INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 1131–1152 (1991)

TURBULENT FLOW CALCULATIONS USING UNSTRUCTURED AND ADAPTIVE MESHES

DIMITRI J. MAVRIPLIS

Institute for Computer Applications in Science and Engineering, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23665, U.S.A.

SUMMARY

A method of efficiently computing turbulent compressible flow over complex two-dimensional configurations is presented. The method makes use of fully unstructured meshes throughout the entire flow field, thus enabling the treatment of arbitrarily complex geometries and the use of adaptive meshing techniques throughout both viscous and inviscid regions of the flow field. Mesh generation is based on a locally mapped Delaunay technique in order to generate unstructured meshes with highly stretched elements in the viscous regions. The flow equations are discretized using a finite element Navier–Stokes solver, and rapid convergence to steady state is achieved using an unstructured multigrid algorithm. Turbulence modelling is performed using an inexpensive algebraic model, implemented for use on unstructured and adaptive meshes. Compressible turbulent flow solutions about multiple-element aerofoil geometries are computed and compared with experimental data.

KEY WORDS Turbulent Navier–Stokes Unstructured Adaptive

1. INTRODUCTION

Although unstructured mesh techniques have been employed extensively in fields such as solid modelling and computational structural mechanics for many years, their use in the field of computational fluid dynamics (CFD) constitutes a relatively recent phenomenon. This situation is probably due to the large overheads generally incurred with unstructured mesh techniques, which can become unacceptable when coupled with the large computational requirements of many CFD problems. The advantages of unstructured meshes lie in the ability they afford for flexibly discretizing arbitrarily complex geometries and in the ease with which they lend themselves to adaptive meshing techniques, which can be employed to accurately resolve complex flows in an efficient manner. In recent years much progress has been made in developing more sophisticated unstructured mesh generation strategies for computational fluid dynamics problems[1–3] as well as in the development of novel and efficient flow solution algorithms.[4–7] However, much of this effort has been directed at two- and three-dimensional inviscid flow problems. The solution of high-Reynolds-number viscous flows, which are of much greater practical interest, introduces additional complications with regards to mesh generation and turbulence modelling. Most attempts at solving viscous flows using unstructured meshes have resorted to hybrid structured–unstructured meshes, where a thin structured mesh is placed in the boundary layer and wake regions and an unstructured mesh is employed in the regions of inviscid flow.[8,9] This type of compromise, however, limits the flexibility of the original unstructured approach. Geometries with close tolerances, where confluent wakes and boundary layers occur,

0271–2091/91/191131–22$11.00 *Received 4 September 1990*

may prove difficult to discretize with a hybrid approach, and the task of performing adaptive meshing throughout the viscous and inviscid regions of flow can be considerably more complex.

This paper describes a method for computing compressible turbulent viscous flows about arbitrary two-dimensional configurations using fully unstructured meshes and incorporating adaptive meshing techniques. The generation of meshes with highly stretched triangular elements in the boundary layer and wake regions is accomplished with a method based on a modified Delaunay triangulation technique.[10] The full Navier–Stokes equations are discretized and solved for on these meshes using an efficient finite element solver which converges rapidly to steady state using an unstructured multigrid strategy.[11] Turbulence modelling is achieved using an inexpensive algebraic model which has been devised specifically for use on unstructured and adaptive meshes.[12]

2. MESH GENERATION

2.1. Initial mesh generation

The initial unstructured mesh is generated in three essentially independent stages. First, a distribution of mesh points and associated stretching vectors is generated throughout the flow field. These points are then joined together in a manner influenced by the local stretching values to form a set on non-overlapping triangular elements which completely fill the domain. The resulting mesh is then smoothed by slightly repositioning the mesh points according to an elliptic smoothing operator discretized on the new mesh.

Although adaptive meshing techniques can be relied upon to increase the mesh resolution in regions of strong flow gradients, a good initial mesh point distribution is essential in order to initially capture all the salient features of the flow and to reduce the number of flow solution adaptivity cycles required to converge the accuracy of the solution process. This is especially true in the case of high-Reynolds-number viscous flows, where a highly stretched, densely packed distribution is required to efficiently resolve the thin shear layers. The initial distribution of mesh points is generated in a geometry-adaptive manner. Points are positioned along all flow field boundaries and along wake lines in a manner which is governed by the local rate of change of slope of the boundary curve and the thickness (height) of the elements to be generated at the boundary. The effect of this distribution method is to cluster points in regions of high boundary curvature and sharp corners, where large flow gradients are expected, as well as to reduce the stretching of the mesh in such regions.

A distribution of interior points is then constructed by generating a series of local hyperbolic structured meshes for each boundary curve or wake line using the previously generated boundary point distribution as an initial condition and prescribing a normal spacing of points as a function of the Reynolds number of the flow to be computed.[13] The union of all the points contained in the various overlapping hyperbolic meshes is then used as the basis for the triangulation. A value of stretching is required at each mesh point and this may be computed from the local ratio of the streamwise to normal point spacings in the local structured meshes, as shown in Figure 1. Initial point clustering in wake regions is achieved by drawing fictitious boundaries which approximate the estimated positions of the wakes. The position of these approximate wake lines is obtained by solving the corresponding inviscid flow problem using an inexpensive panel method solution.[14]

A point-filtering operation based on the aspect ratios of the structured local hyperbolic mesh cells is also employed to remove excessive points in the regions of inviscid flow. The structured mesh cell aspect ratios, as measured by the ratio of streamwise to normal mesh spacing, are large near the geometry walls and wake lines and decrease gradually towards the far field. In regions

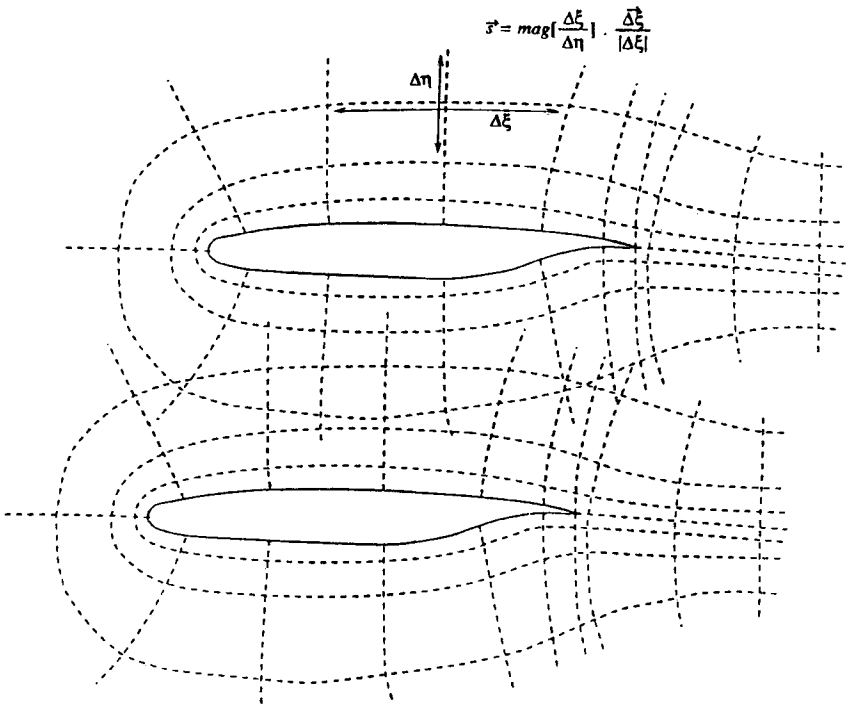


Figure 1. Overlapping structured C-meshes for multi-element aerofoil geometry providing initial definition of mesh point distribution and local stretching factors

where these aspect ratios become smaller than unity, the streamwise point distribution is coarsened by removing selected points, thus maintaining a nearly isotropic point distribution in the inviscid regions of flow. This filtered set of mesh points is then triangulated using a modified Delaunay triangulation criterion.[10] In its original form a Delaunay triangulation of a given set of points tends to produce the most equiangular triangles possible and is thus not well suited for the generation of highly stretched triangulations. The Delaunay criterion is thus modified according to the local stretching values. The local stretching value is used to define a stretched space within which the local mesh point distribution appears more isotropic. The Delaunay triangulation of the points in this stretched space is then constructed. This connected set of points is subsequently mapped back into physical space, thus producing the desired stretched triangulation. The actual construction of the stretched triangulation is performed in a two-stage process. A regular Delaunay triangulation of the mesh points in physical space is initially constructed using Bowyer's algorithm.[15] This initial triangulation is conveniently used to smooth the distribution of stretchings throughout the flow field by providing the basis for discretizing a smoothing operator which is then applied to the stretching values. This Delaunay triangulation is then transformed into a modified (stretched) Delaunay triangulation, based on the smoothed local stretching distribution, by swapping the diagonals according to the Delaunay maximum–minimum angle criterion[10] applied in the stretched space. This criterion provides a basis for deciding whether or not an edge should be swapped by choosing the configuration which maximizes the smallest of the angles formed between the edge to be swapped and its immediate neighbours.

