LEAST SQUARES FINITE ELEMENT SOLUTION OF COMPRESSIBLE AND INCOMPRESSIBLE FLOWS

D. LEFEBVRE, J. PERAIRE AND K. MORGAN*

Department of Aeronautics, Imperial College of Science, Technology and Medicine, London SW7 2BY, UK
*and *Department of Civil Engineering, University College, Swansea SA2 8PP, UK*

ABSTRACT

We investigate the application of a least squares finite element method for the solution of fluid flow problems. The least squares finite element method is based on the minimization of the L2 norm of the equation residuals. Upon discretization, the formulation results in a symmetric, positive definite matrix system which enables efficient iterative solvers to be used. The other motivations behind the development of least squares finite element methods are the applicability of higher order elements and the possibility of using the norm associated to the least squares functional for error estimation. For steady incompressible flows, we develop a method employing linear and quadratic triangular elements and compare their respective accuracy. For steady compressible flows, an implicit conservative least squares scheme which can capture shocks without the addition of artificial viscosity is proposed. A refinement strategy based upon the use of the least squares residuals is developed and several numerical examples are used to illustrate the capabilities of the method when implemented on unstructured triangular meshes.

KEY WORDS Triangular elements Compressible flows Adaptive refinement

INTRODUCTION

A least squares finite element method has been proposed[1,2] for the solution of the steady incompressible Navier–Stokes equations. The method is based on the minimization of the L2 norm of the equation residuals and, using a finite element discretization, results in a symmetric, positive-definite matrix system. This is a major advantage over alternative finite element methods. For example, the mixed velocity–pressure formulation requires the solution of a saddle point problem and is therefore subject to the Ladyshenskaya–Babuska–Brezzi (LBB) stability condition[3,4] which rules out the use of simple element forms. In the last decade, much effort has been devoted to the design of new mixed element forms satisfying the stability requirements and the reader is referred to chapter 12 of Reference 5 for an up-to-date description of the difficulties involved. Petrov–Galerkin formulations[6–8] are an alternative in which the weighting functions are modified in such a way as to retain stability even when equal order interpolations are used for velocity and pressure. However, these methods rely on the choice of upwind parameters and also result in non-symmetric matrix systems when applied to the incompressible Navier–Stokes equations.

In the next section the method originally proposed by Jiang[1,9] for the steady Navier–Stokes equations is implemented using triangular elements in conjunction with an incomplete Choleski conjugate gradient algorithm. The least squares formulation falls into the general category of Petrov- Galerkin formulations while avoiding the above-mentioned difficulties. We also investigate the respective accuracy of linear and quadratic approximations. The performance of the least squares method is illustrated for high Reynolds number cavity flows and various flows past airfoils.

0961–5539/92/020099–15$2.00

Received September 1991

Later, a similar approach is proposed for the solution of the compressible Euler equations. Following recent developments by Jiang and Povinelli[2] for the Euler equations written in a non-conservative form and also by Bruneau[10,11] for the steady Euler equations, we propose a conservative implicit time differenced finite element scheme, applicable on unstructured meshes, which is based on the minimization of the least squares norm of the equations residuals. The scheme is unconditionally stable and we show that the inherent numerical diffusion of the scheme is sufficient to capture shocks without the need for explicitly adding artificial dissipation.

Typical finite element methods for the solution of the Euler equations are the Taylor–Galerkin formulation[12] and the streamline upwind Petrov–Galerkin (SUPG) method[13]. These methods can be applied on unstructured meshes and, with the incorporation of adaptive refinement techniques, have proved to be very successful[12,14,15]. We propose an adaptive refinement technique in which the adaptivity is driven by the norm associated with the least squares formulation. The performance of the least square finite element method and the refinement strategy is illustrated by several numerical experiments.

LEAST SQUARES SOLUTION OF STEADY INCOMPRESSIBLE FLOWS

We consider first the case of the incompressible Navier–Stokes equations. The formulation to be used is first described in some detail.

Vorticity–velocity–pressure formulation

The least squares finite element method is based on the minimization of the L2-norm of the equation residuals. Therefore, the presence of second order derivatives requires the use of expensive C^1 approximations. One possible way to avoid this difficulty is to introduce the vorticity as a new variable and express the viscous terms in terms of the vorticity. Another possibility would be to introduce the stresses as independent variables but we believe this would not result in a cost effective formulation. In two dimensions, the vorticity is defined by:

$$w=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y} \tag{1}$$

where u and v are non-dimensional velocity components. The steady incompressible Navier–Stokes equations can now be written as the first-order dimensionless differential system:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+\frac{\partial p}{\partial x}+\frac{1}{Re}\frac{\partial w}{\partial y}=f_x \tag{2}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+\frac{\partial p}{\partial y}-\frac{1}{Re}\frac{\partial w}{\partial x}=f_y \tag{3}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{4}$$

where p is the pressure, Re the Reynolds number and f_x and f_y are body force components.

The convective terms in the two momentum equations can be linearized by successive substitution. If u^n and v^n denote the velocity components at the previous non-linear step, the least squares method consists in minimizing the functional:

$$\pi=\frac{1}{2}\int\left(w+\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x}\right)^2 \mathrm{d}\Omega+\frac{1}{2}\int\left(u^n\frac{\partial u}{\partial x}+v^n\frac{\partial u}{\partial y}+\frac{\partial p}{\partial x}+\frac{1}{Re}\frac{\partial w}{\partial y}-f_x\right)^2 \mathrm{d}\Omega+$$
$$\frac{1}{2}\int\left(u^n\frac{\partial v}{\partial x}+v^n\frac{\partial v}{\partial y}+\frac{\partial p}{\partial y}-\frac{1}{Re}\frac{\partial w}{\partial x}-f_y\right)^2 \mathrm{d}\Omega+\frac{1}{2}\int\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^2 \mathrm{d}\Omega \tag{5}$$

Minimization of the functional π and interpolation using finite element basis functions leads to the least squares finite element formulation:

$$KU^{n+1}=F \tag{6}$$

where

$$K_{ij}=\int\left(M\cdot N_i+A\frac{\partial N_i}{\partial x}+B\frac{\partial N_i}{\partial y}\right)^{\mathrm{T}}\left(M\cdot N_j+A\frac{\partial N_j}{\partial x}+B\frac{\partial N_j}{\partial y}\right)\mathrm{d}\Omega \tag{7}$$

and

$$F_i=\int\left(M\cdot N_i+A\frac{\partial N_i}{\partial x}+B\frac{\partial N_i}{\partial y}\right)^{\mathrm{T}}f\,\mathrm{d}\Omega \tag{8}$$

Here

$$M=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{pmatrix}\quad A=\begin{pmatrix}1&0&0&0\\u^n&0&1&0\\0&u^n&0&-\dfrac{1}{Re}\\0&-1&0&0\end{pmatrix}\quad B=\begin{pmatrix}0&1&0&0\\v^n&0&0&\dfrac{1}{Re}\\0&v^n&1&0\\1&0&0&0\end{pmatrix}$$

$$f=\begin{Bmatrix}0\\f_x\\f_y\\0\end{Bmatrix}\quad\text{and}\quad U^{n+1}=\begin{Bmatrix}u\\v\\p\\w\end{Bmatrix} \tag{9}$$

and N_i denotes the finite element shape functions used to interpolate the nodal unknowns according to:

$$U^{n+1}=N_iU_i^{n+1} \tag{10}$$

An alternative possibility would be to use different weightings for each equation separately. The condition number of the resulting system would be affected and this could in turn improve the performance of any iterative solver. However, this possibility has not been investigated to date. It is also possible to use different numerical integration rules for each equation, although this cannot be theoretically justified (as in the Galerkin penalty formulation). Our numerical experiments indicate that the use of this procedure greatly affects the quality of the numerical results.

