INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 20, 671–693 (1995)

A SECOND ORDER FINITE ELEMENT METHOD FOR THE SOLUTION OF THE TRANSONIC EULER AND NAVIER–STOKES EQUATIONS

G. S. BARUZZI, W. G. HABASHI AND J. G. GUEVREMONT

Computational Fluid Dynamics Laboratory, Concordia University and CERCA-Centre for Research on Computation and its Applications, Montreal, Quebec, Canada

M. M. HAFEZ

Department of Mechanical and Aerospace Engineering, University of California, Davis, CA, U.S.A.

SUMMARY

The numerical solution of the compressible Euler and Navier–Stokes equations in primitive variables form requires the use of artificial viscosity or upwinding. Methods that are first-order-accurate are too dissipative and reduce the effective Reynolds number substantially unless a very fine grid is used. A first-order finite element method for the solution of the Euler and Navier–Stokes equations can be constructed by adding Laplacians of the primitive variables to the governing equations. Second-order schemes may require a fourth-order dissipation and higher-order elements. A finite element approach is proposed in which the fourth-order dissipation is recast as the difference of two Laplacian operators, allowing the use of bilinear elements. The Laplacians of the primitive variables of the first-order scheme are thus balanced by additional terms obtained from the governing equations themselves, tensor identities or other forms of nodal averaging. To demonstrate formally the accuracy of this scheme, an exact solution is introduced which satisfies the continuity equation identically and the momentum equations through forcing functions. The solutions of several transonic and supersonic inviscid and laminar viscous test cases are also presented and compared to other available numerical data.

KEY WORDS: airfoil; artificial viscosity; upwinding

1. INTRODUCTION

For inviscid flows, schemes for the numerical solution of the Euler equations in primitive variables form require the use of artificial viscosity or of upwinding to eliminate odd–even decoupling and for numerical stability. On the other hand, for the Navier–Stokes equations at high Reynolds numbers, the viscous terms are dominant only in a thin layer outside which the flow is nearly inviscid. The artificial viscosity that is needed for numerical stability in this outer region must be minimized throughout, or eliminated altogether in the viscous layer, in order not to contaminate the numerical solution with artificial dissipation.

In first-order methods, the amount of artificial viscosity necessary for stable solutions is proportional to the mesh size and its detrimental effects are reduced by mesh refinement. It is, however, impractical to use a fine mesh throughout the solution domain, and a cost-effective grid should reflect the disparate characteristic lengths of the viscous and inviscid regions. A practical alternative to fine meshes would be to adopt a higher-order artificial viscosity.

CCC 0271–2091/95/080671–23

A continuous effort has been devoted to achieve such minimum artificial viscosity. For finite difference and finite volume this is accomplished either by using a combination of second- and fourth-order dissipation [1,2] or by using higher-order upwinding.[3,4] For finite elements, Hughes *et al.*[5,6] developed the Petrov–Galerkin and streamline upwinding approaches, and Morgan *et al.*[7] adopted a flux-corrected transport strategy to reduce the effects of the artificial viscosity and improve the quality of the solution.

Baruzzi *et al.*[8,9] have proposed a simple first-order artificial viscosity in the form of Laplacians of the pressure and velocity components, added to the continuity and momentum equations, respectively. To extend this approach to higher order, however, a fourth-order operator would be required, calling for high-order elements. To avoid this, an alternative would be to recast the fourth-order artificial viscosity as the difference of two second-order operators. Thus, it is proposed in this paper to balance the Laplacians of the first-order scheme with correction terms, effectively yielding a fourth-order dissipation. Such a scheme has been tested in the context of viscous incompressible flows,[10] transonic inviscid flows[11] and transonic viscous flows.[12,13] The present paper formalizes this second-order approach for inviscid and viscous transonic flows in a finite element framework.

2. A FIRST-ORDER ARTIFICIAL VISCOSITY SCHEME

The governing equations of viscous, steady compressible flow can be written as

$$\nabla\cdot(\rho\mathbf{V}) = 0 \tag{1a}$$

$$\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla p - \nabla\cdot\boldsymbol{\tau} = 0 \tag{1b}$$

$$H_\infty = \frac{\gamma}{\gamma-1}\frac{p}{\rho} + \frac{\mathbf{V}\cdot\mathbf{V}}{2} \tag{1c}$$

$$\frac{\mu}{\mu_\infty} = \left(\frac{T_\infty + 110°k}{T + 110°k}\right)\left(\frac{T}{T_\infty}\right)^{3/2} \tag{1d}$$

$$\nabla\cdot\boldsymbol{\tau} = \frac{1}{Re}[-\tfrac{2}{3}\nabla(\mu\nabla\cdot\mathbf{V}) + \nabla\times\mu(\nabla\times\mathbf{V}) + 2(\nabla\cdot\mu\nabla)\mathbf{V}]$$

Equation (1c) is the definition of constant total enthalpy, a good approximation for steady flow without heat transfer. Equation (1d) is Sutherland's law for air. After replacing the density in equations (1a) and (1b) by (1c), and lagging the viscosity coefficient during the iterative solution, the system, in two dimensions, reduces to three coupled equations with variables $\{u, v, p\}$.

A pressure dissipation term is introduced in the continuity equation to permit the use of equal order interpolation polynomials for pressure and velocity:

$$\nabla\cdot(\rho\mathbf{V}) - \varepsilon_1\nabla^2 p = 0 \tag{2a}$$

To stabilize the numerical solution of the equations an artificial dissipation (viscosity), proportional to the Laplacian of the velocity, can be introduced in the momentum equation as follows:

$$\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla p - \nabla\cdot\boldsymbol{\tau} - \varepsilon_1\nabla^2\mathbf{V} = 0 \tag{2b}$$

This first-order artificial viscosity scheme successfully met the Pulliam challenge [14] for inviscid non-lifting flows over circular cylinders and ellipses, a set of simple test cases whose accurate numerical solutions, to our knowledge, have yet to be obtained by other methods. The same approach has also been extended to viscous flows.[15,16]

Even though good results have been obtained based on equations (2), the penalty is that conservation of mass and momentum can only be achieved within an error proportional to the artificial dissipation terms. These effects must therefore be minimized.

3. A SECOND-ORDER ARTIFICIAL VISCOSITY SCHEME

The present work addresses the reduction of the artificial dissipation in order to improve the quality of the overall solution to second-order accuracy. Global second-order accuracy can be achieved by introducing fourth-order operators in place of the Laplacians in equation (2). Normally, these operators would require higher-order elements for a correct discretization; however, at the discrete level they can also be recast as the difference of two Laplacian operators, which require only linear elements. It is thus proposed to balance the Laplacian of pressure by the scalar quality $\nabla\cdot\mathbf{F}$ and the Laplacians of velocity by the vector $\nabla\cdot\mathbf{G}$, where $\mathbf{G}$ is a tensor, as follows:

$$\nabla\cdot(\rho\mathbf{V}) - \varepsilon_1\nabla\cdot(\nabla p - \mathbf{F}) = 0 \tag{3a}$$

$$\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla p - \nabla\cdot\boldsymbol{\tau} - \varepsilon_1\nabla\cdot(\nabla\mathbf{V} - \mathbf{G}) = 0 \tag{3b}$$

It is proposed to construct the terms $\mathbf{F}$ and $\mathbf{G}$ such that, at the discrete level, the differences between the Laplacians and the balancing terms yield a fourth-order artificial dissipation. Furthermore, the artificial viscosity terms must incorporate a function that removes the balancing terms in the neighbourhood of a shock, allowing the solution to become locally first-order accurate to avoid spurious oscillations. For these reasons, equations (3a) and (3b) are modified as follows:

$$\nabla\cdot(\rho\mathbf{V}) - \nabla\cdot[(\varepsilon_1 + \varepsilon_2)\nabla p - \varepsilon_2\mathbf{F}] = 0 \tag{4a}$$

$$\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla p - \nabla\cdot\boldsymbol{\tau} - \nabla\cdot[(\varepsilon_1 + \varepsilon_2)\nabla\mathbf{V} - \varepsilon_2\mathbf{G}] = 0 \tag{4b}$$

The coefficients ε_1 and ε_2 denote the first- and second-order artificial viscosity coefficients, which are functions of the solution, such that ε_2 tends to zero in the neighbourhood of a shock and ε_1 tends to zero away from a shock.

With the weak-Galerkin weighted residual method one obtains the following

continuity

$$\iint_A W_i\{[\rho u - (\varepsilon_1 + \varepsilon_2)p_x + \varepsilon_2 f_1]_x + [\rho v - (\varepsilon_1 + \varepsilon_2)p_y + \varepsilon_2 f_2]_y\}\,\mathrm{d}A = 0$$

weak form

$$\iint_A \{[\rho u - (\varepsilon_1 + \varepsilon_2)p_x + \varepsilon_2 f_1]W_{i_x} + [\rho v - (\varepsilon_1 + \varepsilon_2)p_y + \varepsilon_2 f_2]W_{i_y}\}\,\mathrm{d}A$$

$$= \oint_S W_i[\rho\mathbf{V}\cdot\mathbf{n} - (\varepsilon_1 + \varepsilon_2)p_n + \varepsilon_2\mathbf{F}\cdot\mathbf{n}]\mathrm{d}S \tag{5a}$$

x-momentum

$$\iint_A W_i\left\{\left[\rho u^2 + p - \frac{2\mu}{3Re}(2u_x - v_y) - (\varepsilon_1 + \varepsilon_2)u_x + \varepsilon_2 g_{11}\right]_x + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x) - (\varepsilon_1 + \varepsilon_2)u_y + \varepsilon_2 g_{12}\right]_y\right\}\mathrm{d}A = 0$$

weak form

$$\iint_A \left\{ \left[\rho u^2 + p - \frac{2\mu}{3Re}(2u_x - v_y) - (\varepsilon_1 + \varepsilon_2)u_x + \varepsilon_2 g_{11} \right] W_{i_x} + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x) - (\varepsilon_1 + \varepsilon_2)u_y + \varepsilon_2 g_{12} \right] W_{i_y} \right\} \mathrm{d}A$$
$$= \oint_S W_i p \, \mathrm{d}y + \oint_S W_i [\rho u \mathbf{V} \cdot \mathbf{n} - (\varepsilon_1 + \varepsilon_2)u_n + \varepsilon_2 \mathbf{G} \cdot \mathbf{n}] \, \mathrm{d}S$$
$$- \oint_S W_i \left[\frac{2\mu}{3Re}(2u_x - v_y)\frac{\mathrm{d}y}{\mathrm{d}S} - \frac{\mu}{Re}(u_y + v_x)\frac{\mathrm{d}x}{\mathrm{d}S} \right] \mathrm{d}S \tag{5b}$$