A regular Delaunay triangulation of a set of points produces a smoothly varying mesh provided that the mesh points are initially distributed evenly and isotropically. Similarly,

a stretched or modified Delaunay triangulation of a set of points will result in a smoothly varying stretched mesh provided the mesh point and stretching distributions are closely coupled and provided that the stretching distribution varies slowly with respect to the local mesh element size. The geometry-adaptive mesh point distribution and the point-filtering operation previously described have thus been designed to ensure that these criteria are satisfied.

Finally, the stretched triangulation can be smoothed in a postprocessing operation by slightly repositioning the points according to a smoothing operator discretized on the mesh. Once the points have been displaced, the mesh may no longer obey the modified Delaunay criterion; thus the edges may be swapped to recover this property. Multiple smoothing and edge-swapping passes may then be employed to further smooth the mesh.

2.2. Adaptive meshing procedure

Once the initial stretched unstructured mesh has been generated and the flow field has been solved for on this mesh, a new adaptively refined mesh may be constructed by adding new points to the initial mesh in regions where large flow gradients or discretization errors are detected and locally restructuring the mesh. In this work the refinement criterion is based on the undivided difference of pressure and Mach number. Pressure gradients provide a good indication of inviscid flow phenomena such as shocks and expansions, while Mach number variations can be used to identify viscous phenomena such as boundary layers and wakes. Although the mesh point distribution can be refined in the adaptation process, the stretching distribution is held fixed. Thus, in order to maintain a close coupling between the final mesh point distribution and the stretching distribution, an isotropic refinement strategy must be adopted. The variations of pressure and Mach number within each triangular element of the mesh are examined. When these are larger than some fraction of the average variations of pressure and Mach number over all cells of the mesh, three new mesh points are created, one at the midpoint of each of the three edges of the cell. Each new mesh point is assigned a stretching value taken as the average of the stretchings of the two points at either end of the generating mesh edge. The new points are then inserted into the existing mesh by locally restructuring the mesh according to Bowyer's algorithm in the stretched space.[15] For each new mesh point the triangles whose circumcircles are intersected by this new point are located. The union of all intersected triangles forms a convex polygonal region which contains the new point. The existing structure in this region is removed and a new structure is constructed by joining the new point to all the vertices of this polygonal region, as shown in Figure 2. By simultaneously refining all three sides of a mesh element and by assigning average stretching values to the new points, directional biasing of the refinement process is avoided and a smooth distribution of stretching is maintained. The use of Bowyer's algorithm in the stretched space, which provides an effective method for constructing a stretched Delaunay triangulation through sequential point insertion and retriangulation, constitutes an ideal adaptive mesh enrichment strategy and obviates the need for global mesh regeneration.

When new boundary points are introduced, they must be repositioned on the spline definition of the geometry boundary. For curved surfaces this will not coincide with the midpoint of the original boundary mesh edge. For highly stretched meshes the distance between these two locations may in fact be much larger than the average width of the elements in the vicinity of the boundary and a restructuring of the entire region is required, as shown in Figure 3. This is accomplished by drawing the line joining the midpoint of the boundary edge being refined to the spline-displaced position of this new boundary point. The union of all triangular mesh elements intersected by this line, as well as elements whose circumcircles are intersected by the spline-displaced boundary point, is tagged for reconstruction. This entire region is then restructured via

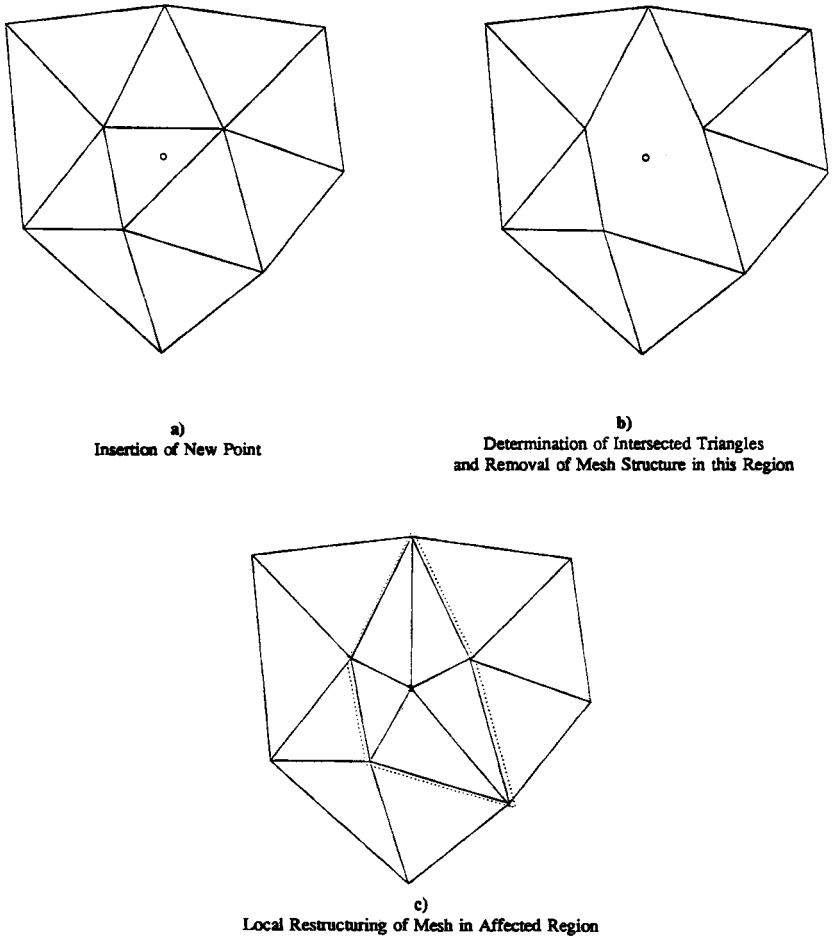


Figure 2. Illustration of Bowyer's algorithm for Delaunay triangulation

the standard Bowyer algorithm, i.e. removing all mesh edges in this region and forming a new structure by joining up the new mesh point to all the vertices of the polygonal region.

3. FLOW SOLUTION

In non-dimensional conservative vector form the Navier–Stokes equations read

$$\frac{\partial w}{\partial t}+\nabla\cdot\mathbf{F}_c=\frac{1}{Re_\infty}\nabla\cdot\mathbf{F}_v, \tag{1}$$

where Re_∞ denotes the overall flow Reynolds number and w represents the conserved variables

$$w=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}, \tag{2}$$

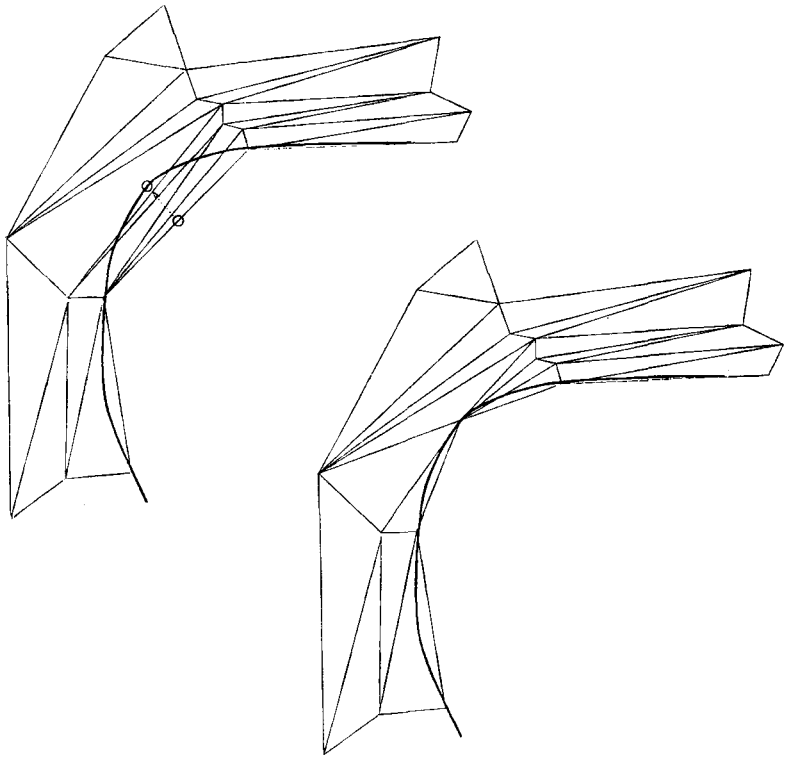


Figure 3. Illustration of adaptive boundary point insertion and local mesh restructuring in regions of high mesh stretching and high boundary curvature