It should be noted that the equation system (6) is symmetric positive-definite and therefore iterative solution techniques can be used. We describe in the next section an incomplete factorization conjugate gradient algorithm.

Numerical results for the incompressible Navier–Stokes equations were reported in References 1 and 2 using a least square formulation and bilinear quadrilateral elements (with 1 integration point). The element chosen here is the quadratic triangle which should perform well to represent boundary layers. An example using linear triangles is also shown later.

Incomplete Choleski conjugate gradient algorithm

An incomplete Choleski conjugate gradient algorithm was used to solve the equation system (6). For large problems, the memory required to perform the complete factorization of K can become prohibitively large. It is therefore advantageous to perform only a partial elimination of K and use this approximate factorization as a preconditioner in the conjugate gradient algorithm. Kershaw[16] proposed an incomplete factorization which discards coefficients that are outside a pre-specified storage area. The technique advocated here was proposed by Ajiz and Jennings[17] and only discards coefficients that fall below a pre-specified level. Since each coefficient is compared to the diagonal terms present at the time, this criterion is not ideal as it does not necessarily reflect the importance of this coefficient. However, this would only be possible if a complete factorization was performed which is precisely what we try to avoid.

The main elements of the algorithm are briefly described below. The coefficient matrix K is factorized according to:

$$K=LL^{\mathrm{T}}-C \tag{11}$$

where L is a lower triangular incomplete Choleski factorization and C is the matrix of coefficients rejected. If the coefficients of L are known for columns 1 to $i-1$ we can compute:

$$l_{ii}l_{ji}=k_{ji}-\sum_{k=1}^{i-1} l_{ik}l_{jk} \tag{12}$$

Off diagonal coefficients are rejected if:

$$l_{ji}^2<\psi^2\left(k_{jj}-\sum_{k=1}^{i-1} l_{jk}l_{jk}\right) \tag{13}$$

where ψ is a specified parameter ranging from 0 to 1. The choice $\psi=0$ corresponds to the complete Choleski factorization.

This technique was found to be very efficient. However, it sometimes led to the loss of positive definiteness and therefore the diagonal scaling advocated by Ajiz and Jennings[17] was also implemented in the code. This scaling ensures that the positive definiteness is conserved and that the factorization will not fail.

For practical computations, the arrays K and L are stored as vectors which contain only the non-zero entries but this requires the addition of some integer vectors to identify the address of each coefficient. The memory required to store the array K is relatively small and it is therefore possible to tackle very large problems if we use larger values of ψ. However, the quality of the preconditioner deteriorates and a larger number of conjugate gradient iterations might be necessary in such cases. In most applications, the technique described here was found to be both efficient and accurate.

Once the incomplete factorization is obtained, the equation system is modified as:

$$L^{-1}KL^{-\mathrm{T}}y=b \tag{14}$$

where $y=L^{\mathrm{T}}U^{n+1}$ and $b=L^{-1}f$.

This system is solved by means of the conjugate gradient method for increments $\Delta U=U^{n+1}-U^n$. Starting from:

$$y=0 \quad \text{and} \quad d^0=r^0=L^{-1}(F-KU^n) \tag{15}$$

we iterate according to:

$$\alpha_k=(r^k)^{\mathrm{T}}r^k/(d^k)^{\mathrm{T}}(L^{-1}KL^{-\mathrm{T}})d^k \tag{16}$$

$$y^{k+1}=y^k+\alpha_k d^k \tag{17}$$

$$r^{k+1}=r^k-\alpha_k(L^{-1}KL^{-\mathrm{T}})d^k \tag{18}$$

$$\beta_k=(r^{k+1})^{\mathrm{T}}r^{k+1}/(r^k)^{\mathrm{T}}r^k \tag{19}$$

$$d^{k+1}=r^{k+1}+\beta_k d^k \tag{20}$$

The node numbering is important since the computation of the preconditioning matrix is based on factorization. A Cuthill–Mackee algorithm[18] has been used to reduce the bandwidth of the K matrix in the computations reported here.

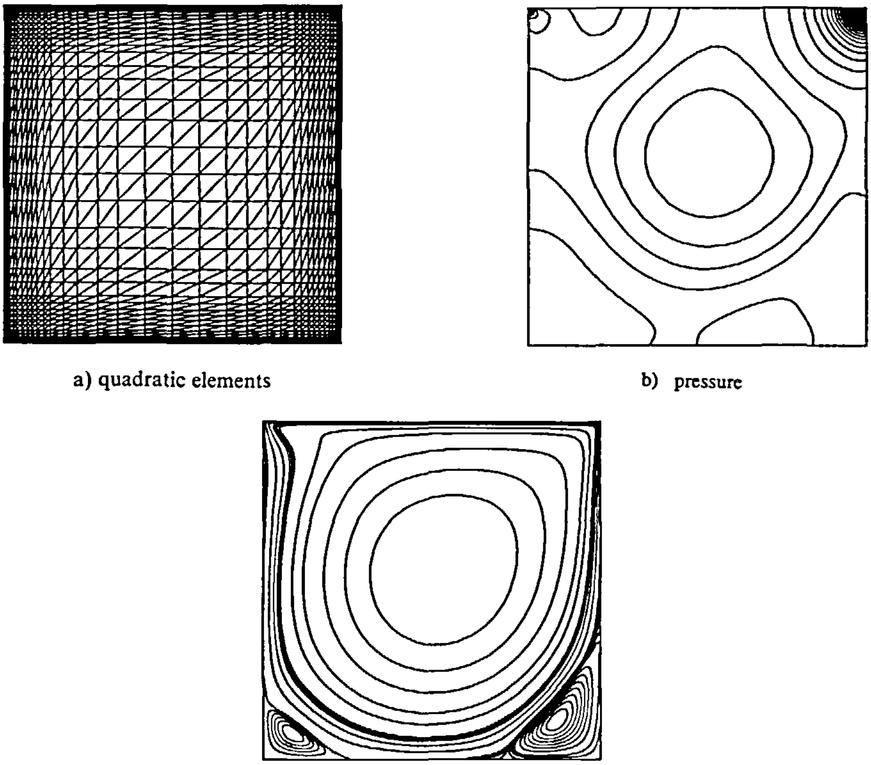
The costly part of the iterative solver is the computation of the incomplete Choleski factorization matrix L. To reduce this cost, the same matrix is used for several iterations when the convergence of the non-linear iteration has reached a certain preset value.

