CHALMERS

Robust Numerical Wall Functions Implemented in OpenFOAM

– new recommendations for near-wall resolution using low-Reynolds-number turbulence models

JON-ANDERS BÄCKAR

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING IN THERMO AND FLUID DYNAMICS

Robust Numerical Wall Functions Implemented in OpenFOAM

– new recommendations for near-wall resolution using low-Reynolds-number turbulence models

JON-ANDERS BÄCKAR

Department of Applied Mechanics
Division of Fluid Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2016

Robust Numerical Wall Functions Implemented in OpenFOAM
– new recommendations for near-wall resolution using low-Reynolds-number turbulence models
JON-ANDERS BÄCKAR

Thesis for the degree of Licentiate of Engineering 2016:11
ISSN 1652-8565
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2016

Robust Numerical Wall Functions Implemented in OpenFOAM
– new recommendations for near-wall resolution using low-Reynolds-number turbulence models
Thesis for the degree of Licentiate of Engineering in Thermo and Fluid Dynamics
JON-ANDERS BÄCKAR
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology

Abstract

This thesis presents two new numerical wall models for computing Reynolds-Averaged-Navier-Stokes (RANS) equations with low-Reynolds-number turbulence models. The objective is to considerably reduce the total central processing unit (CPU) cost of the numerical simulations of wall bounded flows while maintaining the accuracy of any low-Reynolds-number turbulence model.

When calculating turbulent flow problems, a tremendous speed-up may be achieved by decoupling the solution of the boundary layer from the bulk region by use of a *wall function.* However, most wall functions are quite limited and based on assumptions which are not valid in complex, non-equilibrium flows. A decade ago, the *numerical wall function* was born at the University of Manchester [8], [6], solving boundary-layer-type transport equations across the boundary layer on a separate sub-grid. This approach removed most assumptions, but introduced a strong coupling to the turbulence model, and hence, made it cumbersome to implement and maintain.

The present wall functions solve full momentum and energy equations on a sub-grid, using face fluxes of advection and dissipation to transfer the solution to and from the sub-grid. The innovative use of face fluxes, decouples the wall function from the turbulence models' production and dissipation terms, and hence, makes it general to all low-Reynolds-number turbulence models. It has been tested on channel flow, axisymmetric impinging jet, and backward facing step using Launder-Sharma turbulence model [14]. Compared to low-Reynolds-number calculations, the results show perfect agreement to one-sixth of the computational cost.

Further on, an attempt to update the general recommendation on grid design for low-Reynolds-number turbulence models is also made.

Keywords: Computational Fluid Dynamics, OpenFOAM, impinging jet, numerical wall function, speed-up

Acknowledgements

This research has been funded by the Swedish Energy Agency and Volvo AB. I would like to thank my principal supervisor, Professor Lars Davidson, for his endless source of motivation and for sharing his thorough knowledge of turbulence modelling. Niklas Andersson, Lars-Erik Eriksson, Anders Karlsson, Ragnar Lárusson, Lennart Lundqvist and other colleagues at Chalmers and Volvo are acknowledged for fruitful discussions and for showing interest in the project. Authors of, and contributors to, the following open source softwares are also acknowledged:

- OpenFOAM [**greenshields˙openfoam˙2016**] including their third party packages, but also its community at cfd-online [17] for examples and tutorials
- pyfoam [10] for managing OpenFOAM case set-ups
- tox [13], pytest [12] and pathos [15] used to set-up an efficient testing framework
- perf [9] part of the linux kernel, used for profiling
- git [24], trac [7], zotero [22] for keeping track of code, documentation and literacy
- latex, matplotlib [11], and TikZ [23] for graphics and typesetting

In addition the free software Zoom from RotateRight [19] is ackowledged for their GUI to perf. At last I would like to thank my beloved Anne-Sofie and my wonderful children for all their support during the time of this project.

Contents

1 Introduction

In industry, predictions using simulations increase more and more due to cost and lead-time benefits compared to corresponding experiments. To predict industrial flow problems, which often have a turbulent nature, the most common approach is a Reynolds Averaged Navier Stokes (RANS) simulation together with a turbulence model. Considering accuracy and computational cost, the most important aspect of such simulations is how the boundary layer is treated.

The boundary layer is the fluid layer in the immediate vicinity of a wall, or in other words, where the viscous effect is not negligible. It extends to the fully turbulent regime, and even though it only occupies a smaller part of the flow, this region may account for the majority of the computing time. The reason for this relatively high computational cost is that boundary layer flow properties change at a rate typically two or more orders of magnitude faster than elsewhere in the flow.

These high gradients require a very fine computational mesh in order to be resolved accurately. The family of turbulence models that use this strategy of resolving the boundary layer is called low-Reynolds-number (LRN) models. These models use the same set of equations for all parts of the flow and may be accurate for most types of flows but the resulting equation system converges slowly, especially at high Reynolds number. The turbulence models span from simple mixing-length schemes, through two-equation eddy-viscosity models of different complexity, to second-moment closure models.

To mitigate the slow convergence of the LRN models, the boundary layer and the fully turbulent region may be decoupled, thus acknowledging the different computational requirements for the two regions. One of the most common approach, at least in industry, is the high-Reynolds-number (HRN) model together with a "wall function", which uses a coarse mesh where the first cell layer covers the inner boundary layer including the inner part of the log-layer. Instead of solving partial differential equations on a fine mesh, an analytical expression is used to model the flow in the boundary layer. HRN models with wall functions are often less accurate, and they are also sensitive to the mesh resolution close to the wall.Attempts to analytically integrate the transport equations have also been made but then the drawback is the restrictions on the geometry to allow for analytical integration.

A more advanced way of decoupling the boundary layer from the fully turbulent region is to use a "numerical wall function". This wall treatment can be seen as a hybrid of HRN and LRN modelling where the boundary layer is indeed resolved but with a slightly simplified set of partial differential equations compared to what is used in the rest of the domain.

The first numerical wall function in a RANS context, named UMIST-N, was developed by Gant [8] and Craft *et al.* [6]. They divided the wall-adjacent cells into a sub-grid where simplified RANS equations, using some sort of boundary layer assumptions, were solved, including tangential velocity and turbulent quantities. Furthermore, the pressure gradient was assumed to be constant in the wall-normal direction over the sub-grid, and could hence be interpolated from the main-grid. A Dirichlet condition, with interpolated values from the main-grid's first and second wall-adjacent cells, was set on the boundary of the sub-grid, opposite to the wall, for all solved quantities. The calculated wall shear stress, averaged turbulent production and dissipation terms from the sub-grid were then used to replace the ditto in the main-grid equations that yielded results close to a default LRN solution at computing times an order less in magnitude.

A few studies have investigated variations of the UMIST-N model. Myers and Walters [16] simplified the sub-grid equations even further by using a linear profile for the wall-normal velocity and used the 2-D continuity equation to calculate the stream-wise velocity gradient. The convection was neglected in the turbulence equations. In this way, the 2-D boundary layer equations were reduced to 1-D equations for the tangential velocity and turbulent model quantities. Bond and Blottner [2] proposed a similar model for compressible and transient flow by neglecting convection in all transport equations. Chedevergne [4] also developed a similar 1-D model but implemented it in a unstructured code where the sub-grid only covered the main-grid's wall-adjacent cells from the wall up to the centroid of the main cells. He also included compressibility terms into the model equations. Lastly, Wald [25] tried to adapt the UMIST-N for a second-moment closure turbulence model which gave similar results in accuracy as Craft and Gant reported earlier on an axisymmetric impinging jet. However, Wald also conclude that the model is unstable and chooses to not pursue with other geometries. Even though the processing speed of computers is continuously increasing, the CFD-community is generally far from satisfied with available computing resources whether they act in industry or elsewhere. As described in e.g. Spalart [21] HRN and LRN modelling belong to the simpler variants of methods solving turbulent transport equations. Nevertheless, using these relatively simple models for large and complex problems, the computational resources often sets a limit to what can be done. If the same models are used in design-of-experiments or optimisation loops, the computing resources will obviously always be a limiter to what can be achieved with simulations for the next decades.

With this background it is important, for at least the next decades, to acknowledge and deploy turbulence modelling techniques which offer the best compromise between accuracy and computing requirement. The numerical wall function strategy deployed in RANS modelling has existed at least since 2004 but has not yet been widely adopted by the CFD community despite its excellent features of supplying a sweet-spot between HRN and LRN modelling. One important reason for this is that it is not robust enough as indicated by Wald [25]. Another important reason is the close connection to the turbulence model. To support a turbulence model, earlier numerical wall functions need to implement the model's specific terms, making the implementation and maintenance more awkward.

The purpose of this paper is to make the "numerical wall function" robust and thus more attractive to the CFD community. It includes making an implementation in an openly available and unstructured CFD code and relaxing the dependence on a specific turbulence model. There is also an ambition to increase the mathematical stringency of the numerical wall function, and removing most of the interpolation which is commonly used in previous versions.

The new numerical wall treatment model is built upon the work from Craft *et al.*, but it is a more mathematical stringent wall treatment model which is independent of type of turbulence model. This has been achieved by an innovative use of face fluxes, making a two-way connection between the main- and sub-grid. The implementation has been made

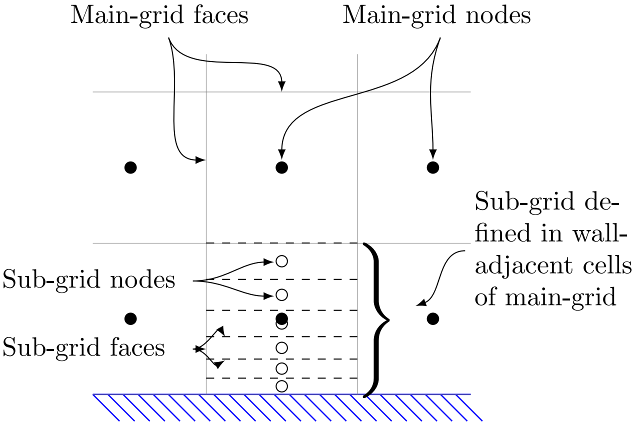


Figure 2.1: *Sketch of sub-grid arrangement, where the height of the sub-grid's cells increases with increasing distance to the wall.*

in the openly distributed CFD package OpenFOAM. It demonstrates the similar excellent compromise between accuracy and computing requirement as the UMIST-N, relaxing the assumptions even further, to only include a constant pressure gradient across the sub-grid, and that the wall-adjacent main-grid cells must be hexagon-like.

2 Sub-grid model

The new numerical sub-grid wall function uses a main-grid covering the whole computational domain but also a sub-grid covering the wall-adjacent walls of the main-grid, see figure 2.1. On the main-grid, the momentum equation is solved in all cells, while the turbulence equations are solved in all cells except for the ones overlapping the sub-grid. On the sub-grid, all transport equations are solved, the pressure gradient is treated as a constant in the wall-normal direction, but updated every iteration with the value taken from the overlapping main-grid cell. To ensure mass balance also on the sub-grid, the mass flux stored at the main-grid's faces is mapped onto the wall-normal faces of the sub-grid, see figure 2.2, and elsewhere the mass flux within the sub-grid is updated to ensure a divergence free mass flow field[1].

There are actually two versions of the present model named the *face flux* and the *wall flux* model. First, the face flux model is outlined, and at the end of section 2.2, describing the implementation, it is explained in what way the wall flux model differ to the face flux model.

The coupling between the main and sub grid uses face flux i.e., fluxes of advection and diffusion over the interface in both directions, see figure 2.4. One advantage with the

[1] For any incompressible flow the divergence of the velocity field equals zero. In OpenFOAM the velocity field is stored in the cell centre, but it also stores the mass or volume flow at the cell faces which is used to calculate the advection terms.

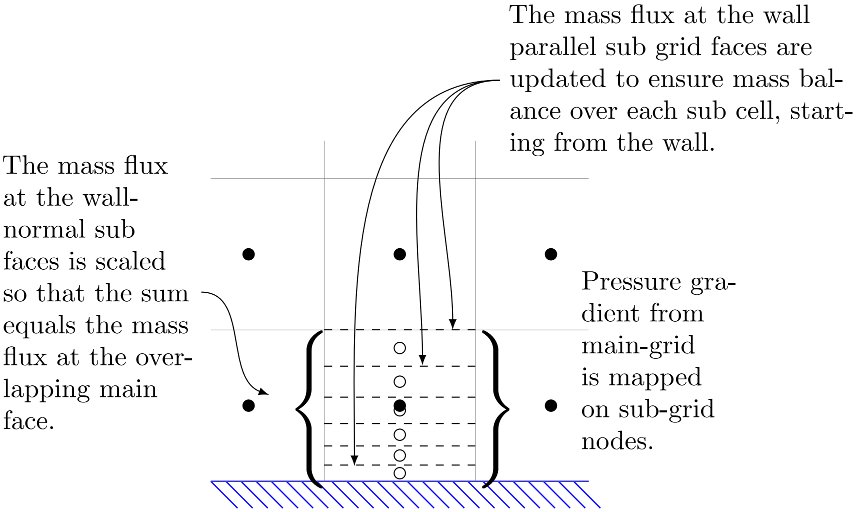

The mass flux at the wall parallel sub grid faces are updated to ensure mass balance over each sub cell, starting from the wall.
The mass flux at the wall-normal sub faces is scaled so that the sum equals the mass flux at the overlapping main face.
Pressure gradient from main-grid is mapped on sub-grid nodes.

Figure 2.2: *Principal sketch how the mapping of mass flux and the interpolation of pressure gradient from the main-grid to the sub-grid is performed.*

face flux model compared to UMIST-N like models is that it is general to all turbulence models as it does not involve the production and dissipation terms which are unique to each and every turbulence model. In other words, earlier numerical wall functions have focused on the source terms of the turbulence equations, while this method only deals with the transport terms of any transport equation.

Other benefits of the present sub-grid models is that continuity is ensured in both main- and sub-grid, and that the only assumption is that the pressure gradient normal to the wall is constant over the sub grid and equal to the pressure gradient in the overlapping main cell.

The sub-grid model has been implemented in the open source package OpenFOAM. The OpenFOAM® (Open Field Operation and Manipulation) CFD Toolbox is a free, open source CFD software package produced by OpenCFD Ltd [18]. It uses a co-located methodology on unstructured polyhedral meshes.

2.1 Governing equations

Full 3D-RANS equations are solved in both the main- and the sub-grid, with the only assumption that the pressure gradient is constant in the sub-grid. Incompressible Reynolds averaged Navier-Stokes in tensor notation reads

$$\overline{u}_{i,i} = 0, \tag{2.1}$$

$$\overline{u}_{i,t} + (\overline{u}_i \overline{u}_j)_{,j} = -\frac{\overline{p}_{,i}}{\rho} + \left[(\nu + \nu_t)(\overline{u}_{i,j} + \overline{u}_{j,i})\right]_{,j} \tag{2.2}$$

temperature, T:

$$T_{,t} + (\overline{u}_i T)_{,i} = \left[\left(\frac{\nu}{\sigma} + \frac{\nu_t}{\sigma_t}\right) T_{,i}\right]_{,i} \tag{2.3}$$

with LRN model Launder-Sharma [14]; turbulent energy, k:

$$k_{,t} + (\overline{u}_i k)_{,i} = \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) k_{,i}\right]_{,i} + P_k - \tilde{\varepsilon} \tag{2.4}$$

and dissipation rate, $\tilde{\varepsilon}$:

$$\tilde{\varepsilon}_{,t} + (\overline{u}_i \tilde{\varepsilon})_{,i} = \left[\left(\nu + \frac{\nu_t}{\sigma_{\tilde{\varepsilon}}}\right) \tilde{\varepsilon}_{,i}\right]_{,i} + c_{\tilde{\varepsilon}1} P_k \tilde{\varepsilon}/k - c_{\tilde{\varepsilon}2} f_2 \tilde{\varepsilon}^2/k + E \tag{2.5}$$

where

$$C_{\tilde{\varepsilon}1} = 1.44, \quad P_k = \nu_t S^2, \quad S \equiv \sqrt{2 S_{ij} S_{ij}}, \quad S_{ij} = 0.5(\bar{u}_{j,i} + \bar{u}_{i,j}) \tag{2.6}$$

and

$$C_{\tilde{\varepsilon}2} = 1.92, \quad f_2 = -0.3 \exp\left(-\min\left[50, k^2 \tilde{\varepsilon}/\nu\right]\right), \quad E = 2\nu\nu_t \left(\overline{u}_{i,jj} \overline{u}_{i,jj}\right) \tag{2.7}$$

The turbulent viscosity is expressed as,

$$\nu_t = C_\mu f_\mu \frac{k^2}{\epsilon}, \quad C_\mu = 0.09, \quad f_\mu = \exp\left(\frac{-3.4}{1 + k^2 \tilde{\varepsilon}/\left(50\nu\right)}\right) \tag{2.8}$$

With HRN model realizable $k - \epsilon$ [20]:

$$k_{,t} + (\overline{u}_i k)_{,i} = \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) k_{,i}\right]_{,i} + P_k - \varepsilon \tag{2.9}$$

$$\varepsilon_{,t} + (\overline{u}_i \varepsilon)_{,i} = \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right) \varepsilon_{,i}\right]_{,i} + c_1 P_k \varepsilon - c_2 \frac{\varepsilon^2}{k + \sqrt{\nu\varepsilon}} \tag{2.10}$$

where

$$C_1 = \max\left[0.43, \frac{\eta}{\eta + 5}\right], \quad \eta = S\frac{k}{\epsilon}, \quad S = \sqrt{2 S_{ij} S_{ij}}, \quad C_2 = 1.9 \tag{2.11}$$

In these equations, P_k represents the generation of turbulence kinetic energy due to the mean velocity gradients, calculated in same manner as standard k-epsilon model. Here the turbulent viscosity takes the form, $\nu_t = C_\mu \frac{k^2}{\epsilon}$, where

$$C_\mu = \frac{1}{A_0 + A_s \frac{kU^*}{\epsilon}}, \quad U^* \equiv \sqrt{S_{ij} S_{ij} + \Omega_{ij} \Omega_{ij}}, \quad \Omega_{ij} = 0.5(\bar{u}_{j,i} - \bar{u}_{i,j}) \tag{2.12}$$

The model constants A_0 and A_s are given by: $A_0 = 4.0, \quad A_s = \sqrt{6}\cos\phi$ where

$$\phi = \frac{1}{3}\cos^{-1}\left(\min\left[1, \max\left[-1, \sqrt{6}W\right]\right]\right), \quad W = \frac{S_{ij} S_{jk} S_{ki}}{\tilde{S}^3}, \quad \tilde{S} = \sqrt{S_{ij} S_{ij}}, \tag{2.13}$$

2.2 Implementation

Mesh

The sub-grid is defined to overlap the wall-adjacent cells in the main-grid, see figure 2.1. The cell sizes are generally non-uniform, and increase in size with increasing distance to the wall using a simple geometric expansion. It is commonly good practice to use a constant and sufficiently small mesh expansion ratio, resulting in a change in mesh spacing which is continuous. However, the concept of having a sub-grid with multiple cell layers in the wall-adjacent cells of the main-grid will lead to mesh size discontinuities at the interface of the main- and sub-grid, in one or the other way. The two cell layers closest to the wall of the main grid are normally close in size for a uniform mesh, but also for a mesh with a moderate expansion ratio. Still, the top cell layer in the sub-grid facing the second cell layer in this main-grid is far smaller in size, dependent on number of cell layers in the sub-grid. To investigate the magnitude of this possible discretization problem three different mesh strategies for the main-grid are investigated labelled *cMesh*, *dMesh*, and *eMesh*.

The *continuous* cMesh is a grid which together with a sub-grid resembles the grid normally used for LRN models and having a uniform mesh expansion ratio, avoiding mesh size discontinuities for most entities, see figure 2.3a. Nevertheless, there is a cell size discontinuity in the main-grid at the location of the interface. This discontinuity is experienced by the main pressure and main velocity as these are the only entities using the relatively large wall-adjacent cells of the main-grid.

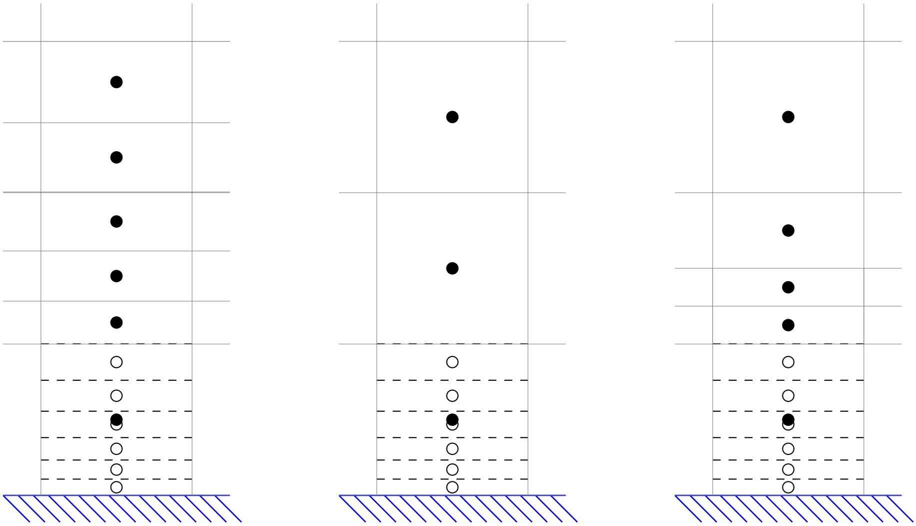
The topology of the *discontinuous* dMesh grid is the same as is typically used with HRN models which often have a uniform mesh spacing. A large mesh size discontinuity will be the result at the interface when going between the top cell in the sub-grid and the cell next to the dMesh's wall cell, see figure 2.3b. For this mesh all solved fields experience a discontinuity except the pressure and the velocity on the main-grid.

