A DIRECTIONALLY ADAPTIVE METHODOLOGY USING AN EDGE-BASED ERROR ESTIMATE ON QUADRILATERAL GRIDS

D. AIT-ALI-YAHIA, W. G. HABASHI AND A. TAM

CFD Laboratory, Department of Mechanical Engineering, Concordia University, 1455 de Maisonneuve Blvd. W, ER 301, Montreal, Quebec, Canada H3G 1M8

M.-G. VALLET

Institut des Matériaux Industriels, 75 Boulevard de Mortagne, Boucherville, Quebec, Canada J4B 6Y4

and

M. FORTIN

Département de Mathématiques et de Statistique, Université Laval, Sainte-Foy, Quebec, Canada G1K 7P4

SUMMARY

The present paper describes a directionally adaptive finite element method for high-speed flows, using an edge-based error estimate on quadrilateral grids. The error of the numerical solution is estimated through its second derivatives and the resulting Hessian tensor is used to define a Riemannian metric. An improved mesh movement strategy, based on a spring analogy, but with no orthogonality constraints, is introduced to equidistribute the lengths of the edges of the elements in the defined metric. The grid adaptation procedure is validated on an analytical test case and the efficiency of the overall methodology is investigated on supersonic and hypersonic benchmarks.

KEY WORDS: Euler equations; directionally adaptive meshes; edge-based error estimate; structured grids; mesh movement; finite element method; high-speed flows

1. INTRODUCTION

The current interest in the area of high-speed flows has increased the need for advanced computational fluid dynamics (CFD) codes, which have become the primary tools for the prediction of aerothermal loads. Such flows are characterized by regions with steep directional gradients of flow variables, embedded in regions where the flow variables vary more smoothly. One approach for improving the solution accuracy of such problems is to apply grid adaptation techniques.

Grid adaptation methods are mainly composed of an error estimate and an adaptive strategy such as mesh movement (r-method), mesh refinement/coarsening (h-method), higher-order interpolation (p-method) or remeshing. Even when the same error estimate is used to assess the accuracy of a solution, the resulting adapted grid depends strongly on the selected adaptation strategy. Classical methods such as grid refinement[1,2] produce isotropic meshes in which the length scales of each element are essentially the same. These methods are optimal only for those flow fields regions possessing nearly

CCC 0271–2091/96/070673–18

Received July 1995

equal gradients in all spatial directions. As a result, directional flow features such as shocks, contact discontinuities and boundary layers are not necessarily adapted efficiently and the number of elements needed to represent them may increase disproportionally with each isotropic refinement.

An alternative approach would be to seek solutions on anisotropic meshes where more resolution is introduced along those directions with rapidly changing flow variables. This idea was introduced by Peraire *et al.*,[3] who used an adaptive remeshing procedure that incorporated directional stretching for the solution of the 2D Euler equations on triangular grids. Anisotropic grids may also be produced by coupling a mesh movement strategy with local isotropic refinement.[4] Kornhuber and Roitzsch[5] proposed an anisotropic strategy based on directed refinement of pairs of triangular elements to resolve boundary layers. Recently Fortin *et al.*[6] used a metric as a measure of error, coupled to an *h–r* strategy, to achieve directionally adapted unstructured grids with high aspect ratios.

The above approaches have primarily been used on unstructured meshes. This trend is mainly driven by the intrinsic ability of triangular elements in 2D and tetrahedral elements in 3D to deal with arbitrary complex geometries. In addition, such meshes provide a natural setting for the implementation of adaptive grid techniques. Unstructured adaptation algorithms can yield highly stretched grids as well as locally refined/coarsened meshes. In contrast, most refinement techniques for structured grids[2,7,8] avoid propagating the refinement to the boundaries by allowing sides to have hanging nodes.

Despite thesee advantages for unstructured meshes, structured grids of quadrilateral elements in 2D and hexahedral elements in 3D are still used with great success in CFD.[8-10] One reason is their ability to include multigrid acceleration techniques in a straightforward manner, while unstructured grids may encounter serious difficulties in generating multilevel grids.[11,12] Moreover, integration of the governing equations on a structured grid requires less CPU time than on an unstructured one for the same number of nodes. Structured grids are also more suitable for turbulence modelling, particularly near solid walls where normals to the wall may be necessary.

Furthermore, a certain degree of grid anisotropy may also be introduced for structured grids through an improved moving-node scheme to be presented in this paper. A moving-node technique was originally introduced by Gnoffo,[13] generalized by Nakahashi and Deiwert[14] in the context of finite volume methods and applied to a finite element method (FEM) by Löhner *et al.*[1] All these schemes are based on a spring analogy where the grid is viewed as a network of springs whose stiffness constants represent a measure of error. The grid vertices are displaced until the equilibrium state of the spring forces is reached. Such techniques are characterized by their low cost and the conservation of nodal connectivity, but often can stall or diverge and tolerate only a limited range of nodal movement.

This paper describes a directionally adaptive FEM using an edge-based error estimate on quadrilateral grids. The use of an appropriate error estimate, combined with the vector nature of spring forces (i.e. their magnitude and direction), permits one to design a convergent adaptive procedure capable of achieving wider nodal movement and a high degree of grid anisotropy. The error of the numerical solution is evaluated using a bound available from finite element interpolation theory. The Hessian of a selected solution variable is computed and then modified to produce a positive definite matrix allowing one to define a measure of error, namely a Riemannian metric. The edge-based error estimate is thus expressed as the length of the edges of the elements in this Riemannian metric. The construction of an anisotropic mesh may thus be interpreted as being a uniform mesh in the defined metric. This metric introduces and controls the magnitude as well as the direction of the grid anisotropy. A mesh movement scheme is then applied as the adaptive strategy, which, in contrast to the spring analogy technique used by Nakahashi and Deiwert,[14] has no constraint on grid orthogonality. This leads to a simple and efficient nodal redistribution algorithm offering a greater range of grid point displacements.

To efficiently attain the goal of mesh independence, the adaptive procedure must be based on concepts that are independent of grid topology and adaptive strategy. Using an edge-based error estimate, the optimal grid for a fixed number of nodes is thus defined as one in which the error is equidistributed over the edges. In the current paper, we restrict ourselves to the application of an r-method on structured meshes, although it can be easily combined with an h-method, to attain the desired error limit.

2. FLOW SOLVER

2.1. *Governing equations*

The conservation form of the mass, momentum and energy equations describing an inviscid compressible flow may be written as

$$\mathbf{Q}_{,t} + \mathbf{F}_{i,i} = 0, \tag{1}$$

where $\mathbf{Q}$ is the vector of the conservative variables and $\mathbf{F}_i$ is the vector of the convective fluxes. Indices i in the above formula refer to the axes of a Cartesian co-ordinate system, a comma denotes partial differentiation and the summation convention is applied. In 2D problems, the components of $\mathbf{Q}$ and $\mathbf{F}_i$ are

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix}, \qquad \mathbf{F}_i = \begin{bmatrix} \rho v_i \\ \rho v_1 v_i + p\delta_{1i} \\ \rho v_2 v_i + p\delta_{2i} \\ (\rho e + p)v_i \end{bmatrix}, \tag{2}$$

where ρ is the density, v_i are the velocity components, p is the pressure, δ_{ij} is the Kronecker delta symbol and e is the specific total internal energy.

