Scientia Iranica, Vol. 12, No. 2, pp 207–216

SCIENTIA
IRANICA

Research Note

Inviscid Compressible Flow Computations on 3D Unstructured Grids

M.T. Manzari[1]

In this paper, an explicit finite element based numerical procedure is presented for simulating three-dimensional inviscid compressible flow problems. The implementation of the first-order upwind method and a higher-order artificial dissipation technique on unstructured grids, using tetrahedral elements, is described. Both schemes use a multi-stage Runge-Kutta time-stepping method for time integration. The use of an edge-based data structure in the finite element formulation and its computational merits are also elaborated. Furthermore, the performance of the two schemes in solving a benchmark problem involving transonic flow about an ONERA M6 wing is compared and detailed solutions are presented.

INTRODUCTION

The simulation of inviscid compressible flow over realistic aerodynamic configurations is a fundamental problem in aircraft industries. Despite many advancements in this field, there is still need for more research to achieve more efficient methodologies in terms of generality, accuracy and speed.

Due to the fact that many practical flow problems are geometrically complex and possess a wide range of length scales, the development of a robust and generalized solution methodology requires a strong ability to deal with such features. The first efforts in CFD relied on structured Cartesian methods. The non-adapted Cartesian grids with stair-cased geometry, however, could not represent the boundaries of the flow domain properly. In the past three decades, the generalization of the structured grid strategy to curvilinear coordinates and the use of multi-block grids made two major impacts on this field, but, the challenge of automatic mesh generation remained an unsolved problem [1]. In the past decades, an alternative Unstructured Grid (UG) method was developed, in which element (cell) connectivities were stored explicitly [2,3]. The UG approach is often employed to accomplish mesh generation and adaptation almost automatically and to resolve the governing partial differential equation without requiring an excessive number of mesh points. Naturally, this new strategy required modified equation solvers in order to overcome the complexity introduced in the data structure, due to having more flexibility in the geometry modeling. In recent years, several CFD practitioners developed efficient and accurate flow solvers, based on the UGs [2,4]. Many UG flow solvers are based on the finite volume method. It is known, however, that the Finite Element Method (FEM) can provide other useful features, such as the use of higher-order elements and a better coupling of the governing equations.

A noticeable improvement in the computational efficiency of the FEM was made by Peraire et al. [4], following the ideas introduced by Barth [5]. They used an edge-based data structure instead of the standard element-based data structure used in the traditional FEM approach. It has been shown that the use of this data structure in 3D simulations results in significantly lower CPU time and smaller memory allocations. The edge-based FEM approach works with edges (or sides) instead of elements. Here, an edge is referred to a line connecting two nodes of an element. The new approach keeps the original FEM formulation intact but re-arranges the discretized equations such that instead of an assembly of element matrices, one is required to assemble the edge contributions. In this way, the computational loops over elements in a standard FEM program are replaced by the loops over edges present in the mesh. This change does not alter the computational properties of the FEM method. It does, however, introduce great flexibility in the method to incorporate many available numerically efficient methods [6-8]. It should

1. *Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran.*

be mentioned that using an edge-based formulation becomes troublesome when higher-order elements are to be used. Another important feature of the edge-based strategy is its suitability for parallel computing [9]. In fact, the edge-based formulations were originally developed to provide better properties for parallel processing.

This paper presents an unstructured grid FEM algorithm for solving 3D inviscid compressible flows. The algorithm is based on an edge-based form of the so-called standard Galerkin formulation. In order to achieve practical formulations suitable for solving realistic flows involving discontinuities such as shocks, the method is stabilized in two different ways. First, the first-order accurate upwind method of Roe [10] is used in connection with a three-stage Runge-Kutta time marching method. Second, an artificial dissipation method, originally developed by Jameson, Schmidt and Turkel [11], is employed to provide a higher-order method. This method also uses a multi-stage Runge-Kutta time-stepping algorithm.

Below, after introducing the governing equations for a 3D compressible inviscid flow, the general FEM formulation is described and the features of the edge-based data structure are explained. Then, the details of implementation of both the Roe's first-order upwind and the Jameson-Schmidt-Turkel (JST) methods in an edge-based FEM context are given. A criterion is given for the choice of time-step size to achieve stable solutions. Finally, the performance of these methods is compared by solving a transonic flow over an ONERA M6 wing and the results are compared with experimental data. The effect of mesh resolution is also studied and the convergence history for the JST method is provided.

GOVERNING EQUATIONS

The system of governing equations describing inviscid compressible flow comprises the equations of mass, momentum and energy. The complete system of governing equations is written in the non-dimensional conservative form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^j}{\partial x_j} = 0, \tag{1}$$

where the unknown vector, $\mathbf{U}$, and the inviscid flux, $\mathbf{F}^j$, are given by:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}, \qquad \mathbf{F}^j = \begin{bmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ (\rho E + p) u_j \end{bmatrix}, \tag{2}$$

for $j = 1, 2, 3$. Here, the dimensionless form employed is based upon the density and velocity of the free-stream and a characteristic length of the problem. In the above equations, t denotes time, x_j the coordinate relative to a Cartesian coordinate system, $ox_1x_2x_3$, u_j the velocity in direction x_j, ρ density, p the pressure and δ_{ij} the Kronecker delta. The total energy per unit mass is defined as $E = e + u_i u_i/2$, where e is the specific internal energy. The fluid is assumed to be an ideal gas with constant specific heat ratio, γ, obeying the equation of state:

$$p = \rho(\gamma - 1)T/\gamma, \tag{3}$$

where T is the temperature.

Let one consider a spatial domain, Ω, which is bounded by a closed surface, Γ, with unit outward normal vector, $\mathbf{u} = (n_1, n_2, n_3)$. To complete the description of the problem governed by Equation 1, the prescription of an initial condition and appropriate boundary conditions are required. For the initial condition, it will be assumed that free-stream values are imposed everywhere in Ω at some time, $t = t^\circ$. At a wall boundary, the slip condition is imposed. This means that the normal component of the velocity vector is cancelled for the nodes on the wall. Far-field boundary conditions are applied using a linearized characteristic analysis in the direction normal to the boundary to correct the computed nodal values obtained at the far-field.

