INT. J. NUM. METH. HEAT FLUID FLOW, VOL. 3, 283–304 (1993)

AN UPWIND UNSTRUCTURED GRID SOLUTION ALGORITHM FOR COMPRESSIBLE FLOW

S. SOLTANI*, K. MORGAN† AND J. PERAIRE*

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

ABSTRACT

An upwind unstructured grid cell-centred scheme for the solution of the compressible Euler and Navier–Stokes equations in two dimensions is presented. The algorithm employs a finite volume formulation. Calculation of the inviscid fluxes is based on the approximate Riemann solver of Roe. Viscous fluxes are obtained from solution gradients computed by a variational recovery procedure. Higher order accuracy is achieved through performing a monotonic linear reconstruction of the solution over each cell. The steady state is obtained by a point implicit time integration of the unsteady equations using local time stepping. For supersonic inviscid flow an alternative space marching algorithm is proposed. This latter approach is applicable to supersonic flow fields containing regions of local subsonic flow. Numerical results are presented to show the performance of the proposed scheme for inviscid and viscous flows.

KEY WORDS Upwind finite volume scheme Euler equations Unstructured grids Compressible flow Navier–Stokes equations

INTRODUCTION

Over the past decade, there has been extensive work on the application of upwinding techniques to the solution of the compressible flow equations. The popularity of upwinding is due to the fact that, in one dimension, it models the underlying physics of the problem better than more conventional centred methods. A generalized formulation of upwind schemes for the Euler equations is given in Reference 1. Amongst the most accurate upwind formulations are the schemes of Roe[2], and of Osher[3]. Although most practical implementations to date have utilized structured computational grids, recently attention has been given to the use of unstructured grids[4–7]. This interest is due to the flexibility that they offer for mesh generation and adaptivity. The favourable properties of upwinding and unstructured grids has lead to an increasing use of the combined methodologies[8–14].

In this paper, we introduce a higher order upwind unstructured grid cell-centred scheme that is capable of solving the two dimensional equations of compressible flow over a wide range of Mach numbers. The layout of the paper is as follows. The governing equations are the laminar compressible Navier–Stokes equations, and these are presented in the next section. The basic finite volume discretization is then described, and the extension to higher order spatial accuracy is explained in the following section. The implicit time integration of the discretized set of equations is also presented, and a space marching approach for the solution of supersonic inviscid flows is described later. A brief discussion of the numerical treatment of the boundary conditions is given followed by numerical results.

0961–5539/93/040283–22$2.00

Received May 1992
Revised February 1993

GOVERNING EQUATIONS

The equations to be solved are the Euler equations, governing the flow of an inviscid compressible fluid, and the full Navier–Stokes equations, governing laminar flow of a compressible viscous fluid. These equations can be expressed in conservation form as:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}_i}{\partial x_i}=\frac{\partial \mathbf{G}_i}{\partial x_i} \qquad i=1, 2 \tag{1}$$

where $\mathbf{U}$ is the vector of unknowns, and $\mathbf{F}_i$ and $\mathbf{G}_i$ represent the inviscid and viscous fluxes respectively in the direction x_i of a Cartesian coordinate system Ox_1x_2. In a non-dimensionalized form[5], these vectors can be written as:

$$\mathbf{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho \varepsilon\end{pmatrix} \quad \mathbf{F}_1=\begin{pmatrix}\rho u\\ \rho u^2+P\\ \rho uv\\ u(\rho\varepsilon+P)\end{pmatrix} \quad \mathbf{F}_2=\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+P\\ v(\rho\varepsilon+P)\end{pmatrix}$$

$$\mathbf{G}_1=\frac{1}{Re_\infty}\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \dfrac{\mu}{Pr}\dfrac{\partial T}{\partial x}+u\tau_{xx}+v\tau_{xy}\end{pmatrix} \quad \mathbf{G}_2=\frac{1}{Re_\infty}\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \dfrac{\mu}{Pr}\dfrac{\partial T}{\partial y}+u\tau_{xy}+v\tau_{yy}\end{pmatrix} \tag{2}$$

where ρ, u, v, ε, P and T represent the density, Cartesian velocity components, specific total energy, pressure and temperature of the fluid respectively. In addition, Re_∞ is the free stream Reynolds number and Pr is the Prandtl number, which is assigned the constant value of 0.72. For a laminar flow, the viscous stresses are related to the velocity gradients according to:

$$\tau_{xx}=2\mu\frac{\partial u}{\partial x}+\lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) \tag{3a}$$

$$\tau_{xy}=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \tag{3b}$$

$$\tau_{yy}=2\mu\frac{\partial v}{\partial y}+\lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) \tag{3c}$$

where it is assumed that $\lambda=-2\mu/3$. The coefficient of viscosity is obtained from Sutherland's law, which in non-dimensionalized form can be written as:

$$\mu=\frac{T_\infty+S}{T+S}\left(\frac{T}{T_\infty}\right)^{3/2} \tag{4}$$

where S is an experimental constant, which for air is taken to be 110 K. In the present work it is assumed that the fluid is a calorically perfect gas with the state equation:

$$P=(\gamma-1)\rho\left(\varepsilon-\frac{1}{2}(u^2+v^2)\right) \tag{5}$$

where $\gamma=c_P/c_v$ is the ratio of specific heats, which for air is taken to be 1.4.

BASIC FINITE VOLUME ALGORITHM

The governing equations are considered in an integral conservation form. For a region Ω with its boundary Γ one can write

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\,d\Omega = \int_\Omega \left(-\frac{\partial \mathbf{F}_i}{\partial x_i}+\frac{\partial \mathbf{G}_i}{\partial x_i}\right) d\Omega \qquad i=1, 2 \tag{6}$$

and application of the divergence theorem results in:

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\,d\Omega = \int_\Gamma (-\mathbf{F}_i+\mathbf{G}_i)n_i\,d\Gamma \tag{7}$$

where $\mathbf{n}=(n_1, n_2)$ is the outward unit vector normal to the boundary. It is assumed that the computational domain is subdivided into a general assembly of either triangular or quadrilateral cells, or a combination of both. The above equation is then satisfied over each cell (control volume) in turn. Assuming a piecewise constant distribution of the unknowns, (7) for a cell c can be approximated as:

$$\frac{\Delta \mathbf{U}_c}{\Delta t_c}=\frac{1}{\Omega_c}\int_{\Gamma_c} (-\mathbf{F}_n+\mathbf{G}_n)\,d\Gamma \tag{8}$$

where $\mathbf{F}_n$ and $\mathbf{G}_n$ represent the normal inviscid and viscous fluxes respectively to the cell sides, and $\Delta t_c=t_c^{m+1}-t_c^m$ is a time step. The actual fluxes are replaced by numerical fluxes and the integral is evaluated using a one point integration over the sides of the cell, resulting in:

$$\Delta \mathbf{U}_c = -\frac{\Delta t_c}{\Omega_c}\sum_{s_c} [\tilde{\mathbf{F}}_{ns}-\tilde{\mathbf{G}}_{ns}]\,\delta_{sc} \tag{9}$$

where δ_{sc} denotes the length of the side s_c. The calculation of the inviscid and the viscous contributions to the right hand side of equation (9) will now be separately considered.

Inviscid fluxes

The inviscid fluxes are calculated using the approximate Riemann solver of Roe[2]. This leads to a crisp representation of shock waves. Moreover, the numerical flux of Roe has been shown to be suitable for accurate modelling of shear layers[15]. This numerical flux, in terms of left and right hand states, can be expressed as:

$$\tilde{\mathbf{F}}_n(\mathbf{U}_L, \mathbf{U}_R)=\frac{1}{2}(\mathbf{F}_n(\mathbf{U}_L)+\mathbf{F}_n(\mathbf{U}_R)-|\mathbf{A}_n(\mathbf{U}_L, \mathbf{U}_R)|(\mathbf{U}_R-\mathbf{U}_L)) \tag{10}$$

where $\mathbf{A}_n=\partial \mathbf{F}_n/\partial \mathbf{U}$ is the mean jacobian matrix which is defined in such a way as to ensure correct jump conditions across discontinuities. The matrix $|\mathbf{A}|$ is defined by:

$$|\mathbf{A}|=\mathbf{R}|\Lambda|\mathbf{R}^{-1} \tag{11}$$

where $\mathbf{R}$ is the matrix of the right eigenvectors of $\mathbf{A}$ and Λ is the diagonal matrix of the eigenvalues of $\mathbf{A}$. The full definition of the entries in these matrices is given in Reference 16. If an eigenvalue of $\mathbf{A}$ vanishes, the numerical flux of Roe may lead to non-physical expansion shocks[1]. Following the remedy of Reference 17, this can be prevented by replacing any diagonal element of $|\Lambda|$ which is less than a threshold, ε_λ, by:

$$\mathrm{diag}(|\Lambda|)=\frac{1}{2}\left(\frac{\lambda^2}{\varepsilon_\lambda}+\varepsilon_\lambda\right) \tag{12}$$

where ε_λ is taken to be a small fraction of the maximum eigenvalue.