As a last grid strategy the *extended* eMesh will be investigated, which is a dMesh where the cell layer above the wall cell layer has been iteratively cut in half until the resulting layer closest to the sub-grid is of equal size as the top layer of the sub-grid, see figure 2.3c. The advantage with this mesh is that the total number of cells can be kept almost as low as in the dMesh at the same time as discontinuities in cell size can be avoided for most entities.

Face flux model

In both grids boundary conditions for all entities are set-up as in a standard wall treatment of the used LRN model with one exception. At the *interface boundary* i.e., at the boundary of the sub-grid which faces the bulk flow, all entities solved for is set to a zero-flow condition[2]. New interface fluxes are then added, see below.

[2] The zero-flow condition is implemented as a derivative of a homogeneous Neumann condition i.e., *zeroGradient* condition in OpenFOAM, where also the advective flux is set to zero.



(a) *cMesh* (b) *dMesh* (c) *eMesh*

Figure 2.3: *Sketch of the topology of the c-, d- and e- main meshes visualizing the difference in cell size discontinuities. Observe that the sub-grid cells, represented by non-filled circles at the node centres and dashed lines at the faces, overlap a larger main cell whose node centre is displayed with a filled circle.*

Continuity As the pressure is not solved on the sub-grid, the continuity needs to be achieved in another way. The mass flux, stored at the wall-normal faces of the sub-grid, is scaled with the corresponding cell faces of the main-grid, see figure 2.2. This is done so that the sum of the mass flux through a group of sub faces equals the mass flux to the overlapping main face. If the number of layers in the sub-grid is larger than one, there are also wall-parallel faces in the sub-grid not overlapping any face of the main-grid. For these sub faces the mass flux is updated to ensure that the sum of the mass flux per cell is zero, which is the same as a divergence free velocity field. The pressure gradient in the sub-grid is calculated by mapping it from the main-grid as a constant per cell. The mapping of the mass flux, as well as the mapping of the pressure gradient, is done before the momentum equation is assembled every iteration. In this way, continuity is secured in all cells also in the sub-grid.

Discretization at the interface The coupling in the main-grid between the wall cell and its neighbour away from the wall are decoupled and replaced by couplings to and from the sub-grid. The couplings between the main- and sub-grid of all entities solved for are handled with face fluxes. These fluxes can be divided into advection and diffusion, which is implemented as follows for the diffusion:

$$S_p = [(1-w)\nu_s + w\nu_m]\frac{1}{\delta y_s + \delta y_m}|S_f| \tag{2.14}$$

$$S_u = S_p\phi \tag{2.15}$$

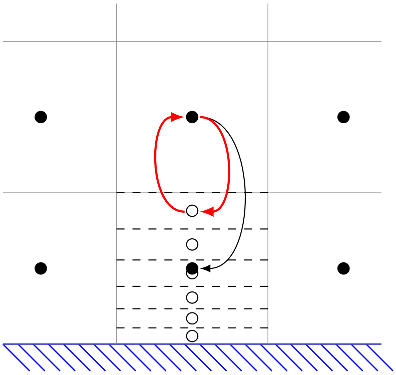


Figure 2.4: *Principal sketch of the coupling in the face flux model where all entities are coupled with face fluxes: to and from the sub-grid's top node,* — *, and within main-grid,* — *. Observe the lack of coupling from the main wall cell which has been decoupled.*

Here, the δy_s and δy_m are the distances from the interfacing face, with face normal area of S_f, to the closest node in the sub- and the main-grid, respectively, see also figure 2.5. The w is a weighting factor calculated as $w = \delta y_s/(\delta y_s + \delta y_m)$, and ϕ is the node value, of the entity in question, next to the interfacing face in the grid not currently being solved for. In this implementation the cross-terms of the diffusion have been neglected, see equation 2.2, as these in general are small compared to the diffusion main term.

For the advection, using the central-differencing scheme, the sources become, superscript refer to main or sub, respectively:

$$\begin{aligned} S_p^s &= (1-w)(\overline{u}_f \cdot S_f) && (2.16)\\ S_u^s &= -w(\overline{u}_f \cdot S_f)\phi^m && (2.17)\\ S_p^m &= -w(\overline{u}_f \cdot S_f) && (2.18)\\ S_u^m &= (1-w)(\overline{u}_f \cdot S_f)\phi^s && (2.19) \end{aligned}$$

where $\overline{u}_f$ is the face velocity, interpolated from the main nodes on each side of the face, ϕ means as before the node value of the entity in question.

Every iteration is started by solving all the transport equations in the sub-grid followed by the same in the main-grid. All source terms are calculated using current values from current grid, together with the earlier values calculated from the other grid.

Gradient at the interface The gradient in a cell is calculated from the face values of the same cell using the Green-Gauss theorem. Thus, to calculate the gradient correctly it is enough to interpolate the cell values to face values. For interpolation to the interface between the main- and sub-grid the values of the interface adjacent cells in both the sub-grid and the main-grid should be used. One example containing gradients are the cross-terms of the diffusion, which in this implementation have been neglected, see previous paragraph. However, the turbulent production includes the gradient of the velocity and must be calculated also at the interface. For the sub-grid one can do this directly by

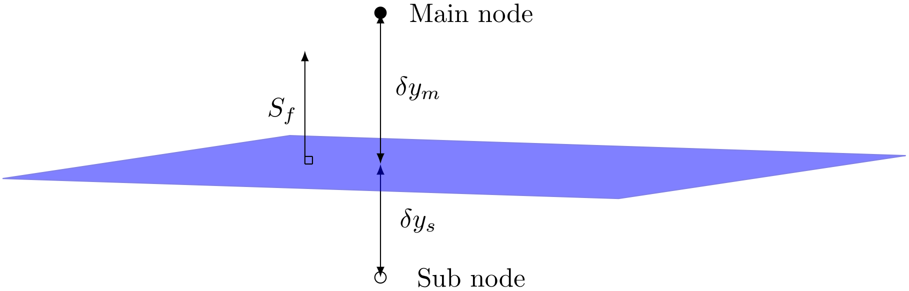


Figure 2.5: *A sketch of the relation between a face and the related nodes on each side at the interface between the main- and the sub-grid.*

setting a Dirichlet condition of the boundary representing the interface equal to the interpolation of the interface adjacent cell of the sub-grid and the cell of the main-grid on the other side of the interface. As there is no boundary at the position of the interface in the main-grid cell, the wall-adjacent cells are temporarily used and set to values so that a cell to face interpolation gives the same value as calculated for the interface boundary of the sub-grid. Directly after the turbulent production field has been calculated the wall-adjacent values in the main-grid are reset to its original values.

Turbulent viscosity in wall cell Earlier it has been mentioned that the turbulent entities are not calculated in the wall-adjacent cells of the main-grid. However, in the momentum equation the turbulent diffusion is needed to be calculated also for these cells. This is done by cell averaging each turbulent entity from the overlapping sub-grid cells. From these averaged values the turbulent viscosity is calculated according to the definition of the specific turbulent model.

Wall flux model

In the wall flux model the coupling in the main-grid from the wall cell to its neighbour further from the wall is not decoupled. Instead, a slip boundary condition for the velocity is set at the wall of the main-grid and the coupling from the sub-grid is in the form of wall shear stress, similar to what is used in UMIST-N, see figure 2.6. The added source terms to the momentum equation in the near wall cells of the main-grid are:

$$S_u = -\nu_s \frac{1}{\delta y} |S_f| \overline{u}_s, \tag{2.20}$$

where ν_s is the sub-grid viscosity, δy is the distance from the wall to the near-wall node in the sub-grid, $|S_f|$ is the cell face area at the wall and $\overline{u}_s$ is the velocity in the near-wall node of the sub-grid. The reasoning behind the use of the wall shear stress as source is that it should equal the sum of the turbulent and viscous shear stress across the boundary layer. However, this sum is neither constant nor equal to the wall shear stress except for the very near-wall part. A similar technique outlined here for the velocity is also used for the temperature.

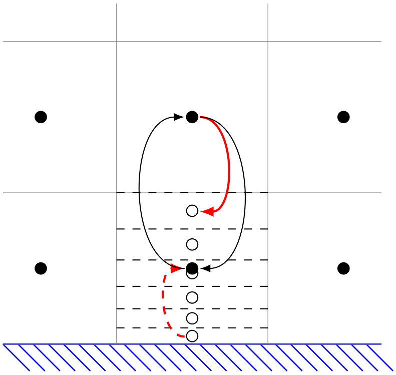


Figure 2.6: *Principal sketch of the velocity coupling from the main- and sub-grid for the wall flux model. The wall shear stress calculated in the sub-grid's wall cell,* – – *, is added to the main-grid's momentum equation as a wall flux. From main- to sub-grid the velocity is coupled with face flux,* — *, as in the face flux model. Within the main-grid the face fluxes,* — *, are untouched. Instead, the Dirichlet wall boundary condition in the main-grid is replaced by a slip condition.*

Reflection upon chosen implementation

Another way of implementing the face flux version of the sub-grid model might be to still use two grids but instead of having a main- and an overlapping sub-grid, one grid could be devoted to continuity and another for the rest of the Navier-Stokes equations. An advantage of using two grids, both covering the whole domain, would be to be able to use all the different discretization types already implemented in OpenFOAM without the need to implement a single line of code regarding discretization. Also, to calculate a correct gradient at the interface would come for free. The disadvantage would be the extra storage needed, and possibly the extra computational effort to transfer values back and forth between the both grids, which would be a small prize to pay for a simpler implementation. A third alternative would be to implement it as a “lazy“ geometrical multi-grid pressure solver which do not solve the pressure in all cells but agglomerates the thin near-wall cells and solve them in groups. Complete continuity is ensured by utilizing the mass flux and pressure mapping outlined in earlier paragraph about continuity in section 2.2. Probably these alternative implementations make the model even more robust as the boundary layer and the bulk now are solved implicitly for all entities except pressure.

2.3 Set-up of test cases and used output

Here the set-up is outlined of the test cases used in the verification and validation of the sub-grid models. For all calculations performed using an LRN model the mesh density is chosen to give an y^+ value of unity with a tolerance of ±15%, but for most cases the tolerance is lower than ±5%. For all solved entities, the central differencing scheme, CD, is used for laminar set-ups, and the linear upwind scheme, LUD, is used for turbulent set-ups.

Channel flow

The channel flow is a simple test case with neither wall-normal advection nor non-zero pressure gradient. Fully developed flow is ascertained by using periodic inlet and outlet boundary condition, together with a pressure gradient as driving source.

Axisymmetric impinging jet

The test case chosen for most verification and validation is the axisymmetric semi-confined impinging jet on a plane surface [1], [5], see also appendix A[3]. The distance between the nozzle and the impinging wall is set to 2 nozzle diameters, the outer radius of impinging wall is set to 10 nozzle diameters, and the half angle of the sector is set to 1 degree. The boundary opposite of the impinging wall is set to symmetric boundary condition. Here, fully developed flow is ascertained by feeding the inlet from a pipe with periodic streamwise boundary conditions.This impinging jet involves both advection and diffusion, and has a non-uniform pressure gradient. Hence, we can here verify if these features are implemented correctly.

For comparisons the skin friction, C_f, and the Nusselt number, Nu, are used. The skin friction is defined as $C_f = -2\tau_w/(\nu U_{in}^2)$, where τ_w is the wall shear stress, ν is the viscosity, and U_{in} is the velocity at the inlet. The Nusselt number is defined as $Nu = \nabla TS_f/(T_w - T_{in})/D$, where ∇TS_f is the wall-normal gradient of the temperature, T_w and T_{in} are the temperatures at the wall and inlet, respectively, and D is the inlet diameter. When an entity itself needs to be highlighted it is plotted along a line orthogonal to the impinging wall at a given distance from the symmetry line.

Backward facing step

The backward facing step case is a good complementary test case as it involves boundary layer separation which is a difficult feature to model with a RANS turbulence model. Here it is set-up in 2D where the full channel height equals six step heights, the length of the channel upstream and downstream of the step equals five and 20 step heights, respectively, see appendix B[4]. All boundaries except the inlet and outlet are modelled as walls. The inlet is fed with fully developed flow from a channel with periodic streamwise boundary conditions. At the outlet homogeneous Neumann conditions are used. The Reynolds number, Re_H, is based on max velocity at the inlet and the step height.

2.4 Verification and Validation

Verification is the procedure of checking whether a system is built correctly, while validation is the procedure of checking that a system fulfils its purpose. Here, verification is used to affirm that certain simulations using the sub grid model, gives more or less identical result

[3] The symmetry line is at $x = 0$ and the impinging wall at $z = 0$

[4] The step is located at $x = 0$ and the bottom wall downstream the step is located at $z = 0$

as a simulation using the default wall treatment i.e., a LRN model. During validation several aspects is tested: accuracy, robustness and speed-up.

Verification

During the verification only grids of cMesh, see figure 2.3a, type is used to mimic LRN with standard wall treatment as close as possible. For the same reason the CD advection scheme would have been used also for turbulent set-ups if it where not due to robustness problems. Instead, the LUD scheme is used for all turbulent set-ups. Recall that the interface advective fluxes are computed using CD. Except for differences in advection scheme a correct implementation would give identical result between the sub-grid model and a default wall treatment of any low Reynolds numbers turbulence model under the following premises:

- the pressure gradient is constant cross the domain covered by the sub-grid which is tested in the turbulent channel flow
- the sub-grid have only one cell layer which is tested for the laminar set-up of the impinging jet

Face advection and diffusion The exchange of fluxes to and from the sub-grid for all entities need to be verified. First the diffusion is tested using a turbulent channel flow, $Re_\tau = 590$, using the Launder-Sharma model. But as turbulence is included, the correction of gradients used in the production term at the interface is also verified. Recall that the the turbulent production is proportional to the square of the velocity gradient.

To test for a combination of advection and diffusion, the laminar axisymmetric impinging jet on a plane surface is used, $Re_D = 450$. This test case also includes a non-uniform pressure gradient, and hence all of these features are tested simultaneously, but must be tested with a single layer in the sub-grid.

Discretization scheme As advection scheme at the interface of the main- and sub grid, only central differencing has been implemented, even though central differencing is known to be unstable for convection, diffusion ratio exceeding two. Thus, to confirm that it is good enough to use central differencing at the interface independent of what advection schemes used elsewhere this has to be tested. To magnify differences a laminar axisymmetric impinging jet case with $Re_D = 450$, having a fairly coarse mesh, is used to compare default wall treatment with the sub face flux model having one single layer for two upwind advection schemes in the main- and sub-grid equations; upwind and linear upwind. A successful test would show that both the default wall treatment and the sub face flux model is affected of the advection scheme in a similar way giving almost identical results of e.g. skin friction and Nusselt number.

Validation

The purpose of the validation is to investigate that the sub-grid model is robust and gives a significant speed-up while delivering almost the same accuracy as the low-Reynolds-number wall treatment. In addition, default LRN is investigated using a similar effective mesh i.e., main- and sub-grid, as used with the sub-grid models. Ideally, any model should be robust to model parameters. Here, two parameters are used to describe the mesh density of the non-standard meshes: the standard y^+ at the wall and the y^+ at the interface between the main- and sub-grid, labelled y_i^+, see e.g. figure 2.3. The y_i^+ parameter refers to the distance from the wall to the interface between the main- and sub-grid in number of wall units i.e., y^+. For most validation tests the turbulent axisymmetric impinging jet is used, and in general the robustness is tested first and then the results of converged solutions are compared to standard LRN. In the paragraph about near-wall resolution also the backward facing step is analysed briefly.

Sensitivity of mesh strategies As mentioned earlier in section 2.2 the concept of having a sub-grid overlapping the wall adjacent wall cells of the main-grid will create cell size discontinuities in one or the other way. Here the two non-standard mesh strategies, dMesh and eMesh, are investigated for the sub-grid models at a relative high y_i^+-value to magnify errors from the discontinuity in cell sizes.

Sensitivity of y^+ Instead of using number of cell layers in the sub-grid as parameter, a specific y^+-parameter has been used. It is important to remember that this parameter is an input parameter governing the mesh density and is not identically equal to the y^+ calculated from the wall shear stress. As stated earlier the difference between targeted and actual y^+ does not differ more than ±5% for most LRN calculations. The y^+-parameter governs the size of the wall-adjacent cell and should be equal or lower than unity for any LRN model. Here the sensitivity of y^+ is compared between default wall treatment and the sub models.

Sensitivity of different y_i^+ values The y_i^+-parameter is a measure of the distance, in wall units, from the wall to the interface between the main- and the sub-grid i.e., the y^+ value for the centre of the wall cell of the main-grid equals the half of y_i^+. The measure is also used for default LRN modelling to define the location of the discontinuity in cell size when a non-standard mesh is used. As the mesh, or model, parameter y_i^+ should be proportional to the boundary thickness this may be difficult to estimate in advance, and thus it is important that the model is robust towards this parameter.

Recommendations for near-wall resolution In [3] recommendations on grid design are given for both HRN and LRN modelling, with a special section on near-wall resolution. In short the criteria for LRN models are:

- y^+ at wall adjacent cells should be less than four but preferably around unity
- mesh size discontinuities should be avoided particularly in regions of large changes
- the number of cells within the distance of $y^+ = 20$ from the wall should be between five or ten dependent on Reynolds number

In section 3.2 an attempt to give more precise recommendations for the tested geometries is made.

Trade-off between accuracy and speed-up The very objective of this work is to optimise the trade-off between accuracy and computational speed. The HRN approach represents the fast, low precision prediction, and the LRN approach represents the slow, high precision prediction. In the LRN approach many more cells are used compared to what is used in HRN and thus has a higher computational cost. To understand if this cost, and also accuracy, is associated with a particular part of the Navier-Stokes equations i.e., continuity or transport of momentum or heat, LRN equipped with an eMesh is also tested, *LRNe*. The main difference between LRNe and the face flux model is if continuity is solved for in the region represented by the sub-grid used in the face flux model[5].

Thus, accuracy comparison is done between standard LRN, LRNe, the face- and wall flux models applied on the impinging wall and pipe wall, where the target $y^+ = 1$. The computational time for all of these model are compared to the computational time of a HRN model.

3 Results

3.1 Verification results

For very simplified set-ups the correctness of the implementation is confirmed.

Verification of face advection and diffusion

In figure 3.1 the results from default LRN and from the face flux and the wall flux models are compared. Already by observing the results in figure 3.1a we can tell that the diffusion scheme has been implemented correctly, as the plots match perfectly. That also the turbulent entities are plotted on top of each other in figure 3.1b demonstrates that also the velocity gradients at the interface have been corrected properly before the turbulent production is evaluated in the cells next to the interface in the main-grid[1].

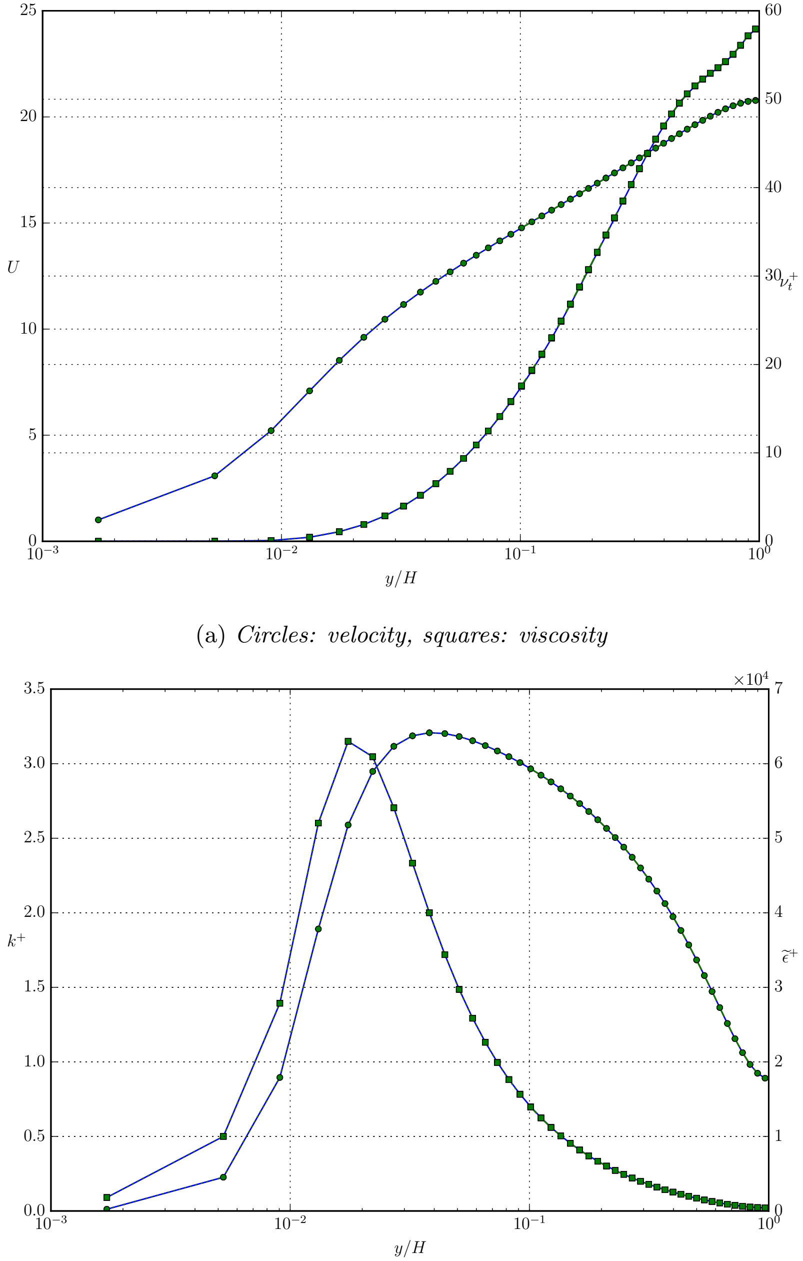
The verification of the CD advection scheme displayed in figure 3.2 shows the same flawless agreement for both skin-friction and heat transfer as earlier presented for the diffusion scheme.