SOLUTION ALGORITHM

A weak variational formulation of the problem is adopted as the starting point for the development of an approximate solution procedure. This weak form can be written as:

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t} W \, d\Omega = \int_\Omega \mathbf{F}^j \frac{\partial W}{\partial x_j} d\Omega - \int_\Gamma \overline{\mathbf{F}}^j n_j W \, d\Gamma, \tag{4}$$

for all suitable weighting functions, W, and for all $t > t^0$. In this expression, an overbar represents a prescribed boundary flux. Then, the region Ω is discretized into an unstructured assembly of tetrahedral elements, with the nodes numbered from 1 to p and the standard linear finite element shape function, N_J, associated with node J, is employed ($\mathbf{U}^{(p)} = \sum_J \mathbf{U}_J(t) N_J(x)$). Due to the compact support of the shape function, N_J, the Galerkin finite element approximation of the problem can be written as:

$$\sum_{E \in I_{\Omega_E}} \int \frac{\partial \mathbf{U}^{(p)}}{\partial t} N_I d\Omega = \sum_{E \in I_{\Omega_E}} \int \mathbf{F}^j(\mathbf{U}^{(p)}) \frac{\partial N_I}{\partial x_j} d\Omega - \sum_{f \in I_{\Gamma_B}} \int \overline{\mathbf{F}}^j(\mathbf{U}^{(p)}) n_j N_I d\Gamma. \tag{5}$$

The left hand side of this equation can be evaluated exactly to give:

$$\sum_{E\in I}\int_{\Omega_E}\frac{\partial \mathbf{U}^{(p)}}{\partial t}N_J d\Omega = \sum_{E\in I}\left[\int_{\Omega_E} N_I N_J d\Omega\right]\frac{d\mathbf{U}_J}{dt}$$

$$= [\mathbf{M}]_{IJ}\frac{d\mathbf{U}_J}{dt}, \tag{6}$$

where Ω_E is the volume of element E which has nodes I, J, K and L as shown in Figure 1, and Γ_B is the area of the boundary face associated to nodes I, J and K. Also, $\mathbf{M}$ denotes the finite element consistent mass matrix, which, for the steady flow analysis of interest here, is replaced by the lumped (diagonal) mass matrix, $\mathbf{M}_L$. For the integrals of the inviscid flux, the following form is used:

$$\mathbf{F}^j = \sum_{I=1}^{p}\mathbf{F}_I^j N_I(x_i), \tag{7}$$

where $\mathbf{F}_I^j = \mathbf{F}^j(\mathbf{U}_I)$. Therefore:

$$\int_{\Omega_E}\mathbf{F}^j(U^{(p)})\frac{\partial N_I}{\partial x_j}d\Omega \approx \frac{\Omega_E}{4}\left[\frac{\partial N_I}{\partial x_j}\right]_E\left(\mathbf{F}_I^j+\mathbf{F}_J^j+\mathbf{F}_K^j+\mathbf{F}_L^j\right),$$

$$\int_{\Gamma_B}\overline{\mathbf{F}}^j(U^{(p)})n_j N_I d\Gamma \approx \frac{\Gamma_B}{12}(2\overline{\mathbf{F}}_I^n+\overline{\mathbf{F}}_J^n+\overline{\mathbf{F}}_K^n). \tag{8}$$

As mentioned before, the integrals appearing in the Galerkin statement can be evaluated using an edge-based data structure. Figure 1 depicts a typical four-node tetrahedral element, along with its nodes, and a typical edge. The resulting semi-discretized equation, at node I of the mesh, is [4]:

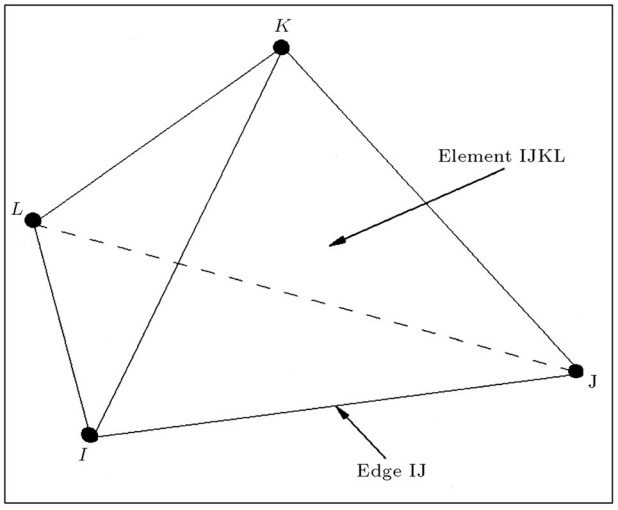

Figure 1. A typical 4-node tetrahedral element and edge.

$$\left[\mathbf{M}_L\frac{d\mathbf{U}}{dt}\right]_I = -\sum_{S=1}^{m_I}\frac{C_{II_S}^j}{2}\left(\mathbf{F}_I^j+\mathbf{F}_{I_S}^j\right)$$

$$+\left\langle\sum_{f=1}^{l_I}D_f\left(4\overline{\mathbf{F}}_I^n+2\overline{\mathbf{F}}_{I_{f_1}}^n+2\overline{\mathbf{F}}_{I_{f_2}}^n+2\mathbf{F}_I^n-\mathbf{F}_{I_{f_1}}^n-\mathbf{F}_{I_{f_2}}^n\right)\right\rangle_I, \tag{9}$$

where the summations extend over the m_I edges, and the l_I boundary faces are connected to node I. The term $\langle\cdots\rangle$ is only non-zero if node I is on the boundary. $C_{II_S}^j$ and D_f are the weight functions associated with edge II_S and face f, respectively. The weights $C_{II_S}^j$ and D_f are computed as:

$$C_{II_S}^j = -\sum_{E\in II_S}\frac{\Omega_E}{2}\left[\frac{\partial N_I}{\partial x_j}\right]_E + \left\langle\sum_{f\in II_S}\frac{\Gamma}{12}n_j\right\rangle,$$

$$D_f = -\frac{\Gamma_f}{24}. \tag{10}$$

Here, n_j is the component in the x_j direction of the unit normal to the boundary face, f.

