Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes

Neal T. Frink*
NASA Langley Research Center, Hampton, Virginia 23665

An upwind scheme is presented for solving the three-dimensional Euler equations on unstructured tetrahedral meshes. Spatial discretization is accomplished by a cell-centered finite-volume formulation using flux-difference splitting. Higher-order differences are formed by a multidimensional linear reconstruction process. The solution gradients required for the higher-order differences are computed by a novel approach that yields highly resolved solutions in regions of smooth flow while avoiding oscillations across shocks without explicitly applying a limiter. Solutions are advanced in time by a three-stage Runge-Kutta time-stepping scheme with convergence accelerated to steady state by local time stepping and implicit residual smoothing. Transonic solutions are presented for two meshes around the ONERA M6 wing and demonstrate substantial accuracy and insensitivity to mesh size.

Introduction

SOLUTION algorithms for the Euler and Navier-Stokes equations on unstructured meshes have evolved rapidly in recent years; e.g., Refs. 1–9. Key motivations behind these developments include the need for more geometric flexibility in constructing quality meshes around complex configurations, and a random data structure to better facilitate adapting the mesh to the physics of the flow. Most of the unstructured algorithms developed to date are based on either the finite element method or central differencing with added dissipation. Only recently has upwind differencing been investigated[7–9] in two dimensions. Upwind differencing utilizes the propagation of information within a mesh in accordance with the theory of characteristics in constructing type-dependent differencing for components of the information traveling in opposite directions in a separate and stable manner. Although this approach is more computationally intensive than central differencing, it does offer the advantages of being more robust and requiring less user interaction.

In this paper, upwind differencing is explored in a three-dimensional unstructured environment. A cell-centered, finite-volume upwind scheme is formulated using flux-difference splitting. Higher-order accuracy is achieved by a multidimensional linear reconstruction process. The solution gradients required for higher-order differencing are computed by a simple new approach that yields highly resolved solutions in regions of smooth flow while avoiding oscillations across shocks without explicitly applying a limiter. Solutions are advanced in time by a three-stage Runge-Kutta time-stepping scheme with convergence accelerated to steady state by local time stepping and implicit residual smoothing.

Governing Equations

The fluid motion is governed by the time-dependent Euler equations for an ideal gas which express the conservation of mass, momentum, and energy for a compressible inviscid nonconducting adiabatic fluid in the absence of external forces. The equations are given below in integral form for a bounded domain Ω with a boundary $\partial\Omega$:

$$\frac{\partial}{\partial t}\iiint_{\Omega} \boldsymbol{Q}\mathrm{d}V + \iint_{\partial\Omega} \boldsymbol{F}(\boldsymbol{Q}) \cdot \hat{\boldsymbol{n}}\mathrm{d}S = 0 \tag{1}$$

where

$$\boldsymbol{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e_0 \end{Bmatrix}$$

and

$$\boldsymbol{F}(\boldsymbol{Q}) \cdot \hat{\boldsymbol{n}} = (\boldsymbol{V} \cdot \hat{\boldsymbol{n}}) \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e_0 + p \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \\ 0 \end{Bmatrix}$$

The equations are nondimensionalized with a reference density ρ_∞ and a speed of sound a_∞. Here $\hat{n}_x$, $\hat{n}_y$, and $\hat{n}_z$ are the Cartesian components of the exterior surface unit normal $\hat{\boldsymbol{n}}$ on the boundary $\partial\Omega$. The Cartesian velocity components are u, v, and w in the x, y, and z directions, respectively. The term e_0 is the total energy per unit volume. With the ideal-gas assumption, the pressure and total enthalpy can be expressed as

$$p = (\gamma - 1)[e_0 - \tfrac{1}{2}\rho(u^2 + v^2 + w^2)]$$

and

$$h_0 = \frac{\gamma}{\gamma - 1}\frac{p}{\rho} + \tfrac{1}{2}(u^2 + v^2 + w^2)$$

where γ is the ratio of specific heats and is prescribed as 1.4 for air.

Equation (1) describes a relationship where the time rate of change of the state vector $\boldsymbol{Q}$ within the domain Ω is balanced by the net flux $\boldsymbol{F}$ across the boundary surface $\partial\Omega$. The domain is divided into a finite number of tetrahedral cells, and Eq. (1) is applied to each cell. The state variables $\boldsymbol{Q}$ are volume-averaged values. It can be shown that the difference equations at each cell volume are satisfied exactly when evaluated at uniform freestream conditions.

Received June 12, 1990; revision received Jan. 30, 1991; accepted for publication Feb. 9, 1991.

*Research Engineer, Transonic Aerodynamics Branch, Applied Aerodynamics Division, Mail Stop 361.

Spatial Discretization

Flux Splitting

Flux quantities are computed using Roe's[10] flux-difference splitting. The flux across each cell face κ is computed using Roe's numerical flux formula:

$$\boldsymbol{F}_\kappa = \tfrac{1}{2}[\boldsymbol{F}(\boldsymbol{Q}_L) + \boldsymbol{F}(\boldsymbol{Q}_R) - |\tilde{\boldsymbol{A}}|\,(\boldsymbol{Q}_R - \boldsymbol{Q}_L)]_\kappa$$

