Rept. CRS4-APPMATH-93-21 CRS4 Cagliary, Italy 1993

Implementation of a $k-\varepsilon$ Model and a Reynolds Stress Model into a Multiblock Code

*Lars Davidson
Applied Mathematics and Simulation Group
CRS4

Abstract

This report describes an implementation of a $k-\varepsilon$ model and a Reynolds Stress Model (RSM) into the `tharros` multiblock code. The `tharros` code, developed by Mulas,[1] is an explicit, compressible, time-marching finite volume code.

The turbulence models have been implemented in such a way that they form a separate part of the program. Input data such as dependent variables (ρ, u, v), laminar viscosity, grid geometry (coordinates, normal vectors of control volume faces), velocity gradient, boundary conditions, are supplied by `tharros` in the `call`-statements, and the turbulence models return turbulent viscosity and, if RSM is used, the Reynolds stresses.

The equations for the turbulent quantities are solved by an implicit finite volume method, which previously has been used by the author.[2,3] A tri-diagonal-matrix-algorithm (TDMA) is used for solving the discretized equations. The convective terms are discretized with upwind/central differencing, and central differencing is used for the diffusive terms.

The fact that a compleatly different method is used for solving the turbulent quantities has both advantages and disadvantages. The implicit solver for the turbulent quantities is more stable than the explicit one. Since the turbulence models reside in a separate part of the program, communicating only via the parameters in the `call`-statements, it is fairly easy to port the implemented turbulence models to another code. The disadvantage is that since the turbulence models are solved with a different method than the mean flow equations, the turbulence models are not well incorporated into the code, and the code is larger than it would have had to be if the explicit solver had been used also for the turbulent quantities.

Acknowledgment

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 Two-layer $k-\varepsilon$ Model **3**

2 The Reynolds Stress Model (RSM) 4
- 2.1 Near Wall Modelling . . . 5
- 2.2 The k and ε-equations . . . 5

3 The Discretization **6**
- 3.1 Convection . . . 7
- 3.2 Diffusion . . . 7
- 3.3 The discretized equation . . . 8
- 3.4 Source terms . . . 10

4 Solver **10**

5 Description of Subroutines **12**
- 5.1 Subroutine `deriv` . . . 12
- 5.2 Subroutine `tratur` . . . 12
  - 5.2.1 Using $k-\varepsilon$ model . . . 12
  - 5.2.2 Using the Reynolds Stress Model (RSM) . . . 14
- 5.3 Subroutine `keps` . . . 14
- 5.4 Subroutine `rsm` . . . 15

6 Test case: Bump A **16**
- 6.1 The Input File `*.bc` . . . 16
- 6.2 Results . . . 17

1 Two-layer $k-\varepsilon$ Model

The two-layer model by Chen & Patel has been used.[6] In the fully turbulent region, the standard $k-\varepsilon$ model is used. The standard k and ε-equations have the form:

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

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.

Near the walls the standard k equation is solved, and the turbulent length scales are prescribed as:

$$\ell_\mu = C_\ell n\,[1 - exp(-R_n/A_\mu)] \tag{4}$$

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

(n is the normal distance from the wall) so that the dissipation term in the k-equation is obtained as:

$$\varepsilon = \frac{k^{3/2}}{\ell_\varepsilon} \tag{6}$$

and the turbulent viscosity as:

$$\mu_t = c_\mu \sqrt{k}\,\ell_\mu \tag{7}$$

The Reynolds number R_n and the constants are defined as

$$\begin{aligned} R_n &= \frac{\sqrt{k}n}{\nu} \\ C_\ell &= \kappa c_\mu^{-3/4}, \quad A_\mu = 70, \quad A_\varepsilon = 2C_\ell \end{aligned}$$

The one-equation model is used near the walls, and the standard $k-\varepsilon$ model in the remaining part of the flow. The matching line is either set at a pre-selected grid line, or where the damping function in square brackets in Eq. 4 takes the value 0.9. See Section 5.2 for further details.

2 The Reynolds Stress Model (RSM)

When computing compressible flow either mass-averaging (compressible) or time-averaging (incompressible) variables can be used. In weakly compressible flow (Mach-number below 2), as in the present study, the incompressible time-averaging concept can be used.

The Reynolds Stress Model neglecting compressibility has the form:[7]

$$\begin{aligned} \underbrace{\frac{\partial}{\partial x_k}\left(\rho U_k \overline{u_i u_j}\right)}_{\text{convection}} &= \underbrace{-\rho\overline{u_i u_k}\frac{\partial U_j}{\partial x_k} - \rho\overline{u_j u_k}\frac{\partial U_i}{\partial x_k}}_{\text{production } P_{ij}} \\ &+ \Phi_{ij} + D_{ij} - \rho\varepsilon_{ij} \end{aligned} \tag{8}$$

The convection and production terms are exact and do not need any modelling assumptions. The pressure strain Φ_{ij} and the dissipation ε_{ij} are modelled as:

$$\begin{aligned} \Phi_{ij} &= \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,w1} + \Phi_{ij,w2} \\ \Phi_{ij,1} &= -c_1\rho\frac{\varepsilon}{k}\left(\overline{u_i u_j} - \frac{2}{3}\delta_{ij}k\right) \\ \Phi_{ij,2} &= -c_2\left(P_{ij} - \frac{2}{3}\delta_{ij}P_k\right) \\ P_k &= \frac{1}{2}P_{jj} \\ \Phi_{ij,w1} &= c_1'\frac{\varepsilon}{k}(\overline{u_k u_m}n_k n_m\delta_{ij} - \frac{3}{2}\overline{u_k u_i}n_k n_j - \frac{3}{2}\overline{u_k u_j}n_i n_k)f\left(\frac{\ell_t}{x_n}\right) \\ \Phi_{ij,w2} &= c_2'\frac{\varepsilon}{k}(\Phi_{km,2}n_k n_m\delta_{ij} - \frac{3}{2}\Phi_{ik,2}n_k n_j - \frac{3}{2}\Phi_{kj,2}n_i n_k)f\left(\frac{\ell_t}{x_n}\right) \\ \varepsilon_{ij} &= \frac{2}{3}\delta_{ij}\varepsilon \end{aligned}$$

The f-function is a damping function which reduces the effect of the wall correction with increasing distance x_n (n denotes normal direction), and which has the form $f = k^{3/2}/(2.55x_n\varepsilon)$.

The diffusion term D_{ij} is modelled using the simple eddy viscosity assumption

$$D_{ij} = \frac{\partial}{\partial x_m}\left(\mu_t\frac{\partial\Phi}{\partial x_m}\right)$$

In previous work[2,3] this simple eddy viscosity model has been compared with the more general GGDH-model by Daly & Harlow,[4] and the difference in the obtained results was found to be negligible.

The Reynolds stresses are stored at the cell centres. In implicit SIMPLE codes apparent viscosity is used in the momentum equations to enhance stability via stability-promoting second derivatives.[8] When using Runge-Kutta solvers for the momentum equations, no such stability

promoting remedies are needed, because the mean flow variables are solved *explicitly*, which means that all terms are at the right-hand-side of the discretized equations, and thus the solution procedure is not sensitive to the explicit adding of the Reynolds stresses.

2.1 Near Wall Modelling

Near the walls the one-equation is used, and the standard high-Re RSM is used in the remaining part of the flow.

