Rept CRS4-APPMATH-93-22, CRS4, Cagliary, Italy,

An Implicit Finite-Volume Solver for the k and ε Equations on Unstructured Meshes With Arbitrary Control Volumes

*Lars Davidson and Luca Stolcis
Applied Mathematics and Simulation Group
CRS4

Abstract

In this work a two-dimensional implicit solver for the $k-\varepsilon$ model on unstructured grids is developed. The solver handles cells of arbitrary form and number of faces. The convective terms are discretized using hybrid upwind/central differencing, and central differences are used for the diffusive terms. The k and ε-equations are solved with point-by-point Gauss-Seidel relaxation. Two different ways are used for treating the wall boundaries: *i)* wall functions, where a rather coarse mesh is used; a *ii)* two-layer model on a fine mesh, in which a one-equation model is employed near the wall and is matched with the standard $k-\varepsilon$ model in the fully turbulent region. This type of solver has previously been used on structured meshes in Refs.1, 2.

The mean flow equations are solved employing an explicit, cell-centered Runge-Kutta solver for unstructured meshes.[3]

The method is used for computing the flow around the RAE2822 airfoil. Several local mesh refinements are done to refine the mesh, demonstrating the flexibility and capability of unstructured methods. The refined mesh is a mixed quadrilateral-triangular mesh, where the number of faces of the cells varies between three and six.

Acknowledgement

The present work was carried out with financial support of the Sardinian Regional Authorities.

*Permanent adress: Thermo and Fluid Dynamics
Chalmers University of Technology S-412 96 Gothenburg, Sweden

Contents

1 Introduction

Computational Fluid Dynamics (CFD) is nowadays being used frequently in the industry. However, when using structured methods, the grid generation for complex geometries remains a major task. Generation of body-conforming grids around three-dimensional bodies usually require considerably much more time in terms of man-power, than the actual flow-field computations. In order to become a useful tool, CFD must be capable of handling complex flow around complex geometries. The lack of generality in treating complex geometries is one of the major reasons which has prevented CFD to become a powerful tool in everyday engineering.

The use of unstructured methods facilitates the grid generation enormously, and automatic methods for triangulation of arbitrary geometries exist.[8] For Navier-Stokes computations of the flow around bodies, a structured body-mesh can be matched with an external (automatically generated) triangulated region.[4]

Local mesh refinement, either adaptive or fixed, is another advantage of unstructured methods. Quadrilaterals are easily split into smaller quadrilaterals or split into triangles, and triangles are readily split into smaller triangles. If, as in the present study, cell-centered methods are adopted, 'hanging' grid points cause no problems, since a cell can have an arbitrary number of grid points (cell vertices).

Considerable work has been presented in the literature on unstructured methods and their applications,[9–12] but only few using transport turbulence models.[4–6]

In the present study a two-dimensional, implicit unstructured solver for the $k-\varepsilon$ turbulence model is developed, which on structured grids has proved to be very stable.[1,2] Hybrid central/upwind differencing is used for the convective terms, and central differencing for the diffusive terms. The solver can handle control volumes of arbitrary number of cell-faces and grid points. The discretized equations are solver by a point-by-point Gauss-Seidel relaxation method. The mean flow equations are solved using an explicit, cell-centered Runge-Kutta method.[3] We present computations around the RAE2822 airfoil. The walls are treated in two ways: either wall functions, using a coarse mesh, or the boundary layer is resolved using a low-Re two-layer $k-\varepsilon$ model. Different meshes are used: only triangles, only quadrilaterals, or mixed mesh containing both triangles and quadrilaterals where the number of faces of the cells varies between three and six.

2 The $k-\varepsilon$ Model

The standard k and ε-equations have the form:

$$\begin{aligned} \frac{\partial}{\partial x_j}\left(\rho U_j k\right) &= \frac{\partial}{\partial x_j}\left[\left(\mu+\mu_t\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon \\ \frac{\partial}{\partial x_j}\left(\rho U_j \varepsilon\right) &= \frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{\varepsilon}{k}\left(c_{1\varepsilon}P_k - c_{2\varepsilon}\rho\varepsilon\right) \end{aligned} \tag{1}$$

and the turbulent viscosity is computed as

$$\mu_t = c_\mu \rho \frac{k^2}{\varepsilon} \tag{2}$$

The production term has the form

$$P_k = \left\{ \mu_t \frac{\partial U_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right\} - \left[\frac{2}{3} \delta_{ij} \frac{\partial U_i}{\partial x_j} \left(\mu_t \frac{\partial U_m}{\partial x_m} + \rho k \right) \right] \tag{3}$$

where the term in square brackets is a dilatation term due to compressibility.