This system is closed with the equation of state for a perfect gas,

$$p = (\gamma - 1)\rho(e - \tfrac{1}{2}v_i v_i), \tag{3}$$

where γ is the ratio of specific heats.

2.2. *Weak Galerkin formulation*

The weak formulation is obtained by minimizing the residuals of (1) over the solution domain. Equation (1) is multiplied by a weight function and integrated over the domain as

$$\int_\Omega (\mathbf{Q}_{,t} + \mathbf{F}_{i,i})\mathbf{W}\mathrm{d}\Omega = 0, \tag{4}$$

where the weight functions $\mathbf{W}$ are identical, in a Galerkin finite element formulation, with the interpolation functions of the unknown variables. By making use of the Gauss divergence theorem, the weak statement

$$\int_\Omega \mathbf{Q}_{,t}\mathbf{W}\mathrm{d}\Omega - \int_\Omega \mathbf{F}_i\mathbf{W}_{,i}\mathrm{d}\Omega + \int_\Gamma \mathbf{F}_i n_i \mathbf{W}\mathrm{d}\Gamma = 0 \tag{5}$$

is obtained, where Γ denotes the boundary of the domain Ω and n_i is the ith component of the outward unit normal to the boundary Γ.

2.3. *Temporal discretization*

The steady-state solution of (5) is obtained by an implicit time-marching formulation, with the discretization of the time-dependent term based on a backward first-order difference. Accordingly, equation (5) is expressed as

$$\int_\Omega \frac{\Delta \mathbf{Q}}{\Delta t}\mathbf{W}\mathrm{d}\Omega - \int_\Omega \mathbf{F}_i^{n+1}\mathbf{W}_{,i}\mathrm{d}\Omega + \int_\Gamma \mathbf{F}_i^{n+1}n_i\mathbf{W}\mathrm{d}\Gamma = 0, \tag{6}$$

where $\Delta\mathbf{Q}$ denotes the increment of the solution after a period of time Δt.

To solve the non-linear semidiscrete form (6), a linearization in time about the time level n is applied,[15] yielding

$$\int_\Omega \mathbf{L}\frac{\Delta \mathbf{U}}{\Delta t}\mathbf{W}\mathrm{d}\Omega - \int_\Omega \mathbf{A}_i\Delta\mathbf{U}\mathbf{W}_{,i}\mathrm{d}\Omega + \int_\Gamma \mathbf{A}_i n_i\Delta\mathbf{U}\mathbf{W}\mathrm{d}\Gamma = \int_\Omega \mathbf{F}_i\mathbf{W}_{,i}\mathrm{d}\Omega - \int_\Gamma \mathbf{F}_i n_i\mathbf{W}\mathrm{d}\Gamma, \tag{7}$$

where $\mathbf{U}=[\rho, v_1, v_2, e]^{\mathrm{T}}$ is the vector of the primitive variables and $\mathbf{L}$ and $\mathbf{A}_i$ are the Jacobian matrices of the vectors $\mathbf{Q}$ and $\mathbf{F}_i$ with respect to $\mathbf{U}$ respectively. The superscript n is dropped for simplicity.

2.4. *Finite element space discretization*

The computational domain is subdivided into quadrilateral elements where the solution vector $\mathbf{U}$ is approximated by bilinear shape functions as

$$\mathbf{U} \approx \mathbf{U}_h = \sum_{J=1}^{N_n} \hat{\mathbf{U}}_J(t)\Phi_J(\mathbf{x}), \tag{8}$$

where $\hat{\mathbf{U}}_J$ are the nodal values of the approximate solution $\mathbf{U}_h$, Φ_J is the shape function associated with node J and N_n is the total number of nodes.

Substituting equation (8) into the variational statement (7), one obtains the algebraic system of equations

$$(\mathbf{M}+\mathbf{K})\Delta\hat{\mathbf{U}} = -\mathbf{R}, \tag{9}$$

where the mass matrix $\mathbf{M}$, the influence matrix $\mathbf{K}$ and the residual vector $\mathbf{R}$ are given by

$$\mathbf{M}_{IJ} = \int_\Omega \frac{1}{\Delta t}\mathbf{L}\Phi_I\Phi_J\mathrm{d}\Omega, \tag{10}$$

$$\mathbf{K}_{IJ} = \int_\Gamma \mathbf{A}_i n_i\Phi_I\Phi_J\mathrm{d}\Gamma - \int_\Omega \mathbf{A}_i\Phi_{I,i}\Phi_J\mathrm{d}\Omega, \tag{11}$$

$$\mathbf{R}_I = \int_\Gamma \mathbf{F}_i n_i\Phi_I\mathrm{d}\Gamma - \int_\Omega \mathbf{F}_i\Phi_{I,i}\mathrm{d}\Omega. \tag{12}$$

A Galerkin FEM gives rise to central difference approximations of differential operators, which are non-dissipative by themselves. In order to suppress the tendency for odd–even node decoupling of the solution and to prevent unphysical oscillations near discontinuities, artificial dissipation terms are added to the governing equations in the form of Laplacians of primitive variables.[15] These additional terms are also treated in an implicit manner and the global algebraic system of equations is solved by a parallel Gauss elimination method.

3. GRID ADAPTATION

Following the classical approach, the optimal mesh for a fixed number of elements may be defined as one in which the error is equidistributed over each element.[16] The adaptive strategy presented in this work, coupled to an edge-based error estimate, specifically aims to equidistribute the error over the edges of the elements. A node-moving scheme is then implemented, relocating the grid points along those directions where the error is high.

3.1. Edge-based error estimate

Consider a 1D problem in which the solution variable g is approximated by g_h with linear interpolation. A local error E_e is defined over an element e to be

$$E_e = g - g_h. \tag{13}$$

By expanding the solution g at one end of the element e and provided that the nodal error is zero, the error E_e for linear interpolation over an element may be cast into the form

$$E_e = \frac{1}{2}x(h-x)\frac{\mathrm{d}^2 g_h}{\mathrm{d}x^2}\bigg|_e, \tag{14}$$

where h represents the element length and x is measured from one end of the element e. The root mean square (RMS) interpolation error over an element spanning the interval $[0, h]$ may be defined as[3]

$$E_e^{\mathrm{RMS}} = \frac{1}{\sqrt{(120)}}h^2\left|\frac{\mathrm{d}^2 g_h}{\mathrm{d}x^2}\right|_e. \tag{15}$$

Thus the interpolation for this 1D problem is proportional to the product of the second derivative and the square of the characteristic length of the element, h.