ρ being the fluid density, u and v the Cartesian velocity components and E the internal energy. $\mathbf{F}_c$ represents the convective flux vector, the components of which are algebraic functions of the conserved variables and the pressure, which itself can be related to the conserved variables through the perfect gas relation. $\mathbf{F}_v$ denotes the viscous flux vector, the components of which are functions of the first derivatives of the conserved variables. Equation (1) represents a set of partial differential equations which must be discretized in space in order to obtain a set of coupled ordinary differential equations, which can then be integrated in time to obtain the steady state solution. Space discretization is performed using a Galerkin finite-element-type formulation. Multiplying equation (1) by a test function ϕ and intergrating over physical space yields

$$\frac{\partial}{\partial t}\iint_\Omega \phi w \,dx\,dy + \iint_\Omega \phi \nabla \cdot \mathbf{F}_c \,dx\,dy = \frac{1}{Re_\infty}\iint_\Omega \phi \nabla \cdot \mathbf{F}_v \,dx\,dy. \tag{3}$$

Integrating the flux integrals by parts and neglecting boundary terms gives

$$\frac{\partial}{\partial t}\iint_\Omega \phi w \,dx\,dy = \iint_\Omega \mathbf{F}_c \cdot \nabla\phi \,dx\,dy - \frac{1}{Re_\infty}\iint_\Omega \mathbf{F}_v \cdot \nabla\phi \,dx\,dy. \tag{4}$$

In order to evaluate the flux balance equations at a vertex P, ϕ is taken as a piecewise linear function which has the value unity at node P and vanishes at all other vertices. Therefore the

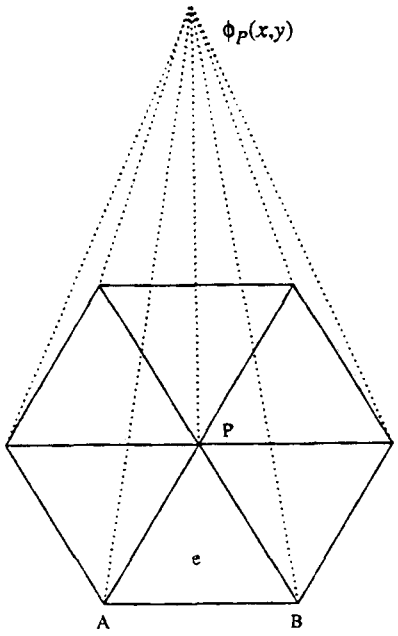


Figure 4. Domain of influence of finite element basis function and equivalent finite volume control volume

integrals in the above equation are non-zero only over triangles which contain the vertex P, thus defining the domain of influence of node P, as shown in Figure 4. To evaluate the above integrals, we make use of the fact that ϕ_x and ϕ_y are constant over a triangle and evaluate spatial derivatives of ϕ and w over a triangle using vertex values by Green's countour integral theorem. The convective fluxes $\mathbf{F}_c$ are taken as piecewise linear functions in space and the viscous fluxes $\mathbf{F}_v$ are piecewise constant over each triangle since they are formed from first derivatives in the flow variables. Evaluating the flux integrals with these assumptions, one obtains

$$\frac{\partial}{\partial t}\int\int_{\Omega}\phi w\,dx\,dy=\sum_{e=1}^{n}\frac{\mathbf{F}_c^A+\mathbf{F}_c^B}{6}\cdot\Delta\mathbf{L}_{AB}-\frac{1}{Re_\infty}\sum_{e=1}^{n}\frac{\mathbf{F}_v^e}{2}\cdot\Delta\mathbf{L}_{AB}, \tag{5}$$

where the summations are over all triangles in the domain of influence, as shown in Figure 4. $\Delta\mathbf{L}_{AB}$ represents the directed (normal) edge length of the face of each triangle on the outer boundary of the domain, $\mathbf{F}_c^A$ and $\mathbf{F}_c^B$ are the convective fluxes at the two vertices at either end of this edge and $\mathbf{F}_v^e$ is the viscous flux in triangle e, e being a triangle in the domain of influence of ϕ. If the integral on the left-hand side of equation (5) is evaluated in the same manner, the time derivatives become coupled in space. Since we are not interested in the time accuracy of the scheme but only in the final steady state solution, we employ the concept of a lumped mass matrix. This is equivalent to assuming w to be constant over the domain of influence while integrating the left-hand side. Hence we obtain

$$\Omega_P\frac{\partial w_P}{\partial t}=\sum_{e=1}^{n}\frac{\mathbf{F}_c^A+\mathbf{F}_c^B}{2}\cdot\Delta\mathbf{L}_{AB}-\frac{1}{Re_\infty}\sum_{e=1}^{n}\frac{3}{2}(\mathbf{F}_v^e\cdot\Delta\mathbf{L}_{AB}), \tag{6}$$

where the factor of 1/3 is introduced by the integration of ϕ over the domain, and Ω_P represents the surface area of the domain of influence of P. For the convective fluxes this procedure is equivalent to the vertex finite volume formulation described in References 4 and 5. For a smoothly varying regular triangulation the above formulation is second-order-accurate.

Additional artificial dissipation terms are required to ensure stability and to capture shocks without producing numerical oscillations. This is necessary for both inviscid and viscous flow computations, since in the latter case large regions of the flow field behave essentially inviscidly and the physical viscosity is not sufficient to guarantee numerical stability for the type of mesh spacings typically employed. Artificial dissipation terms are thus constructed as a blend of a Laplacian and a biharmonic operator in the conserved flow variables. The Laplacian term represents a strong, formally first-order-accurate dissipation which is turned on only in the vicinity of a shock, while the biharmonic term represents a weaker second-order-accurate dissipation which is employed in regions of smooth flow.[5,11] The spatially discretized equations are integrated in time to obtain the steady state solution using a five-stage time-stepping scheme, where the convective terms are evaluated at each stage within a time step and the dissipative terms (both physical and artificial) are only evaluated at the first, third and fifth stages. This particular scheme has been designed to maintain stability in regions where the flow is dominated by viscous effects and to rapidly dampen out high-frequency error components, which is an essential feature for a scheme intended to drive a multigrid algorithm. Convergence is accelerated by making use of local time stepping, implicit residual averaging and an unstructured multigrid algorithm.[11]

The idea of a multigrid strategy is to accelerate the convergence to steady state of a fine grid solution through corrections computed on coarser grids. An initial time step is performed on the fine grid and the flow variables and residuals are then transferred to the coarse grid. A correction equation is constructed on the coarse grid by adding a forcing function to the original discretized equations. This forcing function is formed by taking the difference between the transferred residuals and the residuals of the transferred variables, thus ensuring that the evolution of the coarse grid equations is driven by the fine grid residuals. Hence, when the fine grid residuals vanish, the coarse grid equations are identically satisfied and generate zero corrections. After transferring values down from the fine grid, a time step is performed on the coarse grid and the new values are transferred down to the next coarser grid. When the coarsest grid is reached, the computed corrections are successively interpolated back up to the finest grid and the entire cycle is repeated. In the context of unstructured meshes a sequence of coarse and fine meshes is best constructed by generating the individual meshes independently from one another (as opposed to subdividing a coarse mesh). Thus, in general, the coarse and fine meshes of a given sequence do not have any common mesh points or nested elements. Thus the patterns for transferring the variables, residuals and corrections back and forth between the various meshes of the sequence must be determined in a preprocessing operation, where an efficient tree search algorithm is employed.[5]

Such a multigrid algorithm may be combined with an adaptive meshing strategy in a natural manner. First, a sequence of globally generated meshes is constructed and multigrid time stepping is performed on this sequence until a satisfactorily converged solution is obtained. At this point a new adaptively refined mesh is generated and the transfer patterns for transferring variables from the previous mesh to the new mesh are determined. The flow variables are then transferred to this new mesh, providing a starting solution, and multigrid time stepping is resumed on this new sequence which now contains an additional fine mesh. The process may be repeated, as shown in Figure 5, each time adding a new finer mesh to the sequence until a converged solution of the desired accuracy is obtained.