2.1 Wall Functions

When the mesh is coarse close to the walls, wall functions are used. In these the friction velocity u_* is computed (iteratively) from the law of the wall

$$\frac{U_{||}}{u_*} = \frac{1}{\kappa} \log \left(\frac{E u_* n}{\nu} \right) \tag{4}$$

where $U_{||}$ is the velocity component parallel to the wall, n is the normal distance from the wall, and κ, E are constants (=0.41,9.0). From the friction velocity, the wall shear stress τ_w, k and ε are computed as:

$$\begin{aligned} \tau_w &= \rho u_*^2 \\ k &= c_\mu^{-0.25} u_*^2 \\ \varepsilon &= \frac{u_*^3}{\kappa n} \end{aligned}$$

where c_μ is a constant (=0.09). The k and ε equations are not solved at the cells adjacent to the wall, but are fixed according to the formulae above.

2.2 Two-layer Model

If the grid is fine enough, a one-equation model is used near the wall in which the standard k equation is solved, and the turbulent length scales are prescribed as:

$$\ell_\mu = C_\ell n \left[1 - exp(-R_n / A_\mu) \right] \tag{5}$$

$$\ell_\varepsilon = C_\ell n \left[1 - exp(-R_n / A_\varepsilon) \right] \tag{6}$$

so that the dissipation term in the k-equation is obtained as:

$$\varepsilon = \frac{k^{3/2}}{\ell_\varepsilon} \tag{7}$$

and the turbulent viscosity as:

$$\mu_t = c_\mu \sqrt{k} \ell_\mu \tag{8}$$

The Reynolds number R_n and the constants are defined as

$$R_n = \frac{\sqrt{k}n}{\nu}$$

$$C_\ell = \kappa c_\mu^{-3/4}, \quad A_\mu = 70, \quad A_\varepsilon = 2C_\ell$$

The one-equation model is used near the walls, and the standard $k-\varepsilon$ model in the remaining part of the flow. The matching locations are set where the damping function in square brackets in Eq. 5 takes the value 0.95.

3 The Discretization

Let Φ denote a general variable ($\Phi = k$ or ε). The transport equations for Φ can then be written as

$$\frac{\partial}{\partial x_m}(\rho U_m \Phi) = \frac{\partial}{\partial x_m}(\Gamma_\Phi \frac{\partial \Phi}{\partial x_m}) + \overline{S}^\Phi \tag{9}$$

where $\overline{S}^\Phi$ denotes source per unit volume. If a flux vector J_m containing convection and diffusion is defined as

$$J_m = \rho u_m \Phi - \Gamma_\Phi \frac{\partial \Phi}{\partial x_m} \tag{10}$$

Eq. 9 can be written as:

$$\frac{\partial J_m}{\partial x_m} = \overline{S}^\Phi$$

In vector notation the equation reads:

$$\nabla \cdot \mathbf{J} = \overline{S}^\Phi$$

Integrating this equation over a volume (with volume V and bounding surface A) using Gauss' law, gives:

$$\int_A \mathbf{J} \cdot \mathbf{dA} = \int_V \overline{S}^\Phi dV$$

which for a control volume gives

$$\sum_{i=1}^{N} \{\mathbf{J} \cdot \mathbf{A}\}_i = S^\Phi \tag{11}$$

where N is the (arbitrary) number of faces of a cell, and S^Φ is the total source in the control volume.

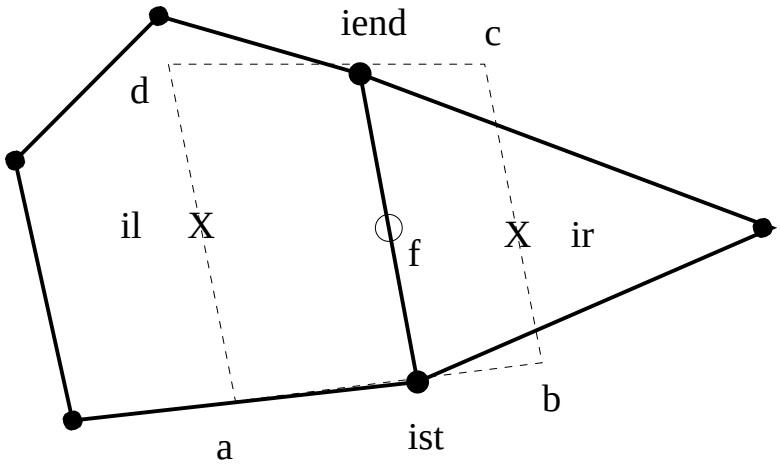

Figure 1: *Two control volumes (cells). Grid points (cell vertices) are marked with black dots (•), and cell centres are marked with X's. The area marked with dashed lines ($a-b-c-d$) is used when computing the gradient at the face (open circle, index* f*).*

3.1 Pointer System

Since we are dealing with unstructured meshes, we need a pointer system which carries information on grid topology. Almost all quantities (e.g. convection, diffusion, derivatives) are computed by looping over cell faces. We thus need a pointer for each face, which gives information about which grid points (cell vertices) form the starting and end points of the face, and which two cells are the adjacent cells. This information is stored in `plist`, which has the form[3]

`plist(i,1)` $=il$

`plist(i,2)` $=ist$

`plist(i,3)` $=iend$

`plist(i,4)` $=ir$

where (see Fig. 1) i is the index of the face, ist and $iend$ are the starting and ending grid points, and il and ir are the left and right cells. Note that the vector product between the vectors $(il)\vec{(}ir)$ and $\hat{z}$ (where $\hat{x}-\hat{y}-\hat{z}$ defines the three Cartesian unit vectors) is positive.

