Solution-Adaptive Grid Generation Using Parabolic Partial Differential Equations

Ralph W. Noack* and Dale A. Anderson†
The University of Texas at Arlington, Arlington, Texas

A solution-adaptive parabolic grid generation scheme has been developed. A new upwind finite-volume formulation for space marching solution of the steady Euler equations is also presented. In addition, a solution-adaptive elliptic grid scheme developed previously and applied to unsteady flow solutions is applied in the present steady-flow context. Solutions for hypersonic flow are computed with the present finite-volume formulation coupled to both the parabolic and elliptic adaptive grid schemes. The results show that the solution-adaptive parabolic scheme produces grids that track the flow features of interest as well as the elliptic grid procedure. This is significant because the parabolic procedure is much faster than an elliptic scheme producing a grid in an order of magnitude of less computational time.

Introduction

THE task of grid generation can be described as the definition of a collection of grid points on which the solution of some equation is to be obtained. The "optimum" collection of points is generally desired, where the exact definition of optimum is open to discussion. Very little in the way of exact guidance is available for the determination of the optimum grid generation technique.

The search for an optimum grid will generally involve an attempt to minimize the error in the solution. One fundamental requirement that arises when trying to minimize the error is that the grid must be smooth. This need for smoothness naturally leads to the use of Laplace's equation as a vehicle for generation of the grid. The popularity of the use of Laplace's equation[1] is a direct result of the smoothness properties of solutions of this equation along with the existence of an extremum principle, which guarantees a one-to-one mapping between the physical and transformed regions. Finally, well-established methods are available to solve Laplace's equation.

The generation of a grid based on the solution of Laplace's equation can consume a large amount of computational time. This consideration led researchers[2] to consider schemes based on hyperbolic-type partial differential equations. The speed advantage of a hyperbolic scheme over an elliptic scheme is great because the hyperbolic generator will provide a grid in approximately one relaxation cycle of the elliptic solver. Yet a hyperbolic scheme has some undesirable properties. Notably, discontinuities in the initial data are propagated into the grid. This can lead to grid crossing or singular mappings, even though the Jacobian is usually specified. Furthermore, the outer boundary grid point locations cannot be specified.

The search for a scheme that combines the smoothing property of elliptic grid equations and the efficiency of hyperbolic grid generation equations led Nakamura[3] to study the use of parabolic partial differential equations as a grid generation method. This entails the solution of an initial value problem so that grids can be generated by a marching procedure just as for hyperbolic schemes. In addition, like the elliptic equations, the parabolic equations are diffusive in nature. This will smooth out any discontinuities which may be in the initial data of the grid. Finally, unlike hyperbolic schemes, parabolic schemes allow the outer boundary grid point locations to be specified.

Regardless of the grid generation scheme used, it is important to cluster grid points in regions of high gradients. Otherwise important features of the flowfield can be missed, and the accuracy of the solution is degraded. For most problems, the solution and hence the regions where grid points should be clustered are not known beforehand. This indicates the need for solution-adaptive grid-generation procedures where the grid-point distribution will evolve with the solution.

In this paper we investigate the use of a solution-adaptive procedure in a parabolic grid-generation scheme. The steady-state Euler equations are solved in a space marching format for hypersonic flow about re-entry vehicles. The Euler equations are solved in a new split-flux finite-volume formulation. A solution-adaptive grid is generated at the body and in the interior to improve the resolution of high gradient regions. Results from the present formulation are compared with results from a solution-adaptive elliptic grid-generation scheme and to solutions using nonadaptive grids.

Grid Generation Using Parabolic Differencing

The generation of grids by solving a parabolic partial differential equation was first proposed for two-dimensional grids by Nakamura.[3] The method starts by considering the elliptic grid-generation equations. These equations, in the form used by Thomas and Middlecoff,[4] are

$$\alpha[r_{\xi\xi}+\phi r_{\xi}]-2\beta r_{\xi\eta}+\gamma[r_{\eta\eta}+\psi r_{\eta}]=0 \tag{1}$$

where $r=x$ or y and

$$\alpha=x_{\eta}^2+y_{\eta}^2; \qquad \gamma=x_{\xi}^2+y_{\xi}^2; \qquad \beta=x_{\xi}x_{\eta}+y_{\xi}y_{\eta}$$

and (x,y) are Cartesian coordinates and (ξ,η) the computational coordinates. The finite-difference equations for an elliptic grid-generation algorithm are formed by approximating all derivatives by second-order centered differences. Assuming that $\Delta\xi=\Delta\eta=1$, and using one-sided first derivatives to maintain diagonal dominance, the following system of equations is

Presented as Paper 88-0315 at the AIAA 26th Aerospace Sciences Meeting, Reno, NV, Jan. 11-14, 1988; received June 7, 1988; revision received July 10, 1989.

*Graduate Research Assistant, Aerospace Engineering Department. Senior Member AIAA.

†Professor, Aerospace Engineering Department. Member AIAA.

obtained

$$ur_{m+1}^{n} - dr_{m}^{n} + lr_{m-1}^{n} = \beta(r_{m-1}^{n-1} - r_{m-1}^{n+1} - r_{m+1}^{n-1} + r_{m+1}^{n+1})$$

$$-2\gamma[(1+\psi^{+})r_{m}^{n+1} + (1-\psi^{-})r_{m}^{n-1}] \tag{2}$$

where

$$u = 2\alpha(1+\phi^{+}), \qquad d = 2[\alpha(2+|\phi|) + \gamma(2+|\psi|)]$$

$$l = 2\alpha(1-\phi^{-})$$

$$\phi^{-} = \min(0,\phi), \qquad \phi^{+} = \max(0,\phi)$$

$$\psi^{-} = \min(0,\psi), \qquad \psi^{+} = \max(0,\psi)$$

where the index m is in the ξ direction, and n is the index in the η direction (which is shown as a superscript in the usual notation for a marching problem). At this point the difference equations are identical to those used for solving the Poisson grid-generation equations. The set of equations could be solved using point or line relaxation techniques. Parabolic differencing of the elliptic grid equations arises by considering the points at the $n+1$ level to be fixed or specified. The resulting difference equations can be shown to be consistent with finite-difference approximations to a parabolic partial differential equation.