Discretization scheme

In figure 3.3 the results from different advection schemes are plotted for both default LRN and for the face flux model. For both entities, skin-friction and heat transfer, it is clear that the results are affected much more by changing advection scheme compared to using default LRN or the face flux model. This makes it plausible that it is enough to only implement a single scheme for the advective face flux, the central differencing scheme, and that the differences using other schemes will be relatively small. However, it should

[5] The other difference is that the face flux model uses an explicit coupling between the entities solved for in the two grids.

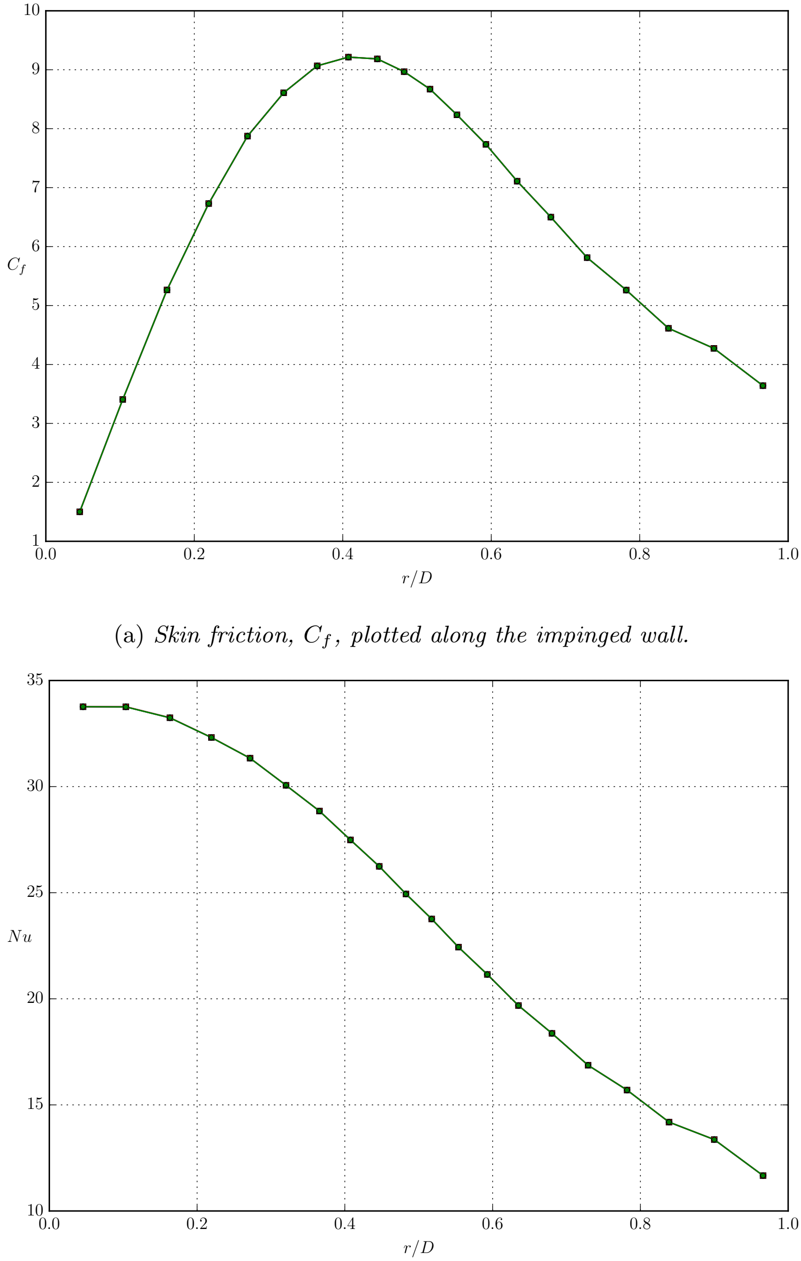
[1] Recall that the turbulent production is proportional to the square of the velocity gradient.



(a) *Circles: velocity, squares: viscosity*

(b) *Circles: turbulent production, squares: turbulent diffusion*

Figure 3.1: *Turbulent channel flow at $Re_\tau = 590$ comparing* — *: default wall treatment with the use of the sub-grid models, having 16 cell layers in the sub-grid;* — *: face flux and* — *: wall flux.*

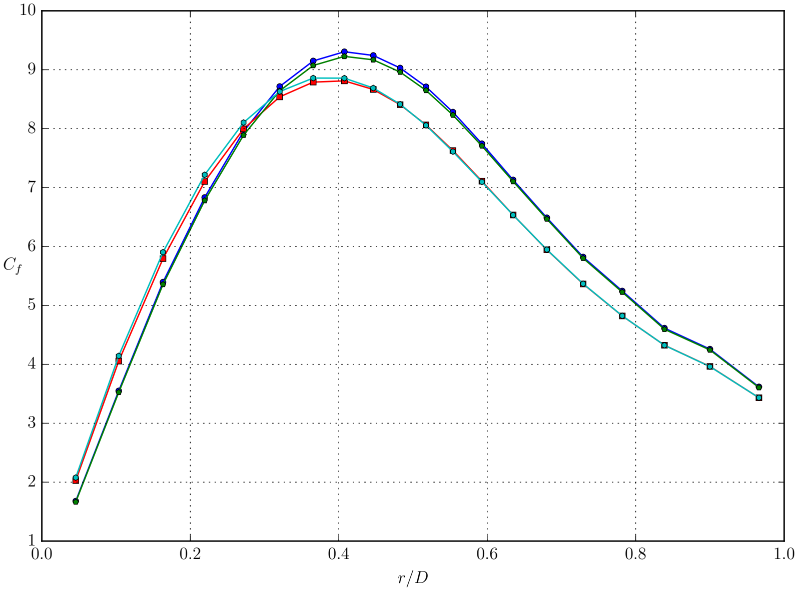


(a) *Skin friction, C_f, plotted along the impinged wall.*

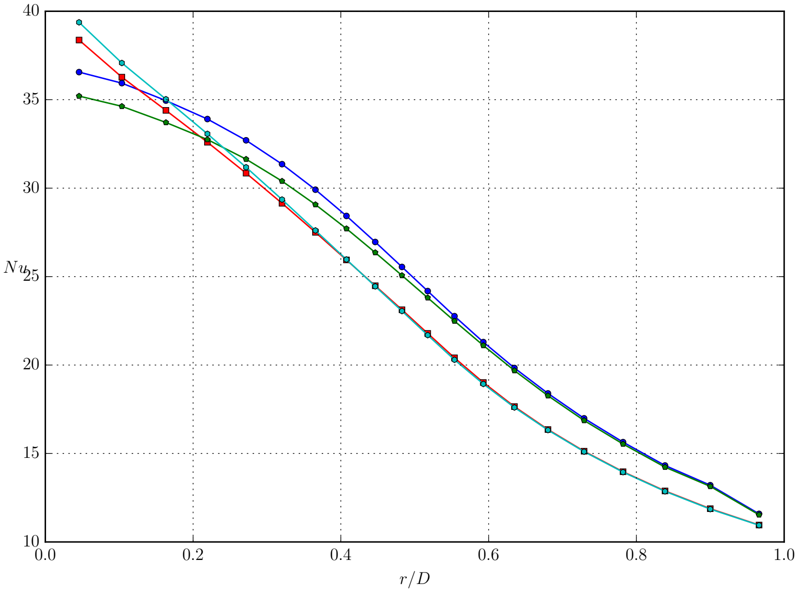
(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.2: *Laminar semi-confined axisymmetric impinging jet at $Re_D = 450$ comparing* 🟥 *default wall treatment with the use of the sub-grid models, having one single cell layer;* 🟦 *face flux and* 🟩 *wall flux.*

be noted that the largest difference between default LRN and the face flux model is seen for the LUD scheme in the heat transfer close to the impinging point, see figure 3.3b.



(a) *Skin friction, C_f, plotted along the impinged wall.*



(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.3: *Laminar semi-confined axisymmetric impinging jet at $Re_D = 450$ comparing differences between default wall treatment, LRN, and the face flux model, FF, having one single cell layer, using different advection schemes; upwind, UD, and linear upwind, LUD. The combinations are represented as follows;* ▬ *: LRN-UD,* ▬ *: FF-UD,* ▬ *LRN-LUD, and* ▬ *: FF-LUD.*

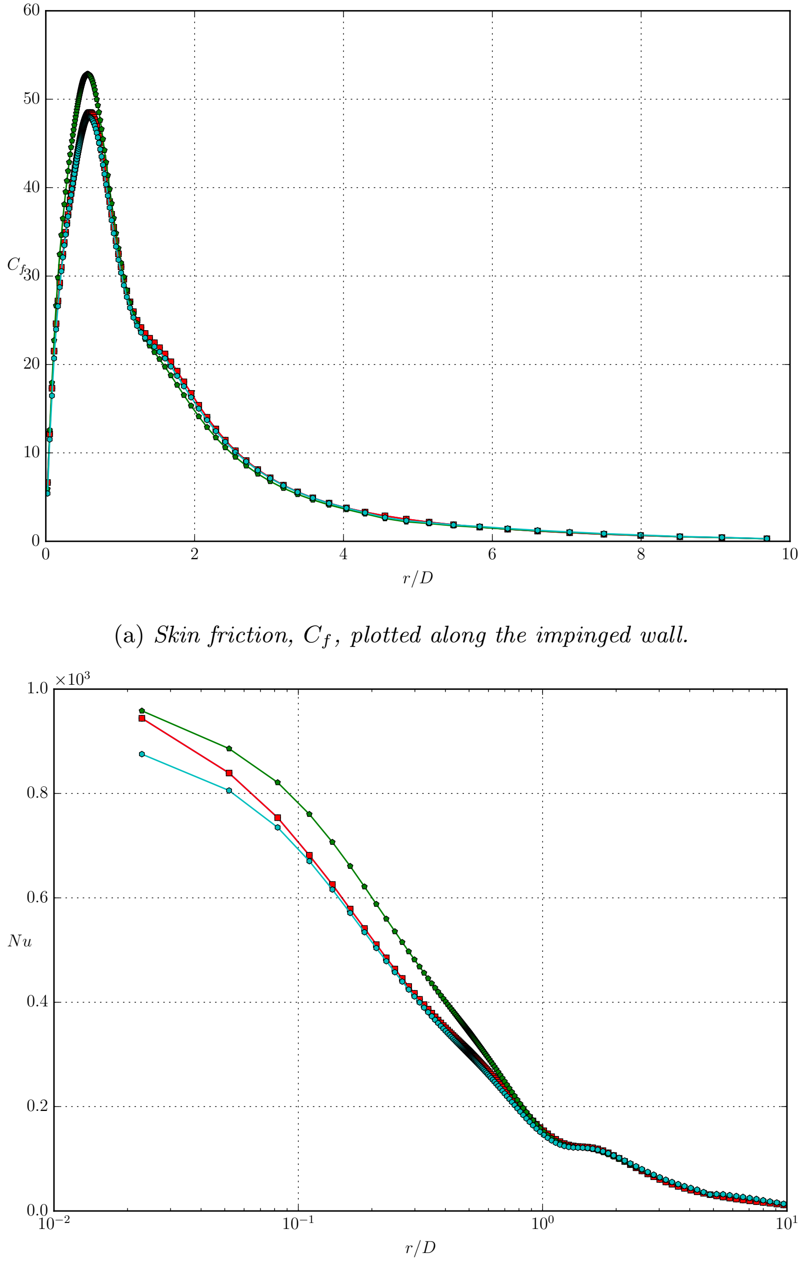
3.2 Validation results

The usefulness of the face flux and wall flux model is investigated regarding robustness and the trade-off between accuracy and speed-up for different set-ups.

Sensitivity of mesh strategies

For the face flux model results are plotted in figure 3.4 comparing the dMesh and eMesh with results from the standard LRN model. The results from the eMesh is very close to the results from default LRN while the results from the dMesh differ significantly. Thus, the discontinuity in cell size leads to severe discretization error. By using the eMesh, which is a refined dMesh, these errors can be almost completely mitigated and still reduce the total number of cells to 7081 compared to 10643 for the cMesh used in the default LRN.

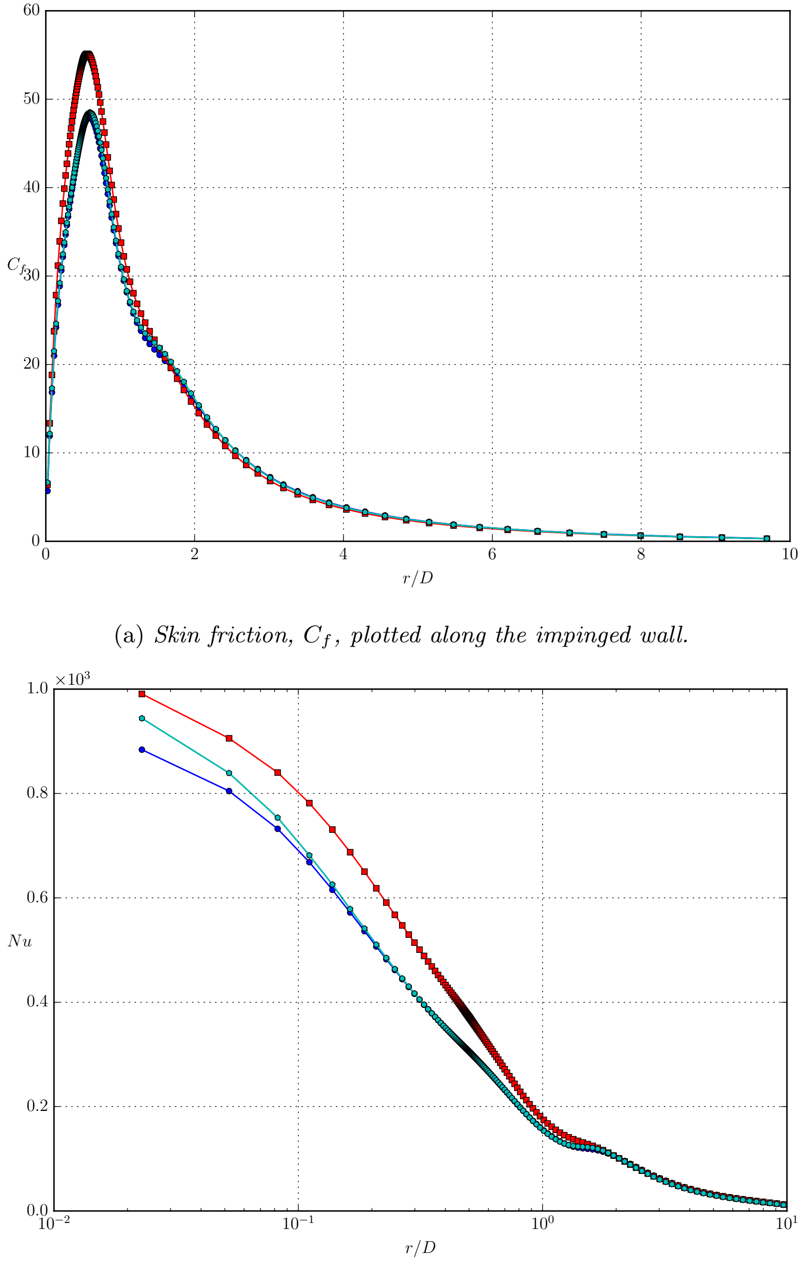
The comparison of dMesh versus the eMesh for the wall flux model displays the same advantages for the eMesh as shown for the face flux model, see figure 3.5. The displayed errors for the dMesh is however slightly larger for the wall flux model than for the face flux model, cf figure 3.4 and figure 3.5. This indicates that the face flux model is slightly less sensitive for cell size discontinuities. Based on the results above the sub-grid models will only use the eMesh in the rest of the evaluation.



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.4: *Launder-Sharma model at $Re_D = 23000$ comparing;* — *: default wall treatment,* — *: face flux model using the dMesh,* — *: face flux model using the eMesh. The interface if the non-standard grids was located at $y_i^+ = 100$.*



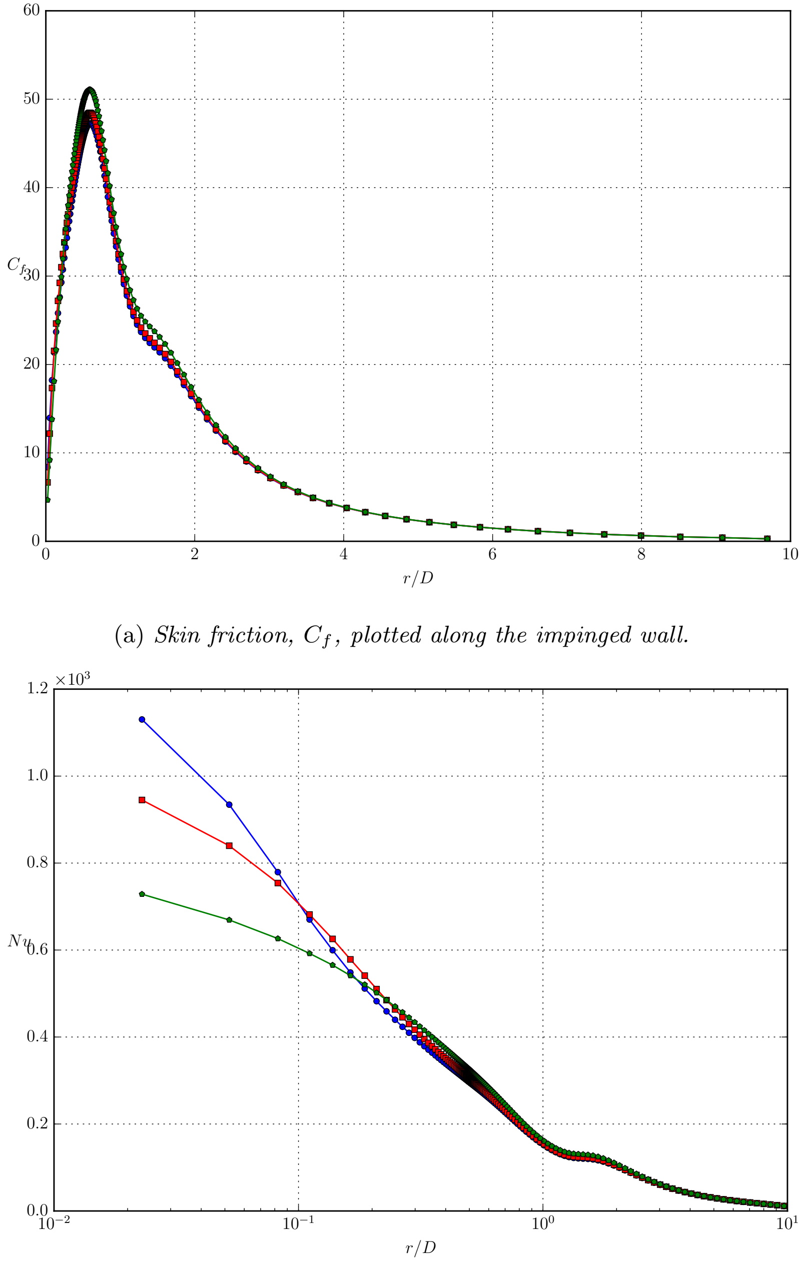
(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.5: *Launder-Sharma model at $Re_D = 23000$ comparing; — : default wall treatment, — : wall flux model using the dMesh, — : wall flux model using the eMesh. The interface in the non-standard grids was located at $y_i^+ = 100$.*

Sensitivity of y^+

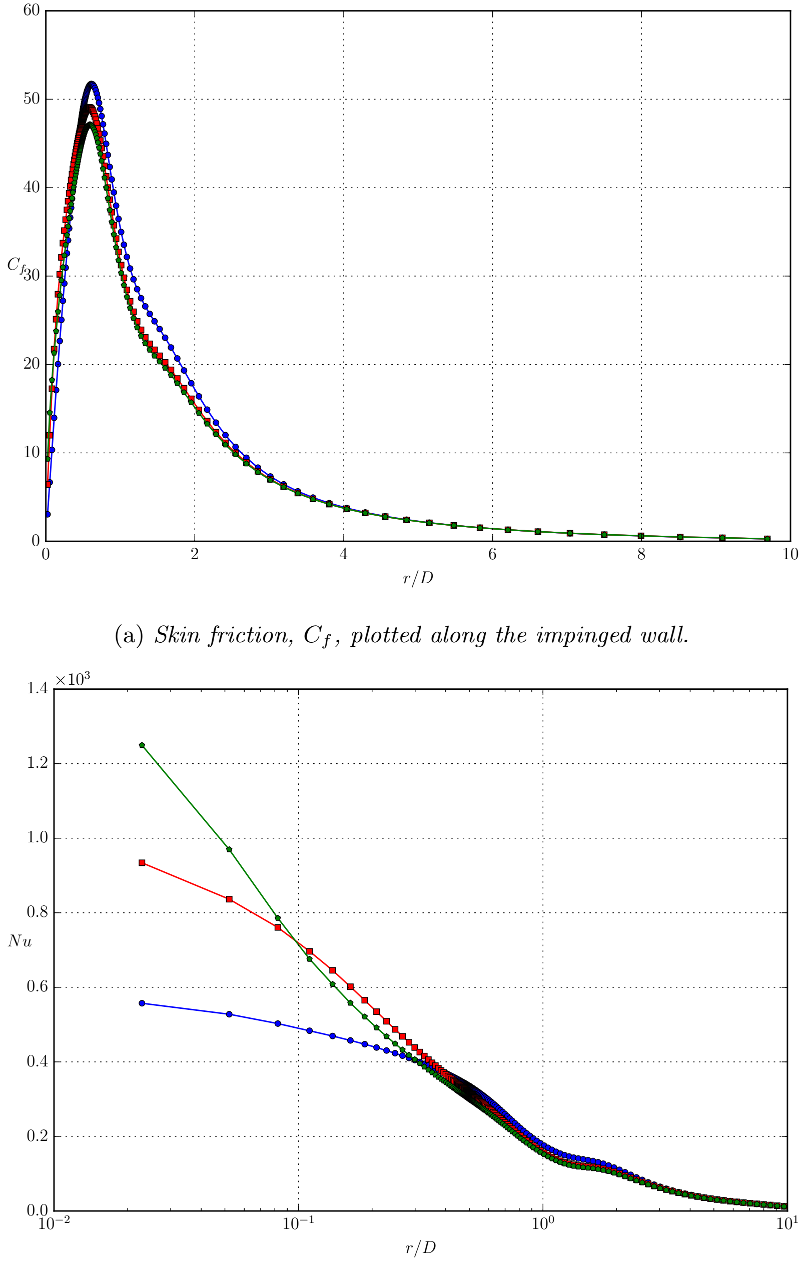
In figure 3.6 and 3.7 the results of different y^+ values at the impinging wall is displayed for the default LRN where the latter uses an eMesh. If y^+ is below unity the skin-friction results seem to be grid independent as expected while the heat transfer might need an even lower value to reach grid independence. The same comparison is shown for the face flux model in figure 3.8 and for the wall flux model in figure 3.9. Also for these models the rule of an y^+ below unity seems fit for an almost mesh independent solution regarding the skin-friction. The correlation between heat transfer and y^+ follows the same trend for the sub-grid models as for default LRN plotted earlier. The observant reader may have noticed that the default LRN with a continuous mesh used a smaller range $y^+ \in [0.5, 2]$ than what was used with the eMesh and the sub-grid models, $y^+ \in [0.3, 3]$, cf figure 3.6, 3.7, 3.8, and 3.9. This is due to the standard LRN cases at the wider span did not converge which prove the eMesh and the sub-grid models to be more robust in this sense.