In practical computations and in most of the problems presented in the next section, small values of ψ (i.e. $0.01 < \psi < 0.1$) are used. It is found that the use of 10 to 30 conjugate gradient iterations ((16) to (20)) is generally sufficient at each non-linear step.

Numerical examples

Driven cavity flow. This is a well-known test example[26]. The Reynolds number in this case is 10^3 and the domain was discretized using a regular 40 × 40 mesh of quadratic triangular elements shown in *Figure 1a*. The pressure contours and flow pattern are displayed in *Figures 1b* and *1c* respectively. It can be observed that the predictions of the least squares finite element method compare favourably with available published data[26,28].

Inviscid flow past a NACA0012 airfoil. This second example is used to compare the effect of using linear and quadratic approximations. The inviscid condition is simply represented by eliminating the vorticity from the two momentum equations. The tangency condition along the airfoil is weakly imposed in a least square sense by simply adding a boundary integral term to enforce zero normal velocity.

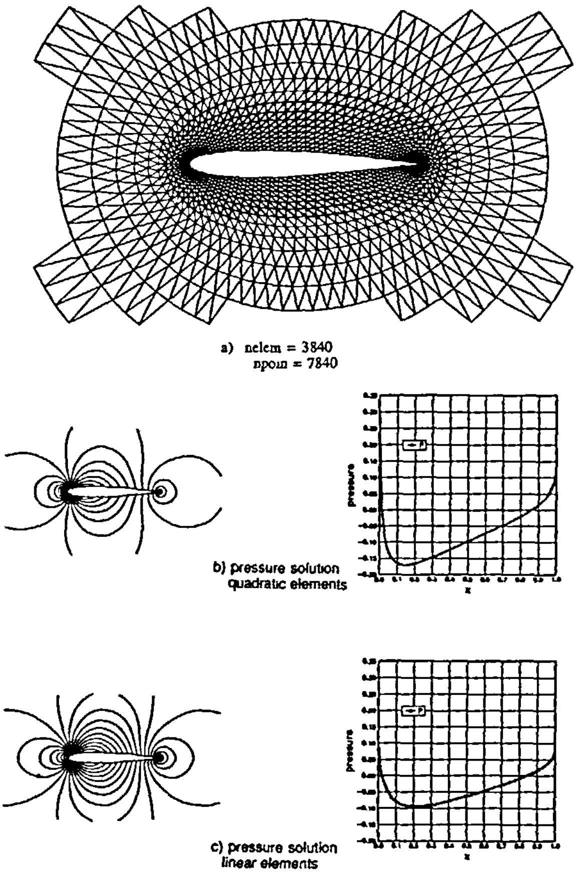


a) quadratic elements

b) pressure

c) streamlines

Figure 1 Driven cavity flow. $Re = 1000$. (a) 40×40 mesh of quadratic elements; (b) pressure solution; (c) selected streamlines

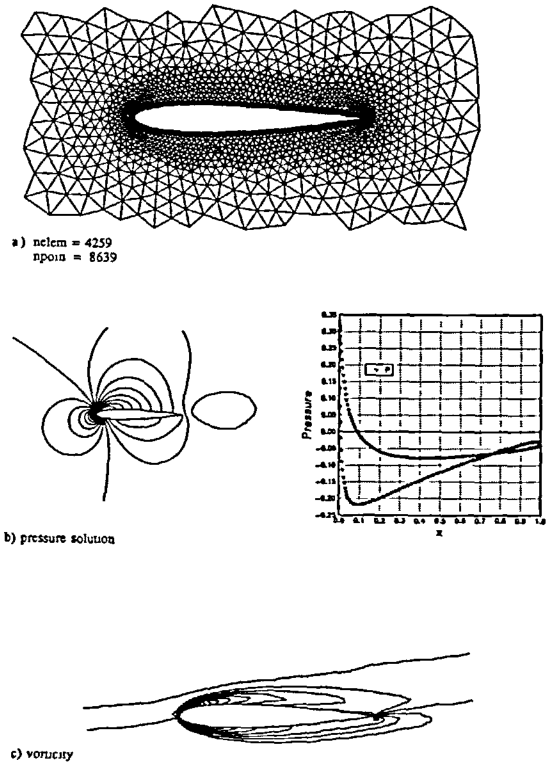


a) nelem = 3840
npoin = 7840

b) pressure solution
quadratic elements

c) pressure solution
linear elements

Figure 2 Inviscid incompressible flow past a NACA0012 airfoil. (a) Detail of the mesh around the airfoil (quadratic elements); (b) pressure contours and distribution on the airfoil obtained with quadratic triangular elements, and (c) pressure contours and distribution on the airfoil obtained with linear triangular elements



a) nelem = 4259
npoin = 8639

b) pressure solution

c) vorticity

Figure 3 Viscous flow past a NACA0012 airfoil at incidence. $Re = 1000$. Angle of attack = 10°. (a) Detail of the mesh near the airfoil; (b) pressure contours and distribution on the airfoil; and (c) vorticity

The mesh of quadratic elements is displayed in *Figure 2a* and consists of 3840 elements and 7840 points. A mesh of linear triangles was obtained by subdividing each quadratic element into four linear elements, thus retaining the same number of points.

The pressure solutions obtained are shown in *Figures 2b* and *2c*. The comparison of the results obtained with the two approximations shows that the linear element is far less accurate than the quadratic element near the leading edge. This difference can probably be attributed to the inability of the linear approximation to properly represent the incompressibility condition.

Viscous flow past a NACA0012 airfoil at incidence. This example involves the simulation of a viscous flow ($Re = 10^3$) past an airfoil at an angle of attack of 10°. The mesh consists of 4259 quadratic elements and 8639 points and is displayed in *Figure 3a.* Note that a structured local mesh consisting of four layers of elements has been added around the airfoil to aid the capture of the boundary layer. The pressure contours and pressure distribution over the airfoil are shown in *Figure 3b.* The vorticity contours are shown in *Figure 3c.*

LEAST SQUARES FINITE ELEMENT SOLUTION OF COMPRESSIBLE FLOWS

Least squares finite element formulations have already been applied with some success for the Euler equations[2,10,11,19]. Jiang *et al.*[2] propose an implicit time differenced least squares formulation for the Euler equations written in terms of non-conservative variables. Numerical examples are included which show the good shock capturing properties of the scheme. Bruneau develops[10] a least squares finite element method which directly solves the steady Euler equations using Newton iterations. Again good shock capturing properties are illustrated. These ideas are extended in this section and applied to the solution of the iso-enthalpic Euler equations written in conservative form. The formulation is based on the minimization of the L2 norm of the time differenced equations. The scheme is implicit, unconditionally stable and the results indicate that the numerical diffusion of the scheme is sufficient to solve shocked flows using linear triangular elements. The norm associated with the least squares formulation provides a natural error norm which is used for mesh refinement.