Since the grid is to be marched from an inner boundary outward, the level $n-1$ points are given from initial data or a previous marching step. Thus the fundamental problem for the parabolic scheme is to define the method of specifying the level $n+1$ points.

The essence of the methods of Nakamura[3,5] and Edwards[6] is that the elliptic grid generation equations given in Eq. (1) are differenced on an unequally spaced grid in the computational plane. With this differencing scheme, the level $n+1$ points are taken to be the outer boundary points. The unequally spaced grid is used only to solve the grid-generation equations. The grid generated by the scheme is used to find the solution of the partial differential equation of interest, generally on an equally spaced computational grid.

It should be noted that most parabolic grid-generation schemes[3,5-7] start from the elliptic grid-generation equations with the grid-control functions ϕ and ψ set equal to zero. One exception is the work of Hodge et al.,[8] in which the grid-control functions are analytically specified to achieve a desired type of clustering. In the present work, the grid-control functions are included and based on the solution to provide a solution-adaptive grid.

Method Using Local Reference Grid

A variation of the parabolic grid-generation method of Nakamura was developed by Noack[7] and has been utilized in space marching solutions to the Euler equations.[7,9] In this method the elliptic grid-generation equations are differenced on a uniformly spaced grid in the computational plane. Thus the difference equations for the method are identical to the elliptic difference equations given by Eq. (2). The equations are parabolized by first generating a local reference grid at the level n and $n+1$ lines. This local reference grid is generated by a hyperbolic/algebraic scheme, which will be described later. The elliptic difference equations are parabolized by setting the r_m^{n+1} grid points equal to the corresponding $n+1$ value in the local reference grid. The nonlinear terms α, β, and γ are evaluated from the grid points in the local reference grid.

The local reference grid is regenerated at each marching step based upon the line of data at the $n-1$ level and the outer boundary. If the reference grid is clustered and orthogonal, then the resulting grid generated by the parabolic scheme will be clustered and nearly orthogonal. This can be seen quite easily as the method can be viewed as applying a Laplacian smoothing function to the level n points with the $n-1$ and $n+1$ points fixed. Thus the new level n line of grid points obtained from the tridiagonal system of equations given by Eq. (2) will be in the neighborhood bound by the previous line of points (level $n-1$) and the reference grid points at the $n+1$ level.

The scheme used to generate the local reference grid is a combination of an explicit hyperbolic grid-generation method and simple interpolation. The technique described herein will be very similar to that described in Ref. 7. However, the development will begin from a slightly different viewpoint.

We first consider the following set of equations obtained by specifying the Jacobian and a measure of the orthogonality of the mesh

$$x_\xi y_\eta - x_\eta y_\xi = \mathcal{J} \tag{3}$$

$$x_\eta x_\xi + y_\eta y_\xi = \beta \tag{4}$$

This is the equation set that Steger and Chaussee[2] solved using an implicit scheme for the standard hyperbolic grid-generation scheme. Choosing the η direction to be the marching direction and solving for x_η, y_η we obtain

$$x_\eta = \frac{x_\xi \beta - y_\xi \mathcal{J}}{x_\xi^2 + y_\xi^2}; \qquad y_\eta = \frac{y_\xi \beta + x_\xi \mathcal{J}}{x_\xi^2 + y_\xi^2} \tag{5}$$

The Jacobian and the orthogonality measure can also be written

$$\mathcal{J} = \mathcal{S}_\eta \mathcal{R}_\xi \sin\theta; \qquad \beta = \mathcal{S}_\eta \mathcal{R}_\xi \cos\theta \tag{6}$$

in which

$$\mathcal{S}_\eta = \sqrt{x_\eta^2 + y_\eta^2}; \qquad \mathcal{R}_\xi = \sqrt{x_\xi^2 + y_\xi^2} \tag{7}$$

where $\mathcal{S}$ is the arc length along a $\xi = \text{const}$ coordinate arc, $\mathcal{R}$ the arc length along an $\eta = \text{const}$ coordinate arc, and θ the angle between the two coordinate lines. For an orthogonal grid $\theta = \pi/2$, $\mathcal{J} = \mathcal{S}_\eta \mathcal{R}_\xi$, and $\beta = 0$. Thus Eqs. (5) reduce to the following explicit hyperbolic grid-generation equations

$$x_\eta = \frac{-y_\xi \mathcal{S}_\eta}{\sqrt{x_\xi^2 + y_\xi^2}}; \qquad y_\eta = \frac{x_\xi \mathcal{S}_\eta}{\sqrt{x_\xi^2 + y_\xi^2}} \tag{8}$$

The arc length derivative in the marching direction $\mathcal{S}_\eta$ is now written as

$$\mathcal{S}_\eta = \Delta\bar{s} R \tag{9}$$

where R is the distance between the point $m,n-1$ and the outer boundary point m, NMAX. The $\Delta\bar{s}$ is a parameter used to control the grid spacing in the η direction and is given in terms of a clustering function

$$\Delta\bar{s} = \frac{\bar{s}_n - \bar{s}_{n-1}}{\bar{s}_{\text{NMAX}} - \bar{s}_{n-1}} \tag{10}$$

The function $\bar{s}$ is the desired normalized distribution of arc length along the $\xi = \text{const}$ arc. For a nonadaptive grid scheme, this distribution is specified analytically. The specification for an adaptive grid scheme will be discussed in a subsequent section.

Finally a grid point $(x_{m,n}^0, y_{m,n}^0)$, which is orthogonal to the $n-1$ line of points, is found by differencing Eq. (8) with

$$x_\eta = x_{m,n}^0 - x_{m,n-1}; \qquad x_\xi = x_{m+1,n-1} - x_{m-1,n-1}$$

$$y_\eta = y_{m,n}^0 - y_{m,n-1}; \qquad y_\xi = y_{m+1,n-1} - y_{m-1,n-1}$$

and solving for grid point $x_{m,n}^0, y_{m,n}^0$.

