INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 15, 193–217 (1992)

A FINITE VOLUME NAVIER–STOKES ALGORITHM FOR ADAPTIVE GRIDS

YANNIS KALLINDERIS

Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712, U.S.A

SUMMARY

A compact, finite volume, time-marching scheme for the two-dimensional Navier–Stokes equations of viscous fluid flow is presented. The scheme is designed for unstructured (locally refined) quadrilateral meshes. An earlier inviscid equation (Euler) scheme is employed for the convective terms and the emphasis is on treatment of the viscous terms. An essential feature of the algorithm is that all necessary operations are restricted to within each cell, which is very important when dealing with unstructured grids. Numerical issues which have to be addressed when developing a Navier–Stokes scheme are investigated. These issues are not limited to the particular Navier–Stokes scheme developed in the present work but are general problems. Specifically, the extent of the numerical molecule, which is related to the compactness of the scheme and to its suitability for unstructured grids, is examined. An approach which considers suppression of odd–even mode decoupling of the solution when designing a scheme is presented. In addition, accuracy issues related to grid stretching as well as boundary layer solution contamination due to artificial dissipation are addressed. Although the above issues are investigated with respect to the specific scheme presented, the conclusions are valid for an entire class of finite volume algorithms. The Navier–Stokes solver is validated through test cases which involve comparisons with analytical, numerical and experimental results. The solver is coupled to an adaptive algorithm for high-Reynolds-number aerofoil flow computations.

KEY WORDS Navier–Stokes Finite volume Viscous terms Unstructured grids

1. INTRODUCTION

Numerical simulations of flow fields are becoming of increasing importance for engineering applications. Various simplified sets of the governing equations have been used, since solution of the full Navier–Stokes equations that govern fluid flow is quite expensive. However, the complexity of flows encountered in modern applications requires solution of the full Navier–Stokes equations. Adaptive algorithms are a new generation of algorithms which change form and structure during the solution process in order to achieve accuracy with minimum computational effort. The first adaptive scheme for turbulent flows was developed in References 1–3. The algorithm uses adaptive, locally embedded quadrilateral grids in order to resolve local flow features.

Currently, efficient inviscid equation (Euler) solvers are being used for applications.[4, 5] Accuracy and efficiency issues have been addressed to some extent and these solvers are quite established. The next step is to extend these Euler schemes to include viscous terms.[1, 3, 6–8] There are special numerical problems associated with the treatment of viscous terms. These terms pose

0271-2091/92/140193-25$17.50

Received 14 September 1990
Revised 7 November 1991

significant accuracy problems which have to be addressed when developing a Navier–Stokes scheme. The viscous terms contain higher-order derivatives compared to the convective terms and therefore a larger numerical molecule is normally required. This results in a scheme with different molecules for the convective and viscous terms, which is undesirable, especially when an unstructured grid algorithm is employed.[1, 3] Such algorithms require a scheme which is compact and with all operations restricted to within each grid cell. Grid stretching poses severe accuracy problems, since numerical approximation of second-order derivatives of the viscous terms is more susceptible to such errors compared to evaluation of the inviscid terms that consist of first-order derivatives.[9]

Central space-differencing schemes are usually susceptible to odd–even decoupling of the solution. Both one- and two-direction sawtooth modes may appear, which makes convergence slower and in some cases may prevent convergence of the algorithm. Usually, artificial dissipation is employed in order to suppress these modes. Special care has to be taken in order to devise a scheme which does not allow these modes to appear. Smoothing may still be required in the inviscid regions of the domain as well as for capturing shocks. However, the presence of artificial dissipation terms in viscous regions may seriously deteriorate the accuracy, since these smoothing terms may 'mask' the physical viscous terms.

The emphasis in the present work is on the design of a numerical scheme for the viscous terms of the Navier–Stokes equations with unstructured grids by taking into account all the above issues. Although the numerical problems are investigated with respect to the specific scheme presented, the conclusions are valid for an entire class of finite volume algorithms. In addition, cases that involve comparisons with available data are important in order to validate the accuracy of the proposed solver.

In the following the discretization of the inviscid terms of the governing equations is presented first. Then the treatment of the viscous terms with embedded grids is described. The numerical issues of conservation, time step restriction, grid-stretching effect on accuracy as well as suppression of the sawtooth modes are addressed in connection with the proposed scheme. Then the artificial dissipation operator is presented and its effect on the accuracy of the viscous region solution is examined together with the determination of optimum smoothing coefficients. Finally the accuracy of the Navier–Stokes solver is investigated through test cases which involve comparisons with analytical, numerical and experimental results.

2. GOVERNING EQUATIONS

The system of two-dimensional Navier–Stokes equations may be written in Cartesian two-dimensional conservation form as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\frac{\partial \mathbf{R}}{\partial x}+\frac{\partial \mathbf{S}}{\partial y}, \tag{1}$$

where

$$\mathbf{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix},\qquad \mathbf{F}=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{pmatrix},\qquad \mathbf{G}=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E+p)v\end{pmatrix}$$

are state and convective flux vectors in the x- and y-direction respectively. The viscous flux vectors are

$$\mathbf{R}=\begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx}+v\tau_{xy}-q_x \end{pmatrix}, \qquad \mathbf{S}=\begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy}+v\tau_{yy}-q_y \end{pmatrix},$$

where τ_{xx}, τ_{yy} and τ_{xy} are viscous stresses and q_x and q_y are heat conduction terms. In the above relations ρ is the density, u and v are the velocity components, E is the total internal energy per unit volume and p is the pressure.

3. INVISCID TERMS

A one-step Lax–Wendroff-type integration scheme due to Ni[5] has been employed for discretization of the convective terms of the Navier–Stokes equations. Omitting the viscous terms, the Euler approximation is

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=0, \tag{2}$$

where $\mathbf{U}$ is the state variable vector and $\mathbf{F}$ and $\mathbf{G}$ are the convective term flux vectors as defined in (1). Integration of the above relation over the area of a cell leads to the equivalent integral relation

$$\frac{\mathrm{d}}{\mathrm{d}t}\iint_{\text{cell area}} \mathbf{U}\,\mathrm{d}x\,\mathrm{d}y+\oint_{\text{cell faces}} (\mathbf{F}\,\mathrm{d}y-\mathbf{G}\,\mathrm{d}x)=0. \tag{3}$$

The first term in the above relation represents the change in time of the state vector over the cell area S and is discretized as $(\Delta\mathbf{U}/\Delta t)S$, where $\mathbf{U}$ is the state vector value at the centre of a cell and $\Delta\mathbf{U}=\mathbf{U}^{n+1}-\mathbf{U}^{n}$ is its change over one time step. The second term in (3) represents the convective fluxes across the cell faces and is computed via the trapezoidal integration rule.

The state vector change in time, $\Delta\mathbf{U}$, at the centre of the cell has to be distributed to the corners. Consider the change $\delta\mathbf{U}$ at a grid node and the Taylor series expansion in time

$$\delta\mathbf{U}=\mathbf{U}^{n+1}-\mathbf{U}^{n}\approx\Delta t\left(\frac{\partial \mathbf{U}}{\partial t}\right)^{n}+\frac{(\Delta t)^2}{2}\left(\frac{\partial^2 \mathbf{U}}{\partial t^2}\right)^{n}. \tag{4}$$

The second-order time term in (4) is calculated following the Lax–Wendroff[10] approach, which replaces time derivatives with spatial derivatives from the governing equation.

Finally the following formulae for the distribution of the time change $\Delta\mathbf{U}$ to the cell corners are derived:

$$(\delta U)_{\mathrm{sw}}=\tfrac{1}{4}(\Delta U-\Delta f-\Delta g), \tag{5a}$$

$$(\delta U)_{\mathrm{nw}}=\tfrac{1}{4}(\Delta U-\Delta f+\Delta g), \tag{5b}$$

$$(\delta U)_{\mathrm{ne}}=\tfrac{1}{4}(\Delta U+\Delta f+\Delta g), \tag{5c}$$

$$(\delta U)_{\mathrm{se}}=\tfrac{1}{4}(\Delta U+\Delta f-\Delta g), \tag{5d}$$

where

$$\Delta f \equiv \frac{\Delta t}{S}(\Delta \mathbf{F} \Delta y^l - \Delta \mathbf{G} \Delta x^l), \tag{6a}$$

$$\Delta g = \frac{\Delta t}{S}(\Delta \mathbf{G} \Delta x^m - \Delta \mathbf{F} \Delta y^m), \tag{6b}$$

$$\Delta \mathbf{F} \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \Delta \mathbf{U}, \qquad \Delta \mathbf{G} \equiv \frac{\partial \mathbf{G}}{\partial \mathbf{U}} \Delta \mathbf{U}.$$

The subscripts sw, nw, ne and se denote the four corners of a grid cell. The cell metric terms Δx^l, Δy^l, Δx^m and Δy^m are defined as

$$\Delta x^l = 0{\cdot}5(x_{\text{nw}} + x_{\text{ne}} - x_{\text{sw}} - x_{\text{se}}), \tag{7a}$$

$$\Delta y^l = 0{\cdot}5(y_{\text{nw}} + y_{\text{ne}} - y_{\text{sw}} - y_{\text{se}}), \tag{7b}$$

$$\Delta x^m = 0{\cdot}5(x_{\text{ne}} + x_{\text{se}} - x_{\text{nw}} - x_{\text{sw}}), \tag{7c}$$

$$\Delta y^m = 0{\cdot}5(y_{\text{ne}} + y_{\text{se}} - y_{\text{nw}} - y_{\text{sw}}), \tag{7d}$$

where *l* and *m* denote the two cell directions. It is important to note that all necessary operations for the above scheme are confined to the interior of each grid cell; no information is required from the exterior of the cell.

The inviscid multiple-grid operator suggested by Ni[5] is employed in order to accelerate wave propagation and therefore convergence. This involves a series of coarser grids which are defined from the immediately finer grid by deleting every other line. The operator uses the distributed changes δU to the corners and redistributes them to coarser grids. A detailed investigation of the inviscid scheme with the multiple-grid accelerator can be found in Reference 11.

4. VISCOUS TERMS

A novel finite volume scheme for the viscous terms of the Navier–Stokes equations will now be developed. The emphasis is on constructing a compact scheme suitable for unstructured grid computations. The change in time of the state variables which is contributed to by the viscous terms is also given by the Taylor series expansion (4). However, the Jacobians $\partial \mathbf{F}/\partial \mathbf{U}$ and $\partial \mathbf{G}/\partial \mathbf{U}$ involved in the expansion include viscous terms as well. This makes the scheme expensive and complicated, especially when the extra terms due to a turbulence model are included.[12] Moreover, the use of the same grid cells for the evaluation of the viscous spatial derivatives results in a larger computational molecule than the one required solely for the inviscid terms. This is an undesirable property when unstructured meshes are employed. Finally, application of boundary conditions becomes difficult owing to the large stencil of the scheme.