When solving the discretized equations, we need also a pointer from a cell to its neighbouring cells. The number of neighbours is arbitrary, and for cell i we store it as:

`info(i,1)` = neighbour 1

`info(i,2)` = neighbour 2

`info(i,.)` = neighbour .

`info(i,n)` = neighbour n

On a mesh containing only triangles we need to store three items in `info` (`plist` contains always four items, irrespectively of type of control volume), for quadrilaterals we have to store one more, and so on. Note that `plist` is stored for each *face*, whereas `info` is stored for each *cell*. On a mesh containing only triangles (quadrilaterals), the number of cell faces is approximately 1.5 (2) times number of cells.

3.2 Convection

The convection, which is the first part of the flux vector $\mathbf{J}$ in Eq. 10, is the scalar product of the velocity vector and the area vector multiplied by the density and the variable Φ. It is computed by looping over faces. For the cell face in Fig. 1 we get:

$$\{\dot{m}\Phi\}_f = \{\rho\mathbf{u}\cdot\mathbf{A}\Phi\}_f = \{\rho\left(u_x A_x + u_y A_y\right)\Phi\}_f \tag{12}$$

where the face values of ρ, and (u_x, u_y) are computed as

$$\begin{aligned}
(u_x)_f &= f_1(u_x)_{il} + (1-f_1)(u_x)_{ir} \\
(v_x)_f &= f_1(v_x)_{il} + (1-f_1)(v_x)_{ir} \\
\rho_f &= f_1\rho_{il} + (1-f_1)\rho_{ir}
\end{aligned}$$

where f_1 is a weight function for linear interpolation.

3.3 Diffusion

Diffusion is the second part of the flux vector $\mathbf{J}$ in Eq. 10, and it has the form:

$$\mathcal{D} = (\mathbf{J}\cdot\mathbf{A})_{\text{diff}} = -\Gamma_\Phi\mathbf{A}\cdot\nabla\Phi$$

For the face in Fig. 1 it gives

$$\begin{aligned}
-\{\Gamma_\Phi\mathbf{A}\cdot\nabla\Phi\}_f &= -\left\{\Gamma_\Phi\left(A_x\frac{\partial\Phi}{\partial x} + A_y\frac{\partial\Phi}{\partial y}\right)\right\}_f = \\
&- \left\{\Gamma_\Phi\left|\mathbf{A}\right|\left(n_x\frac{\partial\Phi}{\partial x} + n_y\frac{\partial\Phi}{\partial y}\right)\right\}_f
\end{aligned} \tag{13}$$

where (n_x, n_y) is unit normal vector of the face, computed as the vector product between the vectors $(il)\vec{(}ir)$ and $\hat{z}$, i.e.

$$\begin{aligned}
n_x &= y_{iend} - y_{ist} \\
n_y &= -(x_{iend} - x_{ist})
\end{aligned}$$

Now we need to evaluate the derivatives $\partial\Phi/\partial x$ and $\partial\Phi/\partial y$ at the face. We do that by applying Green's formula to the volume $a-b-c-d$ (see Fig. 1) surrounding the midpoint of the face, i.e.

$$\frac{\partial\Phi}{\partial x} = \frac{1}{V}\int_S \Phi dS$$

where S $(= a-b-c-d)$ is the surface enclosing the volume V. We get

$$\begin{aligned}\left[\frac{\partial\Phi}{\partial x}n_x\right]_f &= \left\{\frac{n_x}{V}\right\}_f \{(\Phi n_x)_{ir} + (\Phi n_x)_{iend} - (\Phi n_x)_{il} - (\Phi n_x)_{ist}\} \\ \left[\frac{\partial\Phi}{\partial y}n_y\right]_f &= \left\{\frac{n_y}{V}\right\}_f \{(\Phi n_y)_{ir} + (\Phi n_y)_{iend} - (\Phi n_y)_{il} - (\Phi n_y)_{ist}\}\end{aligned} \tag{14}$$

Since the volume $a-b-c-d$ is chosen so that $(n_x)_{il} = (n_x)_f = (n_x)_{ir}$, $(n_y)_{il} = (n_y)_f = (n_y)_{ir}$, $(n_x)_{ist} = (n_x)_{iend}$, and $(n_y)_{ist} = (n_y)_{iend}$, the sum of the two terms in Eq. 14 can be written

$$\begin{aligned}&\left[\frac{\partial\Phi}{\partial x}n_x\right]_f + \left[\frac{\partial\Phi}{\partial y}n_y\right]_f = \\ &= \left\{\frac{1}{V}\right\}_f \left\{\left[n_x^2 + n_y^2\right]_f (\Phi_{ir} - \Phi_{il})\right. \\ &\left. + \left[(n_x)_f (n_x)_{iend} + (n_y)_f (n_y)_{iend}\right](\Phi_{iend} - \Phi_{ist})\right\}\end{aligned} \tag{15}$$