Extending these ideas to the 2D case, the second derivatives can now be replaced by the symmetric Hessian matrix

$$H_{ij} = g_{h,ij}. \tag{16}$$

Since g_h is linear for each element, the second derivatives have no representation. However, a weak formulation, combined with mass lumping, can be applied to recover an estimate of the second derivatives. This yields the expression

$$g_{h,ij}|_I = \frac{\int_{\Omega_I} g_{h,ij}\Phi_I \mathrm{d}\Omega}{\int_{\Omega_I} \Phi_I \mathrm{d}\Omega}, \tag{17}$$

where Ω_I represents the elements sharing node I. After integration of (17) by parts, the nodal values of the Hessian reduce to

$$g_{h,ij}|_J = \frac{\int_{\Gamma_I} g_{h,i}\Phi_I n_j \mathrm{d}\Gamma - \int_{\Omega_I} g_{h,i}\Phi_{I,j}\mathrm{d}\Omega}{\int_{\Omega_I} \Phi_I \mathrm{d}\Omega}. \tag{18}$$

The Hessian matrix given by (16) may be diagonalized as

$$\mathbf{H} = \mathbf{R}(\alpha)\Lambda\mathbf{R}^{\mathrm{T}}(\alpha), \tag{19}$$

where Λ is the diagonal matrix of the eigenvalues of $\mathbf{H}$ and $\mathbf{R}$ is the matrix of the eigenvectors. The transformation Λ is a scaling in the directions of the axes and $\mathbf{R}$ is a rotation with angle α that the eigenvector corresponding to the smallest eigenvalue λ_1 makes with the x_1-axis.

In order to obtain a symmetric, positive definite matrix, the Hessian is modified by taking the absolute value of its eigenvalues.[17] This results in

$$\bar{\mathbf{H}} = \mathbf{R}(\alpha)|\Lambda|\mathbf{R}^{\mathrm{T}}(\alpha) = \mathbf{S}(\alpha)\mathbf{S}^{\mathrm{T}}(\alpha), \tag{20}$$

where $\mathbf{S} = \mathbf{R}\sqrt{|\Lambda|}$. The transformation $\mathbf{S}$ of a unit circle would be an ellipse, rotated through an angle α, whose semi-major and semi-minor axes are the reciprocals of the square roots of the eigenvalues $|\lambda_1|$ and $|\lambda_2|$ respectively (see Figure 1). Therefore, one may obtain a directionally stretched grid by mapping a uniform mesh using the transformation $\mathbf{S}$.

However, in the current approach a mesh with edges of equal length is sought in the transformed plane $\mathbf{S}^{\mathrm{T}}$, where the length of a curve β is given by

$$d(\beta) = \int_0^1 \sqrt{[\mathbf{s}'(l)^{\mathrm{T}}\bar{\mathbf{H}}(l)\mathbf{s}'(l)]}\,\mathrm{d}l \tag{21}$$

and $\mathbf{s}(l)$ is a parametric representation of the curve β.

Since $\bar{\mathbf{H}}$ is a function of the space co-ordinates, equation (21) defines a Riemannian metric. The modified Hessian $\bar{\mathbf{H}}$ is computed and stored on a background mesh and thus the value of $\bar{\mathbf{H}}$ at any position of the domain can be interpolated during the adaptive process on this mesh. The edge-based error estimate can then be numerically evaluated from (21) for each edge of an element.

3.2. *Moving-node scheme*

The adaptive strategy modifies the grid under the guidance of the error estimate to improve the quality of the numerical solution. Thus the use of an appropriate adaptive scheme is crucial for achieving the desired directionally adapted mesh. The proposed strategy relies on a node-moving scheme, also called nodal redistribution. As illustrated in Figure 2, the mesh may be viewed as a

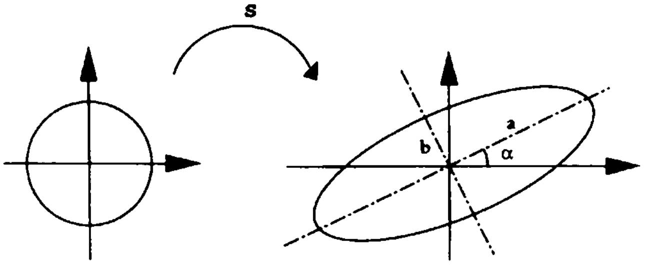

Figure 1. Transformation of a unit circle by S, where $a = |\lambda_1|^{-1/2}$ and $b = |\lambda_2|^{-1/2}$

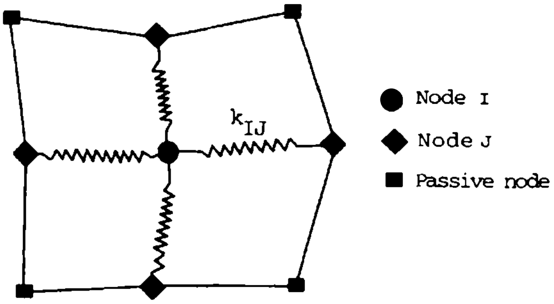

Figure 2. Spring analogy for a patch of elements

network of springs whose stiffness constants represents the edge-based error estimate. The positions of the grid vertices may then be interpreted as the solution of an energy minimization problem. This yields for each vertex I

$$\min_{\mathbf{x}_I} P_I = \min_{\mathbf{x}_I} \sum_J (\mathbf{x}_I - \mathbf{x}_J)^2 k_{IJ}, \tag{22}$$

where P_I denotes the potential energy of the four springs sharing a node I and k_{IJ} are the associated four stiffness constants. These constants may be written specifically as

$$k_{IJ} = \frac{d(\mathbf{x}_I, \mathbf{x}_J)}{\|\mathbf{x}_I - \mathbf{x}_J\|}, \tag{23}$$

where $\|\cdot\|$ indicates the Euclidean norm and $d(\mathbf{x}_I, \mathbf{x}_J)$ is the length of the edge $[\mathbf{x}_I, \mathbf{x}_J]$ in the Riemannian metric defined by (21).

After simplifications, equation (22) reduces to the following system describing the equilibrium state of a spring network:

$$\sum_J (\mathbf{x}_I^{m+1} - \mathbf{x}_J^{m+1}) k_{IJ}^{m} = 0. \tag{24}$$

By lagging $\mathbf{x}_J$ and k_{IJ} at the previous iteration m, equation (24) becomes

$$\Delta \mathbf{x}_I = \frac{\sum_J (\mathbf{x}_J^m - \mathbf{x}_I^m) k_{IJ}^m}{\sum_J k_{IJ}^m} \tag{25}$$

and the position of the vertex I is updated according to the expression

$$\mathbf{x}_I^{m+1} = \mathbf{x}_I^m + \omega \Delta \mathbf{x}_I, \tag{26}$$

where ω is a relaxation factor. The convergence of this scheme can be enhanced by using a Gauss–Seidel algorithm with the latest values of $\mathbf{x}_J$ and k_{IJ} in (25).

In this procedure the boundary nodes are also free to move along their respective curves. The same algorithm as for the internal nodes is applied to compute their new positions, but they are projected back to their corresponding boundary curves.

The grid adaptation procedure may be summarized in the following steps:

```
• read a background mesh and the corresponding solution
• compute $\bar{\mathbf{H}}$ on the background mesh
• current mesh is initialized by an initial mesh
• move the nodes of the current mesh as follows:
  DO $m = 1$, MAXITER
    DO $inod = 1$, NNODE
      Do $iedge = 1$, NEDGE
        • determine $\bar{\mathbf{H}}$ by interpolating on the background mesh
        • compute spring constants by numerical integration of (23)
    ENDDO
    • find new position of inod
    • check quality of resulting elements sharing inod
    • move inod to its new position
  ENDDO
  If (MAXDISP.LE. tolerance) exit the $m$ loop
ENDDO
```

where *NEDGE* represents the number of edges sharing the node *inod* and *MAXDISP* is the maximal displacement of all nodes for a particular iteration *m*.