Viscous fluxes

The gradients of the flow variables which are required in the calculation of the viscous terms are obtained from a variational recovery procedure[9]. A piecewise linear distribution is assumed for the gradients, while the variables are represented by a piecewise constant approximation. Thus, for any flow variable f we can write:

$$f=\sum_c f_c P_c \tag{13a}$$

$$\frac{\partial f}{\partial x_i}=\sum_K \left(\frac{\partial f}{\partial x_i}\right)_K N_K \qquad i=1, 2 \tag{13b}$$

where P_c is the piecewise constant shape function associated with cell c and N_K is the piecewise linear or bilinear shape function associated with vertex K. The nodal gradients are determined from the weighted residual requirement that:

$$\int_\Omega \frac{\partial f}{\partial x_i} N_K \, d\Omega=\int_\Gamma n_i f N_K \, d\Gamma-\int_\Omega f \frac{\partial N_K}{\partial x_i} \, d\Omega \tag{14}$$

Substituting the assumed form of the approximations from (13) into the above equation results in:

$$\left(\frac{\partial f}{\partial x_i}\right)_K=\frac{1}{(M_L)_K}\left[\frac{1}{2}\sum_\Gamma n_i f_\Gamma \delta_\Gamma-\sum_c \int_{\Omega_c} f_c \frac{\partial N_K}{\partial x_i} \, d\Omega\right] \tag{15}$$

for the values of gradients at a cell vertex K[16]. Here M_L denotes the standard lumped mass matrix and the summation over c extends over all the cells surrounding the vertex K. The boundary term in (15) need only be calculated when K lies on the boundary of the computational domain. This approach reproduces the exact gradient, on an arbitrary grid, when the variable f varies linearly.

HIGHER ORDER SPATIAL ACCURACY

The upwind algorithm described in the previous section is only first order accurate in space, which is not sufficient for practical calculations. Higher order accuracy may be achieved through the procedure proposed originally in Reference 11. Using the cell-averaged values a linear reconstruction of the variables is performed in a manner that ensures monotonicity. The left and right hand states, required in the evaluation of the inviscid fluxes in (10), are then determined using these reconstructed values.

The linear representation of a scalar variable f over an arbitrary cell c can be expressed as:

$$f(x, y)=f(x_o, y_o)+\mathbf{r}\cdot\nabla f \tag{16}$$

where $\mathbf{r}$ is the position vector of the point (x, y) with respect to a reference point o. The cell gradients can be determined from the variational recovery procedure[9] described earlier. If the gradient is assumed to be constant over the cell, and if the centroid of the cell is chosen as the reference point, the linear reconstruction given by (16) will be conservative in the sense that:

$$f^a=\frac{1}{\Omega_c}\int_{\Omega_c} f(x, y) \, d\Omega \tag{17}$$

where f^a is the cell-averaged (piecewise constant) value of f over the cell. The linear reconstruction given by (16) may generally exhibit non-physical oscillations near any flow discontinuity. To prevent the appearance of such oscillations, a limiter is applied to the higher order correction

term in the form:

$$f(x, y)=f(x_o, y_o)+\phi \mathbf{r}\cdot\nabla f \tag{18}$$

The cell limiter ϕ is determined in such a way that the value of f over the cell does not exceed the extrema of the cell-averaged values of f in the surrounding cells. Determination of a suitable limiter requires an initial evaluation of the maximum and minimum allowable values of f over each cell. These are obtained as:

$$f^{\max}=\max(f_c, f_i) \tag{19a}$$

$$f^{\min}=\min(f_c, f_i) \tag{19b}$$

where i runs over the surrounding cells. For a linear reconstruction, it is clear that the extrema will occur at cell vertices. The limiter for vertex K may be determined from an application of the defined monotonicity condition as:

$$\phi_K=\begin{cases} \min\left(1, \dfrac{(f^{\min}-f^a)}{(f^K-f^a)}\right); & f^K<f^a \\ \min\left(1, \dfrac{(f^{\max}-f^a)}{(f^K-f^a)}\right); & f^K>f^a \\ 1; & f^K=f^a \end{cases} \tag{20}$$

The limiter for a cell is then obtained as the minimum of the limiters calculated at its vertices. To ensure monotonicity for variables such as density and pressure, the linear reconstruction is performed on the basis of primitive variables ρ, u, v, and P. The resulting scheme has been found to be more stable than that which follows from using the conservation variables in the limiting procedure.

It should be pointed out that due to the limiting procedure involved the higher order scheme is not guaranteed to be globally second order accurate. However, the results of numerical experiments, presented here and in Reference 16, have shown a marked improvement for the higher order scheme over the first order scheme.

IMPLICIT FORMULATION

An evaluation of all the terms on the right hand side of (9) at time level $m+1$ results in a system of the general form:

$$\mathbf{K}\,\Delta\mathbf{U}=\mathbf{f} \tag{21}$$

A point relaxation algorithm similar to the type proposed by Gnoffo[18], and by Stoufflet *et al.*[8] is used to solve the above system of equations. Matrix **K** in (21) is decomposed into a matrix **D** containing the diagonal elements of **K**, and a matrix **E** containing the off-diagonal elements of **K**. The system of equations now takes the form:

$$(\mathbf{D}+\mathbf{E})\,\Delta\mathbf{U}=\mathbf{f} \tag{22}$$

The solution of this equation system is achieved by an iterative procedure, using the recursive relationship:

$$\Delta\mathbf{U}^{k+1}=\mathbf{D}^{-1}(\mathbf{f}-\mathbf{E}\,\Delta\mathbf{U}^k) \tag{23}$$

This is equivalent to a point Gauss–Seidel scheme, which requires only the solution of a 4 by 4 system of equations for each cell. It should be pointed out that the implicit formulation of Reference 9 essentially follows (23) but performs only one iteration per time step. Application

of (23) with a sufficient number of iterations results in faster convergence towards the steady state. To avoid the problem of using too much computer memory, the entries in the sparse matrix E are recalculated at each iteration. Due to the complexity of evaluating the viscous contribution to the matrix E, the off-diagonal viscous contributions to this matrix are ignored. This results in the same formulation as in Reference 9.

For the numerical flux of Roe, (9) in its fully implicit form can be expanded to give:

$$\Delta \mathbf{U}_c = -\frac{\Delta t_c}{\Omega_c} \sum_{s_c} \left\{ \frac{1}{2} [\mathbf{F}_{nc}^{m+1} + \mathbf{F}_{nr}^{m+1} - |\mathbf{A}_{ns}^{m+1}|(\mathbf{U}_r^{m+1} - \mathbf{U}_c^{m+1})] - \tilde{\mathbf{G}}_{ns}^{m+1} \right\} \delta_{sc} \tag{24}$$

where subscripts c and r represent the current cell and its surrounding cells respectively. Following the approach of Reference 9, the values of the unknowns and the fluxes at time level $m+1$ are linearized and written in terms of the values at time level m as:

$$\mathbf{U}_c^{m+1} = \mathbf{U}_c^m + \Delta \mathbf{U}_c \tag{25a}$$

$$\mathbf{F}_{nc}^{m+1} = \mathbf{F}_{nc}^m + \mathbf{A}_{nc}^m \Delta \mathbf{U}_c \tag{25b}$$

$$\tilde{\mathbf{G}}_{ns}^{m+1} = \tilde{\mathbf{G}}_{ns}^* + \mathbf{B}_{ns}^* \Delta \mathbf{U}_c \tag{25c}$$

Here, values of the unknowns at the current cell are considered at time level $m+1$, while the recent available values, indicated by an asterisk, are used for the surrounding cells. Details of the evaluation of the jacobian matrix $\mathbf{B}_{ns}$ for the implicit treatment of the viscous terms can be found in Reference 16. Substituting the expressions for the linearized terms from (25) into (24), and re-arranging results in:

$$\left\{ \mathbf{I} + \frac{\Delta t_c}{2\Omega_c} \sum_{s_c} [|\mathbf{A}_{ns}^*| - 2\mathbf{B}_{ns}^*] \delta_{sc} \right\} \Delta \mathbf{U}_c = -\frac{\Delta t_c}{2\Omega_c} \sum_{s_c} \{ [\mathbf{F}_{nc}^m + \mathbf{F}_{nr}^* - |\mathbf{A}_{ns}^*|(\mathbf{U}_r^* - \mathbf{U}_c^m)] - \tilde{\mathbf{G}}_{ns}^* \} \delta_{sc} \tag{26}$$

Owing to the expected large variation in size of the cells through the mesh, local time stepping is used. This ensures a constant Courant number throughout the mesh. Equation (26) can be applied independently to each cell. Sweeping is normally done in the numbered order through the cells in one time step and in the reverse order for the next step. This results in a random sweeping which depends solely on the mesh numbering. Sweeping can be done according to a user-specified criterion. This can be achieved by renumbering the cells according to the distance from a certain point or line. The effectiveness of the directional sweeping is most obvious for supersonic flow fields[14].

The implicit formulation presented here is based on the first order scheme. The higher order version of the scheme can be made implicit by considering the reconstruction given by (18) at time level $m+1$. Evaluation of the higher order correction terms, especially the limiter ϕ^{m+1}, proves to be difficult and time consuming. Therefore, in practice these terms are determined at time level m and there will be no contribution from the higher order terms to the left hand side of the system of equations. Hence, for the higher order implicit formulation, the left hand side matrix is the same as for the first order scheme. It should be mentioned that this time lagging of the higher order correction terms will introduce some degree of explicitness into the formulation. Therefore, a reduction in the maximum allowable time step can be expected. In practice this reduction in time step is problem dependent and in the most severe cases the time step used for the higher order scheme may be required to be one tenth of that used for the first order scheme.

SPACE MARCHING

It is well known that, in a supersonic flow field, disturbances are carried downstream. This physical property can be employed to devise efficient algorithms for the numerical solution of

supersonic steady state problems. The class of schemes based on this line of reasoning are known as space marching schemes[19–21]. Here, the domain of interest is divided into planes which are nearly normal to the direction of the flow. Since the solution at each of these planes (stations) depends only on the solution in the upstream stations, the steady state equations can be solved completely for each plane and the solution marched downstream. Hence, the solution can be obtained by performing only one sweep over the computational domain.