Formulation

The formulation proposed here considers the Euler equations in conservative form in which the unsteady energy equation has been replaced by the condition of constant enthalpy, which holds at steady state. In 2 dimensions, the resulting hyperbolic system can be expressed as:

$$\frac{\partial U}{\partial t}+\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}=0 \tag{21}$$

where

$$U=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\end{Bmatrix} \quad f=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\end{Bmatrix} \quad g=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\end{Bmatrix} \tag{22}$$

Here ρ, p, u and v denote respectively the density, pressure and velocity components. The pressure is calculated according to the relationship

$$p=\frac{\gamma-1}{\gamma}\rho\left(H_0-\frac{u^2+v^2}{2}\right) \tag{23}$$

where γ is the ratio of specific heats and H_0 is the constant enthalpy calculated from the free stream conditions. Note that the equations are written in a non-dimensional form.

A simple backward difference procedure is used to step forward in time and (21) is replaced by:

$$\frac{\Delta U}{\Delta t}+A^n\frac{\partial \Delta U}{\partial x}+B^n\frac{\partial \Delta U}{\partial y}=-\left(\frac{\partial f^n}{\partial x}+\frac{\partial g^n}{\partial y}\right) \tag{24}$$

where $\Delta U=U^{n+1}-U^n$, Δt is the increment in time and A^n and B^n are the Jacobian matrices which have been linearized in time by setting $A^n=A(U^n)$ and $B^n=B(U^n)$.

The basic least squares method used in the previous section is now applied to minimize the L2 norm of (24) with respect to the increments ΔU. If the domain is discretized using 3-noded triangular elements with piecewise linear interpolations for ΔU in the form:

$$\Delta U=N_i\Delta U_i \tag{25}$$

the least squares finite element formulation becomes

$$K_{ij}\Delta U_j=F_i \tag{26}$$

where

$$K_{ij}=\int\left[N_i+\Delta t\left(A^n\frac{\partial N_i}{\partial x}+B^n\frac{\partial N_i}{\partial y}\right)\right]^{\mathrm{T}}\left[N_j+\Delta t\left(A^n\frac{\partial N_j}{\partial x}+B^n\frac{\partial N_j}{\partial y}\right)\right]\mathrm{d}\Omega \tag{27}$$

and

$$F_i=-\int\Delta t\left[N_i+\Delta t\left(A^n\frac{\partial N_i}{\partial x}+B^n\frac{\partial N_i}{\partial y}\right)\right]^{\mathrm{T}}\left[\frac{\partial f^n}{\partial x}+\frac{\partial g^n}{\partial y}\right]\mathrm{d}\Omega \tag{28}$$

The Jacobian matrices A^n and B^n are assumed to be constant over each element and the fluxes f^n and g^n are computed at the nodes and vary linearly over each element according to:

$$f^n=N_i f(U_i^n) \qquad \text{and} \qquad g^n=N_i g(U_i^n) \tag{29}$$

The right hand side, which should vanish at steady state, is now modified as:

$$F_i=-\int\Delta t N_i^{\mathrm{T}}\left[\frac{\partial f^n}{\partial x}+\frac{\partial g^n}{\partial y}\right]\mathrm{d}\Omega-\int\Delta t^2\left[A^n\frac{\partial N_i}{\partial x}+B^n\frac{\partial N_i}{\partial y}\right]^{\mathrm{T}}\left[A^n\frac{\partial N_j}{\partial x}+B^n\frac{\partial N_j}{\partial y})\right]\mathrm{d}\Omega\, U_j^n \tag{30}$$

for reasons which will be explained in the next section.

The second term in (30) represents the internal dissipation of the least squares formulation and is proportional to Δt. Therefore, the steady solution will be dependent on Δt and it is very important to use local time steps, not only to enhance the convergence but also to avoid oscillations that would occur if the CFL number is too small somewhere in the mesh. The time step is calculated at each node according to:

$$\Delta t=CFL\cdot\frac{h}{\lambda_{\max}} \tag{31}$$

where h is the minimum size of all the elements surrounding the node, $\lambda_{\max}$ is the maximum eigenvalue of the present hyperbolic system (see for example Reference (20)) and CFL is a user prescribed constant.

With the approximation adopted here and the use of linear triangular elements, numerical integration is not needed. The formulation can, however, be easily extended to higher order elements[10]. The incomplete Choleski-conjugate gradient solver described is used to solve the linear system at each time step. For small CFL numbers, more efficient iterative solvers could be used due to the diagonal dominance of K.

It should be noted that, if the Jacobian matrices are symmetric (i.e. written in terms of entropy variables[21,22]), the scheme described here becomes similar to the centred implicit scheme of Lerat[23].

Adaptive refinement strategy based on least squares residuals

We recall that the second term on the right hand side of (30) was modified and written as:

$$\int\Delta t^2\left[A^n\frac{\partial N_i}{\partial x}+B^n\frac{\partial N_i}{\partial y}\right]^{\mathrm{T}}\left[A^n\frac{\partial N_j}{\partial x}+B^n\frac{\partial N_j}{\partial y})\right]\mathrm{d}\Omega\, U_j^n \tag{32}$$

This modification was motivated by the fact that (32) represents the least squares residuals of the steady Euler equations. This naturally defines a norm and a measure of the error which can be used for adaptive refinement[25]. Note also that since this term effectively represents the numerical diffusion of the scheme, it can be anticipated that the error thus defined will be important across shocks or more generally in regions of large variations.

