INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 265–281 (1996)

A PRESSURE CORRECTION METHOD FOR UNSTRUCTURED MESHES WITH ARBITRARY CONTROL VOLUMES

LARS DAVIDSON

Thermo and Fluid Dynamics, Chalmers University of Technology, S-412 96 Gothenburg, Sweden

SUMMARY

A pressure correction procedure for general unstructured meshes is presented. It is a cell-centred, collocated finite volume method and the pressure–velocity coupling is treated using SIMPLEC. The cells can have an arbitrary number of grid points (cell vertices). In the present study the number of faces on the cells varies between three and six. The discretized equations are solved using either a symmetric Gauss–Seidel solver or a conjugate gradient solver with a preconditioner. The method is applied to three two-dimensional test cases in which the flow is incompressible and laminar. The extension to three dimensions as well as to turbulent flow using transport models is straightforward. It can also be extended to handle compressible flow.

KEY WORDS: unstructured; SIMPLE; finite volume; cell-centred

1. INTRODUCTION

Computational fluid dynamics (CFD) is now used frequently in industry. When using structured methods, however, the grid generation for complex geometries remains a major task. The generation of grids for complex geometries usually requires considerably more time in terms of manpower than the actual flow field computations. In order to become a useful tool, CFD must be capable of handling complex flow in complex geometries. The lack of generality in treating complex geometries is one of the major reasons why CFD has not become a powerful tool in everyday engineering.

The use of unstructured methods facilitates the grid generation enormously and there exist automatic methods for triangulation of arbitrary geometries.[1] For Navier–Stokes computations a structured mesh near the boundaries can be matched with an internal or external (automatically generated) triangulated region.[2]

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

There exist a number of papers using unstructured finite volume flow solvers for compressible aerodynamics[2–10] in which algebraic and transport turbulence models have also been used. However, unstructured methods employing pressure correction techniques are not very common. There are a limited number of papers in the literature. Lonsdale and Webster[11] used a staggered grid arrangement and presented three-dimensional, turbulent flow calculations. A number of papers were presented in the 1980s on CVFEMs (control-volume-based finite element methods)[12,13] (see also Reference 14). Thomadakis and Leschziner[15] recently presented a semi-staggered approach and Watterson[16] developed a cell vertex procedure for compressible flow, solving the discretized equations in an

CCC 0271–2091/96/040265–17

Received November 1994
Revised May 1995

explicit time-marching manner with a Runge–Kutta method. The work most closely related to the present study is that of Jiang and Przekwas.[17]

The present pressure correction procedure is a cell-centred, collocated finite volume method for general unstructured meshes. The cells can have an arbitrary number of grid points and cell faces. In the test cases used in the present work, the number of cell faces in the cells ranges between three and six. Central differencing is used for the convective terms together with fourth-order numerical dissipation, while central differencing is employed for the diffusive terms. In order to avoid decoupling between pressure and velocity, the Rhie and Chow interpolation is used when computing the mass flux at cell faces. The discretized equations are solved by either a symmetric point-by-point Gauss–Seidel relaxation method or a conjugate gradient solver with incomplete Cholesky factorization preconditioning.

The finite volume method is presented in some detail in the next section. This is followed by a section that reports the results. The final section gives conclusions and some directions for further work.

2. THE FINITE VOLUME PROCEDURE

2.1. *The discretization*

Let Φ denote a general variable ($\Phi = U$, V or p'). The transport equations for Φ can then be written as

$$\frac{\partial}{\partial x_m}(\rho U_m \Phi) = \frac{\partial}{\partial x_m}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x_m}\right) + \bar{S}^\Phi, \tag{1}$$

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

$$J_m = \rho U_m \Phi - \Gamma_\Phi \frac{\partial \Phi}{\partial x_m}, \tag{2}$$

equation (1) can be written as

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

In vector notation the equation reads

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

Integrating this equation over a volume (with volume $\mathscr{V}$ and bounding surface A) using the Gauss law gives

$$\int_A \mathbf{J} \cdot \mathbf{dA} = \int_{\mathscr{V}} \bar{S}^\Phi \, \mathrm{d}\mathscr{V},$$

which for a control volume gives

$$\sum_{i=1}^{N} \{\mathbf{J} \cdot \mathbf{A}\}_i = \mathbf{S}^\Phi, \tag{3}$$

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

2.2. *Pointer system*

As 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 that gives information about which grid points (cell vertices) form the starting and ending points of the face and which two cells are the adjacent cells. This information is stored in listf, which has the form[8]

listf(i, 1) = il
listf(i, 2) = ist
listf(i, 3) = $iend$
listf(i, 4) = ir

where i is the index of the face, ist and $iend$ are the starting and ending grid points respectively (see Figure 1) and il and ir are the left and right cells respectively. Note that the vectors $\overrightarrow{(il)(ir)}$, $\overrightarrow{(ist)(iend)}$ and $\hat{z}$ form a right-hand-side co-ordinate system.

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

listc(i, 1) = neighbour 1
listc(i, 2) = neighbour 2
listc(i, ·) = neighbour ·
listc(i, n) = neighbour N.

Three items must be stored in listc (listf always contains four items irrespective of the type of control volume) on a mesh containing only triangles, one more must be stored for quadrilaterals and so on. Note that listf is stored for each *face*, whereas listc is stored for each *cell*. On a mesh containing only triangles (quadrilaterals), the number of cell faces is approximately 1·5 (2) times the number of cells.

The pointer system listc, which points to a cell's neighbours, is created from the pointer system for the faces, listf. For the cell in Figure 2, for example, the neighbouring cells for cell i are $i1$, $i2$ and $i3$. The numbering of $i1$, $i2$ and $i3$ in listc(i, j), where $j = i1$, $i2$ or $i3$, is determined by the order in which the connecting faces $if1$, $if2$ and $if3$ are visited when looping over all faces in the calculation domain. Thus, if $if1 < if2 < if3$, then the pointer listc for cell i reads

listc(i, 1) = $i1$
listc(i, 2) = $i2$
listc(i, 3) = $i3$.