It is not always possible to have a completely orthogonal grid given the inner and outer boundary points. Therefore the reference point is switched or relaxed from the point along the orthogonal line near the body to the interpolated point as the grid is marched toward the shock. This allows a body normal grid to meet exactly the specified outer boundary. A point which is interpolated between the $n-1$ point and the NMAX point is given by

$$x^i_{m,n} = x_{m,n-1} + \Delta\bar{s}(x_{m,\mathrm{NMAX}} - x_{m,n-1}) \tag{11a}$$

$$y^i_{m,n} = y_{m,n-1} + \Delta\bar{s}(y_{m,\mathrm{NMAX}} - y_{m,n-1}) \tag{11b}$$

Thus the local reference grid points are given by

$$x_{m,n} = \epsilon_s x^i_{m,n} + (1-\epsilon_s)x^0_{m,n} \tag{12a}$$

$$y_{m,n} = \epsilon_s y^i_{m,n} + (1-\epsilon_s)y^0_{m,n} \tag{12b}$$

where ϵ_s is a switching function that varies from zero at the body to unity at the shock.

Arc Equidistribution Concept

Since the parabolic grid-generation scheme starts from the elliptic grid-generation equations, we first review an adaptive procedure previously developed for the elliptic equations.[10] The extension to a parabolic scheme will then follow.

The concept of equidistribution is often utilized as the starting point for the development of a solution-adaptive grid-generation procedure. A one-dimensional equidistribution law can be written as

$$\mathcal{S}_\eta w_1 = C \tag{13}$$

Here $\mathcal{S}$ is the arc length in physical space, η the computational coordinate, w_1 a weight function that is taken to depend on the solution of the equations of interest, and C is a constant. If the equidistribution law is differentiated with respect to the computational coordinate η, the result is

$$\mathcal{S}_{\eta\eta} + \tilde{\psi}\mathcal{S}_\eta = 0 \tag{14a}$$

$$\tilde{\psi} = \frac{1}{w_1}\frac{\partial w_1}{\partial \eta} \tag{14b}$$

This second-order partial-differential equation is used in the present effort to provide an adaptive distribution of points along computational coordinate curves such as the body surface. We difference Eq. (14a) with second-order centered differences for the second derivative and first-order differences for the first derivative. We choose a forward first difference if $\tilde{\psi}>0$ and a backward first difference if $\tilde{\psi}<0$. The difference equation yields the following recursion formula for the arc length at the grid point n.

$$\mathcal{S}_j = \sum_{i=1}^{j}\Delta\mathcal{S}_i, \qquad \Delta\mathcal{S}_i = \mathcal{S}_i - \mathcal{S}_{i-1} \tag{15a}$$

$$\Delta\mathcal{S}_i = \Delta\mathcal{S}_{i-1}(1+|\tilde{\psi}_i|) \quad \text{if } \tilde{\psi}_i<0 \tag{15b}$$

$$\Delta\mathcal{S}_i = \Delta\mathcal{S}_{i-1}/(1+|\tilde{\psi}_i|) \quad \text{if } \tilde{\psi}_i>0 \tag{15c}$$

Solution-Adaptive Procedure for the Elliptic Equations

Anderson et al.[10] showed that the grid-control functions in the elliptic grid-generation equations given by Eq. (1) can be related to arc equidistribution. The elliptic grid-generation equations when rewritten with arc length along coordinate curves as dependent variables are,[10] for example, for constant ξ arcs

$$\mathcal{S}_{\eta\eta} + \Psi\mathcal{S}_\eta = 0 \tag{16}$$

where

$$\Psi = \psi + (\nu_\eta - 2\mu_\eta)\cot(\nu-\mu) + \frac{\mu_\xi \mathfrak{N}_\eta}{\mathcal{S}_\eta \sin(\nu-\mu)} \tag{17}$$

Here $\mathcal{S}$ is the arc length along the constant ξ arc, $\mathfrak{N}$ the arc length along the constant η arc, ν the angle between the $\xi=$ const coordinate line and the x axis, and μ the angle between the $\eta=$ const coordinate line and the x axis. For a grid which is orthogonal and without curvature, the grid-control function reduces to $\Psi=\psi$ and can be easily identified with the weight function. Thus with ψ specified in terms of weight functions, for moderate curvature and approximately orthogonal grid lines, the result will be a grid based on arc equidistribution. Similar arguments can be applied to the other coordinate direction. The desired grid-control functions then are

$$\psi = \frac{1}{w_1}\frac{\partial w_1}{\partial \eta} \qquad \phi = \frac{1}{w_2}\frac{\partial w_2}{\partial \xi} \tag{18}$$

Thus for grid-generation schemes based on the Poisson equations, a solution-adaptive procedure is easily implemented by evaluating the grid-control functions based on the weight functions as indicated in Eq. (18). Anderson et al.[10] present results for time marching solutions using this formulation. In the present investigation, we will present results comparing grids generated with this adaptive elliptic scheme and the present adaptive parabolic scheme for use in the space marching solution of the steady Euler equations.

Solution-Adaptive Parabolic Grid Procedure

The parabolic grid-generation procedure of Ref. 7 can be easily adapted to provide solution-adaptive grids. This is accomplished by first parabolizing the Poisson grid equations instead of Laplace's equations. The grid-control functions are related to the weight functions by Eq. (18), just as for the elliptic adaptive procedure. With the local reference grid approach to parabolizing the elliptic grid equations, it is the local reference grid which most directly controls the clustering of the grid. Thus for a solution-adaptive parabolic procedure, the spacing on the local reference grid is based on weight functions.

As discussed in a preceding section, the spacing in the η direction on the local reference grid is controlled by the arc length derivative $\mathcal{S}_\eta$. Solution-adaptive spacing could easily be achieved by solving the equidistribution law given by Eq. (13) for the arc length derivative $\mathcal{S}_\eta = C/w_1$. This is not used because the constant C is unknown. We therefore specify the arc length derivative by Eq. (9) just as for the nonadaptive scheme. The desired arc-length distribution $\bar{s}$ is set equal to the arc-length distribution $\mathcal{S}$ obtained by solving the equidistribution law in finite-difference form. The finite difference representation of Eq. (13) could be used since normalizing the arc length distribution removes the dependence on the constant. In order to be consistent with the elliptic equations, we choose to use the second-order partial differential equation form of the equidistribution law given by Eqs. (14). The finite-difference approximation to this equation is given by Eqs. (15). Thus the desired arc-length distribution $\bar{s}$ is given by Eqs. (15), and the grid-control function ψ given by Eq. (18) is used in place of the function ψ.