In the general case the background mesh serves only to compute and store the error estimate and hence could be different from the initial mesh. For the first few cycles of adaptation one can thus develop a crude error estimate on a coarse mesh to adapt a finer grid.

Both flow solver and grid adaptation procedures are placed in an iterative loop which is repeated until the lowest value of a user-specified artificial dissipation coefficient is reached. In the following, each iteration of this loop will be called an adaptive cycle or level.

4. NUMERICAL RESULTS

For all the test cases investigated in this paper, the density solution is the flow variable used for the error estimate, although other choices such as pressure, Mach number or a combination of them could be used. Also, the background mesh is taken to be identical with the adapted mesh of the previous cycle for all adaptation cycles.

4.1. Analytical test case

Since the grid adaptation procedure contains some new aspects, it is important to investigate its effectiveness on an analytical case. The aim of the first example is therefore to demonstrate the capability of this procedure to equidistribute the interpolation error of a given function over the edges. A function, with strong gradients, in the form

$$g(x_1, x_2) = \tan^{-1}[10^3(x_1^4 x_2^4 - 0{\cdot}25)] \tag{27}$$

has been chosen over the domain [0, 2] × [0, 1].

The initial mesh and corresponding isocontours of g are displayed in Figures 3(a) and 3(á) respectively. After 300 iterations of the mesh movement scheme the adapted mesh shown in Figure 3(b) is obtained. As illustrated in Figure 3(b́), this mesh permits a better representation of the function

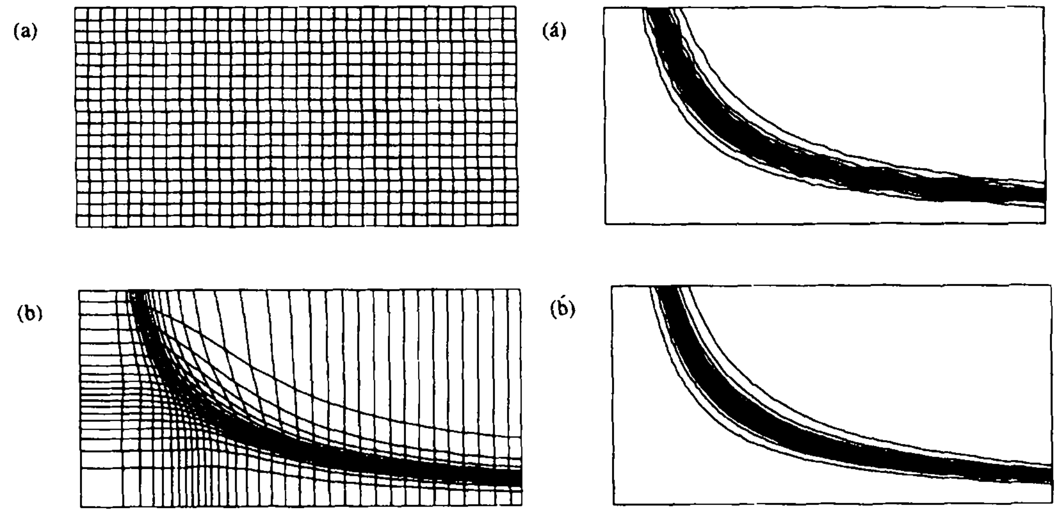

Figure 3. Initial (a) and adapted (b) meshes and corresponding isocontours, (á) and (b́), of g

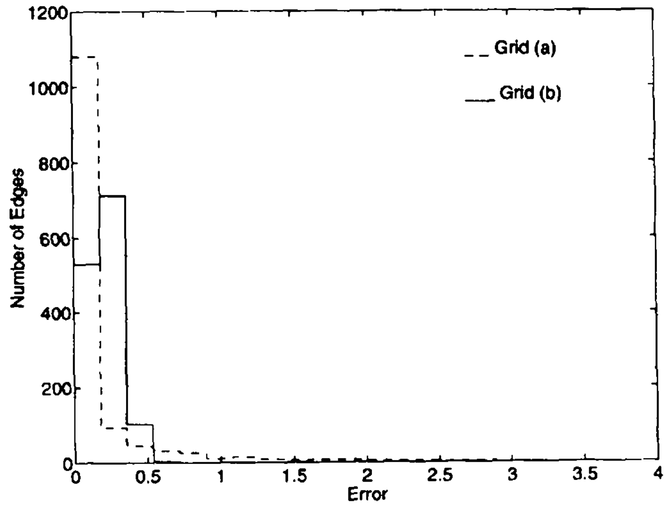

Figure 4. Histogram of error over edges

g. Figure 4 represents a histogram of the number of edges versus the error over these edges. In the ideal case all the edges would have the same error. In practice a nearly Gaussian distribution is obtained, where the maximum error is reduced fivefold.

The plot of the L_2 and L_∞ norms of node displacements in Figure 5 shows that the correction of position vertices dropped by a magnitude of two orders after 700 iterations. In contrast, the L_∞ norm of the edge error stalls at the value of 0·55 after 170 iterations.