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

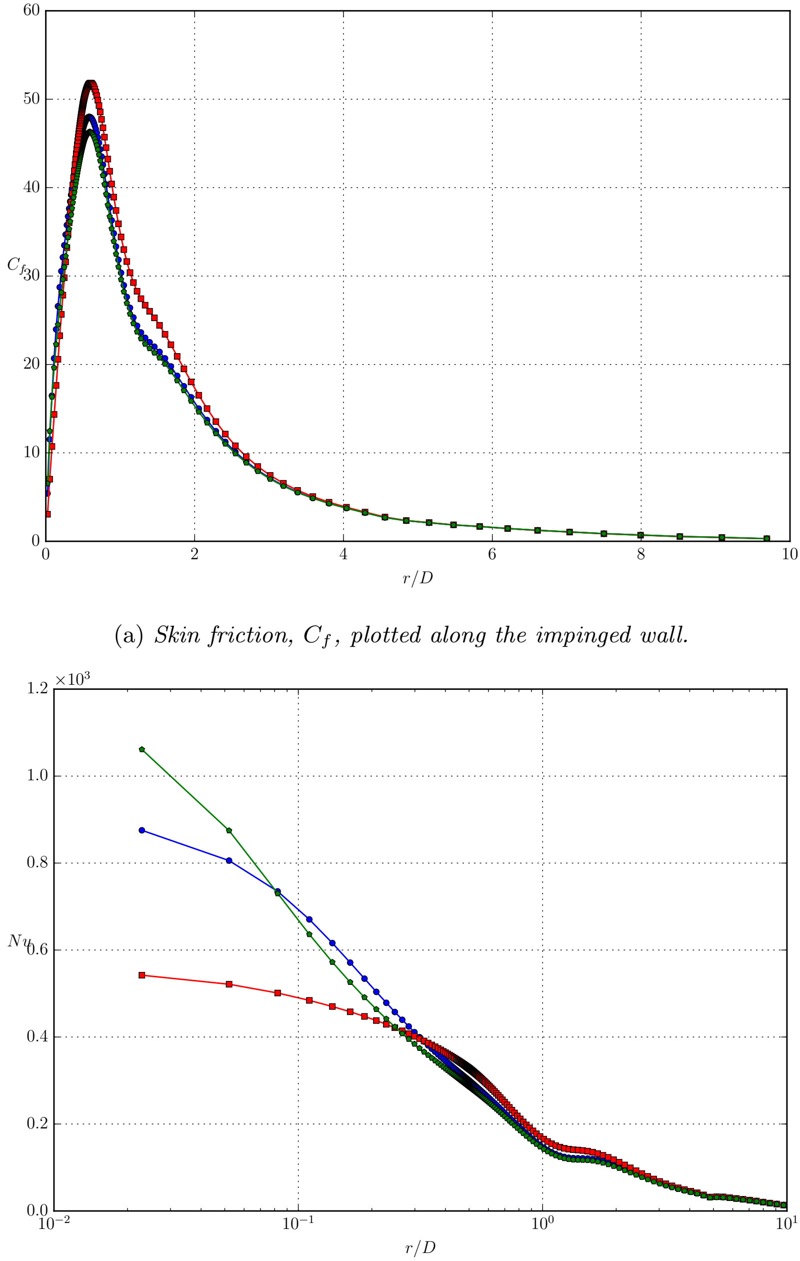
Figure 3.6: *Turbulent semi-confined axisymmetric impinging jet using Launder-Sharma model at $Re_D = 23000$ comparing number of cell layers resulting in different y^+ for default LRN;* ━━ *: $y^+ = 0.5$,* ━━ *: $y^+ = 1$,* ━━ *: $y^+ = 2$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

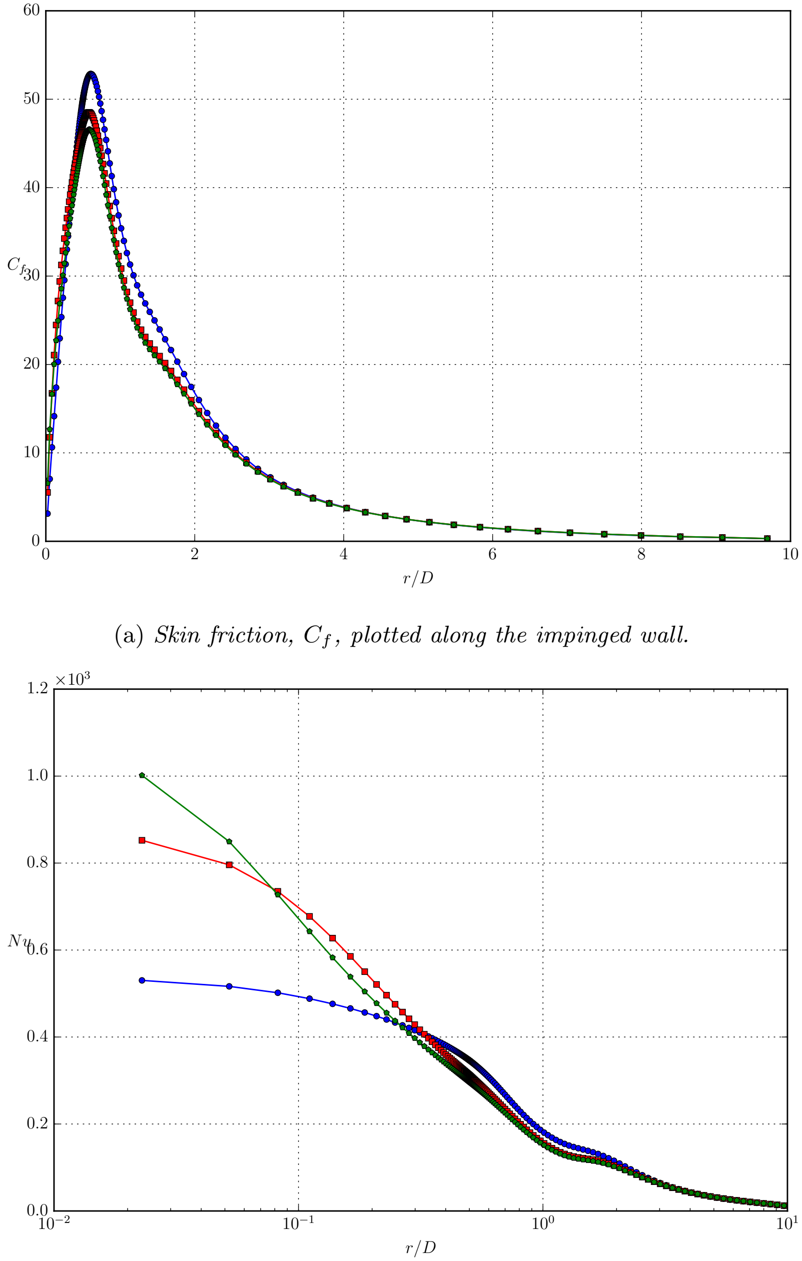
Figure 3.7: *Turbulent semi-confined axisymmetric impinging jet using Launder-Sharma model and an eMesh with $y_i^+ = 200$ at $Re_D = 23000$ comparing number of cell layers resulting in different y^+ for default LRN;* 🟩 *: $y^+ = 0.3$,* 🟥 *: $y^+ = 1$,* 🟦 *: $y^+ = 3$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.8: *Turbulent semi-confined axisymmetric impinging jet using Launder-Sharma model at $Re_D = 23000$ comparing number of cell layers resulting in different y^+ for the face flux sub-grid model, $y_i^+ = 100$; 🟩 : $y^+ = 0.3$, 🟦 : $y^+ = 1$, 🟥 : $y^+ = 3$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.9: *Turbulent semi-confined axisymmetric impinging jet using Launder-Sharma model at $Re_D = 23000$ comparing number of cell layers resulting in different y^+ for the wall flux sub-grid model, $y_i^+ = 200$; — : $y^+ = 0.3$, — : $y^+ = 1$, — : $y^+ = 3$.*

Sensitivity of different y_i^+ values

In table 3.1[2,3] the grid sizes and the required wall time for convergence are summed up for most of the set-ups. A dash in the table represents a set-up which did not converge. The number of dashes in table 3.1 gives a general idea of the robustness of the different wall treatments. Standard LRN with eMesh proved to be the most robust and the wall flux model the least. An interesting note is that no set-up converged for the combination of $Re_D = 71000$ and $y_i^+ = 100$, nor for $Re_D = 220000$ and $y_i^+ = 700$. However, nothing remarkable is visible in these grid topologies compared with the other ones used, cf appendix A.2, where the part of the grids close to the symmetry line is displayed. The reason for not plotting the effective mesh i.e., main- and sub-grid, is that the sub-grid is so dense that the cell layers could not be distinguished from each other. For the two lower Re_D-numbers a smaller y_i^+-value correlates with smaller meshes and lower computing timings but for $Re_D = 220000$ the opposite trend is seen, see table 3.1

Re_D	y_i^+	**LRN eMesh**		**Face flux**		**Wall flux**	
		#Cells	Time [s]	#Cells	Time [s]	#Cells	Time [s]
23000	100	6533	423	6670	368	–	–
	300	6944	471	7081	372	7081	296
	500	7355	497	7492	403	–	–
	700	7629	518	7766	428	–	–
71000	100	–	–	–	–	–	–
	300	9857	1092	10022	1052	10022	877
	500	10187	1142	10352	1082	10352	901
	700	10352	1165	10517	1097	10517	917
220000	100	14387	3867	14582	3195	14582	3223
	300	14192	3810	14387	3100	14387	3083
	500	14192	3802	–	–	–	–
	700	–	–	–	–	–	–

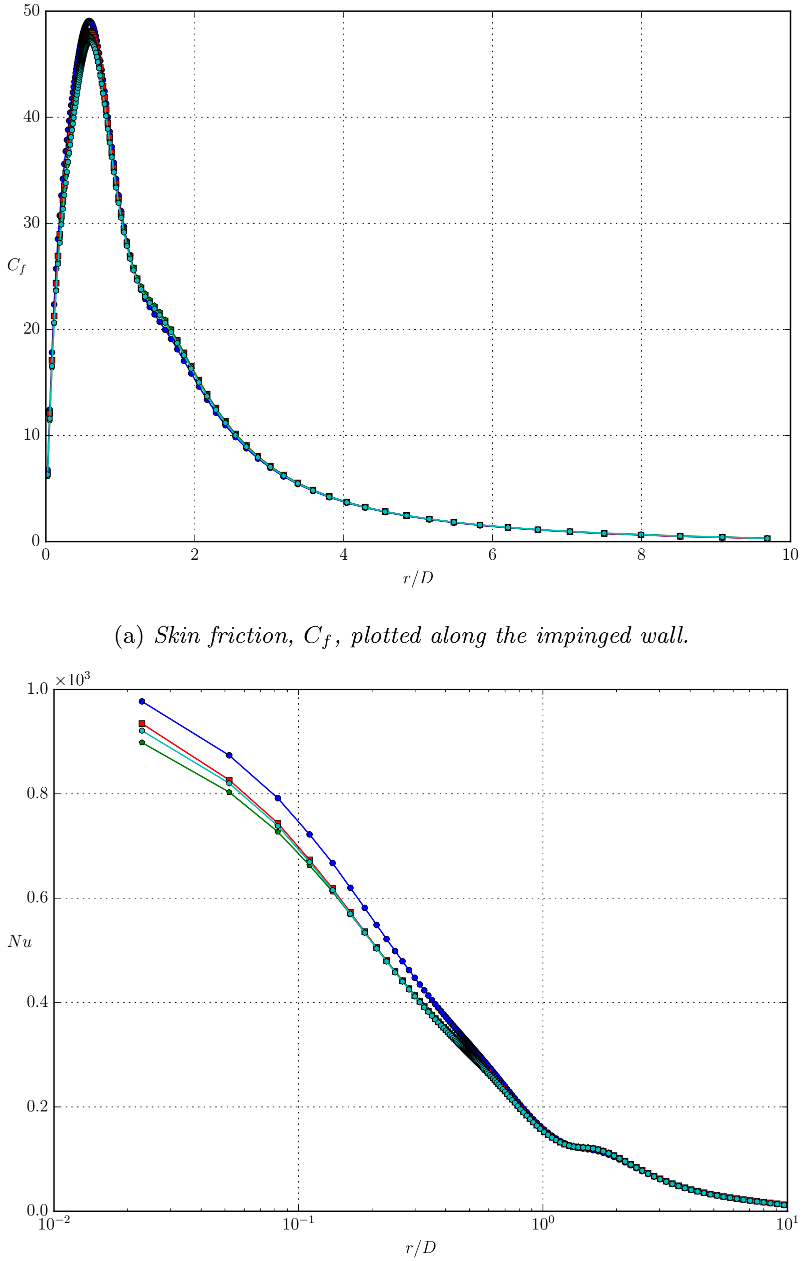
Table 3.1: Number of cells and computing times i.e., wall clock time, for the axisymmetric impinging jet using the Launder-Sharma model for different mesh strategies / wall treatments at the impinging wall.

In the following figures results from calculations using other y_i^+-values than given in the table 3.1 might occur. This happens when a set-up did not converge for the targeted y_i^+ but a set-up using a nearby y_i^+-value did, see e.g. figure 3.12. The results from the default LRN using the eMesh is plotted in figures 3.10, 3.13, and 3.16 for the three different Re_D. Between different y_i^+ the results do not differ much and only small deviations can be spotted around the peak skin-friction and around the impinging area for the heat-transfer. Hence, the LRN with eMesh is very robust towards changes of the grid topology within the tested range. Similar results are shown for the face flux

[2] For all simulations at $Re_D = 23000$ and $y_i^+ = 700$ the resulting y^+ was not kept within targeted tolerance but within $1.1 < y^+ < 1.2$.

[3] For all simulations at $Re_D = 220000$ and $y_i^+ = 100$ the resulting y^+ was not kept within targeted tolerance but within $0.9 < y^+ < 0.95$.

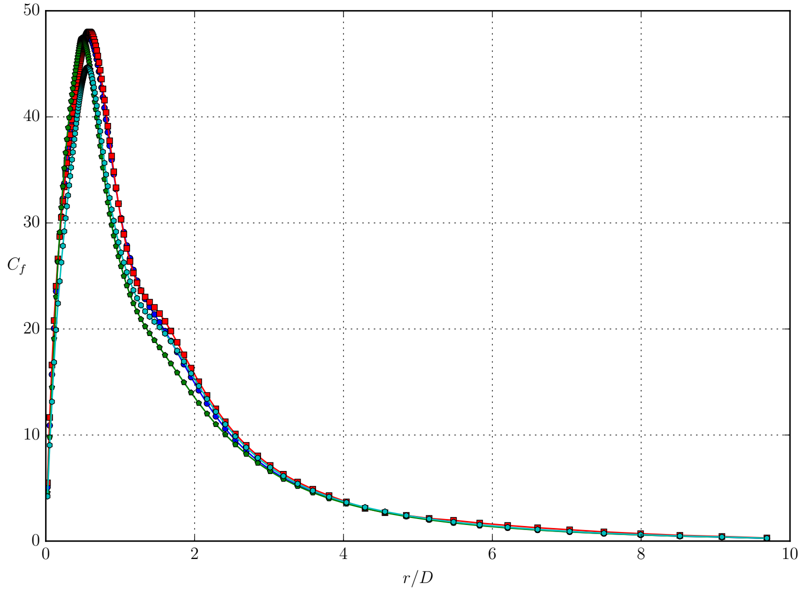
model in figures 3.11, 3.14, and 3.17 except that for $Re_D = 23000$ larger deviations are observed, see figure 3.11a for skin-friction and 3.11b for heat transfer. The results from the wall flux model demonstrated the same level of insensitivity as the face flux model, see figure 3.12, 3.15, and 3.18.



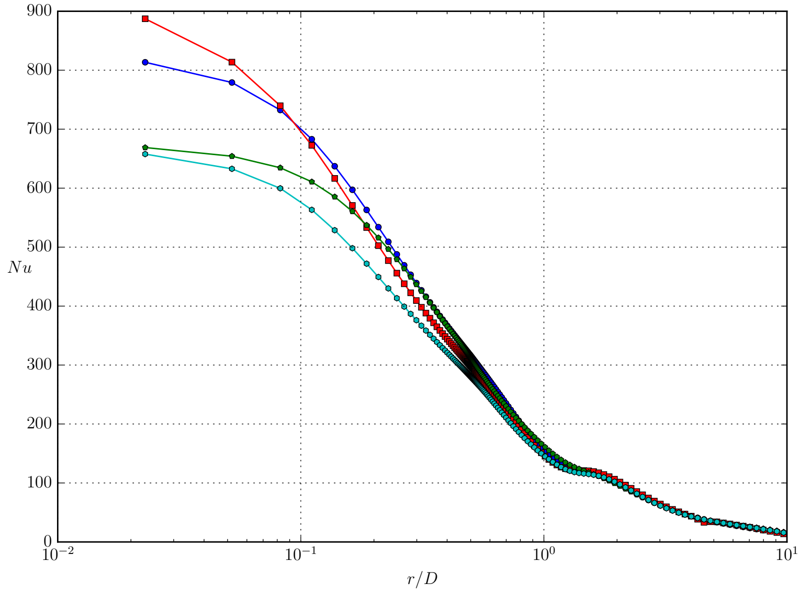
(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.10: *Launder-Sharma model at $Re_D = 23000$ comparing LRN using the eMesh for different y_i^+;* ▬ *:* $y_i^+ = 100$*,* ▬ *:* $y_i^+ = 300$*,* ▬ *:* $y_i^+ = 500$*, and* ▬ *:* $y_i^+ = 700$.