In view of the above considerations, a compact scheme was developed such that all necessary operations are restricted to within each cell. Only the first-order temporal term in the Taylor expansion (4) is kept, which yields a scheme that is first-order accurate in time and which eliminates the need to compute viscous term Jacobians. The development of the scheme starts with a consideration of the viscous part of the Navier–Stokes equations in integral form:

$$\iint \frac{\partial \mathbf{U}}{\partial t}\,\mathrm{d}x\mathrm{d}y = \iint \left(\frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y}\right)\mathrm{d}x\mathrm{d}y = \oint (\mathbf{R}\,\mathrm{d}y - \mathbf{S}\,\mathrm{d}x). \tag{8}$$

The terms represented by $\mathbf{R}$ and $\mathbf{S}$ include stress and heat conduction terms and involve first-order spatial derivatives. Two issues must be addressed when developing a scheme to evaluate the viscous terms in (8): the first is the choice of the cell Ω which will be employed to evaluate the line integral $\oint_\Omega(\mathbf{R}\,dy-\mathbf{S}\,dx)$; the second is the choice of a discretization scheme for the terms $\mathbf{R}$ and $\mathbf{S}$.

A primary concern in the choice of Ω is maintaining the numerical molecule relatively small and identical to that employed for evaluation of the inviscid terms. The staggered cell ABCD shown in Figure 1 is used here and will be referred to as the *primary* cell. A similar cell was employed in Reference 6 for a finite difference discretization scheme of the viscous terms.

The unsteady term in (8) is discretized as $(\Delta\mathbf{U}_{\text{vis}}/\Delta t)S$. The term $\Delta\mathbf{U}_{\text{vis}}$ represents the change in time of the state vector due to the viscous terms only. Performing the line integration around cell ABCD using the midpoint rule results in

$$\begin{aligned}\Delta\mathbf{U}_{\text{vis}}&=\frac{\Delta t}{S}\oint_{\text{ABCD}}(\mathbf{R}\,dy-\mathbf{S}\,dx)\\&=\frac{\Delta t}{S}(+\mathbf{R}_{\text{da}}\Delta y_{\text{da}}-\mathbf{S}_{\text{da}}\Delta x_{\text{da}}\\&\quad+\mathbf{R}_{\text{cd}}\Delta y_{\text{cd}}-\mathbf{S}_{\text{cd}}\Delta x_{\text{cd}}\\&\quad+\mathbf{R}_{\text{bc}}\Delta y_{\text{bc}}-\mathbf{S}_{\text{bc}}\Delta x_{\text{bc}}\\&\quad+\mathbf{R}_{\text{ab}}\Delta y_{\text{ab}}-\mathbf{S}_{\text{ab}}\Delta x_{\text{ab}}).\end{aligned}\tag{9}$$

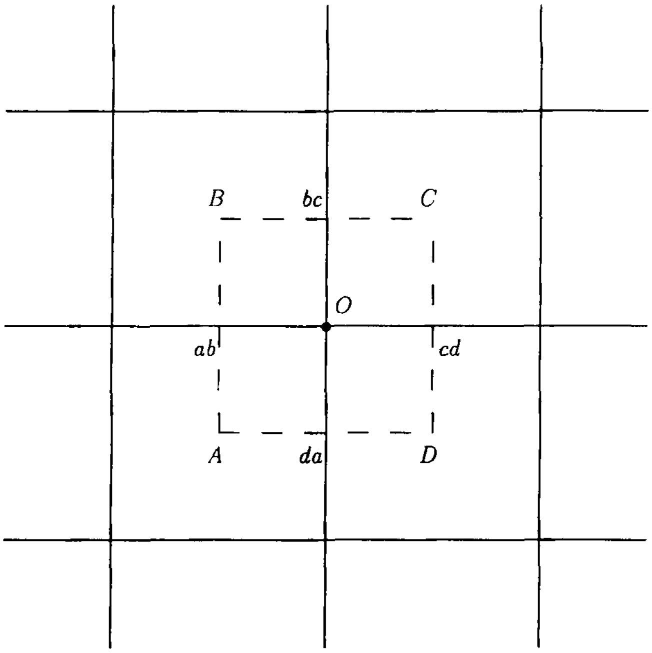


Figure 1. Primary cell for viscous terms discretization

In the above expression the **R**- and S-terms are evaluated at midfaces da, cd, bc and ab of the *primary* cell ABCD. The metric terms are

$$\Delta y_{\mathrm{da}} = y_{\mathrm{d}} - y_{\mathrm{a}}, \qquad \Delta x_{\mathrm{da}} = x_{\mathrm{d}} - x_{\mathrm{a}}, \qquad \text{etc.}$$

Finally, the time step Δt and the area S refer to cell ABCD. It is important to note that the change in time, $\Delta \mathbf{U}_{\mathrm{vis}}$ is evaluated at each grid node directly, as opposed to the inviscid counterpart which is evaluated at the cell centres first and then distributed to the nodes.

It remains to evaluate the terms **R** and S, which consist of first-order spatial derivatives. Green's theorem is applied again in order to discretize the derivatives.[1, 13] The evaluation can be explained by considering the derivative $(\partial u/\partial x)_{\mathrm{cd}}$ of variable u at point cd (Figure 2). The cell with corners at points da, bc, C′ and D′ will be referred to as the *secondary* cell.

Performing the line integration around the *secondary* cell, it follows that

$$\left(\frac{\partial u}{\partial x}\right)_{\mathrm{cd}} = \frac{1}{S_{\mathrm{cd}}} \oint_{(\mathrm{da},\,\mathrm{D}',\,\mathrm{C}',\,\mathrm{bc})} u\,\mathrm{d}y = \frac{1}{S_{\mathrm{cd}}} [+u_{\mathrm{D}}(y_{\mathrm{D}'} - y_{\mathrm{da}}) + u_{\mathrm{O}'}(y_{\mathrm{C}'} - y_{\mathrm{D}'}) + u_{\mathrm{C}}(y_{\mathrm{bc}} - y_{\mathrm{C}'}) + u_{\mathrm{O}}(y_{\mathrm{da}} - y_{\mathrm{bc}})]. \qquad (10)$$

The values of u at cell centres D and C are obtained by averaging the corresponding values at the corners of the grid cells with centre points D and C, and S_{cd} is the area of the *secondary* cell around point cd. Similar cells surrounding points bc, ab and da are employed to evaluate derivatives at these points.

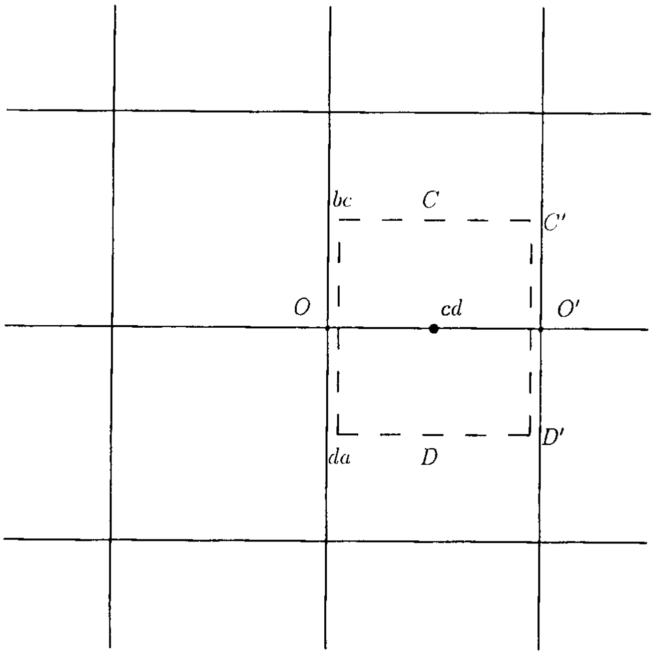


Figure 2. Secondary cell for stress term discretization

4.1. Operations restricted to within each cell

Adaptive algorithms have been developed for more efficient computations. They utilize unstructured grids and may use various sets of governing equations in different regions of the domain.[1] As a consequence it is essential for these algorithms that all scheme operations are restricted to within each cell without any information required from outside the cell. The above described scheme will now be modified to a compact form which is most suitable for unstructured grid computations. Care is taken so that no assumptions are made that would prohibit extension of the 2D scheme to three dimensions.

Equations (9) and (10) require information from the four cells A, B, C and D (Figure 1) which surround grid node O, and scheme operations must be split and allocated to these cells. More specifically, the terms in (9) can be rearranged as

$$\begin{aligned}\Delta\mathbf{U}_{\text{vis}} = \frac{1}{2}\frac{\Delta t}{S}[&+(\mathbf{R}_{\text{D}}+\mathbf{R}_{\text{A}})(+\Delta y^m_{\text{D}}+\Delta y^m_{\text{A}})-(\mathbf{S}_{\text{D}}+\mathbf{S}_{\text{A}})(+\Delta x^m_{\text{D}}+\Delta x^m_{\text{A}})\\&+(\mathbf{R}_{\text{C}}+\mathbf{R}_{\text{D}})(+\Delta y^l_{\text{C}}+\Delta y^l_{\text{D}})-(\mathbf{S}_{\text{C}}+\mathbf{S}_{\text{D}})(+\Delta x^l_{\text{C}}+\Delta x^l_{\text{D}})\\&+(\mathbf{R}_{\text{B}}+\mathbf{R}_{\text{C}})(-\Delta y^m_{\text{B}}-\Delta y^m_{\text{C}})-(\mathbf{S}_{\text{B}}+\mathbf{S}_{\text{C}})(-\Delta x^m_{\text{B}}-\Delta x^m_{\text{C}})\\&+(\mathbf{R}_{\text{A}}+\mathbf{R}_{\text{B}})(-\Delta y^l_{\text{A}}-\Delta y^l_{\text{B}})-(\mathbf{S}_{\text{A}}+\mathbf{S}_{\text{B}})(-\Delta x^l_{\text{A}}-\Delta x^l_{\text{B}})].\end{aligned} \tag{11}$$