Formulation of space marching for the Euler equations

For many practical high speed flight situations, where the body is slender, the direction of the flow around the body does not vary drastically from the free stream direction. For these cases, the domain of interest can be subdivided into a set of planes which are roughly normal to the direction of the flow. The mesh within each of these planes can be completely unstructured.

The equations which are considered are the steady state Euler equations. For realistic flows, even when the free stream Mach number is much higher than unity, regions of subsonic flow can be expected, e.g. in the vicinity of stagnation points. In these subsonic regions (pockets), disturbances are propagated in all directions. As a result, a space marching technique cannot be applied without modification for the subsonic parts of the flow field. An efficient procedure can be obtained by combining the use of space marching in the prevailing supersonic regions of the flow, and time marching in the subsonic regions. In order to have a unified formulation for both supersonic and subsonic regions, the equations are considered in their transient form. The numerical discretization is basically achieved by a finite volume formulation similar to that described earlier.

In the fully supersonic regions of the flow field, starting from the inflow, the time dependent equations can be solved for each marching plane in turn. The finite volume formulation can be written as:

$$\frac{\Delta \mathbf{U}_c}{\Delta t_c} = -\frac{1}{\Omega_c} \sum_{s_c} \tilde{\mathbf{F}}_{ns} \delta_{sc} \tag{27}$$

If a side has both its neighbouring cells in the marching plane then the numerical fluxes, denoted by $\tilde{\mathbf{F}}_{ns}$, are obtained from (10). For the sides that lie on the boundaries of the marching plane the numerical flux is simply calculated using the upwind cell values, and application of an approximate Riemann solver is unnecessary. As long as the component of the flow velocity normal to the marching plane remains supersonic, this procedure is correct. By choosing a large time step (e.g. of the order of 10^6) the transient term in (27) becomes negligible and the formulation leads to an iterative procedure for the steady state equations. The system of equations resulting from a discretization of (27) is solved implicitly, in the marching plane, according to the process which was explained above. It should be noted that the sweeping is now only performed over the number of cells in the marching plane, and not over the total number of cells, thereby achieving fast convergence for fully supersonic flows. Once the solution is obtained for a marching plane, the same procedure is repeated for the next plane downstream. In this manner one sweep over the domain will result in the steady state solution being obtained.

For supersonic flow with subsonic pockets, the problem is to track down and identify the subsonic regions and to solve the equations in those regions by using a time marching scheme. The subsonic regions are identified by using a small CFL number in the first sweep in each plane. The reason for using a small Courant number is that if the flow at steady state in a particular plane is subsonic, a relaxation procedure must be used and this calls for a sensible choice of the time step size and a consideration of downstream conditions. If the local Mach number in all cells of the current marching plane is supersonic, the calculations will proceed downstream. If a Mach number of less than unity is encountered, the current plane is recorded as pertaining to the relaxation region. The plane recorded as being subsonic and its two adjacent

planes are taken to be the relaxation region and the solution in these three planes is time marched until convergence to steady state. If the solution in any of the two adjacent planes becomes subsonic, the relaxation region is extended by one plane in the direction where subsonic flow were detected and time marching is resumed in the updated relaxation region. In this manner the relaxation region can grow in both upstream and downstream directions until the boundary of the subsonic pocket reaches its full extension. When full convergence in the subsonic region is achieved, the space marching procedure is continued downstream.

The extension to higher order accuracy follows from the process described above. There are, however, some modifications necessary to comply with the requirements of the space marching technique. Evaluation of the gradients has to be modified to use only the physically relevant information. Consider the marching plane *mp*, where the flow is fully supersonic, and its neighbouring planes. From the cell-averaged values of a variable f, the corresponding value at a vertex K is calculated from:

$$f_K = \frac{\sum_{c \in mp \text{ or } mp-1} \Omega_c f_c^a}{\sum_{c \in mp \text{ or } mp-1} \Omega_c} \tag{28}$$

It should be noted that here the summation extends only over the cells in the marching plane and its neighbouring upstream plane. If the plane is in the subsonic region, the summation takes place over all the surrounding cells and the formulation will be similar to a variational recovery procedure[9]. The cell gradients follow from the use of the standard linear or bilinear shape functions together with the calculated vertex values.

Mesh generation and adaptivity for space marching

For space marching schemes, the process of mesh generation requires some consideration. It was mentioned in the introduction that unstructured grids provide great flexibility in handling complex realistic geometries. However, space marching techniques, due to their design, require a degree of structure in the mesh. Here, a compromise is arrived at by using grids which exhibit a structure in the direction of the flow but are otherwise unstructured. McGrory *et al.*[22] have reported a similar approach.

One of the main advantages of using unstructured grids is the ease with which mesh refinement can be carried out. The mesh refinement procedure is more restricted for space marching calculations, since the structure of the marching planes must be preserved. Local mesh enrichment has the property that the underlying structure of the mesh does not change and so is ideal for the present purpose. The mesh enrichment method used here follows the ideas presented by Morgan and Peraire[5].

BOUNDARY CONDITIONS

No special treatment is needed for the inflow and outflow boundaries. The approximate Riemann solver automatically selects the appropriate boundary conditions according to the sign of the characteristic velocities.

For the Euler equations, the physical boundary condition at the slip wall is that the normal component of the velocity to the wall is zero. To help achieve this, a set of imaginary cells is introduced inside the wall boundary. The values of the velocity components for these cells are set so that the average interface velocities satisfy the tangency condition. For computational purposes, the values of two other variables (density and pressure) must also be specified at the imaginary cells inside the wall. These values are taken to be the same as the values inside the

domain. For problems in which the solution can be expected to possess a certain degree of symmetry, computational efficiency will usually demand that the solution is computed for only one half of the domain. This will introduce a type of boundary across which all the flow conditions are the same. The satisfaction of the symmetry requirement is achieved by the same procedure as that described for a slip wall.

The boundary condition specific to the Navier–Stokes equations is the no-slip wall. The no-slip condition for the velocity at a wall is imposed by a similar approach to that taken by Gnoffo[18]. Using the imaginary cell concept, all the velocity components at these cells are taken to be zero. For an isothermal wall, the temperature at these cells is fixed at the wall temperature. For an adiabatic wall the condition $\partial T/\partial n=0$ is satisfied by taking the temperature at the imaginary cell to be the same as the temperature at the adjacent cell inside the domain. For the pressure, the boundary layer assumption is employed and the pressure at the imaginary cell is taken to be the same as the pressure from inside the computational domain. Other variables can then be determined from the equation of state. This procedure for modelling the no-slip wall boundary condition is equivalent to assuming that the wall is half a cell size outside the domain. Considering the very small size of the cells that are usually employed near the no-slip wall to capture the boundary layer, this should not introduce any serious error into the calculations.

NUMERICAL RESULTS

Apart from the first two examples presented in this section, the other examples are solved using an adaptive refinement approach. The mesh adaptivity for the third and fourth examples is achieved by complete remeshing of the computational domain using the advancing front technique[6]. For brevity, in each case only the solution on the final mesh employed is discussed. In the final example, which illustrates the application of the proposed space marching technique, mesh adaptivity is obtained by a mesh enrichment approach[5].

Inviscid transonic aerofoil

The first example consists of Mach 0.8 inviscid flow past a NACA 0012 aerofoil at an angle of attack of 1.25°. This is an AGARD Fluid Dynamics Panel test case[23] (Test Case 01). A 120 by 29 O-mesh is used for this problem. Detail of the grid near the aerofoil is depicted in *Figure 1*. The solution contours for pressure coefficient, C_p, using the first order upwind scheme is presented in *Figure 2*. The results obtained by the explicit and implicit versions of the scheme are identical.

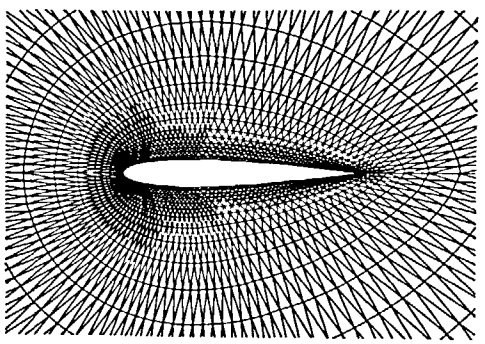


Figure 1 Details of O-mesh for $M=0.8$ inviscid flow past a NACA 0012 aerofoil

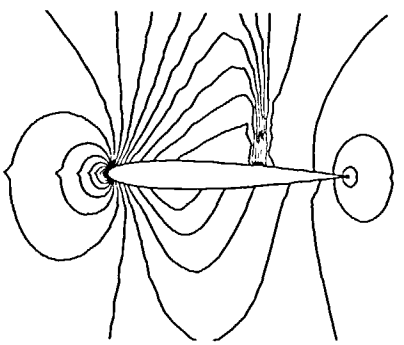


Figure 2 NACA 0012; contours of pressure coefficient obtained by using the first order scheme

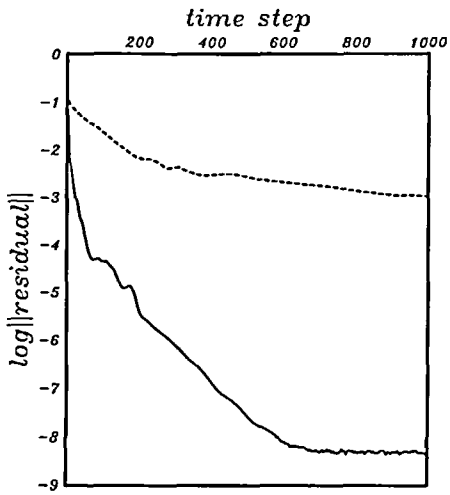


Figure 3 NACA 0012; comparison between explicit and implicit convergence histories. – – – explicit; —— implicit