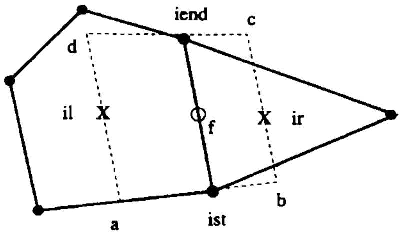

Figure 1. Two control volumes (cells). Grid points (cell vertices) are denoted by full circles and cell centres by crosses. The area denoted by broken lines (a–b–c–d) is used when computing the gradient at the face (open circle, index f)

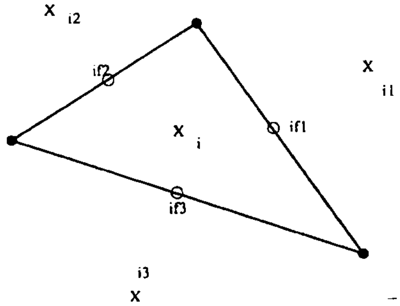

Figure 2. A triangular control volume i with its three faces $if1$, $if2$ and $if3$ and three neighbours $i1$, $i2$ and $i3$. Grid points are denoted by full circles, cell centres by crosses and cell faces by open circles

2.3. *Convection*

The convective term, which is the first part of the flux vector **J** in equation (2), 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 Figure 1 we obtain the normal velocity component as

$$U^f = \{Un_x + Vn_y\}_f \tag{4}$$

((n_x, n_y) is the unit normal vector), where the face values are computed as

$$U_f = f_1 U_{il} + (1 - f_1)U_{ir}, \qquad V_f = f_1 V_{il} + (1 - f_1)V_{ir}$$

and f_1 is a weight function for linear interpolation.

2.4. *Numerical dissipation*

2.4.1. Rhie–Chow pressure term. Collocated grid systems are often used in structured finite volume methods based on SIMPLE.[18–21] In order to prevent oscillations in the pressure field, a third derivative term in pressure, invented by Rhie and Chow,[22] is added to the convecting (normal) velocities. This term is added to the convections when calculating the continuity error in the pressure correction equation; it is not added to the convections in the U and V equations. It is often implemented as the difference between two first derivative terms, one evaluated at the centre of the cell and the other at the cell face. This third derivative term can be regarded as a stabilizing dissipation term similar to the fourth-order dissipation term added to all equations in time-marching compressible codes.[23–25] It has been found that in buoyancy-affected flows it is essential that this term is implemented as a true artificial dissipation term.[26]

In the present work this pressure gradient (which is a vector) is evaluated at the cell centres il and ir and is projected on (taking the dot product with) the normal vector at the face n_i, i.e.

$$\left\{\frac{\partial p}{\partial x_i}\right\}_{il} n_i, \qquad \left\{\frac{\partial p}{\partial x_i}\right\}_{ir} n_i.$$

At the cell face the pressure gradient is evaluated and projected on n_i as

$$\left\{\frac{\partial p}{\partial x_i}\right\}_f n_i = \frac{p_{ir} - p_{il}}{\overrightarrow{(il)(ir)}}.$$

The third derivative term is then obtained as

$$\frac{p_{ir} - p_{il}}{\overrightarrow{(il)(ir)}} - \left(f_1\left\{\frac{\partial p}{\partial x_i}\right\}_{il} + (1 - f_1)\left\{\frac{\partial p}{\partial x_i}\right\}_{ir}\right)n_i. \tag{5}$$

The mass flux at the cell faces can now be written as (see equations (4) and (5))

$$\dot{m}_f = \{\rho A U^f\}_f - \left\{\frac{\rho \mathcal{V} A}{a_P}\right\}_f \left[\frac{p_{ir} - p_{il}}{\overrightarrow{(il)(ir)}} - \left(f_1 \left\{\frac{\partial p}{\partial x_i}\right\}_{il} + (1 - f_1)\left\{\frac{\partial p}{\partial x_i}\right\}_{ir}\right) n_i\right], \tag{6}$$

where A and $\mathcal{V}$ denote the surface area and cell volume respectively and a_P is the discretized diagonal coefficient in the momentum equations (see equation (15)).

2.4.2. Fourth-order numerical dissipation. The convective terms in the U and V equations are discretized using central differences. In order to damp odd–even oscillations, a fourth-order dissipation term is added. This is computed by first evaluating the gradient $\Phi_{,i}n_i$ (the subscript comma denotes derivation with respect to x_i). The vector $\Phi_{,i}n_in_j$ is the gradient (at a face) normal to a face. The divergence of $\Phi_{,i}$ is obtained with the Gauss law

$$\Phi_{,ii} = \frac{1}{\delta \mathcal{V}} \int_A \Phi_{,i}n_in_jn_j \, \mathrm{d}A = \frac{1}{\delta \mathcal{V}} \int_A \Phi_{,i}n_i \, \mathrm{d}A.$$

In discrete form we replace the integral by a sum

$$f \equiv \Phi_{,ii}\delta \mathcal{V} = \sum_{k=1}^{N} \Phi_{,j}n_j\delta A \tag{7}$$

summed over all faces (index k) and where N is the number of faces of a cell. The fourth derivative is computed by repeating the procedure, now letting the Laplacian operate on f. The dissipation term is added in conservative form. The diagonal coefficient in the momentum equations, a_P, is used as a scaling factor (which also gives the correct dimension on the dissipation term):

$$\text{dissipation term} = \epsilon_4 \sum_{i=1}^{n} a_P f_{,j} n_j \delta A, \tag{8}$$

where ϵ_4 is a fixed constant. The term $\Phi_{,j}n_j\,\delta A$ in equation (7) is simply taken as $(\Phi_{ir} - \Phi_{il})\Delta z$; see Figure 1. Since we are dealing with two-dimensional flow, $\Delta z = 1$. The term $f_{,j}n_j\delta A$ in equation (8) is computed in the same way.

The fourth-order numerical dissipation was found to be necessary only in one of the present test cases (backward-facing step; see Section 3.2), where the value $\epsilon_4 = 0{\cdot}02$ was used. In the other two test cases in Section 3, pure central differencing was used with $\epsilon_4 = 0$.