Each of the terms $\mathbf{R}_{\text{A}}$, $\mathbf{S}_{\text{A}}$, $\mathbf{R}_{\text{B}}$, $\mathbf{S}_{\text{B}}$, etc. consists of terms related to cells A and B respectively. More precisely, $\mathbf{R}_{\text{A}}$ denotes the part of the stress term **R**-expression at points da and ab which originates in cell A. The rest of the **R**- and S-terms have analogous meanings. The metric terms Δx^m, Δy^m and Δx^l, Δy^l denote distances along the two directions m and l of a cell and have been defined in (7). The next step assumes that there is no appreciable stretching in the grid, which implies that $\Delta y^m_{\text{D}}\approx\Delta y^m_{\text{A}}$, $\Delta x^m_{\text{D}}\approx\Delta x^m_{\text{A}}$, etc. and $\mathbf{S}_{\text{A}}\approx\mathbf{S}_{\text{B}}\approx\mathbf{S}_{\text{C}}\approx\mathbf{S}_{\text{D}}$. It is also assumed that $\Delta t_{\text{A}}\approx\Delta t_{\text{B}}\approx\Delta t_{\text{C}}\approx\Delta t_{\text{D}}=\Delta t$. The above assumption is already implicitly made when using Green's theorem in order to obtain values at the centres of cells. The grid-stretching issue will be investigated in Section 5.4. In practice it has been found that levels of stretching up to 30% are acceptable by the scheme. On the basis of the above assumptions, equation (11) becomes

$$\begin{aligned}\Delta\mathbf{U}_{\text{vis}} = \frac{\Delta t}{S}[&+(+\mathbf{R}_{\text{D}}\Delta y^m_{\text{D}}+\mathbf{R}_{\text{A}}\Delta y^m_{\text{A}})-(+\mathbf{S}_{\text{D}}\Delta x^m_{\text{D}}+\mathbf{S}_{\text{A}}\Delta x^m_{\text{A}})\\&+(+\mathbf{R}_{\text{C}}\Delta y^l_{\text{C}}+\mathbf{R}_{\text{D}}\Delta y^l_{\text{D}})-(+\mathbf{S}_{\text{C}}\Delta x^l_{\text{C}}+\mathbf{S}_{\text{D}}\Delta x^l_{\text{D}})\\&+(-\mathbf{R}_{\text{B}}\Delta y^m_{\text{B}}-\mathbf{R}_{\text{C}}\Delta y^m_{\text{C}})-(-\mathbf{S}_{\text{B}}\Delta x^m_{\text{B}}-\mathbf{S}_{\text{C}}\Delta x^m_{\text{C}})\\&+(-\mathbf{R}_{\text{A}}\Delta y^l_{\text{A}}-\mathbf{R}_{\text{B}}\Delta y^l_{\text{B}})-(-\mathbf{S}_{\text{A}}\Delta x^l_{\text{A}}-\mathbf{S}_{\text{B}}\Delta x^l_{\text{B}})]\\=\frac{\Delta t}{S}[&+(+\mathbf{R}^{\text{W}}_{\text{D}}\Delta y^m_{\text{D}}+\mathbf{R}^{\text{E}}_{\text{A}}\Delta y^m_{\text{A}})-(+\mathbf{S}^{\text{W}}_{\text{D}}\Delta x^m_{\text{D}}+\mathbf{S}^{\text{E}}_{\text{A}}\Delta x^m_{\text{A}})\\&+(+\mathbf{R}^{\text{S}}_{\text{C}}\Delta y^l_{\text{C}}+\mathbf{R}^{\text{N}}_{\text{D}}\Delta y^l_{\text{D}})-(+\mathbf{S}^{\text{S}}_{\text{C}}\Delta x^l_{\text{C}}+\mathbf{S}^{\text{N}}_{\text{D}}\Delta x^l_{\text{D}})\\&+(-\mathbf{R}^{\text{E}}_{\text{B}}\Delta y^m_{\text{B}}-\mathbf{R}^{\text{W}}_{\text{C}}\Delta y^m_{\text{C}})-(-\mathbf{S}^{\text{E}}_{\text{B}}\Delta x^m_{\text{B}}-\mathbf{S}^{\text{W}}_{\text{C}}\Delta x^m_{\text{C}})\\&+(-\mathbf{R}^{\text{N}}_{\text{A}}\Delta y^l_{\text{A}}-\mathbf{R}^{\text{S}}_{\text{B}}\Delta y^l_{\text{B}})-(-\mathbf{S}^{\text{N}}_{\text{A}}\Delta x^l_{\text{A}}-\mathbf{S}^{\text{S}}_{\text{B}}\Delta x^l_{\text{B}})].\end{aligned} \tag{12}$$

The superscripts of the **R**- and **S**-terms refer to cell faces. For example, $\mathbf{R}^{\text{W}}_{\text{D}}$ implies the part of the line integration around the *secondary* cell (da, bc, C′, D′), which corresponds to the west face of cell D. Each cell contributes a part of the line integration around the *primary* cells which surround its corners, and the cell contributions to the change in time, $\Delta\mathbf{U}_{\text{vis}}$, of state vector **U** due to the viscous terms follow from (12):

$$(\Delta U_{\mathrm{vis}})_{\mathrm{sw}}=\frac{\Delta t}{S}\,[(+\mathbf{R}^{\mathrm{S}}\Delta y^{l}-R^{\mathrm{W}}\Delta y^{m})-(+\mathbf{S}^{\mathrm{S}}\Delta x^{l}-\mathbf{S}^{\mathrm{W}}\Delta x^{m})], \tag{13a}$$

$$(\Delta U_{\mathrm{vis}})_{\mathrm{nw}}=\frac{\Delta t}{S}\,[(+\mathbf{R}^{\mathrm{N}}\Delta y^{l}+\mathbf{R}^{\mathrm{W}}\Delta y^{m})-(+\mathbf{S}^{\mathrm{N}}\Delta x^{l}+\mathbf{S}^{\mathrm{W}}\Delta x^{m})], \tag{13b}$$

$$(\Delta U_{\mathrm{vis}})_{\mathrm{ne}}=\frac{\Delta t}{S}\,[(-\mathbf{R}^{\mathrm{N}}\Delta y^{l}+\mathbf{R}^{\mathrm{E}}\Delta y^{m})-(-\mathbf{S}^{\mathrm{N}}\Delta x^{l}+\mathbf{S}^{\mathrm{E}}\Delta x^{m})], \tag{13c}$$

$$(\Delta U_{\mathrm{vis}})_{\mathrm{se}}=\frac{\Delta t}{S}\,[(-\mathbf{R}^{\mathrm{S}}\Delta y^{l}-\mathbf{R}^{\mathrm{E}}\Delta y^{m})-(-\mathbf{S}^{\mathrm{S}}\Delta x^{l}-\mathbf{S}^{\mathrm{E}}\Delta x^{m})], \tag{13d}$$

where the subscripts denote the four corners of the cell.

It remains to describe how first-order derivative expressions such as in equation (10) are split into separate contributions from the cells involved. From (10) we obtain

$$\begin{aligned}(u_x)_{\mathrm{cd}}&=\frac{1}{S_{\mathrm{cd}}}\,[U_{\mathrm{O}'}(y_{\mathrm{C}'}-y_{\mathrm{D}'})+U_{\mathrm{C}}(y_{\mathrm{bc}}-y_{\mathrm{C}'})+U_{\mathrm{O}}(y_{\mathrm{da}}-y_{\mathrm{bc}})+U_{\mathrm{D}}(y_{\mathrm{D}'}-y_{\mathrm{da}})]\\ &=\frac{1}{S_{\mathrm{C}}}\,[U_{\mathrm{O}'}(y_{\mathrm{C}'}-y_{\mathrm{O}'})+U_{\mathrm{C}}(y_{\mathrm{bc}}-y_{\mathrm{C}'})+U_{\mathrm{O}}(y_{\mathrm{O}}-y_{\mathrm{bc}})]\\ &\quad+\frac{1}{S_{\mathrm{D}}}\,[U_{\mathrm{O}'}(y_{\mathrm{O}'}-y_{\mathrm{D}'})+U_{\mathrm{O}}(y_{\mathrm{da}}-y_{\mathrm{O}})+U_{\mathrm{D}}(y_{\mathrm{D}'}-y_{\mathrm{da}})],\end{aligned} \tag{14}$$

where the previous assumption that the areas S_{C} and S_{D} of grid cells C and D are equal is employed again. The first term on the RHS of equation (14) corresponds to $\mathbf{R}^{\mathrm{S}}_{\mathrm{C}}$, while the second term refers to $\mathbf{R}^{\mathrm{N}}_{\mathrm{D}}$.

Summarizing, the evaluation of the change in time of state variables due to the viscous terms consists of two basic steps: the first step computes the parts of the stress and heat conduction terms which correspond to each cell (equation (14)), while the second step evaluates the changes $\Delta\mathbf{U}_{\mathrm{vis}}$ at the corner nodes of each cell (equation (13)). It is important to note that the viscous scheme operations are decoupled entirely from the operations for the inviscid terms, which makes it simple to switch off viscous term evaluations in inviscid regions (equation adaptation).[1, 3]

The CPU time consumed is 0·007 s per node per iteration on the Vax/750 computer. The scheme requires eight words of memory per node for the state vector **U** and the distribution $\delta\mathbf{U}$. Also, four words per node are required to store the second-order differences of the fourth-order smoothing operator. Additional storage may be used for the time-step and the various metrics.

5. NUMERICAL ISSUES FOR VISCOUS TERMS

The present scheme for the viscous terms is examined with respect to numerical issues that were addressed when developing the algorithm. It should be emphasized that the same issues have to be examined during development of an entire class of finite-volume Navier–Stokes schemes. The analysis of Sections 5 and 6 is valid for a wide range of central-type space-differencing algorithms. An approach of examining the existence of odd–even modes in the solution will be presented. Then the conservation property will be examined. Next the inviscid and diffusion stability limitations will be compared, followed by an investigation of the effect of grid stretching on the accuracy.

5.1. Odd–even solution decoupling

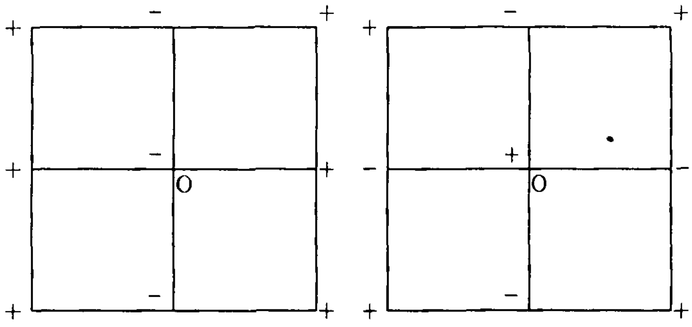
A novel consideration when constructing a numerical scheme is presented in this subsection. The present viscous term scheme was designed so as to prevent the appearance of oscillations in the solution. Central space-differencing schemes are usually susceptible to odd–even decoupling. Frequently the line integrations admit sawtooth modes as valid solutions. Both one- and two-direction odd–even modes may appear as shown in Figure 3, where + and – indicate deviations from a mean.