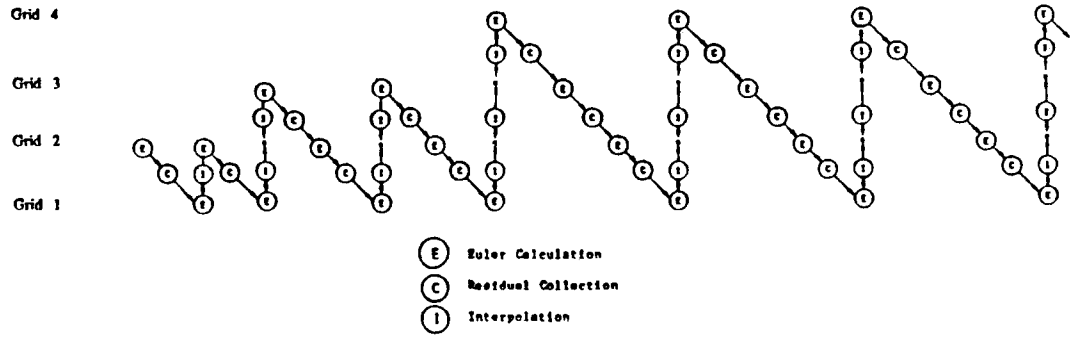


Figure 5. Full multigrid algorithm employed in conjunction with adaptive meshing strategy

4. TURBULENCE MODELLING

4.1. General procedure

The most widespread turbulence models in use currently are either of the multiple-field-equation type or of the algebraic type. While field equation models (such as the K–ε model) can be discretized and solved for on unstructured meshes in a straightforward manner, the solution of additional field equations can be considerably expensive, especially in near-wall regions where the equations may become very stiff numerically. Algebraic models, on the other hand, are relatively inexpensive to compute and have demonstrated generally superior accuracy and reliability for limited classes of problems, such as high-Reynolds-number attached flows over streamlined bodies. However, such models typically require information concerning the distance of each mesh point from the nearest wall. Turbulence length scales, which are related to the local boundary layer or wake thickness, are determined by scanning the appropriate flow values along specified streamwise stations. In the context of unstructured meshes, mesh points and thus flow variables do not naturally occur at regular streamwise locations. Thus lines normal to the walls and viscous layers must be created and flow variables interpolated onto these lines in order that turbulence length scales may be determined. This type of approach has previously been implemented for supersonic ramp geometries by Rostand.[16] However, in the present work, more complex geometries must be accommodated. A more sophisticated manner of generating normal mesh stations can be devised using structured hyperbolic mesh generation techniques.[13] Thus a distribution of normal mesh lines which do not cross over each other, and containing a smoothly varying normal distribution of points, can be obtained by generating a structured hyperbolic mesh about each geometry component, based on the boundary point distribution of the global unstructured mesh on each component.[12] These normal mesh lines are terminated if they intersect a neighbouring geometry component, thus ensuring that turbulence quantities in any given region of the flow field are only dependent on the viscous layers and walls which are directly visible from that location (compare Figures 7 and 12). At each time step in the flow solution phase, flow variables are interpolated onto the background turbulence mesh stations and the Baldwin–Lomax[17] algebraic model is employed to compute eddy viscosity values along these stations. The standard Baldwin–Lomax model must be modified slightly in order to enable the treatment of flows with multiple boundary layers and wakes. Thus the search for the maximum moment of vorticity, which is usually employed to determine a turbulence length scale, is limited to the region between the wall (or wake centreline) and the first point of vanishing vorticity off the

wall (or wake centreline). Since a change in sign of the vorticity occurs in the region between two confluent shear layers, turbulence length scales associated with vorticity fluctuations due to neighbouring shear layers are ignored in this manner. The eddy viscosities are then interpolated back onto the unstructured mesh for subsequent use in the flow solver. In regions where multiple background turbulence meshes overlap, the multiple eddy viscosity values (one from each mesh) interpolated back to the unstructured mesh are weighted by a factor proportional to the inverse of the distance from the corresponding wall, thus producing a smooth distribution of eddy viscosity throughout the flow field. Flow variables can be repeatedly interpolated back and forth between the background turbulence mesh stations and the global unstructured mesh by computing and storing the interpolation weights and addresses in a preprocessing operation prior to the flow solution phase. This is accomplished by first triangulating the point distribution of the local background turbulence meshes and then making use of the same efficient search routines employed in the unstructured multigrid transfer process.

4.2. Adaptive meshing procedure

When an adaptive meshing strategy is employed, the background turbulence meshes must be adapted in a manner analogous to the refinement of the global unstructured mesh. A refinement field variable is thus constructed which is assigned the value 1·0 in regions where the unstructured mesh is refined and 0·0 in the remainder of the flow field. This variable is then interpolated onto the background turbulence meshes and used to determine the regions which require refinement. Since these background meshes have previously been triangulated for interpolation purposes, they may be refined by inserting new points and restructuring locally using Bowyer's Delaunay triangulation algorithm[15] as described previously. However, these new points must be added in a manner which preserves the original structure (that of normal mesh stations) of the background meshes. Hence only new points along existing mesh stations are permitted, and when a new boundary point is inserted, an entire new turbulence mesh station extending up to the outer boundary of the background mesh is created.[12]

After each adaptation process the transfer patterns for interpolation between the newly refined global unstructured mesh and background meshes must be recomputed. In the context of the multigrid strategy the turbulence model is only executed on the finest grid of the sequence. The computed eddy viscosity values are then interpolated up to the coarser unstructured meshes where they are employed in the multigrid correction equations. The whole process is very efficient and, in general, the turbulence model is found to require less than 10% of the total time required within a single multigrid cycle. Memory requirements are, however, increased by close to 50%, since extra variables and transfer coefficients must be stored for the background turbulence mesh stations.

5. RESULTS

5.1. Single-aerofoil geometry

As an initial test case the turbulent flow over an RAE 2822 aerofoil has been computed. The freestream Mach number is 0·729, the Reynolds number is $6{\cdot}5 \times 10^6$, the corrected incidence is 2·31° and transition is fixed at 0·03 chords. This constitutes a well documented test case (Case 6) for turbulent transonic flow[18] which can be used to validate the present solver. The unstructured mesh employed for this case is depicted in Figure 6. This mesh contains 13 751 points, of which 210 are on the aerofoil surface. The average normal spacings of the triangles on the aerofoil

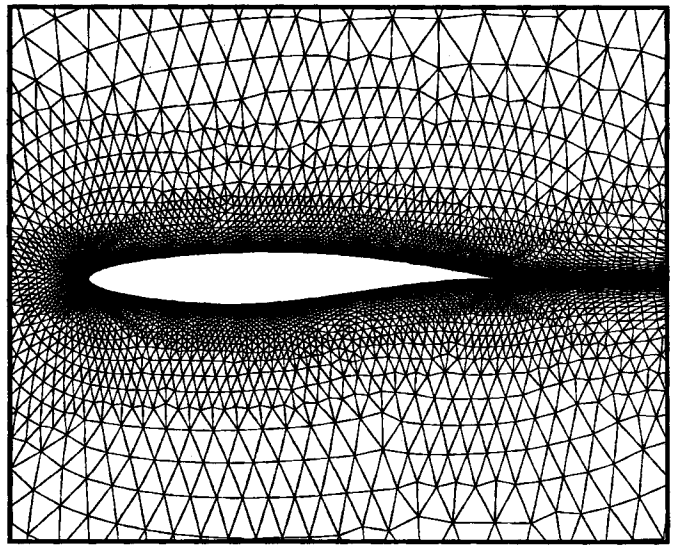


Figure 6. Fully unstructured mesh with high stretching employed for computing turbulent flow past an RAE 2822 aerofoil (number of points = 13 751)

surface is 0·00001 chords, resulting in cell aspect ratios of the order of 1000 : 1 near the wall. The background turbulence mesh stations employed for computing the algebraic turbulence model, which contain a total of 13 372 points, are depicted in Figure 7. The computed surface pressure distribution and skin friction distribution are displayed in Figures 8 and 9 respectively, where they are compared with experimental data from Reference 18. Both quantities are seen to compare favourably with the experimental results, and the computed lift coefficient of 0·7403 is