The matching of the one-equation model and the k and ε-equations does not pose any problems but gives a smooth distribution of μ_t and ε across the matching line. If the Reynolds stresses in the one-equation region are computed from Boussinesq approximation, this gives rise to non-continuity in the Reynolds stresses across the matching line,[2] since the one-equation model gives more or less isotropic normal Reynolds stresses, whereas RSM gives highly non-isotropic Reynolds stresses, which results in a jump in the profiles of $\overline{u^2}$ and $\overline{v^2}$ across the matching line. This problem is less serious for the shear stress. In the present study the stresses in the one-equation region are computed as:

$$\begin{aligned}
\rho\overline{uv} &= -\mu_t\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right) \\
\rho\overline{u^2} &= \rho k\left(\frac{\overline{u^2}}{k}\right)_{\text{outer}} \\
\rho\overline{v^2} &= \rho k\left(\frac{\overline{v^2}}{k}\right)_{\text{outer}}
\end{aligned}$$

where μ_t is calculated using Eq. 7, and $(\overline{u^2}/k)_{\text{outer}}$ and $(\overline{v^2}/k)_{\text{outer}}$ are the ratios of $\overline{u^2}$, $\overline{v^2}$ and k just outside the matching line, and these ratios are supposed to remain constant all the way to the wall. In this way the distribution of the stresses is smooth across the matching line. It seems, however, that the velocity field is fairly insensitive to the treatment of the normal stresses in the one-equation region.

2.2 The k and ε-equations

In a RSM there are two equivalent sets of equations which can be solved. Either $\overline{u^2}, \overline{v^2}, \overline{w^2}, \overline{uv}$ and ε or $\overline{u^2}, \overline{v^2}, k, \overline{uv}$ and ε. The latter set has been chosen in the present study. The main reason for this choice is that the k-equation has to be solved in the one-equation region. The k and ε-equations have the same form as in the $k-\varepsilon$ model (see Eq. 1), except that the production term retain its original form, i.e.

$$P_k = -\overline{u_i u_j}\frac{\partial U_i}{\partial x_j}. \tag{9}$$

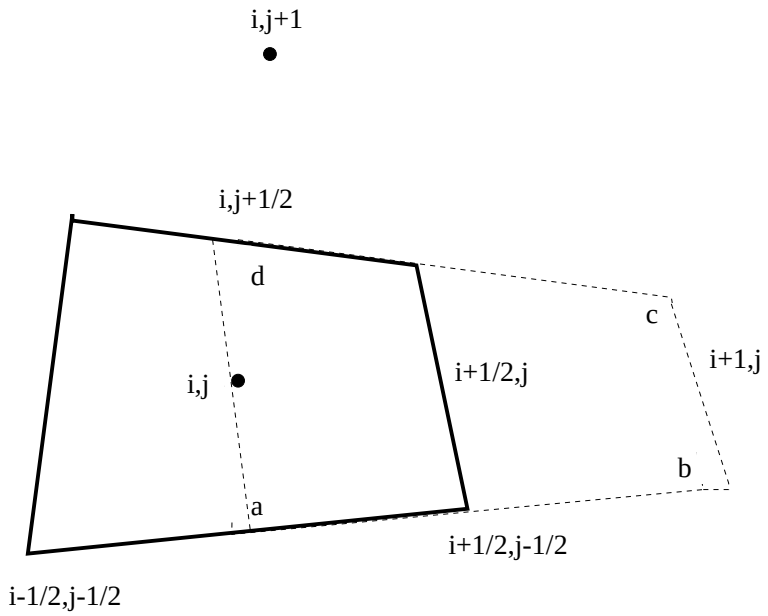

Figure 1: *A control volume at cell centre* (i, j).

3 The Discretization

Let Φ denote a general variable ($\Phi = k, \varepsilon$, $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$). The transport equations for the turbulent quantities can then all be written as

$$\frac{\partial}{\partial x_m}(\rho U_m \Phi) = \frac{\partial}{\partial x_m}(\Gamma_\Phi \frac{\partial \Phi}{\partial x_m}) + \overline{S}^\Phi \tag{10}$$

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

$$J_m = \rho u_m \Phi - \Gamma_\Phi \frac{\partial \Phi}{\partial x_m} \tag{11}$$

Eq. 10 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 (in, for simplicity, one dimension)

$$\{\mathbf{J} \cdot \mathbf{A}\}_{i+1/2} + \{\mathbf{J} \cdot \mathbf{A}\}_{i-1/2} = S^\Phi \tag{12}$$

where S^Φ is the total source in the control volume.

Note that for Cartesian coordinates (where $u \geq 0$) the positive sign in front of the contribution for the low-I $(i-1/2)$ turns into a negative one because the scalar product between the two vectors $(\mathbf{J}\cdot\mathbf{A})$ is negative.

3.1 Convection

The convection, which is the first part of the flux vector $\mathbf{J}$ in Eq. 11, is the scalar product of the velocity vector and the area vector multiplied by the density and the variable Φ. For the high-I face it gives (see Fig. 1):

$$\{\dot{m}\Phi\}_{i+1/2} \quad = \quad \{\rho\mathbf{u}\cdot\mathbf{A}\Phi\}_{i+1/2} \tag{13}$$

3.2 Diffusion

Diffusion is the second part of the flux vector $\mathbf{J}$ in Eq. 11, and it has the form:

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

For the high-I face (at $i+\frac{1}{2},j$ in Fig. 1), for example, it gives

$$\begin{aligned} -\{\Gamma_\Phi\mathbf{A}\cdot\nabla\Phi\}_{i+\frac{1}{2},j} \quad &= \quad -\left\{\Gamma_\Phi\left(A_x\frac{\partial\Phi}{\partial x}+A_y\frac{\partial\Phi}{\partial y}\right)\right\}_{i+\frac{1}{2},j} = \\ &- \quad \left\{\Gamma_\Phi\,|\mathbf{A}|\left(n_x\frac{\partial\Phi}{\partial x}+n_y\frac{\partial\Phi}{\partial y}\right)\right\}_{i+\frac{1}{2},j} \end{aligned} \tag{14}$$

where (n_x,n_y) is the outwards unit normal vector of face $i+\frac{1}{2},j$. Now we need to evaluate the derivatives $\partial\Phi/\partial x$ and $\partial\Phi/\partial y$ at $i+\frac{1}{2},j$. We do that by applying Green's formula to the volume $a-b-c-d$ (see Fig. 1) surrounding $i+\frac{1}{2},j$, i.e.

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

where S is the surface enclosing the volume V. We get

$$\begin{aligned} \left[\frac{\partial\Phi}{\partial x}n_x\right]_{i+\frac{1}{2},j} &= \left\{\frac{n_x}{V}\right\}_{i+\frac{1}{2},j}\left\{(\Phi n_x)_{i+1,j}+(\Phi n_x)_{i+\frac{1}{2},j+\frac{1}{2}}-(\Phi n_x)_{i,j}-(\Phi n_x)_{i+\frac{1}{2},j-\frac{1}{2}}\right\} \\ \left[\frac{\partial\Phi}{\partial y}n_y\right]_{i+\frac{1}{2},j} &= \left\{\frac{n_y}{V}\right\}_{i+\frac{1}{2},j}\left\{(\Phi n_y)_{i+1,j}+(\Phi n_y)_{i+\frac{1}{2},j+\frac{1}{2}}-(\Phi n_y)_{i,j}-(\Phi n_y)_{i+\frac{1}{2},j-\frac{1}{2}}\right\} \end{aligned} \tag{15}$$

Note that on the high faces ($i+1,j$ and $i+\frac{1}{2},j+\frac{1}{2}$), the normal vectors are *outwards* pointing vectors, but on the low faces (i,j and $i+\frac{1}{2},j-\frac{1}{2}$) the normal vectors point *into* the control volume. Since the volume $a-b-c-d$ is chosen so that $(n_x)_{i+1,j}=(n_x)_{i,j}=(n_x)_{i+\frac{1}{2},j}$ and