y-momentum

$$\iint_A W_i \left\{ \left[\rho uv - \frac{\mu}{Re}(u_y + v_x) - (\varepsilon_1 + \varepsilon_2)v_x + \varepsilon_2 g_{21} \right]_x + \left[\rho v^2 + p - \frac{2\mu}{3Re}(2v_y - u_x) - (\varepsilon_1 + \varepsilon_2)v_y + \varepsilon_2 g_{22} \right]_y \right\} \mathrm{d}A = 0$$

weak form

$$\iint_A \left\{ \left[\rho uv - \frac{\mu}{Re}(u_y + v_x) - (\varepsilon_1 + \varepsilon_2)v_x + \varepsilon_2 g_{21} \right] W_{i_x} + \left[\rho v^2 + p - \frac{2\mu}{3Re}(2v_y - u_x) - (\varepsilon_1 + \varepsilon_2)v_y + \varepsilon_2 g_{22} \right] W_{i_y} \right\} \mathrm{d}A$$
$$= \oint_S W_i p \, \mathrm{d}x + \oint_S W_i [\rho v \mathbf{V} \cdot \mathbf{n} - (\varepsilon_1 + \varepsilon_2)v_n + \varepsilon_2 \mathbf{G} \cdot \mathbf{n}] \, \mathrm{d}S$$
$$- \oint_S W_i \left[\frac{\mu}{Re}(u_y + v_x)\frac{\mathrm{d}y}{\mathrm{d}S} - \frac{2\mu}{3Re}(2v_y - u_x)\frac{\mathrm{d}x}{\mathrm{d}S} \right] \mathrm{d}S \tag{5c}$$

with W_i denoting the weight functions. At the discrete level $f_1, f_2, g_{11}, g_{12}, g_{21}, g_{22}$ are nodal variables and would be represented in the finite element framework as

$$\{\mathbf{F}, \mathbf{G}\} = \sum_{k=1}^{4} \{\mathbf{F}_k, \mathbf{G}_k\} N_k$$

where $\{\mathbf{F}_k, \mathbf{G}_k\}$ are the nodal values of $\{\mathbf{F}, \mathbf{G}\}$ and N_k are the finite element shape functions. Let

$$\psi = \kappa^{(2)}\varepsilon\left(\max_k T_j^k\right), \quad \phi = \max(0, \kappa^{(4)}\varepsilon - \psi)$$

where

$$T_j^k = \frac{\left| \max_k p_j^k + \min_k p_j^k - 2p_j \right|}{\max_k p_j^k + \min_k p_j^k - 2p_j}$$

The notation $(\cdot)_j^k$ refers to all the neighbours k of node j.

The artificial viscosity coefficients are similar to those of References 1 and 2. The functions ε_1 and ε_2 are

$$\varepsilon_1 = (|\mathbf{V}| + c)\cdot\psi, \quad \varepsilon_2 = (|\mathbf{V}| + c)\cdot\phi$$

where c is the speed of sound. The artificial viscosity is controlled by the parameter ε and the values of the coefficients $\kappa^{(2)}$ and $\kappa^{(4)}$ are similar to those of References 1 and 2, namely $\kappa^{(2)} = \frac{1}{4}$ and $\kappa^{(4)} = 0{\cdot}009$.

After replacing the density by equation (1c), the equation system (5) is linearized by Newton's method, discretized with bilinear quadrilateral isoparametric elements and integrated by Gauss–Legendre quadrature with (3×3) Gauss points. At each iteration, a fully coupled, sparse linear algebraic system of the form

$$[J]\{\Delta U\} = \alpha\{\text{Res}\} \tag{6}$$

is solved by a direct solver. Here $\Delta U = \{\Delta u, \Delta v, \Delta p\}$, $\{\text{Res}\}$ is the residual and α is a relaxation factor. In this deferred correction approach for the introduction of the second-order artificial viscosity, the added balancing terms $f_1, f_2, g_{11}, g_{12}, g_{21}, g_{22}$ are functions of the solution at the previous iteration; hence the Jacobian matrix $[J]$ is unaltered with respect to the corresponding first-order approach. The residual, however, is of higher-order accuracy.

4. BOUNDARY CONDITIONS

The following boundary conditions are imposed, with all contour integrals of equations (5) evaluated, unless otherwise noted:

Inflow: u, v, p are specified.

Walls: For viscous flows $u, v = 0$. For inviscid flows, the term $\rho\mathbf{V}\cdot\mathbf{n}$ in the continuity equation contour integral vanishes.

Outflow: For a subsonic outflow, p is specified in the contour integrals of the momentum equations. For a supersonic outflow, no back pressure is specified.

For all other boundary conditions the practice has been to replace the governing equations by extrapolation formulae, locally unidimensional equations or Riemann invariants. However, since the finite element shape functions are smooth continuous polynomials over the element, this situation can be avoided altogether. Hence, once the influence matrices of the elements on the boundary are assembled into a global one, no further treatment is required for quantities such as pressure, velocities, etc. The advantage is therefore that the discrete governing equations are not replaced by simpler approximations at the boundary.

5. ARTIFICIAL VISCOSITY BALANCING TERMS

The balancing terms **F** and **G** can be obtained either from tensor identities and nodal averaging, as shown in the first proposed scheme outlined below, or simply through the smoothing provided by nodal averaging, as in the second proposed scheme. The balancing terms are functions of the solution at the previous iteration and cannot be linearized due to the intermediate procedure required for their evaluation.

5.1. Scheme 1

The term **F** of equation (3a) is obtained from the original momentum equations

$$\mathbf{F} = f_1\mathbf{i} + f_2\mathbf{j} = \nabla p = -\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla\cdot\boldsymbol{\tau} \tag{7a}$$

where

$$f_1 = -\left\{\left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right]_x + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]_y\right\} \tag{7b}$$

$$f_2 = -\left\{\left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]_x + \left[\rho v^2 - \frac{2\mu}{3Re}(2v_y - u_x)\right]_y\right\} \tag{7c}$$

The artificial term of equation (3a), $\nabla^2 p - \nabla \cdot \mathbf{F}$, does not vanish identically, even in the steady state when the discrete momentum equations are satisfied, since the divergence of the momentum equations is not necessarily zero in a discrete sense. Nevertheless, conservation of mass is improved by at least an order of magnitude over the original method represented by (2a).

Similarly, the momentum equations are modified by adding artificial viscosity terms proportional to the Laplacian of the velocity components, with these terms balanced by a tensor obtained from the following identity:

$$\nabla^2 \mathbf{V} - \nabla(\nabla \cdot \mathbf{V}) + \nabla \times (\nabla \times \mathbf{V}) = 0 \tag{8a}$$

$$\nabla^2 \mathbf{V} - \nabla s + \nabla \times \omega = 0 \tag{8b}$$

The tensor $\mathbf{G}$ therefore becomes

$$g_{11} = s, \quad g_{12} = -\omega, \quad g_{21} = \omega, \quad g_{22} = s$$

The correction terms $f_1, f_2, g_{11}, g_{12}, g_{21}, g_{22}$ are evaluated at the nodes in accordance with the standard weak-Galerkin framework. For f_1, for example, the Galerkin integral is

$$\iint_A W_i \left\{f_1 + \left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right]_x + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]_y\right\} \mathrm{d}A = 0 \tag{9}$$

Let

$$f_1 = \sum_{j=1}^{4} \overline{f_{1j}} N_j$$

where N_j represents the finite element shape functions and $\overline{f_{1j}}$ denotes the value of f_1 at the four nodes of an element. Upon integration by parts,

$$\begin{aligned}
\sum_{j=1}^{4} \overline{f_{1j}} \iint_A W_i N_j \,\mathrm{d}A = {} & \iint_A \left\{\left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right] W_{i_x} + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right] W_{i_y}\right\} \mathrm{d}A \\
& - \oint_S W_i \left\{\left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right]\frac{\mathrm{d}y}{\mathrm{d}S} - \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]\frac{\mathrm{d}x}{\mathrm{d}S}\right\} \mathrm{d}S \\
[M]\{\overline{f_{1j}}\} = {} & \iint_A \left\{\left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right] W_{i_x} + \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right] W_{i_y}\right\} \mathrm{d}A \\
& - \oint_S W_i \left\{\left[\rho u^2 - \frac{2\mu}{3Re}(2u_x - v_y)\right]\frac{\mathrm{d}y}{\mathrm{d}S} - \left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]\frac{\mathrm{d}x}{\mathrm{d}S}\right\} \mathrm{d}S
\end{aligned} \tag{10}$$

where $[M]$ is the mass matrix and the integral on the right-hand side is a function of the solution at the previous iteration.

The other variables can be written as

$$f_2 = -\left\{\left[\rho uv - \frac{\mu}{Re}(u_y + v_x)\right]_x + \left[\rho v^2 - \frac{2\mu}{3Re}(2v_y - v_x)\right]_x\right\}$$

$$g_{11} = -\nabla\cdot\mathbf{V}, \quad g_{12} = \nabla\times\mathbf{V}, \qquad g_{21} = -\nabla\times\mathbf{V}, \quad g_{22} = -\nabla\cdot\mathbf{V}$$

where $g_{22} = g_{11}, g_{21} = -g_{12}$ and

$$\iint_A W_i(g_{11} + \nabla\cdot\mathbf{V})\,\mathrm{d}A = 0 \quad \iint_A W_i(g_{12} - \nabla\times\mathbf{V})\,\mathrm{d}A = 0$$

In summary, for this scheme, four new variables, f_1, f_2, g_{11}, g_{12}, need to be defined at the nodes as functions of the variables $\{u, v, p\}$ at the previous iteration.