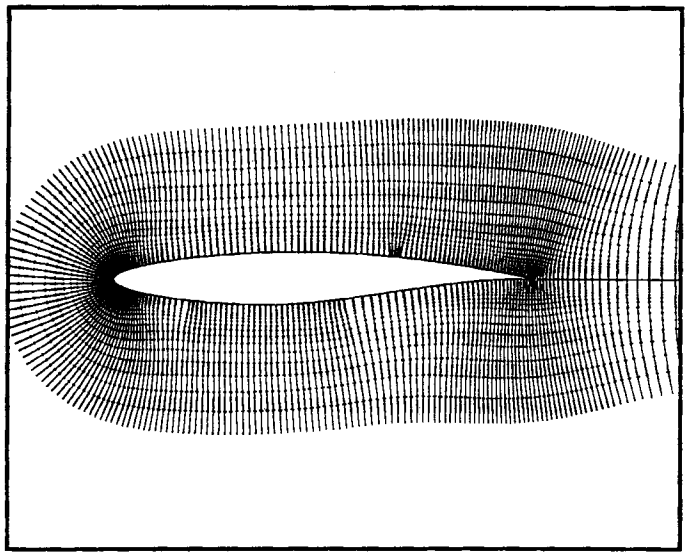


Figure 7. Turbulence stations employed for computing flow past an RAE 2822 aerofoil (number of points = 13 372)

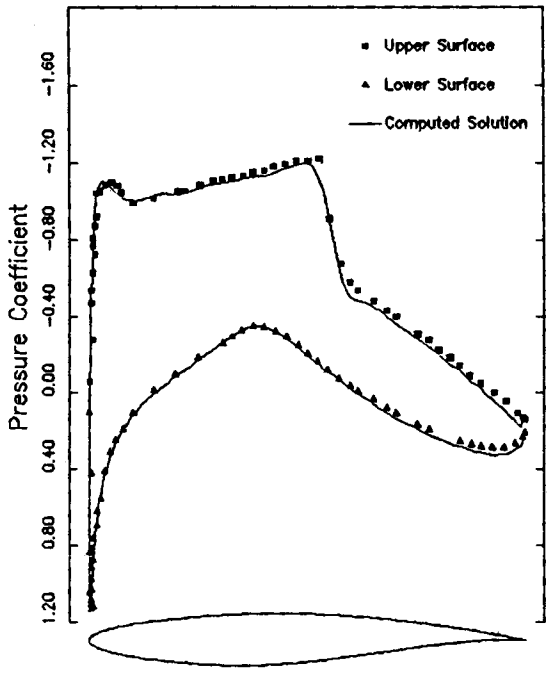


Figure 8. Comparison of computed surface pressure with experimental measurements for flow over an RAE 2822 aerofoil (Mach = 0·729, $Re = 6{\cdot}5 \times 10^6$, incidence = 2·31°)

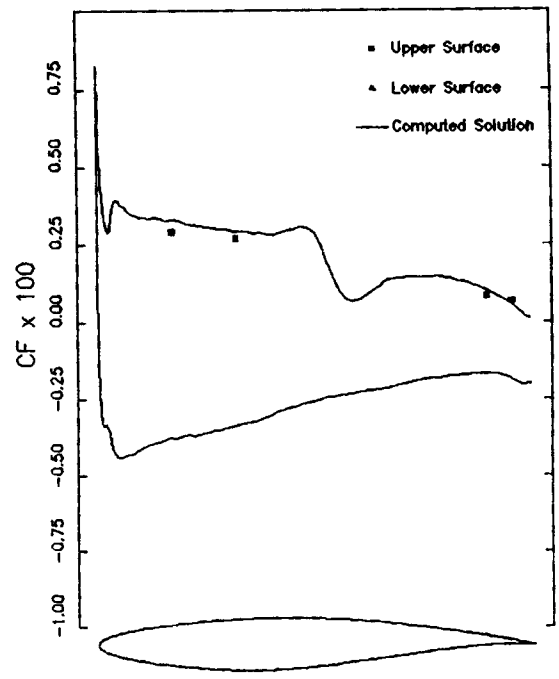


Figure 9. Comparison of computed skin friction with experimental measurements for flow over an RAE 2822 aerofoil (Mach = 0·729, $Re = 6{\cdot}5 \times 10^6$, incidence = 2·31°)

well within the range reported in previously published computational solutions using structured meshes.[19] A total of five meshes were employed in the multigrid sequence, with the coarsest mesh containing only 98 points. The convergence rate for this case, as measured by the decrease in the RMS average of the density residuals throughout the flow field, versus the number of multigrid cycles is depicted in Figure 10. An average residual reduction of 0·955 per multigrid cycle is achieved on the finest grid, resulting in a decrease of the residuals by four orders of magnitude over 200 cycles. Furthermore, the lift and drag coefficients were converged to four significant figures within 90 cycles. Since each multigrid cycle requires roughly 1·4 CPU seconds on a single processor of the CRAY-YMP computer, engineering solutions could thus be obtained in approximately 2 min for this case.

5.2. Two-element aerofoil geometry

The next test case consists of a main aerofoil with a leading edge slat, which has been the subject of extensive wind tunnel tests as part of a programme aimed at determining the effectiveness of slats as manoeuvring devices for fighter aircraft.[20] The test conditions consist of a freestream Mach number of 0·5, a chord Reynolds number of $4{\cdot}5 \times 10^6$ and a corrected incidence of 7·5°. Under these conditions the flow becomes supercritical and a small shock is formed on the upper surface of the slat. The adapted mesh used to compute this flow is depicted in Figure 11. The mesh contains a total of 35 885 points, of which 418 are on the surface of the main aerofoil and 421 on the surface of the slat. The minimum normal spacing at the wall is 0·00001

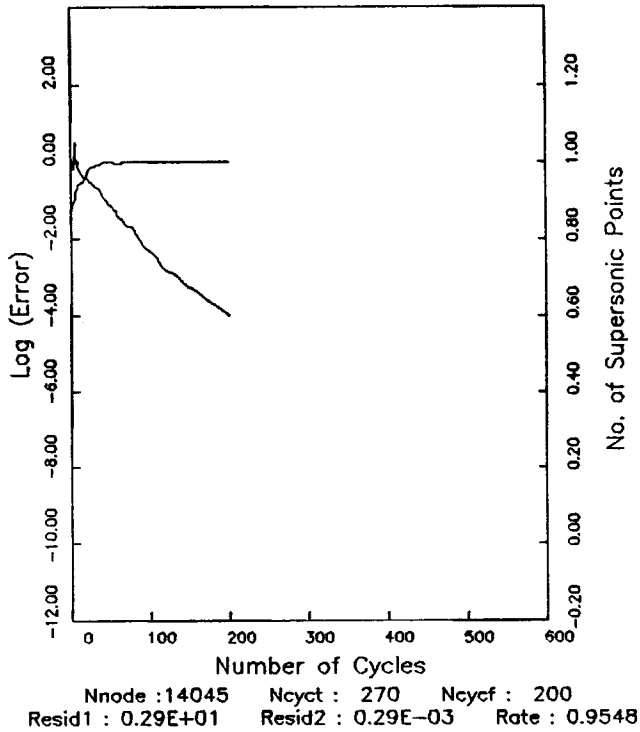
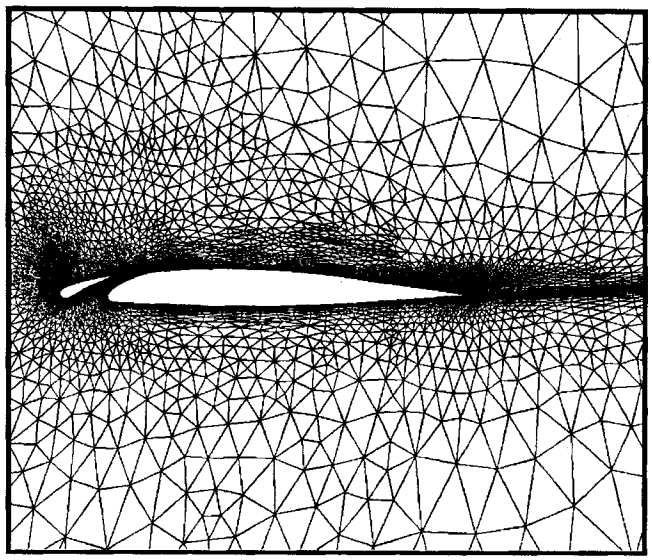
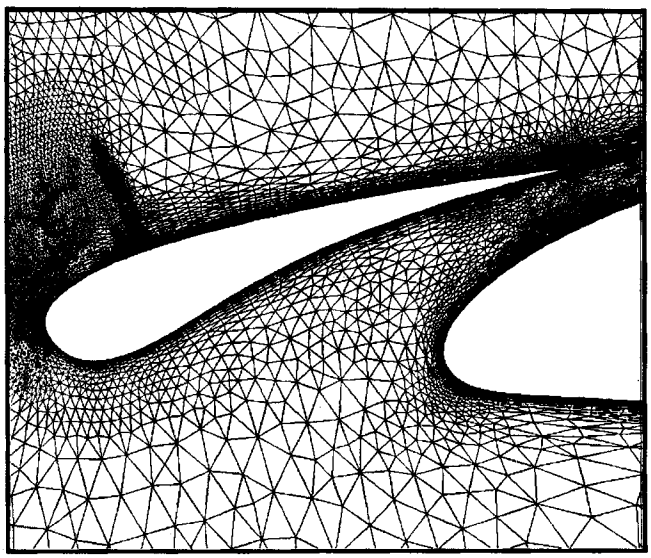