2.5. *Diffusion*

Diffusion is the second part of the flux vector $\mathbf{J}$ in equation (2) and has the form

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

For the face f in Figure 1 we have

$$-\{\Gamma_\Phi \mathbf{A} \cdot \nabla\Phi\}_f = -\left\{\Gamma_\Phi\left(A_x\frac{\partial\Phi}{\partial x} + A_y\frac{\partial\Phi}{\partial y}\right)\right\}_f = -\left\{\Gamma_\Phi|\mathbf{A}|\left(n_x\frac{\partial\Phi}{\partial x} + n_y\frac{\partial\Phi}{\partial y}\right)\right\}_f, \tag{9}$$

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

$$n_x = \frac{y_{iend} - y_{ist}}{d}, \qquad n_y = -\frac{x_{iend} - x_{ist}}{d}$$

$$d = \sqrt{[(y_{iend} - y_{ist})^2 + (x_{iend} - x_{ist})^2]}.$$

We must now evaluate the derivatives $\partial\Phi/\partial x$ and $\partial\Phi/\partial y$ at the face. This is done by applying Green's formula to the volume *a–b–c–d* (see Figure 1) surrounding the midpoint of the face, i.e.

$$\frac{\partial\Phi}{\partial x}=\frac{1}{\mathcal{V}}\int_A \Phi n_x \,\mathrm{d}A, \qquad \frac{\partial\Phi}{\partial y}=\frac{1}{\mathcal{V}}\int_A \Phi n_y \,\mathrm{d}A,$$

where A $(=a\text{–}b\text{–}c\text{–}d)$ is the surface enclosing the volume $\mathcal{V}$. We obtain

$$\begin{aligned}\left[\frac{\partial\Phi}{\partial x}n_x\right]_f &= \left\{\frac{n_x}{\mathcal{V}}\right\}_f \{(\Phi n_x A)_{ir}+(\Phi n_x A)_{iend}-(\Phi n_x A)_{il}-(\Phi n_x A)_{ist},\\ \left[\frac{\partial\Phi}{\partial y}n_y\right] &= \left\{\frac{n_y}{\mathcal{V}}\right\}_f \{(\Phi n_y A)_{ir}+(\Phi n_y A)_{iend}-(\Phi n_y A)_{il}-(\Phi n_y A)_{ist}.\end{aligned} \tag{10}$$

Note that the exact co-ordinates of the volume *a–b–c–d* are not used. The definition is that its face vector areas are defined as $(\mathbf{n}A)_{il}=(\mathbf{n}A)_f=(\mathbf{n}A)_{ir}$ and $(\mathbf{n}A)_{ist}=(\mathbf{n}A)_{iend}$. The areas $(\mathbf{n}A)_{il}$ and $\delta(\mathbf{n}A)_{ir}$ are centred at their respective nodes. The normal vectors of $(\mathbf{n}A)_{ist}$ and $(\mathbf{n}A)_{iend}$ are normal to $\overrightarrow{(il)(ir)}$. The sum of the two terms in equation (10) can be written as

$$\begin{aligned}\left[\frac{\partial\Phi}{\partial x}n_x\right]_f+\left[\frac{\partial\Phi}{\partial y}n_y\right]_f &= \left\{\frac{1}{\mathcal{V}}\right\}_f \{A_f(\Phi_{ir}-\Phi_{il})\\ &\quad + A_{iend}[(n_x)_f(n_x)_{iend}+(n_y)_f(n_y)_{iend}](\Phi_{iend}-\Phi_{ist})\}\end{aligned} \tag{11}$$

using $(n_x^2+n_y^2)_f=1$. The second line represents orthogonal diffusion and the last line represents non-orthogonal diffusion, which vanishes on orthogonal grids, i.e. when $\overrightarrow{(il)(ir)}\cdot\overrightarrow{(ist)(iend)}=0$. The non-orthogonal terms are mostly negligible on smooth meshes based on quadrilaterals and often also on smooth meshes based on triangles. For stretched triangles, however, the non-orthogonal terms are not negligible.[10] Equations (9)–(11) can now be written as

$$\begin{aligned}-\{\Gamma_\Phi \mathbf{A}\cdot\nabla\Phi\}_f &= -\left\{\frac{\Gamma_\Phi|\mathbf{A}|}{\mathcal{V}}\right\}_f \{A_f(\Phi_{ir}-\Phi_{il})\\ &\quad + A_{iend}[(n_x)_f(n_x)_{iend}+(n_y)_f(n_y)_{iend}](\Phi_{iend}-\Phi_{ist},)\}\end{aligned} \tag{12}$$

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

$$\{\Gamma_\Phi \mathbf{A}\cdot\nabla\Phi\}_{f,\mathrm{ort}}=D_f\Gamma_{\Phi f}(\Phi_{ir}-\Phi_{il}). \tag{13}$$

2.6. *The discretized equation*

The numbering of the coefficients in `listc` is determined in the same way as the numbering of neighbouring cells when creating `listc` from `listf` (see Section 2.2). Thus, if $i1<i2<i3$ in Figure 2, then

`a(i, 1)` = `i1` is the connection coefficents from i to $i1$
`a(i, 2)` = `i2` is the connection coefficents from i to $i2$
`a(i, 3)` = `i3` is the connection coefficents from i to $i3$.

Combining equations (3), (6) and (13) for a cell with an arbitrary (*N*) number of neighbours gives

$$\{\dot{m}_{if1}\Phi_{if1} - D_{if1}(\Phi_{i1} - \Phi_i)\} + \{\dot{m}_{if2}\Phi_{if2} - D_{if2}(\Phi_{i2} - \Phi_i)\} + \cdots + \{\dot{m}_{ifN}\Phi_{ifN} - D_{ifN}(\Phi_{iN} - \Phi_i)\} = S^{\Phi}, \tag{14}$$

where the non-orthogonal diffusion terms have been included in the source term. Note that the mass fluxes are computed with a *cell's* outward-pointing normal vector, which means that the sign of the mass flux $\dot{m}_f$ depends on in which cell it is used, cell *il* or *ir* (see Figure 1). The cell face values of Φ are estimated with central differencing. The resulting discretized equation is

$$\begin{aligned}
a_P\Phi_i &= a_{if1}\Phi_{i1} + a_{if2}\Phi_{i2} + \cdots + a_{ifN}\Phi_{iN} + S_U^{\Phi},\\
a_{if1} &= -\tfrac{1}{2}\dot{m}_{if1} + D_{if1},\\
a_{if2} &= -\tfrac{1}{2}\dot{m}_{if2} + D_{if2},\\
&\ \vdots\\
a_{ifN} &= -\tfrac{1}{2}\dot{m}_{ifN} + D_{ifN},\\
a_P &= a_{if1} + a_{if2} + \cdots + a_{ifN} - S_P.
\end{aligned} \tag{15}$$

The coefficients in equation (15) 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. It is this counter which determines the values of $i1, i2, \ldots, iN$ ($1 \geq i1, i2, \ldots, iN \geq N$). In the grids used in the present work, the number of faces of the control volumes in the mesh varies between three and six.

The treatment of the source terms is described in detail in Reference 27.