Here $\boldsymbol{Q}_L$ and $\boldsymbol{Q}_R$ are the state variables to the left and right of the interface κ. The matrix $\tilde{\boldsymbol{A}}$ is computed from evaluating

$$\boldsymbol{A} \equiv \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}$$

with Roe-averaged quantities such as

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}$$

$$\tilde{u} = (u_L + u_R\sqrt{\rho_R/\rho_L})/(1 + \sqrt{\rho_R/\rho_L})$$

$$\tilde{v} = (v_L + v_R\sqrt{\rho_R/\rho_L})/(1 + \sqrt{\rho_R/\rho_L})$$

$$\tilde{w} = (w_L + w_R\sqrt{\rho_R/\rho_L})/(1 + \sqrt{\rho_R/\rho_L})$$

$$\tilde{h}_0 = (h_{0_L} + h_{0_R}\sqrt{\rho_R/\rho_L})/(1 + \sqrt{\rho_R/\rho_L})$$

$$\tilde{a}^2 = (\gamma - 1)(\tilde{h}_0 - (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2)/2)$$

so that

$$\boldsymbol{F}(\boldsymbol{Q}_R) - \boldsymbol{F}(\boldsymbol{Q}_L) = \tilde{\boldsymbol{A}}[\boldsymbol{Q}_R - \boldsymbol{Q}_L]$$

is satisfied exactly. Introducing the diagonalizing matrices $\tilde{\boldsymbol{T}}$ and $\tilde{\boldsymbol{T}}^{-1}$, and the diagonal matrix of eigenvalues $\boldsymbol{\Lambda}$, then $|\tilde{\boldsymbol{A}}|$ is defined as

$$|\tilde{\boldsymbol{A}}| = \tilde{\boldsymbol{T}}\,|\tilde{\boldsymbol{\Lambda}}|\,\tilde{\boldsymbol{T}}^{-1}$$

The term

$$|\tilde{\boldsymbol{A}}|\,(\boldsymbol{Q}_R - \boldsymbol{Q}_L) = \tilde{\boldsymbol{T}}\,|\tilde{\boldsymbol{\Lambda}}|\,\tilde{\boldsymbol{T}}^{-1}\,\Delta\boldsymbol{Q}$$

in Roe's flux formula can be reduced to three $\Delta\boldsymbol{F}$ flux components, each of which is associated with a distinct eigenvalue:

$$\tilde{\boldsymbol{T}}\,|\tilde{\boldsymbol{\Lambda}}|\,\tilde{\boldsymbol{T}}^{-1}\,\Delta\boldsymbol{Q} = |\Delta\tilde{\boldsymbol{F}}_1| + |\Delta\tilde{\boldsymbol{F}}_4| + |\Delta\tilde{\boldsymbol{F}}_5|$$

with

$$|\Delta\tilde{\boldsymbol{F}}_1| = |\tilde{U}|\left\{\left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2}\right)\begin{bmatrix}1\\ \tilde{u}\\ \tilde{v}\\ \tilde{w}\\ (\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)/2\end{bmatrix} + \tilde{\rho}\begin{bmatrix}0\\ \Delta u - \hat{n}_x\Delta U\\ \Delta v - \hat{n}_y\Delta U\\ \Delta w - \hat{n}_z\Delta U\\ \tilde{u}\Delta u + \tilde{v}\Delta v + \tilde{w}\Delta w - \tilde{U}\Delta U\end{bmatrix}\right\}$$

$$|\Delta\tilde{\boldsymbol{F}}_{4,5}| = |\tilde{U} \pm \tilde{a}|\left(\frac{\Delta p \pm \tilde{\rho}\tilde{a}\Delta U}{2\tilde{a}^2}\right)\begin{bmatrix}1\\ \tilde{u} \pm \hat{n}_x\tilde{a}\\ \tilde{v} \pm \hat{n}_y\tilde{a}\\ \tilde{w} \pm \hat{n}_z\tilde{a}\\ \tilde{h}_0 + \tilde{U}\tilde{a}\end{bmatrix}$$

where $\tilde{U} = \tilde{u}\hat{n}_x + \tilde{v}\hat{n}_y + \tilde{w}\hat{n}_z$ and $\Delta U = \hat{n}_x\Delta u + \hat{n}_y\Delta v + \hat{n}_z\Delta w$.

For a first-order scheme, the state of the primitive variables at each cell face are set to the cell-centered averages on either side of the face.

Higher-Order Scheme

Following the notions presented by Barth and Jespersen[7] in two dimensions, a higher-order scheme is formulated by reconstructing a multidimensional piecewise linear solution of the cell-averaged data. Higher-order interpolations for estimating the state at each cell face are achieved by expanding the solution with a Taylor series in the neighborhood of each cell center (x_0, y_0, z_0):

$$\begin{aligned}\boldsymbol{q}(x, y, z) &= \boldsymbol{q}(x_0, y_0, z_0) + \nabla\boldsymbol{q}_0 \cdot \Delta\boldsymbol{r} + \mathcal{O}(\Delta r^2)\\ &\approx \boldsymbol{q}(x_0, y_0, z_0) + \boldsymbol{q}_x|_0 \cdot (x - x_0)\\ &\quad + \boldsymbol{q}_y|_0 \cdot (y - y_0) + \boldsymbol{q}_z|_0 \cdot (z - z_0)\end{aligned} \tag{2}$$

where

$$\boldsymbol{q} \equiv [\rho, u, v, w, p]^T$$

This formulation requires solution-gradient information at the cell centers. The gradients are computed from an approximate form of the exact relation:

$$\int_\Omega \nabla\boldsymbol{q}\,dV = \oint_{\partial\Omega} \boldsymbol{q}\hat{\boldsymbol{n}}\,dS \tag{3}$$

The three-dimensional solution gradient at the cell center is estimated by computing the surface integral of Eq. (3) for some closed surface $\partial\Omega$:

$$\nabla\boldsymbol{q}_0 = \frac{1}{V_\Omega}\oint_{\partial\Omega} \boldsymbol{q}\hat{\boldsymbol{n}}\,dS \tag{4}$$

In two dimensions, Eq. (4) represents a line integral around some closed path around the cell center.