The second line represents orthogonal diffusion, and the last line represents non-orthogonal diffusion which vanishes on orthogonal grids. Eqs. (13-15) now can be written:

$$\begin{aligned}&-\{\Gamma_\Phi \mathbf{A}\cdot\nabla\Phi\}_f = \\ &-\left\{\frac{\Gamma_\Phi|\mathbf{A}|}{V}\right\}_f \left\{\left[n_x^2 + n_y^2\right]_f (\Phi_{ir} - \Phi_{il})\right. \\ &\left. + \left[(n_x)_f (n_x)_{iend} + (n_y)_f (n_y)_{iend}\right](\Phi_{iend} - \Phi_{ist})\right\}\end{aligned} \tag{16}$$

where the second line will be treated implicitly, and the last line will be treated explicitly, using values at previous time level n. For the sake of conciseness, we rewrite the orthogonal part of diffusive as

$$\{\Gamma_\Phi \mathbf{A}\cdot\nabla\Phi\}_{f,\mathrm{ort}} = D_f (\Phi_{ir} - \Phi_{il}) \tag{17}$$

3.4 The discretized equation

Combining Eqs. 11, 12 and 17 for a triangular cell (see Fig. 2) gives

$$\begin{aligned}&\{\dot{m}_{if1}\Phi_{if1} - D_{if1}(\Phi_{i1} - \Phi_i)\} + \{\dot{m}_{if2}\Phi_{if2} - D_{if2}(\Phi_{i2} - \Phi_i)\} + \\ &\{\dot{m}_{if3}\Phi_{if3} - D_{if3}(\Phi_{i3} - \Phi_i)\} = S^\Phi\end{aligned} \tag{18}$$

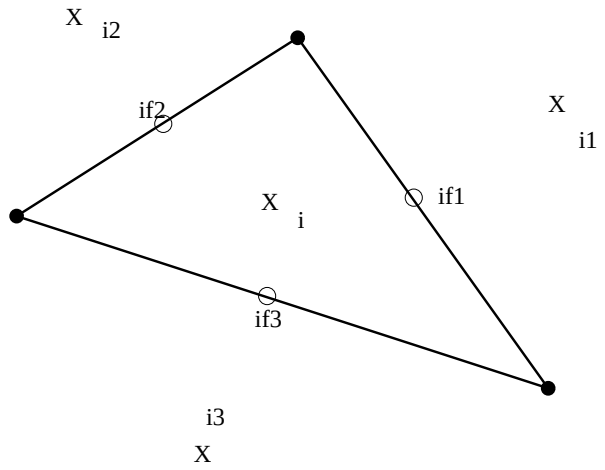

Figure 2: *A triangular control volume* i *with its three neighbours* i1, i2 *and* i3. *Grid points are marked with black dots (•), cell centres with X's, and cell faces with open circles.*

where the non-orthogonal diffusion terms have been included in the source term. The cell face values of Φ are estimated with hybrid central/upwind differencing.[7,16] The resulting discretized equation reads[7]

$$\begin{aligned} a_P\Phi_i &= a_{if1}\Phi_{i1} + a_{if2}\Phi_{i2} + a_{if3}\Phi_{i3} + S_U^\Phi \\ a_{if1} &= \max\left\{-\frac{1}{2}\dot{m}_{if1} + D_{if1}, 0, -\dot{m}_{if1}\right\} \\ a_{if2} &= \max\left\{-\frac{1}{2}\dot{m}_{if2} + D_{if2}, 0, -\dot{m}_{if2}\right\} \\ a_{if3} &= \max\left\{-\frac{1}{2}\dot{m}_{if3} + D_{if3}, 0, -\dot{m}_{if3}\right\} \\ a_P &= a_{if1} + a_{if2} + a_{if3} - S_P \end{aligned} \tag{19}$$

where n denotes number of neighbours (for the triangle $n = 3$).

The coefficients in Eq. 19 are calculated by looping over cell faces. We have a cell counter which is augmented by one every time a cell is adjacent to a visited cell face, and it is this counter which determines the values of $i1, i2, i3$ ($1 \geq i1, i2, i3 \geq 3$). The discretized equation above is given for a control volume with three faces, but the control volumes can have arbitrary number of faces. In the refined grid low-Re number grid (see below), the number of faces of the control volumes in the mesh varies between three and six.

The treatment of the source terms are described in detail in Ref. 7.