$(n_y)_{i+1,j} = (n_y)_{i,j} = (n_y)_{i+\frac{1}{2},j}$, the sum of the two terms in Eq. 15 can be written

$$\left[\frac{\partial \Phi}{\partial x} n_x\right]_{i+\frac{1}{2},j} + \left[\frac{\partial \Phi}{\partial y} n_y\right]_{i+\frac{1}{2},j} =$$

$$= \left\{\frac{1}{V}\right\}_{i+\frac{1}{2},j} \left\{ (\Phi_{i+1,j} - \Phi_{i,j}) \left[n_x^2 + n_y^2\right]_{i+\frac{1}{2},j} \right. \tag{16}$$

$$+ (n_x)_{i+\frac{1}{2},j} \left[(n_x)_{i+\frac{1}{2},j+\frac{1}{2}} \Phi_{i+\frac{1}{2},j+\frac{1}{2}} - (n_x)_{i+\frac{1}{2},j-\frac{1}{2}} \Phi_{i+\frac{1}{2},j-\frac{1}{2}}\right]$$

$$\left. + (n_y)_{i+\frac{1}{2},j} \left[(n_y)_{i+\frac{1}{2},j+\frac{1}{2}} \Phi_{i+\frac{1}{2},j+\frac{1}{2}} - (n_y)_{i+\frac{1}{2},j-\frac{1}{2}} \Phi_{i+\frac{1}{2},j-\frac{1}{2}}\right] \right\}$$

The second line represents orthogonal diffusion, and the two last lines represent non-orthogonal diffusion which vanishes on orthogonal grids. Eqs. (14-16) now can be written:

$$- \{\Gamma_\Phi \mathbf{A} \cdot \nabla \Phi\}_{i+\frac{1}{2},j} =$$

$$- \left\{\frac{\Gamma_\Phi |\mathbf{A}|}{V}\right\}_{i+\frac{1}{2},j} \left\{ (\Phi_{i+1,j} - \Phi_{i,j}) \left[n_x^2 + n_y^2\right]_{i+\frac{1}{2},j} \right. \tag{17}$$

$$+ (n_x)_{i+\frac{1}{2},j} \left[(n_x)_{i+\frac{1}{2},j+\frac{1}{2}} \Phi_{i+\frac{1}{2},j+\frac{1}{2}} - (n_x)_{i+\frac{1}{2},j-\frac{1}{2}} \Phi_{i+\frac{1}{2},j-\frac{1}{2}}\right]$$

$$\left. + (n_y)_{i+\frac{1}{2},j} \left[(n_y)_{i+\frac{1}{2},j+\frac{1}{2}} \Phi_{i+\frac{1}{2},j+\frac{1}{2}} - (n_y)_{i+\frac{1}{2},j-\frac{1}{2}} \Phi_{i+\frac{1}{2},j-\frac{1}{2}}\right] \right\}$$

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

$$\{\Gamma_\Phi \mathbf{A} \cdot \nabla \Phi\}_{i+\frac{1}{2},j,\mathrm{ort}} = D_{I,i+1/2,j} (\Phi_{i+1,j} - \Phi_{i,j}) \tag{18}$$

3.3 The discretized equation

Combining Eqs. 12, 13 and 18 (in 1D) gives

$$\left\{\dot{m}_{i+1/2}\Phi_{i+1/2} - D_{I,i+1/2}(\Phi_{i+1} - \Phi_i)\right\} - \left\{\dot{m}_{i-1/2}\Phi_{i-1/2} - D_{I,i-1/2}(\Phi_i - \Phi_{i-1})\right\} = S^\Phi \tag{19}$$

where the non-orthogonal diffusion terms have been included in the source term. Note that $\dot{m}_{i+1/2}$ denotes the mass flux *out* of the control volume, whereas $\dot{m}_{i-1/2}$ is the mass flux *into* the control volume. The problem now arises for the convection terms: how to estimate $\Phi_{i\pm1/2}$? It is well known that second order central differencing causes unphysical occilations in Euler calculations where no viscous damping effects are present. However, when the diffusive (viscous) transport is of the same order as the convective transport the use of central differencing poses no problems. The upwind scheme ($\Phi_{i+1/2} = \Phi_i$ if $\dot{m}_{i+1/2} \geq 0$ and $\Phi_{i+1/2} = \Phi_{i+1}$ otherwise) is always stable but only first order accurate. The compromise chosen in this work is one that is commonly used in codes based on SIMPLE-methods, namely hybrid central/upwind differencing.[9] In this scheme central differencing is used when diffusion is large enough (Peclet number smaller than two), and upwind differencing otherwise (the Peclet number P_e is the

local Reynolds number based on the cell size, $P_e = U\delta x/(\nu_t + \nu)$). The switch from central differencing to upwind differencing is chosen at $P_e = 2$ in order to ensure that the coefficients in the linearized equation (Eq. 22) be positive. For the $i + \frac{1}{2}$ face, for example, one obtains

$$\{\text{convection} - \text{diffusion}\}_{i+1/2} = \{\dot{m}\Phi\}_{i+1/2} - D_{I,i+1/2}(\Phi_{i+1} - \Phi_i) =$$
$$\max\left\{\frac{1}{2}\dot{m}_{i+1/2} + D_{I,i+1/2}, 0, \dot{m}_{i+1/2}\right\}\Phi_i - \max\left\{-\frac{1}{2}\dot{m}_{i+1/2} + D_{I,i+1/2}, 0, -\dot{m}_{i+1/2}\right\}\Phi_{i+1} \quad (20)$$

Equation 20 into 19 gives

$$\max\left\{\frac{1}{2}\dot{m}_{i+1/2} + D_{I,i+1/2}, 0, \dot{m}_{i+1/2}\right\}\Phi_i - \max\left\{-\frac{1}{2}\dot{m}_{i+1/2} + D_{I,i+1/2}, 0, -\dot{m}_{i+1/2}\right\}\Phi_{i+1}$$
$$- \max\left\{\frac{1}{2}\dot{m}_{i-1/2} + D_{I,i-1/2}, 0, \dot{m}_{i-1/2}\right\}\Phi_{i-1} + \max\left\{-\frac{1}{2}\dot{m}_{i-1/2} + D_{I,i-1/2}, 0, -\dot{m}_{i-1/2}\right\}\Phi_i = S^\Phi \quad (21)$$

This can be rewritten, using the continuity equation ($\dot{m}_{i+1/2} - \dot{m}_{i-1/2} = 0$) and linearizing the source terms as $S^\Phi = S_P^\Phi\Phi + S_U$, which gives

$$a_{P,i}\Phi_i = a_{E,i}\Phi_{i+1} + a_{W,i}\Phi_{i-1} + S_{U,i}^\Phi \quad (22)$$

where

$$\begin{aligned} a_E &= \max\left\{-\frac{1}{2}\dot{m}_{i+1/2} + D_{I,i+1/2}, 0, -\dot{m}_{i+1/2}\right\} \\ a_W &= \max\left\{\frac{1}{2}\dot{m}_{i-1/2} + D_{I,i-1/2}, 0, \dot{m}_{i-1/2}\right\} \\ a_P &= a_W + a_E - S_P \end{aligned}$$

Index P, E, W denote *P*oint, *E*ast and *W*est, respectively.