The spacing in the ξ direction on the local reference grid is controlled primarily by the spacing along the initial data line, which is the body. Hence it is important that a solution-adaptive procedure be used to determine the placement of grid points on the body. The preceding arc equidistribution method

is used in the present study. The arc-length distribution $\mathfrak{N}$ along the body is given by Eqs. (15) with $\mathfrak{N}$ replacing $\mathfrak{S}$ and ϕ given by Eq. (18) replacing $\hat{\psi}$.

Weight Function

The weight functions used in the present study are a combination of first derivatives and the curvature of the solution variable:

$$w_1 = 1 + c_{11}|f_\eta| + c_{12}\frac{|f_{\eta\eta}|}{(1+f_\eta^2)^{3/2}} \tag{19a}$$

$$w_2 = 1 + c_{21}|f_\xi| + c_{22}\frac{|f_{\xi\xi}|}{(1+f_\xi^2)^{3/2}} \tag{19b}$$

where the solution is adapted to the solution variable f. For the present study, the solution is adapted to the static pressure. The parameters $c_{11}, c_{12}, c_{21}, c_{22}$ are constants used to control the amount of adaption.

Additional Procedures

The adaptive parabolic and elliptic grid schemes described in preceding sections will generate qualitatively pleasing adaptive grids for static test cases. However, difficulties were encountered when the adaptive schemes were used in the solution of the Euler equations where the grid is coupled to the solution. This is because the adaptive grid changes rapidly as shocks develop and propagate into the flowfield. This rapid movement of the grid causes the flowfield solution scheme to become unstable. Several different procedures were implemented to control the movement of the grid and to stabilize the solution scheme.

A smoothing operator is applied at various stages in the generation of the adaptive grid. This smoothing is required for several reasons. The first is that numerical approximations to derivatives are required at several points in the adaptive procedure. Smoothing the derivatives will reduce the high-frequency variations in the derivatives, which can occur because of the numerical approximations. In addition we wish to track general flow features and not transient spikes, which may occur due to errors introduced by the numerical scheme.

A smoothing operator[11] is applied as follows.

For one-dimension, such as at the body:

$$\bar{f} = f + \frac{\omega_s}{4} f_{\xi\xi} \tag{20a}$$

For two-dimensions, such as in the interior:

$$\bar{f} = f + \frac{\omega_s}{8}(f_{\xi\xi} + f_{\eta\eta}) \tag{20b}$$

where f is the function to be smoothed and ω_s is the smoothing parameter. Multiple passes of the smoothing operator are also used. The smoothing is applied to the pressure before the weight functions are calculated from Eqs. (19). The weight functions are then smoothed before the grid-control functions are obtained from Eq. (18). Finally, the grid-control functions are smoothed before use in Eqs. (2) and (15).

The actual rate of change of the grid from one marching location to the next is controlled by under-relaxing the change in the grid-control functions ϕ and ψ. This is required to prevent the grid from moving too rapidly in response to regions with shocks, which develop as the solution is marched downstream. In the present study we use

$$\psi^{l+1} = \psi^l + \omega_r \Delta\psi \tag{21}$$

where ψ^{l+1} is the new value of the grid-control function, ψ^l the value from the previous grid plane, and ω_r a relaxation parameter. A typical value of the relaxation parameter is $\omega_r = 0.05$. The change $\Delta\psi$ from one step to the next is limited as follows

$$\Delta\psi = \min\left(\frac{\omega_r}{2}[\psi_{\max} + \tilde{\psi}^{l+1}], \tilde{\psi}^{l+1} - \psi^l\right) \tag{22}$$

where $\psi_{\max}$ is the maximum value of $\tilde{\psi}^{l+1}$ over the whole grid. The ψ^{l+1} is the grid-control function given by Eq. (18), and ψ^l is the grid-control function from the previous plane of data.

From Eqs. (15) we see that grid-control functions are related to the ratio of two adjacent arc increments. Thus limiting the grid-control function will limit how rapidly the mesh can stretch in each coordinate direction. This is desirable as large variations in the grid point spacing can degrade the solution accuracy. It was also found that setting limits on the magnitude of the grid-control functions increased the stability of the whole solution procedure. The present study limited both ϕ and ψ in magnitude to less than 0.2.

Finally, the adaption-control parameters c_{11}, c_{12}, c_{21}, c_{22} are allowed to vary linearly with the η coordinate. This was added as a result of the observation that the grid tended to cluster to the strongest shocks, which for the cases investigated were at the body. This removed grid points from the regions with weaker shocks such as the outer bow shock. It was found that significantly higher adaption parameters could be used away from the body. Thus the control parameters in the computed results were typically six times greater at the outer boundary than at the body. This significantly improved the adaption to the outer bow shock and maintained a stable scheme.

Flowfield Solution Methodology

The steady, three-dimensional Euler equations representing conservation of mass and momentum with a Cartesian-base coordinate system, after transforming to the computational space (ξ,η,ζ), can be written in the strong conservation law form as

$$\frac{\partial E}{\partial \zeta} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \tag{23}$$

where

$$E = \mathfrak{J}\bar{E} \tag{24a}$$

$$F = \hat{\xi}_z\bar{E} + \hat{\xi}_x\bar{F} + \hat{\xi}_y\bar{G} \tag{24b}$$

$$G = \hat{\eta}_z\bar{E} + \hat{\eta}_x\bar{F} + \hat{\eta}_y\bar{G} \tag{24c}$$

$$\hat{\xi}_x = \mathfrak{J}\xi_x; \quad \hat{\xi}_y = \mathfrak{J}\xi_y; \quad \hat{\xi}_z = \mathfrak{J}\xi_z$$

$$\hat{\eta}_x = \mathfrak{J}\eta_x; \quad \hat{\eta}_y = \mathfrak{J}\eta_y; \quad \hat{\eta}_z = \mathfrak{J}\eta_z \tag{25}$$

where $\mathfrak{J} = x_\xi y_{h\eta} - y_\xi x_\eta$ is the Jacobian of the inverse tranformation. In additon, $\bar{q} = (\rho, \rho u, \rho v, \rho w)$ is the usual vector of conserved variables and the fluxes are $\bar{E}$, $\bar{F}$, $\bar{G}$. The condition that the total enthalpy is a constant is used in place of the energy equation. The assumption of a perfect gas is also made. In addition the velocity in the z direction, w, is assumed to be greater than the local acoustic speed.