We define the error for each element as:

$$E_i=\frac{1}{\Omega_e}\int\left[A^n\frac{\partial N_i}{\partial x}+B^n\frac{\partial N_i}{\partial y}\right]^{\mathrm{T}}\left[A^n\frac{\partial N_j}{\partial x}+B^n\frac{\partial N_j}{\partial y}\right]\mathrm{d}\Omega\, U_j^n \tag{33}$$

and the error indicator as:

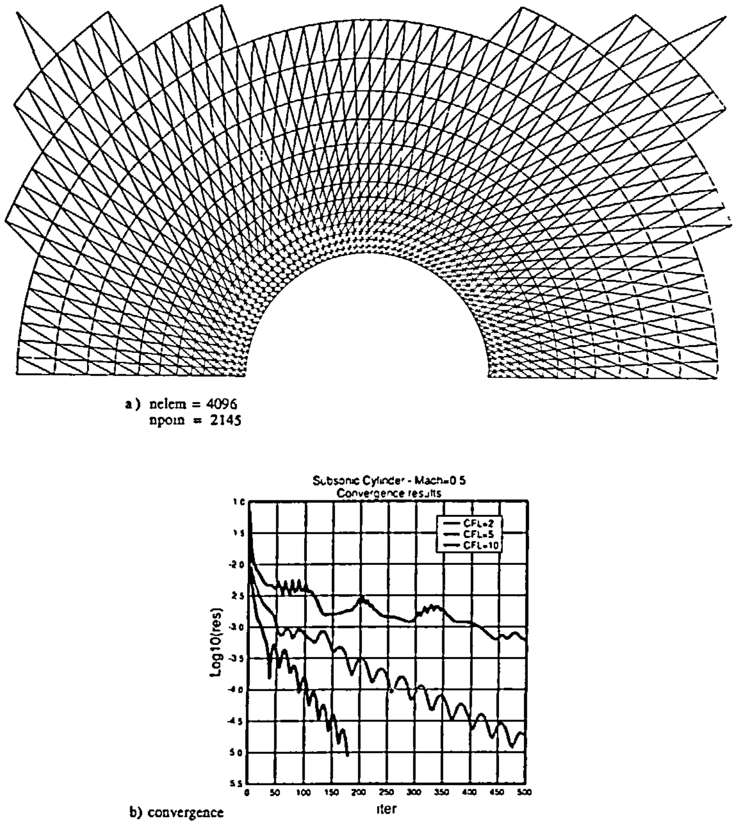
$$\beta = \frac{E_l}{\max E_l} \tag{34}$$

The refinement stretegy is then to refine all these elements for which the indicator is larger than a certain specified value. Let Ω_R denote the region of elements for which refinement is needed. Each triangle of Ω_R is subdivided into four smaller triangles and, to avoid the problem of 'hanging nodes', transition elements are introduced. If a transition element has one side along the boundary of Ω_R it is subdivided into 2. If a transition element has two sides along the boundary of Ω_R, it is subdivided into 3.

Various mesh enhancement techniques (i.e. to avoid badly deformed elements) can also be added. These techniques are described in detail in Reference 24.

Numerical examples

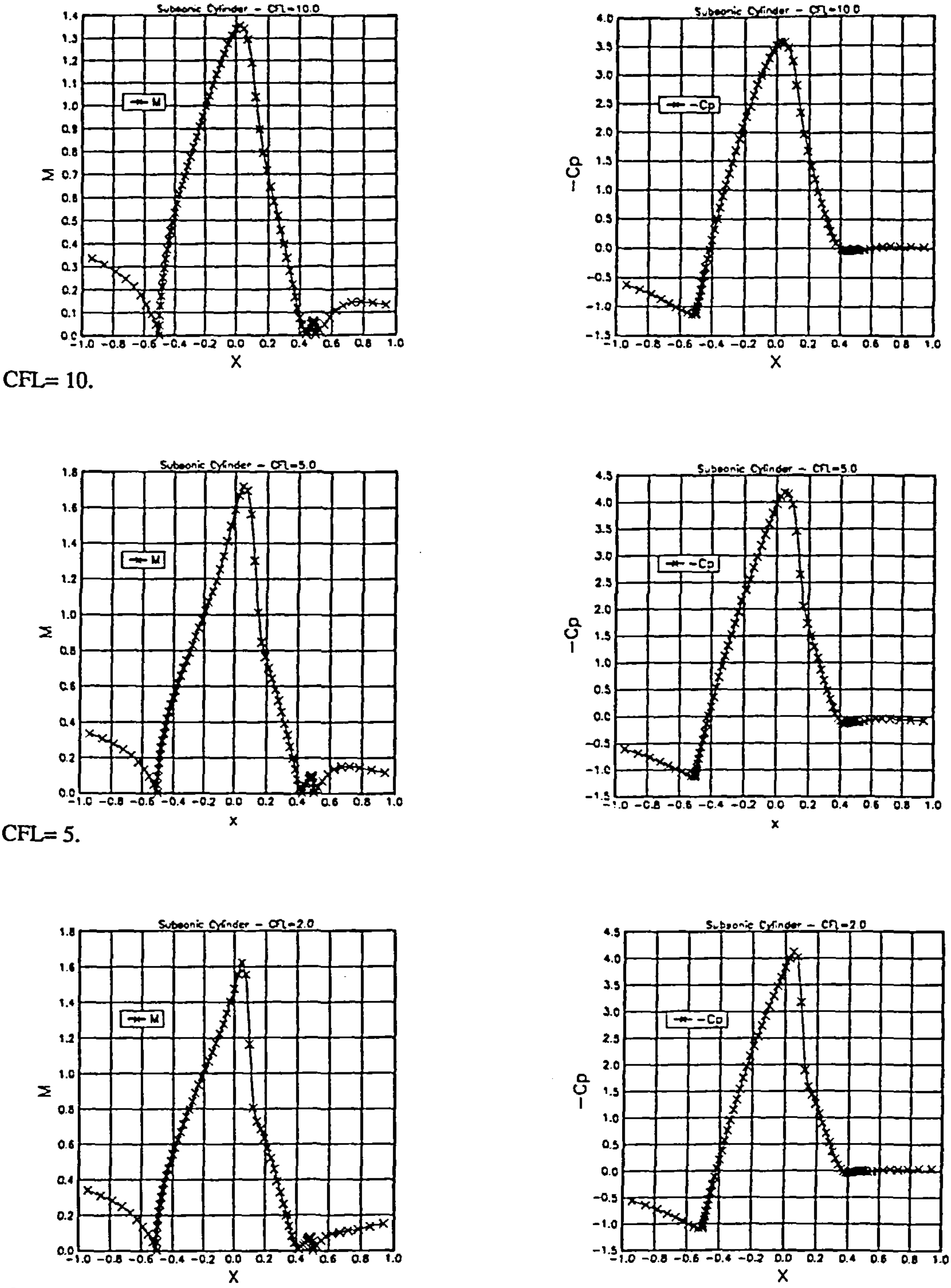
Subsonic flow past a cylinder. The first example is a case of subsonic flow past a cylinder. The free stream Mach number is 0.5. Due to the symmetry, only half of the cylinder is considered. A detail of the mesh of linear triangular elements is displayed in *Figure 4a.* The global mesh consists of 4096 elements and 2145 points and the outer boundary is situated at approximately 20 diameters from the cylinder. As in the inviscid incompressible case, the tangency condition



a) nelem = 4096
npoin = 2145

b) convergence

Figure 4 Subsonic inviscid flow past a cylinder (Mach=0.5). (a) Detail of the mesh; (b) convergence of the L2 norm of the time derivative of density for various *CFL* numbers;



c) CFL= 10.

d) CFL= 5.

e) CFL= 2.