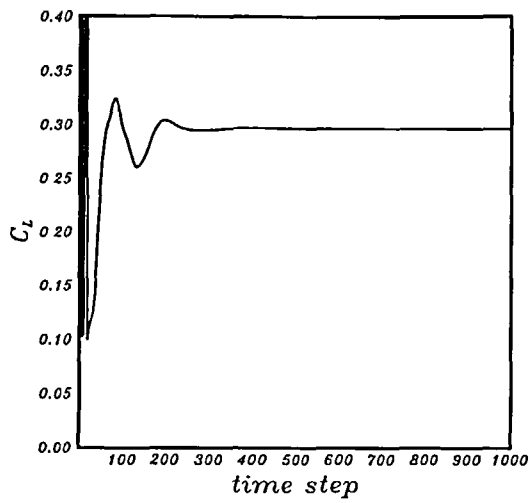


Figure 4 NACA 0012; lift coefficient history for the first order solution

During the implicit time stepping the Courant number was increased to 1000. A comparison between the convergence curves for the explicit and the implicit versions of the scheme is presented in *Figure 3*. This comparison clearly illustrates the advantage of the implicit time integration in obtaining fast convergence towards the steady state. The lift coefficient history for the implicit case is given in *Figure 4*. *Figure 5* presents the pressure coefficient contours obtained by using the proposed higher order algorithm. A comparison between the first order and the higher order results for the variation of C_P, M, and entropy deviation, E_d, along the surface of the aerofoil are depicted in *Figure 6*. As can be seen the higher order scheme resolves the weak shock at the bottom, which is highly smeared by the first order upwind scheme. The false entropy created by the higher order scheme is much smaller than that created by the first order method. According to Reference 23 the solution by Lytton (Contribution No. 3) was one of the most accurate solutions to this problem. The mesh in that study had 158 points on the aerofoil, which is higher than the number used here. The distribution of C_P and M from Lytton's solution are also given in *Figures 6a* and *6b*. On the upper surface of the aerofoil the higher order solution compares quite well with the solution of Reference 23. The position of the shock on the lower surface differs by about 0.05 chord length in the two solutions and the higher order scheme gives a larger undershoot after this shock. Nevertheless, the overall agreement between the two solutions is very good. The lift coefficient history for the higher order solution is presented in *Figure 7*. The lift coefficient obtained with the first order scheme is 0.295 while the higher order solution gives a value of 0.332, which is closer to the value of 0.346 as computed by Lytton[23].

Supersonic viscous flow past a flat plate

This example consists of a developing laminar boundary layer over an insulated flat plate. The problem definition is:

Free stream Mach number	4.0
Reynolds number	4.0×10^6
Free stream temperature	218 K

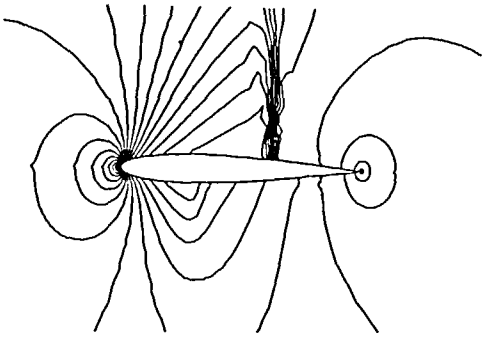
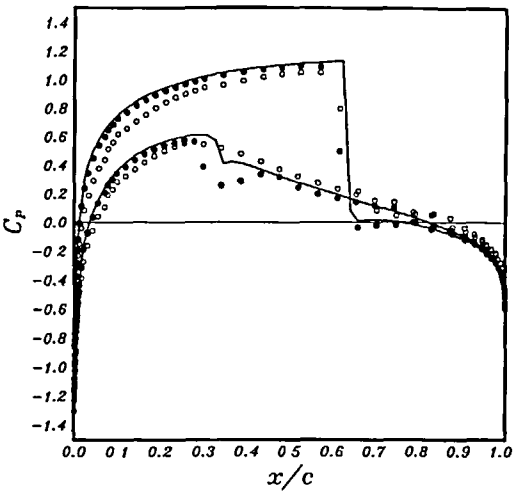
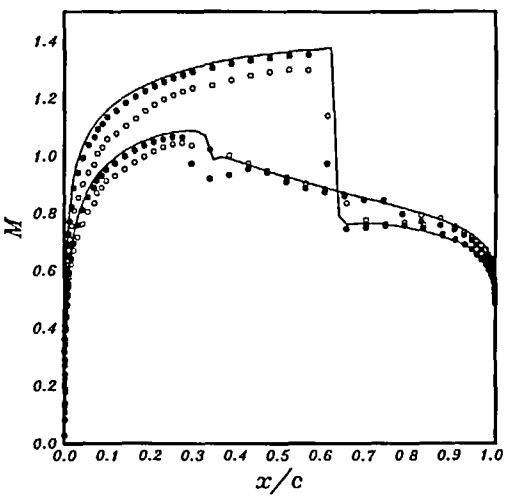


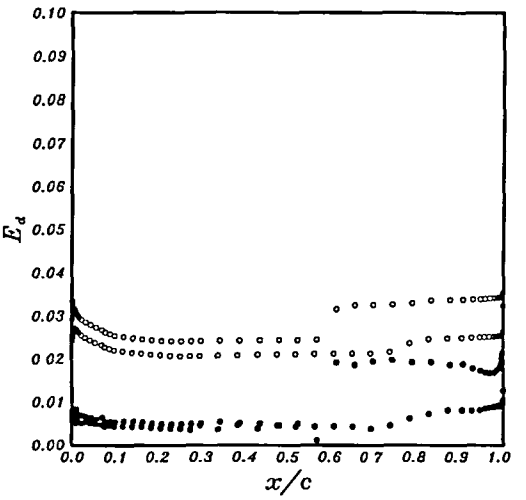
Figure 5 NACA 0012; contours of pressure coefficient obtained by using the higher order scheme



(a) pressure coefficient



(b) Mach number



(c) entropy deviation. ○, First order; ●, higher order; ——, Lytton (1985)

Figure 6 NACA 0012; comparison between the first order and the higher order results for:

A Prandtl number of 0.75 is used for this problem. A structured mesh of triangles consisting of 20,000 cells and 10,201 points as shown in *Figure 8* is employed. The non-dimensional plate length is 1 and the distance of the first layer of grid points from the wall is 0.0001. For this laminar boundary layer a similarity solution exists[24], and can be obtained by introducing the new variable:

$$y' = \frac{y}{x}\sqrt{Re_x}$$

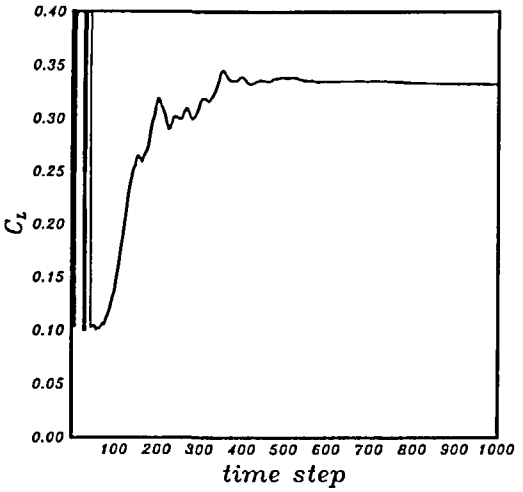


Figure 7 NACA 0012; lift coefficient history for the higher order solution

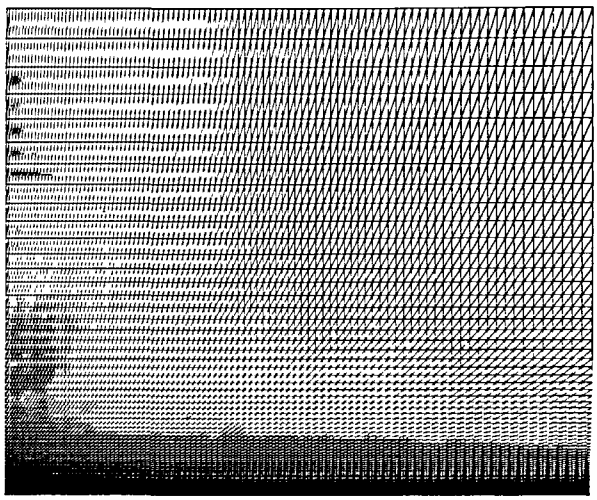


Figure 8 Mesh used for Mach 4 viscous flow past a flat plate

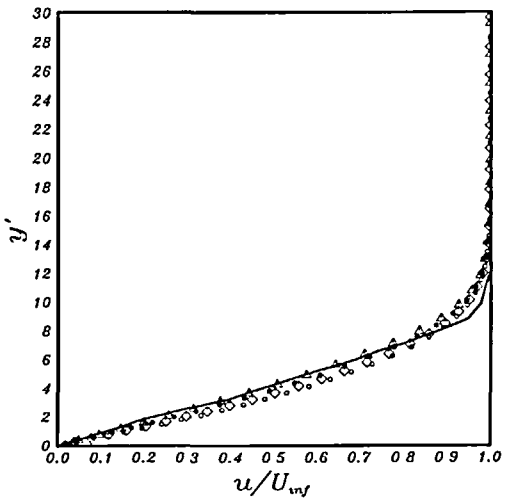


Figure 9 Flat plate; velocity profiles obtained by using the first order scheme. ——, Analytical; △, $x=0.3$; ●, $x=0.6$; ◇, $x=0.7$; ○, $x=0.9$