It is well known that global conservation might be violated by finite-difference schemes for the strong conservation form of transformed partial differential equations for cases with time-dependent transformations. Thomas and Lombard[12] developed a geometric conservation law (GCL), which must be solved together with the partial differential equations of interest in order to overcome this problem. For an adaptive grid, the transformation will be changing rapidly as the solution progresses. Attempting to solve the finite-difference representation of the partial differential equations with an adaptive

grid without regard for geometric conservation will introduce large errors in the solution. In the present study, we found that the grid will adapt to these errors and further degrade the solution.

Hindman[13] showed that by casting the equations in either the chain rule or the weak conservation form, the solution of a GCL is not required. The weak conservation form of the steady Euler equations was solved in Ref. 9 for cases where the changing geometry induced grid motion. The present research effort quickly pointed out several weaknesses in using the weak conservation form of the governing equations and in solving the GCL to maintain global conservation.

Thomas and Lombard state that the GCL should be differenced with the same scheme as used to solve the flow conservation laws.[12] When the flow is solved with a simple scheme this can be done quite easily. However, it is not at all obvious how to difference the GCL when a split scheme is used to solve the flow equations.

The weak conservation law form of the governing equations will conserve mass, etc., only if the metrics do not vary. Thus solutions of the weak form of the flow conservation laws will not properly capture shocks for an adaptive grid as the metrics will be changing because of the adaption. This was found to result in the shock being located in different positions when the amount of adaption was changed.

Because of these difficulties, the finite-difference scheme for Eq. (23) was abandoned, and a new code was developed based on the finite-volume formulation. Although written as a differential equation, Eq. (23) can be interpreted as representing the flux balance over a control volume of the conserved quantities. As we are solving the steady form of the equations, the unsteady term representing the accumulation of mass, etc., in the volume is not needed.

The metric terms given by Eqs. (25) represent the directed areas of each cell face. The directed areas are calculated as half the vector cross product of the two diagonal vectors joining the four vertex points defining a cell interface.[14] An axis normal grid is assumed so that the solution is marched in planes perpendicular to the z axis. Figure 1 presents a schematic of the grid and control volume along with the fluxes at each face. The vertices of the cell face at each axis normal plane are chosen to be at the centroid of the four neighboring grid points. Thus the centroid of the $\zeta = \text{const}$ cell faces will be approximately at a grid point.

A flux vector splitting technique is used as the basis for an upwind algorithm. Matus[15] also utilized a flux-split method in space marching solutions to steady Euler equations where the fluxes were split according to the sign of the eigenvalues of the steady Euler equations. In the present effort, the fluxes of the steady equations are split according to the signs of eigenvalues of the unsteady Euler equations. This splitting technique was used successfully in the finite-difference solution of the steady Euler equations for hypersonic flow about complex geometries.[9] Splitting the fluxes of the steady-flow equation in this manner is valid as the eigenvalues of the steady equations will be of the same sign as the eigenvalues of the unsteady equations.[9]

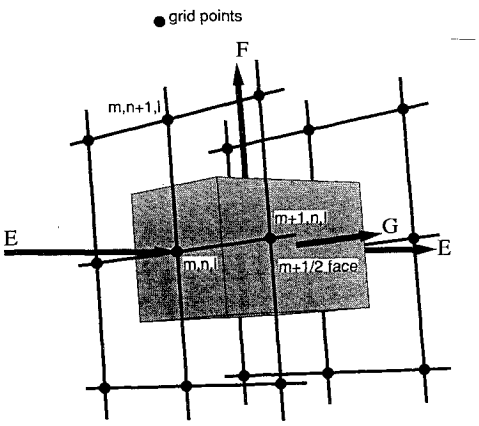

Fig. 1 Schematic of volume, grid, and face fluxes.

Several methods exist for splitting the fluxes at the cell faces. Van Leer[16] developed a nonlinear splitting of the fluxes, which has been extended to two[17] and three[14] dimensions. Steger and Warming[18] developed a technique that is linear in terms of the eigenvalues. Reklis and Thomas[19] present another splitting which is, in essence, a spectral decomposition of the Steger-Warming splitting. The present study utilizes this spectral decomposition to split the flux into three pieces, one for each of the three distinct eigenvalues.

The fluxes at the cell faces are split based on the eigenvalues at the cell faces. The cell-face eigenvalues are obtained by calculating eigenvalues on each side of the face based on the geometric quantities at the cell faces and the conserved variables obtained from the adjacent grid point on that side of the face. The two eigenvalues are then averaged to find the eigenvalues at the cell faces.

The solution is advanced by balancing the fluxes through the faces of the volume. As the flow is assumed to be steady, the time rate change of the mass, momentum, and energy within the volume is zero. Hence the flux balance provides the flux exiting the face at the end of the marching step. This flux is then decoded to provide the conserved and physical variables at the new axial location. The flux balance is given by

$$E^{l+1}_{m,n} = E^{l}_{m,n} - [F^{l+1/2}_{m,n+1/2} - F^{l+1/2}_{m,n-1/2} + G^{l+1/2}_{m+1/2,n} - G^{l+1/2}_{m-1/2,n}] \tag{26}$$

where l is the index in the ζ direction and is shown as a superscript in the usual notation for the marching direction. The subscripts m and n are the indices in the ξ and η directions, respectively.