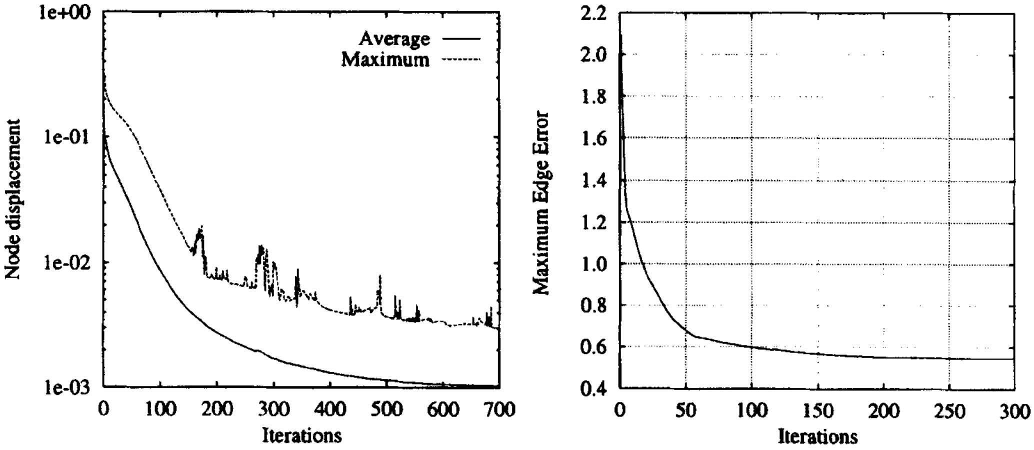

Figure 5. Node displacement and maximum error-edge convergence histories

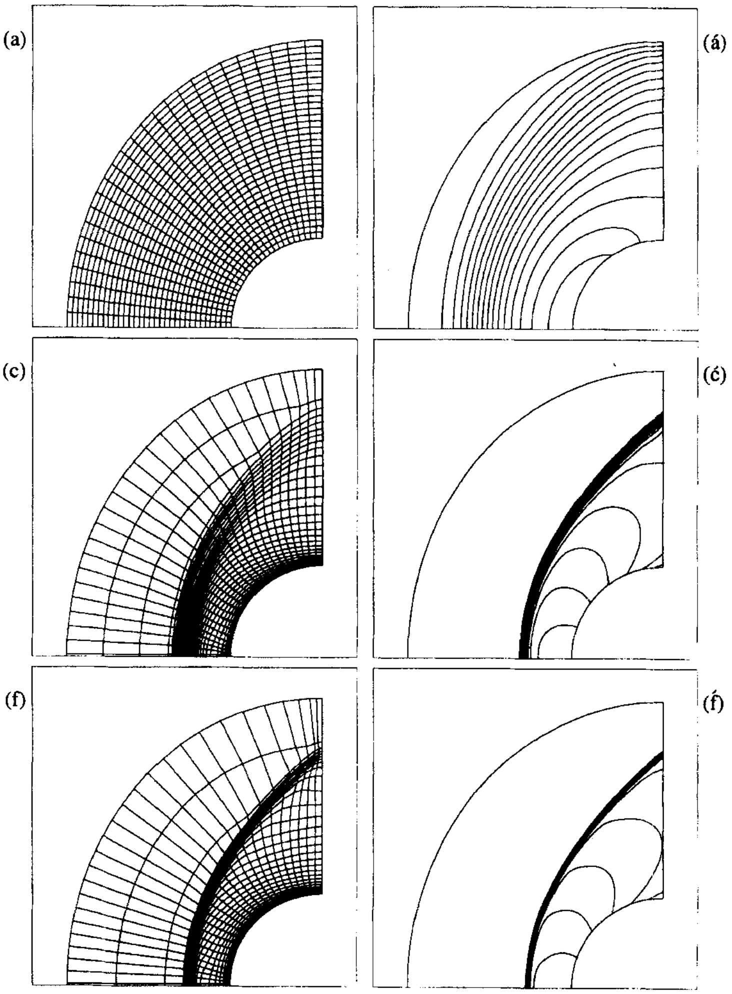

Figure 6. Adapted grids and corresponding Mach number contours for cycles (a), (c) and (f)

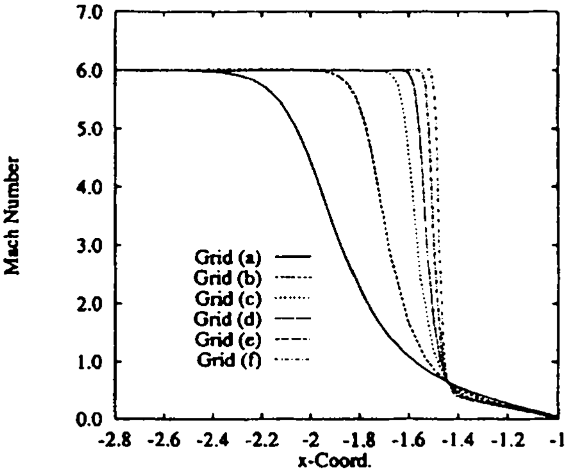

Figure 7. Enhancement of Mach number distribution on stagnation line with grid adaptation cycles

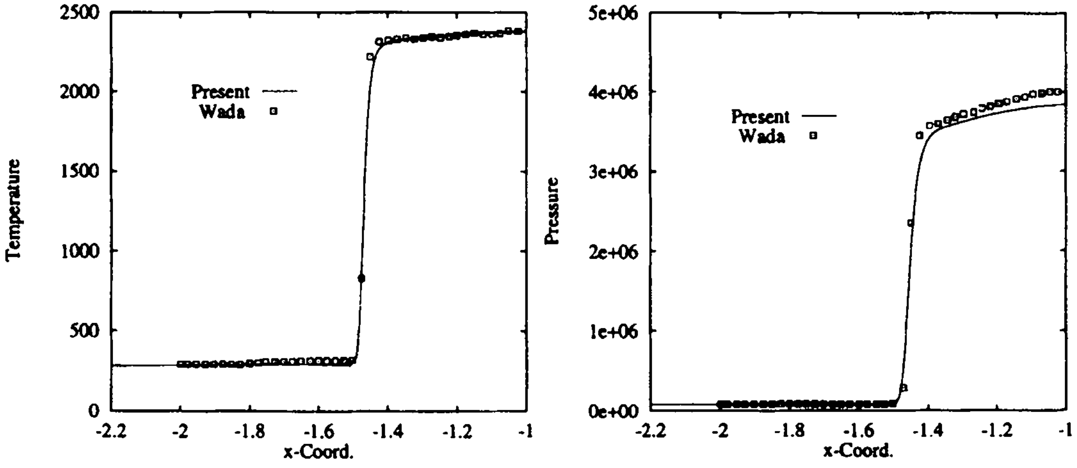

Figure 8. Comparison of stagnation line results of present approach with other numerical results

4.2. Hypersonic flow over a cylinder

In this example the proposed methodology is applied to a hypersonic flow over a cylinder. This calculation was performed with a freestream Mach number of 6 and zero angle of attack. The initial finite element mesh, shown in Figure 6(a), is composed of 33 × 53 nodes distributed uniformly in both directions. The corresponding Mach number contours of the flow are depicted in Figure 6(á).

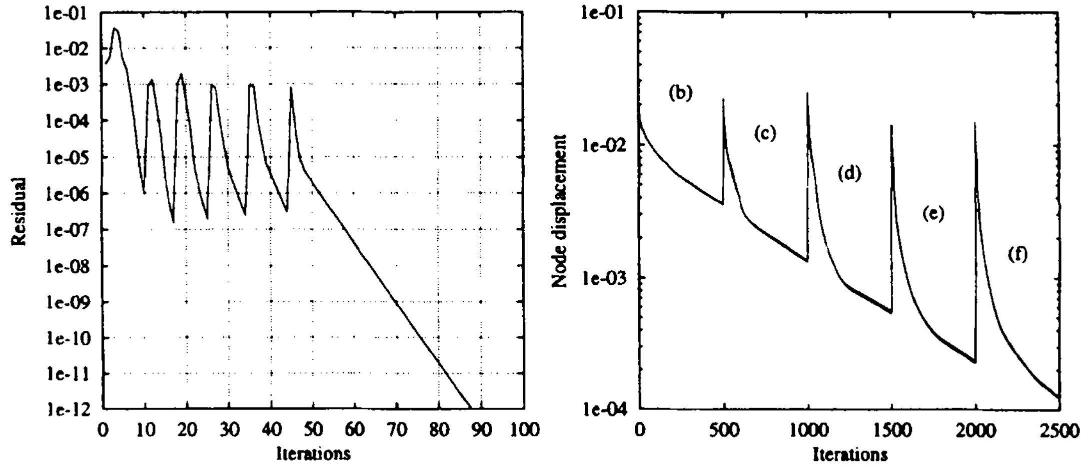

Figure 9. Convergence histories of the flow solver and the adaption procedure

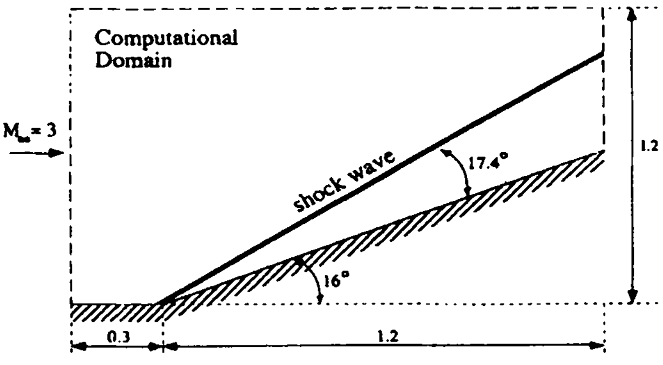

Figure 10. Definition of compression corner test case

The adapted meshes in Figure 6(c) and 6(f) require two and five levels of adaptation respectively. The corresponding flow field contours, Figure 6(c) and 6(f), demonstrate the benefits of the grid adaptation in resolving a detached bow shock.