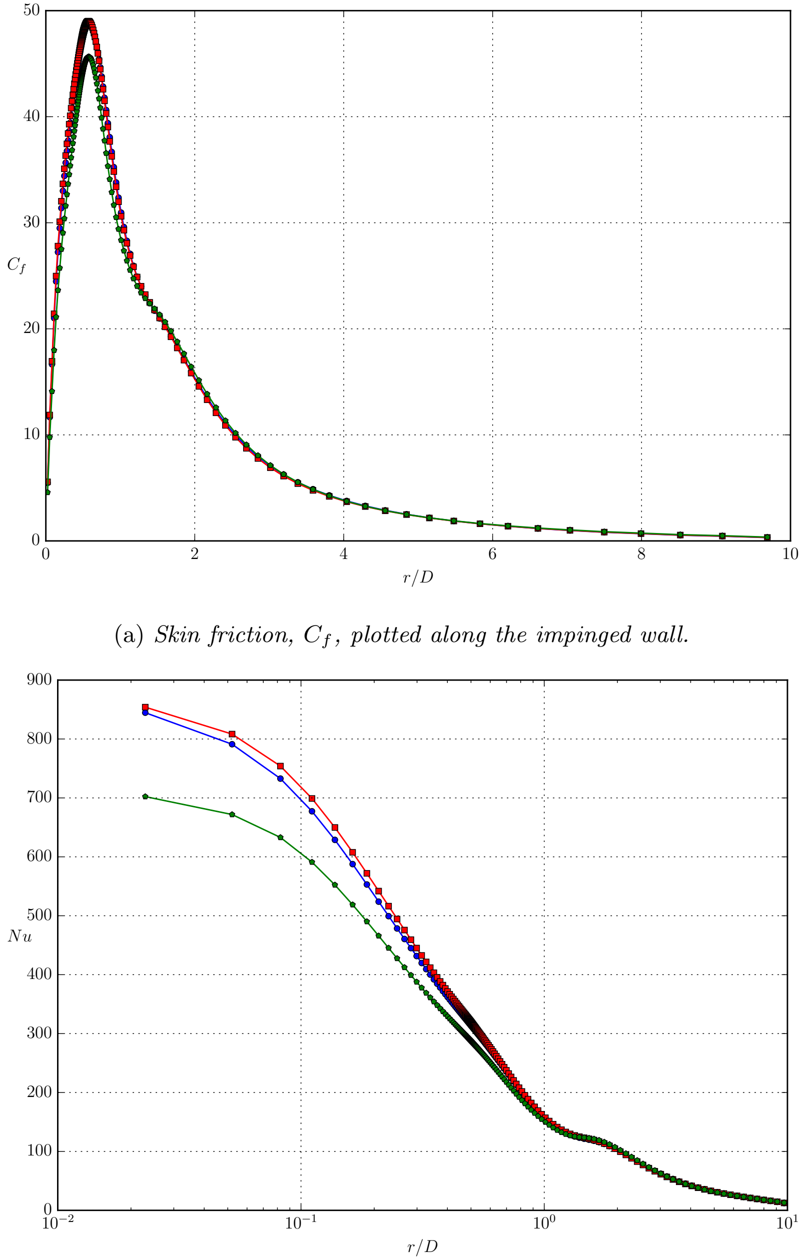


(a) *Skin friction, C_f, plotted along the impinged wall.*



(b) *Heat transfer, Nu, plotted along the impinged wall.*

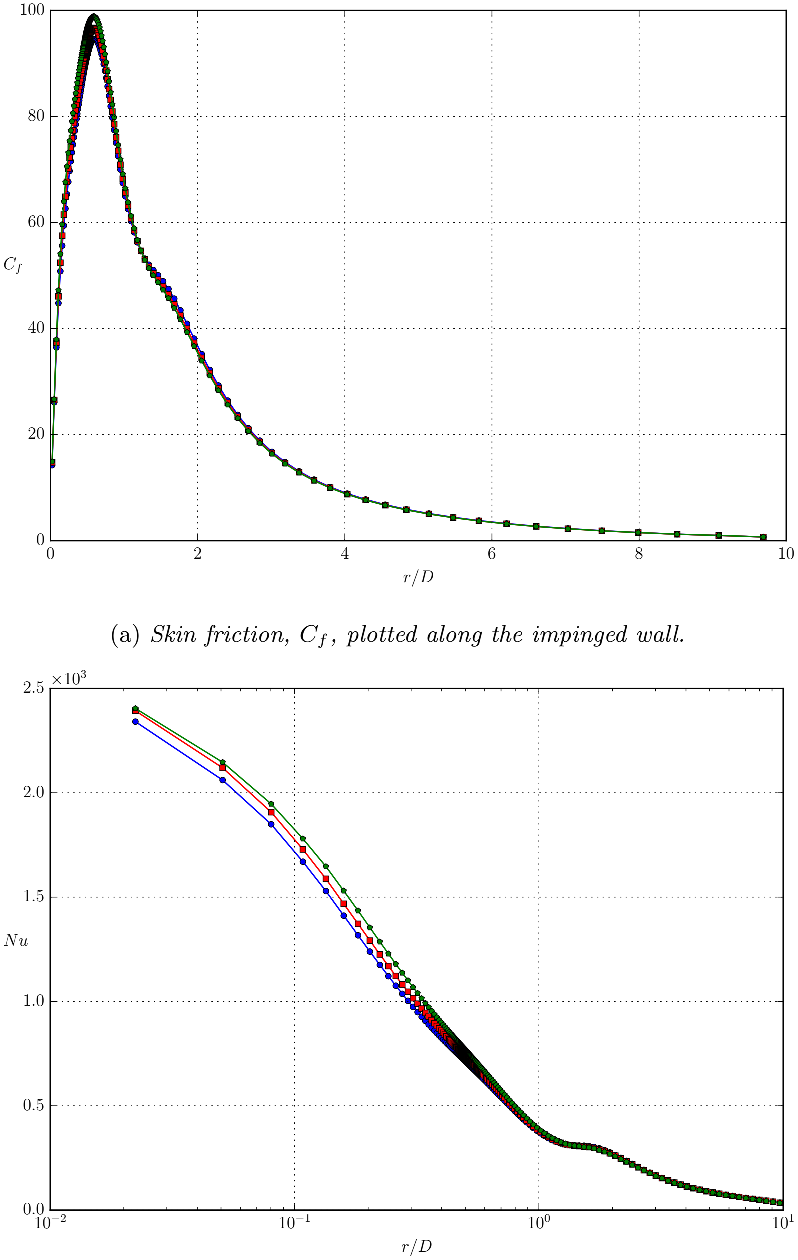
Figure 3.11: *Launder-Sharma model at $Re_D = 23000$ comparing the face flux model using the eMesh for different y_i^+;* 🟥 *: $y_i^+ = 100$,* 🟦 *: $y_i^+ = 300$,* 🟦 *: $y_i^+ = 500$, and* 🟩 *: $y_i^+ = 700$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

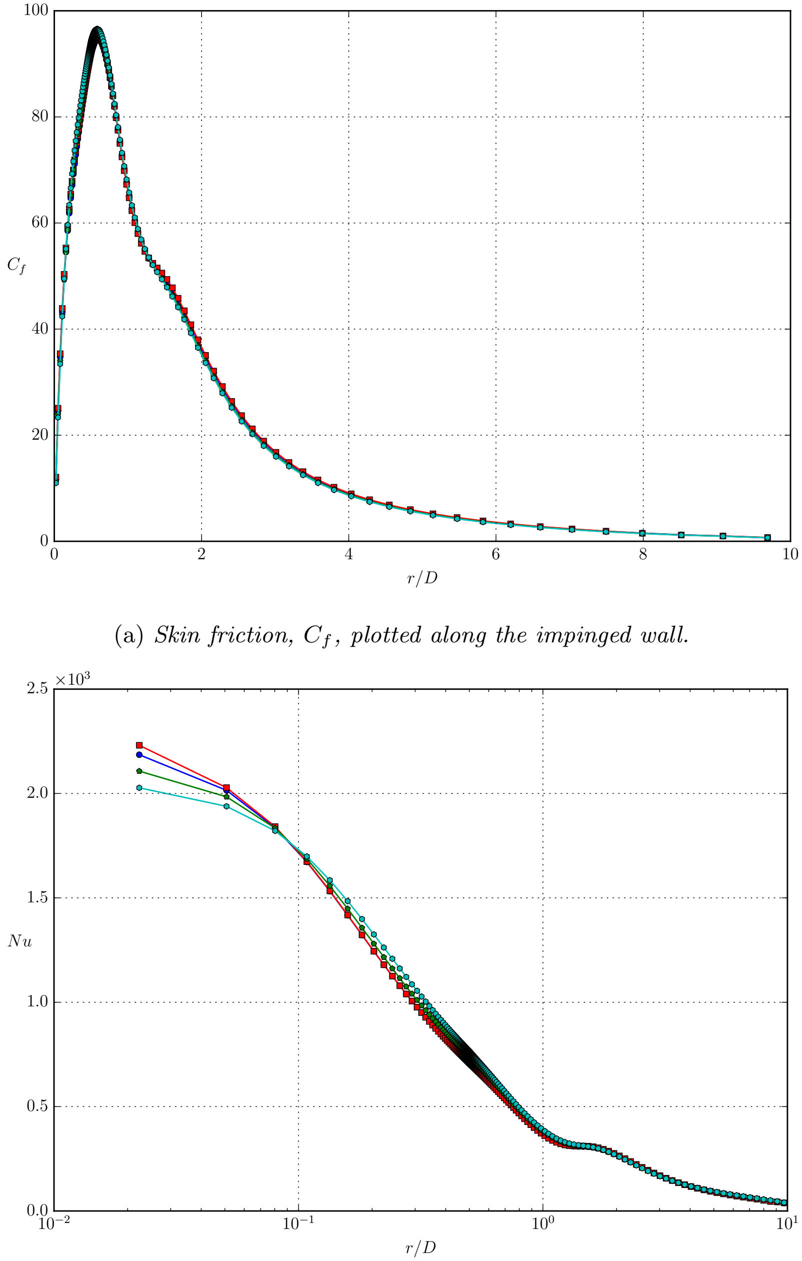
Figure 3.12: *Launder-Sharma model at $Re_D = 23000$ comparing the wall flux model using the eMesh for different y_i^+;* 🟦 *: $y_i^+ = 200$,* 🟥 *: $y_i^+ = 300$, and* 🟩 *: $y_i^+ = 400$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

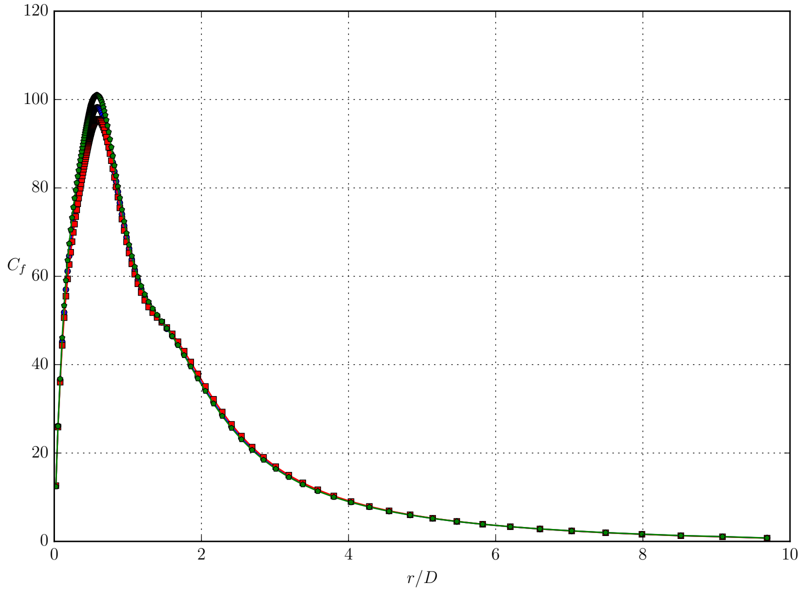
Figure 3.13: *Launder-Sharma model at $Re_D = 71000$ comparing LRN using the eMesh for different y_i^+; ━ : $y_i^+ = 300$, ━ : $y_i^+ = 500$, and ━ : $y_i^+ = 700$.*



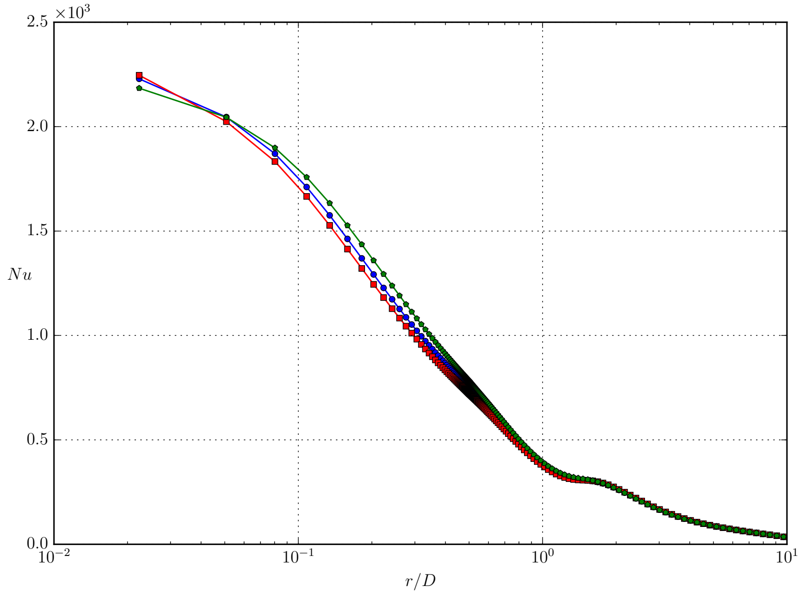
(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.14: *Launder-Sharma model at $Re_D = 71000$ comparing the face flux model using the eMesh for different y_i^+; — : $y_i^+ = 200$, — : $y_i^+ = 300$, — : $y_i^+ = 500$, and — : $y_i^+ = 700$.*

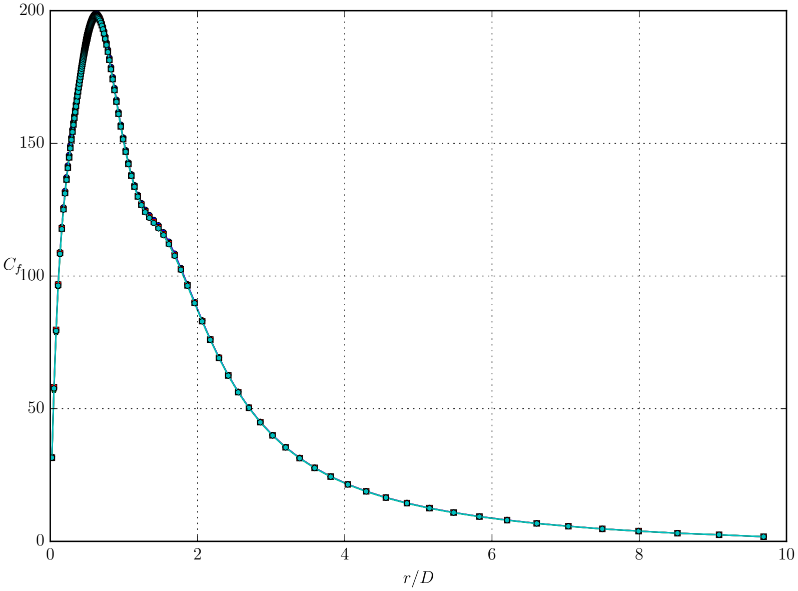


(a) *Skin friction, C_f, plotted along the impinged wall.*

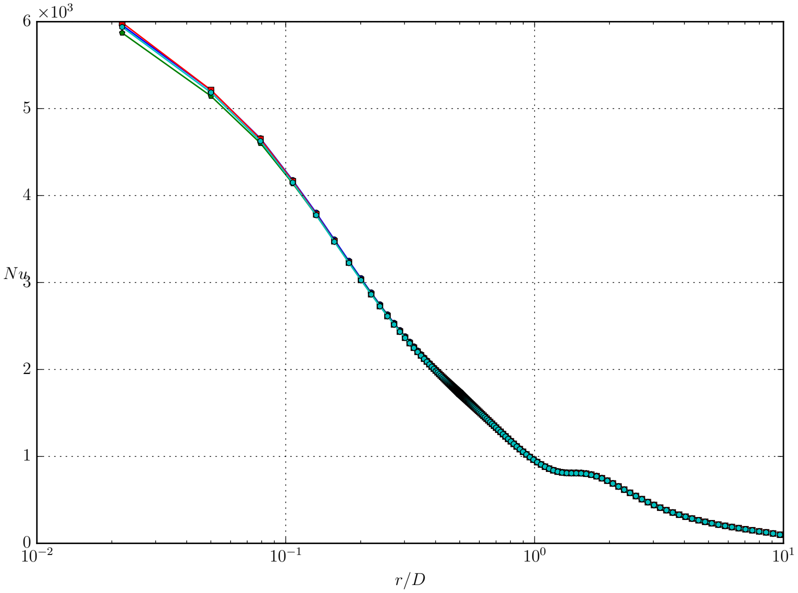


(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.15: *Launder-Sharma model at $Re_D = 71000$ comparing the wall flux model using the eMesh for different y_i^+;* 🟥 *: $y_i^+ = 300$,* 🟦 *: $y_i^+ = 500$, and* 🟩 *: $y_i^+ = 700$.*

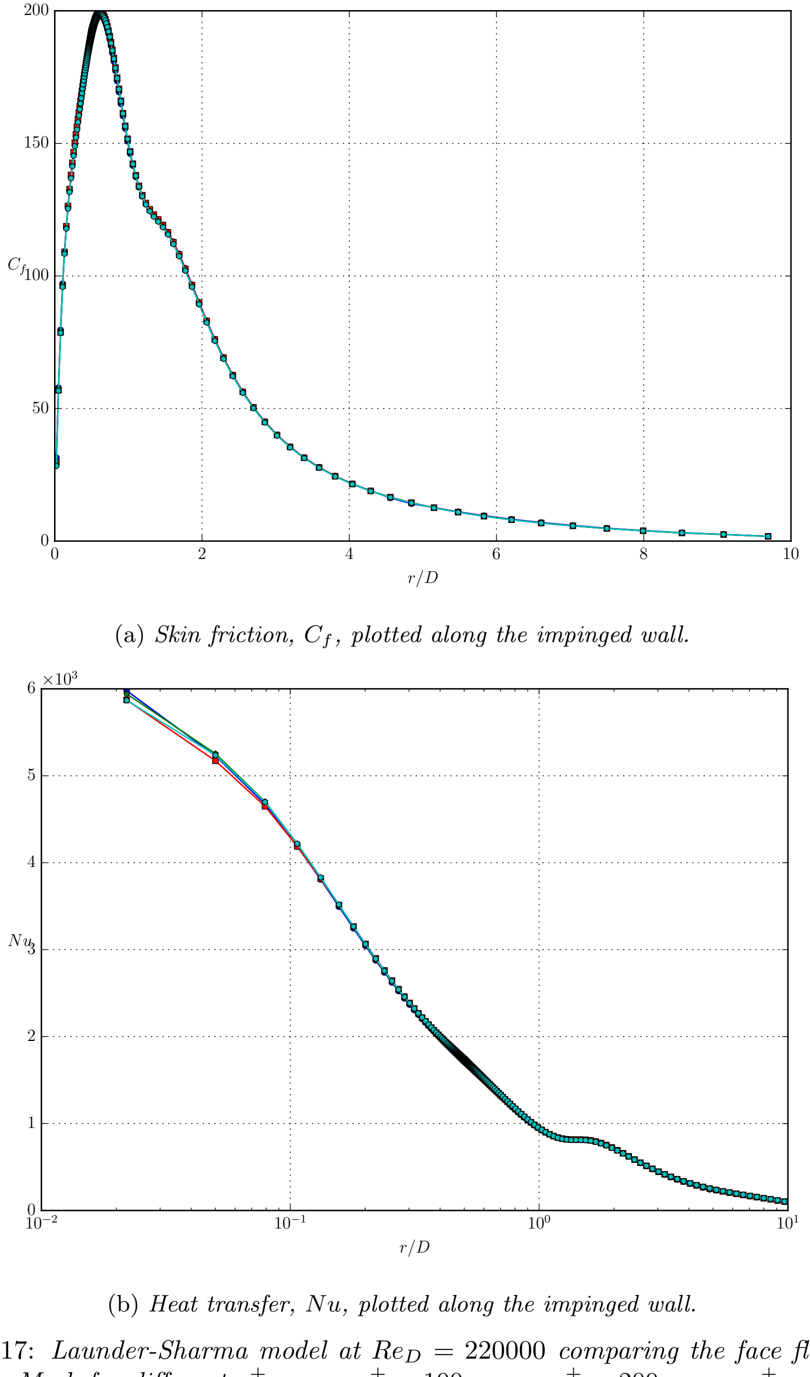


(a) *Skin friction, C_f, plotted along the impinged wall.*



(b) *Heat transfer, Nu, plotted along the impinged wall.*

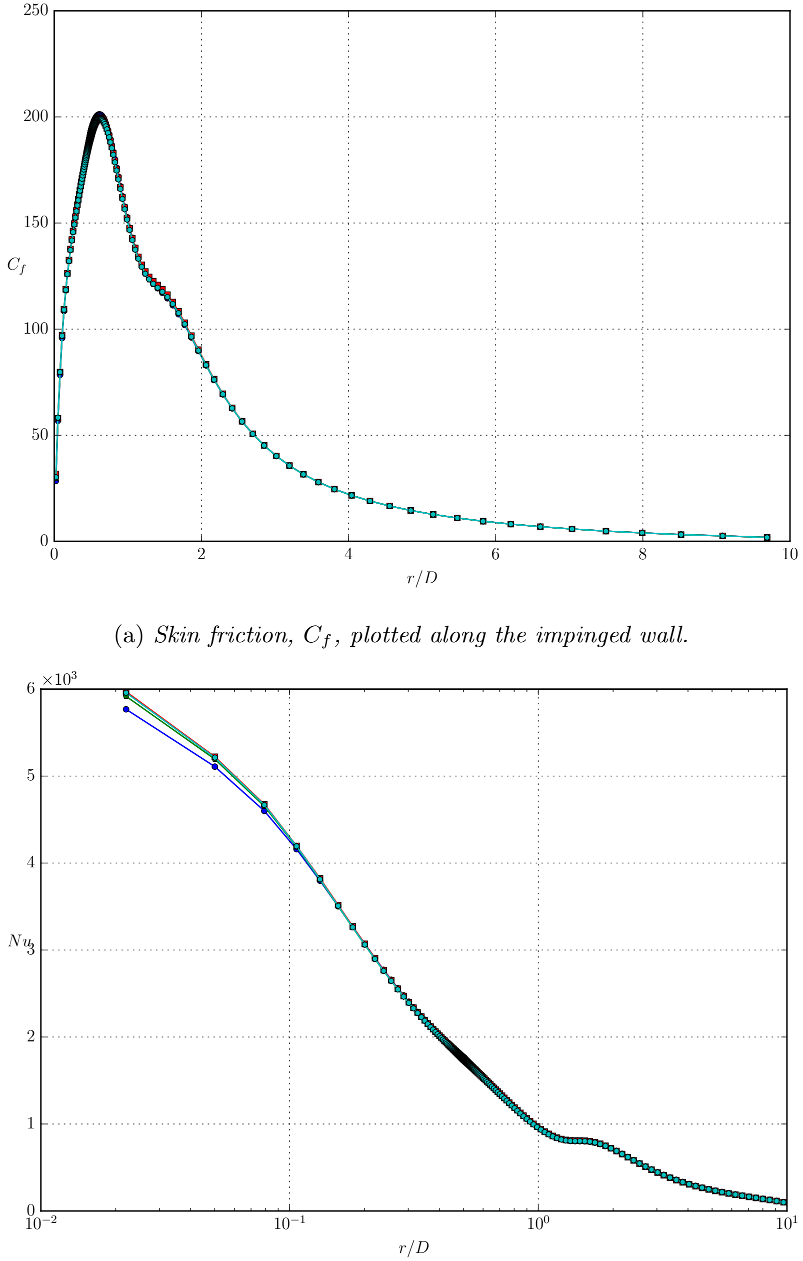
Figure 3.16: *Launder-Sharma model at $Re_D = 220000$ comparing LRN using the eMesh for different y_i^+;* 🟦 *: $y_i^+ = 100$,* 🟥 *: $y_i^+ = 300$,* 🟦 *: $y_i^+ = 500$, and* 🟩 *: $y_i^+ = 600$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.17: *Launder-Sharma model at $Re_D = 220000$ comparing the face flux model using the eMesh for different y_i^+; — : $y_i^+ = 100$, — : $y_i^+ = 200$, — : $y_i^+ = 300$, and — : $y_i^+ = 400$.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.18: *Launder-Sharma model at $Re_D = 220000$ comparing the wall flux model using the eMesh for different y_i^+;* ━ *: $y_i^+ = 100$,* ━ *: $y_i^+ = 300$,* ━ *: $y_i^+ = 400$, and* ━ *: $y_i^+ = 600$.*

Recommendations for near-wall resolution

Without a thorough analysis an attempt on updating the recommendations on grid design given in [3] is made for the axisymmetric impinging jet and the backward facing step.