The appearance of these modes makes convergence slower and in some cases may prevent achievement of a steady state. Usually artificial dissipation is employed in order to suppress the modes. In the present scheme for the viscous terms special care was taken in order to devise a scheme which does not allow sawtooth modes to appear. It turns out that the choice of the *secondary* cells plays a crucial role in the behaviour of the scheme for odd–even decoupling of the solution. The choice of *asymmetric* overlapping *secondary* cells has the effect of suppressing the one- and two-direction odd–even modes shown in Figure 3. In the first case the integration around node O makes a positive contribution to this node, while in the second case a negative contribution is provided. This can be observed by substituting a sawtooth solution of the form $+1, -1$ into the viscous contribution formulae (13). Therefore in both types of sawtooth modes the scheme tends to suppress them.

Instead of a staggered *secondary* cell, another possibility would use averaging in order to obtain the face stresses. For the cd node (Figure 1), averaging such as $R_{\mathrm{cd}}=\frac{1}{2}(R_{\mathrm{C}}+R_{\mathrm{D}})$ can be used. In this case the grid cells C and D are used as *secondary* cells in order to evaluate R_{C} and R_{D}. However, this method allows the two-direction odd–even mode to exist and additional smoothing is required in order to stabilize the scheme. Since the inviscid part of the scheme allows odd–even modes, smoothing is required within the inviscid portions of the flow domain.

5.2. Conservation

The Navier–Stokes equations express conservation of mass, momentum and energy. The flow domain is divided into smaller volumes and the numerical scheme is then applied on each of the cells. For global accuracy of the scheme it is important to conserve the above flow-quantities over each individual cell and therefore over the entire domain. Schemes that accomplish this are said to have the *conservation property*.



(a) one direction mode (b) two directions mode

Figure 3. Odd–even solution modes

A definition of conservation can be derived by considering the viscous part of the equations integrated throughout the domain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\iint_{\text{domain}} \mathbf{U}\mathrm{d}x\mathrm{d}y = \oint_{\text{domain boundary}} \mathbf{R}\mathrm{d}y - \mathbf{S}\mathrm{d}x. \tag{15}$$

The domain is divided into smaller control surfaces, the cells, and the above expression leads to the following general definition of conservation:

$$\sum_{\text{nodes}} \frac{1}{\Delta t}\,\delta\mathbf{U}_{\text{node}}\,A_{\text{node}} = \text{boundary terms.} \tag{16}$$

In the above the A_{node} are suitable areas which surround the grid nodes. For the present scheme the change $\delta\mathbf{U}_{\text{node}}$ at each node is the sum of the contributed changes from the adjoining cells as given by formulae (13). Therefore

$$\sum_{\text{cells}} \frac{1}{\Delta t}\left(\sum_{1}^{4} \delta\mathbf{U}_{\text{corner}}\,A_{\text{cell}}\right) = \text{boundary terms.} \tag{17}$$

It is apparent that the distribution formulae (13) satisfy the above equation, since the sum of the contributions from each cell to its four corners, $\sum_1^4 \delta\mathbf{U}_{\text{corner}}$, is zero, implying that the scheme is conservative. Similarly, the distribution formulae (5) for the inviscid terms show that the scheme for the inviscid terms is also conservative.

The flow through a 40% bump in a channel (Mach number 1.4, Reynolds number 10^4) is employed as a test case in order to investigate conservation of the scheme. An oblique shock forms at the leading edge of the bump. Figure 4 illustrates the mass flow across the channel and shows that there is no mass error between the inlet and the outlet. The jump in the distribution around $X = 0{\cdot}1$ is due to the addition of artificial dissipation in order to capture the oblique shock wave.

5.3. CFL versus diffusion stability restrictions

Using central space and forward time differencing, the stability limitation for the model 1D wave equation $u_t + cu_y = 0$ is $c\Delta t/\Delta y \leq 1$ (CFL limitation), while the corresponding stability restriction for the 1D model diffusion equation $u_t = \nu u_{yy}$ is $\nu\Delta t/(\Delta y)^2 \leq \frac{1}{2}$. Comparing the two time step limitations, we obtain

$$\frac{\Delta t_{\text{vis}}}{\Delta t_{\text{inv}}} = \frac{\frac{1}{2}Re(\Delta y)^2}{M_\infty \Delta y} = \tfrac{1}{2}\,\Delta y Re \frac{1}{M_\infty}, \tag{18}$$

where it is considered that $\nu \sim 1/Re$ and $c \sim 1/M_\infty$. It is apparent that in most common cases the CFL stability restriction is much more severe than the viscous time step limitation. Only in cases of low Re and/or high M_∞, can the viscous limitation be severe.

In the present scheme a combination of the two limitations was employed. Specifically,

$$\Delta t = \omega \min\left(\frac{\Delta m}{|u| + c + \nu/\alpha\Delta m}, \frac{\Delta l}{|v| + c + \nu/\alpha\Delta l}\right), \tag{19}$$

where Δm and Δl are the cell dimensions in the directions m and l respectively, c is the speed of sound, u and v are the velocity components, ν is the kinematic viscosity coefficient and $\alpha = \frac{1}{2}$ is the diffusion stability limitation. Lastly, ω is a safety factor equal to $0{\cdot}9$.

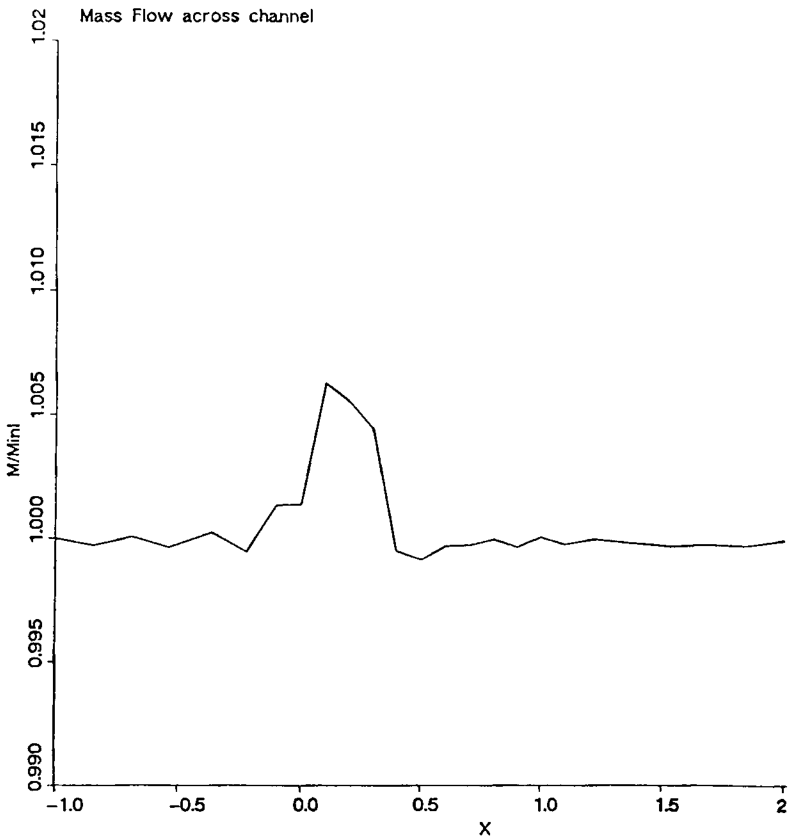


Figure 4. Check of mass conservation—4% bump in a channel ($M_\infty = 1{\cdot}4$, $Re = 10^4$)

5.4. Grid-stretching effect on accuracy

The viscous term scheme is second-order-accurate in space for a uniform Cartesian mesh. Grid stretching and skewness reduce the spatial accuracy to first-order. Generally, the numerical approximation of second-order derivatives of the viscous terms is more susceptible to grid-related errors compared to the evaluation of the inviscid terms.[9] The spatial accuracy of the viscous term discretization therefore merits special attention.

Green's formula provides a derivative value which is averaged over the cell area but does not necessarily apply at the centre of the cell. In fact, the substitution $(\partial u/\partial x)_O = (1/S)\oint u\mathrm{d}y$, which evaluates a derivative at point O of a cell with area S, holds only if O is located at the centroid of the cell. This can be seen from a Taylor series expansion of $\partial u/\partial x(x, y)$ around the point (x_0, y_0) so that

$$\iint_S \frac{\partial U}{\partial x}\mathrm{d}S = \iint_S \frac{\partial u}{\partial x}(x_0, y_0)\mathrm{d}S + \iint_S (x - x_0)\frac{\partial^2 u}{\partial x^2}(x_0, y_0)\mathrm{d}S + \iint_S (y - y_0)\frac{\partial^2 u}{\partial y^2}(x_0, y_0)\mathrm{d}S + \dots .$$

Therefore

$$\frac{\partial u}{\partial x}(x_0, y_0)=\frac{1}{S}\iint_S \frac{\mathrm{d}U}{\partial x}\,\mathrm{d}S$$

holds exactly only if

$$\iint_S (x-x_0)^m(y-y_0)^n \mathrm{d}S=0 \tag{20}$$

for $m, n=0, 1, \ldots$, which means that (x_0, y_0) must lie at the centroid of the cell. This is not the case if grid stretching is present.

Consider the one-dimensional case which evaluates a quantity u at point A ($\overline{u_A}$ say) using the values of quantities at neighbouring points A^+ and A^-:

$$\overline{u_A}=\frac{1}{\Delta S}\int_{A^-}^{A^+} u\mathrm{d}S=\frac{1}{\Delta S}\tfrac{1}{2}(u_{A^-}+u_{A^+})\Delta S. \tag{21}$$

Taylor expanding u_{A^-} and u_{A^+} and substituting in (21) gives

$$\overline{u_A}=u_A+\tfrac{1}{2}(\Delta S^+-\Delta S^-)\frac{\partial u}{\partial s}+\tfrac{1}{4}[(\Delta S^+)^2+(\Delta S^-)^2]\frac{\partial^2 u}{\partial s^2}+\ldots$$

$$\Rightarrow \overline{u_A}=u_A+\tfrac{1}{2}(\Delta S^+-\Delta S^-)\frac{\partial u}{\partial s}+\ldots, \tag{22}$$

where ΔS^+ and ΔS^- are the distances between points A and A^+ and points A^- and A respectively. The derivative terms on the right-hand side represent the error that results when approximating $\overline{u_A}$. Without stretching ($\Delta S^+=\Delta S^-$), an evaluation of $\overline{u_A}$ is second-order-accurate, but reduces to first-order when stretching is present.

In order to restrict all scheme operations within each cell, it is necessary to assume that the four cells which surround each node (cells A, B, C and D in Figure 1) have equal metrics. However, the error introduced by grid stretching is of the same order as that introduced by applying Green's theorem. Therefore these assumptions do not deteriorate the spatial accuracy of the scheme any further. In practice it has been found that stretching factors less than 30% are acceptable and do not introduce appreciable inaccuracies.