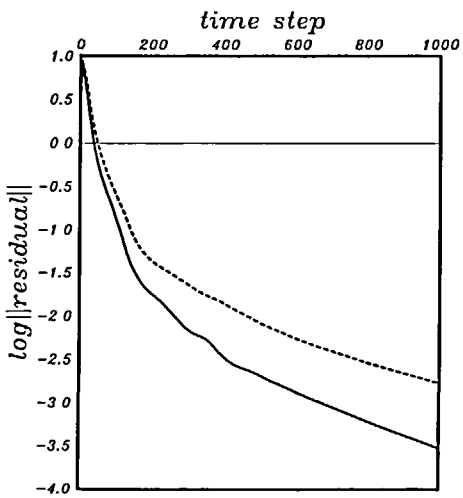


Figure 10 Flat plate; convergence history showing the effect of implicit formulation of the viscous terms. ——, Gauss–Seidel; – – –, Gauss–Jacobi

In *Figure 9* the profiles of the non-dimensionalized velocity, u/U_∞, against y' at four stations, i.e.

$$x=0.3; \quad x=0.5; \quad x=0.7; \quad x=0.9$$

computed by the use of the basic first order method are plotted together with the analytical solution. As can be seen the numerical results exhibit a thicker boundary layer which indicates excessive diffusion. This is due to the inherent numerical viscosity of first order upwinding. The solution profiles at different stations do not match very well, which could be related to a poor determination of density in the boundary layer. Typical convergence curves for this case are

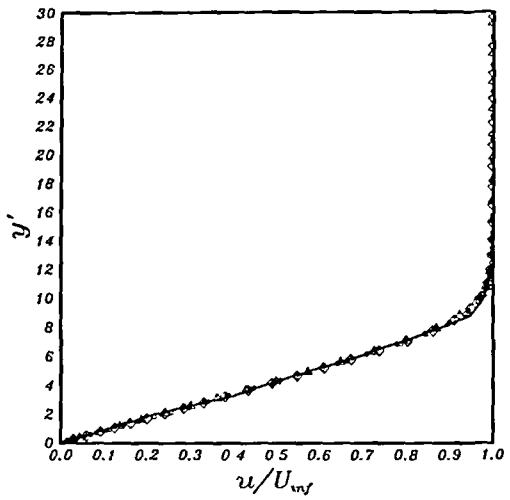


Figure 11 Flat plate; velocity profiles obtained by using the higher order scheme (allowable extrema obtained by using the cells sharing a side with any cell). ——, Analytical; △, $x=0.3$; ●, $x=0.5$; ◇, $x=0.7$; ○, $x=0.9$

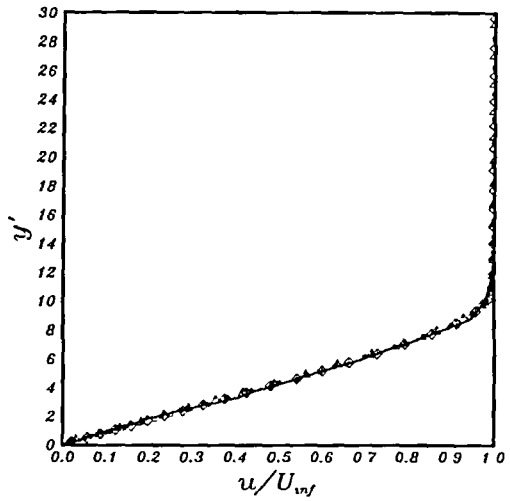


Figure 12 Flat plate; velocity profiles obtained by using the higher order scheme (allowable extrema obtained by using the cells sharing a vertex with any cell). ——, Analytical; △, $x=0.3$; ●, $x=0.5$; ◇, $x=0.7$; ○, $x=0.9$

depicted in *Figure 10*. A Courant number of 10 is used for these calculations. In the higher order solution, separate limiters are used for each equation in order to minimize the effect of artificial diffusion. The evaluation of the maximum and minimum allowable values is based on the three neighbouring cells sharing a side with the current cell. The non-dimensionalized velocity profiles are presented in *Figure 11*. Comparing with *Figure 9* the results are clearly improved. The computed velocity profiles lie on top of each other and are fairly close to the analytical solution. A less diffusive reconstruction can be achieved by using all the surrounding cells to determine the maximum and minimum allowable values. The results obtained with this approach are illustrated in *Figure 12*. The velocity profiles are in excellent agreement with the analytical solution. This emphasises the importance of the discretization of the inviscid terms on the accuracy of Navier–Stokes calculations.

Compression corner in hypersonic viscous flow

This example consists of a hypersonic flow past an 18° compression corner. The problem specification is as follows

Free stream Mach number	14.07
Reynolds number	7.2×10^4
Free stream temperature	90 K
Wall temperature	297 K

The adaptively refined mesh for this problem is shown in *Figure 13*. The mesh consists of 6323 triangular cells and 3265 points. In order to accurately model the boundary layer, 3720 cells are arranged in fifteen structured layers near the no-slip wall. The distance of the first layer of grid points from the solid wall is 0.0004. The solution contours of density, C_p, and M, using the higher order scheme are given in *Figure 14*. Variation of the coefficients of heat transfer, C_h, and skin friction, C_f, and of the normalized pressure, P/P_∞, along the wall are depicted in *Figure 15*. These values are plotted together with a solution obtained by Thomas[25] at the NASA

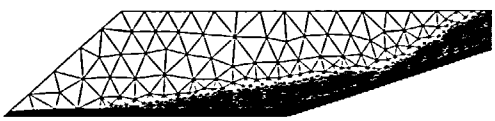
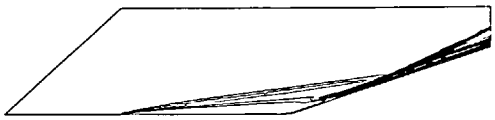
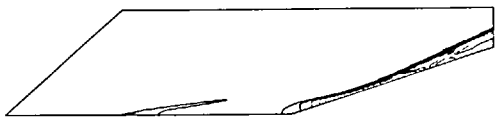


Figure 13 Mesh used for $M=14.07$ viscous flow past an 18° compression corner



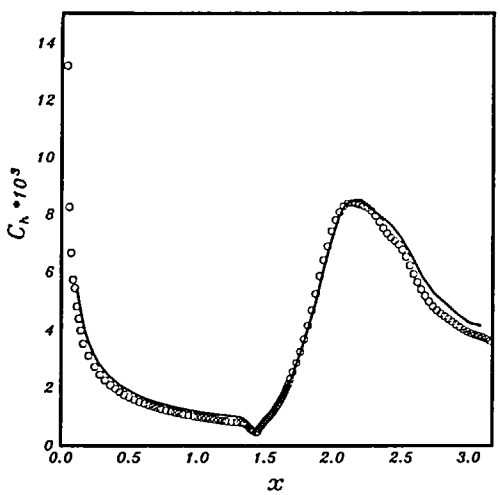
(a) density



(b) pressure coefficient

(c) Mach number, for the higher order solution

Figure 14 Compression corner; contours of:



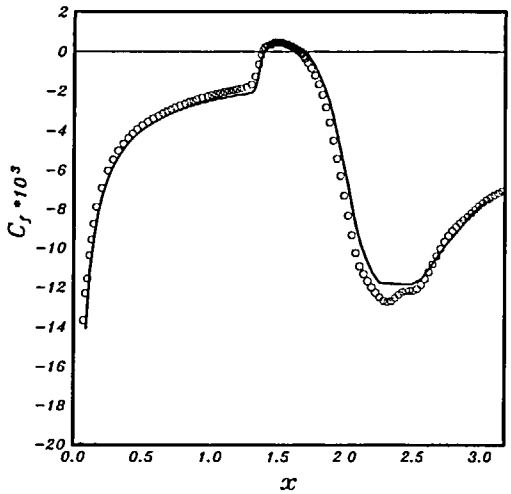
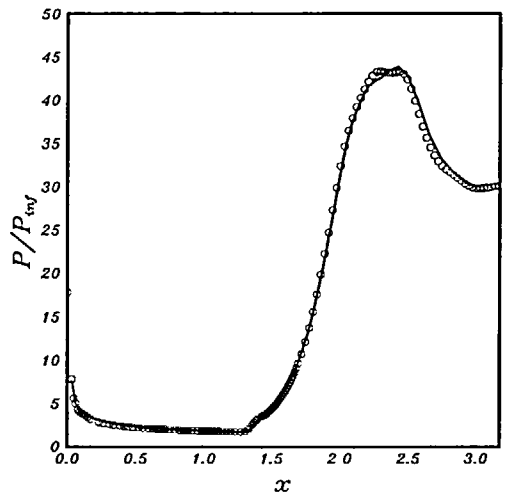


Figure 15 Compression corner; variations of: (a) heat transfer rate coefficient; (b) skin friction coefficient;



(c) non-dimensionalized pressure; along the wall. —— Thomas (1990); O present higher order