Figure 10. Convergence rate for flow over an RAE 2822 aerofoil as measured by the RMS average of the density residuals versus the number of multigrid cycles



(a)



(b)

Figure 11. Adaptively generated unstructured mesh about two-element aerofoil (number of nodes = 35 885)

chords and cells of aspect ratios up to 1000:1 are observed. A total of seven meshes were employed in the multigrid sequence, with the last three meshes generated adaptively. Figure 12 illustrates the adaptively generated turbulence mesh stations employed by the algebraic turbulence model for this case. The computed Mach contours in the flow field are depicted in Figure 13, where a crisp resolution of the small localized shock provided by the adaptive meshing technique

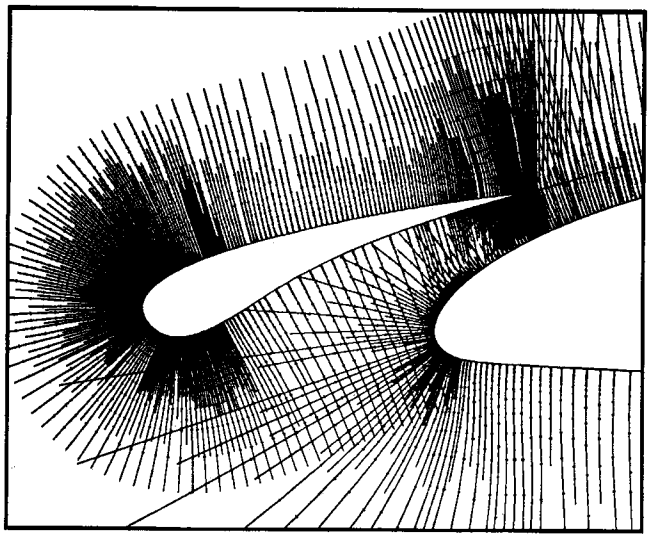
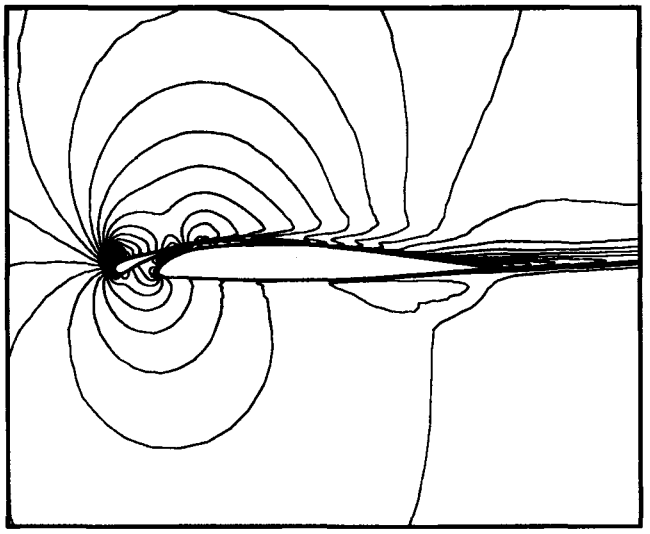


Figure 12. Illustration of turbulence mesh stations employed in algebraic model (number of points = 43 566)

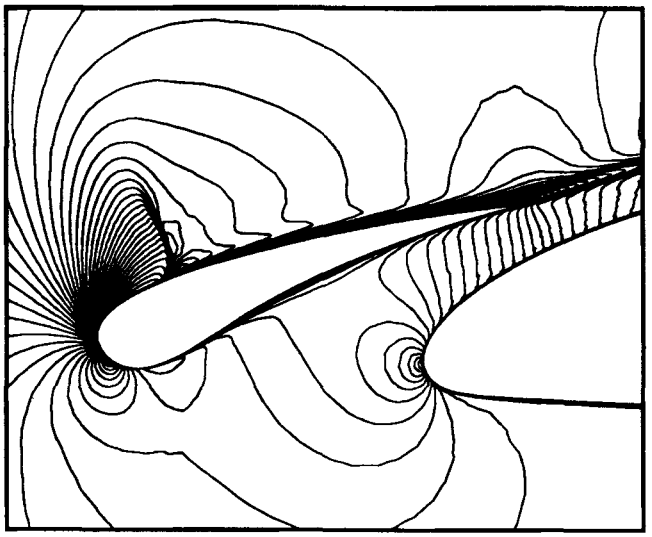
is observed. A good correlation between the computed and experimental surface pressure coefficients is displayed in Figure 14. This method predicts a small zone of separated flow behind the shock, which has not been detected in previous calculations,[20] while also providing a more accurate prediction of shock location and subsequent pressure recovery. The solution of this case required 15 min on a single processor of a CRAY-YMP, during which the fine grid residuals were reduced by three orders of magnitude over 200 multigrid cycles, as shown in the convergence history plot of Figure 15.

5.3. Four-element aerofoil geometry

The final test case consists of a four-element aerofoil configuration. This represents a truly complex geometry involving regions with sharp corners which is not easily amenable to structured mesh techniques and is of considerable practical interest since it relates to the design of high-lift devices for commercial aircraft. A multigrid sequence of six meshes was employed to compute the flow over this configuration, with the last two meshes generated adaptively. The finest mesh of the sequence, which contains a total of 48 691 points, is depicted in Figure 16. The mesh contains 243 points on the main aerofoil, 327 points on the leading edge slat, with 208 and 247 points located on the vane and flap respectively. The height of the smallest cells at the wall is of the order of 0·00002 chords for each aerofoil element and cell aspect ratios up to 500:1 are observed. The computed Mach contours for this case are depicted in Figure 17. The freestream Mach number is 0·1995, the chord Reynolds number is $1{\cdot}187 \times 10^6$ and the corrected incidence is 16·02°. Under these conditions the flow remains entirely subcritical. Compressibility effects are nevertheless important owing to the large suction peaks generated about each aerofoil. For example, in the suction peak on the upper surface of the leading edge slat the local Mach number achieves a value of 0·77. The computed surface pressure coefficients are compared with experimental wind tunnel data[21] in Figure 18. Good overall agreement, including the prediction of the height of the suction peaks, is observed. This case provides a good illustration of the importance of adaptive meshing in practical aerodynamic calculations. Adequate resolution of the strong



(a)



(b)

Figure 13. Computed Mach contours for flow over a two-element aerofoil configuration (Mach = 0·5, $Re = 4{\cdot}5 \times 10^6$, incidence = 7·5°)

suction peak on the upper surface of the slat can only be achieved with a very fine mesh resolution in this region. Failure to adequately capture this large suction peak results in the generation of numerical entropy, which is then convected downstream, thus contaminating the solution in the downstream regions and degenerating the global accuracy of the solution. Because these suction peaks are very localized, they are efficiently resolved with adaptive techniques. In order to obtain a similar resolution using global mesh refinement, of the order of 200 000 mesh points would be required, greatly increasing the cost of the computation. The convergence history for this case, as

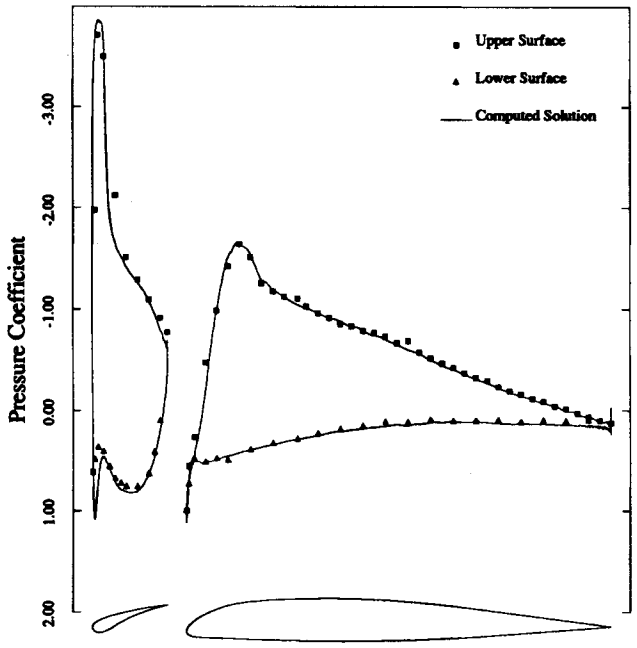


Figure 14. Comparison of computed surface pressure distribution with experimental wind tunnel data for flow over two-element aerofoil configuration (Mach = 0·5, $Re = 4{\cdot}5 \times 10^6$, incidence = 7·5°)