2.7. *The pressure correction equation*

The pressure correction equation is derived from the continuity equation as in standard pressure correction procedures.[28] The discrete form of the continuity equation for a cell with an arbitrary number of faces (neighbours) can be written as

$$\dot{m}_{i1} + \dot{m}_{i2} + \cdots + \dot{m}_{ifN} = 0.$$

The mass flux at a face $\dot{m}_f$ is taken from equation (6) (see comment below equation (14)). The mass fluxes are split into an old value (previous iteration) $\dot{m}_f^*$ and a correction $\dot{m}'_f$, i.e.

$$\dot{m}_f = \dot{m}_f^* + \dot{m}'_f.$$

Using the truncated momentum equations, we obtain a relation between pressure and velocity so that

$$\dot{m}'_f = \rho_f A_f (U'\gamma^f = \left(\frac{\rho D\gamma}{a_P}\right)_f (p'_{il} - p'_{ir}), \tag{16}$$

where D_f is the diffusion coefficient in equation (13). The pressure correction equation can now be cast in the standard form of equation (15) with the coefficients

$$\begin{aligned} a_{if1} &= \left(\frac{\rho D\mathcal{V}}{a_P}\right)_{if1}, \\ a_{if2} &= \left(\frac{\rho D\mathcal{V}}{a_P}\right)_{if2}, \\ &\vdots \\ a_{ifN} &= \left(\frac{\rho D\mathcal{V}}{a_P}\right)_{ifN}, \\ S_U &= -\dot{m}^*_{if1} - \dot{m}^*_{if2} - \cdots - \dot{m}^*_{ifN}. \end{aligned} \tag{17}$$

After having solved the pressure correction equation, the pressure, mass flux and node velocities are corrected as

$$\begin{aligned} &\dot{m}_f = \dot{m}^*_f + \dot{m}'_f = \dot{m}^*_f + \rho_f A_f (U')^f = \dot{m}^*_f + \left(\frac{\rho D\mathcal{V}}{a_P}\right)_f (p'_{il} - p'_{ir}), \\ &p = p^* + p' \qquad U = U^* - \frac{\mathcal{V}}{a_P}\frac{\partial p'}{\partial x}, \qquad V = V^* - \frac{\mathcal{V}}{a_P}\frac{\partial p'}{\partial y}. \end{aligned} \tag{18}$$

Care must be taken when evaluating the derivatives of p' at boundaries. The p' field must satisfy $\partial p'/\partial n$ at all boundaries.

2.8. *The solution procedure*

The solution procedure is summarized below. It is similar to standard pressure correction procedures.[28]

1. The coefficients in the momentum equations are computed from equation (15) (looping over faces).
2. The numerical dissipation is computed (if $\epsilon_4 \neq 0$) from equation (8) (looping over faces).
3. The coefficient a_P in the U equation is computed from equation (15) and underrelaxation is introduced. This is done by looping over cells.
4. The discretized U equation is solved using a symmetric Gauss–Seidel solver (looping over cells).
5. Steps 2–4 are repeated for the V equation.
6. The coefficients for the p' equation are computed from equation (17) (looping over faces).
7. The discretized p' equation is solved (looping over cells).
8. The pressure and velocities are corrected using equation (18) (looping over cells). The mass flux is corrected using equation (18) (looping over faces).
9. The continuity error is computed (looping over faces).
10. Steps 1–9 are repeated until convergence is achieved.

3. RESULTS

The finite volume solver presented in the previous section is used here to calculate laminar flow in three two-dimensional configurations: backward-facing step, diverging channel and skewed driven cavity.

3.1. *Boundary conditions*

Zero normal gradient is prescribed at *all* boundaries for the pressure and pressure correction. No slip is used at the walls for the velocities. Zero streamwise gradient, $\partial/\partial x = 0$, is prescribed at the outlet.

3.2. *Backward-facing step*

Gartling[29] and Blosch *et al.*[30] have presented calculations for this configuration. The Reynolds number is 800 and the height of the step is half the channel width. The length of the calculation domain is 16 channel heights (the same as in Reference 29 and 30). At the inlet, U is set as a parabolic profile and V is set to zero.

The refined grid is shown in Figure 3. A structured 40 × 20 grid (quadrilaterals) is refined in the two recirculation regions. The refinement is done by splitting each quadrilateral into four new quadrilaterals. Note that this grid contains two cells which have six faces. These two cells are located at the corners of the coarse part of the mesh ($x \simeq 1{\cdot}8$, $y \simeq 0{\cdot}45$ and $x \simeq 5$, $y \simeq 0{\cdot}55$) and have refined cells as neighbours on two sides. This is a good illustration of the flexibility of the present method.

In Figure 4 the predicted U profiles are compared with the calculations in References 29 and 30 (the U profiles predicted in Reference 29 and 30 are identical). As can be seen, the U profile predicted with a 160 × 40 mesh (the same as in Reference 30) is identical with the benchmark calculation. The refined mesh produces a U profile much closer to the benchmark calculation than does the original 40 × 20 mesh.

The pressure contours are shown in Figure 5. It can be seen that there are some oscillations in the pressure near the interface of the refinement at $x = 5$. This is not very surprising considering the abrupt change in cell size across the interface, making the interpolation to faces inexact (see equation (4)). Apart from at the interfaces of the refinements, the pressure field is smooth.

3.3. *Diverging channel*

This configuration was used in a workshop in 1982 and the results were presented in Reference 31. The Reynolds number is 10. The inlet velocities are set as

$$U = 3(y - \tfrac{1}{2}y^2), \qquad V = 0.$$

First a fairly coarse grid based on triangles was used (see Figure 6(a)). Twenty-one cells were used along the lower (wall) boundary and 10 along the other three boundaries. The interior was filled with triangles, giving a grid with 319 cells. As can be seen in Figure 7, this gives a fairly good prediction of the vorticity at the wall, ω_w, although an increase in ω_w near the outlet is observed. The grid is refined near the outlet by doubling the number of cells on the outlet boundary (see Figure 6(b)), giving a mesh with 621 cells. It is seen in Figure 7 that the wall vorticity is better predicted near the outlet using this refined mesh.