The flux at the inflow axial face is formed from the conserved variables stored at the centroid of that cell face. As mentioned in the preceding discussion and indicated in Eq. (26), the flux balance will provide the flux at the outflow axial face. The half indices in Eq. (26) indicate that the crossflow faces lie between grid points (see Fig. 1) and the quantities used to form the fluxes on these faces must be extrapolated to the face. The type of extrapolation scheme used determines the character of the difference approximation to the governing partial differential equations.

The algorithm used to approximate the flux balance can be viewed as a finite-volume version of the Warming and Beam[20] upwind version of the MacCormack scheme.[21] Whitfield and Janus[22] previously utilized a similar scheme in the solution of the unsteady Euler equations.

The scheme proceeds in a predictor-corrector fashion as follows. For the predictor step, fluxes on the cross flow faces are

$$F^{l+1/2}_{m,n+1/2} = F(q^{l}_{m,n+1/2}); \qquad G^{l+1/2}_{m+1/2,n} = G(q^{l}_{m+1/2,n}) \tag{27}$$

where the notation indicates that $F^{l+1/2}_{m,n+1/2}$ is evaluated from the flux formula given by Reklis and Thomas[19] with the appropriate cell-face areas and extrapolated conserved variables. The fluxes are formed from the flux pieces associated with the three distinct eigenvalues. Each piece is evaluated from the conserved variables extrapolated in the direction appropriate for each eigenvalue. Thus F_1 is formed from q extrapolated based upon the sign of λ_1. The extrapolation in the η direction for the predictor is

$$q^{l}_{m,n+1/2} = q^{l}_{m,n+1} \qquad \text{if } \lambda \geq 0 \tag{28a}$$

$$q^{l}_{m,n+1/2} = q^{l}_{m,n} \qquad \text{if } \lambda < 0 \tag{28b}$$

with a similar procedure for the extrapolation in the ξ direc-

tion. Once the fluxes at each face are formed, the end-face is found from Eq. (26) and is then decoded to yield the predicted values of the conserved variables $q^{l+1}_{m,n}$ at the new axial location.

For the corrector step, the fluxes on the crossflow faces are

$$F^{l+1/2}_{m,n+1/2}=\frac{1}{2}\left[F(q^{l}_{m,n+1/2})+F(\overline{q^{l+1}_{m,n+1}})\right] \tag{29a}$$

$$G^{l+1/2}_{m+1/2,n}=\frac{1}{2}\left[G(q^{l}_{m+1/2,n})+G(\overline{q^{l+1}_{m+1/2,n}})\right] \tag{29b}$$

The extrapolation in the η direction for the corrector is

$$q^{l}_{m,n+1/2}=q^{l}_{m,n+1}+\theta(q^{l}_{m,n+1}-q^{l}_{m,n+2}) \tag{30a}$$

$$\overline{q^{l+1}_{m,n+1/2}}=\overline{q^{l+1}_{m,n+1}} \qquad \text{If } \lambda<0 \tag{30b}$$

$$q^{l}_{m,n+1/2}=q^{l}_{m,n}+\theta(q^{l}_{m,n}-q^{l}_{m,n-1}) \tag{30c}$$

$$\overline{q^{l+1}_{m,n+1/2}}=\overline{q^{l+1}_{m,n}} \qquad \text{If } \lambda\geq 0 \tag{30d}$$

with a similar procedure for the extrapolation in the ξ direction. Once again, the fluxes at each face are formed, the end face flux is found from Eq. (26) and is then decoded to yield the corrected values of the conserved variables $q^{l+1}_{m,n}$ at the new axial location.

The parameter θ controls the order of the scheme. When $\theta=0$, a first-order scheme is obtained, and $\theta=1$ results in a second-order scheme. Flux limiting is achieved when θ depends upon the solution gradients. These limiters are intended to reduce a second-order scheme to first order in regions of steep gradients. Examples of such limiters can be found in Ref. 17. For the present study, unless otherwise specified, we set $\theta=0.1$.

Boundary Conditions

The location of the outer boundary, corresponding to $\eta=\eta_{\max}$, is specified, and freestream conditions are imposed. Thus any shocks, including the bow shock, will be captured in the present study.

The body-boundary condition procedure extrapolates quantities from the interior and enforces tangency. The pressure is first extrapolated to the body with the following

$$p^{l+1}_{m,1}=p^{l+1}_{m,2}+\hat{\theta}(p^{l+1}_{m,2}-p^{l+1}_{m,3}) \tag{31}$$

where $\hat{\theta}$ is a parameter used to control the order of the extrapolation. When $\hat{\theta}=0$, the zeroth order extrapolation results, whereas $\hat{\theta}=1$ gives first-order extrapolation. Difficulties were encountered with $\hat{\theta}=1$ so that the present study used $\hat{\theta}=0.1$. The density at the body is evaluated from the extrapolated pressure and the entropy at the first grid point off the body ($n=2$). The velocity components at the body are then obtained by taking the velocity vector at the first grid point off the body, subtracting the component normal to the body, and stretching the magnitude so that the total enthalpy is equal to the freestream value.

Results

The present solution-adaptive procedure is tested for hypersonic flow about a two-dimensional wedge and a re-entry vehicle with drag flaps. The flowfields about both vehicles contain embedded shocks, which are to be captured by the solution procedure. In both cases, freestream conditions are used as the initial plane of data at the apex of the vehicle.