Implementation of the numerical method proposed by Barth and Jespersen for solving the integral of Eq. (4) was not straightforward in three dimensions. A simpler higher-order method has been developed and is described in the following. The general approach is to 1) coalesce surrounding cell information to the vertices or nodes of the candidate cell, and 2) evaluate the surface integral of Eq. (4) on the faces of the tetrahedral cell with the midpoint-trapezoidal rule. For illustration purposes, the present approach is first sketched for a two-dimensional problem as shown in Fig. 1. Estimates of the solution are determined at each node by a weighted average of the surrounding cell-centered solution quantities. It is assumed in the nodal averaging procedure that the known values of the solution are concentrated at the cell centers, and that the contribution to a node from the surrounding cells is in-

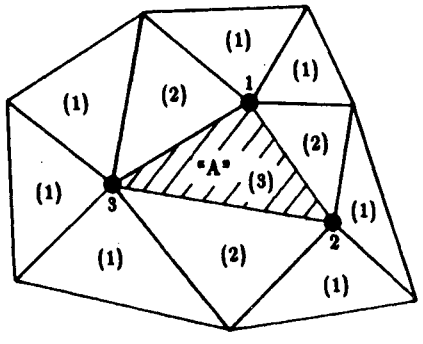

Fig. 1 Cluster of two-dimensional triangular cells.

versely proportional to the distance from each cell centroid to the node:

$$\boldsymbol{q}_n = \left(\sum_{i=1}^{N} \frac{\boldsymbol{q}_{0,i}}{r_i}\right) \Big/ \left(\sum_{i=1}^{N} \frac{1}{r_i}\right) \tag{5}$$

where for two dimensions

$$r_i = [(x_{0,i} - x_n)^2 + (y_{0,i} - y_n)^2]^{\frac{1}{2}}$$

and for three dimensions

$$r_i = [(x_{0,i} - x_n)^2 + (y_{0,i} - y_n)^2 + (z_{0,i} - z_n)^2]^{\frac{1}{2}}$$

The subscripts n and $0,i$ refer to the node and surrounding cell-centered values, respectively. Upon knowing the solution at the nodes, the solution to Eq. (4) can be numerically evaluated by applying the midpoint-trapezoidal rule to the three sides of the triangle.

Once the gradient is computed in a cell, $\boldsymbol{q}$ can be expanded to the cell edge with a Taylor series to achieve higher-order accuracy. Because of the relative positions and number of times each cell has been used in the gradient computation (numbers enclosed in parenthesis in Fig. 1), the cell-interface values are biased toward cell A. For example, if the state is being computed on edge 1–2 from the state and gradient at the centroid of cell A, the interpolation will utilize information from surrounding cells on both sides of the edge, but be weighted in the direction of cell A. Thus, this procedure has the general features of the three-point interpolation formulas commonly used in upwind-biased structured schemes.[11]

Upwind schemes generally require the use of limiter functions to obtain smooth higher-order solutions around flow discontinuities, which has been the experience with the schemes of Refs. 7–9. Although the present method has yet to be applied in two dimensions, experience with its application in three dimensions has shown that limiting need not be applied explicitly, and that the method correctly captures shocks without oscillations. Although this unexpected result is beneficial, it seems unlikely that a high-order scheme could capture discontinuities smoothly without some form of artificial dissipation being added. It is quite possible that the application of the averaging procedure across discontinuities could introduce additional dissipation and a local reduction in accuracy as is characteristic of limiters. This result may also be attributed in part to the use of the method in three dimensions and therefore on relatively coarse meshes. Application of the method in two dimensions on a sufficiently fine mesh may require some form of limiting to eliminate oscillations. With respect to limiting overshoots in the expansion of Eq. (2), it can be reasoned that when the averaging procedure of Eq. (5) is applied at a node, the resulting $\boldsymbol{q}_n$ represents a weighted mean value of the surrounding solution, i.e., $\boldsymbol{q}_n$ is bounded by the extrema of the surrounding solution. Furthermore, in three dimensions the summation of Eq. (5) accesses an average of 20–22 cells for each node, which results in a smoothing of errors introduced from the surrounding solution. Thus, the expansion will have been smoothed and bounded by the procedure and should not introduce new extrema into the solution.

The accuracy of the higher-order scheme has not been formally determined. It is known that the Taylor-series expansion and the midpoint-trapezoidal rule are both second-order accurate. Thus, the key to accuracy for this method lies in the quality of the averaged solution at the nodes. The goal in designing the averaging procedure was to construct a good approximation of the state at the nodes. No attempt was made to preserve the average. Several procedures were investigated with known test functions, both linear and nonlinear, on an arbitrary tetrahedral grid. Among the alternate procedures were those based on cell volumes, and on the inverse distances from cell center to node raised to a power. Error was assessed by an rms average of the local errors relative to the exact solution. Volume averaging yielded the poorest result, while averaging based on the inverse distance raised to the first power [Eq. (5)] worked best. An application of this averaging procedure to a linear function on an irregular mesh does not reconstruct the function exactly, but produces a good approximation.

Time Integration

A semidiscrete form of the governing equations reads

$$V_i \frac{\partial \boldsymbol{Q}_i}{\partial t} + \boldsymbol{R}_i = 0, \qquad i = 1, 2, 3, \ldots \tag{6}$$

where

$$\boldsymbol{R}_i = \sum_{j=\kappa(i)} \boldsymbol{F}_{i,j} \Delta S_{i,j}$$

$\boldsymbol{R}_i$ is the residual accrued by summation of the fluxes through the four faces κ of a tetrahedral cell i. These equations are integrated in time using a fully explicit m-stage Runge-Kutta time-stepping scheme developed by Jameson et al.[12]:

$$\begin{aligned}
\boldsymbol{Q}_i^{(0)} &= \boldsymbol{Q}_i^n \\
\boldsymbol{Q}_i^{(1)} &= \boldsymbol{Q}_i^{(0)} - \alpha_1 \frac{\Delta t}{V_i} \boldsymbol{R}_i^{(0)} \\
&\ldots \\
\boldsymbol{Q}_i^{(m-1)} &= \boldsymbol{Q}_i^{(0)} - \alpha_{m-1} \frac{\Delta t}{V_i} \boldsymbol{R}_i^{(m-2)} \\
\boldsymbol{Q}_i^{(m)} &= \boldsymbol{Q}_i^{(0)} - \alpha_m \frac{\Delta t}{V_i} \boldsymbol{R}_i^{(m-1)} \\
\boldsymbol{Q}_i^{n+1} &= \boldsymbol{Q}_i^{(m)}
\end{aligned} \tag{7}$$

where the superscript n denotes the time level, and the parenthetical superscripts the stage of the Runge-Kutta time stepping. The coefficients α_1 to α_m are defined as

$$\alpha_k = \frac{1}{m - k + 1}, \qquad k = 1, \ldots, m$$

These values of α_k will give m-order accuracy in time for a linear equation. For steady-state calculations, it is not necessary to employ the time-accurate Runge-Kutta coefficients. Coefficients that optimize speed of convergence could be used, but were not for the present results. Preliminary calculations were made using both a three-stage and four-stage scheme. The solution and convergence characteristics were essentially identical. Thus, a three-stage scheme was used for the calculations presented in this paper.