6. ARTIFICIAL DISSIPATION

Smoothing accomplished by explicitly adding dissipation is employed by a large number of existing schemes, especially those concerned with compressible flows. There are two main types of such artificial smoothing: one is used to capture shocks, the second is designed to damp spurious oscillations throughout the field and to suppress odd–even decoupling of the solutions.[4, 14] The smoothing operators are cast in a form suitable for unstructured grids. All operations are split in such a way that no information is required from outside each cell. These second- and fourth-order smoothing operators are examined in detail in order to address the relevant issues of viscous region contamination by artificial smoothing and the degradation of order with grid stretching.

6.1. Unstructured grid formulation

Second-order smoothing provides the damping necessary to smear a shock, which ideally has zero thickness, in such a way that oscillations are avoided. Since such smoothing is required only

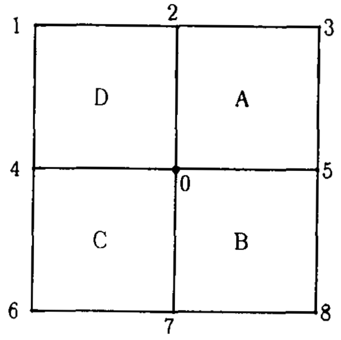


Figure 5. Node and cell designations for smoothing operators

in shock regions, a switch is employed to turn it off elsewhere. Consider the specific discretization for a cell vertex scheme at node 0 (Figure 5).

The node receives contributions from each of the four surrounding cells: that from cell A is

$$S^{(2)}_{0A} = +\left|\frac{(P_3+P_5)-(P_2+P_0)}{P_3+P_5+P_2+P_0}\right| [(u_3+u_5)-(u_2+u_0)]$$
$$+\left|\frac{(P_2+P_3)-(P_0+P_5)}{P_2+P_3+P_0+P_5}\right| [(u_2+u_3)-(u_0+u_5)]; \tag{23}$$

similarly, from cell D it is

$$S^{(2)}_{0D} = +\left|\frac{(P_1+P_4)-(P_2+P_0)}{P_1+P_4+P_2+P_0}\right| [(u_1+u_4)-(u_2+u_0)]$$
$$+\left|\frac{(P_1+P_2)-(P_4+P_0)}{P_1+P_2+P_4+P_0}\right| [(u_1+u_2)-(u_4+u_0)]. \tag{24}$$

Similar expressions furnish contributions from cells C and B. The above is first-difference, but after the node receives contributions from surrounding cells it becomes second-difference. Pressure differences in the switch are normalized by the sum of the pressures at the four corners of each cell. The sum of the smoothing distributions to the four nodes of each cell is zero, which implies that the above operator is conservative.

The fourth-order smoothing that is used away from shocks to suppress odd–even modes and damp spurious oscillations is turned off near shocks because it is destabilizing. The operator is formed in two steps. The second-order difference operator is formed in the first step (Figure 5):

$$D^2_{0A} = u_0 + u_2 + u_3 + u_5 - 4u_0,$$
$$D^2_{0B} = u_7 + u_0 + u_5 + u_8 - 4u_0,$$
$$D^2_{0C} = u_6 + u_4 + u_0 + u_7 - 4u_0,$$
$$D^2_{0D} = u_4 + u_1 + u_2 + u_0 - 4u_0. \tag{25}$$

The second step duplicates the first, replacing state variables by second-order differences from the first step:

$$
\begin{aligned}
-D^4_{0\mathrm{A}} &= D^2_0 + D^2_2 + D^2_3 + D^2_5 - 4D^2_0,\\
-D^4_{0\mathrm{B}} &= D^2_7 + D^2_0 + D^2_5 + D^2_8 - 4D^2_0,\\
-D^4_{0\mathrm{C}} &= D^2_6 + D^2_4 + D^2_0 + D^2_7 - 4D^2_0,\\
-D^4_{0\mathrm{D}} &= D^2_4 + D^2_1 + D^2_2 + D^2_0 - 4D^2_0.
\end{aligned}
\tag{26}
$$

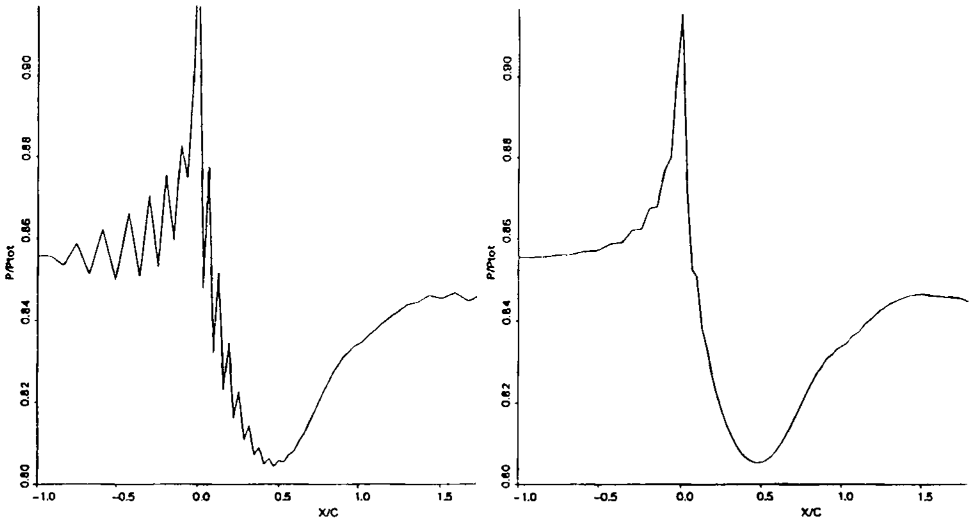
6.2. Smoothing coefficients

The combined second- and fourth-order smoothing operator has the form

$$\delta u_i = \tfrac{1}{4}\,\sigma_2 S_i^2 - \tfrac{1}{4}\max(0, \sigma_4 - \sigma_2 \Delta P) D_i^4, \tag{27}$$

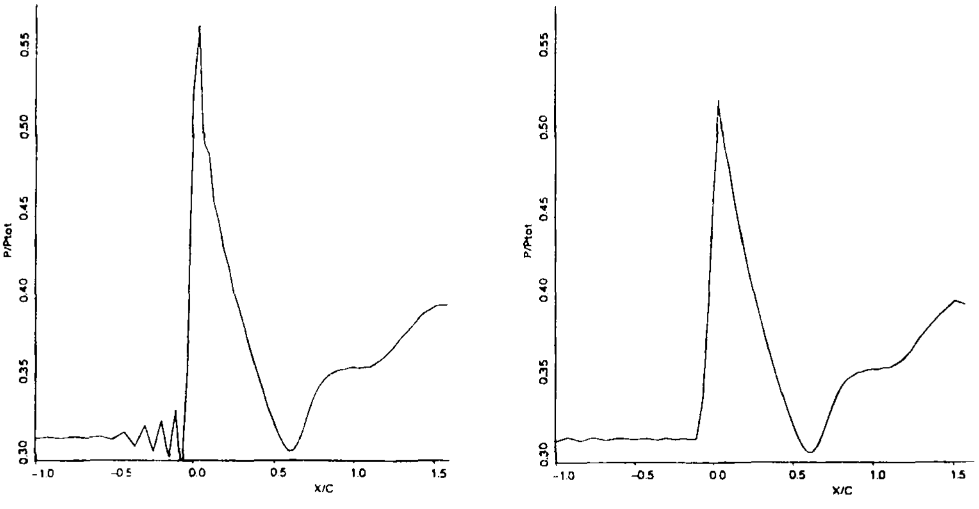
where S_i^2 and D_i^4 are the second- and fourth-order operators discussed above, σ_2 and σ_4 are the corresponding smoothing coefficients and ΔP is the pressure switch. Near shocks the term $\sigma_2 \Delta P$ dominates σ_4 and therefore $\max(0, \sigma_4 - \sigma_2 \Delta P)$ vanishes and switches off the fourth-order smoothing.

Numerical experiments have been carried out to determine optimum values for the smoothing coefficients. The example considers flow at Mach number 0·5 and 1·4 and with Reynolds number 8×10^3 in a channel with a bump and a 65×33 grid. The bump is located between $X = 0$ and 1. Figure 6 shows C_{P} distributions for the subsonic case with and without fourth-order smoothing only and it is clear that even a small amount of dissipation suppresses the sawtooth mode. Similarly, a supersonic case ($M_\infty = 1{\cdot}4$) was used to study the second-order smoothing coefficient. Figure 7 shows C_{P} wall distributions with a fourth-order smoothing coefficient $\sigma_4 = 0{\cdot}0005$ and



(a) without smoothing (b) with smoothing $\sigma_4 = 0.0005$

Figure 6. Effect of fourth-order smoothing on odd–even decoupling (surface pressure distribution)



(a) smoothing coeff. $\sigma_2 = 0.05$ (b) smoothing coeff. $\sigma_2 = 0.40$

Figure 7. Shock capturing with second-order smoothing (surface pressure distribution)

shock-smoothing coefficients σ_2 of 0·05 and 0·40. Clearly the value of 0·40 smears the shock excessively, while the value of 0·05 is too small and preshock oscillations appear. A reasonable choice of smoothing coefficients is $\sigma_2 = 0{\cdot}20$ and $\sigma_4 = 0{\cdot}0004$. Although the values were determined for relatively low Reynolds number, they have been found to be valid for high-Reynolds-number computations as well. This is due to the fact that smoothing scales with mesh size, which is normally smaller for higher Reynolds numbers.

6.3. *Boundary layer contamination*

Smoothing is still required in the inviscid regions since the inviscid term discretization scheme allows odd–even modes to exist. The presence of smoothing in viscous regions may seriously deteriorate the accuracy. Figure 8 demonstrates contamination of the boundary layer by smoothing via wall skin friction coefficient distributions for values of the smoothing coefficient σ_4 equal to 0·004 and zero. Both the way in which smoothing terms affect the viscous layer solution as well as the resolution requirements to avoid the error are of interest.