3.5 Problems with Very Stretched Triangular Cells

Problems have been experiences with very stretched triangular cells. When the boundary layer is resolved using the two-layer $k - \varepsilon$ model, we have cells with very high aspect ratios ($\simeq 500$). This gives problems with regard of the diffusion term $|\mathbf{A}|\mathbf{n} \cdot \nabla\Phi$, which on the face where it is to be evaluated is decomposed on the two coordinate directions defined by $(il)\vec{(}ir) = \hat{\xi}$ and $(ist)\vec{(}iend) = \hat{\eta}$, see Fig. 3. When the cells are very stretched the angle γ approaches 180^o, and

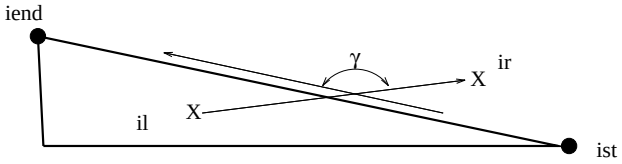

Figure 3: *A stretched triangular control volume. Grid points are marked with black dots (•), cell centres with X's, and cell faces with open circles. The two coordinate directions for computing the diffusion term* $|\mathbf{A}|\mathbf{n}\cdot\nabla\Phi$ *are defined by* $(il)\vec{(}ir) = \hat{\xi}$ *and* $(ist)\vec{(}iend) = \hat{\eta}$.

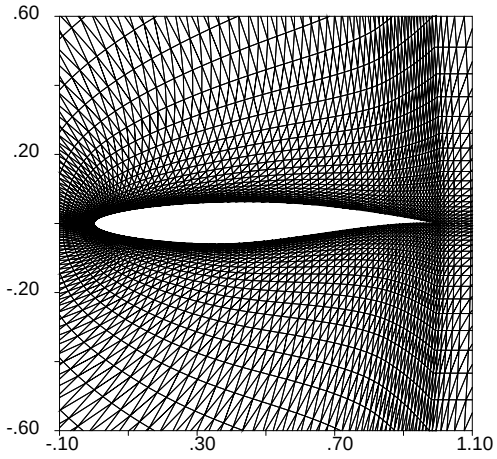

Figure 4: *A* 200×40 *mesh where each quadrilateral is split into two triangles. Wall functions are used on this mesh.*

the $\hat{\xi} - \hat{\eta}$ becomes very il-conditioned, since $\hat{\xi}$ and $\hat{eta}$ become almost parallel ($\hat{\xi}\cdot\hat{\eta} > 0.99$). In the present work this problem is solved by using quadrilaterals near the wall, where the aspects ratios are large.

4 The RAE2822 airfoil

The RAE2822 airfoil is used as test case. The experiments are taken from Cook *et al.*[19] Transition is imposed according to experiments at the lower and upper side at $x = 0.03$.

4.1 Wall functions

When computing this flow, we start with a structured C-mesh, where each quadrilateral is split into two triangles, see Fig. 4. The mesh is fairly coarse near the airfoil, and we use wall functions when computing the wall stress in the mean flow equations, and k and ε are fixed from the friction velocity u_*, see Section 2.1. In Figs. 5 and 6 the wall pressure, and the convergence history are shown. For comparison, results obtained solving the k and ε-equations with an explicit Runge-Kutta solver[5,17,18] (the same solver as for the mean-flow equations),

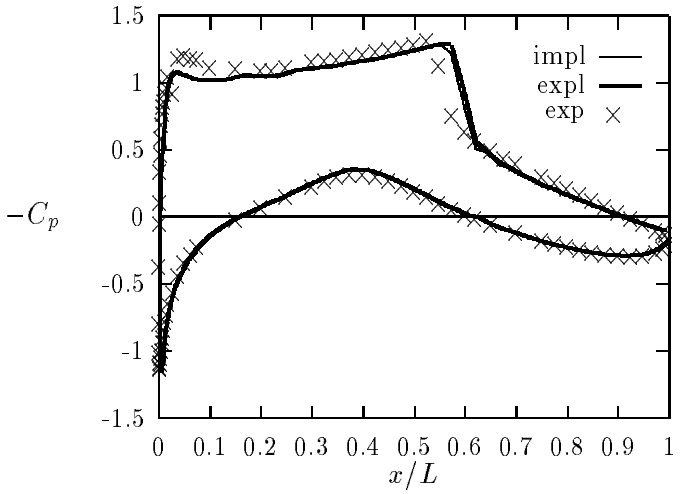

Figure 5: Pressure coefficient. Comparison between implicit and explicit solver for the k and ε-equations. Wall functions are used.

are also included. As can be seen from Fig. 5, the results are more or less identical (as they should; the same turbulence model is used, but the discretized equations are solved with two different solvers), and from Fig. 6 we see that the system of equations (mean-flow equations and turbulence model) converge slightly better when the implicit solver is used.

4.2 Low-Re Two-Layer Model

Here we use meshes which are finer near the wall, and we use the Low-Re two-layer $k-\varepsilon$ model (see Section 2.2). The first mesh is a structured C-mesh, refined near the wall, with 200×40 mesh lines (quadrilaterals). For reasons explained in Section 3.5, it is not triangulated. This mesh is too coarse, and the (negative) pressure peak at the leading edge is not captured (see Fig. 7), and as a result the shock is predicted earlier. It seems that we obtain a good prediction of the shock (see Fig. 7), but is purely coincidental, and is a result of the poor prediction ahead of the shock.