Local Time Stepping

Local time stepping accelerates convergence to steady state by advancing the solution at each cell in time at a Courant-Friedricks-Leury (CFL) number near the local stability limit. The expression for the local time step was derived with the aid of a two-dimensional stability anaysis presented in Ref. 9:

$$\Delta t_i \leq \nu \frac{V_i}{A_i + B_i + C_i} \tag{8}$$

with

$$A_i = (|u_i| + a_i) S_i^{(x)}$$

$$B_i = (|v_i| + a_i)S_i^{(y)}$$

$$C_i = (|w_i| + a_i)S_i^{(z)}$$

where v is the CFL number, V_i is the cell volume, a_i is the local speed of sound, and $S_i^{(x)}$, $S_i^{(y)}$, and $S_i^{(z)}$ are the projected areas of cell i in the x, y, and z directions, respectively. The local time steps are determined at the beginning of a run, either at the initialization or before a restart.

Implicit Residual Smoothing

The maximum time step can be further increased by increasing the support of the scheme through implicit averaging of the residuals[13] with their neighbors. The residuals are filtered through a smoothing operator (which is essentially the Laplacian operator for a uniform grid):

$$\overline{\boldsymbol{R}}_i = \boldsymbol{R}_i + \varepsilon \nabla^2 \overline{\boldsymbol{R}}_i$$

where

$$\nabla^2 \overline{\boldsymbol{R}}_i = \sum_{j=\kappa(i)} (\overline{\boldsymbol{R}}_j - \overline{\boldsymbol{R}}_i)$$

The summation uses residuals from the neighboring cells that share the faces κ with cell i. The resulting set of equations can easily be solved by using Jacobi iteration

$$\overline{\boldsymbol{R}}_i^{(m)} = \left(\boldsymbol{R}_i + \varepsilon \sum_{j=\kappa(i)} \overline{\boldsymbol{R}}_j^{(m-1)}\right) \Big/ \left(1 + \varepsilon \sum_{j=\kappa(i)} 1\right) \quad (9)$$

A value for ε of about 0.5 is suggested in Ref. 14 to maintain a strongly diagonally dominant coefficient matrix. In practice, two Jacobi iterations are adequate to give a good approximation of $\overline{\boldsymbol{R}}_i$ at all cell centers. Residual smoothing was performed during every stage of the Runge-Kutta time cycle and allowed a doubling of the time step.

Boundary Conditions

For the solid boundaries such as the wing and centerplane, the flow trangency condition is imposed by setting the velocities on the boundary faces to their cell-center values and then subtracting the component normal to the solid surface. Density and pressure boundary conditions are simply set to the cell-centered value. A condition of zero mass and energy flux through the surface is ensured by setting the left and right states of solid boundary faces equal to the boundary conditions prior to computing the fluxes with Roe's approximate Riemann solver. This technique only permits a flux of the pressure terms of the momentum equations through a solid boundary.

Characteristic boundary conditions are applied to the outer far-field subsonic boundary using the fixed and extrapolated Riemann invariants corresponding to the incoming and outgoing waves. The incoming Riemann invariant is determined from the freestream flow and the outgoing invariant is extrapolated from the interior domain. The invariants are used to determine the locally normal velocity component and speed of sound. At an outflow boundary, the two tangential velocity components and the entropy are extrapolated from the interior, while at an inflow boundary they are specified as having far-field values. These five quantities provide a complete definition of the flow in the far field.

Results

Solutions were computed for the ONERA M6 wing on two tetrahedral meshes at transonic conditions: $M_\infty = 0.84$, and $\alpha = 3.06$ deg. The wing has a leading-edge sweep of 30 deg, an aspect ratio of 3.8, a taper ratio of 0.56, and symmetrical airfoil sections. The two meshes, shown in Figs. 2 and 3, were generated by the advancing front technique.[15] The wing has

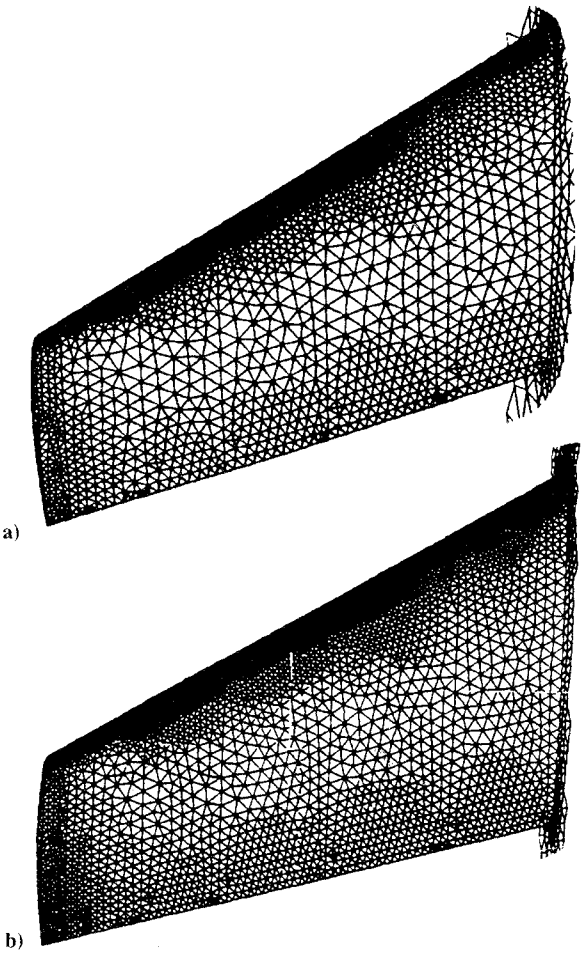

Fig. 2 Upper surface mesh for ONERA M6 wing: a) mesh 1; and b) mesh 2.

a root chord of 0.67 and a semispan b of 1.0 with a rounded tip. The computational domain is bounded by a rectangular box with boundaries at $-6.5 \le x \le 11.0$, $0.0 \le y \le 2.5$, and $-6.5 \le z \le 6.5$. Mesh size specifications are listed in Table 1.