The second- and fourth-order derivatives usually take large values within the viscous region and therefore smoothing contaminates the viscous layer more than other regions. For the model diffusion equation $u_t = \nu u_{yy}$ we have

$$u_t = \nu u_{yy} + \frac{|u| + c}{\mathrm{CFL}} \sigma_2 \Delta P \Delta y u_{yy} + \frac{|u| + c}{\mathrm{CFL}} \nu_4 (\Delta y)^3 u_{yyyy}, \tag{28}$$

where the last two terms represent first- and third-order errors introduced by the two smoothing operators. The CFL stability condition $((|u| + c)\Delta t/\Delta x = \mathrm{CFL})$ was used in order to eliminate the time step Δt from the above expression. To ensure that the artificial viscosity is much less than the

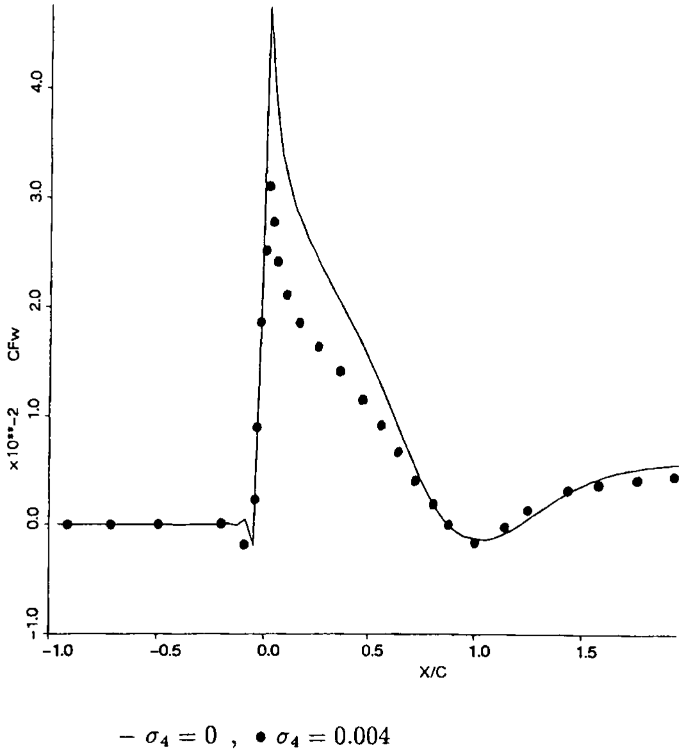


Figure 8. Contamination of boundary layer solution by smoothing (skin friction coefficient distribution)

physical viscosity,

$$\frac{|u|+c}{\mathrm{CFL}}\sigma_2 \Delta P \Delta y \ll v,$$

which yields

$$\sigma_2 \ll \frac{\mathrm{CFL}}{Re \Delta P \Delta y}, \tag{29}$$

in which the Reynolds number is specially defined as $Re \equiv (|u|+c)L/v$. If ΔP is negligible across the shear layer, then relatively large values of σ_2 can be used. However, ΔP can be large near features such as shocks and expansions. For example, for the case $\mathrm{CFL}=1$, $Re=10^6$, $\Delta P = 0{\cdot}1$ and $\Delta y = 10^{-4}$ we obtained $\sigma_2 \ll 0{\cdot}1$.

For a fixed value of the smoothing coefficient σ_2 the resolution of the shear layer should be such that

$$O\left(\frac{v u_{yy}}{[(|u|+c)/\mathrm{CFL}]\sigma_2 \Delta P \Delta y u_{yy}}\right) \gg 1 \Rightarrow N \equiv \frac{\delta}{\Delta y} \gg \frac{Re_\delta \sigma_2 \Delta P}{\mathrm{CFL}}, \tag{30}$$

N being the number of grid points across the shear layer and $Re_\delta = (|u|+c)\delta/v$. For the case $\mathrm{CFL}=1$, $Re_\delta = 10^3$, $\Delta P = 0{\cdot}1$ and $\sigma_2 = 0{\cdot}1$ the number of points within the layer should be $N \gg 10$, so that the real viscous term dominates. Re_δ is based on $|u|+c$ and therefore in nearly imcompressible cases it can take very large values, which implies that considerable resolution is needed.

For a fixed value of the smoothing coefficient ν_4 the resolution of the shear layer should be such that

$$O\left(\frac{\nu u_{yy}}{[(|u|+c)/\mathrm{CFL}]\,\nu_4(\Delta y)^3 u_{yyyy}}\right)\gg 1\Rightarrow N\gg\left(\frac{Re_\delta \nu_4}{\mathrm{CFL}}\right)^{1/3}. \tag{31}$$

Comparison with relation (30) shows that this resolution requirement is much less severe as a result of the higher order of this operator. For example, if CFL $=1$, $Re_\delta=10^3$ and $\nu_4=0{\cdot}001$, the number of points within the layer is $N\gg 1$.

6.4. Grid stretching increases smoothing error

The above smoothing operators introduce first- and third-order errors only for uniformly spaced grids. The actual order can be demonstrated by considering a one-dimensional stretched grid. A second-order smoothing operator without the pressure switch has the form $S_0^2=u_1-2u_0+u_{-1}$, with 1, 0 and -1 being consecutive grid points.

A Taylor series expansion about point 0 leads to $S_0^2=(h^+-h^-)u_x+\frac{1}{2}[(h^+)^2+(h^-)^2]u_{xx}+\ldots$, where h^+ and h^- are the distances between points 0 and 1 and points -1 and 0 respectively. Assuming geometric stretching and with $\alpha\equiv h^+/h^-$, the operator becomes

$$S_0^2=h^-(\alpha-1)u_x+\tfrac{1}{2}(h^-)^2(\alpha^2+1)u_{xx}+\ldots. \tag{32}$$

The first-order term $h^-(\alpha-1)u_x$ appears, which increases the smoothing error and makes the operator *dispersive* rather than *dissipative*. Similarly, the fourth-order difference operator $D_0^4=u_{-2}-4u_{-1}+6u_0-4u_1+u_2$ has the following form for a geometrically stretched mesh:

$$D_0^4=(\alpha-2)(\alpha-1)h^-u_x+2\alpha(1+\alpha/4)(\alpha^2+1)(h^-)^2u_{xx}+\ldots. \tag{33}$$

Again the first-order term $(\alpha-2)(\alpha-1)h^-u_x$ increases the error and makes the smoothing *dispersive*. It is to be noted that geometrically stretched meshes are widely employed for viscous computations. The dissipative terms in equations (32) and (33) are positive for any α, which precludes the possibility of having negative damping for some α.

7. SOLVER VALIDATION

The accuracy of the Navier–Stokes numerical scheme has been investigated for both laminar and turbulent cases. Channel as well as aerofoil geometries were considered. An adaptive algorithm that uses both grid and equation adaptation was coupled to the solver for the aerofoil cases. An initial quadrilateral grid is automatically embedded in local regions of the domain in which important flow features exist. The embedding is achieved through subdivision of grid cells. Also, the scheme switches to the simpler inviscid flow (Euler) equations outside the viscous regions. Comparisons were made with analytical, numerical and experimental results. The cases include a Blasius boundary layer and a 5% bump in a channel. Also, high-Reynolds-number aerofoil cases in both subsonic and transonic flow were employed. Additional cases including two-element aerofoils are presented in Reference 1. The reported CPU times refer to an Alliant FX/8 computer with three processors of approximately 1·5 Mflops speed.

The boundary conditions include wall and inflow/outflow conditions. At a wall there is no slip ($u=v=0$), the temperature is either prescribed or its gradient is set to zero (adiabatic wall) and the normal-to-wall pressure gradient is set to zero. At inflow and outflow boundaries the 1D unsteady inviscid theory of characteristics dictates the number of variables that will be either specified or extrapolated. At an inflow boundary in subsonic flow, total pressure, total temperature and flow angle (v/u) are specified, while the Riemann invariant $J^{-} \equiv u-2c/(\gamma-1)$ is extrapolated from the interior of the domain. At an outflow boundary in subsonic flow, density and u- and v-velocities are extrapolated, while pressure is prescribed.

A Blasius boundary layer is computed with $M_{\infty}=0{\cdot}1$ and $Re=10^3$. A Blasius profile $(X=1)$ was specified at the inlet in order to avoid the leading edge singularity. A grid of 33×41 nodes was employed with 12 points within the boundary layer at the middle of the flat plate $(X=1{\cdot}50)$. The smallest grid normal spacing at the wall was 0·004 and a constant grid stretching factor of 1·1 was employed in the normal-to-surface direction. No artificial dissipation was added. Figure 9 compares the computed and corresponding analytical u-velocity profiles at position $X=1{\cdot}50$.

The second test was a 5% bump in a channel in a subsonic flow at $M_{\infty}=0{\cdot}5$ and $Re=8\times10^3$. This is the same case that was employed in Section 6.2 for determination of the fourth-order smoothing coefficient. A 65×33 grid with minimum wall normal spacing equal to 2.5×10^{-3} is used. The skin friction (C_f) curve for this example is compared in Figure 10 with the results of References 6 and 15. Employment of the multiple-grid accelerator of Reference 5 reduced the required CPU time to convergence by a factor of five.

7.1. *Adaptive grid aerofoil cases*

The subsonic and transonic flow for an NACA 0012 aerofoil was considered and comparisons were made with experimental results obtained by an AGARD group.[16] The Baldwin–Lomax[17] algebraic model was employed for the turbulent flow computations.

The subsonic flow conditions were $M_{\infty}=0{\cdot}50$, $Re=2{\cdot}91\times10^6$ and $\alpha=1{\cdot}77°$, where both the angle of attack and Mach number values are those suggested in Reference 16 to take into account wind tunnel wall effects.

An initial C-mesh of 33×17 points was employed with two levels of refinement, resulting in a final number of 5225 cells within the domain. The minimum grid normal spacing at the aerofoil leading edge was 9×10^{-5} chord lengths, while that for the trailing edge region was 9×10^{-4}. The