Figure 4 (*continued*) Subsonic inviscid flow past a cylinder (Mach=0.5). Mach number and pressure coefficient distribution for (c) *CFL*=10, (d) *CFL*=5, and (e) *CFL*=2 is imposed weakly via the least squares functional and the pressure is corrected to obtain a conservative approximation of the normal momentum equation.

The main purpose of this first example is to illustrate the influence of the choice of the *CFL* number on accuracy and convergence. *Figure 4b* illustrates the convergence of the L2 norm of the time derivative of the density for various *CFL* numbers. As expected, the convergence rate improves with increasing *CFL* numbers. The oscillations in the convergence history are due to the variations of the local time steps. The influence of the *CFL* number on the computed solution is illustrated in *Figures 4c*, *d* and *e*. For *CFL*=2, the shock is well represented but oscillations before the shock are appearing. As the *CFL* number increases, the shock is diffused and has almost disappeared for a *CFL* number of 10.

Supersonic flow past a cylinder and mesh refinement. The next example consists of a supersonic flow past a cylinder and illustrates the refinement technique described above. The free stream Mach number is 3. Due to the symmetry, only half of the cylinder is considered. The first mesh used is a structured polar mesh which consists of 4096 elements and 2145 points (*Figure 5a*). The time step is chosen as to obtain a uniform *CFL* number of 5. The pressure solution obtained with this mesh is displayed in *Figure 5d.*

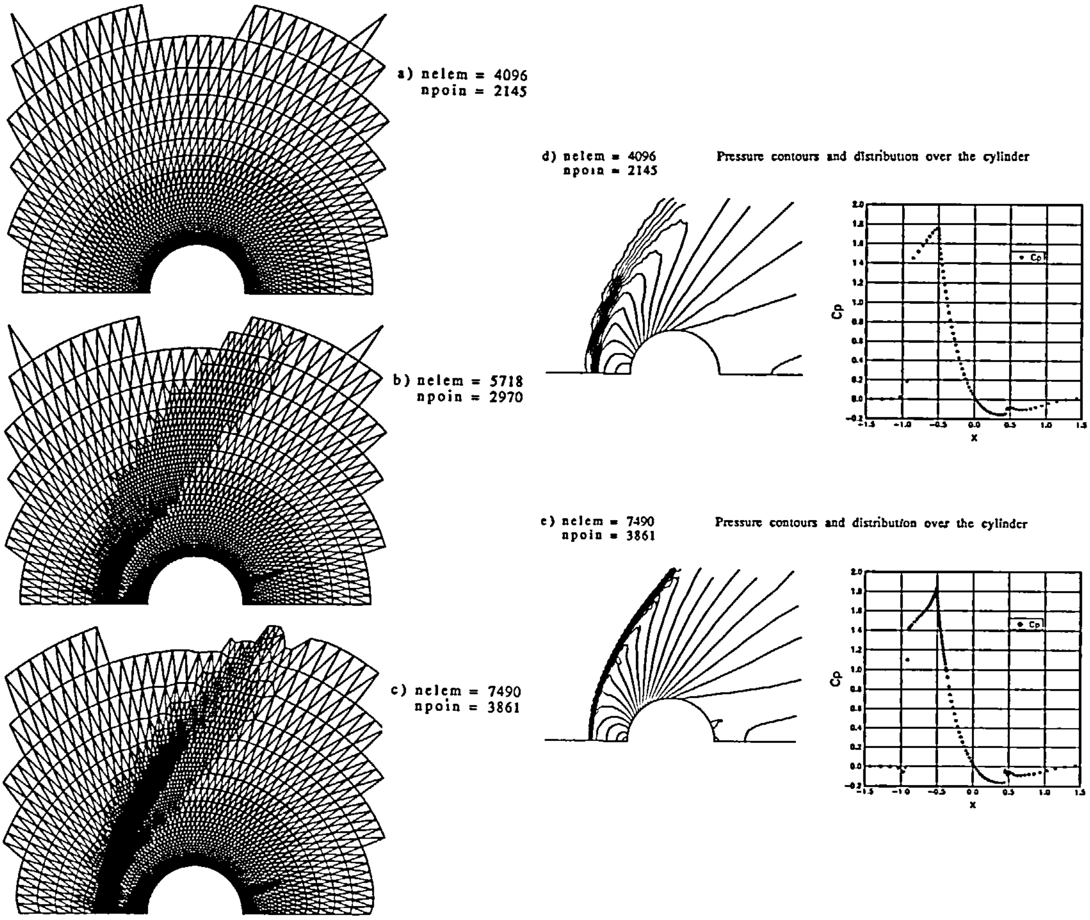
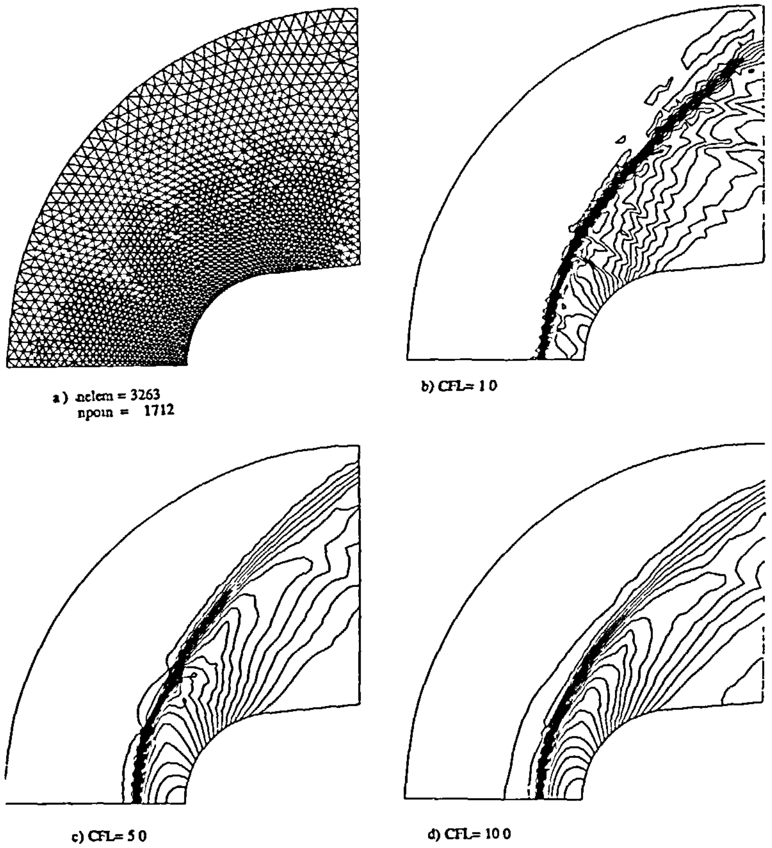