The extension to two dimensions is straight-forward and the resulting form of the discretized equation reads

$$\begin{aligned} a_{P,i,j}\Phi_{i,j} &= a_{E,i,j}\Phi_{i+1,j} + a_{W,i,j}\Phi_{i-1,j} + a_{N,i,j}\Phi_{i,j+1} + a_{S,i,j}\Phi_{i,j-1} + S_{U,i,j}^\Phi \\ a_{E,i,j} &= \max\left\{-\frac{1}{2}\dot{m}_{i+1/2,j} + D_{I,i+1/2,j}, 0, -\dot{m}_{i+1/2,j}\right\} \\ a_{W,i,j} &= \max\left\{\frac{1}{2}\dot{m}_{i-1/2,j} + D_{I,i-1/2,j}, 0, \dot{m}_{i-1/2,j}\right\} \\ a_{N,i,j} &= \max\left\{-\frac{1}{2}\dot{m}_{i,j+1/2} + D_{J,i,j+1/2}, 0, -\dot{m}_{i,j+1/2}\right\} \\ a_{S,i,j} &= \max\left\{\frac{1}{2}\dot{m}_{i,j-1/2} + D_{J,i,j-1/2}, 0, \dot{m}_{i,j-1/2}\right\} \\ a_{P,i,j} &= a_{W,i,j} + a_{E,i,j} + a_{S,i,j} + a_{N,i,j} - S_{P,i,j} \end{aligned} \quad (23)$$

where indices i, j have been added as to remind that the a-coefficients and the source terms are matrices.

3.4 Source terms

The source terms are taken to be constant in the control volume, i.e.

$$\int_V S^\phi dV = \overline{S}^\Phi \delta V = S^\Phi$$

In general, the source term is linearized as[9]

$$S = S_P \Phi + S_U$$

and S_P is included in the diagonal of the matrix (see Eq. 23), since it, when negative, enhances the diagonal dominance of the coefficient matrix. In general, the purpose of including source terms is to enhance the diagonal dominance. But, when solving for turbulence quantities (such as k, $\overline{u^2}$, $\overline{v^2}$ and ε), which during the whole course of the iteration process must stay positive (divergence will otherwise rapidly occur), an efficient way of ensuring that the variable stay postive, is to put *all* negative source terms in S_P. So, if we have a source term R which can be both positive and negative, we put it in S_U when positive and S_P when negative, i.e.

$$S_U = \max(R, 0)$$

$$S_P = \min(R/\Phi, 0)$$

One example of a source term which can be both positive and negative, is the dilatation term in the $k-\varepsilon$ model (term in square brackets in Eq.3). The first term in Eq.3 is always postive, and is therefore put in S_U. The dissipation $-\rho\varepsilon\delta V$ term in the k-equation (see Eq.1) is always negative, and is therefore put in S_P, i.e.

$$S_P = -\rho\varepsilon/k\delta V$$

A convenient way to fix a variable at a chosen value (usually zero) is to use S_P and S_U. If we want to fix Φ to Φ_{fix}, we set $S_P = -\texttt{great}$ (`great` is some some large value, e.g. 10^{10}) and $S_U = \texttt{great} \cdot \Phi_{fix}$, which gives us:

$$\Phi = \left(a_E\Phi_{i+1,j} + a_W\Phi_{i-1,j} + a_N\Phi_{i,j+1} + a_S\Phi_{i,j-1} + \texttt{great} \cdot \Phi_{fix}\right)/a_P \tag{24}$$

As $\texttt{great} \gg a_W, a_E, a_N, a_S$ we get

$$\Phi \simeq \texttt{great} \cdot \Phi_{fix}/\texttt{great} = \Phi_{fix} \tag{25}$$

4 Solver

A tri-diagonal solver is used, and since it solves linear equations along *lines* (say I-line), Eq. 23 is rewritten so that the contributions to point (i,j) from $(i,j-1)$ and $(i,j+1)$ are transferred to the right-hand-side, which in matrix form gives

$$\begin{pmatrix} a_{P,1} & -a_{E,1} & & & & & \\ -a_{W,2} & a_{P,2} & -a_{E,2} & & & & \\ & & \cdots\cdots\cdots & & & & \\ & & & -a_{W,i} & a_{P,i} & -a_{E,i} & \\ & & & \cdots\cdots\cdots & & & \\ & & & & -a_{W,ni} & a_{P,ni} & \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \cdots \\ \Phi_i \\ \cdots \\ \Phi_{ni} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_i \\ \cdots \\ b_{ni} \end{pmatrix}$$

where

$$b_i = S_{U,i,j} + a_{N,i,j}\Phi_{i,j+1} + a_{S,i,j}\Phi_{i,j-1}$$

Eliminating the $a_{W,2}$-element by adding to row 2 row 1 multiplied with $a_{W,2}/a_{P,1}$ gives

$$\begin{pmatrix} a_{P,1} & -a_{E,1} & & & & & \\ & A_{P,2} & -a_{E,2} & & & & \\ & & \cdots\cdots\cdots & & & & \\ & & & -a_{W,i} & a_{P,i} & -a_{E,i} & \\ & & & \cdots\cdots\cdots & & & \\ & & & & -a_{W,ni} & a_{P,ni} & \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \cdots \\ \Phi_i \\ \cdots \\ \Phi_{ni} \end{pmatrix} = \begin{pmatrix} b_1 \\ B_2 \\ \cdots \\ b_i \\ \cdots \\ b_{ni} \end{pmatrix}$$

where

$$B_2 = b_2 + \frac{a_{W,2}b_1}{a_{P,1}}$$
$$A_{P,2} = a_{P,2} - \frac{a_{W,2}a_{E,1}}{a_{P,1}}$$

Doing this for each row, and thereby eliminating the lower diagonal, gives

$$\begin{pmatrix} a_{P,1} & -a_{E,1} & & & & \\ & A_{P,2} & -a_{E,2} & & & \\ & & \cdots\cdots\cdots & & & \\ & & & A_{P,i} & -a_{E,i} & \\ & & & \cdots\cdots\cdots & & \\ & & & & & A_{P,ni} \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \cdots \\ \Phi_i \\ \cdots \\ \Phi_{ni} \end{pmatrix} = \begin{pmatrix} b_1 \\ B_2 \\ \cdots \\ B_i \\ \cdots \\ B_{ni} \end{pmatrix} \tag{26}$$

where

$$\begin{aligned} B_i &= b_i + \frac{a_{W,i}B_{i-1}}{A_{P,i-1}} \\ A_{P,i} &= a_{P,i} - \frac{a_{W,i}a_{E,i-1}}{A_{P,i-1}} \end{aligned} \tag{27}$$

Equation 26 can be written as

$$A_{P,i}\Phi_i - a_{E,i}\Phi_{i+1} = B_i \tag{28}$$

where $A_{P,i}$ and B_i are defined in Eq. 27. The linear equation system in Eq. 23 can now be solved along each i-line by first calculating all $A_{P,i}$ and B_i defined in Eq. 27, and then calculating Φ_{i+1} from Eq. 28.

Above in this section expressions have been derived to solve the linear equation system in Eq. 23 along I-lines. For solving Eq. 23 along J-lines, Eq. 23 is rewritten so that the contributions to point (i,j) from $(i-1,j)$ and $(i+1,j)$ are transferred to the rigth-hand-side, and the subsequent expressions are derived in exactly the same way as above. In this way Eq. 23 can be solved, alternating, along I-lines and J-lines.

At each time step the Φ-equation is solved twice (once along I-lines and once along J-lines).

5 Description of Subroutines

5.1 Subroutine `deriv`

For laminar flow and when using Baldwin-Lomax model, the viscosity is in `deriv` computed at the cell vertices. When using $k-\varepsilon$ (`ieuns=4`) and RSM (`ieuns=5`), all turbulent quantities, are stored at cell centres, and thus the viscosity is computed at the centre of the control volumes.

5.2 Subroutine `tratur`

The subroutine `tratur` is called from `solver` when a transport turbulence model is to be used. The different transport turbulence models are:

`ieuns=4` $k-\varepsilon$ turbulence model

`ieuns=5` Reynolds Stress Model (RSM)

In `tratur` we have a loop over `maxit` iterations and a loop over all blocks. Inside these loops the subroutines `calct` and `deriv` are called. In the former the temperature is computed, and in the latter the velocity and temperature derivatives are computed; in `deriv` the laminar viscosity and heat conductivity are also computed.