The inviscid flux contribution can be rewritten as:

$$\sum_{S=1}^{m_I}\frac{C_{II_S}^j}{2}\left(\mathbf{F}_I^j+\mathbf{F}_{I_S}^j\right) = \sum_{S=1}^{m_I}\frac{\ell_{II_S}}{2}\left(\mathbf{F}_I^j S_{II_S}^j + \mathbf{F}_{I_S}^j S_{II_S}^j\right)$$

$$= \sum_{S=1}^{m_I}\ell_{II_S}\mathbf{F}_{II_S}, \tag{11}$$

where:

$$\mathbf{C}_{II_S} = \left(C_{II_S}^1, C_{II_S}^2, C_{II_S}^3\right),$$

$$\ell_{II_S} = |\mathbf{C}_{II_S}|,$$

$$S_{II_S}^j = \frac{C_{II_S}^j}{\ell_{II_S}}. \tag{12}$$

The formulation given in Equation 9 represents a central difference type of approximation and is prone to produce spurious numerical oscillations and/or numerical instabilities when solving convection dominant flow problems. In order to produce a practical scheme, a consistent numerical flux is substituted for the actual inviscid flux. Two alternatives are considered in this work and are described below.

ROE'S FIRST-ORDER UPWIND METHOD

This is a popular scheme due to Roe [10] and is widely used in both incompressible and compressible flow solvers. Although only a first-order form of the method is used here, various higher-order formulations

are also available based on this scheme. The main reason for using Roe's method in this work is that it is a highly robust and, computationally, fairly inexpensive technique. Therefore, its performance is used for comparison purposes to assess the performance of the higher-order accurate (JST) method.

Roe's first-order upwind flux-difference splitting method can be implemented by replacing the actual flux, $\mathbf{F}_{II_S}$, by a consistent numerical flux, F_{II_S}, as:

$$\mathbf{F}_{II_S} = \frac{1}{2}\left\{\mathbf{F}_I^j S_{II_S}^j + \mathbf{F}_J^j S_{II_S}^j - |\mathbf{A}_{II_S}|\,(\mathbf{U}_{I_S} - \mathbf{U}_I)\right\}, \tag{13}$$

where the Jacobian matrix, $\mathbf{A}_{II_S}$, is defined by:

$$\mathbf{A}_{II_S} = \mathbf{A}\,(\mathbf{U}_I, \mathbf{U}_{I_S}) = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)_{II_S}, \tag{14}$$

and is evaluated in the direction of weight coefficient vector, $\mathbf{C}_{II_S}$, using Roe's averaging procedure [10]. For the Euler system in 3D, the Jacobian matrix along any arbitrary direction (n_1, n_2, n_3) can be diagonalized as $|\mathbf{A}| = \mathbf{R}^{-1}|\mathbf{\Lambda}|\mathbf{R}$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$, in which $\lambda_1 = \lambda_2 = \lambda_3 = V, \lambda_4 = V + c$ and $\lambda_5 = V - c$ are the eigen-values of the system. Here, $V = n_1 u_1 + n_2 u_2 + n_3 u_3$, u_j being the velocity component in direction x_j, and c is the local speed of sound. The dissipation term can be efficiently computed as [12]:

$$\begin{aligned}
&|\mathbf{A}_{II_S}|\,(\mathbf{U}_{I_S} - \mathbf{U}_I) \\
&= |\widetilde{V}| \left\{ \left(\Delta\rho - \frac{\Delta p}{\widetilde{c}^2}\right) \begin{bmatrix} 1 \\ \widetilde{u}_1 \\ \widetilde{u}_2 \\ \widetilde{u}_3 \\ 0.5(\widetilde{u}_1^2 + \widetilde{u}_2^2 + \widetilde{u}_3^2) \end{bmatrix} \right. \\
&\left. + \widetilde{\rho} \begin{bmatrix} 0 \\ \Delta u_1 - n_1 \Delta V \\ \Delta u_2 - n_2 \Delta V \\ \Delta u_3 - n_3 \Delta V \\ \widetilde{u}_1 \Delta u_1 + \widetilde{u}_2 \Delta u_2 + \widetilde{u}_3 \Delta u_3 + \widetilde{V} \Delta V \end{bmatrix} \right\} \\
&+ |\widetilde{V} + \widetilde{c}| \left(\frac{\Delta p + \widetilde{\rho}\widetilde{c}\Delta V}{2\widetilde{c}^2}\right) \begin{bmatrix} 1 \\ \widetilde{u}_1 + n_1\widetilde{c} \\ \widetilde{u}_2 + n_2\widetilde{c} \\ \widetilde{u}_3 + n_3\widetilde{c} \\ \widetilde{h} + \widetilde{U}\widetilde{c} \end{bmatrix} \\
&+ |\widetilde{V} - \widetilde{c}| \left(\frac{\Delta p + \widetilde{\rho}\widetilde{c}\Delta V}{2\widetilde{c}^2}\right) \begin{bmatrix} 1 \\ \widetilde{u}_1 - n_1\widetilde{c} \\ \widetilde{u}_2 - n_2\widetilde{c} \\ \widetilde{u}_3 - n_3\widetilde{c} \\ \widetilde{h} - \widetilde{V}\widetilde{c} \end{bmatrix},
\end{aligned} \tag{15}$$

where $\Delta V = n_1\Delta u_1 + n_2\Delta u_2 + n_3\Delta u_3$ and the total enthalpy is $h = E + p/\rho$. The Roe average quantities (denoted by $\sim$) are defined as:

$$\begin{aligned}
\widetilde{\rho} &= \sqrt{\rho_L \cdot \rho_R}, \\
\widetilde{u}_1 &= \left(u_{1L} + u_{1R}\cdot\sqrt{\rho_R/\rho_L}\right) / \left(1 + \sqrt{\rho_R/\rho_L}\right), \\
\widetilde{u}_2 &= \left(u_{2L} + u_{2R}\cdot\sqrt{\rho_R/\rho_L}\right) / \left(1 + \sqrt{\rho_R/\rho_L}\right), \\
\widetilde{u}_3 &= \left(u_{3L} + u_{3R}\cdot\sqrt{\rho_R/\rho_L}\right) / \left(1 + \sqrt{\rho_R/\rho_L}\right), \\
\widetilde{h} &= \left(h_L + h_R\cdot\sqrt{\rho_R/\rho_L}\right) / \left(1 + \sqrt{\rho_R/\rho_L}\right), \\
\widetilde{c}^2 &= (\gamma - 1)\left(\widetilde{h} - (u_1^2 + u_2^2 + u_3^2)/2\right).
\end{aligned} \tag{16}$$