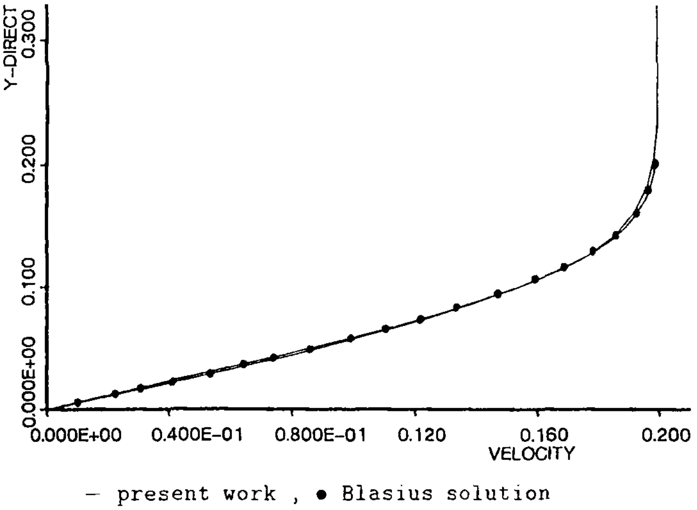


Figure 9. Comparison of velocity profiles for a Blasius boundary layer

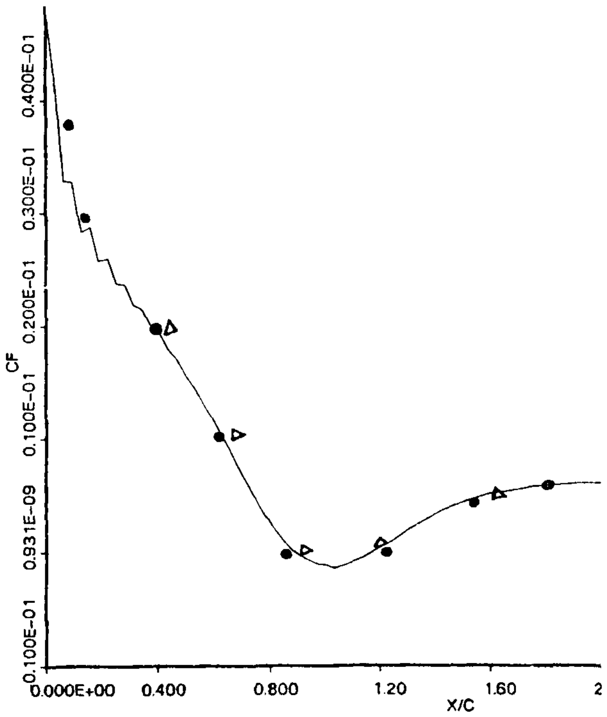


Figure 10. Comparison of skin friction distribution with previous numerical results—5% bump in a channel ($M_\infty = 0{\cdot}5$, $Re = 8 \times 10^3$): —, present work; •, Reference 6; △, Reference 15

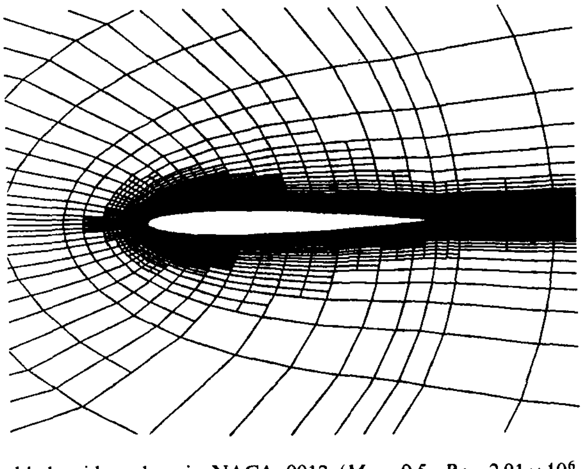


Figure 11. Two-level embedded grid—subsonic NACA 0012 ($M_\infty = 0{\cdot}5$, $Re = 2{\cdot}91 \times 10^6$, $\alpha = 1{\cdot}77°$); horizontal scale enlarged

spacing in the streamwise direction in the leading and trailing edge regions was 0·002 and 0·026 respectively. Figure 11 illustrates the embedded grid. The case took 4000 iterations to converge (reduction in residual magnitude by three orders) and required 1·8 h. The resulting flow field is depicted in Figure 12 in terms of Mach number contour plots. The experiment[16] provided

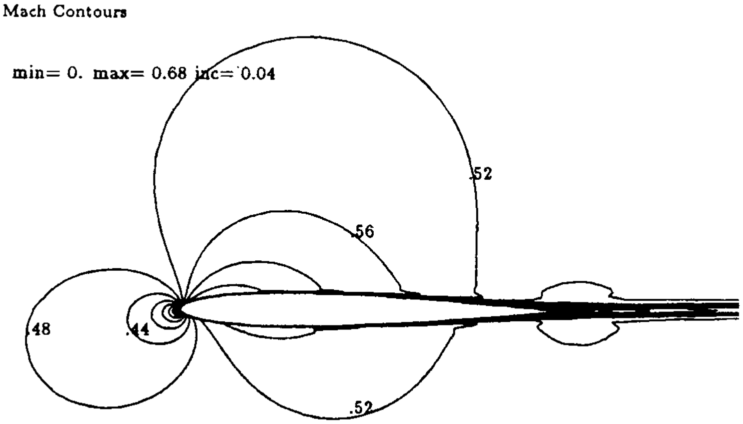


Figure 12. Flow field around subsonic NACA 0012 ($M_\infty = 0{\cdot}5$, $Re = 2{\cdot}91 \times 10^6$, $\alpha = 1{\cdot}77°$); horizontal scale enlarged

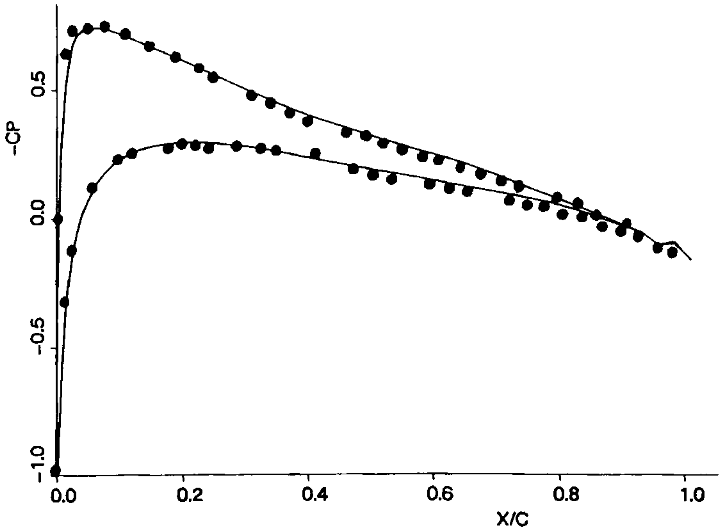


Figure 13. Comparison of pressure coefficient distribution with experiment—subsonic NACA 0012 ($M_\infty = 0{\cdot}5$, $Re = 2{\cdot}91 \times 10^6$, $\alpha = 1{\cdot}77°$): —, present work; •, experiment[16]

pressure distribution data (Figure 13); comparison shows very good agreement between numerics and measurements. The computed C_L of 0·192 compares very well with the experimental value of 0·195.

The transonic flow conditions were $M_\infty = 0{\cdot}754$, $Re = 3{\cdot}76 \times 10^6$ and $\alpha = 3{\cdot}02°$. Again the angle of attack and Mach number values are those suggested in Reference 16 to account for wind tunnel wall effects. An initial C-mesh of 65 × 41 points was applied with the far-field boundary placed 15 chord lengths away from the aerofoil. Three levels of embedding were introduced by the algorithm; the final grid is illustrated in Figure 14 and the final number of cells within the domain is 40440. The minimum grid normal spacing at the aerofoil leading edge is 2×10^{-5} chord

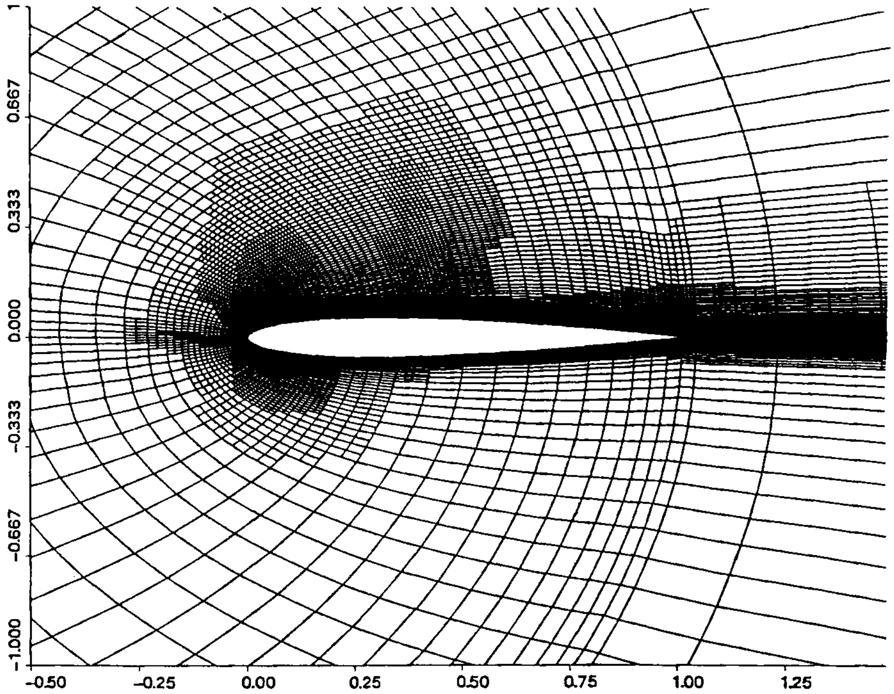


Figure 14. Final three-level embedded grid—transonic NACA 0012 ($M_\infty = 0{\cdot}754$, $Re = 3{\cdot}76 \times 10^6$, $\alpha = 3{\cdot}02°$)

lengths, while the spacing at the trailing edge region is 2×10^{-4}. The spacing in the streamwise direction at the leading edge region is 3×10^{-4}, while the corresponding spacing at the trailing edge is 0·004. The case took 5000 iterations to converge and consumed 8·5 h of computing time.

Figure 15 illustrates the flow field in terms of Mach number contours. A shock forms on the suction side at 40% of the chord, with the Mach number just upstream of the shock being 1·31. The boundary layer on the suction side of the aerofoil starts to thicken upstream of the shock and separates at $X = 0{\cdot}82$ close to the trailing edge. On the other hand, the pressure-side boundary layer is considerably thinner and remains attached to the surface. The wiggles that appear just upstream of the shock are a result of odd–even modes. They exist owing to the use of low values of artificial viscosity so that the solution within the viscous region would not be contaminated. Such oscillations do not affect the shock location, which is predicted accurately.

Interesting flow physics is revealed in an enlarged view of the shock–boundary layer interaction region (Figure 16). The severe adverse pressure gradient induced by the normal shock causes the boundary layer to thicken considerably and eventually to separate at the foot of the normal shock. A separation bubble is formed and it is captured in detail by the adaptive algorithm. The boundary layer separates at $X = 0{\cdot}36$ and reattaches at $X = 0{\cdot}52$ chord lengths. It should be noted that the appearance of such separation bubbles is dependent on the turbulence model being employed.