At the end of the loop the subroutine `keps` is called, and if `ieuns=5` the subroutine `rsm` is also called.

5.2.1 Using $k-\varepsilon$ model

Normally it is possible to start $k-\varepsilon$ computations from scratch, i.e. using inlet boundary conditions as homogeneous initial conditions (all turbulent quantities=0). If we are doing a restart, where the restart file has been obtained with Baldwin-Lomax, it may be worth while to initialize k and ε. This is can be done by setting `maxit` in the beginning of subroutine `tratur` to, say, 100 (by default `maxit=1`). 100 iterations (`= maxit`) solving k and ε are then carried out.

The call to `keps` looks like:

```
      call keps(work(iro),work(iuu),work(ivv),work(iux),work(iuy),
     .  work(ivx),work(ivy),work(ikk),work(ixx),
     .  work(iyy),work(inix),work(iniy),work(injx),
     .  work(injy),work(ivol),work(imut),
     .  work(idnorm),work(iae),work(ian),work(iaw),
     .  work(ias),work(iapo),work(iap),work(isp),
     .  work(isu),work(iconve),work(iconvn),work(imulam),
     .  work(igen),work(idilat),work(irleps),
     .  ni,nj,ist1,ien1,jji1,npbl1,ist2,ien2,jmax1,jmax2,
     .  jji2,npbl2,iis1,iie1,iis2,iie2,nseg1,nseg2,
     .  maxit,iterk,iter,ibloc)
```

All indices `index` in `work(index)` are pointers in the `work`-array. The indices have the following meaning:

- `iro,iuu,ivv` are the depending variables density, and the two velocity components
- `iux,iuy,ivx,ivy` are the velocity gradients, i.e. $\partial U/\partial x$, etc, stored at the cell vertices

- `work(ikk)` is the first element of the k-field. `keps` has then access also to the other turbulent quantities (ε, $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$) since they are stored in the `work`-array after k
- `ixx,iyy` coordinates of the cell vertices
- `isix,isiy` normal vector components of the $i-1/2, j$ faces
- `isjx,isjy` normal vector components of the $i, j-1/2$ faces
- `ivol` cell volume
- `imut` dynamic turbulent viscosity.

which all are global variables in `tharros`. The arrays `work(idnorm), work(iae), work(ian), work(iaw), work(ias), work(iapo), work(iap), work(isp), work(isu), work(iconve), work(iconvn), work(imulam), work(igen), work(idilat), work(irleps)` are work-arrays, and are used locally in `keps`, except two of them (`work(idnorm)` and `work(imulam)`) which serve as input to `rsm`.

Boundary conditions at walls when using the $k-\varepsilon$ model or RSM are given in the same way as for Baldwin-Lomax. Note that in `tharros` only two sides, $j=1$ (wall 1) and $j=nj-1$ (wall 2) can be defined as walls. The following parameters are used, which all are global variables in `tharros`:

- `nj,nj` number of grid lines in i and j-direction
- `ist1(n) [ist2(n)],ien1(n)[ien2(n)]` starting and ending i-grid line of segment n at wall 1 [2] which defines the turbulent part of segment n ($n \leq 5$)
- `npbl1(n) [npbl2(n)]` number of grid lines at segment n at which the one-equation model is to be used at wall 1 [2] ($n \leq 5$). If npbl1(n) [npbl2(n)] is negative, the location of the matching line is not chosen along a fix grid line, but it is taken where the damping function in square brackets in Eq. 4 is equal to 0.9. The local j-index of the grid line is stored in `jmax1(i)` [`jmax2(i)`]. Note that the damping function often goes to zero in the laminar far field region, where we don't want to use the one-equation model. In order to prevent this, `jmax1(i)` [`jmax2(i)`] is not allowed to exceed `abs(npbl1(i))` [`abs(npbl2(i))`]. Note that `jmax1` and `jmax2` are one-dimensional arrays, which are dimensioned to 300. This means that a wall cannot have more than 300 grid nodes. If it has more, `jmax1` & `jmax2` have to be re-dimensioned.
- `jji1(n) [jji2(n)]` j-grid line which defines wall 1 [2] at segment n ($n \leq 5$)
- `iis1(n) [iis2(n)], iie1(n), [iie2(n)]` starting and ending i-grid line of segment n at wall 1 [2] of segment n ($n \leq 5$)
- `nseg1 [nseg2]` number of wall segments at wall 1 [2] ≤ 5

5.2.2 Using the Reynolds Stress Model (RSM)

Normally it is possible to start RSM computations from scratch, i.e. using inlet boundary conditions as homogeneous initial conditions (all turbulent quantities=0). If we are doing a restart, where the restart file has been obtained with $k-\varepsilon$, it may be worth while to initialize the Reynolds stresses. This is can be done by setting `maxit` in the beginning of subroutine `tratur` to, say, 100 (by default `maxit=1`). 100 iterations (`= maxit`) solving k,ε, $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$ are then carried out.

The call to `rsm` looks like:

```
call rsm(work(iro),work(iux),work(iuy),work(ivx),
.  work(ivy),work(ikk),work(ixx),work(iyy),
.  work(inix),work(iniy),work(injx),work(injy),
.  work(ivol),work(imut),work(idnorm),
.  work(iae),work(ian),
.  work(iaw),work(ias),work(iapo),work(iap),
.  work(isp),work(isu),work(imulam),work(iconve),
.  work(iconvn),work(igen),ni,nj,ist1,ien1,jji1,
.  npbl1,ist2,ien2,jji2,npbl2,iis1,iie1,iis2,iie2,
.  jmax1,jmax2,nseg1,nseg2,iterk,maxit,iter)
```

where all variables were explained in the previous subsection.

5.3 Subroutine `keps`

In the following the subroutine `keps` is described in some detail.

Part 1 The starting end ending indices of i are changed in order to secure that the turbulent viscosity in the one-eqution region is computed also at dummy cells ($i=0$ and $i=ni$). This is necessary at boundaries which are block interfaces.
Caution must be taken at the first iteration when k and ε are zero. Therefore if `iter`=1 and `iterk=1` we ensure that k and ε have some finite values (=`small`=10^{-10}).

Part 2 The normal distance from the walls is computed. It is simply computed as $\sqrt{(x_p-x_{wall})^2+(y_p-y_{wall})^2}$ from both walls, and the normal distance is taken as the minimum of the two and is stored in `dnorm`. The index `ii` is used in order to compute `dnorm` correctly also at dummy nodes $i=0$ and $i=ni$.

Part 3 The mass fluxes at the high faces ($i+\frac{1}{2},j$ and $i,j+\frac{1}{2}$) are computed and are stored in `conve` and `convn`.

Part 4 Here the loop which solves k and ε starts. If no initialization is to be done, then `maxit=1`. The turbulent viscosity is first computed in the whole field, and then it is re-computed near the walls using the one-equation model. This is done for the `npbl1(n)-1` first nodes near wall 1 and `nj-npbl2(n)-2` last nodes at wall 2. Note that `npbl1(n)` and `npbl2(n)` refer to *grid lines*. The turbulent viscosity is stored in `mut`.

Part 5 The dilatation term (term in square brackets in Eq. 3) and the production terms (see Eqs. 3,9) are computed.

Part 6 If $k-\varepsilon$ model is being used, k is solved first, and then ε, which has proved to be stable during the first time-steps starting from $k = \varepsilon = 10^{-10}$. However, if RSM is used, ε is solved first and then k, because the coefficients (a_E, a_W, a_N and a_S) are the same for the k-equation and the $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$-equations, which means that the coefficients do not need to be re-computed in subroutine `rsm`.
The coefficients and non-orthogonal diffusion are calculated in `nonort` as described in Section 3.