Here, R and L represent the state variables to the right and left of the interface between the two nodes I and I_S. To construct a time marching method, a three-stage time stepping Runge-Kutta scheme is employed. This leads to:

$$\begin{cases} \mathbf{U}_I^{(0)} = \mathbf{U}_I^n \\ \mathbf{U}_I^{(k)} = \mathbf{U}_I^n - \alpha_k \Delta t [\mathbf{M}_L]^{-1} \mathbf{R}_I^{(k-1)} \quad \text{for } k = 1, 2, 3, \\ \mathbf{U}_I^{n+1} = \mathbf{U}_I^{(3)} \end{cases} \tag{17}$$

where $\mathbf{U}_I^n$ denotes the solution at node I at time $t = t^n, \Delta t = t^{n+1} - t^n$ and $\mathbf{R}_I$ is the right hand side of Equation 9. The values $\alpha_1 = 0.6, \alpha_2 = 0.6$ and $\alpha_3 = 1$ are adopted for the Runge-Kutta coefficients [13]. A local time stepping approach is used to accelerate the convergence rate towards the steady-state.

It should be added that various techniques for implementation of higher-order upwind methods in the context of unstructured grids exist [7,8]. Most of these techniques construct a compact stencil to approximate the interface values to be used for higher-order inviscid flux computations.

JST ARTIFICIAL DISSIPATION METHOD

As a second scheme, the artificial dissipation scheme, due to Jameson et al., is used [11]. The flux function associated with this method consists of a blend of stabilization and discontinuity capturing operators, with a pressure sensor controlling the magnitude of the discontinuity capturing term. The solution is advanced by an explicit three-stage Runge-Kutta scheme. In the present context, the diffusion, D_I, added at a general node, I, is constructed as a blend of approximations to

second order and fourth order operators as:

$$\mathbf{D}_I = \sum_{S=1}^{m_I} \frac{36\min(\lambda_I, \lambda_{I_S})}{m_I + m_{I_S}} \Big(\varepsilon^{(2)}_{II_S} \frac{\mathbf{U}_{I_S} - \mathbf{U}_I}{m_I} - \varepsilon^{(4)}_{II_S} (\nabla^2 \mathbf{U}_{I_S} - \nabla^2 \mathbf{U}_I) \Big), \quad (18)$$

where the second order operator is approximated according to:

$$\nabla^2 \mathbf{U}_I \approx \frac{1}{m_I} \sum_{S=1}^{m_1} (\mathbf{U}_{I_S} - \mathbf{U}_I). \quad (19)$$

Here, λ is the maximum eigen-value of the Jacobian matrix, $l_j \partial \mathbf{F}^j / \partial \mathbf{U}$, in absolute value, where $\mathbf{I} = (l_1, l_2, l_3)$ is the unit vector in the direction of the edge, II_S. The parameters, $\varepsilon^{(2)}_{II_S}$ and $\varepsilon^{(4)}_{II_S}$, are defined by:

$$\varepsilon^{(2)}_{II_S} = \kappa^{(2)} \max(P_I, P_{I_S}),$$

$$\varepsilon^{(4)}_{II_S} = \max(0, \kappa^{(4)} - \kappa^{(2)} \max(P_I, P_{I_S}), \quad (20)$$

where $\kappa^{(2)}$ and $\kappa^{(4)}$ are user specified parameters (typically 0.4 and 0.2) and:

$$P_I = \sum_{S=1}^{m_I} (p_{I_S} - p_I) / \sum_{S=1}^{m_I} (p_{I_S} + p_I), \quad (21)$$

is the nodal value of a pressure switch. A three stage Runge-Kutta procedure is again employed to advance the solution from time level $t = t^n$ to time level $t = t^{n+1} = t^n + \Delta t$. Within each time step, this scheme is implemented in the form:

$$\begin{cases} \mathbf{U}_I^{(0)} = \mathbf{U}_I^n \\ \mathbf{U}_I^{(k)} = \mathbf{U}_I^n - \alpha_k \Delta t [\mathbf{M}_L]^{-1} (\mathbf{R}_I^{(k-1)} - \mathbf{D}_I^{(0)}) \\ \qquad \text{for } k = 1, 2, 3 \\ \mathbf{U}_I^{n+1} = \mathbf{U}_I^{(3)} \end{cases}, \quad (22)$$

Here, $\mathbf{R}_I^{k-1}$ represents the right hand side of Equation 9 computed at the stage $k-1$, while the added diffusion, $\mathbf{D}_I$, is held constant at the value computed at t^n. The values of α_k are the same as given previously.

An alternative form of adding artificial dissipation is the matrix dissipation technique. Swanson and Turkel [14] have shown that this form can improve the accuracy of the solution at the expense of higher CPU time.

TIME STEP SIZE

As mentioned before, in this work, a local time stepping technique is used. The local time stepping accelerates convergence by advancing the solution at each element in time at a CFL number near the local stability limit. The expression for the local time step is derived with the aid of a 2D stability analysis [15] and is given as:

$$(\Delta t)_I = 2\mathrm{CFL}[\mathbf{M}_L]_I \left[\sum_{S=1}^{m} \ell_{II_S} |\lambda_{II_S}| \right]^{-1}, \quad (23)$$

where λ_{II_S} is the maximum eigen-value of the system and CFL denotes the Courant number, which is constant in the domain. A typical value, CFL = 2, was used for the calculations presented in this paper. Finally, it should be noted that the use of a local time stepping technique is allowed only because the correct modeling of the transient development of the flow was not of interest here and for the steady-state computations this technique is accurate.

NUMERICAL RESULTS

To demonstrate the performance of the solution algorithms described above, the transonic inviscid flow over an ONERA M6 wing is solved. This is a well established benchmark problem and extensive experimental [16] and numerical [12,17] data exist for it. For the test case studied here, the Reynolds number is Re = 11.72E06, the Mach number M = 0.8395 and the angle of incidence $\alpha = 3.06$. Since this is a high Reynolds number flow problem, it is frequently used for the assessment of inviscid flow solvers. The flow exhibits interesting 3D flow features, including a λ-shape shock-shock interaction.