The computations were performed using the three-stage Runge-Kutta time-stepping scheme with local time stepping and implicit residual smoothing. All time steps were advanced with a CFL number of 4.0. The convergence history of the L_2-norm (rms average of all residuals) is shown in Fig. 4 where both runs show a decrease of approximately 2.5 orders of magnitude. The solutions were started from freestream initial conditions with the first-order scheme and run until the L_2-norm decreased by at least one order of magnitude. The first-order solution was then used as an initial conditions for the higher-order computations. The solution for mesh 1 was advanced 300 time steps with the first-order scheme and 1420 steps with the higher-order scheme. Similarly, mesh 2 was advanced 460 steps first-order and 1730 steps higher-order. The reason for the spikes observed in the residual plot of Fig. 4b is not clear. They may be related to local disturbances induced by the shock waves moving through the irregular mesh as the solution converges to steady state. An additional 1000 iterations were computed on mesh 2 to verify that no further spiking occurred.

Before plotting the results in this paper, the *computed* face-centered boundary quantities were *averaged* to the boundary nodes using Eq. (5). This step was necessary in order to utilize existing contouring software. (It is also the same step used in the gradient computation.) Since there are approximately one-half the number of boundary nodes as boundary faces (see Table 1), some of the available spatial resolution is absent

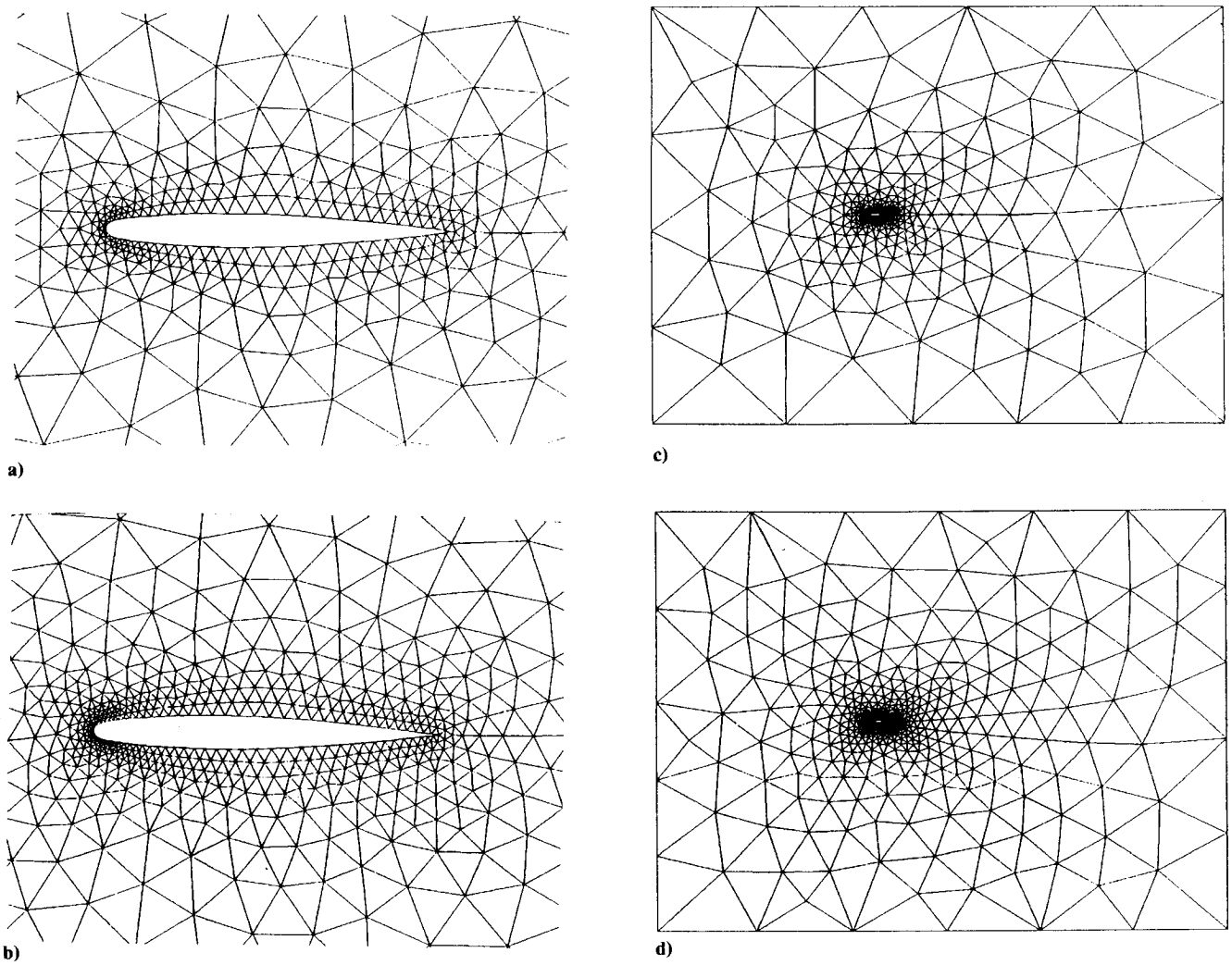

Fig. 3 Mesh at symmetry plane for ONERA M6 wing: a) mesh 1 (near-field); b) mesh 2 (near-field); c) mesh 1 (far-field); and d) mesh 2 (far-field).