Part 7 The source term (production, dissipation and dilatation) are computed and added into `su` or `sp` depending on their sign (see Section 3.4). Note that when using $k-\varepsilon$ the production term in the ε-equation is computed differently than when RSM is used. During the first time steps it is necessary to compute this term in this way, otherwise the turbulent viscosity grows to rapidly due to too slow growth of ε. In RSM the production term has the form $c_{1\varepsilon}\varepsilon/kP_k$. If we, when using $k-\varepsilon$, write $P_k = \mu_t G$ where G contains the velocity derivatives (see Eq. 3), we get $c_\mu \rho c_{1\varepsilon} kG$. (Note that this is not true in the one-equation region, but what we do with the ε-equation in this region is irrelevant, since ε is re-calculated after that the transport equation has been solved). The former expression contains the dissipation ε at taken at the previous time-level n, which during the first time steps will be too small, and thus the production term too small, which gives too small ε and too large μ_t.

Part 8 k, ε and turbulent viscosity are set to zero (by setting `sp=-great`, see Eqs. 24,25) in the laminar region.

Part 9 The equations are under-relaxated (using standard under-relaxation factor of 0.5), and k and ε are solved in subroutine `tdma` using tdma (see Section 4). Immediately after ε has been solved (and before k is to be solved), ε is re-calculated in the one-equation region according to Eq. 6.

Part 10 At the last iteration (`iter=ncyc, iterk=maxit`), the location of the matching line (`jmax1` & `jmax2`) are printed out.

5.4 Subroutine `rsm`

In the following the subroutine `rsm` is described in some detail. Many details have been explained in the previous subsection, and or not repeated here.

Part 1 Caution must be taken at the first iteration when $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$ are zero. Therefore if `iter`=1 and `iterk=1` we ensure that they have some finite values (=`small`=10^{-10}).

Part 2 The normal distance from the walls is computed. Then the damping functions f_x, f_y and f_{xy} are computed.

Part 3 Here the loop which solves $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$ starts. The non-orthogonal diffusion is computed in `nonort`.
The production terms, pressure-strain term, and the dissipation terms are includes in `sp` or `su` depending on sign (see Section 3.4. The treatment of the source terms is described in detail in Ref. 10.

Part 4 $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$ are set to zero (by setting `sp=-great`, see Eqs. 24,25) in the laminar region.

Part 5 The equations are under-relaxated (using standard under-relaxation factor of 0.5), and k and ε are solved in subroutine `tdma` using tdma (see Section 4).

Part 6 In the one-equation region the Reynolds stresses are computed according to Eq. 9.

6 Test case: Bump A

6.1 The Input File `*.bc`

All indata is given in the same way as when Baldwin-Lomax turbulence model is used. In the present test case, we have a wall at $j = 1$ along which the flow is turbulent for all i-grid lines. In the boundary condition file `*.bc` we prescribe that side 1 is a wall as follows: as

```
1
13 1 97 1 1
1.0 97.0 -11.
```

The first line sets number of segments on the $j = 1$ boundary.
At the second line, the items have the following meaning:

Item 1 =13 means that it is a wall

Item 2 defines the first i-grid line

Item 3 defines the last i-grid line

Item 4 defines the first j-grid line

Item 5 defines the first j-grid line

At the third line, the items have the following meaning:

Item 1 defines the first i-grid line where the flow is turbulent

Item 2 defines the last i-grid line where the flow is turbulent

Item 3 the number of grid lines that the cover the one-equation region. In this example, the one-equation model is used in the ten cells closest to wall 1 (covered by eleven grid lines). If the boundary would have been a high-j boundary (wall 2), the number still indicates how many grid lines that cover the one-equation region.
If the one-equation model not is to be used at all, insert a zero (=0.0).
If, as in the present test case, the matching line should be computed automatically (where the damping term in square brackets in Eq. 4 is 0.9), set a negative value (the number of grid lines where the one-equation model is used is smaller than `abs(npbl1)` [`abs(npbl2)`]), see Section 5.2.1.

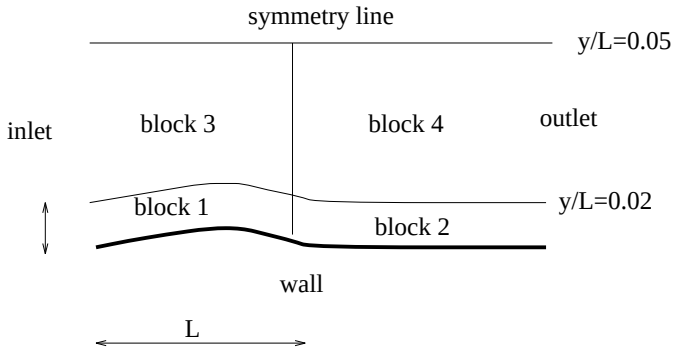

Figure 2: *The configuration. A four-block mesh is used.*

	Baldwin-Lomax	$k-\varepsilon$	RSM
CPU	1550	1650	1850

Table 1: *CPU time (in seconds) for 1000 time steps using the different turbulence models.*

6.2 Results

The flow in a channel with a bump (Case A, in Ref.11) is used as test case. This flow case was also computed in the EUROVAL project[12] and in Ref.3. A four-block mesh with 49×17 grid lines in each is used (see Fig. 2), and the flow is predicted using both $k-\varepsilon$ and RSM.

The convergence plot for Baldwin-Lomax, $k-\varepsilon$ and RSM, using a second-order TVD scheme for the mean flow variables, are shown in Fig. 2 (the Baldwin-Lomax calculations were carried out on a single-block mesh, since it cannot be used on the multi-block mesh; the single-block mesh corresponds to the four-block mesh). All calculations are started from scratch, i.e. the initial conditions for the mean flow quantities are inflow conditions everywhere, and k, $\overline{u^2}$, $\overline{v^2}$ and $\overline{uv}$ are set to zero. As can be seen from Fig. 3, the convergence is actually better with both the $k-\varepsilon$ and the RSM than with the Baldwin-Lomax model.

The $k-\varepsilon$ model and RSM do not require much more CPU time than the Baldwin-Lomax model, partly because the two former models are called only once at each time step, whereas Baldwin-Lomax is used every (five) Runge-Kutta stage, and partly because the mean flow equation solver requires considerably computational effort. From Table 1 it can be seen that the $k-\varepsilon$ model and the RSM requires only 7 and 20 %, respectively, more CPU than the Baldwin-Lomax model.

In the figures below all quantities have been made non-dimensionalized with the stagnation speed of sound a_0 and the bump length L.

The wall pressure at the lower wall is presented in Fig. 4, and profiles of U, $\overline{uv}$, $\overline{u^2}$ and $\overline{v^2}$ are presented in Figs. 5-8, and the results are very similar to those obtained in Refs. 3,12. The

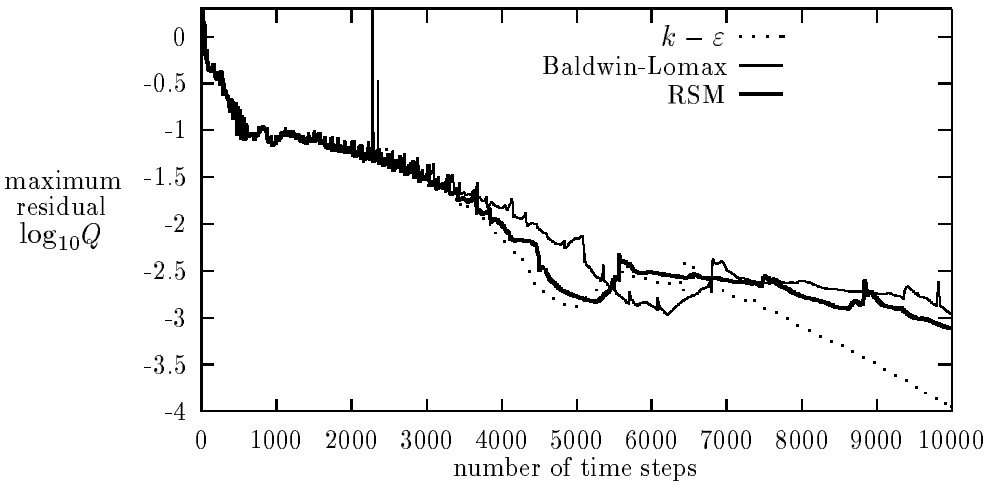

Figure 3: Convergence history. On the ordinate the maximum (of ρ, ρu, ρv and ρE) rms-residual is shown. All calculation were started from scratch. A single-block mesh was used for the Baldwin-Lomax calculations.