5.2. *Scheme 2*

In this proposed scheme, the Laplacians of the artificial viscosity can be balanced in a slightly different manner:

$$\nabla\cdot(\rho\mathbf{V}) - \varepsilon_1[\nabla^2 p - \nabla\cdot(\nabla\hat{p})] = 0 \tag{11a}$$

$$\nabla\cdot(\rho\mathbf{V}\mathbf{V}) + \nabla p - \nabla\cdot\tau - \varepsilon_1[\nabla^2\mathbf{V} - \nabla\cdot(\nabla\mathbf{U})] = 0 \tag{11b}$$

The vector $\nabla\hat{p}$ and the Cartesian tensor $\nabla\mathbf{U}$ are constructed as follows:

$$\nabla\hat{p} = f_1\mathbf{i} + f_2\mathbf{j} = p_x\mathbf{i} + p_y\mathbf{j}$$

$$\nabla\mathbf{U} = \mathbf{U}_x + \mathbf{U}_y = (g_{11} + g_{21})\mathbf{i} + (g_{12} + g_{22})\mathbf{j} = (\mathbf{u}_x + \mathbf{v}_x)\mathbf{i} + (\mathbf{u}_y + \mathbf{v}_y)\mathbf{j}$$

where $f_1, f_2, g_{11}, g_{12}, g_{21}, g_{22}$ are evaluated within the weak-Galerkin framework, outlined in (9) and (10), such that

$$\iint_A W_i(f_1 - p_x)\,\mathrm{d}A = 0, \quad \iint_A W_i(g_{11} - u_x)\,\mathrm{d}A = 0, \quad \iint_A W_i(g_{21} - v_x)\,\mathrm{d}A = 0$$

$$\iint_A W_i(f_2 - p_y)\,\mathrm{d}A = 0, \quad \iint_A W_i(g_{12} - u_y)\,\mathrm{d}A = 0, \quad \iint_A W_i(g_{22} - v_y)\,\mathrm{d}A = 0$$

This is a more general scheme allowing a balanced artifical viscosity for the fully energy equation, should it be included. In this approach, the number of extra variables to be computed increases from four to six. However, one should note that the mass matrix in both schemes is identical for all the additional variables and needs to be assembled and decomposed only once, in a post-processing step, during each Newton iteration. Therefore, the increase in the number of variables of Scheme 2 represents a minor penalty in terms of computational cost.

Four different methods can be used to compute the nodal values of the additional variables: the mass matrix of equation (10) (CMM), the lumped mass matrix (LMM) and, as an alternative to the Galerkin averaging, extrapolation from the Gauss points to the nodes (EXT) through a 9th-order polynomial[17] and, finally, least-squares fitting (LSQ) involving sampling the derivatives of the variables at all the Gauss points of the elements surrounding each node.[18]

Note that the solutions obtained with the second-order schemes are less sensitive to the value of the artificial viscosity coefficient than those obtained with the first order one. The convergence rate of the second-order schemes, however, is directly controlled by the value of the artificial viscosity coefficient since the balancing terms are lagged from the previous iteration and do not

contribute to the Jacobian of the Newton linearization. A partial remedy to this situation is presented in the following section.

6. SEMI-IMPLICIT SECOND-ORDER BALANCING TERMS

The various schemes presented above for the construction of the second-order balancing terms have one common feature: the balancing terms are obtained from an averaging procedure. Indeed, some of the schemes are based on a sophisticated but expensive weighted-averaging procedure, the Galerkin integral. The complexity of this procedure requires both the solution of a matrix and that the balancing terms be lagged at the previous iteration, with a detrimental effect on both the convergence rate and the solution time. Since the solution of (6), the linear system of coupled equations of the Navier–Stokes FEM scheme, can be an expensive undertaking when a direct solver is used, a fully implicit scheme with quadratic convergence would be desirable to minimize solution cost. A fully implicit second-order scheme, however, requires a wider finite element stencil and would produce a system of equations with twice the current bandwidth, for which direct solvers would prove expensive and unwieldy. Hence, a compromise must be sought.

The Galerkin weighted averaging can be replaced by a simpler averaging procedure which requires neither integration nor the solution of an intermediate system of equations. This scheme will be outlined through the construction of the balancing term f_1, with the procedure being identical for the other five balancing terms introduced in Scheme 2.

6.1. Scheme 3

The definition of f_1 is

$$f_1 = p_x$$

Introducing the finite element discretization for the derivative yields the following equation:

$$f_1 = \sum_{j=1}^{4} \bar{p}_j \frac{\partial N_j}{\partial x}$$

where f_1 is intended to be a nodal value but p_x is defined only inside the element in the discrete approximation. The nodal value of f_1 is obtained by an area averaging of the values of the derivatives computed at the centroids of the elements surrounding each node, such that

$$f_1 = \frac{1}{A_{\text{total}}} \sum_{l=1}^{M} \left(A_l \sum_{j=1}^{4} \bar{p}_j \frac{\partial N_j}{\partial x}\bigg|_c \right)_l \tag{12}$$

where $\bar{p}_j$ is the value of pressure at the nodes of the element and

$$A_{\text{total}} = \sum_{l=1}^{M} A_l$$

is the sum of the areas of the element surrounding each node, M is the number of elements surrounding a node and the subscript c indicates the centroid of the element.

The expression for f_1 in equation (12) is substituted directly into equations (5a) and is eventually discretized like a nodal-based quantity by the finite element method. The areas of the elements and the derivatives of the shape functions are constants for stationary grids, hence since the balancing terms can be recast in terms of the dependent variables, the intermediate averaging procedure is eliminated and the balancing terms can be linearized with Newton's method.

Since a full linearization would double the bandwidth, compared to the schemes where the balancing terms are lagged, a partial linearization is suggested. The balancing terms contributions to the right-hand side of equation (6) must be included in their entirety. Some elements have balancing terms stiffness matrices which can only be partially assembled into the Jacobian matrix without expanding the bandwidth. It was found that if these stiffness matrices were partially assembled, the Jacobian matrix would lose its topological symmetry and the iterative procedure would diverge. Therefore, only the balancing terms stiffness matrices which can be fully assembled should be included into the Jacobian matrix to ensure stability. This is possible because the unknowns in equation (6) are the changes in the solution and the Jacobian matrix is a preconditioner for the system of equations. This procedure is stable, conserves the topological symmetry and, by including more information into the Jacobian matrix, produces better convergence rates.

7. SOME PARALLELIZATION STRATEGIES

The finite element discretization shown above leads to large sparse linear systems of equations which are not well conditioned. Two solution strategies are available: direct and iterative solvers. An efficient direct solver has been developed for the current two-dimensional applications for which speed rather than memory reduction is the main requirement. For three-dimensional problems, the memory requirements of direct solvers being prohibitive, one must resort to conjugate gradient-like iterative solvers.[19]

The Gauss elimination algorithm is divided into two processes: the matrix factorization and the back-substitution, with the former being the most expensive and the one that benefits most from parallelization.

The matrix is stored with a constant bandwidth and its indices are reversed so that it resides in memory with all rows arranged in sequential order, in the same way in which it is accessed during row elimination. The factorization step, being the most time consuming, receives particular attention: IF statements are eliminated by storing selected critical parameters into three integer vectors of length N, the number of unknowns. The first vector maps the number of rows scheduled for decomposition below each diagonal entry, the second maps the address of the end of each row and the third is used during the back-substitution to map the number of the unknowns already solved that affect the calculation of the current unknown. The three vectors are assembled before the global iterative process and need not be evaluated repeatedly if the matrix size remains constant.

The factorization step is represented schematically by three nested loops: the first loop sweeps all the entries along the diagonal of the matrix, the middle loop sweeps all the entries in each column under the diagonal and finally the inner loop handles the factorization of all the entries in the rows. Either of the inner or middle loops can be parallelized, with the parallelization of the middle loop generating less synchronization overhead and hence more performance gains. An efficient factorization is constructed so that each processor handles the multiplication and addition of a complete row. If the bandwidth is kept constant, a strict synchronization of the processors is no longer necessary as all handle an equal amount of work, with little idle time at the completion of the row operation spent waiting for the other processors to complete their task. An additional gain can be obtained by unrolling the inner row factorization loop. In this case, the processors memory caches handle the I/O of data in a more efficient manner, reducing the overhead required for fetching and storing data by increasing the size of the data blocks processed. Furthermore, on vector architectures the inner loop could also be vectorized.

The solver is written in FORTRAN, contains no machine-dependent instructions and parallelizes automatically on Silicon Graphics parallel workstations with the Power FORTRAN Accelerator software. Figure 1 shows the performance gains obtainable with this solver on an SGI

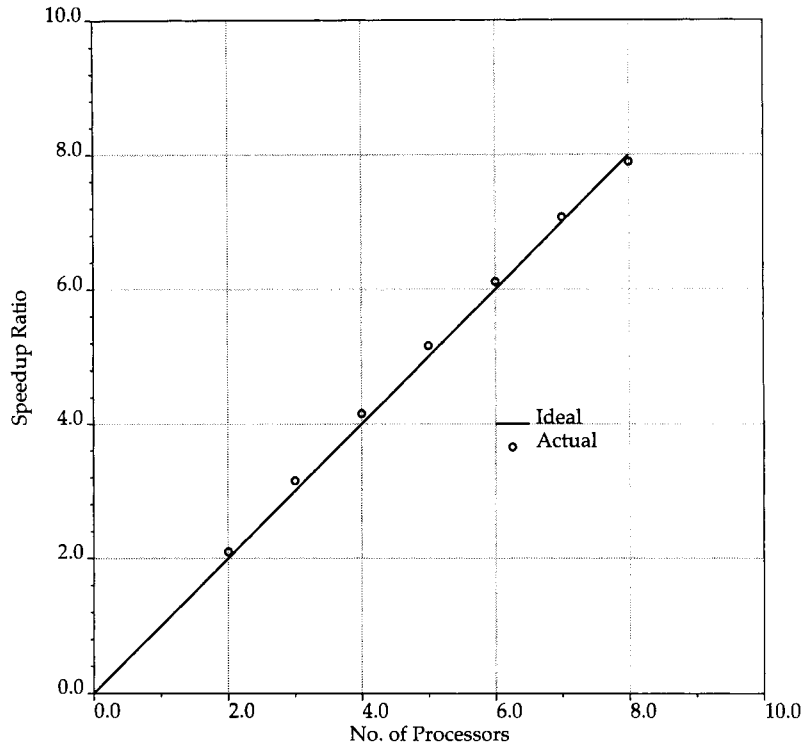

Figure 1. Performance of the direct solver on an 8-processor SGI Power Series 280 GTX with 25 MHz MIPS R3000 processors

Power Series 280 GTX shared memory computer with eight 25 MHz MIPS R3000 processors. The problem considered is transonic flow over a NACA 0012 airfoil at $M_\infty = 0{\cdot}8$ and 1.25° angle of attack, discretized using a 254×30 C-grid. The matrix size is $(23\,637 \times 371)$ and the solution requires 98 s with eight processors. Figure 2 shows the performance of the same solver on a Silicon Graphics Challenge workstation equipped with twenty 50 MHz R4400 processors. The problem considered as $M_\infty = 2$ supersonic flow in a channel with a 4 per cent circular arc airfoil using a (80×200) grid. The matrix size is $(47\,920 \times 1209)$ and the solution requires 295 s with 20 processors.