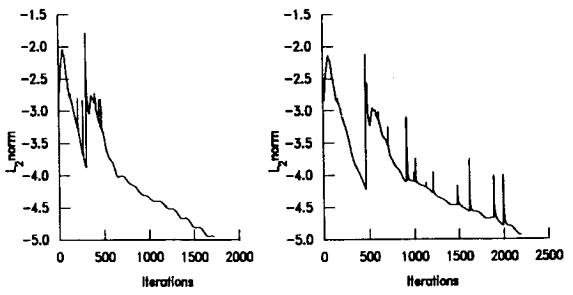

Fig. 4 Convergence history of ONERA M6 wing: a) mesh 1; and b) mesh 2 ($M_\infty = 0.84$, $\alpha = 3.06$ deg).

Table 1 Mesh size specifications

	Mesh 1	Mesh 2
Total cells	108,755	231,507
Global boundary faces	9,858	16,984
Faces on wing surface	8,807	15,279
Total nodes	20,412	42,410
Global boundary nodes	4,931	8,494
Nodes on wing surface	4,434	7,680

Table 2 Force and moment coefficients

	Mesh 1	Mesh 2
C_L	0.2904	0.2911
C_D	0.0132	0.0123
C_m	−0.1724	−0.1726
C_{RBM}	0.1283	0.1285

from the plotted results. However, the force and moment coefficients listed in Table 2 were computed by integrating the face-centered boundary pressures, and thus reflect the full spatial resolution of the solution. The coefficients for lift, drag, pitching moment, and wing root bending moment are based on reference quantities of $S_{\text{ref}} = 0.5255$, $\bar{c} = 0.67$, and $b_{\text{ref}} = 1.0$. The pitching moment is referenced about the wing apex.

A comparison of wing surface pressure contours for the two meshes is presented in Fig. 5 with contour intervals of $\Delta(p/p_\infty) = 0.02$. The results show very little sensitivity to mesh size. The primary differences occur with the resolution of the aft shock in the midchord region where the larger cells (see Fig. 2) limit the spatial resolution. A solution-adapted mesh could enhance the solution quality in vicinity of the shock.[16]

Pressure contours at the plane of symmetry are presented in Fig. 6 and again exhibit very little sensitivity to mesh size. The shock has been captured but is smeared due to the coarseness of the mesh (see Fig. 3).

Figure 7 shows the effect of mesh size on the streamwise surface C_p distribution at six span stations. Both the present unstructured results and structured Euler solutions from Ref. 17 are plotted in comparison to experimental data at a Rey-

nolds number of 11.7×10^6 (Ref. 18), corresponding to conditions for which viscous effects are relatively small. The computations on both tetrahedral meshes 1 and 2 agree well with experiment and demonstrate comparable accuracy with the structured-grid calculations. The primary effect of mesh size is confined to regions of large changes in gradient such as the leading-edge suction peak and the shock, where the finer mesh 2 yields slightly better spatial resolution.

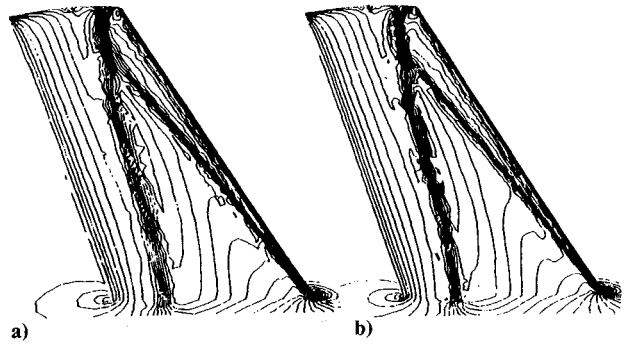

Fig. 5 Upper surface pressure contours for ONERA M6 wing: a) mesh 1; and b) mesh 2 ($M_\infty = 0.84$, $\alpha = 3.06$ deg, $\Delta(p/p_\infty) = 0.02$).

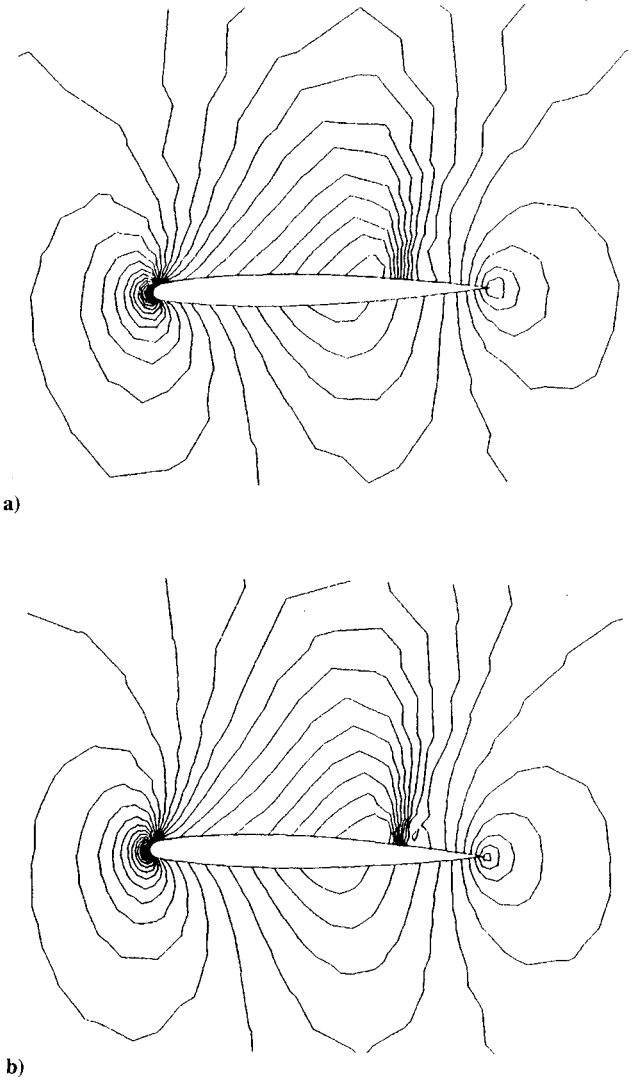

Fig. 6 Pressure contours at symmetry plane for ONERA M6 wing: a) mesh 1; and b) mesh 2 ($M_\infty = 0.84$, $\alpha = 3.06$ deg, $\Delta(p/p_\infty) = 0.02$).