RSM predicts a stronger interaction between the shock and the boundary layer due to its ability of taking streamline curvature into account, and the RSM predicts a larger separation. This is also seen from the skin frictions in Fig. 9.

References

[1] MULAS, M., private communication, CRS4, Cagliari, 1993.

[2] DAVIDSON, L. and RIZZI, A. – Navier-Stokes Stall Predictions Using an Algebraic Stress Model", *J. Spacecraft and Rockets*, Vol. 29(6), pp. 794-800, 1992 (see also AIAA-paper 92-0195, Reno, Jan. 1992).

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

[4] DALY, B.J. and HARLOW, F.H. – Transport Equations of Turbulence, *Phys. Fluids*, **13**, pp. 2634-2649, 1970.

[5] 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.

[6] 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.

[7] GIBSON, M.M. and YOUNIS, B.A. – Calculation of Swirling Jets With a Reynolds Stress Closure, *Phys. Fluids*, **29**, pp. 38-48, 1986.

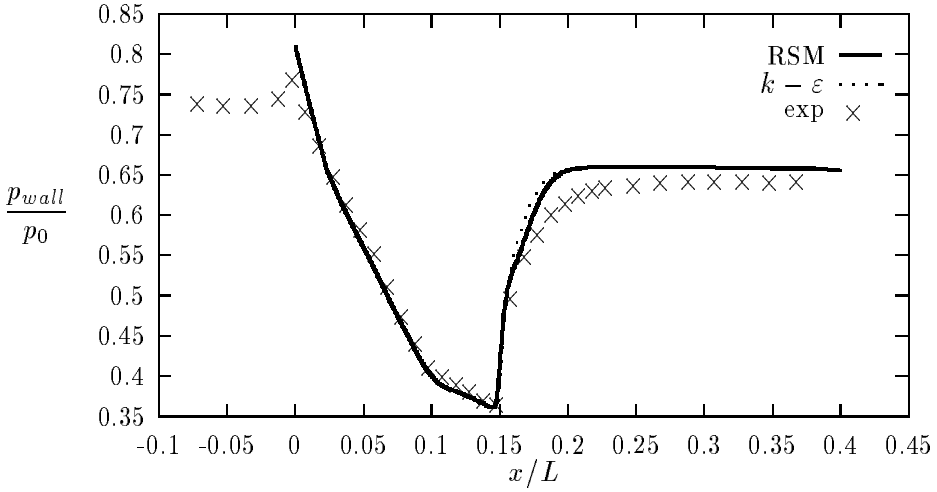

Figure 4: Wall pressure.