During the Newton iteration two processes dominate execution time: assembly and factorization of the global matrix. With the factorization parallelized as shown above, the assembly becomes the dominant process and it too should be parallelized in the interest of overall execution performance. The global matrix is the sum of all element matrices and is normally assembled in an element-by-element sequence. The process can be easily parallelized provided that the stiffness matrices of neighbouring elements are not assembled and stored in the global matrix simultaneously, to avoid summation conflicts. A simple scheme consists in dividing the solution domain into n blocks of roughly equal size and assigning the assembly of each block of elements to a processor. This strategy involves a minimum of synchronization overhead. For complicated geometries some checking of the element assembly sequence on the interfaces of the blocks is required to eliminate potential conflicts. More complicated schemes based on checkerboarding or colouring the elements can also be applied, if necessary. The process of parallelizing

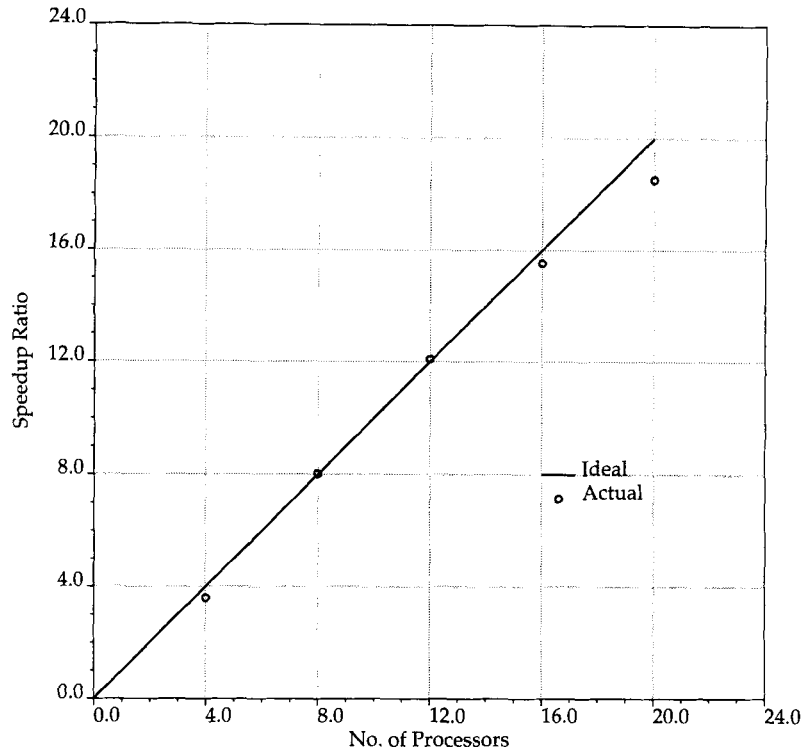

Figure 2. Performance of the direct solver on an 20-processor SGI Challenge equipped with 50 MHz MIPS R4400 processors

the assembly is inherently simple; however, the performance gain is not as dramatic as that of the matrix solver since the loop lengths in the stiffness matrices assembly are short.

8. RESULTS

A test case with an exact solution is proposed to verify the accuracy of the three second-order schemes. The Navier–Stokes equations are solved on a square domain, with a uniform mesh spacing, satisfying the following exact solution:

$$\rho = 2\sin(x)\sin(y), \quad u = \sin(x)\cos(y), \quad v = -\cos(x)\sin(y) \quad 0{\cdot}5 \leqslant x \leqslant 0{\cdot}7, \quad 0{\cdot}5 \leqslant y \leqslant 0{\cdot}7$$

The original continuity equation, equation (1), is satisfied identically and a forcing function for each of the momentum equations can be derived from the exact solution. Five progressively refined grids have been used: (16 × 16), (24 × 24), (32 × 32), (48 × 48) and (64 × 64) elements. The Reynolds number is set at 1000. An error norm defined as

$$\text{Error} = \iint_A |U_{\text{Exact}} - U_{\text{Numerical}}| \, dA$$

is calculated for density, velocities and pressure. In Figure 3 the $\{\rho, u, v, p\}$ errors of Scheme 1, for both the lumped and consistent matrices, do not tend to zero as the mesh is refined, suggesting