The geometry of the problem involves an ONERA-M6 wing and a symmetry plane. In Figure 2, a typical surface grid generated for this geometry is shown. The wing span is $b = 1.196$ and its mean aerodynamic cord is $c = 0.646$. It should be noted here

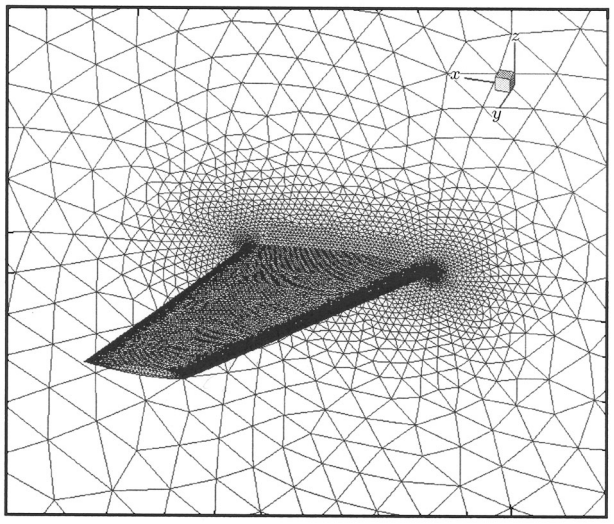

Figure 2. Wing and symmetry plane surface grid.

that all lengths are non-dimensional. In order to study the effect of surface mesh resolution, two surface grids shown in Figure 3 are used. The coarse grid has 20658 triangular surface elements, 309795 tetrahedral volume elements and 56380 nodal points. The fine grid has 29982 triangular surface elements, 449637 tetrahedral volume elements and 81830 nodal points.

Figure 4 shows two views of the surface pressure contour plots obtained using the JST method. It is seen that a λ-shape shock is captured. Figure 5 shows pressure contour plots at various wing sections; $y/b = 0.2, 0.65, 0.8$ and 0.9. Figure 6 shows the convergence history of the solution for the first 2000 time

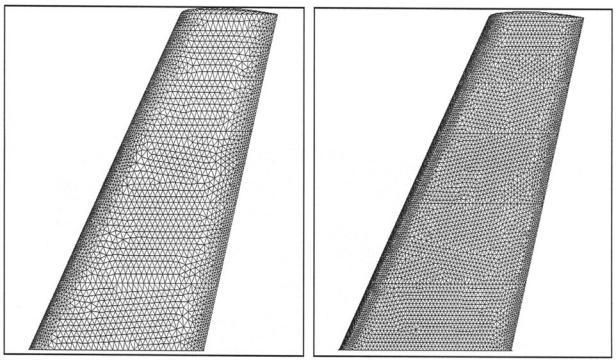

Figure 3. Wing surface grids (left: coarse, right: fine).

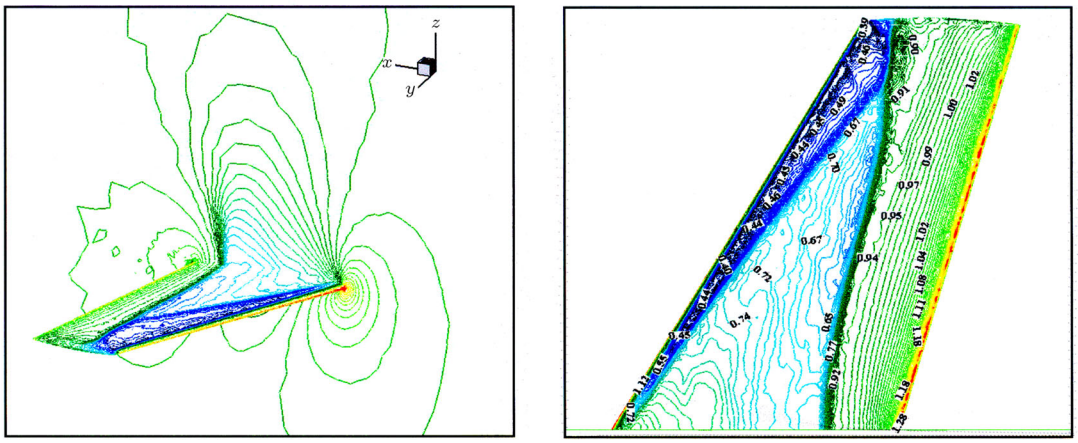

Figure 4. Surface pressure contour plots obtained by JST.

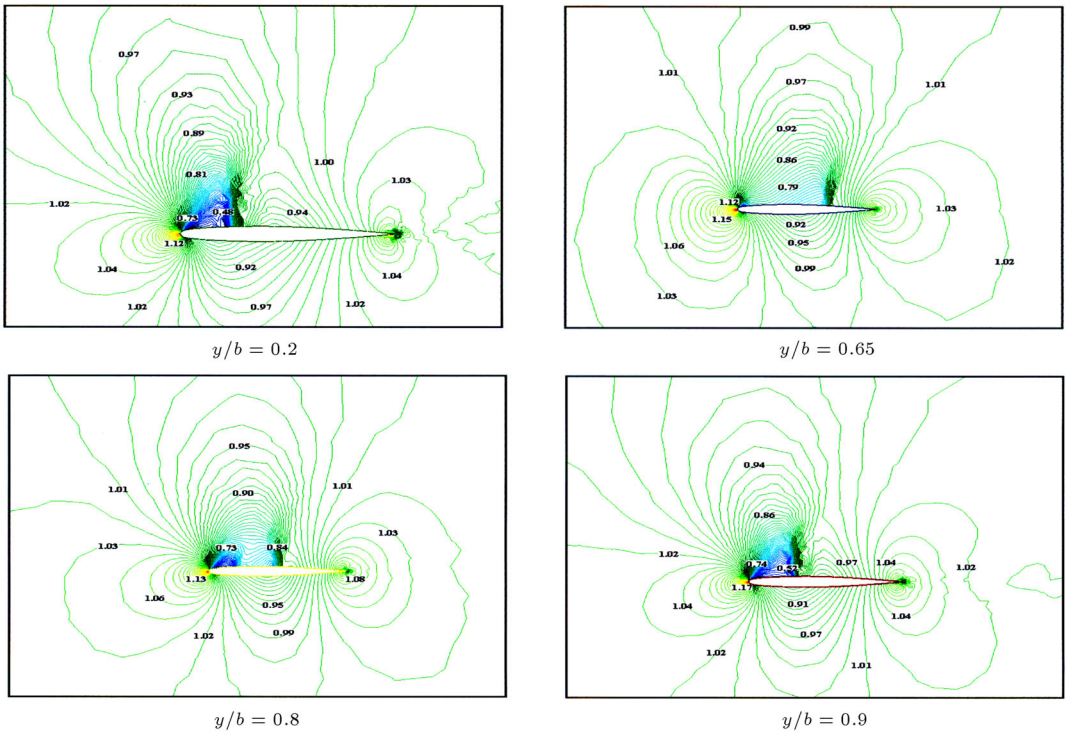

Figure 5. Pressure contour plots obtained using JST method at different wing sections.