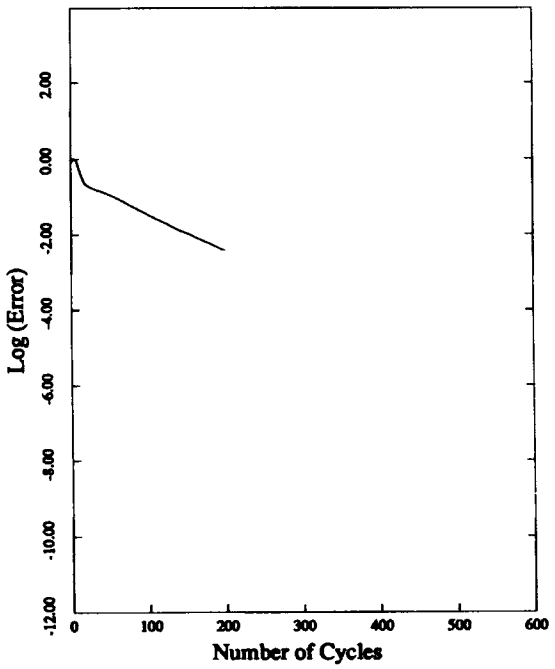
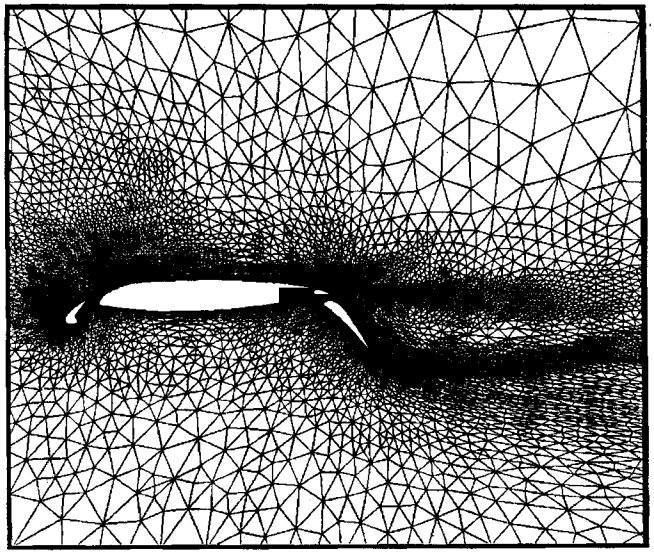
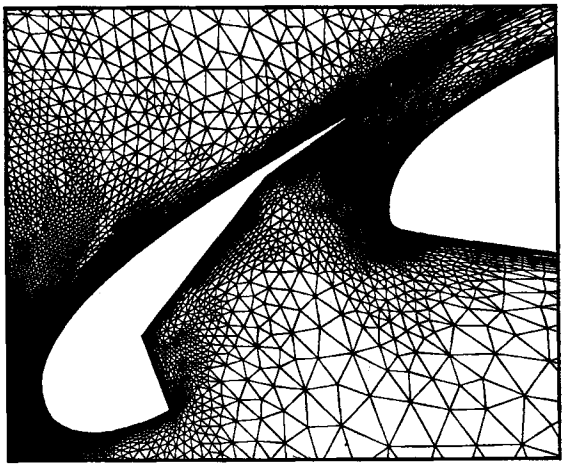


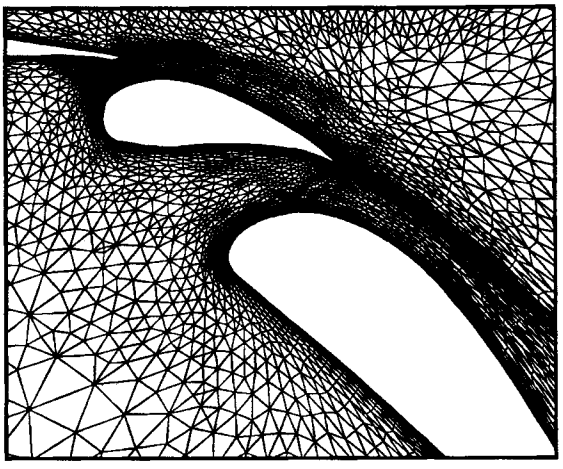
Figure 15. Convergence rate as measured by RMS average of the density residuals versus the number of multigrid cycles for flow past a two-element aerofoil



(a)



(b)



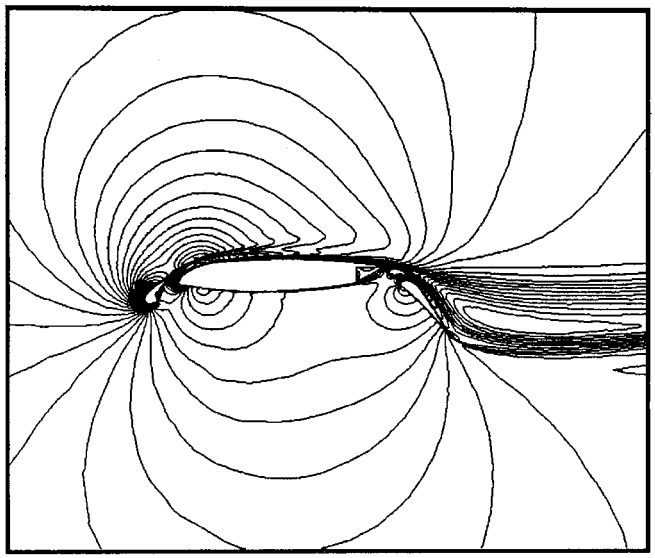
(c)

Figure 16. Adaptively generated unstructured mesh about four-element aerofoil (number of nodes = 48 691)

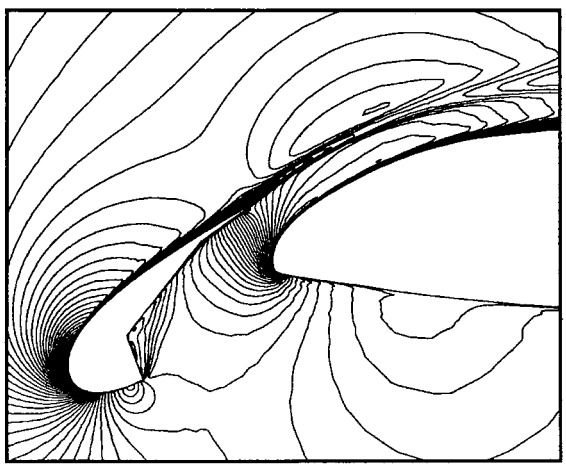
measured by the density residuals and the total lift coefficient versus the number of multigrid cycles, is depicted in Figure 19. A total of 400 multigrid cycles were executed, which required roughly 35 min of single-processor CRAY-YMP time and 14 Mwords of memory.