Figure 7 illustrates the enhancement of the Mach number distribution on the stagnation line with grid adaptation cycles. The profiles of temperature and pressure along the stagnation streamline are compared in Figure 8 with the results of Wada and Liou.[18] Overall, this figure reveals good agreement between the adapted solution and Wada's results. Slight discrepancies in pressure distributions are, however, observed near the wall owing to the different manner of implementing the wall boundary conditions. It must be pointed out that Wada's solution was predicted on a finer grid with an advection upwind splitting method (AUSM), proven superior to most standard schemes of flux-difference splitting and flux-vector splitting.

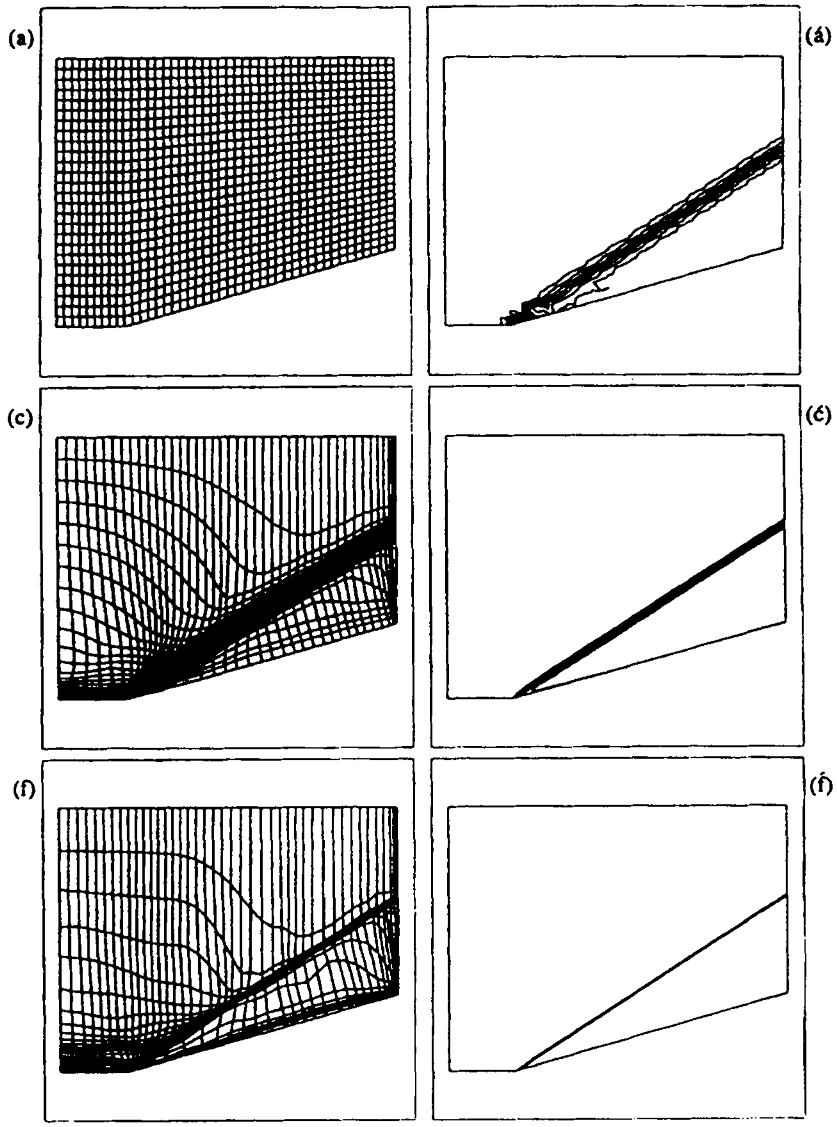

Figure 11. Adapted grids and corresponding density contours for cycles (a), (c) and (f)

The convergence histories of the flow solver and the grid adaptation procedure are presented in Figure 9, where the jumps in the curves represent the beginning of each new cycle. For a given value of the artificial dissipation coefficient the L_2 norm of the flow residual at that cycle is lowered by three orders of magnitude and then the mesh nodes are displaced 500 times. Figure 9 shows that the use of a local-time stepping technique with a high CFL number of 350 permits one to reach machine accuracy in few iterations.

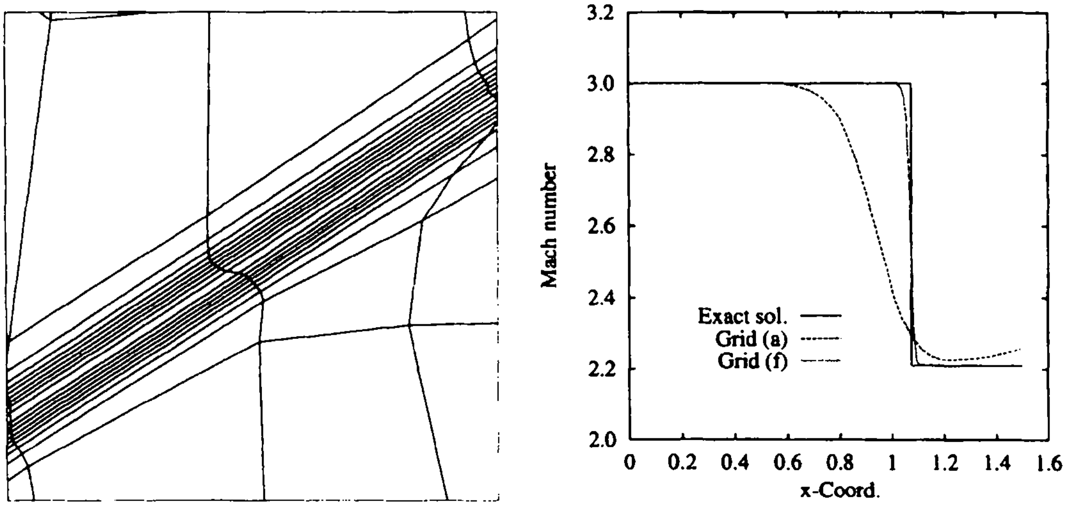

Figure 12. Magnification of grid in shock region (left) and distribution of Mach number of $y = 0{\cdot}5$ (right)