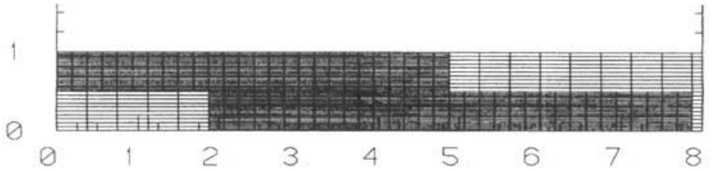

Figure 3. Backward-facing step. Upper and lower boundaries are walls. Inlet at the left boundary at $0 \le y \le 0{\cdot}5$. Outlet at the right boundary at $x = 16$. A structured 40 × 20 mesh refined in two regions: $0 \le x \le 5$, $0{\cdot}5 \le y \le 1$ and $2 \le x \le 8$, $0 \le y \le 0{\cdot}5$

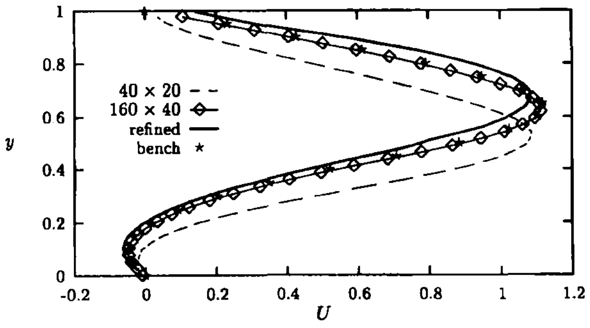

Figure 4. Backward-facing step. Calculated U profiles at $x=7$

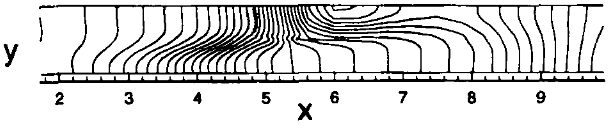

Figure 5. Backward-facing step. Calculated pressure contours

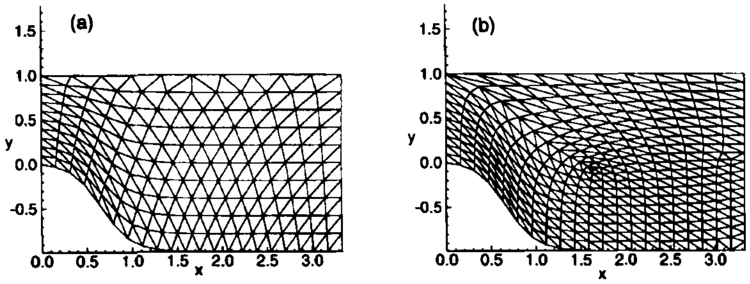

Figure 6. Diverging channel. Upper boundary is a symmetry line. Lower boundary is a wall. The inlet is at the left boundary and the right boundary is the outlet. (a) Coarse grid and (b) refined grid

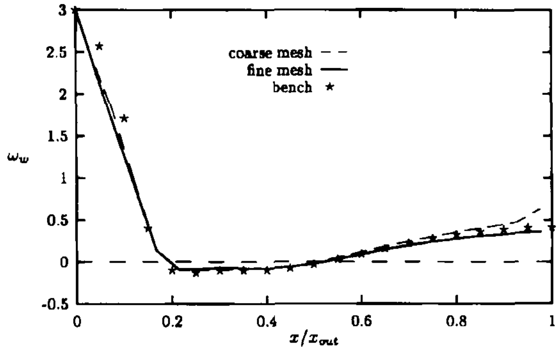

Figure 7. Diverging channel. Wall vorticity ω_w

The pressure contours are presented in Figure 8. It is seen that there are some oscillations in the pressure in the upper-left corner when using the refined mesh (Figure 8(b)). The triangles in this region are fairly distorted. The pressure field for the coarse grid is much smoother (Figure 8(a)) and so also is the mesh (Figure 6(a)). The dissipation term stemming from Rhie–Chow interpolations does not seem to have any adverse effect. This term was reduced by a factor of 10 without noticeably affecting the pressure field. The convergence rate, however, deteriorated significantly. The reason why the pressure field is sensitive to distorted cells is probably related to inexact interpolation from nodes to faces according to equation (4). A true bilinear interpolation would probably be better, but this is not easily accomplished on a cell-centred mesh. This is one advantage of cell vertex methods, where bilinear interpolation is easily implemented using base functions.

3.4. *Skewed driven cavity*

The configuration is shown in Figure 9. The flow in this configuration has been computed by Demirdžić *et al.*[32] and Oosterlee *et al.*[33] It is a configuration which gives very skewed cells if only quadrilaterals are used. The grid used in the present study is a mixed grid with both triangles and quadrilaterals; see Figure 10. The quadrilaterals in the grids in Figure 10 are fairly rectangular. The coarse and fine meshes in Figure 10 contain 1531 and 2707 cells respectively.

The U profile along the centreline CL1 (see Figure 9) and the V profile along $y=0{\cdot}5$ (CL2) are compared with benchmark computations in Figures 11 and 12 respectively for two different Reynolds numbers $Re=U_{\rm w}L/\nu$. As can be seen, the present predictions are in good agreement with the benchmark computations. For the higher Reynolds number the V profile with the fine mesh is slightly better predicted. Pressure contours are shown in Figure 13. They are smooth and regular except in the left part of the domain, where some oscillations appear. It should be noted that the pressure variations in this region are very small.

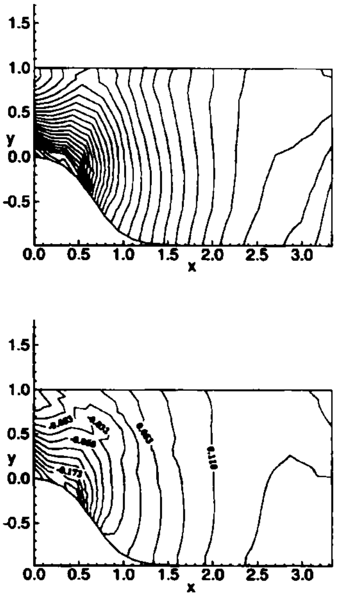

Figure 8. Diverging channel. Contours of constant pressure. (a) Coarse mesh and (b) refined mesh

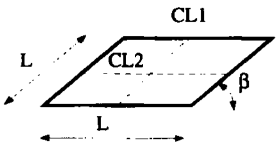

Figure 9. Skewed driven cavity. Configuration. Upper boundary is moving with velocity U_w. $\beta = 30°$

3.5. *Convergence data*

In this subsection the convergence data for the three cases are presented. For the pressure correction either a symmetric Gauss–Seidel solver is used or a conjugate gradient solver with incomplete Cholesky factorization preconditioning. For the momentum equations a symmetric Gauss–Seidel solver is used. A preconditioned biconjugate gradient solver was also tested for the momentum equations but was not found to improve the convergence data compared with the Gauss–Seidel solver. The conjugate gradient solvers in the SLAP library have been used (available at netlib).