The chordwise entropy distributions presented in Fig. 8 are defined by the following relation:

$$\gamma p/\rho^\gamma - 1$$

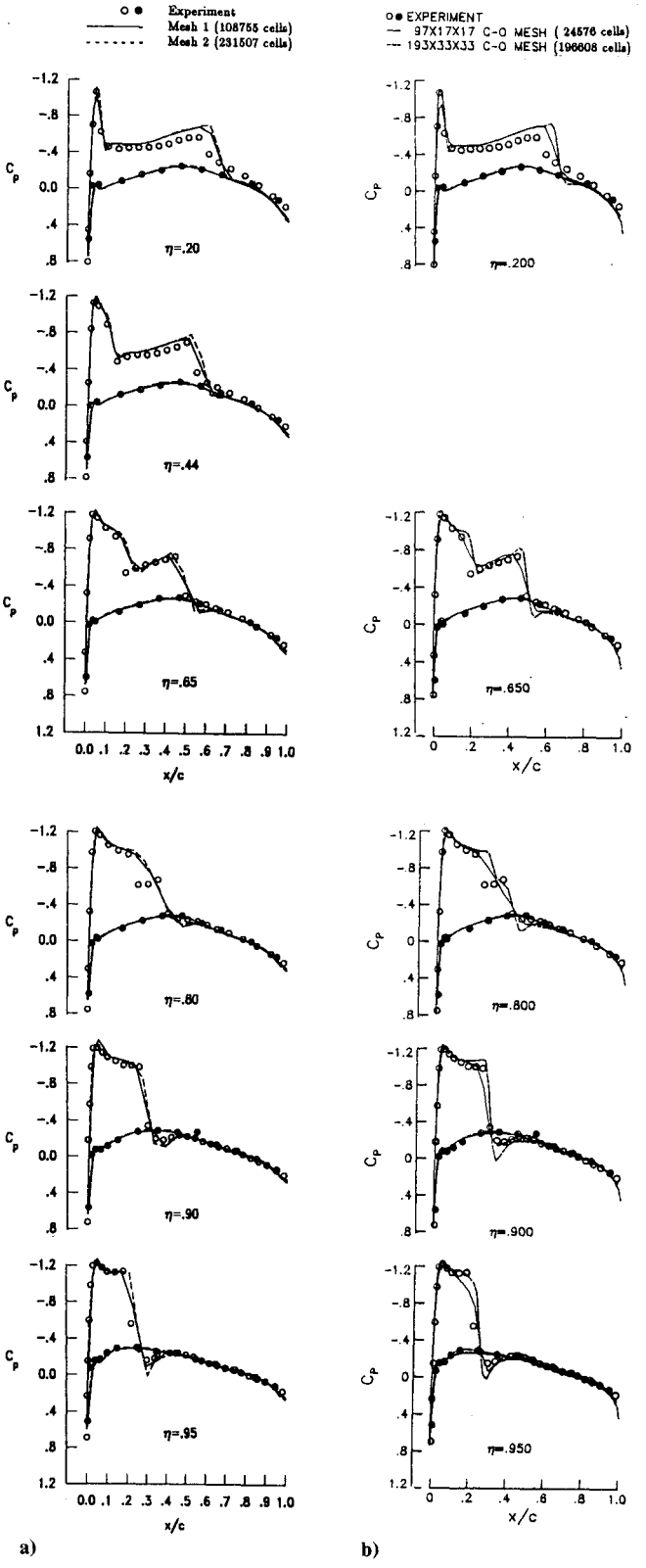

Fig. 7 Comparison of experimental data with unstructured and structured inviscid solutions: a) unstructured; and b) structured (Ref. 17) (ONERA M6 wing, $M_\infty = 0.84$, $\alpha = 3.06$ deg).

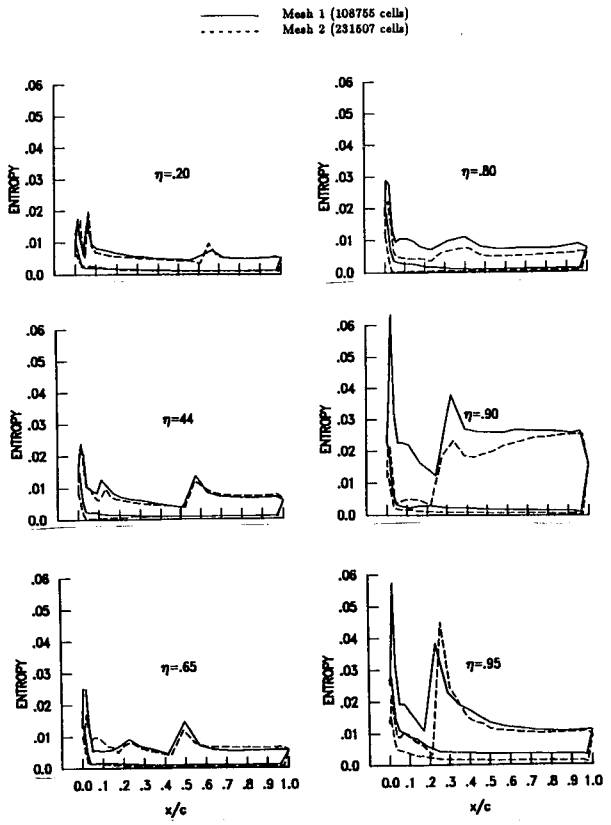

Fig. 8 Effect of mesh size on chordwise entropy distribution for ONERA M6 wing (M_∞ = 0.84, α = 3.06 deg).