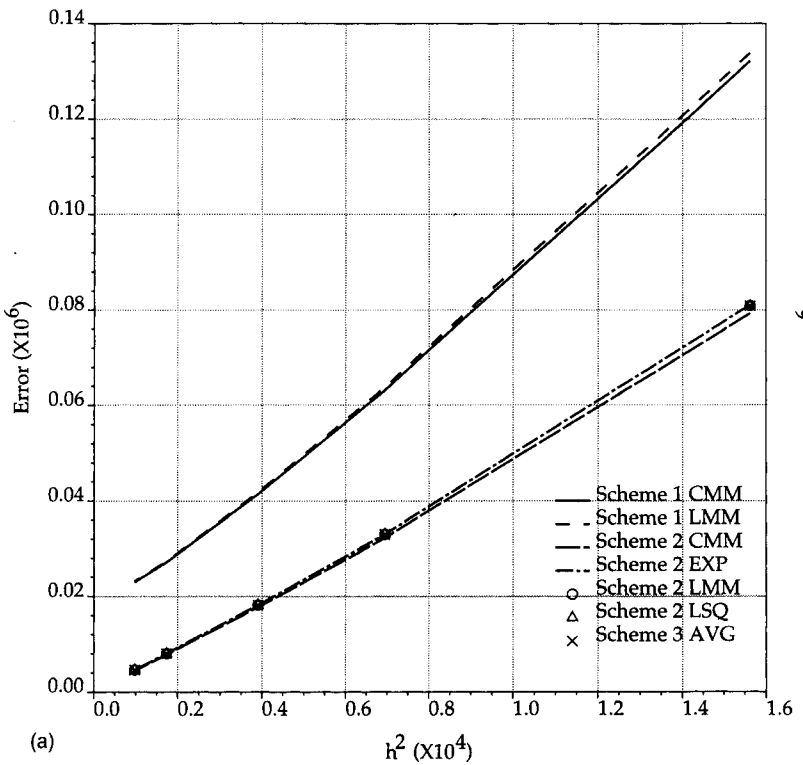

Figure 3(a). Order of accuracy test: density error as a function of h^2

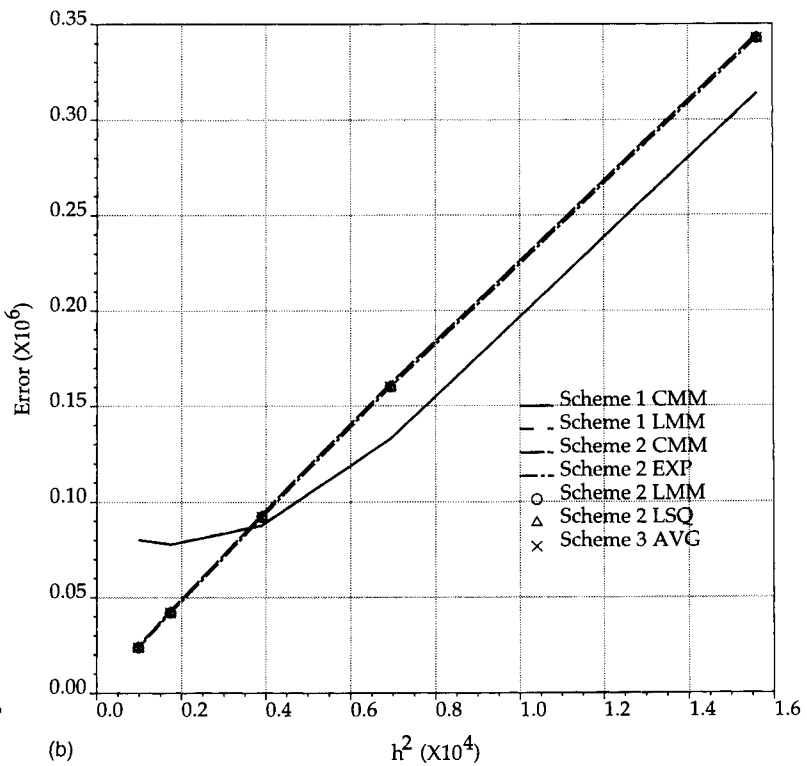

Figure 3(b). Order of accuracy test: u-velocity error as a function of h^2

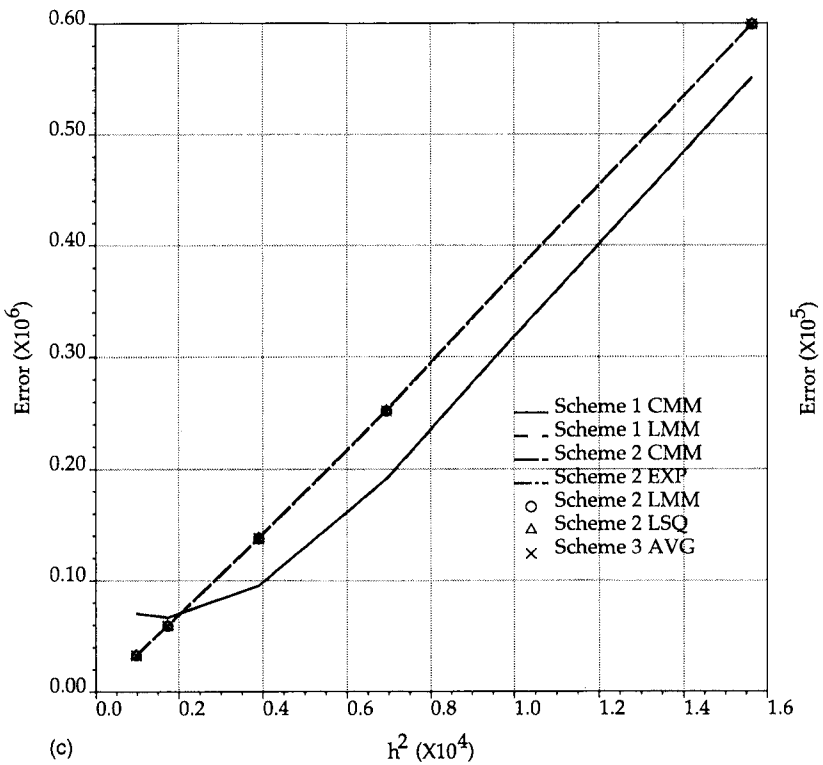

Figure 3(c). Order of accuracy test: v-velocity error as a function of h^2

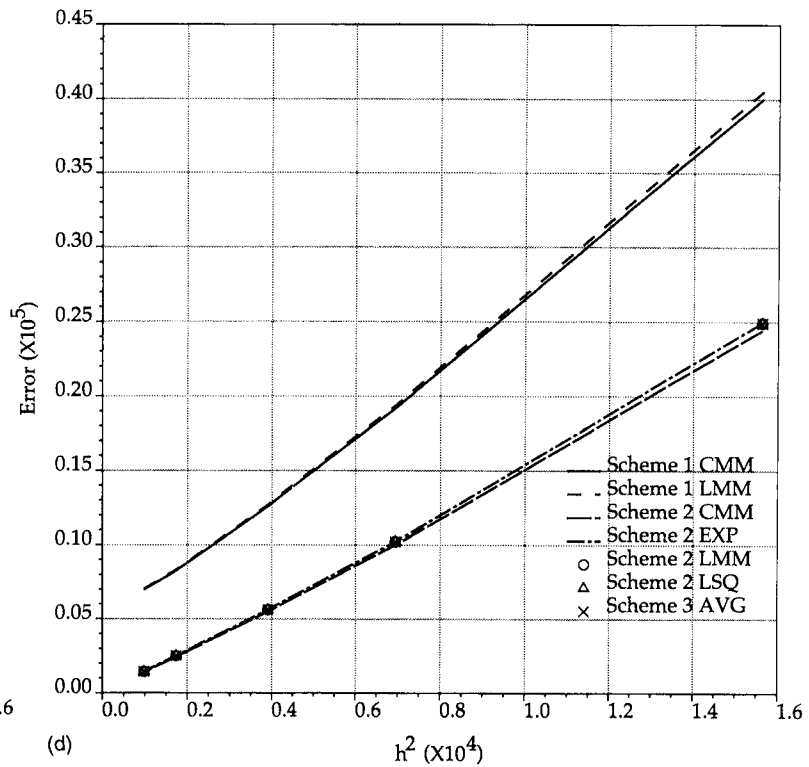

Figure 3(d). Order of accuracy test: pressure error as a function of h^2

that only Schemes 2 and 3 are second-order accurate irrespective of the method used to compute the balancing terms.

The second test case is for supersonic inviscid flow over a 15° wedge, at $M_\infty = 2$. The solution procedure is started with the first-order scheme and uniform artificial viscosity coefficients. The artificial viscosity parameters are initially set to $\varepsilon_1 = 0{\cdot}05$, $\varepsilon_2 = 0$ uniformly throughout and ε_1 is lowered to 0·01 and 0·005 at the intermediate RMS residual value of 10^{-6}. The first grid adapting is performed and the code is switched to second-order accuracy with the balancing terms of Scheme 3 and artificial viscosity parameter $\varepsilon = 0{\cdot}2$. After the intermediate residual is reached a second adaptation is carried out and the solution is finally allowed to converge to the final residual of 10^{-8} with $\varepsilon = 0{\cdot}1$. The relaxation factor in (6) is set to $\alpha = 0{\cdot}5$ to stabilize the second-order solution in the presence of strong normal and oblique shocks. The (120×48) grid was initially nearly uniform in the streamwise direction and during the solution procedure it was adapted using a mesh adapting strategy based on a spring analogy.[20] The initial, intermediate and final grids are shown in Figure 4. The Mach number contours obtained from the three grids are shown in Figure 5. The final grid was obtained by specifying minimum and maximum mesh spacings of 3·5 and 150 per cent of the average mesh size along each grid line, respectively.

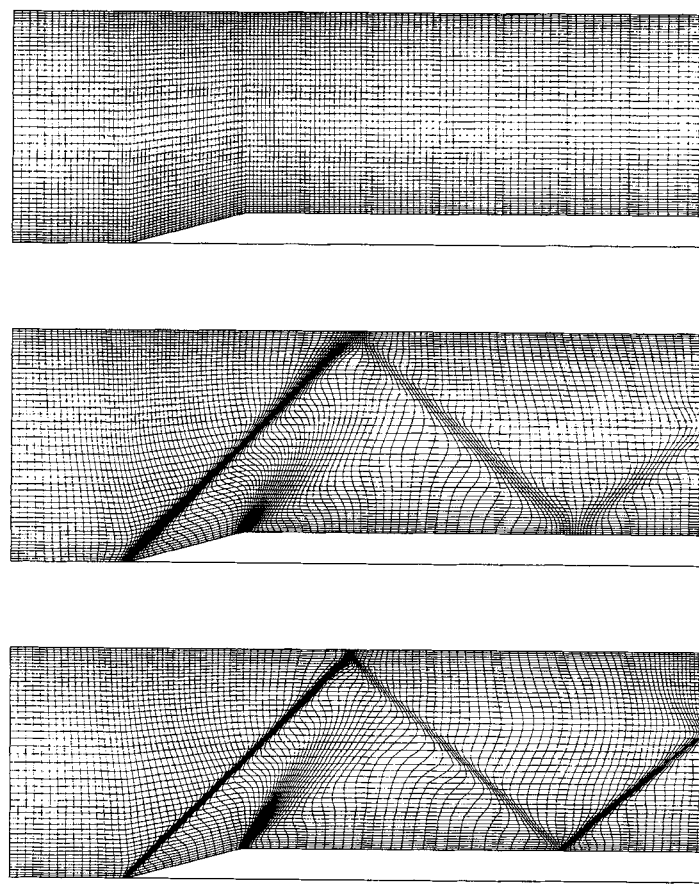

Figure 4. Supersonic flow over a 15° wedge: initial, intermediate and final adapted grid

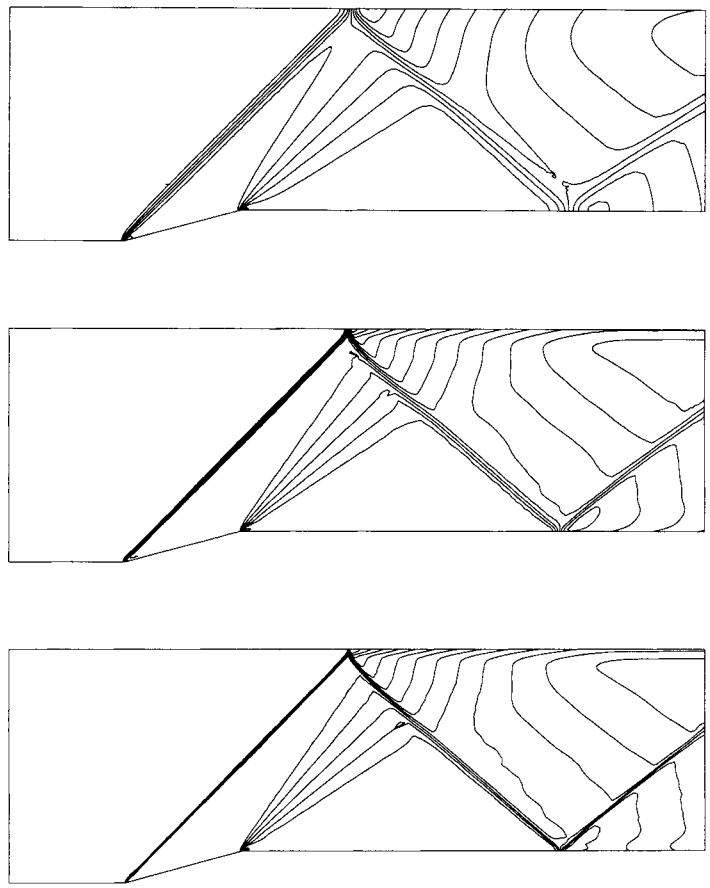

Figure 5. Supersonic flow over a 15° wedge: initial, intermediate and final Mach number contours ($M_{min} = 0{\cdot}65$, $\Delta M = 0{\cdot}05$)

Figure 6 is the convergence curve which shows the effects of the five cycles of artificial viscosity and mesh refinement. Solution times are of the order of 30 s per iteration for Scheme 3 on a Silicon Graphics Challenge computer, with four 150 MHz R-4400 processors.

The third test is the AGARD01 case,[21] namely that of transonic inviscid flow over a NACA0012 airfoil, at $M_\infty = 0{\cdot}8$ and 1·25° angle of attack. A (200 × 32) O-grid is used, with 200 elements on the airfoil surface (Figure 7). For this test case, the solution is started with the second-order dissipation but with the viscosity coefficient set to a large value in order to stabilize the Newton method. This high value is then unloaded in 4 successive steps, $\varepsilon = 0{\cdot}25, 0{\cdot}15, 0{\cdot}10$ and 0·05, converging at each value to an intermediate residual of 5×10^{-6}, with the last step carried to 10^{-12}. The relaxation factor in (6) is set to $\alpha = 0{\cdot}5$ to stabilize the second-order solution in the presence of the strong shock. The Mach number and pressure contours are shown in Figures 8 and 9, while the Mach number distribution on the surface is compared in Figure 10 to the results of Pulliam and Barton[21] obtained on a (561 × 65) C-grid and those of Jameson[22] on a (320 × 64) O-grid. Note that in spite of the finer grids, the results of Pulliam and Barton and those of Jameson show shocks captured across several grid points. Convergence of the solution is shown in Figure 11. The saw-tooth appearance of the convergence history is due to the four