The first test case is a 5-deg, two-dimensional wedge. The freestream conditions are Mach 4 and zero angle of attack. First-order solutions for parabolic adaptive and uniform grids are compared to results for a second-order solution on a uniform grid. The weight function [given by Eqs. (19)] for the adapted grid used parameter values of $c_{11}=0.1$, $c_{12}=10$ at the body and 60 times these values at the outer boundary. The uniform grid is shown in Fig. 2 with the grid generated with the present solution-adaptive parabolic scheme shown in Fig. 3. For clarity the initial portions of the grid and contour have been omitted. The solution is marched in the z direction generating a new grid at each step with the z axis aligned with the wedge surface. As indicated by the arrow in Figs. 2 and 3, the freestream velocity is inclined at 5 deg to the wedge surface. These grids are generated as (x,y) planes at each z location, even though only a line is required for the two-dimensional case. The adaptive grid tracks the shock well and shows significant clustering in the region of the shock. Pressure contours for the first-order uniform and adaptive grids are also shown in Figs. 2 and 3. A comparison of the pressure profile at the last marching step is shown in Fig. 4. Comparing the first-order solutions with and without the adaptive grid shows that the adaptive grid significantly reduces the width of the shock. The captured shock in the first-order adaptive solutions is as sharp as the second-order solution on a uniform grid. The oscillations resulting from the dispersion of the second-order scheme cause the shock to extend over a wider region in comparison with the monotone, first-order adaptive results. These results show that by using a solution-adaptive grid, first-order schemes can produce monotone solutions with the resolution of second-order solutions.

The present solution-adaptive procedure bases the weight functions and grid-control function upon differences on the computational domain. A similar formulation can be derived based upon derivatives in the physical plane. We use the present approach because the solution and hence its derivatives in the computational plane will always be bounded. This is not true for the latter approach. Even though the change between two points is bounded, the derivative in the physical plane will increase unboundedly as the physical spacing is decreased. Some method is required to prevent the grid from collapsing to zero spacing. A disadvantage of the present scheme is that the adaptive grid expands with the physical domain. This is readily apparent in Fig. 2.

The final test case is Mach 10 flow about a 6-deg, sharp-cone re-entry vehicle, that has two drag flaps deployed at a deflection of 15 deg from the vehicle centerline. The flow is assumed to be symmetric about the pitch plane. The geometry is shown in Fig. 5. Radii have been used to round the sharp corners and fillet regions of the flap. Figures 6-8 present grids generated using the present method for parabolic adaptive, elliptic-adaptive, and parabolic without adaption along with the pressure contours for the solution on the grids. The solutions were calculated on the half plane with the assumption of

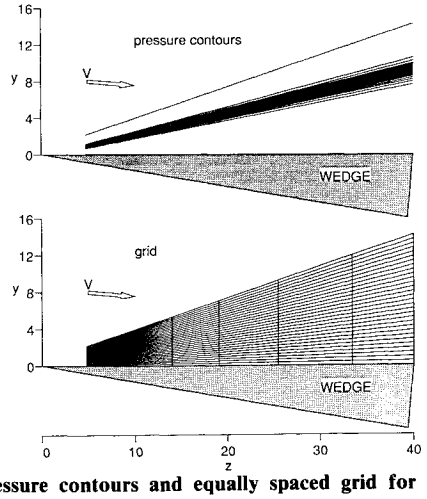

Fig. 2 Pressure contours and equally spaced grid for two-dimensional wedge.

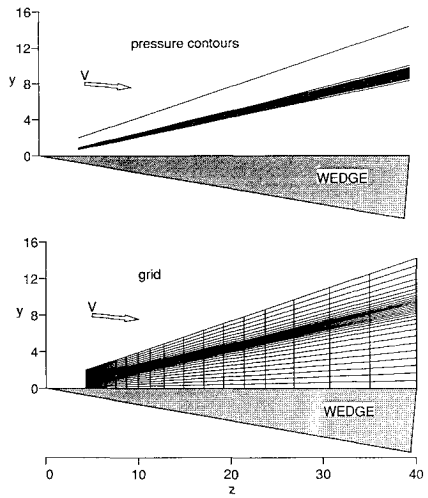

Fig. 3 Pressure contours and adaptive parabolic grid for two-dimensional wedge.

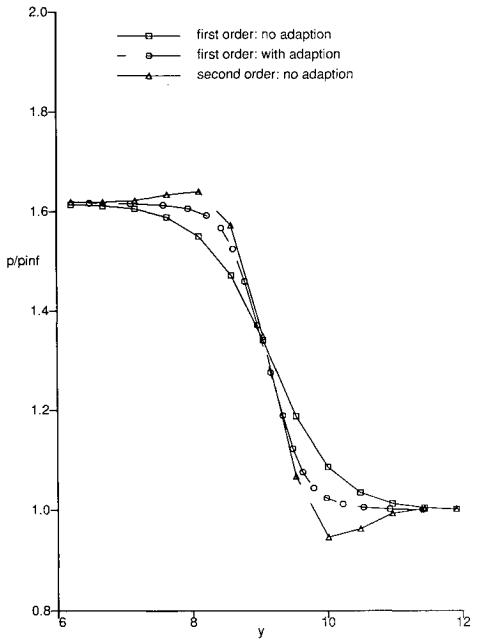

Fig. 4 Comparison of pressure profile for two-dimensional wedge.

symmetry about the pitch plane. The solutions also are symmetric about the yaw plane. The figures also show a magnified view of the grid and contours in the region of the flap. The adaptive-parabolic whereas elliptic schemes cluster the grid significantly in the regions of the captured shocks. The parabolic grids are body normal, whereas the elliptic grid has pulled away from the concave corner. The grid generated with the elliptic scheme does provide somewhat tighter adaption to the shocks except in the concave region. The pressure profiles at the flap centerline are presented in Fig. 9 for solutions using the adaptive parabolic, adaptive elliptic, and nonadaptive parabolic grid schemes. The improved prediction of the shocks with the adaptive grid schemes is readily apparent. The inner flap shock is significantly stronger than the outer bow shock, and the clustering is predominantly in the flap-shock region. The smearing of the outer bow shock is evident indicating the need for tighter adaption to this weaker shock. The variation of the adaption parameters radially improves the adaption to the bow shock but is still insufficient. The sharpness of the shock for the adaptive grids demonstrates the improved results in comparison with the nonadaptive grid. The parabolic adaptive grid also improves the resolution of the shock in the fillet region over the elliptic adaptive grid as shown by the pressure contours.