6. CONCLUSIONS

A method for efficiently computing turbulent compressible flows over complex two-dimensional configurations has been presented. By employing fully unstructured meshes throughout the entire flow field, arbitrary geometries can be handled and adaptive meshing techniques may be

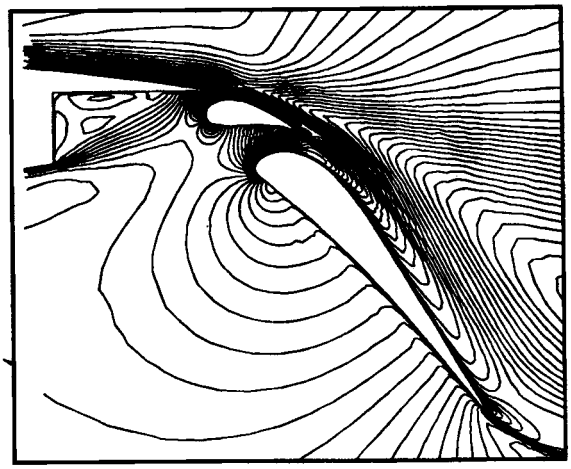


(a)



(b)

Figure 17. (Continued)



(c)

Figure 17. Computed Mach contours for flow over four-element aerofoil (Mach = 0·1995, $Re = 1{\cdot}187 \times 10^6$, incidence = 16·02°)

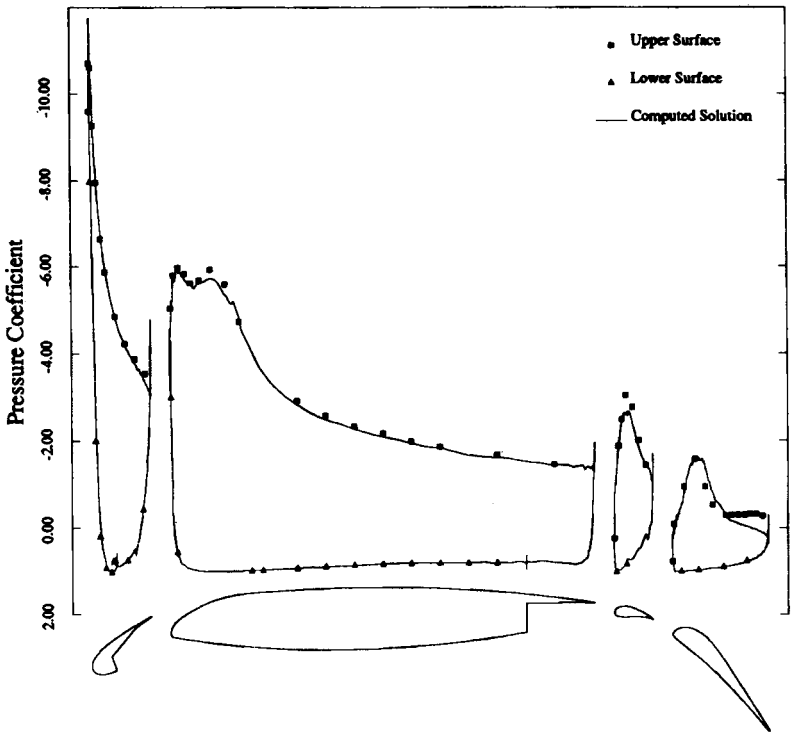


Figure 18. Comparison of computed surface pressure distribution with experimental wind tunnel data for flow over four-element aerofoil configuration (Mach = 0·1995, $Re = 1{\cdot}187 \times 10^6$, incidence = 16·02°)

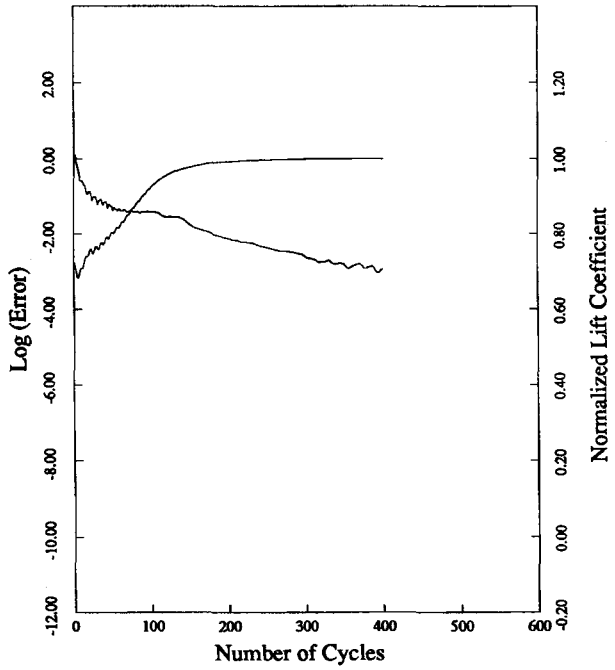


Figure 19. Convergence as measured by the computed lift coefficient and the density residuals versus the number of multigrid cycles for flow past a four-element aerofoil

extensively exploited. The method is efficient in that engineering solutions may generally be obtained in less than 100 multigrid cycles and the turbulence model requires less than 10% of the total time required to compute a solution. Unstructured mesh algorithms incur substantial overheads owing to the random nature of their datasets and generally run up to three times slower than their structured counterparts on present-day supercomputers owing to the indirect addressing and gather–scatter operations which must be employed. However, adaptive meshing techniques enable the accurate resolution of complex flow fields with a relatively small number of points, easily outweighting the penalties incurred by the use of random datasets. In the future an adaptivė meshing technique capable of modifying the local mesh stretching as well as the mesh point distribution should be pursued. Further turbulence-modelling research is also required, either refining the current algebraic model or implementing an alternative field equation model for cases with large regions of separated flows.

REFERENCES

1. T. J. Baker, 'Three dimensional mesh generation by triangulation of arbitrary point sets', *AIAA Paper 87-1124, Proc. AIAA 8th Computational Fluid Dynamics Conf.*, June 1987.
2. R. Lohner, P. Parikh and C. Gumbert, 'Interactive generation of unstructured grids for three dimensional problems', in S. Sengupta, J. Hauser, P. R. Eisman and J. F. Thompson (eds), *Proc. 2nd Int. Conf. on Numerical Grid Generation in Computational Fluid Dynamics*, Pineridge, Swansea, 1988, pp. 687–697.
3. J. Peraire, J. Peiro, L. Formaggia and K. Morgan, 'Adaptive numerical solutions of the Euler equations in 3-D using finite elements', in D. Dwoyer, Y. Hussaini and R. Voigt (eds), *Lecture Notes in Physics, Proc. 11th Int. Conf. on Numerical Methods in Fluid Dynamics*, Williamsburgh, VA, 1988, Springer, New York, 1988, pp. 469–473.
4. A. Jameson, T. J. Baker and N. P. Weatherill, 'Calculation of inviscid transonic flow over a complete aircraft', *AIAA Paper 86-0103*, January 1986.

5. D. J. Mavriplis, 'Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes', *AIAA J.*, **26**, 824–831 (1988).
6. T. J. Barth and D. C. Jespersen, 'The design and application of upwind schemes on unstructured meshes', *AIAA Paper 89-0366*, January 1989.
7. B. Stoufflet, J. Periaux, F. Fezoui and A. Dervieux, 'Numerical simulation of 3-D hypersonic Euler flows around space vehicles using adapted finite elements', *AIAA Paper 87-0560*, January 1987.
8. N. Nakahashi, 'FDM–FEM zonal approach for viscous flow computations over multiple bodies', *AIAA Paper 87-0604*, January 1987.
9. D. G. Holmes and S. Connell, 'Solution of the 2-D Navier–Stokes equations on unstructured adaptive grids', *AIAA Paper 89-1932, Proc. AIAA 9th Computational Fluid Dynamics Conf.*, Buffalo, NY, June 1989.
10. D. J. Mavriplis, 'Adaptive mesh generation for viscous flows using Delaunay triangulation', *J. Comput. Phys.*, in the press.
11. D. J. Mavriplis, A. Jameson and L. Martinelli, 'Multigrid solution of the Navier–Stokes equations on triangular meshes', *AIAA Paper 89-0120*, January 1989.
12. D. J. Mavriplis, 'Algebraic turbulence modeling for unstructured and adaptive meshes', *AIAA Paper 90-1653*, June 1990.
13. J. L. Steger and R. L. Sorenson, 'Use of hyperbolic partial differential equations to generate body-fitted grids', in R. E. Smith (ed.), *Numerical Grid Generation Techniques, NASA CP-2166*, 1980.
14. L. B. Wigton, The Boeing Company, private communication.
15. A. Bowyer, 'Computing Dirichlet tessalations', *Comput. J.*, **24**, 162–166 (1981).
16. P. Rostand, 'Algebraic turbulence models for the computation of two-dimensional high speed flows using unstructured grids', *Int. j. numer. methods fluids*, in the press.
17. B. S. Baldwin and H. Lomax, 'Thin layer approximation and algebraic model for separated turbulent flows', *AIAA Paper 78-275*, 1978.
18. P. H. Cook, M. A. McDonald and M. C. P. Firmin, 'Aerofoil RAE 2822 pressure distributions and boundary layer and wake measurements', *AGARD Advisory Report No. 138*, May 1979.
19. L. Martinelli and A. Jameson, 'Validation of a multigrid method for the Reynolds averaged equations', *AIAA Paper 88-0414*, January 1988.
20. G. Volpe, 'A multigrid method for computing the transonic flow over two closely-coupled airfoil components', *ICAS Paper 84-1.4.3, 14th ICAS Congr.*, Toulouse, September 1984.
21. L. B. Wigton, The Boeing Company, private communication.