In order to create a better mesh, it is refined. The mesh is refined in three steps:

Refinement 1: on the upper side of the airfoil ($0.4 \leq x \leq 1.3$; $0.02 \leq n \leq 0.15$) [n is normal distance from the wall] are split into two triangles; the region close to the wall is excluded in order to avoid very stretched triangles (see Section 3.5)

Refinement 2: a partition of the refined triangles ($0.5 \leq x \leq 0.9$; $0.02 \leq n \leq 0.08$) are split into four new triangles

Refinement 3: the cells close to the wall ($0 \leq x \leq 1$; $0 \leq n \leq 0.02$) are split into four quadrilaterals.

The three different types of grid refinements are shown in Fig. 8.

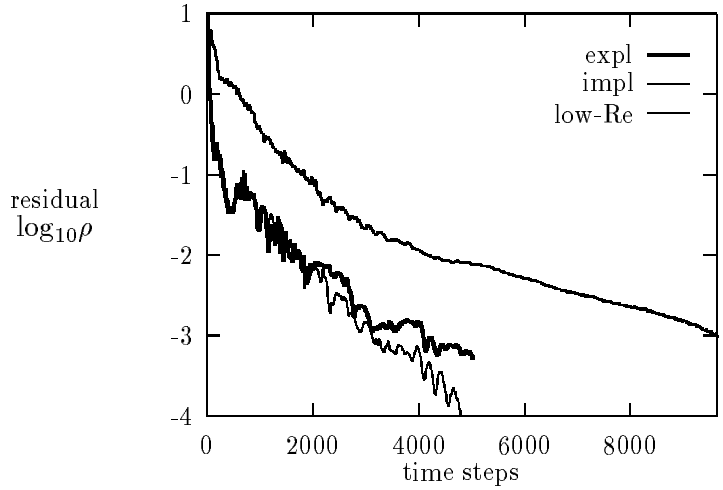

Figure 6: Convergence history comparing the explicit and the implicit $k-\varepsilon$ solver. Also the low-Re $k-\varepsilon$ convergence history (implicit solver) on the refined mesh is shown. On the ordinate the ρ rms-residual is shown.

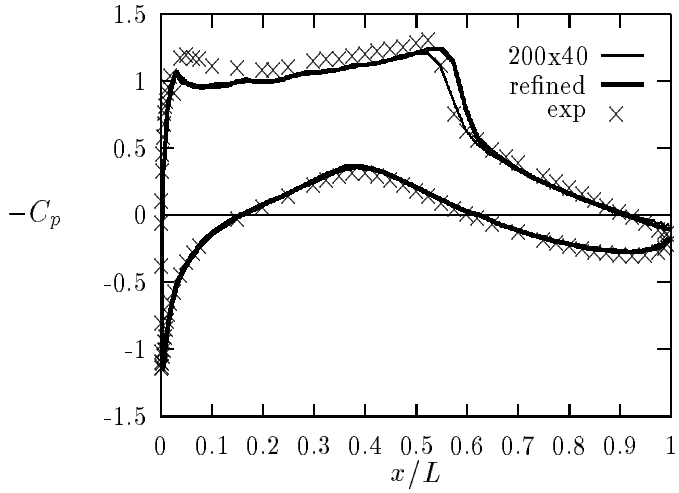

Figure 7: Pressure coefficient. Comparison between calculations on the 200×40-low-Re mesh and the refined mesh. Low-Re two-layer model is used together with the implicit solver.

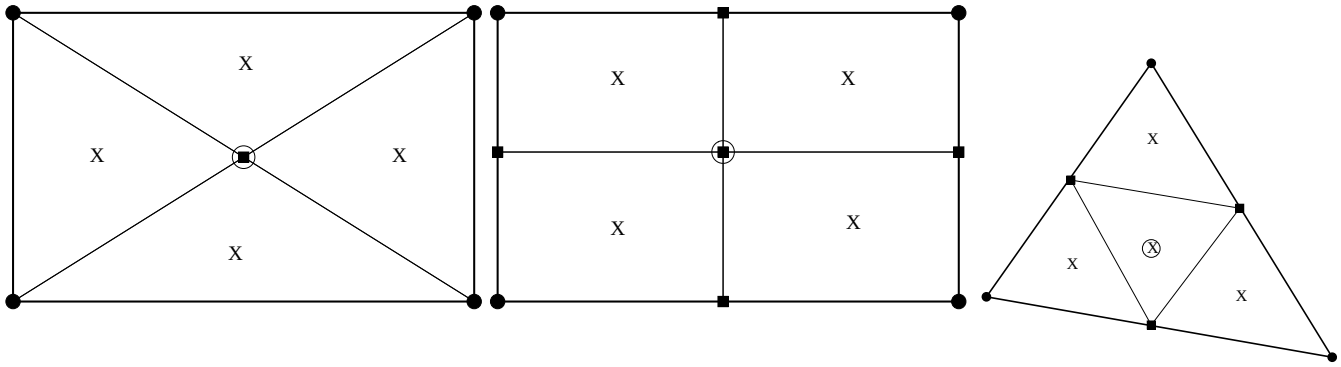

Figure 8: *Three types of grid refinements.* ●*: grid point (cell vertex); X: new cell;* ◯*: deleted cell;* ■*: new grid point.*