Twenty sweeps on the pressure correction equation are made at each iteration when using the Gauss–Seidel solver, during which the residual is reduced typically from 1 to 0·8. The residual is defined as (see equations (15) and (17))

$$R^{\Phi} = \sum_{\text{all cells}} |a_P \Phi_i - a_{if1}\Phi_{if1} - a_{if2}\Phi_{if2} - \cdots - a_{ifN}\Phi_{iN} - S_U^{\Phi}|, \tag{19}$$

with $\Phi = p'$.

When using the conjugate gradient solver, the residual is reduced from 1 to 0·2, which takes between five and 20 sweeps. Driving down the residual to 0·1 did not have any large influence on the convergence data. The conjugate gradient solver is thus considerably more efficient than the Gauss–Seidel solver in solving the pressure correction equation. Unfortunately, this difference is not large

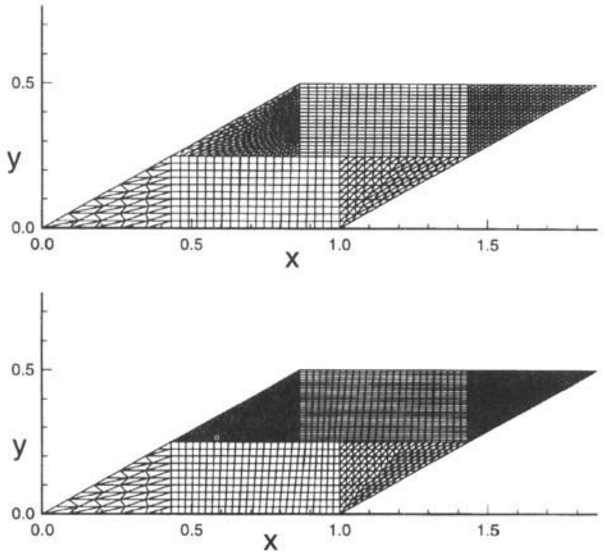

Figure 10. Skewed driven cavity. Two grids. Coarse (top) and refined (bottom)

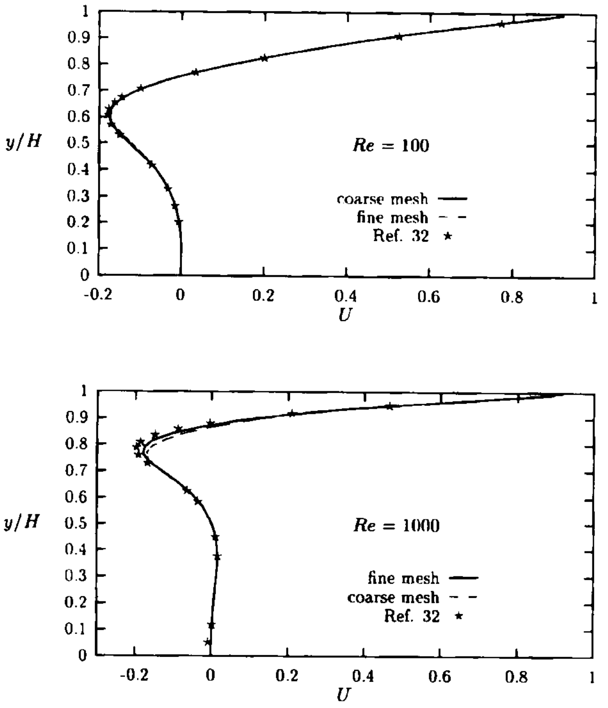

Figure 11. Skewed driven cavity. U velocity profiles along centreline CL1 (see Figure 9). $Re = 100$ (top) and 1000 (bottom)

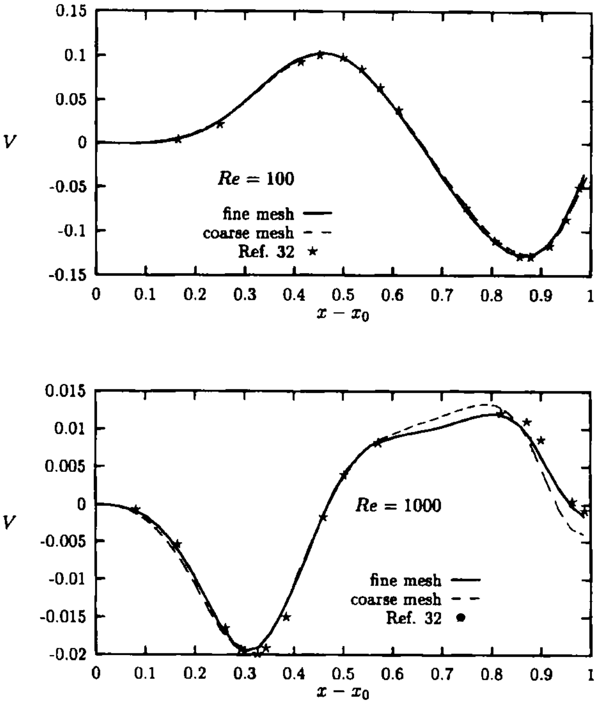

Figure 12. Skewed driven cavity. V velocity profiles along centreline CL2 (see Figure 9). $Re = 100$ (top) and 1000 (bottom)

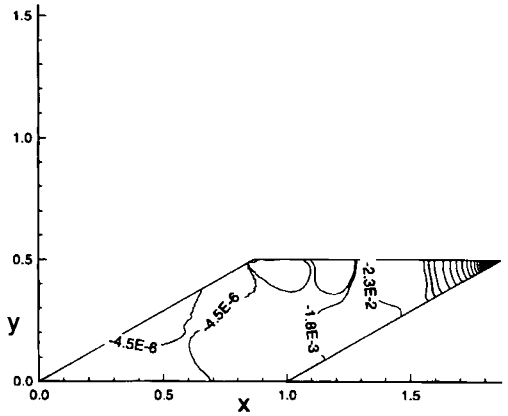

Figure 13. Skewed driven cavity. Contours of constant pressure. $Re = 1000$

enough for the convergence of the global equation system; see Figure 14 and Table I. The residual in Figure 14 is defined as

$$R = \max(R^{\Phi}), \quad \text{with } \Phi = U, \ V, \ p',$$

where R^{Φ} is taken from equation (19).

The only case where it is worthwhile to use a better solver is the backward-facing step. However, it should be remembered that these cases are small model cases.

4. CONCLUSIONS AND FUTURE WORK

A pressure correction procedure for two-dimensional, incompressible laminar flow has been presented. It is a cell-centred finite volume method and can be applied to general unstructured meshes in which the cells can have an arbitrary number of grid points and cell faces. The present method was applied to three test cases and it was demonstrated that the method gives results in agreement with benchmark calculations.