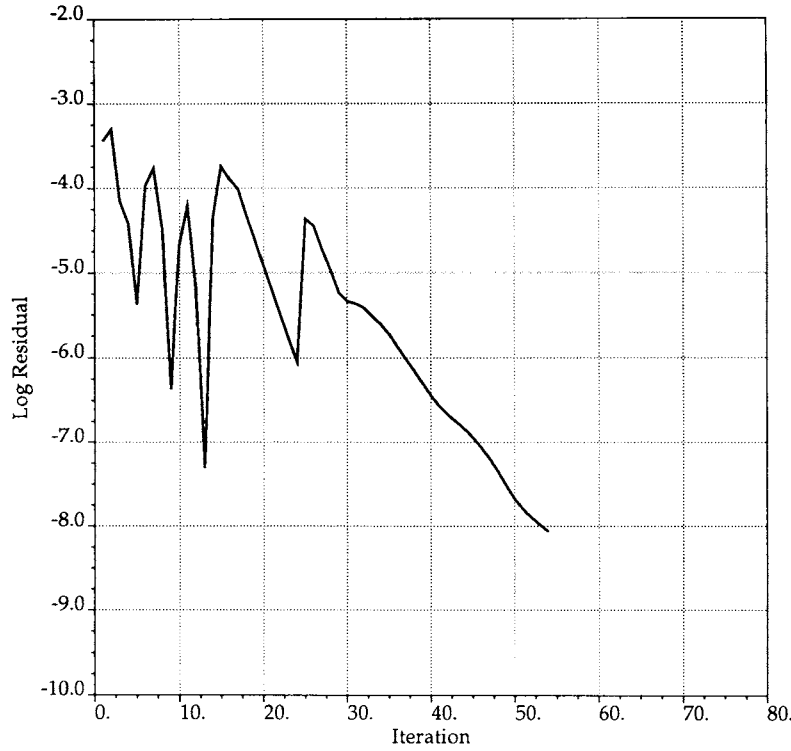

Figure 6. Convergence history for supersonic flow over a 15° wedge

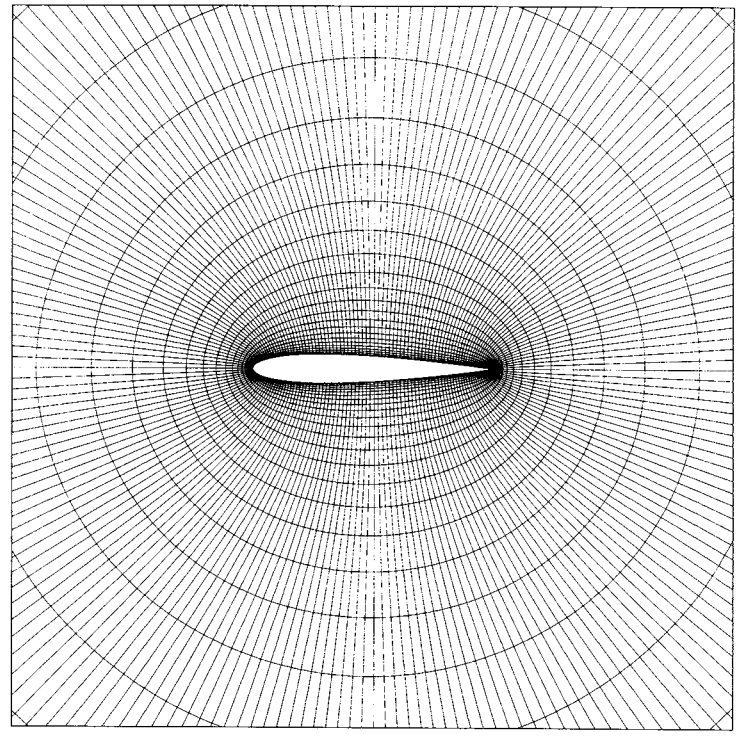

Figure 7. Detail of the (200 × 32) O-grid around a NACA0012 airfoil

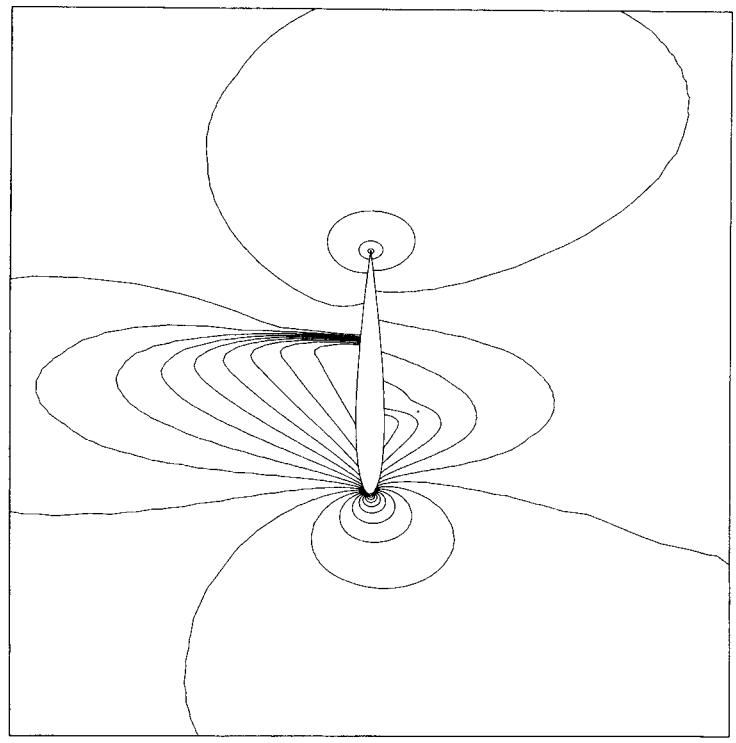

Figure 9. Inviscid flow over a NACA0012 airfoil at $M = 0{\cdot}8$ and $\alpha = 1{\cdot}25°$: non-dimensional pressure contours ($P_{min} = 0$, $\Delta P = 0{\cdot}05$)

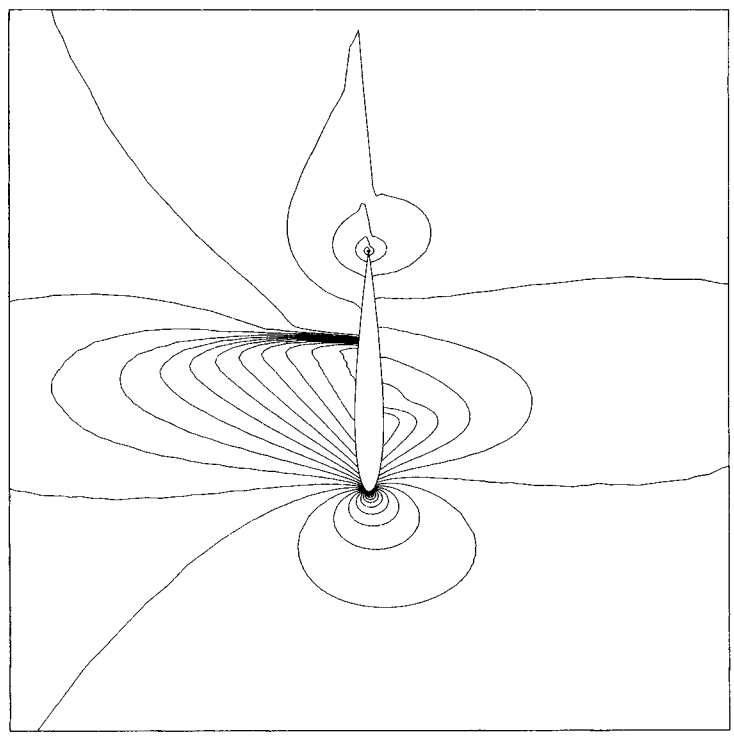

Figure 8. Inviscid flow over a NACA0012 airfoil at $M = 0{\cdot}8$ and $\alpha = 1{\cdot}25°$: Mach number contours ($M_{min} = 0{\cdot}1$, $\Delta M = 0{\cdot}05$)

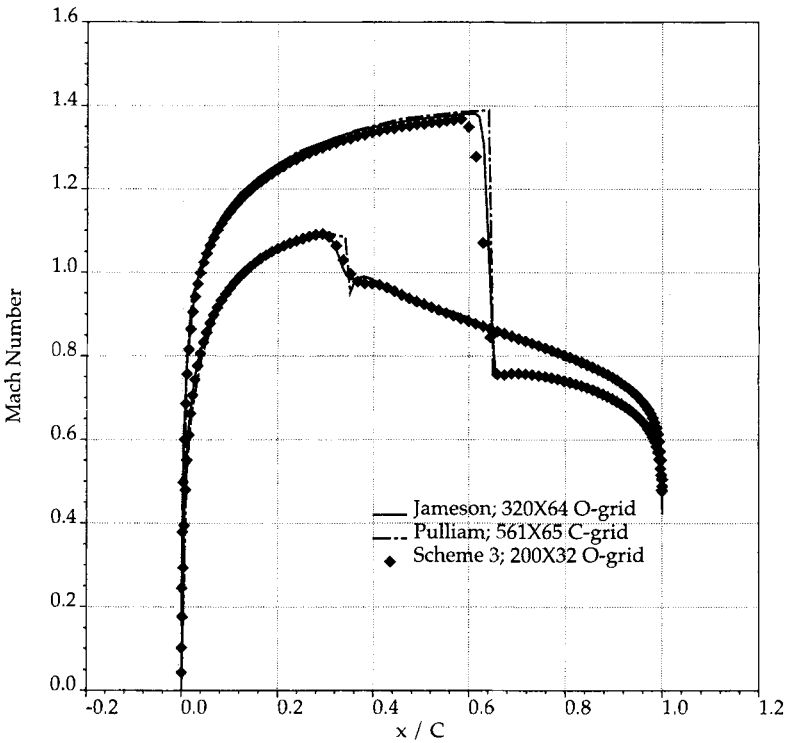

Figure 10. Inviscid flow over a NACA0012 airfoil at $M = 0{\cdot}8$ and $\alpha = 1{\cdot}25°$: surface Mach number distribution

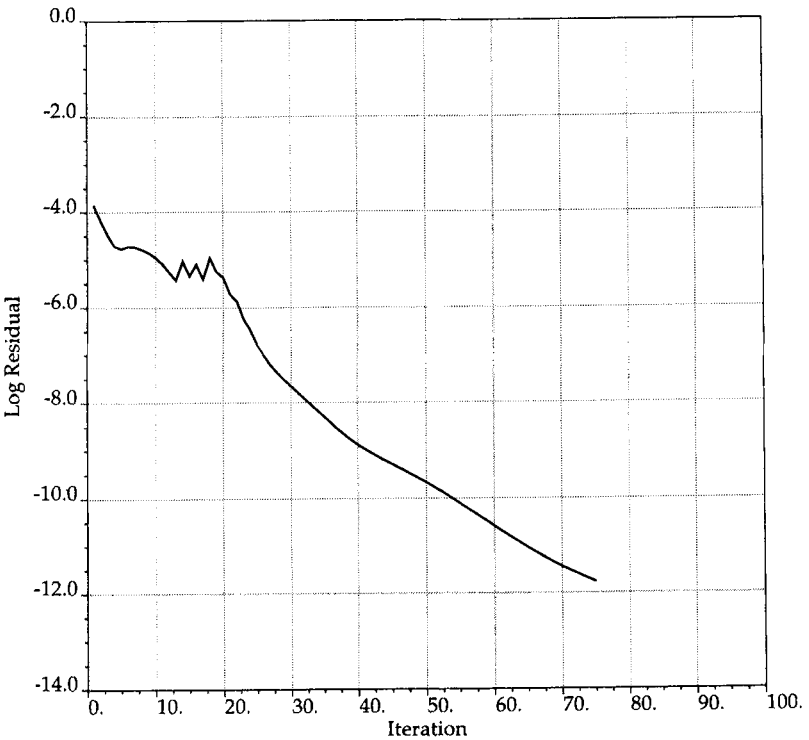

Figure 11. Inviscid flow over a NACA0012 airfoil at $M = 0{\cdot}8$ and $\alpha = 1{\cdot}25°$: convergence history, showing the effects of four artificial viscosity cycles

artificial viscosity cycles. Solutions times are of the order of 33 s per iteration on a Silicon Graphics Challenge computer with eight 150 MHz R-4400 processors, achieving speeds of 20 Mflops per processor.