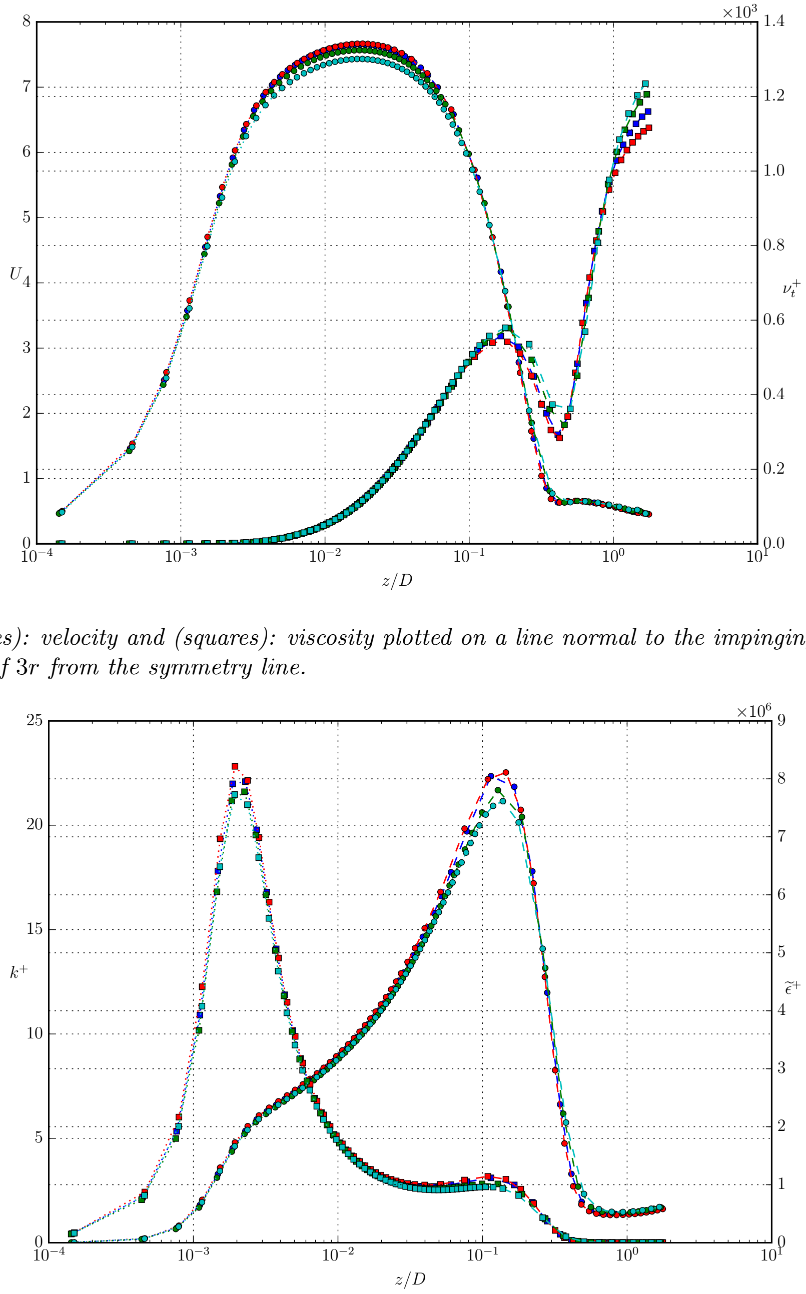
Turbulent axisymmetric impinging jet As guidance the velocity, the viscosity, the turbulent production and dissipation are plotted in figure 3.19. The distance of $3R$ from the symmetry line is chosen because the wall-normal gradients are highest at this line. For this set-up on $y_i^+ \in [200, 700]$ converged but going lower made the simulation diverge. In the plot the levels on y_i^+ correspond to a z/D-range of roughly $[0.025, 0.07]$. The most obvious feature for this range of z/D-values is that the gradient of the $\tilde{\varepsilon}$ is quite flat, see figure 3.19b which occurs around the peak velocity, see figure 3.19a.

Thus, with the hypothesis that the wall normal gradient of the turbulent dissipation is the quantity that is most sensitive to large cell size discontinuities the following recommendation is given:

- start from the wall with a small enough cell size to ensure a y^+-value close to unity
- use a relatively low and continuous expansion ratio of the cell size up to a wall distance where the gradient of $\tilde{\varepsilon}^+$ is low
- further out a higher stretching factor of around two is allowed to quickly reach a cell size fitted to the bulk flow and which gives low aspect ratios of the cells' faces.

In this way, the mesh may still resolve the boundary layer enough without using indispensable number of cells in the region outside the boundary layer.

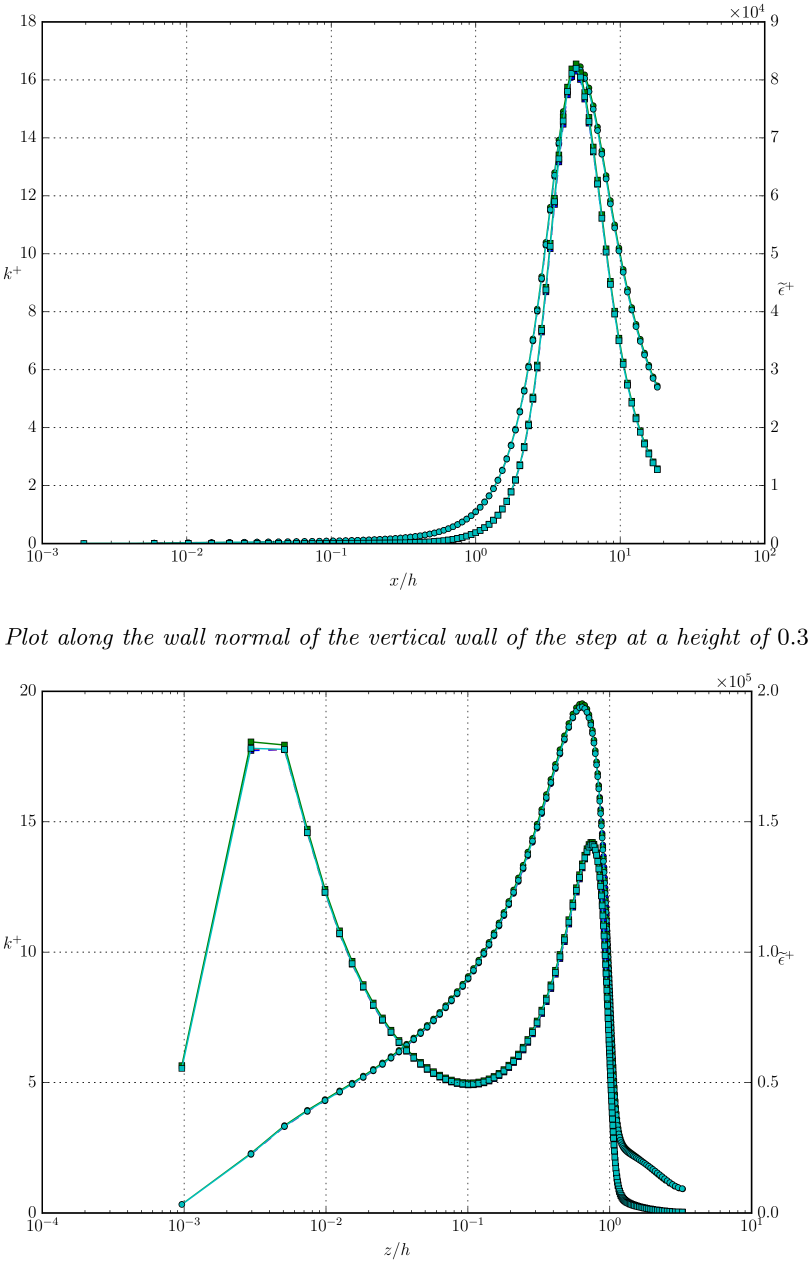
Backward facing step The recommendation in previous paragraph is here applied for the backward facing step which includes boundary layer separation, see appendix B. The walls enclosing the separation i.e., the vertical wall at the step and the floor downstream the step, do not have large enough regions where the gradient of the turbulent dissipation are low and hence it is not worthwhile applying different expansion ratios to them, see figures 3.20a and 3.20b. However the upper wall and the bottom wall upstream of the step show clear regions with low gradient of $\tilde{\varepsilon}$, see figure 3.21a and 3.21b, respectively. But applying the recommendation described above is only successful for the upper wall. Application to the lower wall affects also the resolution of the mesh downstream the step and the shear layer above the boundary layer separation will probably not be enough resolved. Different values on the y_i^+ has been investigated but no setting giving a converged solution has been found. All in all the updated near-wall resolution of the upper wall gives a moderate speed-up of around 30% and hence of limited use for flows including boundary layer separation. For these kind of flows the given recommendations in [3] seems fit where there is a strong coupling between the outer flow and the boundary layer.



(a) *(Circles): velocity and (squares): viscosity plotted on a line normal to the impinging wall at a distance of* 3*r from the symmetry line.*

(b) *(Circles): turbulent energy and (squares): turbulent dissipation plotted on a line normal to the impinging wall at a distance of* 3*r from the symmetry line.*

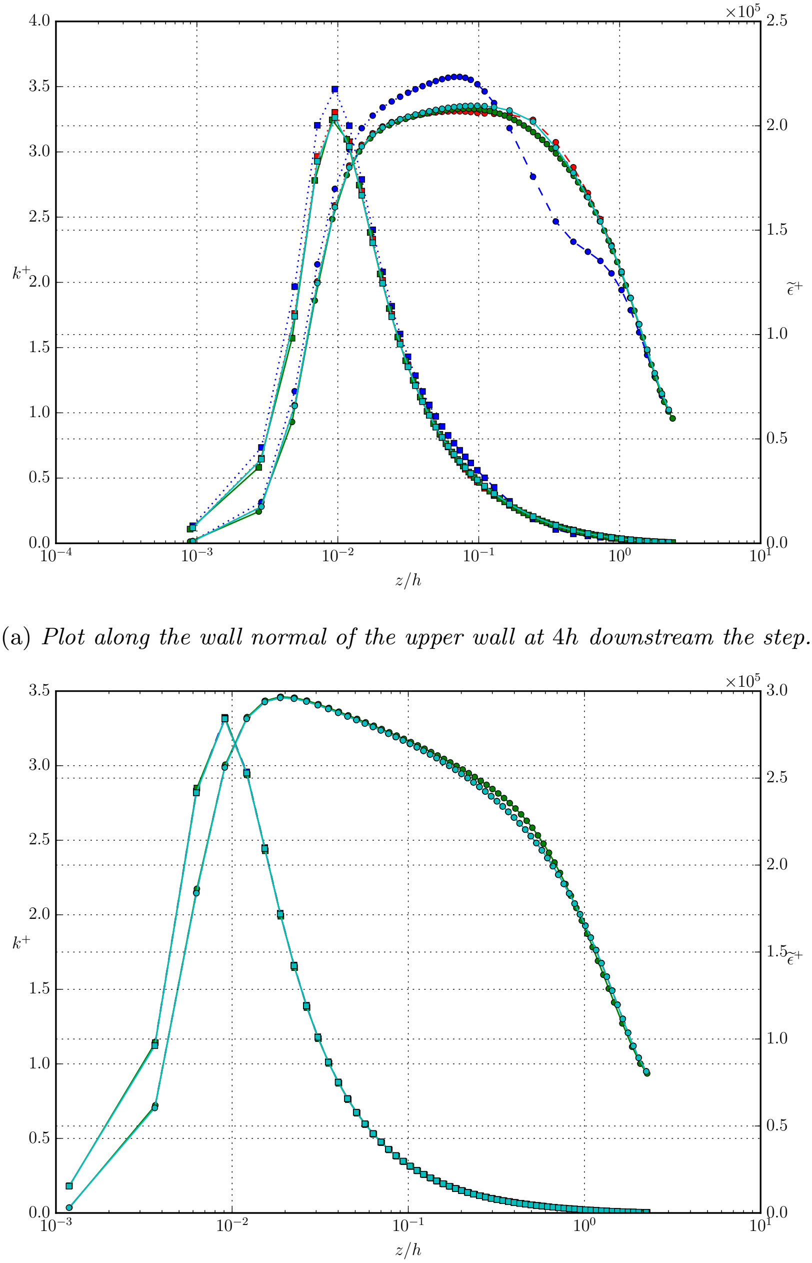
Figure 3.19: *Launder-Sharma model at Re_D = 71000 comparing the face flux model using the eMesh for different y_i^+;* — *: $y_i^+ = 200$,* — *: $y_i^+ = 300$,* — *: $y_i^+ = 500$, and* — *: $y_i^+ = 700$.*



(a) *Plot along the wall normal of the vertical wall of the step at a height of* 0.3*h*.

(b) *Plot along the wall normal of the bottom floor at* 4*h downstream the step.*

Figure 3.20: *Backward facing step using Launder-Sharma model at* $Re_H = 25000$. *Turbulent production, circles, and turblent dissipation, squares, plotted along wall normals from different walls for different near-wall strategies;* 🟩 *: default LRN,* 🟦 *: LRNe,* 🟦 *: face flux, and* 🟥 *: wall flux model.*



(a) *Plot along the wall normal of the upper wall at 4h downstream the step.*

(b) *Plot along the wall normal at the top of the step.*

Figure 3.21: *Backward facing step using Launder-Sharma model at* $Re_H = 25000$. *Turbulent production, circles, and turblent dissipation, squares, plotted along wall normals from different walls for different near-wall strategies;* 🟩 *: default LRN,* 🟦 *: LRNe,* 🟦 *: face flux, and* 🟥 *: wall flux model.*

Trade-off between accuracy and speed-up

Table 3.2: Computing times for the axisymmetric impinging jet at different Re_D for different near-wall strategies; HRN: realizable $k-\varepsilon$-model with standard wall function, SWF, LRN: Launder-Sharma $k-\tilde{\varepsilon}$-model, LRNe: standard LRN with eMesh, LRN-FF: face flux model, and LRN-WF: wall flux model.

$\boldsymbol{Re_D}$ **= 23000**	**HRN**	**LRN**	**LRNe**	**LRN-FF**	**LRN-WF**
Pipe wall treatm.	SWF ($y^+ = 40$)	Std	eMesh	Sub-grid ($y_i^+ = 40$)	
Impinging wall treatm.	SWF ($y^+ = 80$)	Std	eMesh	Sub-grid ($y_i^+ = 300$)	
Number of cells	726	10643	2898	881(+24+2052)	
No. of iterations	2021	6880	3496	3496	1377
Wall Clock Time [s]	23	638	109	133	58
Relative Speed-up	**28**	**1**	**6**	**5**	**11**

(a)

$\boldsymbol{Re_D}$ **= 71000**	**HRN**	**LRN**	**LRNe**	**LRN-FF**	**LRN-WF**
Pipe wall treatm.	SWF ($y^+ = 40$)	Std	eMesh	Sub-grid ($y_i^+ = 40$)	
Impinging wall treatm.	SWF ($y^+ = 80$)	Std	eMesh	Sub-grid ($y_i^+ = 300$)	
Number of cells	1853	15137	4847	1990(+24+2916)	
No. of iterations	40224	10944	4816	4634	2413
Wall Clock Time [s]	724	1610	291	312	182
Relative Speed-up	**2**	**1**	**6**	**5**	**9**

(b)

$\boldsymbol{Re_D}$ **= 220000**	**HRN**	**LRN**	**LRNe**	**LRN-FF**	**LRN-WF**
Pipe wall treatm.	SWF ($y^+ = 40$)	Std	eMesh	Sub-grid ($y_i^+ = 40$)	
Impinging wall treatm.	SWF ($y^+ = 80$)	Std	eMesh	Sub-grid ($y_i^+ = 300$)	
Number of cells	3086	20822	7868	4066(+24+3888)	
No. of iterations	6122	16747	6591	6507	4643
Wall Clock Time [s]	313	3440	689	721	560
Relative Speed-up	**11**	**1**	**5**	**5**	**6**

(c)

In table 3.2[4] grid sizes, number of solver iterations, and wall clock timings are summarized for different wall treatments. Here, the used wall treatment is applied to both walls i.e., impinging- and pipe wall, and the reduction in number of cells and computational timing will thus be more pronounced compared to earlier simulation, cf table 3.1 and 3.2. The grids used together with the HRN model have the fewest cells and deliver in general

[4] The eMesh used with default LRN has exactly the same topology as the effective mesh i.e., the main- and sub-grid, used with the sub-grid models. Observe that the sum of the total number of cells will differ as the wall adjacent cells in the main-grid overlapping the sub-grid do not exist in the eMesh.

the highest speed-up (10–30) compared to the default LRN model. The simulation at $Re_D = 71000$ displays a remarkable exception where the speed-up is only around two due to difficulties converging the velocity with the HRN model. Next in line is the wall flux model which takes roughly double the time as the HRN model despite having relatively many more cells. The reason is the convergence speed in number of iterations where this model excel. The face flux model and the LRN used with an eMesh demonstrate very similar results both in number of iterations and wall clock timing which indicates the analogy between the two. This also means that the major cost of a near-wall refined mesh is the propagation of diffusion and not the continuity i.e., calculation of pressure, which one might think earlier numerical wall treatments have assumed [6], [8], [16], [4], and [25]. However similar speed-up as shown here was demonstrated for UMIST-N having almost as many cells as the LRN set-up and where the gain in speed came almost solely from fewer number of iterations. A plausible explanation is that ensuring continuity and the use of face flux makes the model more robust but also slower to converge compared to the UMIST-N model. Nevertheless, the overview of the wall clock timings for all the set-ups displayed in figure 3.22 is a manifest of the considerable speed-up for any of the tested wall treatments investigated.

In figures 3.23, 3.24, and 3.25 the skin-friction and heat transfer at the impinging wall are presented. The general trend is that all wall treatments predict the flow with high Reynolds number better than flow with a lower value. As earlier shown in section 3.2, the difference between the sub-grid models and the default LRN set-ups are small, and the largest deviations are found for lower Re_D around the impinging area. In contrast the result from the HRN model differ significantly from the other models. The peak skin-friction is under-predicted with 30–50% and the Nusselt number is more than a magnitude lower around the impinging area. On the other hand the HRN model predicts both the skin-friction and the heat transfer well close to the outlet where the flow is close to fully developed and the assumptions of the law of the wall is fulfilled.