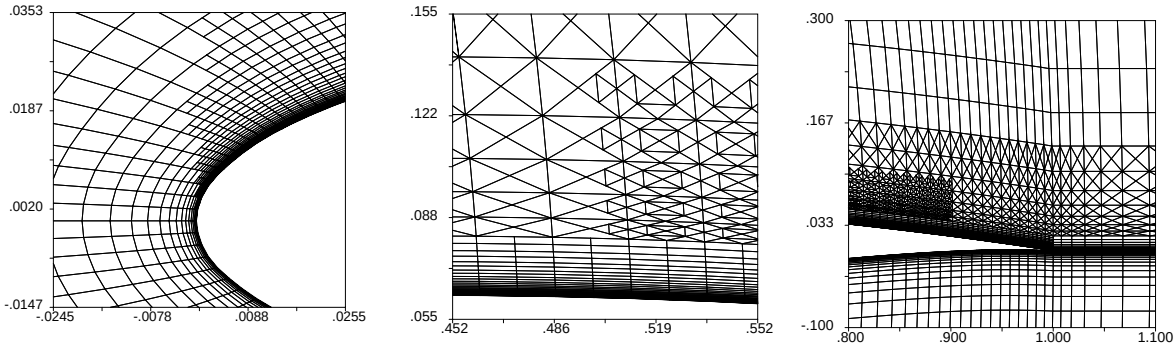

Figure 9: *Blowups of the refined mesh. To the left: the leading edge; in the middle: the mesh in the shock region; to the right: the trailing edge.*

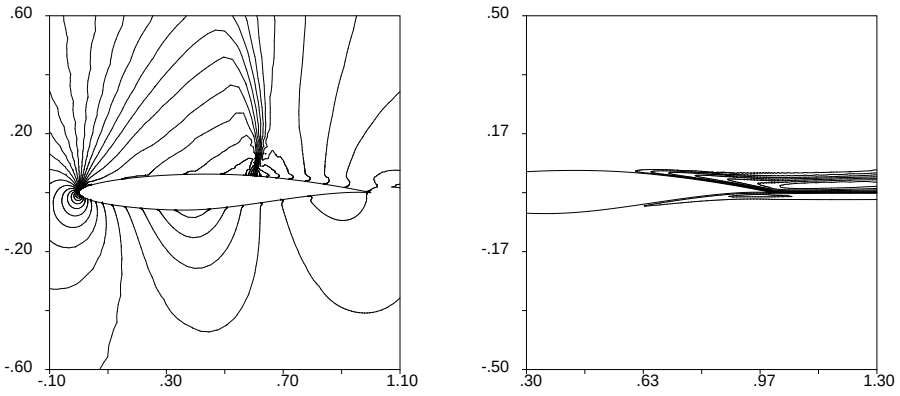

Figure 10: *Contours of pressure (left) and turbulent viscosity (right) contours. Refined mesh.*

Blowups of the refined mesh are shown in Fig. 9.

From Fig. 7 it is seen that the refined mesh gives a shock location at the same position as the wall functions. However, the pressure peak is still too low. Thus it seems that the wall functions give a better representation of the flow ahead of the shock. However, using the low-Re model should give a more accurate representation of the boundary layer flow, since we resolve the boundary layer. The transition process is probably predicted quite differently with the wall functions and with the low-Re model.

The contours of pressure and turbulent viscosity are presented in Fig. 10, and it is seen that the contours are smooth, also across the mesh refinements.

5 Summary

An implicit solver for the $k-\varepsilon$ model on two-dimensional general unstructured grids has been presented. The discretized equations are solved using point-by-point Gauss-Seidel relaxation. The mean flow equations are solved using an explicit, cell-centred Runge-Kutta solver. The flow around the RAE2822 airfoil has been computed on various meshes.

i) A coarse triangulated mesh where wall functions were used for treating the wall boundary.

ii) A finer mesh containing only quadrilaterals, on which a low-Re two-layer $k-\varepsilon$ model was used.

iii) Mesh *ii)* refined; it was refined near the wall, splitting the quadrilaterals into four smaller ones, and the quadrilaterals in the shock region were split, in two steps, into smaller triangles. The number of faces of the control volumes vary between three and six.