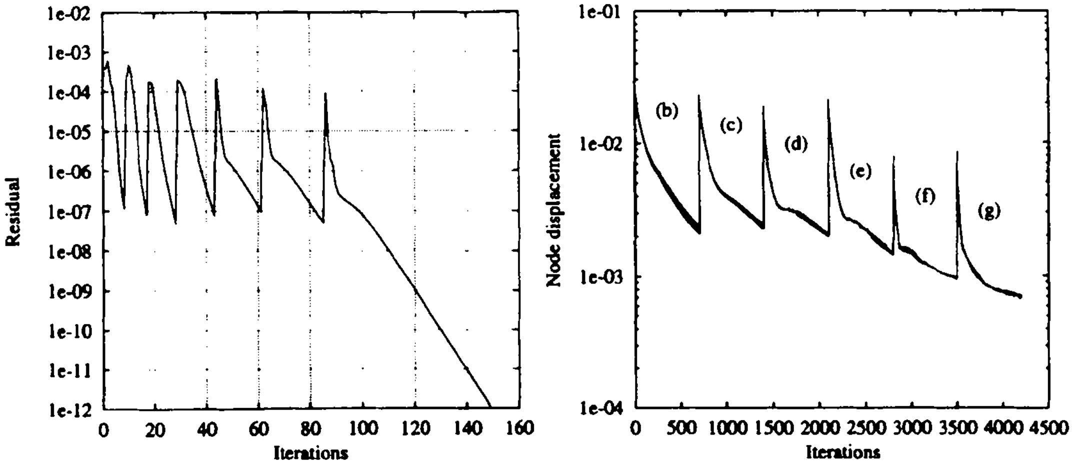

Figure 13. Convergence histories of flow solver (left) and adaption procedure (right)

4.3. Supersonic compression corner

The third test case is a flow at Mach 3 over a 16° ramp (see Figure 10). This example tests certain features of the algorithm, including the resolution of the oblique shock and its proper angle. The initial coarse mesh composed of 44 × 27 nodes is shown in Figure 11(a), while the resulting density contours are presented in Figure 11(á). The results after two and five levels of adaptation are depicted in Figures 11(c), 11(ć) and 11(f), 11(f́) respectively. Comparison of Figure 11(á) and 11(f́) demonstrates the important role of the grid adaptation in capturing a sharp shock at the correct angle.

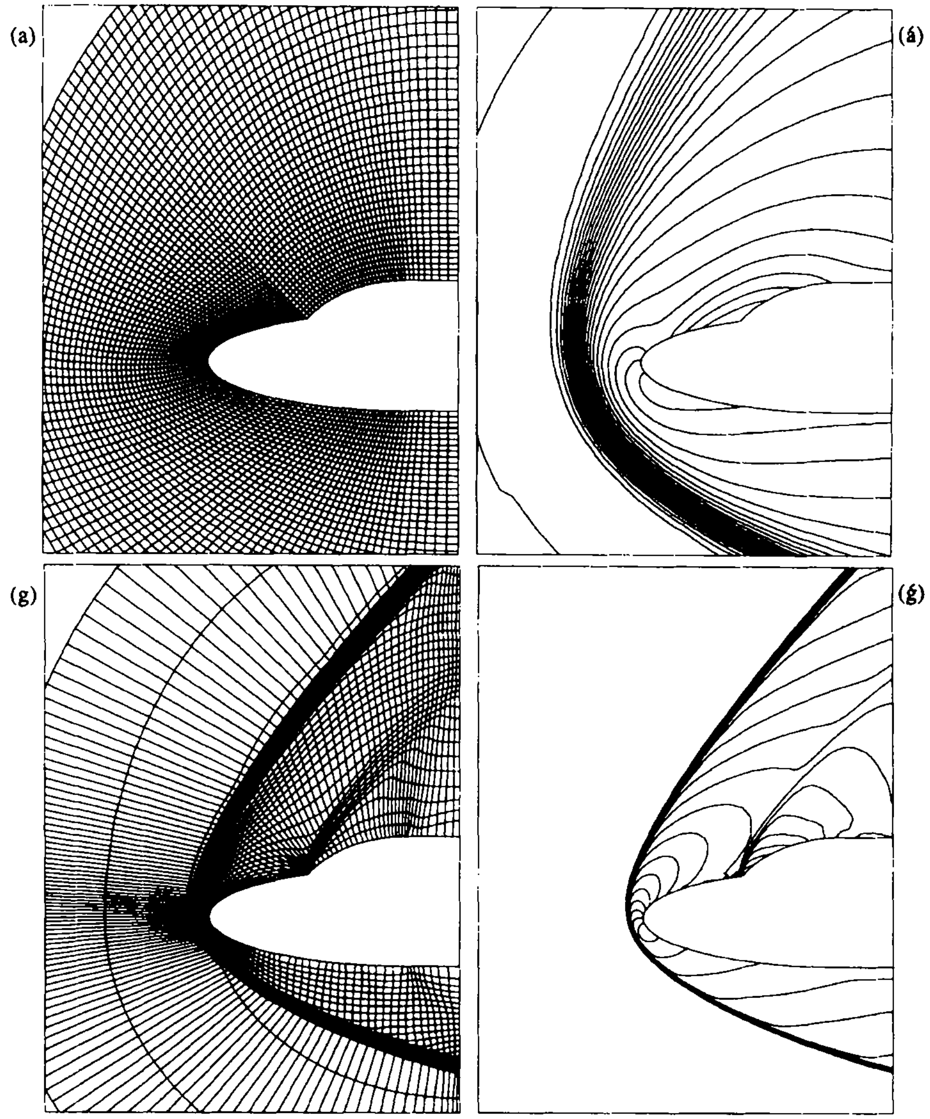

Figure 14. Initial and final adapted grids with corresponding Mach number contours

The magnification of the mesh in the shock region presented in Figure 12 shows that the quadrilateral elements are oriented in the direction of the shock with an aspect ratio as high as 50. Figure 15 clearly illustrates the superiority of the adapted solution in approximating the exact solution. In fact, the use of an appropriate grid also allows the reduction of the artificial dissipation coefficient by a factor of 10. The convergence history of the flow solver and the node displacement convergence histories are presented in Figure 13.

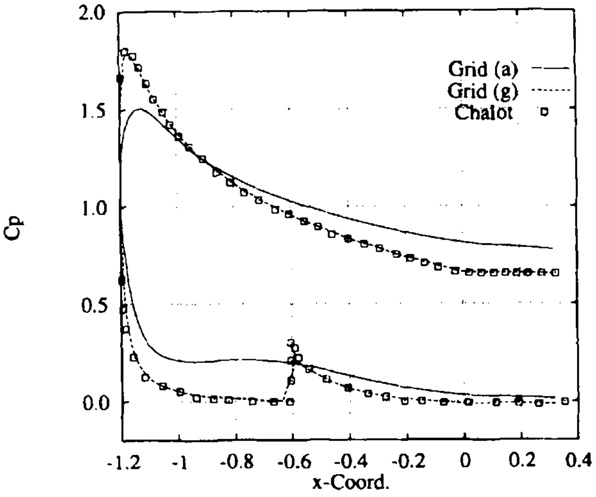

Figure 15. Enhancement of the C_p distribution on the body wall with grid adaptation cycles