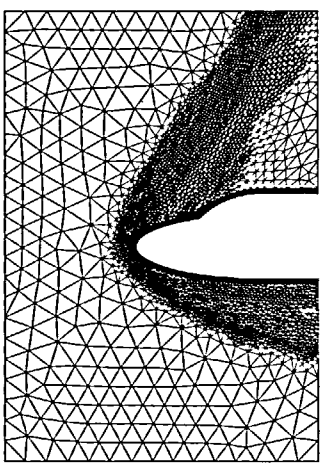


Figure 16 Mesh used for $M=8.15$ viscous flow past a double-ellipse

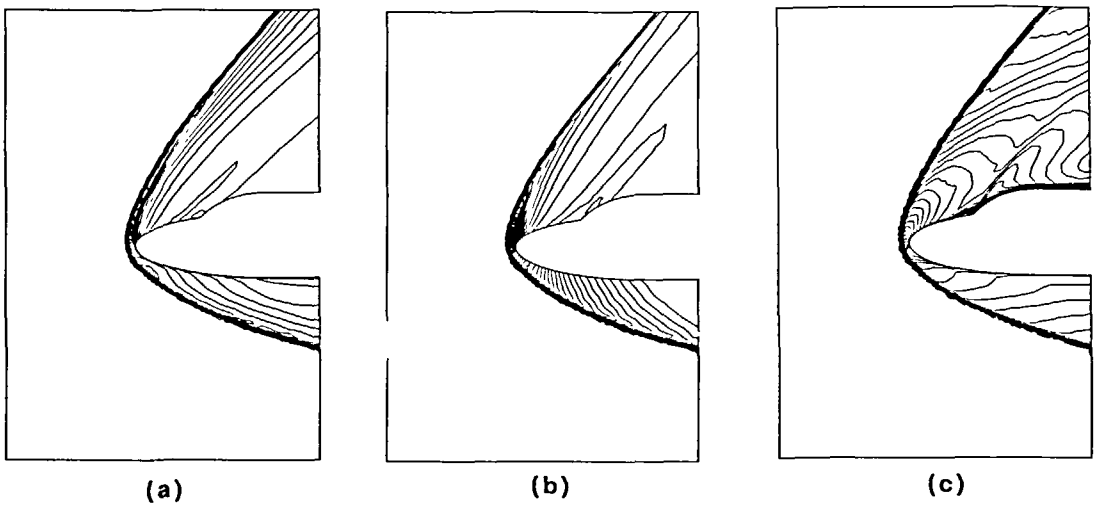


Figure 17 Double-ellipse; contours of: (a) density; (b) pressure coefficient; (c) Mach number; for the higher order solution

Langley Research Center using a third order accurate code (CFL3D) on a structured grid. The present method predicts higher values of skin friction and heat transfer at the reattachment point, but the two solutions generally compare well. The pressure along the wall does not seem to be a very sensitive parameter, and the plots of P/P_∞ are similar.

Double-ellipse in hypersonic viscous flow

The fourth example considered is hypersonic flow past a double-ellipse configuration at 30° angle of attack. The problem definition is:

Free stream Mach number	8.15
Reynolds number	1.67×10^7
Free stream temperature	56 K
Wall temperature	288 K

The final mesh employed for this problem has 8149 cells, of which 1870 are stretched quadrilateral cells arranged in ten layers in the vicinity of the solid surfaces. The number of points in the mesh is 5172. The mesh is depicted in *Figure 16*. The first layer of grid points is located at a distance of 0.0025 from the surface of the configuration. The solution contours of density, C_P, and M, obtained with the higher order scheme, are depicted in *Figure 17*. Variation of the heat transfer, skin friction, and the pressure coefficient for this solution are shown in *Figure 18*. Features of the solution are, the strong bow shock which is captured sharply, high heating at the stagnation point, and the flow separation region at the foot of the canopy.

Supersonic inviscid flow past a blunt body

The final example consists of a supersonic flow past a symmetrical blunt body. The free stream Mach number of the flow is 6.57, and $\gamma = 1.38$. The solution to this problem exhibits a subsonic region behind the bow shock. Due to symmetry, the computational grid covers only half of the domain. The initial mesh used for this problem is shown in *Figure 19a*. The mesh consists of 419 triangular cells and 243 points. The solution is obtained via the space marching form of

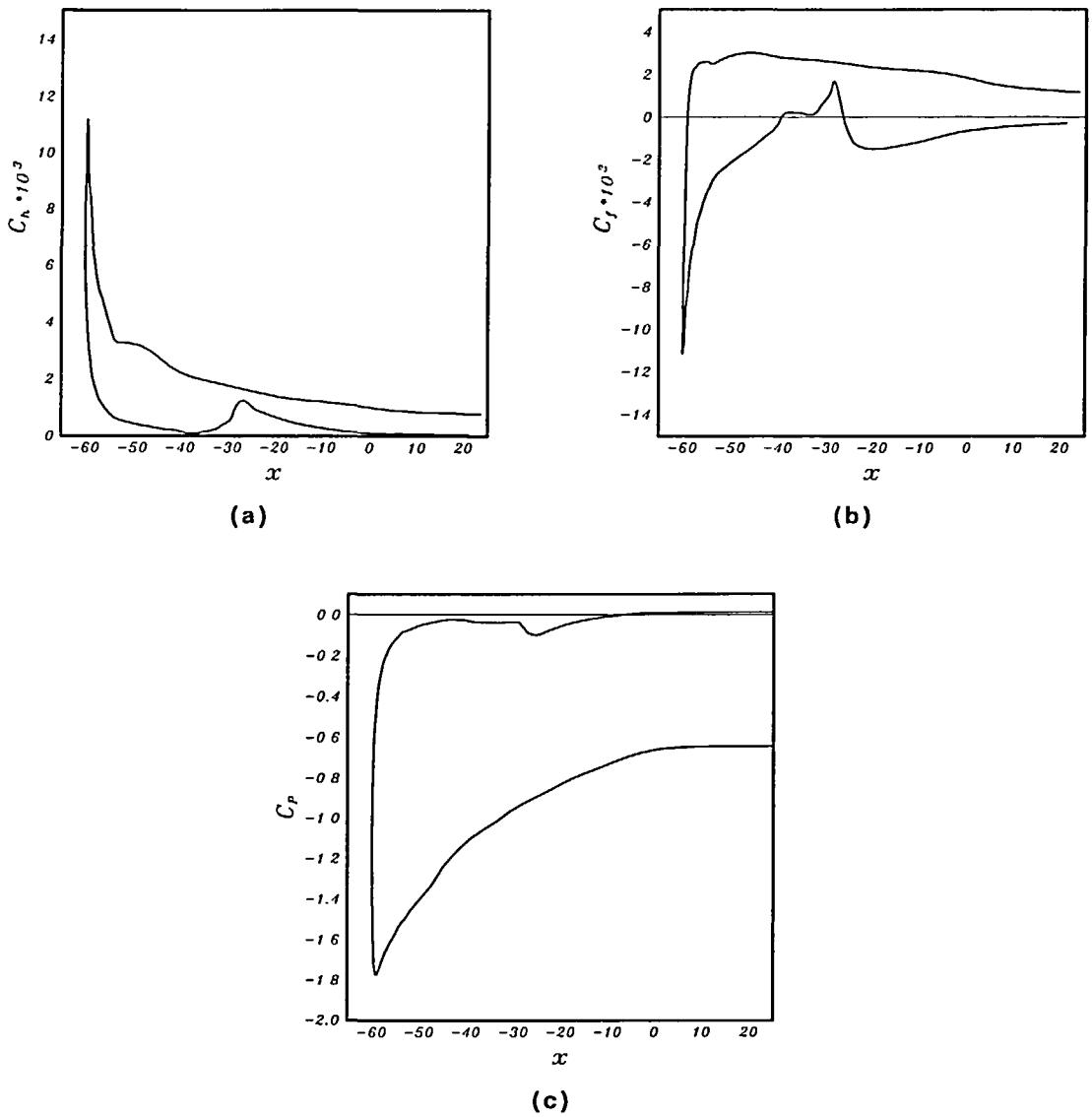


Figure 18 Double-ellipse; variations of: (a) heat transfer rate coefficient; (b) skin friction coefficient; (c) pressure coefficient; along the wall

the scheme. The number of marching planes is 15. The results obtained with a Streamwise Upwind Petrov–Galerkin (SUPG) method on a structured grid are also available for this problem. Calculations for this example are performed by using the higher order scheme. In *Figure 19b* the solution contours of pressure coefficient for the space marching calculations are given. Using the values of density along the axis and over the surface of the blunt body, a comparison is made between the space marching and the SUPG results which is shown in *Figure 20*. The corresponding comparisons for the velocity components at the exit are presented in *Figure 21*. Two levels of mesh enrichment are used to enhance the accuracy of the solution. There are 1121 and 2043 cells in the first and second refined meshes respectively. These are

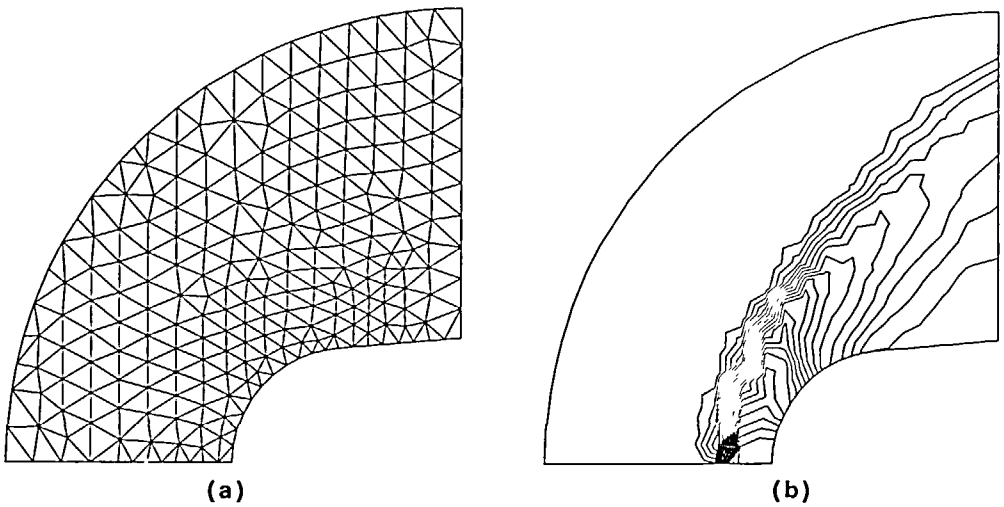


Figure 19 Mach 6.57 inviscid flow past a blunt body; (a) first mesh; (b) pressure coefficient contours obtained by using the space marching scheme