Problems were encountered when very stretched triangles were used near the wall. The problem was traced to the diffusion term. At the contol volumes faces the gradient is decomposed onto two coordinate axis $\hat{\xi}-\hat{\eta}$ defined by the two cell-centres, and the direction of the face. On stretched triangles the $\hat{\xi}-\hat{\eta}$ system becomes very il-conditioned, since $\hat{\xi}$ and $\hat{\eta}$ are almost parallel ($\hat{\xi}\cdot\hat{\eta}>0.99$). This problem was avoided by using quadrilaterals near the wall.

References

[1] DAVIDSON, L. and RIZZI, A. – Navier-Stokes Stall Predictions Using an Algebraic Stress Model", *J. Spacecraft and Rockets*, **29**, pp. 794-800, 1992 (see also AIAA-paper 92-0195, Reno, Jan. 1992).

[2] DAVIDSON, L. – Reynolds Stress Transport Modelling of Shock/Boundary-Layer Interaction, AIAA-paper 93-2936, AIAA 24th Fluid Dynamics Conference, Orlando, July 1993.

[3] STOLCIS, L. and JOHNSTON, L.J. – Solution of the Euler Equations on Unstructured Grids for Two-Dimensional Compressible Flow, *Aeronat. J*, pp. 181-195, June/July 1990.

[4] HOLMES, D.G. and CONNEL, S.D. – Solution of the 2D NAvier-Stokes Equations on Unstructured Adaptive Grids, AIAA-paper 89-1932-CP, 1989.

[5] STOLCIS, L. and JOHNSTON, L.J. – Near-wall turbulence models and numerical solution of the Reynolds-averaged Navier-Stokes equations using unstructured grids, Lecture Notes in Physics, **414**, pp. 200-204, Napolitano M. and Sabetta F. Editors, Springer Verlag, 1993

[6] MAVRIPLIS, D.J. and MARTINELLI, L. – Multigrid Solution of Compressible Turbulent Flow on Unstructured Meshes Using a Two-Equation Model, AIAA-paper 91-0237, 1991.

[7] DAVIDSON, L. – Implementation of a $k-\varepsilon$ Model and a Reynolds Stress Model into a Multiblock Code, Rept. CRS4-APPMATH-93-21, CRS4, Cagliari, 1993.

[8] BARTH, T.J. – On Unstructured Grids and Solvers, in *Computational Fluid Dynamics*, Lecture Notes 1990-03, von Karman Institute for Fluid Dynamics, March 5-9, 1990.

[9] HWANG, C.J. and YANG, S.Y. – Locally Implicit Total Variation Diminishing Schemes on Mixed Quadrilateral-Triangular Meshes, *AIAA J.* **31**, pp. 2008-2015, 1993.

[10] MATHUR, J.S. and WEATHERILL, N.P. – The Simulation of Inviscid, Compressible Flows Using an Upwind Kinetic Method on Unstructured Grids, *Int. J. Numer. Meth. Fluids*, **15**, pp. 59-82, 1992.

[11] PAN, D. and CHENG, D.P. – Upwind Finite-Volume Navier-Stokes Computations on Unstructured Triangular Meshes, *AIAA J.*, **31** pp. 1618-1625, 1993.

[12] KALLINDERS, Y. – A Finite Volume Navier-Stokes Algorithm for Adaptive Grids, *Int. J. Numer. Meth. Fluids*, **15**, pp. 193-217, 1992.

[13] WOLFSHTEIN, M. – The Velocity and Temperature Distribution in One-Dimensional Flow with Turbulence Augmentation and Pressure Gradient, *Int. J. Mass Heat Transfer*, **12**, pp. 301-318, 1969.

[14] CHEN, H.C. and PATEL, V.C. – Practical Near-Wall Turbulence Models for Complex Flows Including Separation, AIAA Paper No 87-1300, Honolulu, June 1987.

[15] HUANG, P.G. and LESCHZINER, M.A. – Stabilisation of Recirculating-Flow Performed with Second-Moment Closures and Third-Order Discretization, 5th Turbulent Shear Flow, pp. 20.7-20.12, Cornell, 1985.

[16] PATANKAR, S.V. – *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.

[17] STOLCIS, L. and JOHNSTON, L.J. – Compressible Flow Calculations Using a Two-Equation Turbulence Model and Unstructured Grids, Numerical Methods in Laminar and Turbulent Flow, Vol. 7, pp. 922-931, Taylor C., Chin J.H. and Homsy G.M Editors, Pinerdge Press, 1991

[18] STOLCIS, L. and JOHNSTON, L.J. – Computation of the Viscous Flow Around Multi-Element Aerofoils Using Unstructured Grids, Notes on Numerical Fluid Mechanics, **35**, pp. 311-320, Vos J.B, Rizzi A. and Ryhming I.L. Editors, Vieweg Verlag, 1992

[19] Cook P.H., McDonald M.A. and Firmin, M.C.P – Aerofoil RAE 2822 - Pressure Distributions and Boundary Layer and Wake Measurements, AGARD AR 138, pp. A6-1 to A6-77, 1979.