It is a simple matter to include transport turbulence models in the present method.[10] More work is needed in extending the method to three dimensions, but it is straightforward. Pressure correction methods are increasingly being used for compressible flow,[25,34–36] which means that the method can be extended to include compressible flows.

If the method is to be applied to high-Reynolds-number flow, it would be desirable to use a better discretization scheme than central differencing plus fourth-order numerical dissipation. Schemes used in structured methods such as MUSCL[37] (second-order, bounded) QUICK[38] (third-order, unbounded) and CHARM[39] (third-order, bounded) are not applicable without modification, owing to the problem of finding a node two cells upstream. Consider Figure 15, where we want to estimate the face value at face f when setting up the discretized equation for node i. Assume that the flow is directed from right to left. Using QUICK or MUSCL, for example, we need two nodes upstream of face f. We have node $i2$ and we could of course use nodes $i1$, $i2$ and $i3$ (possibly more nodes) to interpolate to point p. This would require a very complex pointer system, however. A better way is probably to use reconstruction schemes,[6,40,41] namely to compute the gradient in node $i2$ and then use Taylor expansion to obtain the value at point p, i.e.

$$\Phi_p = \Phi_{i2} + h\frac{\partial \Phi}{\partial \zeta} + \frac{h^2}{2}\frac{\partial^2 \Phi}{\partial \zeta^2},$$

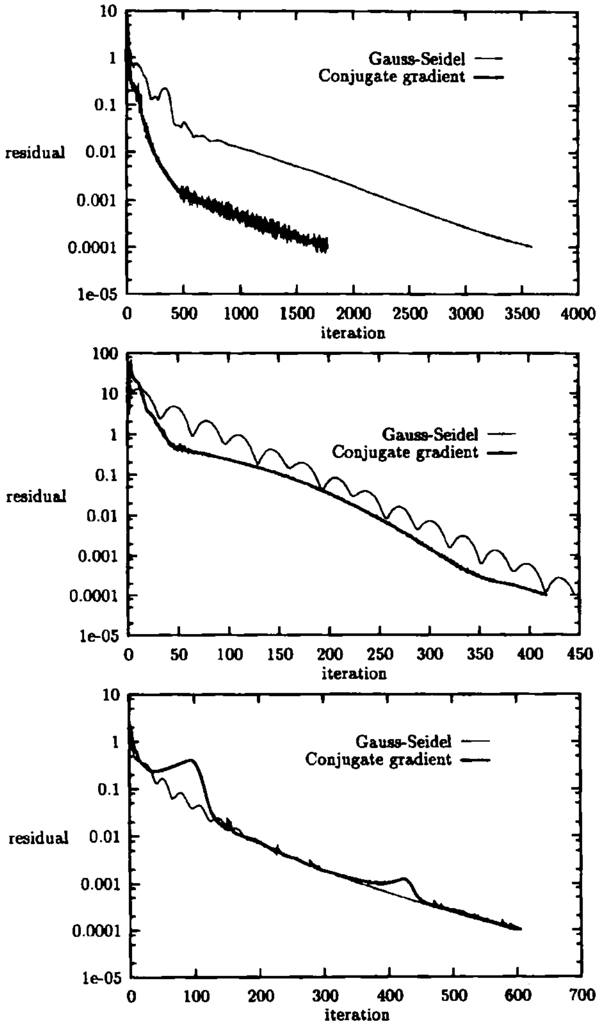

Figure 14. Convergence data comparing Gauss–Seidel solver and conjugate gradient with incomplete Cholesky factorization preconditioning for pressure correction equation. Backward-facing step (top), diverging channel (middle) and skewed driven cavity (bottom)

Table I. Convergence data. Symmetric Gauss–Seidel is used for the momentum equations. For the pressure correction equation either symmetric Gauss–Seidel (G–S) or conjugate gradient with incomplete Cholesky factorization preconditioning (CG) is used

Configuration	Solver	CPU (s)	Iterations
Backward-facing step	G– S	768	3597
Backward-facing step	CG	446	1769
Diverging channel	G–S	32	447
Diverging channel	CG	30	418
Driven cavity	G–S	112	597
Driven cavity	CG	118	606

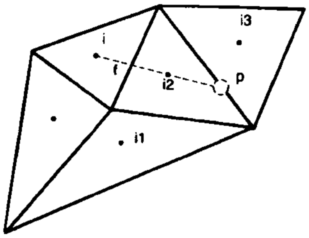

Figure 15. Reconstruction schemes employing two nodes upstream of a face when using higher-order discretization schemes

where h is the distance between $i2$ and p and $\zeta = \overrightarrow{(i)(i2)}$. Usually the last term containing the second derivative is omitted.

A further issue that needs attention is convergence acceleration. Multigrid methods, which have proved to be very efficient in structured methods,[42] would probably reduce the CPU time considerably.