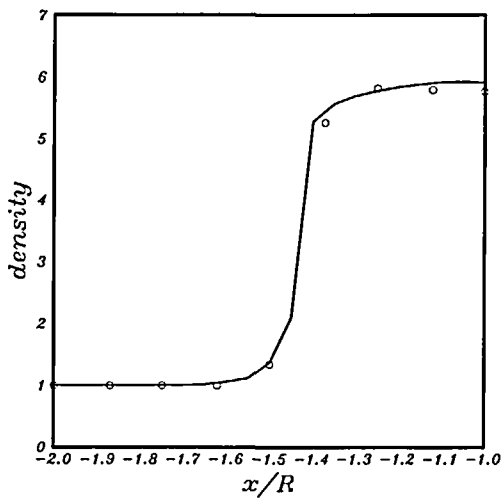


Figure 20 Blunt body; variation of density along the axis and on the surface for the first mesh. —— SUPG; O, space marching

presented in *Figures 22a* and *25a*. In each case, the converged solution is used as the initial condition for the next refined grid. Solution contours of C_P for the first refined mesh are given in *Figure 22b*. *Figure 23* illustrates the variation of density along the axis and over the surface of the blunt body, for the space marching and the SUPG computations on the second mesh. The values of velocity components along the outflow boundary are shown in *Figure 24*. The corresponding results for the second refined mesh are presented in *Figures 25b*, *26*, and *27*. As can be seen from these Figures the solution obtained with the space marching scheme compares

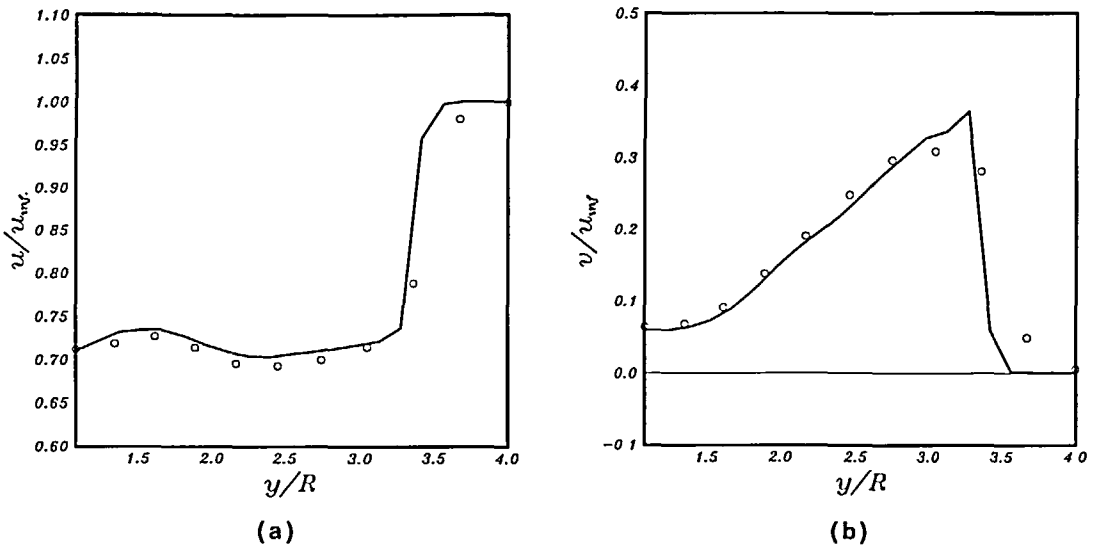


Figure 21 Blunt body; variations of: (a) u-velocity; (b) v-velocity; along the outflow for the first mesh. ——, SUPG; O, space marching

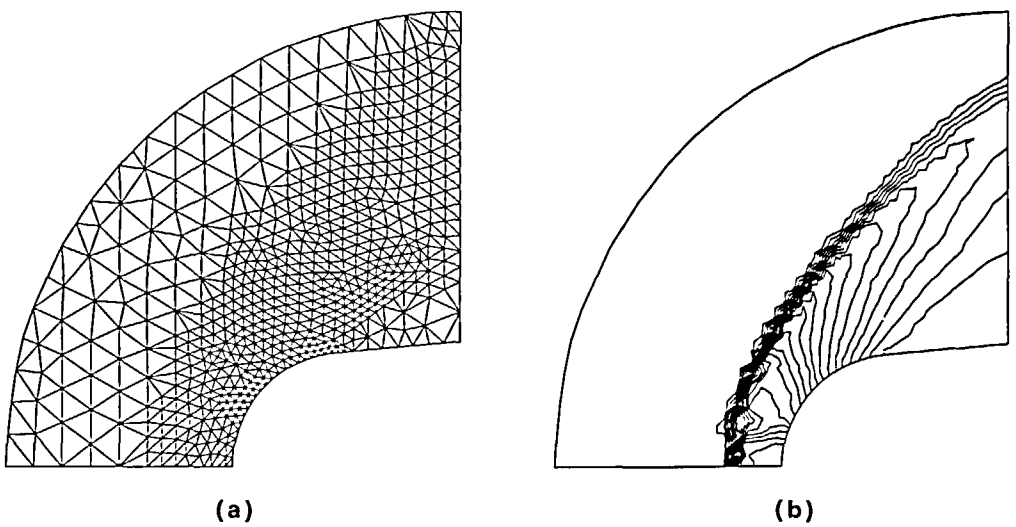


Figure 22 Mach 6.57 inviscid flow past a blunt body; (a) second mesh; (b) pressure coefficient contours obtained by using the space marching scheme

well with the SUPG results. The resolution of the shock wave, even on a coarser mesh, is sharper in the space marching results. The converged steady state solution, in each case, has been obtained by one pass over the solution domain. No monotonicity problem was observed for the higher order solution.

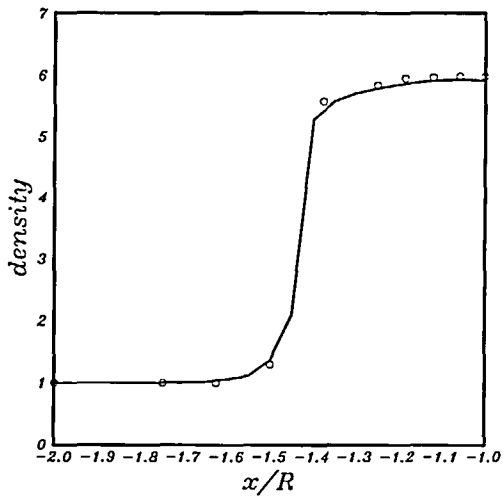


Figure 23 Blunt body; variation of density along the axis and on the surface for the second mesh. ——, SUPG; O, space marching

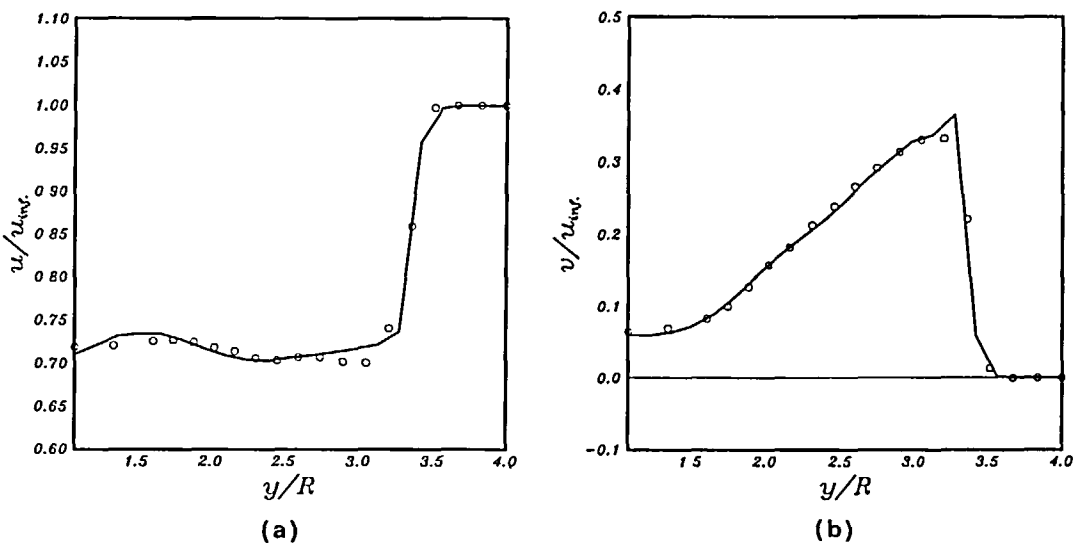


Figure 24 Blunt body; variations of: (a) u-velocity; (b) v-velocity; along the outflow for the second mesh. ——, SUPG; O, space marching

CONCLUSIONS

An upwind finite volume scheme for solving the compressible Euler and laminar Navier–Stokes equations on unstructured grids has been developed. Higher order spatial accuracy is achieved via a linear reconstruction of the primitive variables, which leads to enhanced accuracy over the first order results without producing spurious oscillations. It has been shown that the