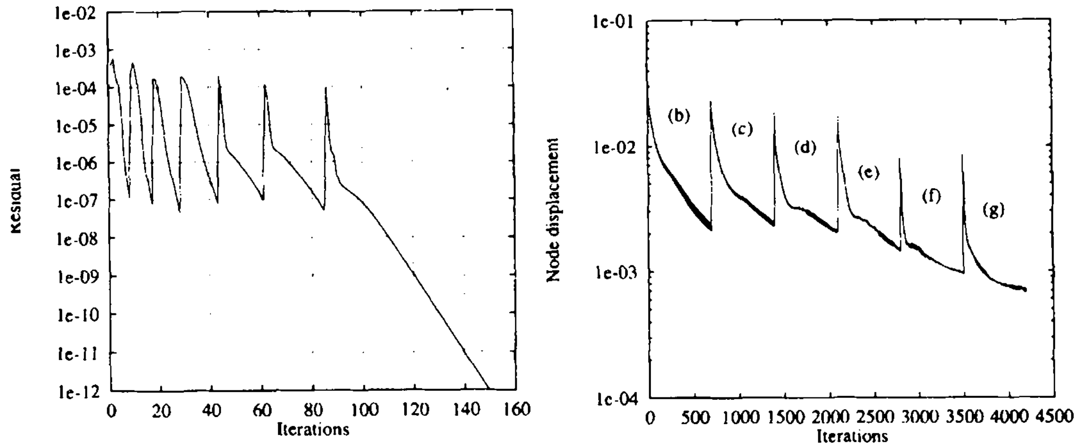

Figure 16. Convergence history of flow solver

4.4. Hypersonic flow over a double ellipse

In this test case a double-ellipse profile is placed in a Mach 8 flow at 30° angle of attack. Since it was introduced in the Workshop on Hypersonic Flows for Reentry Problems in 1990, this benchmark has become a great challenge for testing compressible flow solvers. The flow field is characterized by a strong detached shock wave followed by a moderate canopy shock. Therefore the use of standard flow solvers on a uniform grid tends to produce smeared shocks.

The computations were initiated on 45 × 124 (5580) grid nodes, Figure 14(a), and the corresponding solution is shown in Figure 14(á). The adapted grid, Figure 14(g), is obtained after six levels of adaptation. The adapted solution is presented in Figure 14(ģ) and it can be clearly seen that the detached and canopy shocks are well-resolved.

The body pressure coefficient (C_p) distributions of initial and adapted solutions are compared with Chalot *et al.*'s results[19] in Figure 15. This plot demonstrates the important role of the adapted grid in shock capturing through mesh alignment and the ensuing reduction of the required artificial dissipation in the flow solver. Chalot's computation was carried out by an FEM on an adapted triangular mesh of 6721 nodes. As displayed in Figure 16, full convergence of the flow solver requires 150 iterations, while 700 iterations are used for each adaptation cycle.

5. CONCLUSIONS

An adaptive finite element method with directional features, using an edge-based error estimate on quadrilateral meshes, has been described and applied to high-speed flows. The error of the numerical solution is measured by it second derivatives and the resulting Hessian tensor is used to define a Riemannian metric. An improved mesh movement strategy with no orthogonality constraints is introduced to equidistribute the lengths of the edges of the elements in the defined metric. The adaptive procedure has been proven to be effective on an analytical test case, where a nearly Gaussian distribution of the error is obtained. The flow solver, combined with the proposed grid adaptation method is then validated on a supersonic compression corner, capturing the oblique shock with high resolution and the correct angle. The methodology is also tested on hypersonic flows and other numerical results have been correctly reproduced on coarser meshes. Current work involves the extension of the approach to 3D hexahedral and tetrahedral grids.

ACKNOWLEDGEMENTS

The authors would like to thank the NSERC (Natural Sciences and Engineering Research Council of Canada) and Quebec's FCAR (Fonds pour la Formation de Chercheurs et l'aide à la Recherche) for their financial support of this work.

REFERENCES

1. R. Löhner, K. Morgan and O. C. Zienkiewicz, 'Adaptive grid refinement for the compressible Euler equation', in I. Babuška, O. C. Zienkiewicz, J. Gago and A. Oliveira (eds), *Accuracy Estimates and Adaptive Refinements in Finite Element Computation*, Wiley, New York, 1986.
2. J. T. Oden, T. Strouboulis and P. Devloo, 'Adaptive finite element methods for high-speed compressible flows', *Int. j. numer. methods fluids*, 7, 1211–1228 (1987).
3. J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz, 'Adaptive remeshing for compressible flow computations', *J. Comput. Phys.*, **72**, 449–466 (1987).
4. B. Palmerio, 'A two-dimensional FEM adaptive moving-node method for steady Euler flow simulation', *Comput. Methods Appl. Mech. Eng.*, **71**, 315–340 (1988).
5. R. Körnhuber and R. Roitzsch, 'On adaptive grid refinement in the presence of internal or boundary layers', *Impact Comput. Sci. Eng.*, **2**, 40–72 (1990).
6. M. Fortin, M.-G. Vallet, D. Poirier and W. G. Habashi, 'Error estimation and directionally adaptive meshing', *AIAA Paper 94-2221*, 1994.
7. Y. G. Kallinderis and J. R. Baron, 'Adaptation methods for a new Navier–Stokes algorithm', *AIAA J.*, **27**, 37–43 (1989).
8. R. L. Davis and J. F. Dannenhoffer, '3-D adaptive grid-embedding Euler technique', *AIAA Paper 93-0330*, 1993.
9. A. Jameson, 'Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flows', *Int. J. Comput. Fluid Dyn.*, **4**, 171–218 (1995).
10. A. Arnone, 'Multigrid methods for turbomachinery Navier–Stokes calculations', in W. Habashi (ed.), *Solution Techniques for Large- Scale CFD Problems*, 1995, Wiley pp. 294–332.

11. M. H. Lallemand, H. Steve and A. Dervieux, 'Unstructured multigridding by volume agglomeration: current status', *Comput. Fluids*, **21**, 397–433 (1992).
12. D. J. Mavriplis, 'Three-dimensional unstructured multigrid for the Euler equations', *AIAA J.*, **30**, 1753–1761 (1992).
13. P. A. Gnoffo, 'A finite-volume, adaptive grid algorithm applied to planetary entry flowfields', *AIAA J.*, **21**, 1249–1254 (1983).
14. K. Nakahashi and G. S. Deiwert, 'Self-adaptive grid method with application to airfoil flow', *AIAA J.*, **25**, 513–520 (1987).
15. D. Ait-Ali-Yahia, W. G. Habashi and G. S. Baruzzi, 'A finite element method for hypersonic reacting flows', in Dhaubhadel, Habashi and Engleman (eds), *Advances in Finite Element Analysis in Fluid Dynamics*, FED Vol. 200, ASME, New York, 1994, pp. 11–19.
16. J. T. Oden and L. Demkowicz, 'A survey of adaptive finite element methods in computational mechanics', in A. K. Noor and J. T. Oden (eds), *State-of-the-Art Surveys on Computational Mechanics*, ASME, New York, 1989, pp. 441–467.
17. M.-G. Vallet, 'Génération de maillages éléments finis anisotropes et adaptatifs', *Ph.D. Thesis*, Université Pierre et Marie Curie, Paris VI, 1992.
18. Y. Wada and M. S. Liou, 'A flux-splitting scheme with high-resolution and robustness for discontinuities', *AIAA Paper 94-0083*, 1994.
19. F. Chalot, T. J. R. Hughes, Z. Johan and F. Shakib, in *Workshop on Hypersonic Flows for Reentry Problems*, 1990.