Figure 5 Supersonic flow past a cylinder (Mach=3; *CFL*=5). (a) Initial mesh; (b) first mesh refinement; (c) second mesh refinement; (d) pressure solution obtained with the initial mesh; (e) pressure solution obtained with the final mesh



a) nelem = 3263
npoin = 1712

b) CFL= 1 0

c) CFL= 5 0

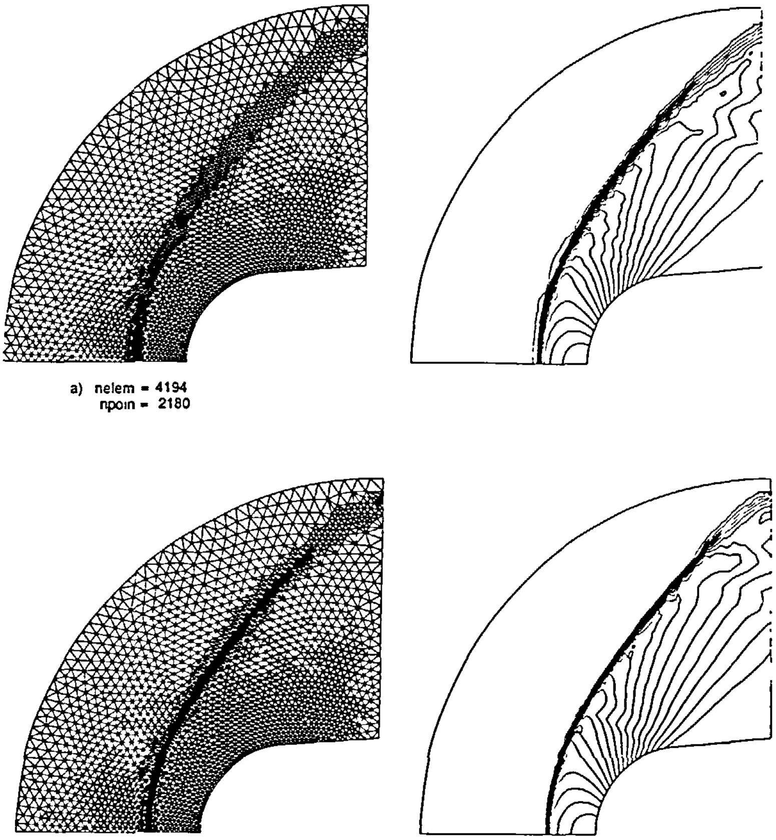
d) CFL= 10 0

Figure 6 Supersonic flow past a blunt body (Mach = 6.57). (a) Discretization of the domain and pressure solution for (b) *CFL* = 1, (c) *CFL* = 5, and (d) *CFL* = 10

Two successive mesh refinements are performed from this initial solution. The first mesh refinement is illustrated in *Figure 5b*. The new mesh consists of 5718 elements and 2970 points. It can be observed that the error indicator is not only active across the bow shock but also near the stagnation point and across the rear shock. This is further illustrated by a second mesh refinement depicted in *Figure 5c*. The new mesh consists of 7490 elements and 3861 points. The mesh 'smoothing' technique described in Reference 24 was used for this mesh to remove badly deformed elements. The pressure solution obtained with this mesh is displayed in *Figure 5e* and the convergence was achieved after less than 200 time steps starting from the solution obtained with the previous mesh.

Supersonic blunt body and mesh refinement. In this example, we illustrate the capability of the least squares method to capture strong shocks without the addition of artificial viscosity. The test problem is that of a supersonic flow past a blunt body. The free stream Mach number is 6.57. The geometry and discretization of the domain is shown in *Figure 6a*. The mesh consists of 3263 elements and 1712 points. Free stream conditions are imposed at the inlet and the outlet boundary is left free.

The influence of the choice of the *CFL* number is illustrated in *Figures 6b*, *c* and *d*. For *CFL* = 1, the solution is oscillatory and the convergence slow. These oscillations disappear for higher *CFL* numbers. Note that, for *CFL* = 5, the shock is captured within two elements.



a) nelem = 4194
npoin = 2180

b) nelem = 5291
npoin = 2730

Figure 7 Supersonic flow past a blunt body (Mach = 6.57) and mesh enrichment (a) First mesh refinement and pressure solution; (b) second mesh refinement and pressure solution

Figure 7 illustrates the mesh refinement strategy described earlier. The solutions were obtained using a *CFL* number of 5. A first refinement is shown in *Figure 7a*. The new mesh consists of 4194 elements and 2180 points and it can be observed that the error indicator is only active along the shock. This is further illustrated by the second refinement shown in *Figure 7b*. The new mesh now consists of 5291 elements and 2730 points and the improvement of the computed solution is evident.

Transonic flow past a NACA0012 at incidence. As a last example, we consider the problem of a transonic flow past a NACA0012 airfoil with an angle of attack of 1 degree. The free stream Mach number is 0.85. The first mesh is depicted in *Figure 8a* and consists of 3840 elements and 2000 points. The first solution was obtained using a *CFL* number of 5.

A refinement is illustrated in *Figure 8b*. The new mesh consists of 5740 elements and 2969 points. It should be noted that in this case the error indicator is not only active across the shocks but also near the leading and trailing edge where the resolution was insufficient in the first mesh. The pressure and Mach number solutions obtained with this second mesh are shown in *Figures 8c* and *d* respectively. It should be noted that a thin boundary layer is present on the upper and lower surfaces of the airfoil. This explains the small discrepancies in the shock locations when compared to highly accurate solutions of this test problem[27].