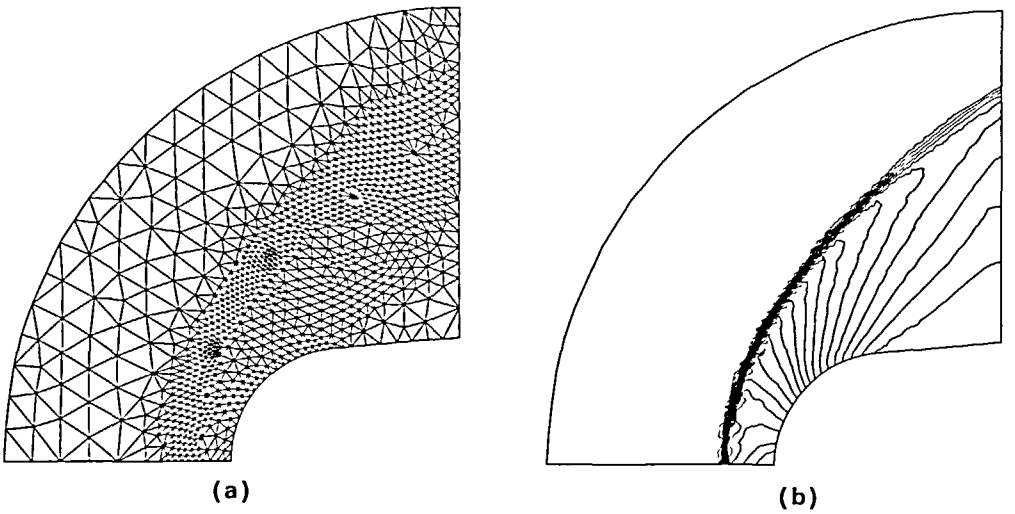


Figure 25 Mach 6.57 inviscid flow past a blunt body; (a) third mesh; (b) pressure coefficient contours obtained by using the space marching scheme

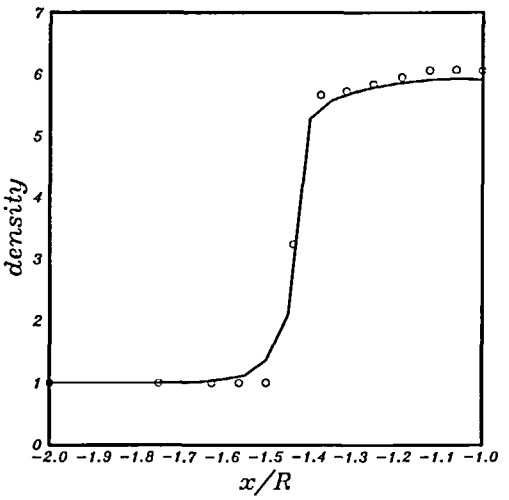


Figure 26 Blunt body; variation of density along the axis and on the surface for the third mesh. ——, SUPG; O, space marching

discretization of the inviscid terms has a significant effect on the accuracy of the computed boundary layers. Convergence towards steady state is achieved by use of a point implicit solver using local time stepping. Numerical examples indicate that the scheme performs well for a wide range of problems. The use of space marching in conjunction with unstructured grids for supersonic inviscid problems has been examined. An algorithm for detecting the imbedded subsonic regions within the supersonic flow field is introduced, and the results of using this approach have been shown to compare well with the results of a time marching scheme.

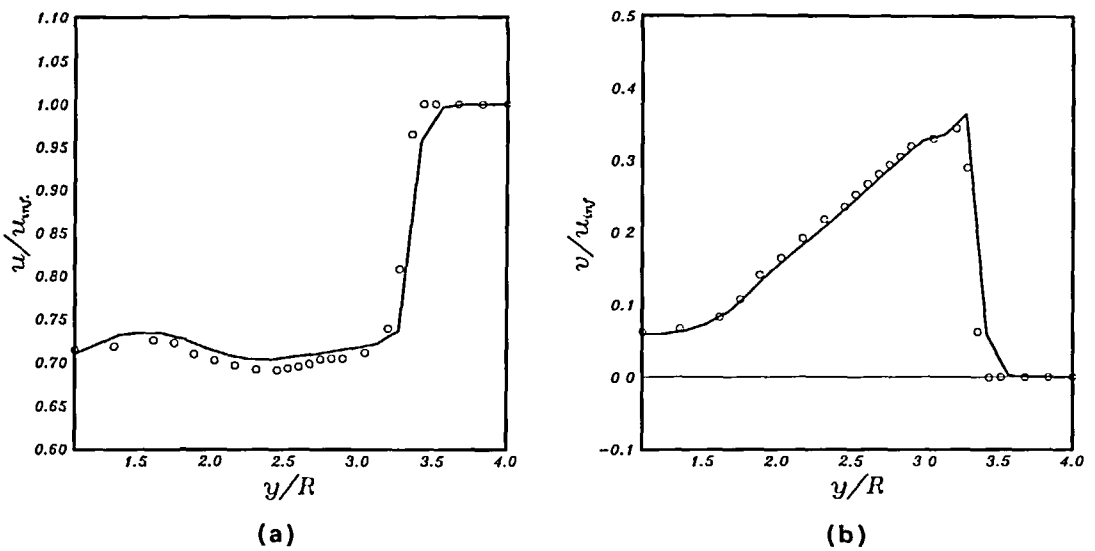


Figure 27 Blunt body; variations of: (a) u-velocity; (b) v-velocity; along the outflow for the third mesh. ——, SUPG; O, space marching

Further research work can be carried out in two areas. First, one can attempt to improve the numerical algorithm in terms of accuracy. The spatial accuracy of the scheme can be improved by performing a quadratic reconstruction of the variables. Moreover, it has been argued that some form of multidimensional Riemann solver can enhance the accuracy of the numerical scheme, especially for highly distorted grids. Secondly, one may wish to extend the adaptivity of the proposed scheme to encompass a wider range of problems. Research towards the extension of the proposed scheme to 3-D and the inclusion of turbulence modelling is being carried out and the initial results are encouraging.

ACKNOWLEDGEMENTS

The authors express their thanks to the Aerothermal Loads Branch, NASA Langley Research Center, for their partial support of this work under NASA Research Grant Number NAGW-1809.

REFERENCES

1 Roe, P. L. Characteristic-based schemes for the Euler equations, *A. Rev. Fluid Mech.*, **18**, 337–365 (1986)
2 Roe, P. L. Approximate Riemann solvers, parameter vectors and difference schemes, *J. Comp. Phys.*, **43**, 357–372 (1981)
3 Osher, S. Shock modelling in aeronautics, *Numerical Methods for Fluid Dynamics*, (Eds. Morton and Baines), Academic Press, New York, pp. 179–217 (1982)
4 Jameson, A., Baker, T. J. and Weatherhill, N. P. Calculation of inviscid transonic flow over a complete aircraft, *AIAA Paper No. 86-0202* (1986)
5 Morgan, K. and Peraire, J. Finite element methods for compressible flows, *Von Karman Instit. Fluid Dynamics (Lect. Series 1987-04)* (1987)
6 Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O. C. Adaptive remeshing for compressible flow computations, *J. Comp. Phys.*, **72**, 449–466 (1987)
7 Mavriplis, D. J. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes, *AIAA Paper No. 88-3707* (1988)

8 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 No. 87-0560* (1987)
9 Thareja, R. R., Stewart, J. R., Hassan, O., Morgan, K. and Peraire, J. A point implicit unstructured grid solver for the Euler and Navier–Stokes equations, *AIAA Paper No. 88-0036* (1988)
10 Venkatakrishnan, V. and Barth, T. J. Application of direct solvers to unstructured meshes for the Euler and Navier–Stokes equations using upwind schemes, *AIAA Paper No. 89-0364* (1989)
11 Barth, T. J. and Jesperson, D. C. The design and application of upwind schemes on unstructured meshes, *AIAA Paper No. 89-0366* (1989)
12 Whitaker, D. L., Grossman, B. and Löhner, R. Two-dimensional Euler computations on a triangular mesh using an upwind finite-volume scheme, *AIAA Paper No. 89-0470* (1989)
13 Batina, J. T. Three-dimensional flux-split Euler scheme involving unstructured dynamic meshes, *AIAA Paper No. 90-1649* (1990)
14 Soltani, S., Morgan, K. and Peraire, J. An unstructured grid cell-centred scheme for compressible flow, *Proc. NUMETA 90 Conf. Swansea*, pp. 902–913 (1990)
15 Van Leer, B., Thomas, J. L., Roe, P. L. and Newcome, R. W. A comparison of numerical flux vector formulas for the Euler and Navier–Stokes equations, *AIAA Paper No. 87-1104* (1987)
16 Soltani, S. An upwind scheme for the equations of compressible flow on unstructured grids, *PhD Thesis*, Department of Aeronautics, Imperial College, University of London (1991)
17 Harten, A. High resolution schemes for hyperbolic conservation laws, *J. Comp. Phys.*, **49**, 357–393 (1983)
18 Gnoffo, P. A. Application of program LAURA to three dimensional AOTV flow fields, *AIAA Paper No. 86-0565* (1986)
19 Schiff, L. B. and Steger, J. L. Numerical simulation of steady supersonic viscous flow, *AIAA Paper No. 79-0130* (1979)
20 Chakravarthy, S. R. and Szema, K. Y. An Euler solver for three-dimensional flows with subsonic pockets, *AIAA Paper No. 85-1703* (1985)
21 Lawrence, S. L., Tannehill, J. C. and Chaussee, D. S. An upwind algorithm for the parabolized Navier–Stokes equations, *AIAA Paper No. 86-1117* (1986)
22 McGrory, W. D., Walters, R. W. and Löhner, R. A three-dimensional space marching algorithm for the solution of the Euler equations on unstructured grids, *AIAA Paper No. 90-0014* (1990)
23 Test Cases for Inviscid Flow Field Methods, *AGARD Advis. Rep. No. 211* (1985)
24 Van Driest, E. R. Investigation of laminar boundary layer in compressible fluids using the Crocco method, *NACA Techn. Note 2597* (1952)
25 Thomas, J. L. personal communication (1990)