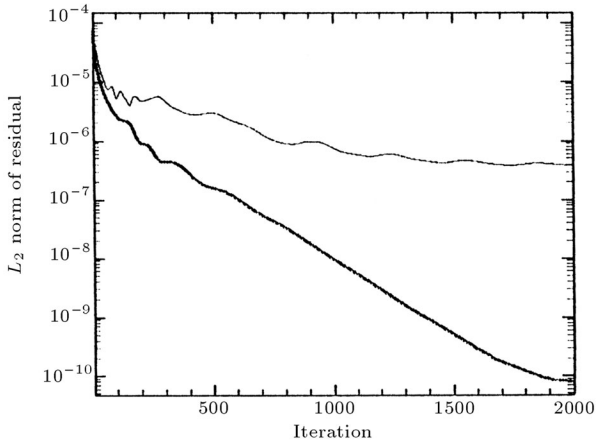

Figure 6. Convergence history of JST (thin line) and first-order upwind (thick line) methods.

steps of the computations, starting from a free-stream condition. Each time step of the calculations takes about six seconds on a Pentium III 600 processor. This is typical convergence behavior for the JST method and can be improved using multi-grid techniques. It should be mentioned that the solution procedure was allowed to continue up to 10000 time steps but no change in the results was observed.

In Figure 7, the pressure coefficients obtained by the JST method are compared with the experimental data. It is seen that the results are in fairly close agreement with the data in most sections. A better shock representation can be achieved by a local mesh refinement, which is outside the scope of this paper.

The problem was also solved using Roe's first-order upwind method and the pressure coefficients are

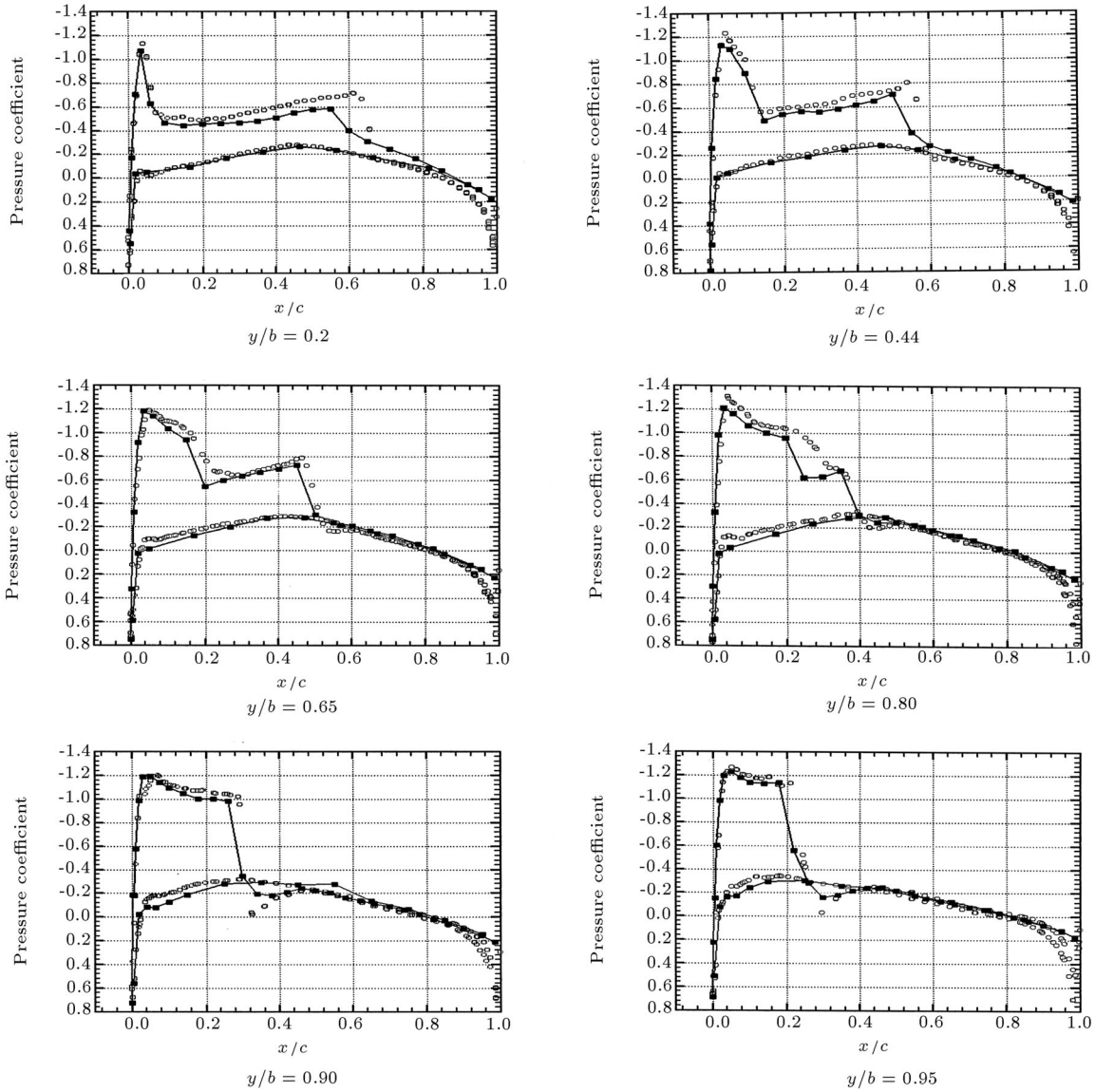

Figure 7. Comparison of calculated pressure coefficient distribution at different wing sections with experimental data. (○: JST, □: Experiment).