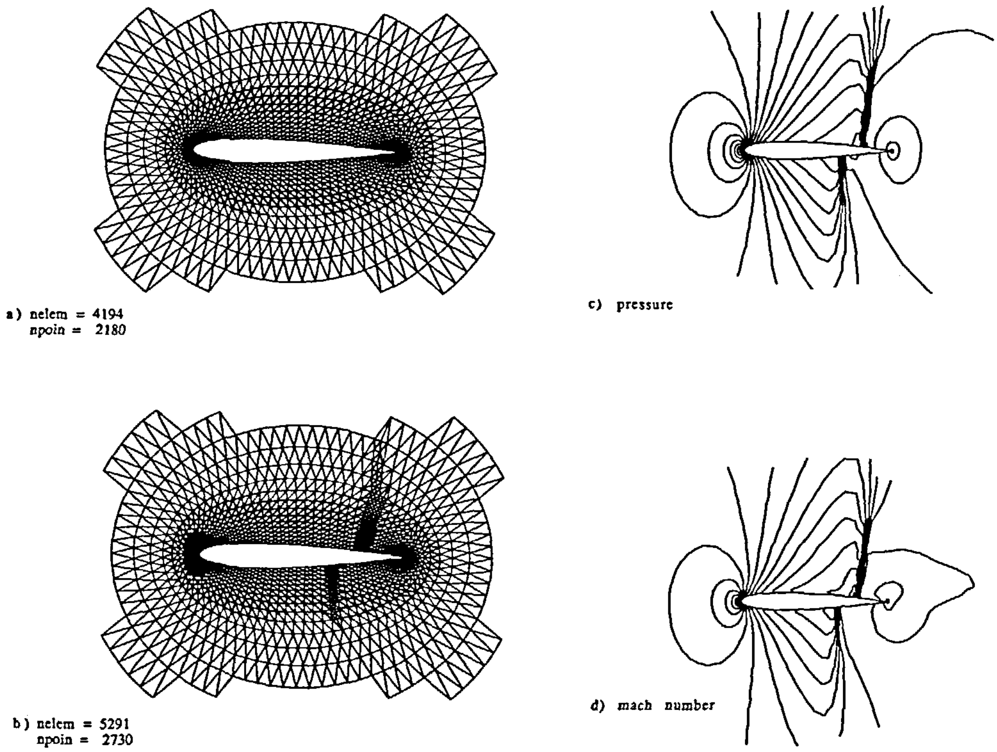


Figure 8 Transonic flow past a NACA0012 airfoil (angle of attack = 1°, Mach = 0.85). (a) Initial mesh; (b) first mesh refinement; (c) pressure contours ($CFL = 5$); (d) Mach number contours ($CFL = 5$)

CONCLUSIONS

We have presented some early results obtained with a least squares finite element method using triangular elements and unstructured meshes. The method is both simple and accurate. In particular, a number of complex compressible flow problems have been solved using the least squares formulation without any added artificial dissipation and improved solutions have been obtained with the refinement technique based on the least squares residuals.

We believe the method has some valuable merits and we are currently working on its extension to solve the full Euler system and also the compressible Navier–Stokes equations. Another possibility we are investigating, is the use of higher order elements. Early experiments with the full Euler system revealed that the method can become unstable in complex flow situations. The same observation was made with higher order elements. Our work will be directed towards improving the efficiency of the formulation in such cases.

ACKNOWLEDGEMENTS

We would like to acknowledge the support of Dassault Aviation (France) which enabled this work to be performed.

REFERENCES

1 Jiang, B. N. Least-squares finite element methods with element-by-element adaptive refinement, *PhD Dissertation*, University of Texas, Austin (1986)

2 Jiang, B. N. and Povinelli, L. A. Least squares finite element method for fluid dynamics, *NASA TM 102352-ICOMP-89-23* (1989)

3 Babuska, I. The finite element method with Lagrange multipliers, *Num. Math.*, **20**, 179–192 (1973)

4 Brezzi, F. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers, *R.A.I.R.O.*, **8**, 129–151 (1974)

5 Zienkiewicz, O. C. and Taylor, R. L. *The Finite Element Method*, 4th Edn, Vol. 1, Ch. 12, McGraw-Hill, New York (1989)

6 Brooks, A. N. and Hughes, T. J. R. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, *Comp. Meth. Appl. Mech. Eng.*, **32**, 199–259 (1982)

7 Hughes, T. J. R., Franca, L. P. and Balestra, M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal order interpolations, *Comp. Meth. Appl. Mech. Eng.*, **59**, 85–99 (1986)

8 de Sampaio, P. A. B. A Petrov–Galerkin formulation for the incompressible Navier–Stokes equations using equal order interpolation for velocity and pressure, *Int. J. Num. Meth. Eng.* in press

9 Jiang, B. N. A least squares finite element method for incompressible Navier–Stokes problems, *NASA TM 102385-ICOMP-89-28* (1989)

10 Bruneau, C. H. Computation of hypersonic flows by a finite element least squares method, *Proc. 7th Int. Conf. Finite Element Methods in Fluids*, University of Alabama Press, Hunstville, pp. 757–762 (1989)

11 Bruneau, C. H., Laminie, J. and Chattot, J. J., Computation of 3D vortex flows past a flat plate at incidence through a variational approach of the full steady Euler equations, *Int. J. Num. Meth. Fluids*, **9**, 305–323 (1989)

12 Morgan, K. and Peraire, J. Finite element methods for compressible flows, *Von Karman Inst. Fluid Dynamics, Lecture Series 1987–04* (1987)

13 Hughes, T. J. R. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier–Stokes equations, *Int. J. Num. Meth. Fluids*, **7**, 1261–1275 (1987)

14 Stoufflet, B., Periaux, J., Fezoui, F. and Dervieux, A. Numerical simulation of 3-D hypersonic Euler flows around space vehicles using adapted finite elements, *AIAA paper 87-0560* (1987)

15 Peraire, J., Peiro, J., Formaggia, L., Morgan, K. and Zienkiewicz, O. C. Finite element Euler computations in three dimensions, *AIAA paper 88-0032* (1988)

16 Kershaw, D. S. The incomplete Choleski–conjugate gradient method for the iterative solution of systems of linear equations, *J. Comp. Phys.* **26**, 43–65 (1968)

17 Ajiz, M. A. and Jennings, A. A robust incomplete Choleski–conjugate gradient algorithm, *Int. J. Num. Meth. Eng.*, **20**, 949–966 (1984)

18 Cuthill, E. H. and Mckee, J. M. Reducing the bandwidth of sparse symmetric matrices, *Sparse Matrices and their Applications* (Eds. D. J. Rose and R. A. Willoughby), Plenum Press, New York, pp 157–160 (1972)

19 Jiang, B. N. and Carey, G. F. A stable least squares finite element method for non-linear hyperbolic problems, *Int. J. Num. Meth. Fluids*, **8**, 933–942 (1988)

20 Hirsch, C. *Numerical Computation of Internal and External Flows*, Vol. 2, Wiley, New York (1990)

21 Hughes, T. J. R., Franca, L. P. and Mallet, M. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics, *Comp. Meth. Appl. Mech. Eng.*, **59**, 223–231 (1986)

22 Harten, A. On the symmetric form of systems of conservation laws with entropy, *J. Comp. Phys.*, **49**, 151–164 (1983)

23 Lerat, A. and Sides, J. Efficient solution of the steady Euler equations with a centered implicit method, *Proc. Int. Conf. Num. Meth. Fluid Dynamics, Oxford* (1988)

24 Peraire, J. A finite element method of convection dominated flows, *PhD Dissertation*, University of Wales, Swansea (1986)

25 Jiang, B. N. and Carey, G. F. Adaptive refinement for least squares finite elements with element-by-element conjugate gradient solution, *Int. J. Num. Meth. Eng.* **24**, 569–580 (1987)

26 Ghia, U., Ghia, K. N. and Shin, C. T. High-Re solutions for incompressible flowusing the Navier–Stokes equations and a multigrid method, *J. Comp. Phys.*, **48**, 387–411 (1982)

27 Pulliam, T. H. and Barton, J. T. Euler computations of AGARD working group 07. Airfoil test cases, *AIAA paper 85-0018* (1985)

28 Bruneau, C. H. and Jouron, C. An efficient scheme for solving steady incompressible Navier–Stokes equations, *J. Comp. Phys.*, **89**, 389–413 (1990)