The procedure accuracy may be examined by comparing the experimental surface pressure coefficient distribution with the corresponding numerical result (Figure 17). The shock location is predicted reasonably accurately but is somewhat smeared. A fourth level of embedding which would provide a 'crisper' shock, was disallowed owing to computer limitations. The agreement

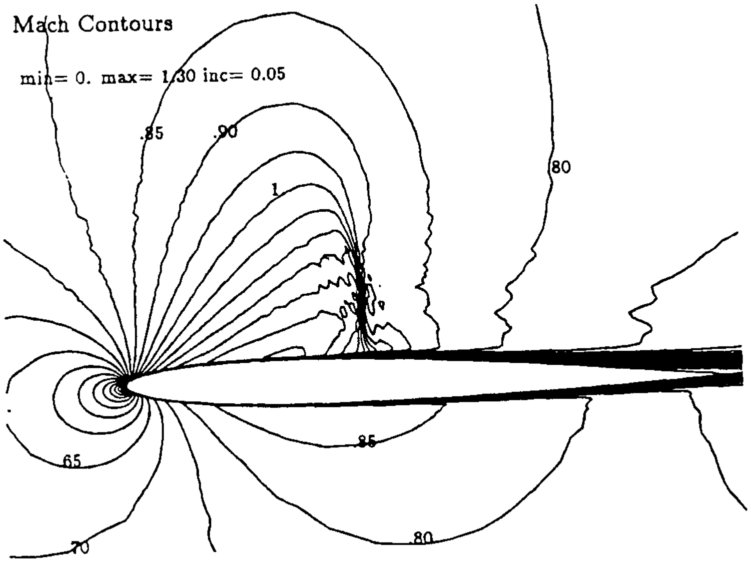


Figure 15. Flow field around transonic NACA 0012 ($M_\infty=0{\cdot}754$, $Re=3{\cdot}76\times10^6$, $\alpha=3{\cdot}02°$)

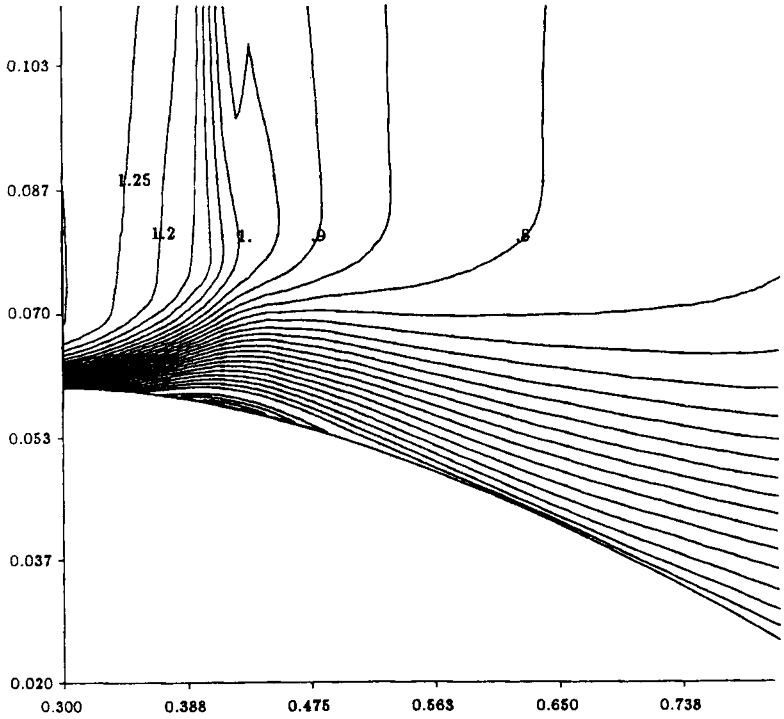


Figure 16. Separation bubble at foot of shock—transonic NACA 0012 ($M_\infty=0{\cdot}754$, $Re=3{\cdot}76\times10^6$, $\alpha=3{\cdot}02°$)

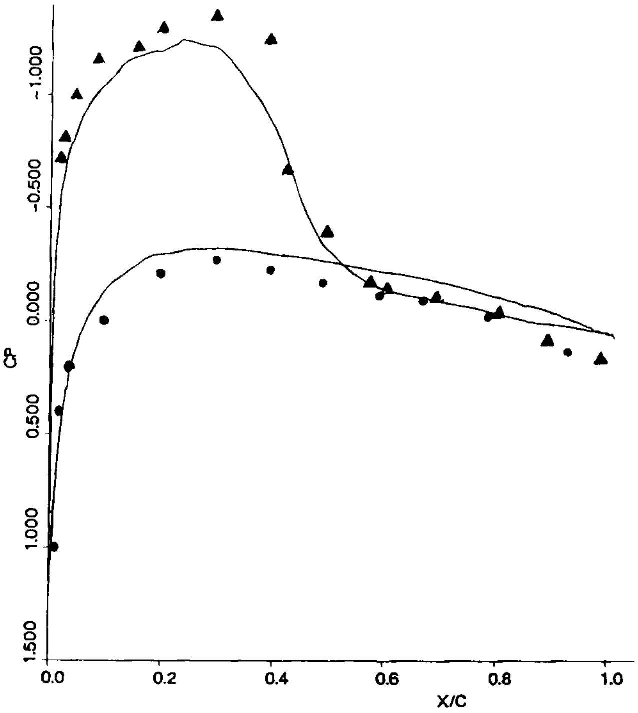


Figure 17. Comparison of pressure coefficient distribution with experiment—transonic NACA 0012 ($M_\infty = 0{\cdot}754$, $Re = 3.76 \times 10^6$, $\alpha = 3{\cdot}02°$): —, embedded mesh; •, pressure side;[15] ▲, suction side[16]

remains good downstream of the shock. However, as the trailing edge is approached, the boundary layer does not resist the adverse pressure gradient and separates, causing the pressure distribution to tend to level out. Such trailing edge separation is not observed in the experiment. The algebraic turbulence model employed is believed to be largely responsible fot this behaviour, as has been concluded by comparative studies of different turbulence models for transonic aerofoils.[18] The pressures on the pressure and suction sides match at the trailing edge and the somewhat lower pressure level at the suction side influences the pressure-side distribution, causing it to deviate slightly from the experimental results. The deviation is approximately the same over most of the pressure surface. Unfortunately, corresponding measurements for skin friction were not performed.

8. CONCLUDING REMARKS

A compact, finite volume, explicit scheme for the two-dimensional Navier–Stokes equations of viscous fluid flow was developed. The scheme is suitable for adaptive algorithms, which employ locally embedded quadrilateral meshes, as well as equation adaptation. Extension of a Euler solver to include viscous terms requires consideration of numerical problems related to treatment of the viscous terms. These issues are addressed in order to design an accurate Navier–Stokes solver. The issues were investigated with respect to the specific scheme presented. However, the approach of the investigation as well as the conclusions are valid for an entire class of finite volume algorithms.

The inviscid term scheme is a one-step Lax–Wendroff-type scheme. Treatment of the viscous terms required employment of two staggered cells, one for evaluation of first-order derivatives and the other for second-order derivatives. Scheme operations were split in such a way that no additional information is required from outside each cell when integrating that cell, which is an important property for an unstructured grid solver. No assumptions were made that would prohibit extension of the present 2D scheme to three dimensions. The scheme is conservative. Grid stretching introduces a severe accuracy error to evaluation of the second-order derivatives of the viscous terms.

The present viscous term scheme with the staggered cell arrangement suppresses both the one- and two-direction sawtooth modes. However, artificial dissipation is still required for capturing shocks and for suppresion of oscillations in the inviscid region of the domain. Optimum values of the smoothing coefficients can be obtained by employing test cases. The solution in the viscous regions is very sensitive to smoothing. The magnitude of the artificial dissipation terms must be much smaller than the magnitude of the real viscous terms, which gives expressions for the smoothing coefficients as well as for the required number of grid points across a boundary layer in order to avoid boundary layer contamination. Also, grid stretching increases the smoothing error and makes the operator dispersive instead of dissipative. Finally, the test cases employed demonstrate both the accuracy and robustness of the algorithm. These cases include both subsonic and transonic flows with channel and aerofoil geometries. The Navier–Stokes scheme was coupled with an adaptive algorithm for the high-Reynolds-number aerofoil cases.

ACKNOWLEDGEMENT

The author would like to thank Professor J. R. Baron of the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology for his helpful discussions and comments.

REFERENCES

1. Y. Kallinderis, 'Adaptation methods for viscous flows', *Ph.D. Thesis, CFDL-TR-89-5*, Department of Aeronautics and Astronautics, MIT, 1989.
2. J. Kallinderis and J. R. Baron, 'Unsteady and turbulent flow using adaptation methods', in *Lecture Notes in Physics, Vol. 323*, Springer, Berlin, 1988, pp. 326–330.
3. Y. Kallinderis and J. R. Baron, 'Adaptation methods for a new Navier–Stokes algorithm', *AIAA J.*, **27**, 37–43 (1989).
4. A. Jameson, W. Schmidt and E. Turkel, 'Numerical solutions of the Euler equations by finite-volume methods using Runge–Kutta time-stepping schemes', *AIAA Paper 81-1259*, 1981.
5. R.-H. Ni, 'A multiple grid scheme for solving the Euler equations', *AIAA J.*, **20**, 1565–1571 (1982).
6. R. L. Davis, R.-H. Ni and J. E. Carter, 'Cascade viscous flow analysis using the Navier–Stokes equations', *AIAA Paper 86-0033*, 1986.
7. R. C. Swanson and E. Turkel, 'A multistage time-stepping scheme for the Navier–Stokes equations', *AIAA Paper 85-0035*, 1985.
8. L. Martinelli, A. Jameson and F. Grasso, 'A multigrid method for the Navier–Stokes equations', *AIAA Paper 86-0208*, 1986.
9. J. Kallinderis and J. R. Baron, 'The finite volume approach for the Navier–Stokes equations', in C. A. Brebbia (ed.), *Computational Methods in Viscous Aerodynamics*, Elsevier, Amsterdam, 1990, pp. 117–146.
10. P. D. Lax and B. Wendroff, 'Systems of conservation laws', *Commun. Pure Appl. Math.*, **13**, 217–237 (1960).
11. J. F. Dannenhoffer, 'Grid adaptation methods for complex two-dimensional transonic flows', *Sc.D. Thesis, CFDL-TR-87-10*, Department of Aeronautics and Astronautics, MIT, 1987.
12. L. Cambier, V. Couaillier and J. P. Veuillot, 'Numerical solution of the Navier–Stokes equations', *Rech. Aerosp., No. 1988-2*, 1988.
13. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*, Springer, New York, 1983.
14. T. H. Pulliam, 'Artificial dissipation models for the Euler equations', *AIAA Paper 85-0438*, 1985.
15. R. V. Chima and G. M. Johnson, 'Efficient solution of the Euler and Navier–Stokes Equations with a vectorized multiple-grid algorithm', *AIAA J.*, **23**, 23–32 (1985).

16. J. J. Thibert, M. Granjacques and L. H. Ohman, 'NACA 0012 airfoil', *AGARD AR 138*, 1979.

17. B. S. Baldwin and H. Lomax, 'Thin-layer approximation and algebraic model for separated flows', *AIAA Paper 78-257*, 1978.

18. T. L. Holst, 'Viscous Transonic Airfoil Workshop—compendium of results', *AIAA Paper 87-1460*, 1987.