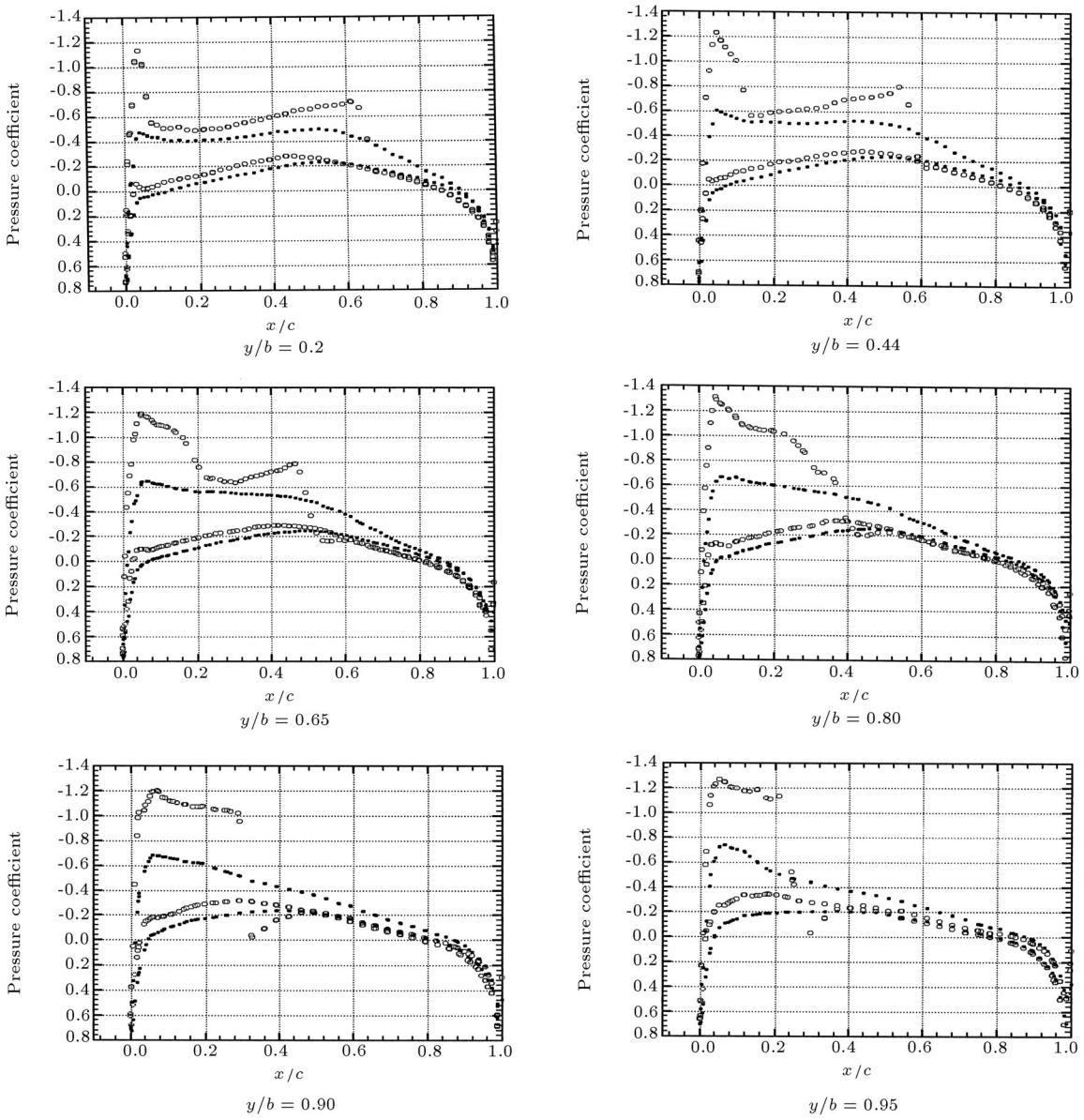

Figure 8. Comparison of calculated pressure coefficient distribution at different wing sections using different methods. (○: JST, □: First-order upwind).

compared with those of the JST method in Figure 8. As expected from a first-order scheme, the results are much more diffuse than the higher-order accurate JST method and most of the shock-shock interaction features of the problem are lost. The solution, however, is obtained much faster with only two seconds taken for each time-step. A valuable use of the first-order method can be during the start-up of the solution from the free-stream condition, where most of the higher-order methods encounter difficulties and require ad hoc treatments. For a comparison reason, the convergence history of the solution, using Roe's first order upwind method, is also shown in Figure 6. The curve shows that the solution has converged almost to machine zero within 2000 iterations.

Finally, to assess the mesh dependence of the solution, the problem was solved using the coarse mesh shown in Figure 3. A comparison of the results obtained on two meshes using the JST is shown in Figure 9. Again, it is seen that the improvement in the solution is restricted to the regions of high pressure gradients.

CONCLUSIONS

It has been demonstrated that both the first-order upwind and JST methods can be successfully incorporated in a finite element formulation for the simulation of compressible inviscid flows. While the JST method produces highly accurate results, both in the smooth