The fourth case is for transonic viscous flow over a NACA0012 airfoil, at $M_\infty = 0{\cdot}9$, $Re = 5000$ and 0° angle of attack. Details of the (200×48) C-grid, with 120 elements on the surface, are shown in Figure 12. The solution was started by marching in Reynolds number from 1000 to 5000 with uniform first-order dissipation and $\varepsilon_1 = 0{\cdot}005$, $\varepsilon_2 = 0$. Figure 13 shows the convergence curve with machine accuracy reached in 10 Newton iterations. The fully converged first-order scheme was then used as input for the second-order schemes 1, 2 and 3.

To illustrate the difference between first- and second-order accuracy of the finite element schemes, consider the following: the scale factor of the natural viscosity terms of (2b) is $1/Re$ while the scale factor of the artificial viscosity terms is ε_1. The overall scale factor of the combined viscosities is $1/Re' = (1/Re) + \varepsilon_1 = 0{\cdot}0052$ for this particular test case, yielding an effective Reynolds number, Re', of 192·3. The second-order-accurate scheme at this Reynolds number and $M_\infty = 0{\cdot}9$ produces results that the almost identical to those of the first-order scheme at $Re = 5000$, as can be seen by comparing the Mach number contours on the upper and lower halves of Figure 14. The uniform artificial viscosity of the first-order scheme has therefore effectively lowered the Reynolds number by a factor of 26! Note also that above $Re = 200$ the first-order artificial viscosity dominates over the natural viscosity, hence the value of ε_1 must be kept lower than the inverse of the Reynolds number. The only way to achieve this objective is to

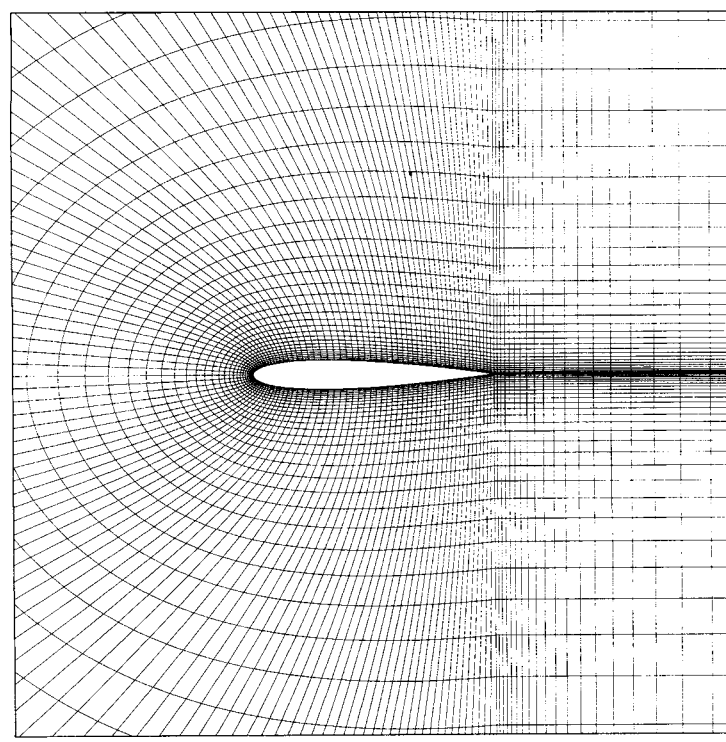

Figure 12. Detail of the (200×48) C-grid for laminar transonic viscous flow over a NACA0012 airfoil

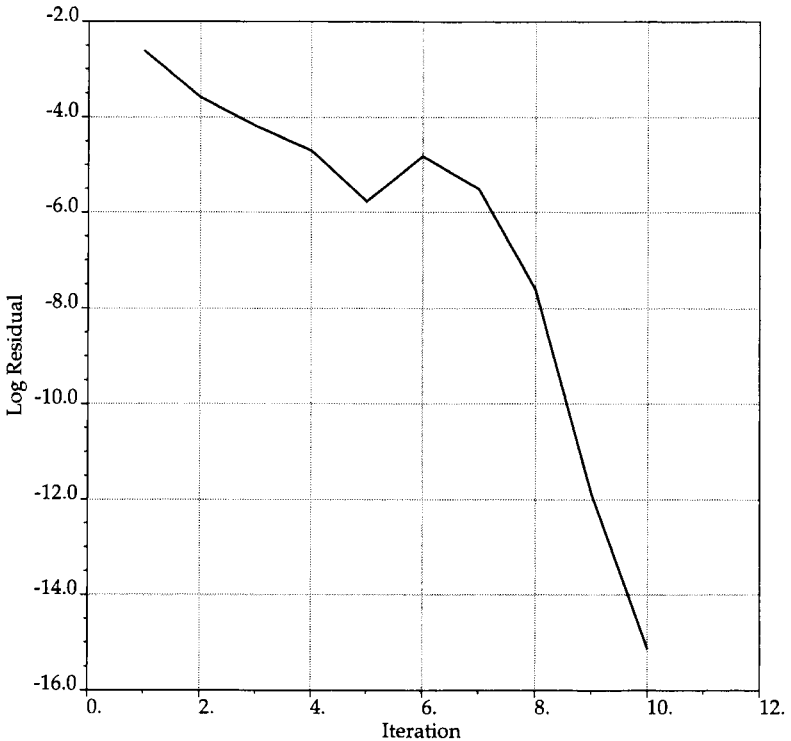

Figure 13. Laminar transonic viscous flow over a NACA0012 airfoil at $M = 0{\cdot}9$, $Re = 5000$ and $\alpha = 0°$: convergence history of the first-order solution

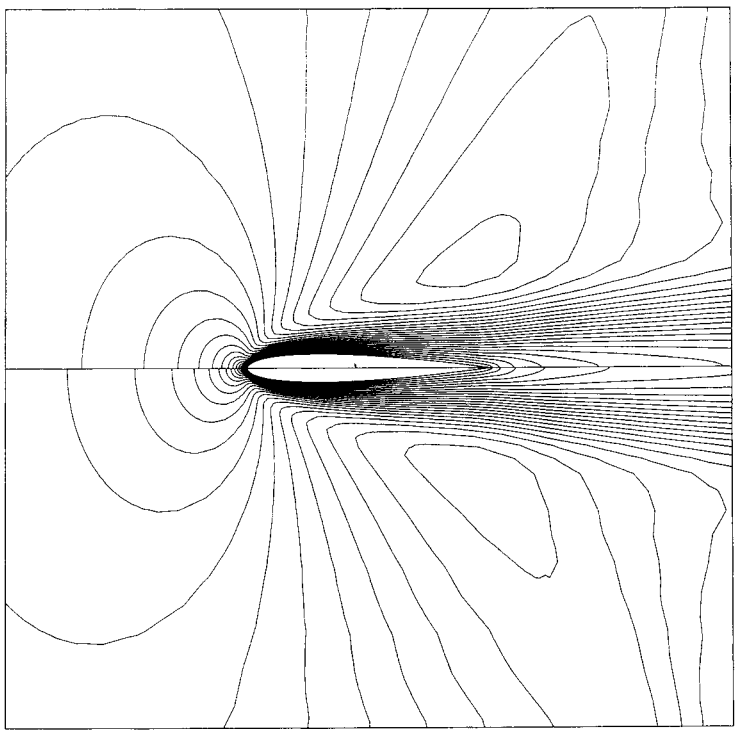

Figure 14. Comparison of the Mach number contours of the first-order solution at $M = 0{\cdot}9$, $Re = 5000$ and $\alpha = 0°$, at the top of the figure, with those of the second-order solution at $M = 0{\cdot}9$, $Re = 192{\cdot}3$ and $\alpha = 0°$ at the bottom ($M_{min} = 0$, $\Delta M = 0{\cdot}05$)

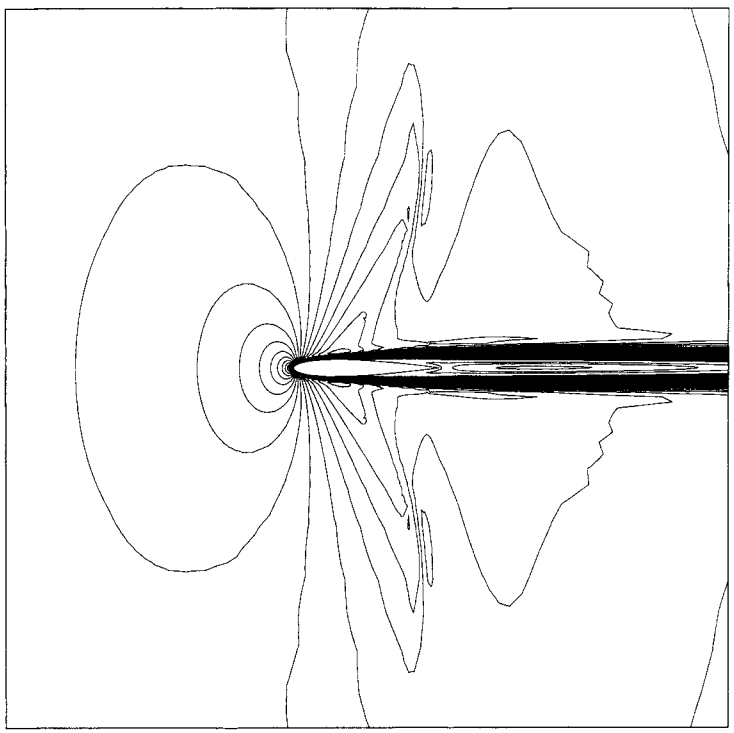

Figure 15. Laminar transonic viscous flow over a NACA0012 airfoil at $M = 0{\cdot}9$, $Re = 5000$ and $\alpha = 0°$: Mach number contours of the second-order solution, Scheme 3 ($M_{min} = 0$, $\Delta M = 0{\cdot}05$)