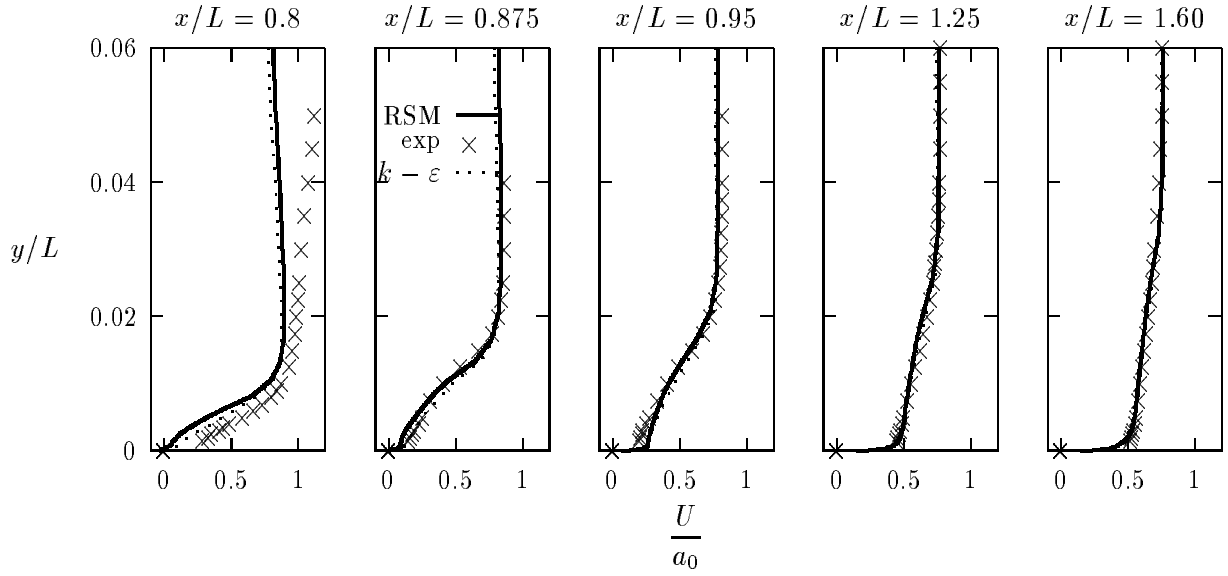

Figure 5: Predicted U-velocities compared with experiments

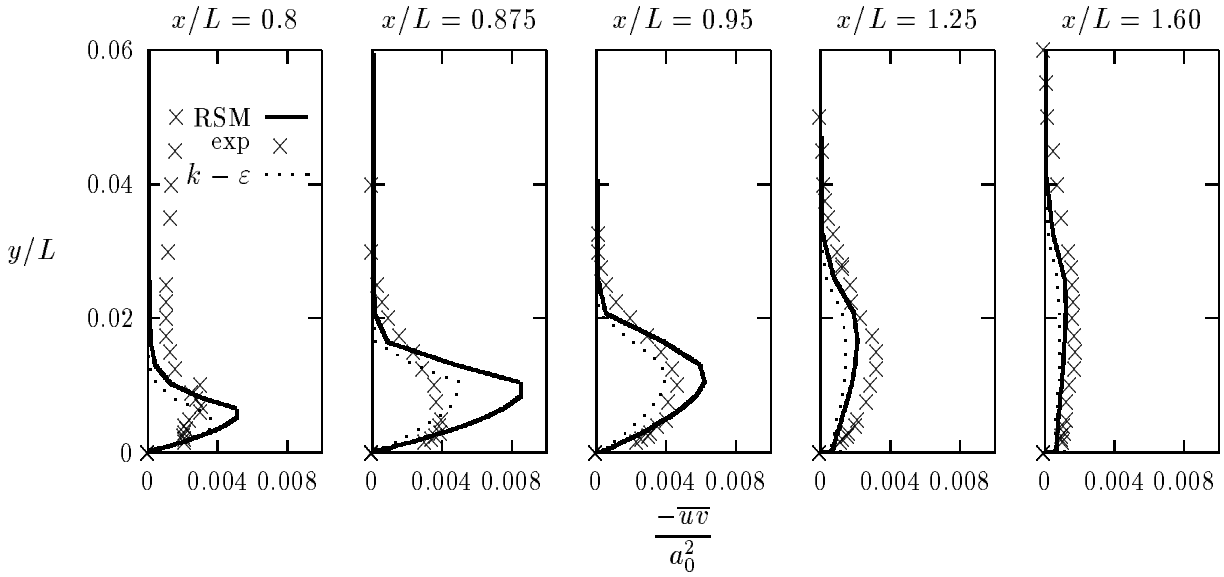

Figure 6: Predicted shear stresses compared with experiments

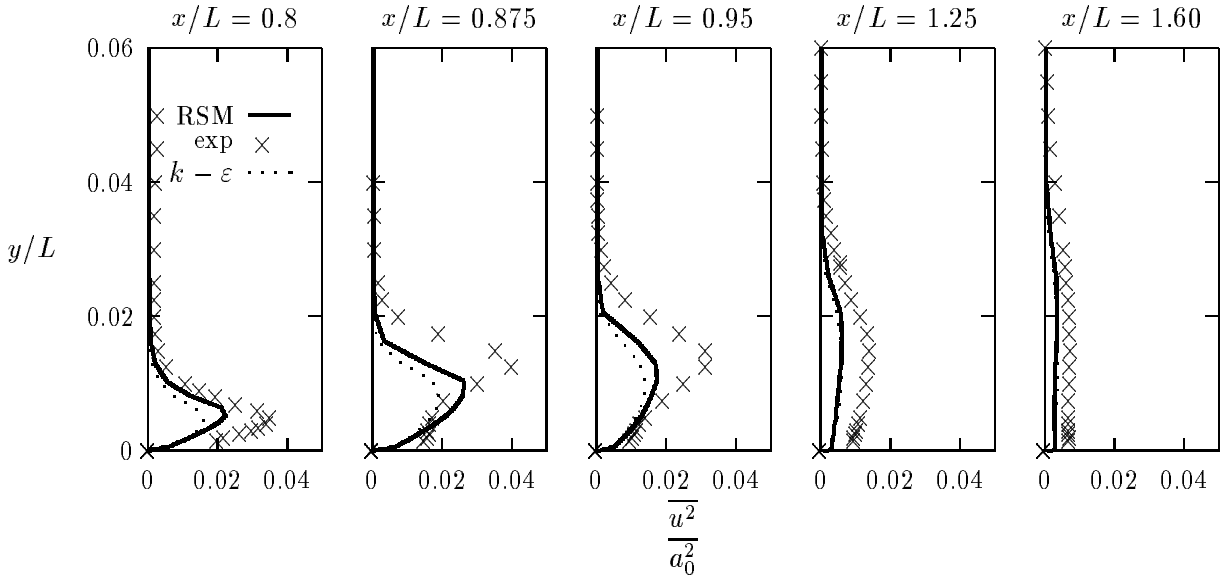

Figure 7: Predicted normal stresses $\overline{u^2}$ compared with experiments

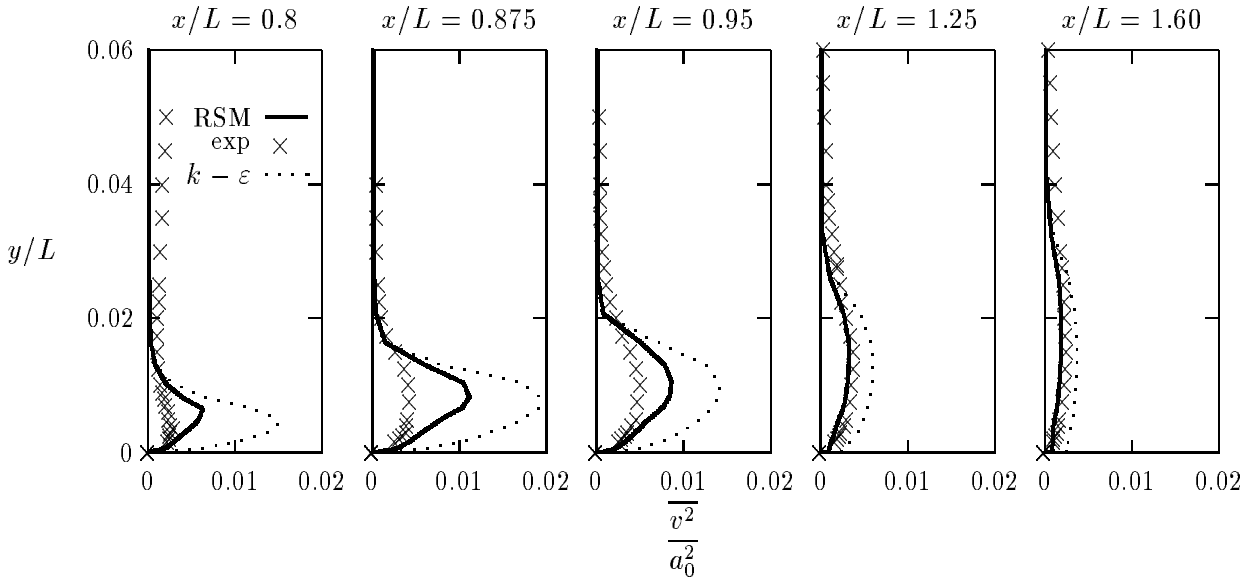

Figure 8: Predicted normal stresses $\overline{v^2}$ compared with experiments

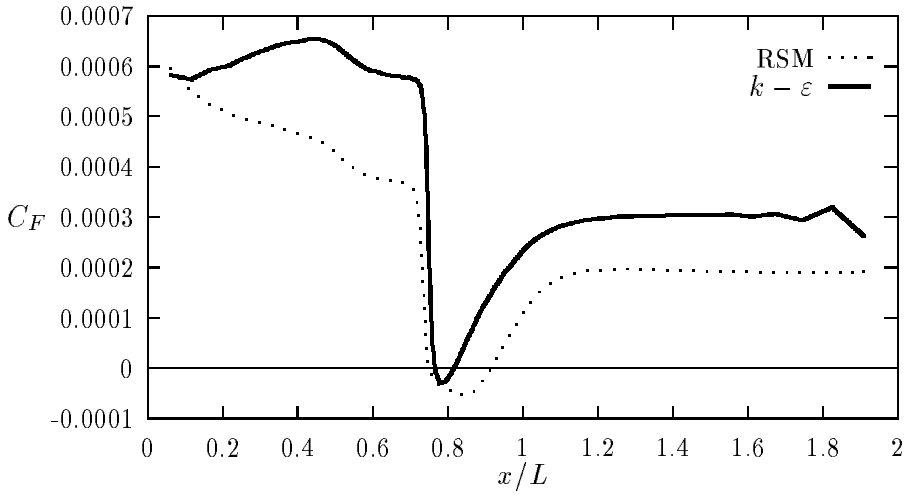

Figure 9: Predicted skin frictions.

[8] 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.

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

[10] FARHANIEH, B., DAVIDSON, L. and SUNDÉN, B. – Second Moment Closure for Calculation of Turbulent Recirculating Flows in Complex Geometries with Collocated Variable Arrangement, *Int. J. Numer. Meth. Fluids*, Vol. 16, pp. 525-544, 1993.

[11] DELERY, J. – Experimental Investigation of Turbulence Properties in Transonic Shock/Boundary-Layer Interactions, *AIAA J.*, **21**, pp. 180-185, 1983.

[12] EUROVAL – A European Initiative on Validation of CFD-codes, W. Haase, F. Brandsma, E. Elsholz, M. Leschziner and D. Schwamborn (Eds.), Notes on Numerical Fluid Mechanics, Vieweg Verlag, Vol. 42, 1993.