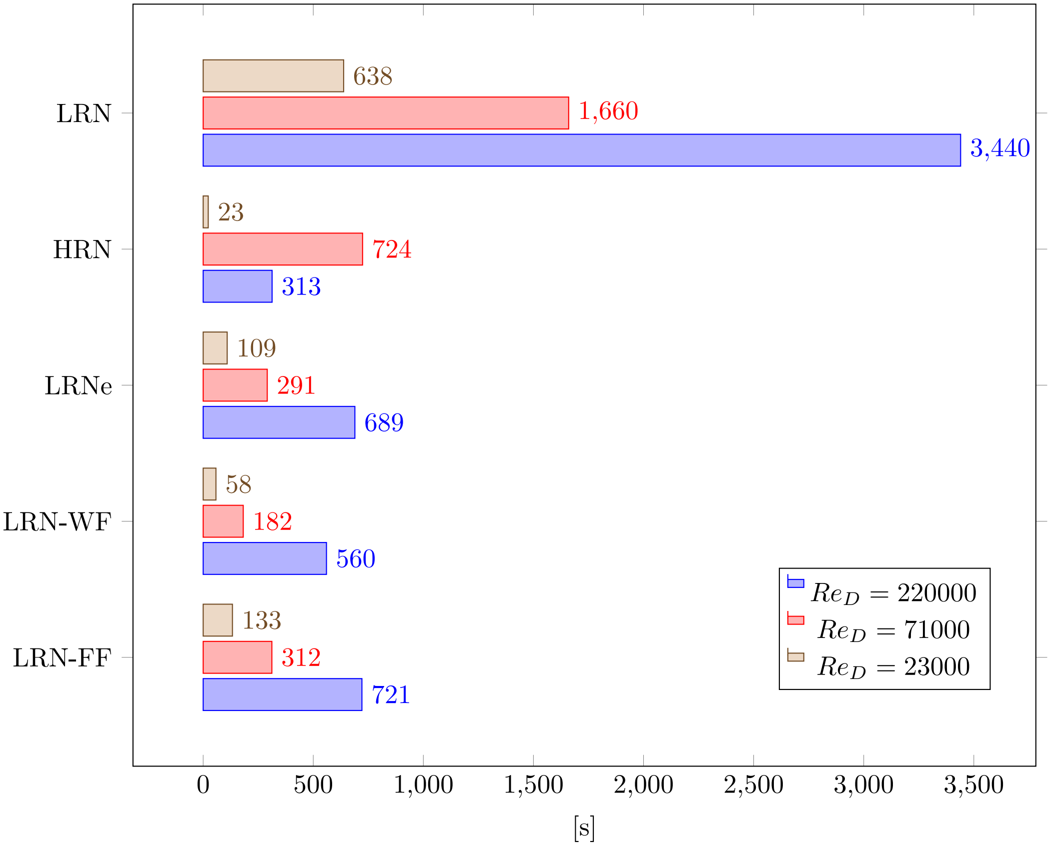
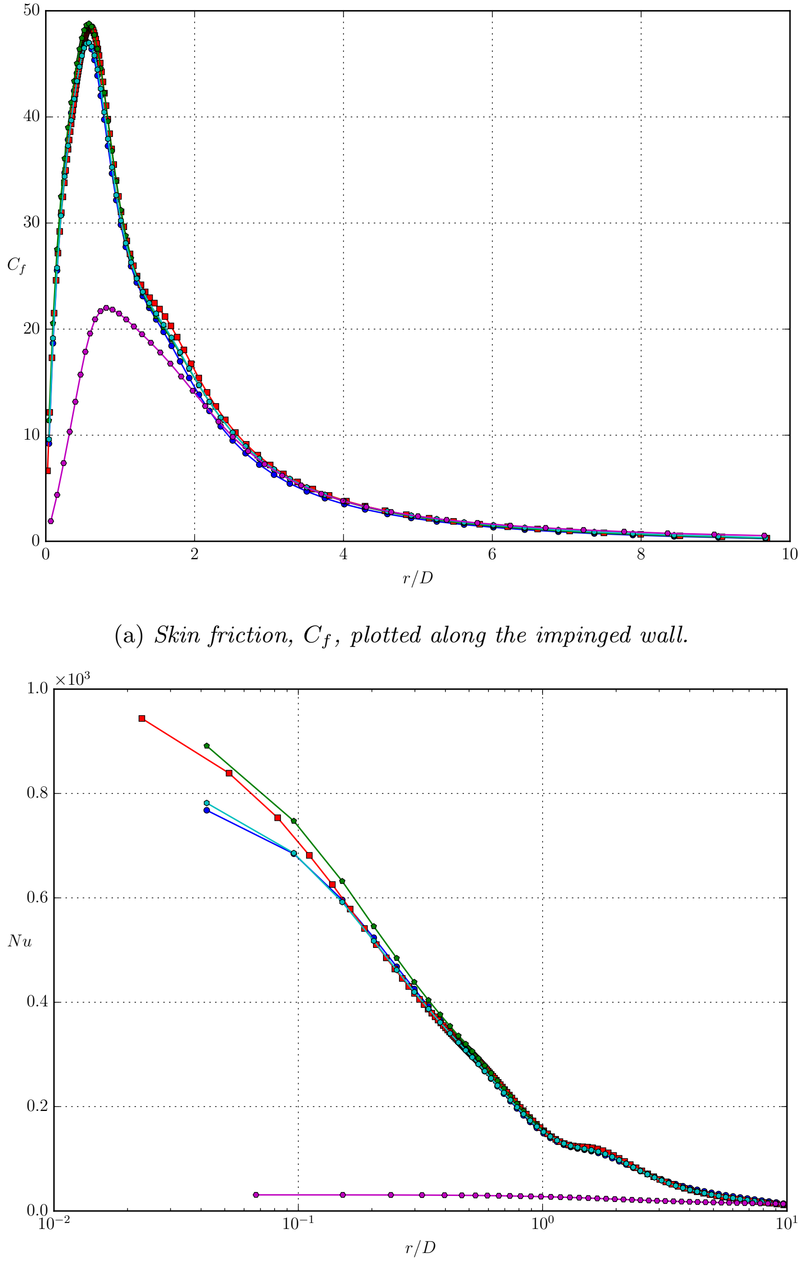


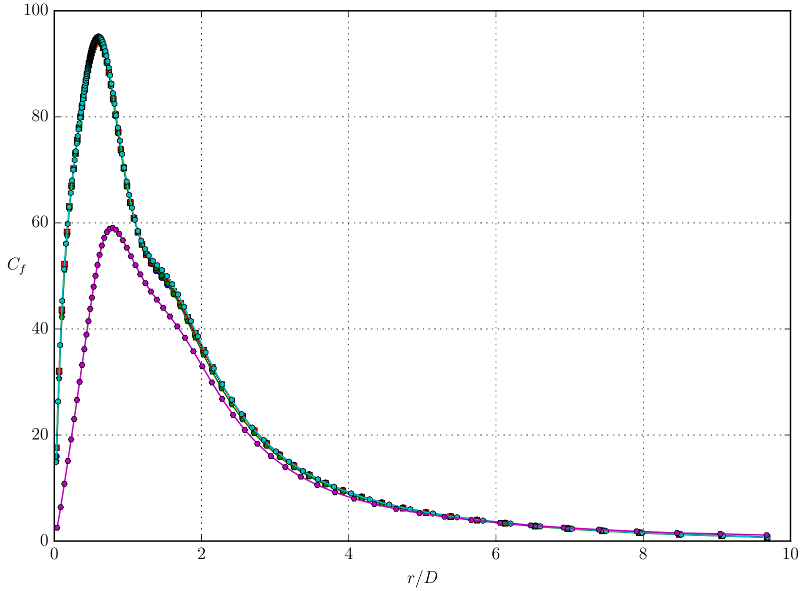
Figure 3.22: *Computing times for the axisymmetric impinging jet at different Re_D for different near-wall strategies; HRN: realizable $k-\varepsilon$-model with standard wall function, SWF, LRN: Launder-Sharma $k-\tilde{\varepsilon}$-model, LRNe: standard LRN with eMesh, LRN-FF: face flux model, and LRN-WF: wall flux model.*



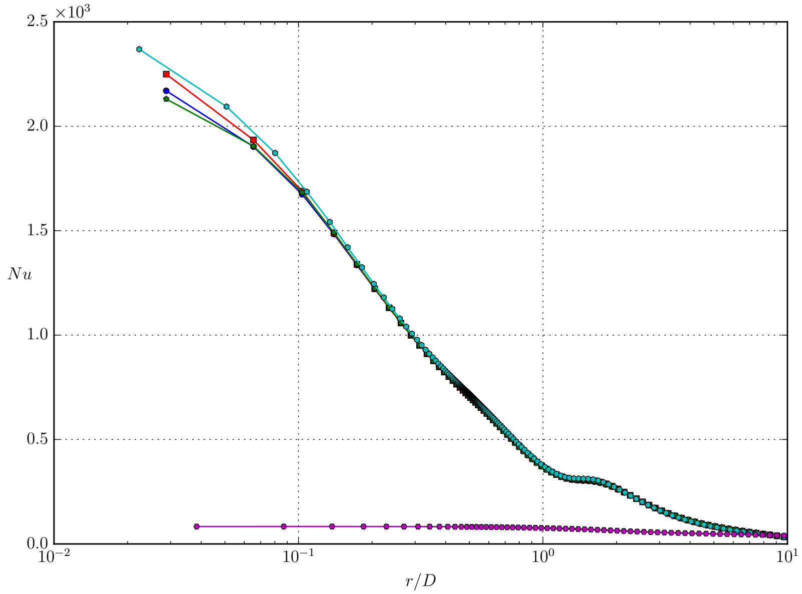
(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.23: $Re_D = 23000$ *where near-wall strategies are applied to both the impinging- and pipe wall, comparing;* — *: default LRN,* — *: LRN with eMesh,* — *face flux,* — *: wall flux,* — *: default HRN.*

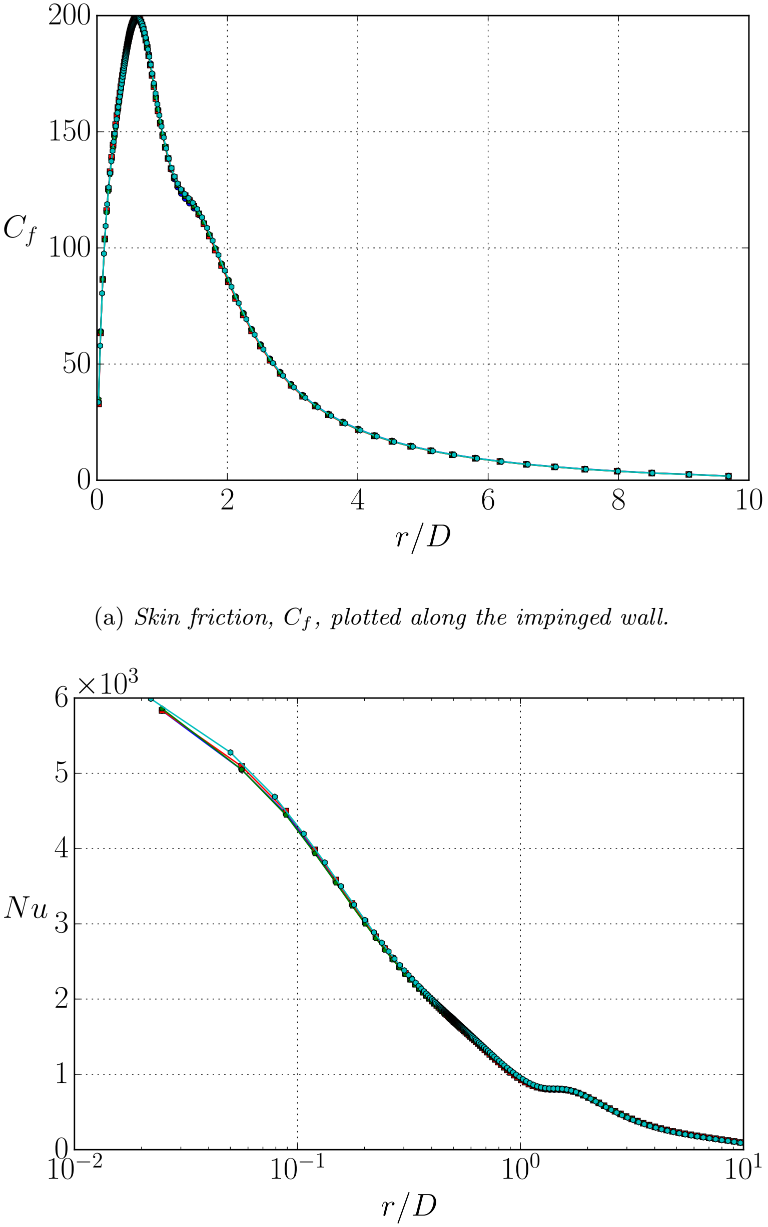


(a) *Skin friction, C_f, plotted along the impinged wall.*



(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.24: $Re_D = 71000$ *where near-wall strategies are applied to both the impinging- and pipe wall, comparing;* 🟦 *: default LRN,* 🟥 *: LRN with eMesh,* 🟩 *face flux,* 🟦 *: wall flux,* 🟪 *: default HRN.*



(a) *Skin friction, C_f, plotted along the impinged wall.*

(b) *Heat transfer, Nu, plotted along the impinged wall.*

Figure 3.25: $Re_D = 220000$ *where near-wall strategies are applied to both the impinging- and pipe wall, comparing;* 🟥 *: default LRN,* 🟪 *: LRN with eMesh,* 🟦 *face flux,* 🟩 *: wall flux,* 🟦 *: default HRN.*

4 Concluding remarks

Two new numerical wall functions, independent of the turbulence model, have been implemented in the open source CFD package OpenFOAM. They use a *sub-grid* which overlaps the wall-adjacent cells in an ordinary *main-grid*. The coupling between the two grids uses face fluxes, giving a stringent mathematical implementation from a finite volume perspective. Continuity on the sub-grid is obtained by ensuring the mass flux over the faces overlapping a main-grid face is equal to that of the main-grid face' before cell internal mass flux in the sub-grid is updated to give a divergence-free velocity field in each and every cell of the sub-grid. The results show excellent agreement with default low-Reynolds-number calculations, while the computing requirement is somewhere between the requirements of high-Reynolds-number models with wall functions and those of low-Reynolds-number models.

Compared to earlier numerical wall functions [6], [8], [16], [4] and [2], presented in the introduction, the present new numerical wall functions are the first ones not being restricted to a specific turbulence model due to the use of face fluxes and the implementation in the highly modular CFD package OpenFOAM. This independence makes it very general, and should work directly with any low-Reynolds-number model constituted by transport equations[1]. Earlier methods have instead used Dirichlet boundary conditions with interpolated values from the main-grid, together with averaged volume sources of the turbulent production and dissipation terms from the sub-grid, for the coupling between the two meshes. The new wall functions are also the first ones to ensure continuity on the sub-grid, which is the most plausible reason for being very robust for a wide range of Reynolds number and insensitive to changes in size of the near-wall main-grid cells.

However, an even more robust set-up was found using a standard low-Reynolds-number model with an optimized mesh that relaxes the general recommendations for grid generation of the near-wall resolution. This finding, resulting in similar speed-up as the implemented numerical wall functions, proves that the major calculation cost of the boundary layer is the diffusion and not the continuity for steady-state flows. To summarize, a number of inferences can be drawn:

- for complex flow, HRN modelling is inaccurate, especially for heat transfer
- new recommendations for grid design on near-wall resolution can drastically reduce the computing times required for LRN modelling for certain types of flows, e.g. an impinging jet, but not for other types, e.g. a backward-facing step
- no advantages have been demonstrated with the face flux model compared to using the LRN turbulence model with an AMS mesh
- the numerical wall flux model, i.e. NWF, can give slightly larger speed-up but may suffer from robustness problems

[1] It should also work with any extra transport equation of any scalar or vector. The present numerical wall functions could also be adapted to a low-Reynolds-number model, which includes other type of equations, e.g. elliptic ones.

The numerical wall functions, and the relaxed grid design recommendations, bring a sweet-spot between accuracy and computing time for the tested low-Reynolds-number model, which is beneficial for many turbulent flow problems, but especially for complex industrial cases. The results from the set-ups presented here are in excellent agreement with the investigated turbulence model, and will probably be for any low-Reynolds-number model. The computing requirement is up to an order of magnitude less than for a default set-up of a low-Reynolds-number model.

The most compelling strategy is of course the relaxed grid design recommendation, i.e. AMS, as this does not require any alteration of the CFD code. To increase the understanding of what speed-up can be expected using AMS for different flows, more testing is needed.

References

[1] J. Baughn, A. Hechanova and X. Yan. An Experimental-Study of Entrainment Effects on the Heat-Transfer from a Flat Surface to a Heated Circular Impinging Jet. *J. Heat Transf.-Trans. ASME* **113**.4 (Nov. 1991). WOS:A1991HP76100034, 1023–1025. ISSN: 0022-1481. DOI: 10.1115/1.2911197.

[2] R. B. Bond and F. G. Blottner. Derivation, implementation, and initial testing of a compressible wall-layer model. *Int. J. Numer. Meth. Fluids* **66**.9 (30th July 2011), 1183–1206. ISSN: 1097-0363. DOI: 10.1002/fld.2309. URL: http://onlinelibrary.wiley.com/doi/10.1002/fld.2309/abstract (visited on 30/09/2015).

[3] M. Casey and T. Wintergerste. *Ercoftac, Best practice Guidelines of Industrial Computational Fluid Dynamics of Single-Phase Flows.* Ercoftac, 2000. (Visited on 20/05/2016).

[4] F. Chedevergne. Advanced wall model for aerothermodynamics. *International Journal of Heat and Fluid Flow* **31**.5 (Oct. 2010). WOS:000281918600017, 916–924. DOI: 10.1016/j.ijheatfluidflow.2010.05.002.

[5] D. Cooper et al. Impinging jet studies for turbulence model assessment I. Flow-field experiments. *International Journal of Heat and Mass Transfer* **36**.10 (1993), 2675–2684. URL: http://www.sciencedirect.com/science/article/pii/S0017931005802042 (visited on 23/05/2013).

[6] T. Craft et al. A new wall function strategy for complex turbulent flows. *Numerical Heat Transfer, Part B: Fundamentals.* Numerical Heat Transfer, Part B: Fundamentals **45**.4 (2004), 301–318. ISSN: 1040-7790. DOI: 10.1080/10407790490277931.

[7] Edgewall. *Trac is a web-based software project management and bug/issue tracking system emphasizing ease of use and low ceremony.* URL: http://trac.edgewall.org.

[8] S. E. Gant. 'Development and Application of a New Wall Function for Complex Turbulent Flows'. PhD thesis. Dept. of Mechanical, Aerospace and Manufacturing Engineering, UMIST, 2002. URL: http://www.finite.org.uk/.

[9] B. Gregg. *Perf Wiki.* URL: https://perf.wiki.kernel.org/index.php/Main_Page (visited on 28/04/2016).

[10] B. Gschaider. *pyFoam: A python library to control OpenFOAM-runs and manipulate OpenFOAM-data.* Version 0.6.4. URL: https://openfoamwiki.net/index.php/Contrib/PyFoam.

[11] J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering* **9**.3 (May 2007), 90–95. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.55.

[12] H. Krekel et al. *pytest: helps you write better programs.* Version 2.7. URL: http://pytest.org.

[13] H. Krekel et al. *tox aims to automate and standardize testing in Python.* Version 1.7.0. URL: http://tox.readthedocs.io.

[14] B. Launder and B. Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer* **1**.2 (Nov. 1974), 131–137. ISSN: 0094-4548. DOI: 10.1016/0094-4548(74)90150-7. URL: http://www.sciencedirect.com/science/article/pii/0094454874901507 (visited on 12/12/2013).

[15] M. McKerns and M. Aivazis. *pathos: a framework for heterogeneous computing.* 2010. URL: http://trac.mystic.cacr.caltech.edu/project/pathos.

[16] S. Myers and D. Keith Walters. 'A one-dimensional subgrid near wall treatment for turbulent flow CFD simulation'. *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FED.* Vol. 261 FED. 2005, pp. 577–585. ISBN: 0-7918-4219-3 978-0-7918-4219-5.

[17] C. Online. *OpenFOAM – CFD Online Discussion Forums.* URL: http://www.cfd-online.com/Forums/openfoam/ (visited on 28/04/2016).

[18] OPENFOAM®. *Version 2.3.x.* 2015. URL: http://www.openfoam.org/ (visited on 09/05/2016).

[19] RotateRight. *Zoom: a GUI for the perf Linux profiler.* Version 3.3.3. URL: http://www.rotateright.com/.

[20] T.-H. Shih et al. A new k-ϵ eddy viscosity model for high reynolds number turbulent flows. *Computers & Fluids* **24**.3 (Mar. 1995), 227–238. ISSN: 0045-7930. DOI: 10.1016/0045-7930(94)00032-T. URL: http://www.sciencedirect.com/science/article/pii/004579309400032T (visited on 02/05/2016).

[21] P. R. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow* **21**.3 (June 2000), 252–263. ISSN: 0142-727X. DOI: 10.1016/S0142-727X(00)00007-2. URL: http://www.sciencedirect.com/science/article/pii/S0142727X00000072 (visited on 23/05/2016).

[22] S. Takats et al. *Zotero [zoh-TAIR-oh] is a free, easy-to-use tool to help you collect, organize, cite, and share your research sources.* Version 4.0.29.8. URL: https://www.zotero.org.

[23] T. Tantou. *PGF and TikZ – Graphic systems for TeX.* URL: https://sourceforge.net/projects/pgf/.