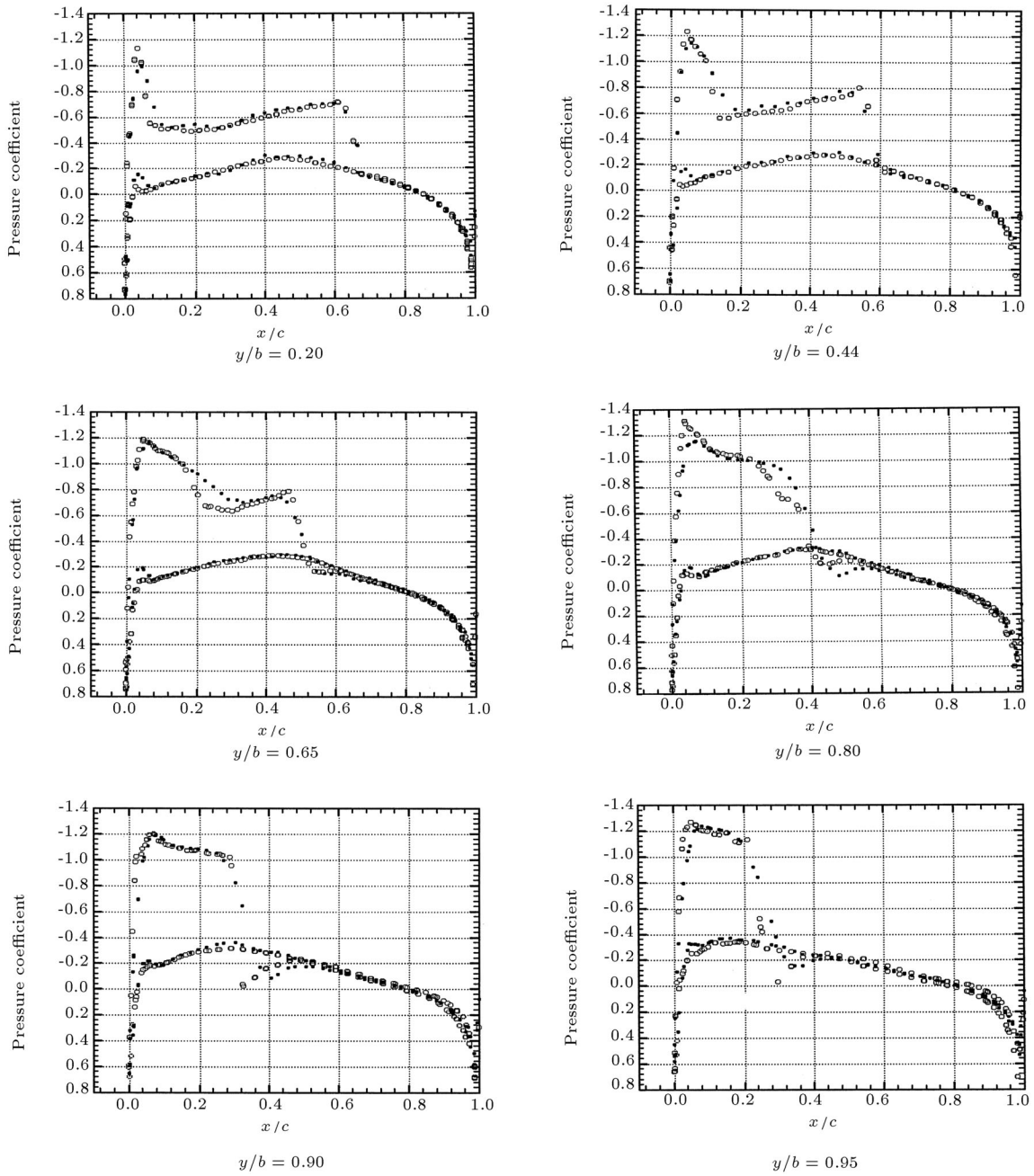

Figure 9. Comparison of calculated pressure coefficient distribution at different wing sections for different meshes using JST (○: fine, □: coarse).

and discontinuous flow regions for the problem studied here, the first-order upwind method, as expected, produces a more diffuse solution. It is proposed that the latter may be used as a start-up scheme for a higher-order scheme, in order to avoid numerical difficulties encountered when solving highly complex flow problems. Finally, it was shown that the JST method is only sensitive to mesh resolution wherever significant gradients are involved. In such cases, the lack of enough mesh resolution results in a smeared solution.

REFERENCES

1. Aftosmis, M.J. "Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries", *VKI Lecture Series 1997-02* (1997).

2. Mavriplis, D.J. and Levy, D.W. "Transonic drag prediction using an unstructured multigrid solver", *ICASE Report No. 2002-5* (2002).

3. Pirzadeh, S. "Three-dimensional unstructured viscous grids by the advancing-layers method", *AIAA Journal*, **34**, pp 43-49 (1996).

4. Peraire, J., Peiro, J. and Morgan, K. "Finite element multigrid solution of Euler flows past installed aero-engines", *Comp. Mech.*, **11**, pp 433-451 (1993).
5. Barth, T. "Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes", *AIAA Paper 91-0721* (1991).
6. Morgan, K. and Peraire, J. "Unstructured grid finite element methods for fluid mechanics", *Reports on Progress in Physics*, **61**, pp 569-638 (1998).
7. Lyra, P.R.M. and Morgan, K. "A review and comparative study of upwind biased schemes for compressible flow computation. Part I: 1-D first order schemes", *Archives of Computational Methods in Engineering*, **7**, pp 19-55 (2000).
8. Lyra, P.R.M. and Morgan, K. "A review and comparative study of upwind biased schemes for compressible flow computation. Part III: Multidimensional extension on unstructured grids", *Archives of Computational Methods in Engineering*, **9**, pp 207-256 (2003).
9. Manzari, M.T., Hassan, O., Morgan, K. and Weatherill, N.P. "Turbulent flow computations on 3D unstructured grids", *Finite Elements in Analysis and Design*, **30**, pp 335-352 (1998).
10. Roe, P. "Approximate Riemann solvers, parameter vectors and difference schemes", *J. Comp. Phys.*, **43**, pp 357-372 (1981).
11. Jameson, A., Schmidt, W. and Turkel, E. "Numerical simulation of the Euler equations by finite volume method using Runge-Kutta time stepping schemes", *AIAA Paper 81-1259* (1981).
12. Frink, N.T., Parikh, P. and Pirzadeh, S. "A fast upwind solver for the Euler equations on three-dimensional unstructured meshes", *AIAA Paper 91-0102* (1991).
13. Jameson, A. "Transonic flow calculations", Princeton University, Report MAE 1751 (1984).
14. Swanson, R.C. and Turkel, E. "On central-difference and upwind schemes", *J. Comp. Phys.*, **101**, pp 292-306 (1992).
15. Giles, M. "Energy stability analysis of multi-step methods on unstructured meshes", *MIT CFD Laboratory Report CFDL-TR-87-1* (1987).
16. Schmitt, V. and Charpin, F. "Pressure distributions on the ONERA-M6-wing at transonic Mach numbers", *Experimental Data Base for Computer Program Assessment*, Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138 (May 1979).
17. Parikh, P. "Application of a scalable, parallel, unstructured grid based Navier-Stokes solver", *AIAA-2001-2584 Paper, 15th AIAA CFD Conference* (June 2001).