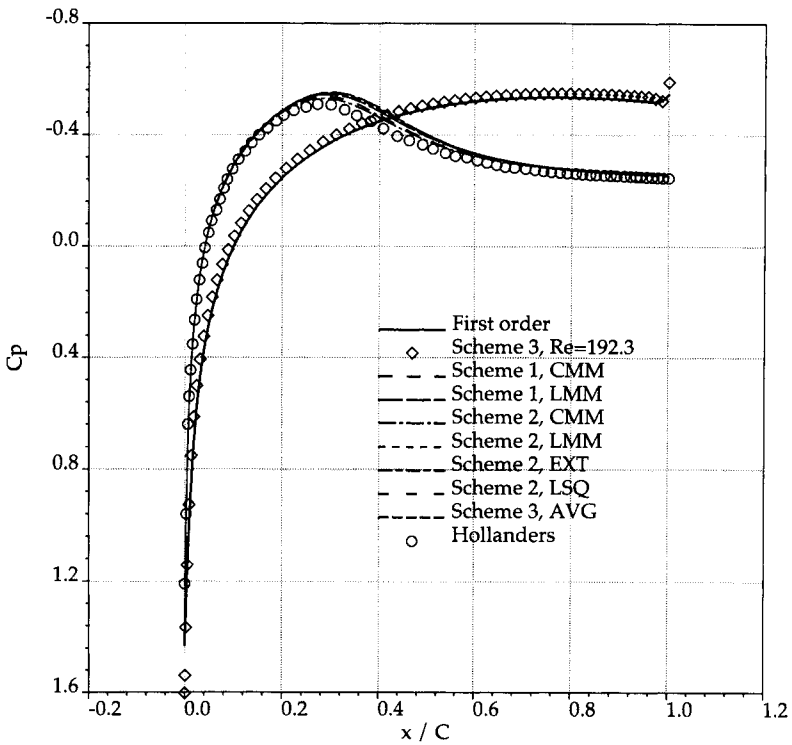

Figure 16. Cp distributions of the three second-order schemes for transonic laminar viscous flow over a NACA0012 airfoil at $M = 0{\cdot}9$, $Re = 5000$ and $\alpha = 0°$

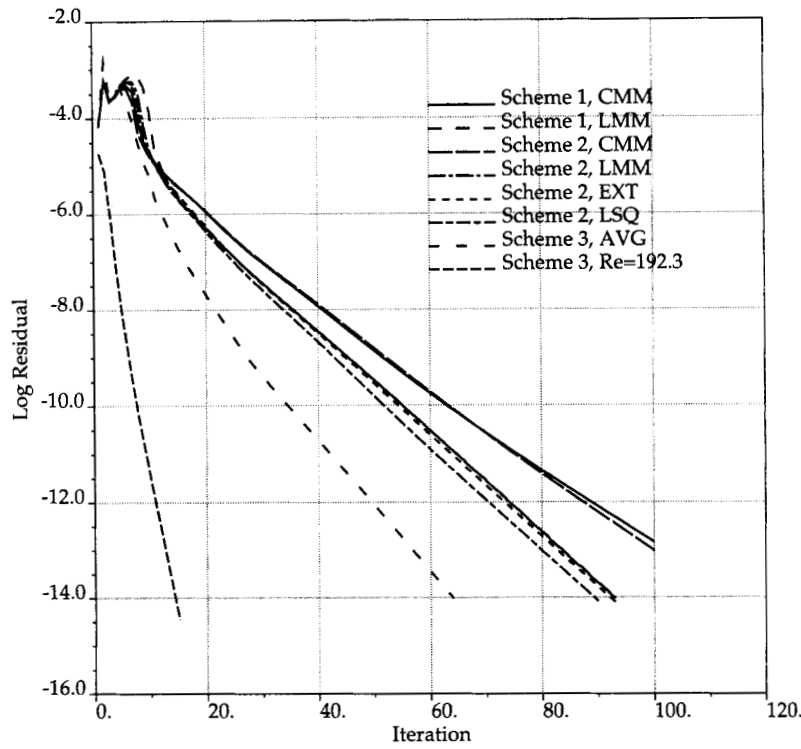

Figure 17. Convergence histories of the three second-order schemes for transonic laminar viscous flow over a NACA0012 airfoil at $M = 0{\cdot}9$, $Re = 5000$ and $\alpha = 0°$

refine the grid progressively as the Reynolds number increases, with the solution cost quickly becoming prohibitive.

Figure 15 shows the Mach number contours obtained for $M_\infty = 0{\cdot}9$ and $Re = 5000$ with Scheme 3. The artificial viscosity parameters were set at $\varepsilon_1 = 0$, $\varepsilon_2 = 0{\cdot}001$ throughout the solution domain. The relaxation factor in (6) is set to $\alpha = 0{\cdot}9$ to stabilize the second-order solution. The improvement in the results achieved with the second-order scheme is noticeable. The surface Cp distribution obtained with the first and all the second-order schemes is compared in Figure 16 to the results of the finite volume scheme of Hollanders and Ravalason.[23] Note again the good agreement between the Cp distribution of the first-order scheme at $Re = 5000$ and the second-order scheme at $Re = 192{\cdot}3$ Figure 17 is a comparison of the convergence rates of Schemes 1, 2 and 3 using the various methods to compute the balancing terms. Schemes 1 and 2 with balancing terms evaluated via a consistent mass matrix (CMM) have the slowest convergence rates. Convergence is improved by switching to other methods of evaluating the second-order balancing terms. Scheme 3 has the best convergence rate for this transonic test case. Solution times were of the order of 136 s per iteration on a Silicon Graphics Challenge computer with four 150 MHz R-4400 processors.

9. CONCLUSIONS

Numerical solutions for transonic inviscid and viscous laminar flows, using a higher-order dissipation, have been presented. The scheme is based on a second-order artificial viscosity

constructed with simple Laplacians of the primitive variables $\{u, v, p\}$, balanced by additional terms obtained either from the derivatives of the primitive variables or the momentum equations or from a tensor identity. The governing equations are linearized with Newton's method and integrated with a Galerkin finite element approach. The unknowns are solved for simultaneously, in a fully coupled manner, with an efficient parallel direct solver. The second-order accuracy of the scheme has been formally demonstrated through a simple test for which an exact solution exists. Work is underway to apply efficient iterative schemes for the solution of the coupled system for three-dimensional applications as in Reference 19.

REFERENCES

1. W. Schmidt and A. Jameson, 'Euler Solvers as an Analysis Tool for Aircraft Aerodynamics', in W. G. Habashi (ed.), *Advances in Computational Transonics*, Pineridge Press, Swansea, 1985, pp. 371–404.
2. T. H. Pulliam, 'Implicit finite-difference methods for the Euler equations', in W. G. Habashi (ed.), *Advances in Computational Transonics*, Pineridge Press, Swansea, 1985, pp. 503–542.
3. P. L. Roe, 'Characteristics-based schemes for the Euler equations', *Ann. Rev. Fluid Mech.*, **18**, 337–365 (1986).
4. B. Van Leer, J. Thomas, P. L. Roe and R. Newsome, 'A comparison of numerical flux formulas for the Euler and Navier–Stokes equations', *AIAA Paper 87-1104*, 1987.
5. T. J. R. Hughes and A. Brooks, 'A theoretical framework for Petrov–Galerkin methods with discontinuous weight functions: application to the streamline upwind procedure', in R. H. Gallagher *et al.* (eds.), *Finite Element in Fluids*, Vol. 4, Wiley, New York, 1982, pp. 47–65.
6. T. E. Tezduyar and T. J. R. Hughes, 'Finite element formulation for convection dominated flows with particular emphasis on the compressible Euler equations', *AIAA Paper 83-0125*, 1983.
7. O. Hassan, K. Morgan and J. Peraire, 'An implicit finite element method for high speed flows', *AIAA Paper 90-0402*, 1990.
8. G. S. Baruzzi, W. G. Habashi and M. M. Hafez, 'Non-unique solutions of the Euler equations', in W. F. Ballhaus, Jr and Y. M. Haussaini (eds.), *Advances in Fluid Dynamics*, Springer, New York, 1989, pp. 1–10.
9. G. S. Baruzzi, W. G. Habashi and M. M. Hafez, 'Finite element solutions of the Euler equations for transonic external flows', *AIAA J.*, **29**, 1886–1893 (1991).
10. M. M. Hafez and M. Soliman, 'Numerical solution of the incompressible Navier–Stokes equations in primitive variables on unstructured grids', *AIAA Paper 91-1561*, 1991.
11. G. Fernandez and M. M. Hafez, 'Finite element simulation of compressible flow with shocks', *AIAA Paper 91-1551*, 1991.
12. G. S. Baruzzi, W. G. Habashi and M. M. Hafez, 'A second order method for the finite element solution of the Euler and Navier–Stokes equations', *Proc. 13th Int. Conf. on Numerical Methods in Fluid Dynamics*, Springer, Roma, Italy, July 1992, pp. 509–513.
13. G. S. Baruzzi, W. G. Habashi and M. M. Hafez, 'An improved finite element method for the solution of the compressible Euler and Navier–Stokes equations', *Proc. 1st European Computational Fluid Dynamics Conf.* (ECCOMAS), Elsevier, Brussels, Belgium, September 1992, Vol. 2, pp. 643–650.
14. T. H. Pulliam, 'A computational challenge: Euler solution for ellipses', *AIAA J.* **28**, 1703–1704, (1990).
15. M. F. Peeters, W. G. Habashi and B. Q. Nguyen, 'Finite element solution of the incompressible Navier–Stokes equations by a Helmholtz velocity decomposition', *Int. j. numer. methods fluids*, **13**, 135–144 (1991).
16. M. F. Peeters, W. G. Habashi, B. Q. Nguyen and P. L. Kotiuga, 'Finite element solutions of the Navier–Stokes equations for compressible internal flows', *AIAA J. Propulsion and Power*, **8**, 192–198 (1992).
17. E. Hinton, F. C. Scott and R. E. Ricketts, 'Local least squares stress smoothing for parabolic isoparametric elements', *Int. j. numer. methods eng.*, **9**, 235–239 (1975).
18. H. P. Langtangen, 'A method for smoothing derivatives of multilinear finite element fields', *Commun. appl. numer. methods*, **5**, 275–281 (1989).
19. L. C. Dutto, W. G. Habashi, M. Fortin and M. P. Robichaud, 'Parallelizable block-diagonal preconditiones for 3D viscous compressible flow calculations', *AIAA Paper 93-3309, Proc. 11th AIAA Computationl Fluid Dynamics Conf.*, Orlando, Florida, Vol. 2, July 1993, pp. 135–143.
20. G. S. Baruzzi, 'Structured mesh grid adapting based on a spring analogy', *Proc. CFD'93 Conf.*, CERCA (Centre for Research on Computation and its Applications), Montreal, June 1993, pp. 425–436.
21. T. H. Pulliam and J. T. Barton, 'Euler computations of AGARD Working Group 07 airfoil test cases', *AIAA Paper 85-0018*, 1985.
22. H. Viviand, 'Test cases for inviscid flow field methods', AGARD Advisory Report No. 211, 1985, pp. 6–21, 22.
23. H. Hollanders and W. Ravalason, 'Résolution des équations de Navier–Stokes en fluide compressible par méthode implicite', *La Recherche Aérospatiale*, No. 1, Jan.-Feb. 1986, pp. 23–46.