REFERENCES

1. T. J. Barth, 'On unstructured grids and solvers', in *Computational Fluid Dynamics*, Lecture Notes 1990-03, von Karman Institute for Fluid Dynamics, Brussels, 1990.
2. D. G. Holmes and S. D. Connel, 'Solution of the 2D Navier–Stokes equations on unstructured adaptive grids', *AIAA Paper 89-1932-CP*, 1989.
3. D. J. Mavriplis and L. Martinelli, 'Multigrid solution of compressible turbulent flow on unstructured meshes using a two-equation model', *Int. j. numer. methods fluids*, **18**, 887–914 (1994).
4. C. J. Hwang and S. Y. Yang, 'Locally implicit total variation diminishing schemes on mixed quadrilateral–triangular meshes', *AIAA J.*, **31**, 2008–2015 (1993).
5. J. S. Mathur and N. P. Weatherill, 'The simulation of inviscid, compressible flows using an upwind kinetic method on unstructured grids', *Int. j. numer. methods fluids*, **15**, 59–82 (1992).
6. D. Pan and D. P. Cheng, 'Upwind finite-volume Navier–Stokes computations on unstructured triangular meshes', *AIAA J.*, **31**, 1618–1625 (1993).
7. Y. Kallinders, 'A finite volume Navier–Stokes algorithm for adaptive grids', *Int. j. numer. methods fluids*, **15**, 193–217 (1992).
8. L. Stolcis and L. J. Johnston, 'Solution of the Euler equations on unstructured grids for two-dimensional compressible flow', *Aeronaut. J.*, 181–195 (1990).
9. L. Stolcis and L. J. Johnston, 'Computation of the viscous flow around multi-element aerofoils using unstructured grids', in J. B. Vos, A. Rizzi and I. L. Rhyming (eds), *Notes on Numerical Fluid Mechanics*, Vol. 35, Vieweg, Braunschweig, 1992, pp. 311–320.
10. L. Davidson and L. Stolcis, 'An efficient and stable solution procedure of compressible turbulent flow on general unstructured meshes using transport turbulence models', *AIAA Paper 95-0342*, 1995.
11. R. D. Lonsdale and R. Webster, 'The application of finite volume methods for modelling 3D incompressible flow on unstructured mesh', in C. Taylor, P. Gresho, R. L. Sani and J. Haüser (eds), *Proc. 6th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow*, Vol. 6, Pineridge, Swansea, 1989, pp. 1615–1626.
12. B. R. Baliga and S. V. Patankar, 'A control-volume finite element method for two-dimensional fluid flow and heat transfer', *Numer. Heat Transfer*, **6**, 245–261 (1983).
13. C. Prakash and S. V. Patankar, 'A control volume-based finite element method for solving the Navier–Stokes equations using equal-order velocity– pressure interpolation', *Numer. Heat Transfer*, **8**, 259–280 (1985).
14. C. Masson, H. J. Saaba and B. R. Baliga, 'Collocated equal-order control-volume finite element method for two-dimensional axisymmetric incompressible fluid flow', *Int. j. numer. methods fluids*, **18**, 1–26 (1994).
15. M. Thomadakis and M. A. Leschziner, 'Numerical simulation of viscous incompressible flows using a pressure-correction method and unstructured grids', in S. Wagner, E. H. Hirschel, J. Périaux and R. Piva (eds), *Proc. 2nd Eur. Computational Fluid Dynamics Conf.*, Wiley, New York, 1994, pp. 325–332.
16. J. K. Watterson, 'A pressure-based flow solver for the three- dimensional Navier–Stokes equations on unstructured and adaptive meshes', *AIAA Paper 94-2358*, 1994.
17. Y. Jiang and A. J. Przekwas, 'Implicit, pressure-based incompressible Navier–Stokes equations solver for unstructured meshes', *AIAA Paper 94-0305*, 1994.
18. M. Peric, R. Kessler and G. Scheuerer, 'Comparison of finite-volume numerical methods with staggered and collocated grids', *Comput. Fluids*, **16**, 389–403 (1988).

19. F. S. Lien, 'Computational modelling of 3D flow in complex ducts and passages', *Ph.D. Thesis*, University of Manchester, 1992.
20. S. Majumdar, W. Rodi and J. Zhu, 'Three-dimensional finite-volume method for incompressible flows with complex boundaries', *ASME J. Fluids Eng.*, **114**, 496–503 (1992).
21. S. Johansson, L. Davidson and E. Olsson, 'Numerical simulation of vortex shedding past triangular cylinders at high Reynolds number', *Int. j. numer. methods fluids*, **16**, 859–878 (1993).
22. C. M. Rhie and W. L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA J.*, **21**, 1525–1532 (1983).
23. A. Jameson, W. Schmidt and E. Turkel, 'Numerical solutions of the Euler equations by finite volume methods with Runge–Kutta time stepping schemes', *AIAA Paper 81-1259*, 1981.
24. A. Rizzi, 'Damped Euler-equation method to compute transonic flow around wing-body combinations', *AIAA J.*, **20**, 1321–1328 (1982).
25. G. Zhou and L. Davidson, 'A pressure based Euler scheme for transonic internal and external flow simulation', *Int. J. Comput. Fluid Dyn.*, **5**, 169–188 (1995).
26. P. Johansson and L. Davidson, 'A modified collocated SIMPLEC algorithm applied to buoyancy affected turbulent flow using a multigrid solution procedure', *Numer. Heat Transfer B*, **28**, 39–57 (1995).
27. L. Davidson, 'Implementation of a k–ε model and a Reynolds stress model into a multiblock code', *Rep. CRS4-APPMATH-93- 21*, CRS4, Cagliari, 1993.
28. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
29. D. K. Gartling, 'A test problem for outflow boundary conditions—flow over a backward-facing step', *Int. j. numer. methods fluids*, **11**, 953–967 (1990).
30. E. Blosch, W. Shyy and R. Smith, 'The role of mass conservation in pressure-based algorithms', *Numer. Heat Transfer B*, **24**, 415- 429 (1993).
31. M. Napolitano and P. Orlando, 'Laminar flow in complex geometry: a comparison', *Int. j. numer. methods fluids*, **5**, 667–683 (1985).
32. I. Demirdžić, Ž. Lilek and M. Perić, 'Fluid flow and heat transfer test problems for non-orthogonal grids: bench-mark solutions', *Int. j. numer. methods fluids*, **15**, 329–354 (1992).
33. C. W. Oosterlee, P. Wesseling, A. Segal and E. Brakkee, 'Benchmark solutions for the incompressible Navier–Stokes equations in general co- ordinates on staggered grids', *Int. numer. methods fluids*, **17**, 301–321 (1993).
34. J. J. McGuirk and G. Page, 'Shock capturing using a pressure-correction method', *AIAA Paper 89-0561*, 1989.
35. F. S. Lien and M. A. Leschziner, 'A pressure–velocity solution strategy for compressible flow and its application to shock/boundary layer interaction using second-moment turbulence closure', *ASME J. Fluids Eng.*, **115**, 717–725 (1993).
36. G. Zhou, L. Davidson and E. Olsson, 'Turbulent transonic airfoil flow simulation using a pressure based algorithm', *AIAA J.*, **33**, 42–47 (1995).
37. B. Van Leer, 'Towards the ultimate conservative difference scheme. V. A second-order sequel to Godonov's method', *J. Comput. Phys.*, **32**, 101–136 (1979).
38. B. P. Leonard, 'A stable and accurate convective modelling based on quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
39. G. Zhou, L. Davidson and E. Olsson, 'Transonic inviscid/turbulent airflow flow simulations using a pressure based method with higher order schemes', *Proc. 14th Int. Conf. on Numerical Methods in Fluid Dynamics*, Bangalore, July 1994.
40. T. Barth and D. C. Jespersen, 'The design and application of upwind schemes on unstructured meshes', *AIAA Paper 89-0366*, 1989.
41. M. Aftosmis, D. Gaitone and T. S. Tavares, 'On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes', *AIAA Paper 94-0415*, 1994.
42. P. Johansson and L. Davidson, 'A full multigrid method applied to turbulent flow using the SIMPLEC algorithm together with a collocated grid arrangement', *Multigrid Methods IV*, Birkhaüser, 1994, pp. 245–256.