For these supercritical inviscid solutions, the flow should be isentropic ahead of the shocks with zero entropy production. Entropy should only be produced through the shock waves and convected downstream thereafter. Any generation of entropy apart from shocks or even an unphysical reduction of entropy can be only interpreted as numerical error. Figure 8 shows the presence of erroneous entropy peaks at the leading edge approaching a level of 0.06 for the coarser mesh 1. Refinement of the mesh (mesh 2) results in a reduction in these entropy peaks to a level of approximately 0.02. The finer mesh 2 also produces a further decrease of the entropy convected downstream of the erroneous peaks. Further reductions in the peaks may be possible by additional mesh refinement or by applying alternate boundary conditions that maintain a zero gradient of entropy normal to the solid surface.[19] The entropy rise through the shocks generally overshoots but settles to a constant value.

The solutions were computed on a single processor of the NASA Langley Research Center Voyager CRAY-2S. All coding within the flow-solver portion of the code vectorizes with the standard Fortran compiler. The computations were performed with three stages of the Runge-Kutta scheme and two Jacobi iterations for the implicit residual smoothing. Mesh 1 required approximately 3 h and mesh 2 required 8 h of CRAY-2S time. The dimensioned memory requirements are shown in Table 3 and the computation times per cycle in Table 4.

For comparison, structured Euler codes generally require 40–50 words of memory per cell and 25–35 μs of CRAY-2S time per cell for higher-order solutions.

To avoid possible confusion, it should be noted that a three-dimensional structured mesh of hexahedral elements contains the same number of nodes as cells (asymptotically), whereas an unstructured mesh of tetrahedral elements generally contains between five and six times more cells than nodes. The results shown in Fig. 7 demonstrate that comparable accuracy

Table 3 Dimensional memory requirements

	Mesh 1	Mesh 2
Total	7,959,706	16,817,182
Per cell	73	73
Per node	390	397

Table 4 Computational time per cycle

	Per cell, μs	Per node, μs
First order	18	97
Higher order	65	350

can be achieved between structured and cell-centered unstructured codes where the number of cells are of the same order. Thus, it is important to make comparisons based on the number of unknowns computed, i.e., the number of cells for a cell-centered scheme.

Concluding Remarks

An upwind scheme for solving the three-dimensional Euler equations on unstructured tetrahedral meshes has been presented. The algorithm consists of a time-explicit cell-centered finite-volume formulation using flux-difference splitting. Higher-order accuracy was achieved by a multidimensional linear reconstruction process. A simple method was proposed for computing gradients that permitted smooth higher-order solutions to be obtained around shocks without explicitly applying a limiter. Transonic solutions are computed for two meshes around an ONERA M6 wing with a CFL number of 4 using a three-stage Runge-Kutta time-stepping scheme, local time stepping, and implicit residual smoothing. Results showed good quantitative agreement with experimental data and demonstrated comparable accuracy with referenced structured-grid upwind solutions. Contours of surface pressure showed very little sensitivity to mesh size. Some mesh sensitivity was observed in the surface entropy distributions.

Acknowledgments

The author wishes to thank his Ph.D. advisor, R. W. Walters of Virginia Polytechnic Institute and State University, for many stimulating discussion. Special appreciation is also extended to Shahyer Pirzadeh and Paresh C. Parikh of ViGYAN and to Clyde R. Gumbert of NASA Langley for generating the additional coarse mesh and assisting with the postprocessing of solutions.

References

[1]Jameson, A., and Baker, T. J., "Improvements to the Aircraft Euler Method," AIAA Paper 87-0452, Jan. 1987.

[2]Mavriplis, D. J., "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes," *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 213–221.

[3]Stoufflet, B., Périaux, J., Fezoui, F., and Dervieux, A., "Numerical Simulation of 3-D Hypersonic Euler Flows Around Space Vehicles Using Adapted Finite-Elements," AIAA Paper 87-0560, Jan. 1987.

[4]Desideri, J. A., and Dervieux, A., "Compressible Flow Solvers Using Unstructured Grids," *von Karman Institute Lecture Series*, von Karman Institute, Brussels, Belgium, March 1988, pp. 1–115.

[5]Rostand, P., and Stoufflet, B., "TVD Schemes to Compute Compressible Viscous Flows on Unstructured Meshes," *Nonlinear Hyperbolic Equations*, edited by Josef Ballmann and Rolf Jeltsch, F. Vieweg, Publisher, Braunschweig, Germany, 1989, pp. 510–520.

[6]Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aeroelastic Analysis," AIAA Paper 89-1189, April 1989.

[7]Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[8]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 89-0364, Jan. 1989.

[9]Whitaker, D. L., "Two-Dimensional Euler Computations on a Triangular Mesh Using an Upwind, Finite-Volume Scheme," Ph.D. Dissertation, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, 1988.

[10]Roe, P. L., "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337–365.

[11]Thomas, J. L., and Walters, R. W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," *AIAA Journal*, Vol. 25, No. 4, 1987, pp. 527–534.

[12]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite-Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[13]Jameson, A., and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper 83-1929, July 1983.

[14]Mavriplis, D., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes," *AIAA Journal*, Vol. 26, No. 7, 1988, pp. 824–831.

[15]Löhner, R., and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing Front Method," AIAA Paper 88-0515, Jan. 1988.

[16]Slack, D. C., Walters, R. W., and Löhner, R., "An Interactive Adaptive Remeshing Algorithm for the Two-Dimensional Euler Equations," AIAA Paper 90-0331, Jan. 1990.

[17]Anderson, W. K., Thomas, J. L., and Whitfield, D. L., "Multigrid Acceleration of the Flux-Split Euler Equations," *AIAA Journal*, Vol. 26, No. 6, 1988, pp. 649–654.

[18]Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA M6-Wing at Transonic Mach Number," AGARD Advisory Rept. 138, May 1979.

[19]Barton, J. T., and Pulliam, T. H., "Airfoil Computations at High Angles of Attack, Inviscid and Viscous Phenomena," *AIAA Journal*, Vol. 24, No. 5, 1986, pp. 705–714.