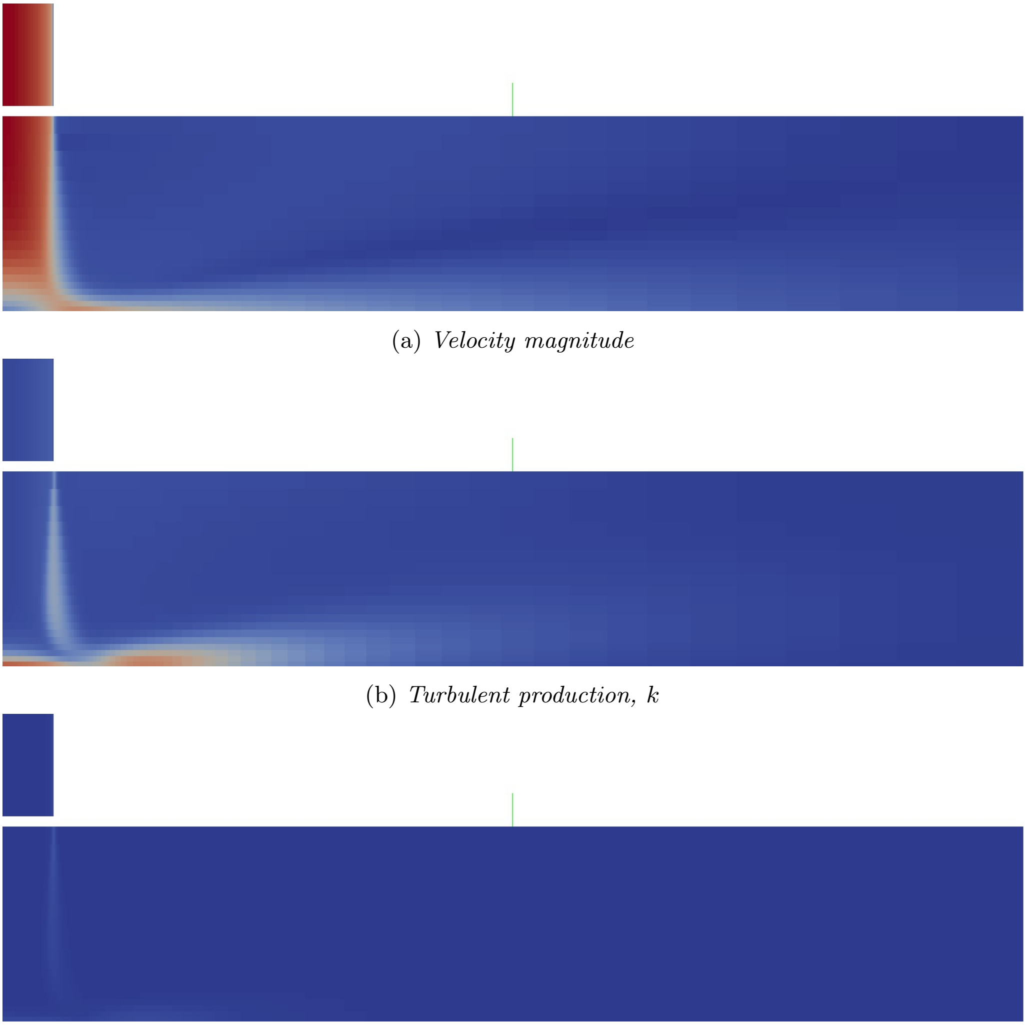
[24] L. Torvalds, J. Hamano and many others. *Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.* URL: https://git-scm.com.

[25] J.-F. Wald. 'Lois de paroi adaptatives pour un modèle de fermeture du second ordre dans un contexte industriel'. PhD thesis. de l'Université de Pau et des Pays de l'Adour, 2016.

Appendices

A Axisymmetric impinging jet

A.1 Flow pattern



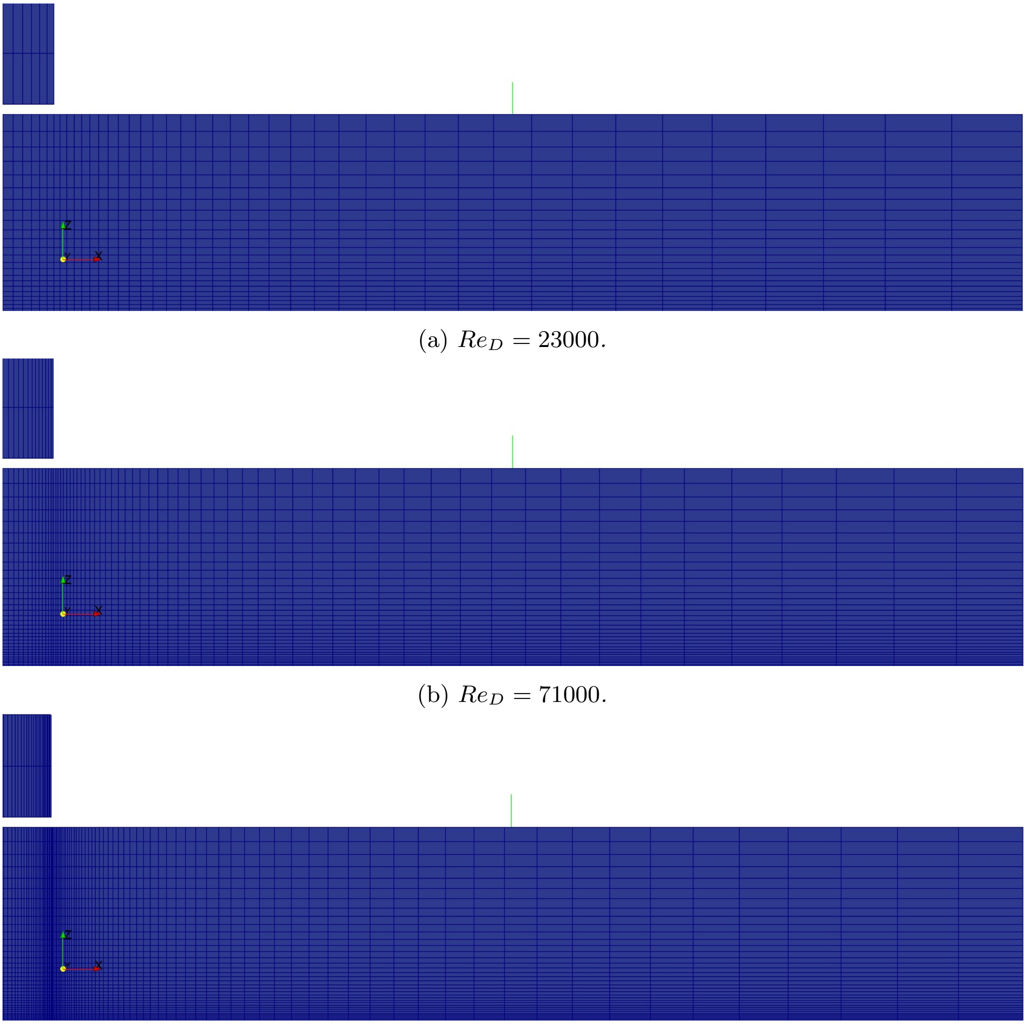
(a) *Velocity magnitude*

(b) *Turbulent production, k*

(c) *Turbulent dissipation, $\tilde{\varepsilon}$*

Figure A.1: *Flow pattern of the axisymmetric impinging jet at $Re_D = 71000$ using the Launder-Sharma model.*

A.2 Meshes

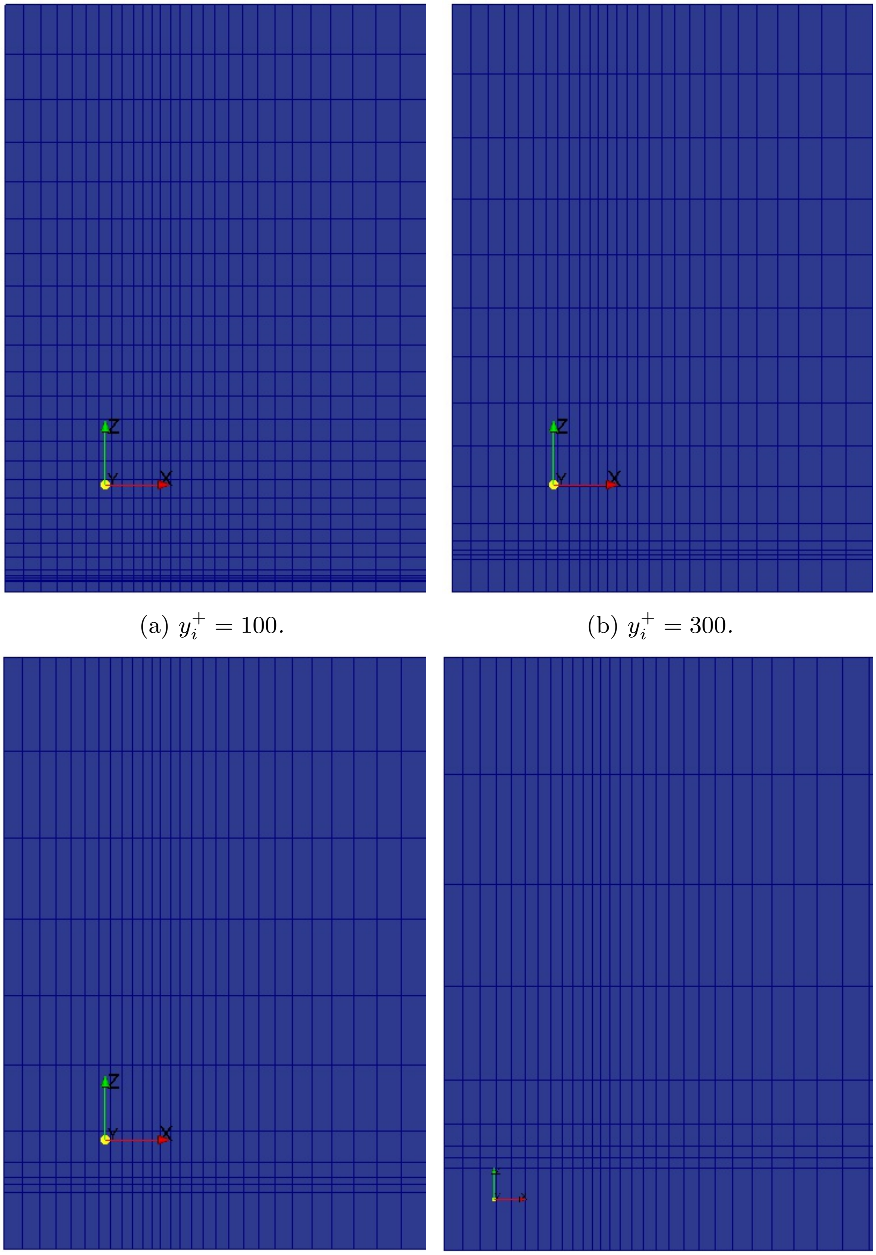


(a) $Re_D = 23000$.

(b) $Re_D = 71000$.

(c) $Re_D = 220000$.

Figure A.2: *Grids used in the turbulent axisymmetric semi-confined impinging jet with HRN model at different Reynolds number, $y^+ = 80$ for both pipe and impinging walls.*



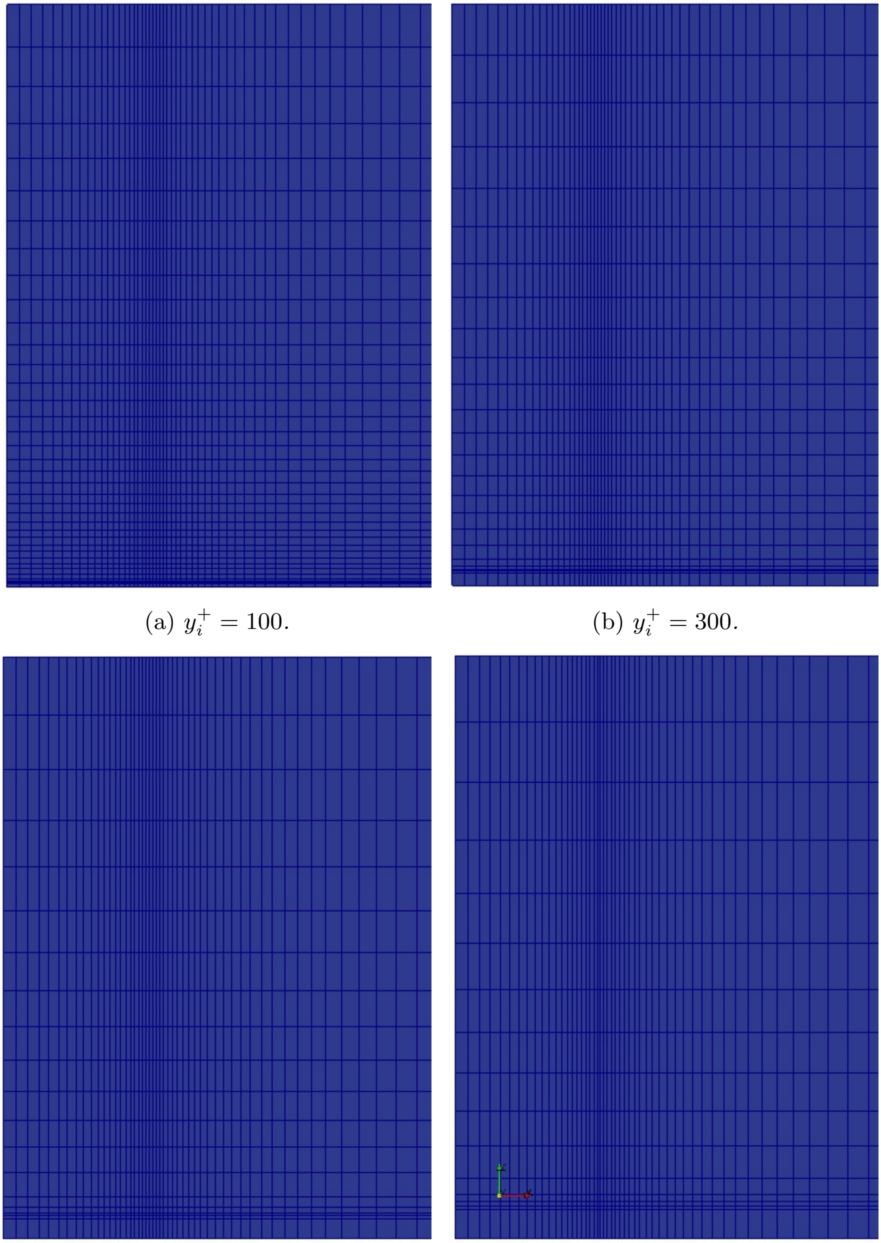
(a) $y_i^+ = 100$.

(b) $y_i^+ = 300$.

(c) $y_i^+ = 500$.

(d) $y_i^+ = 700$.

Figure A.3: *Part of the main-grids used with the sub-grid models, $r = z = 0$ at the lower left corner of the plots, used in the turbulent axisymmetric semi-confined impinging jet at $Re_D = 23000$ for different y_i^+.*



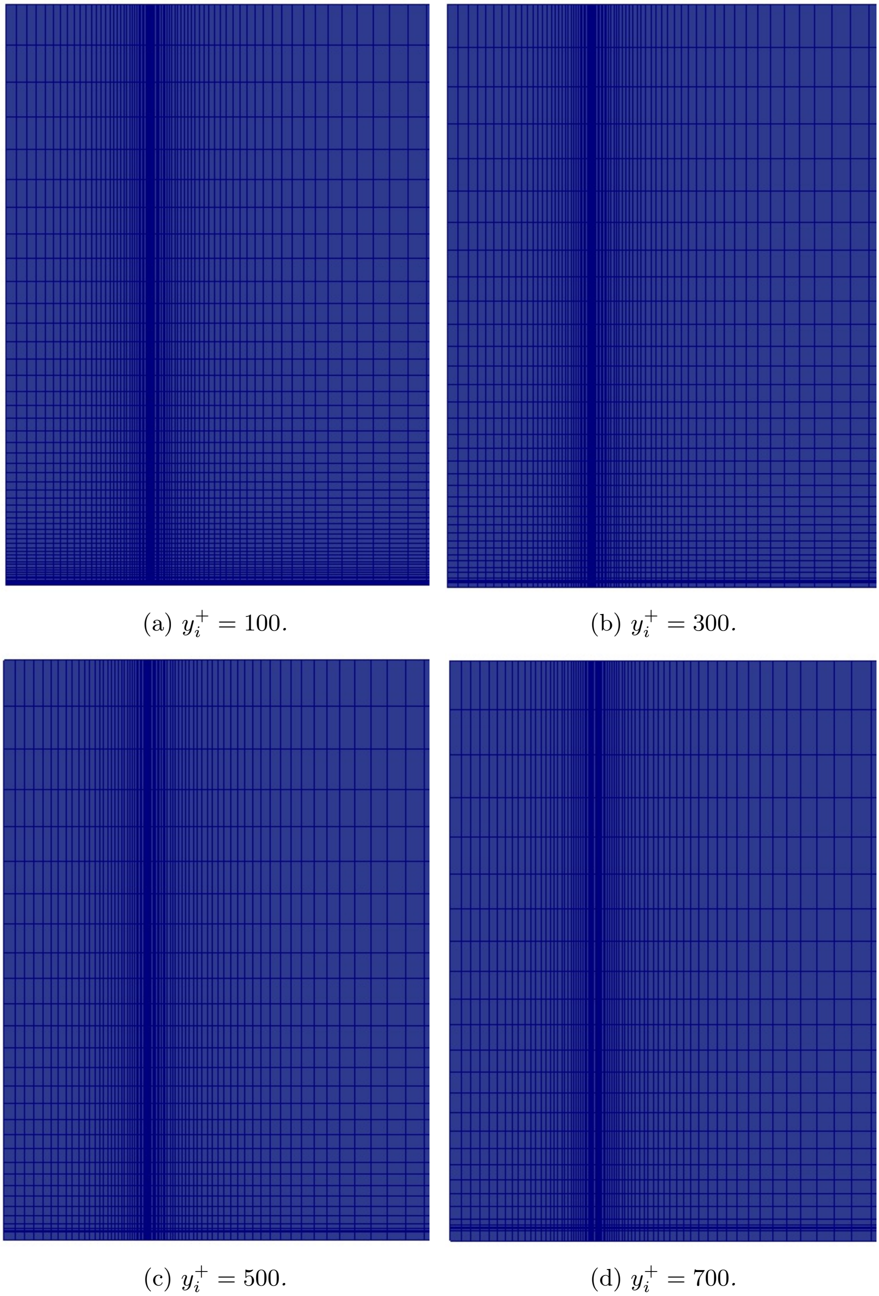
(a) $y_i^+ = 100$.

(b) $y_i^+ = 300$.

(c) $y_i^+ = 500$.

(d) $y_i^+ = 700$.

Figure A.4: *Part of the main-grids used with the sub-grid models, $r = z = 0$ at the lower left corner of the plots, used in the turbulent axisymmetric semi-confined impinging jet at $Re_D = 71000$ for different y_i^+.*



(a) $y_i^+ = 100$.

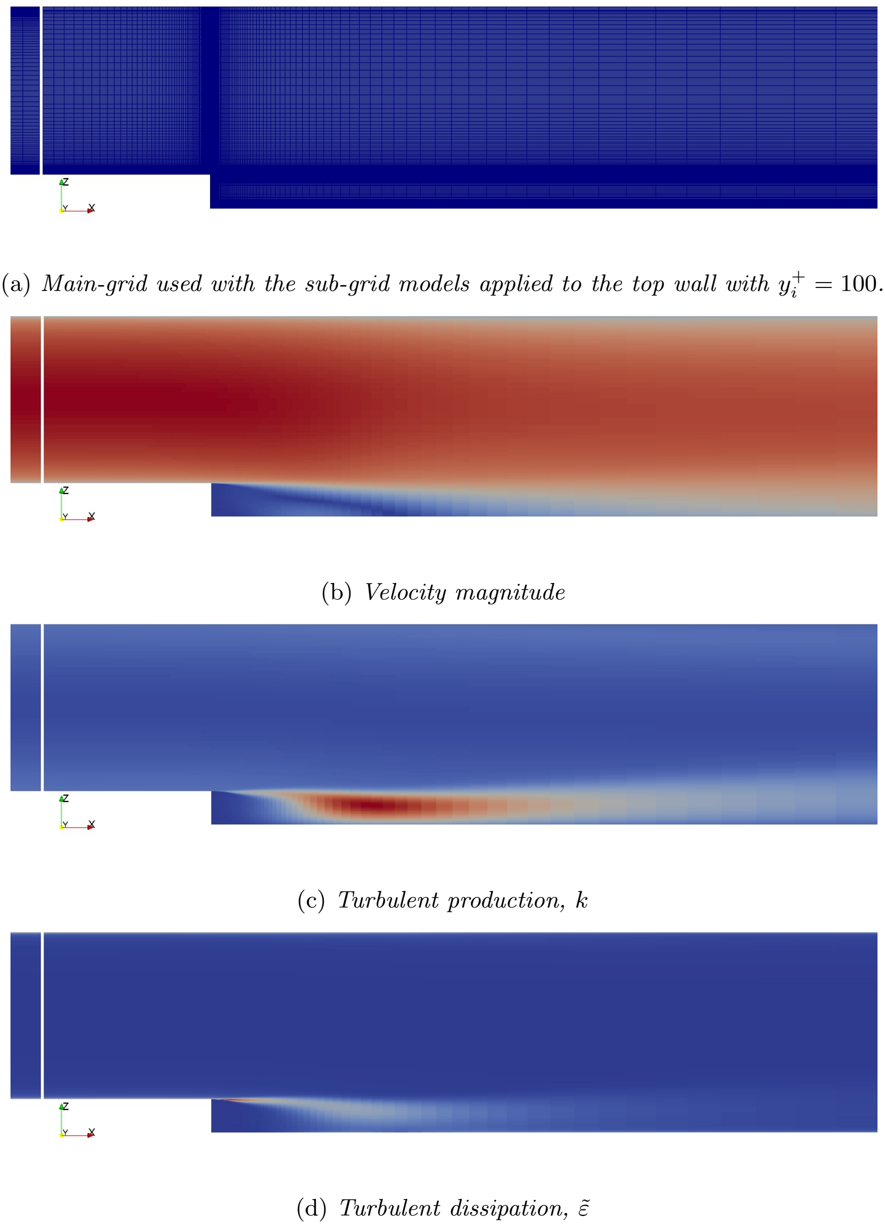
(b) $y_i^+ = 300$.

(c) $y_i^+ = 500$.

(d) $y_i^+ = 700$.

Figure A.5: *Part of the main-grids used with the sub-grid models,* $r = z = 0$ *at the lower left corner of the plots, used in the turbulent axisymmetric semi-confined impinging jet at* $Re_D = 220000$ *for different* y_i^+.

B Backward facing step



(a) *Main-grid used with the sub-grid models applied to the top wall with* $y_i^+ = 100$.

(b) *Velocity magnitude*

(c) *Turbulent production, k*

(d) *Turbulent dissipation, $\tilde{\varepsilon}$*

Figure B.1: *Mesh and flow pattern of the backward facing step at* $Re_H = 25000$ *using the Launder-Sharma model.*