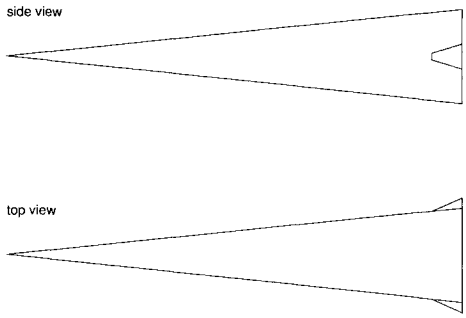

Fig. 5 Re-entry vehicle geometry.

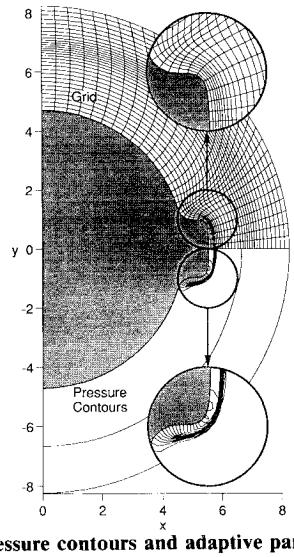

Fig. 6 Pressure contours and adaptive parabolic grid.

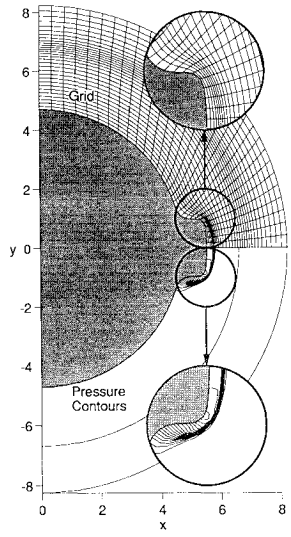

Fig. 7 Pressure contours and adaptive elliptic grid.

The results presented in this paper were computed on an Alliant FX-8 with four processors. This is a vector-parallel machine whose compiler automatically recognizes vector and concurrent loops. The present code contains no machine-specific code and was written so that the Alliant compiler would take maximum advantage of the machine architecture. For comparison purposes, the present code executes approximately 4.5 times faster on a single processor Cray XMP. The approximate execution time on the Alliant for a solution with the parabolic scheme without adaption was 0.51 s/marching step for a 61×31 grid. The time for the parabolic scheme with adaption is 0.66 s/step. Approximately one-third of the execution time is used in generating the parabolic grid. For the

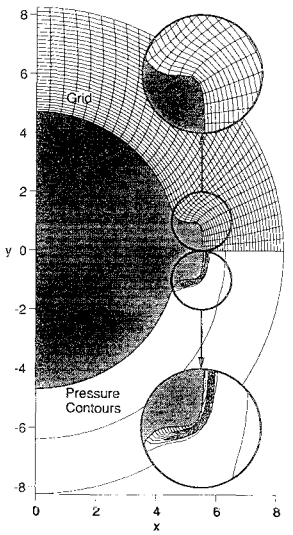

Fig. 8 Pressure contours and parabolic grid without adaption.

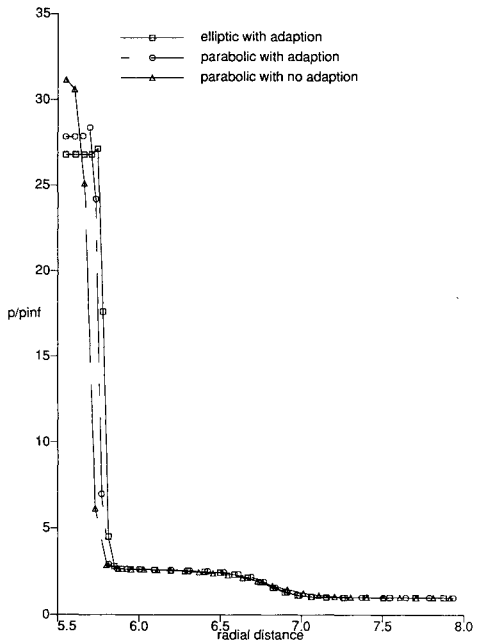

Fig. 9 Comparison of pressure profile at flap centerline.

elliptic grid solutions, the execution time was approximately 22 s/marching step. The elliptic equations are solved with point relaxation for a fixed relaxation parameter of 1.7.

Conclusions

A previously developed parabolic grid-generation scheme has been extended to generate solution-adaptive grids. A new upwind finite-volume method has been described for the space marching solution of the steady Euler equations. The adaptive parabolic grid scheme is used with this finite-volume code to predict the hypersonic flow about two geometries. An adaptive elliptic grid scheme developed previously and applied to unsteady flow solutions is also used in the present steady-flow simulation. The grid is adapted to the captured embedded shocks.

The present solution-adaptive parabolic formulation produces grids which track the flow features of interest as well as the elliptic grid procedure. The parabolic procedure is significantly faster than the elliptic scheme producing a grid in an order of magnitude of less computational time. It was found that use of an adaptive grid provides the potential to capture shocks with a first-order scheme as sharply as a second-order scheme on a uniform grid.

Acknowledgment

This research was partially sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-85-0195.

References

[1]Thompson, J., "Elliptic Grid Generation," *Numerical Grid Generation*, edited by J. F. Thompson, Elsevier, New York, 1982, pp. 79–105.

[2]Steger, J. L. and Chaussee, D. S., "Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 1, No. 4, Dec. 1980.

[3]Nakamura, S., "Marching Grid Generation Using Parabolic Partial Differential Equations," *Numerical Grid Generation,* edited by J. F. Thompson, Elsevier, New York, 1982, pp. 775–786.

[4]Thomas, P. D. and Middlecoff, J. F., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," *AIAA Journal,* Vol. 18, No. 6, 1980, pp. 652–656.

[5]Nakamura, S., "Noniterative Grid Generation Using Parabolic Partial Differential Equations for Fuselage-Wing Flow Calculations," *Lecture Notes in Physics*, Vol. 170, Springer-Verlag, New York, 1982, pp. 775–786.

[6]Edwards, T. A., "Noniterative Three-Dimensional Grid Generation Using Parabolic Partial Differential Equations," AIAA Paper 85-0485, 1985.

[7]Noack, R. W., "Inviscid Flow Field Analysis of Maneuvering Hypersonic Vehicle Using the SCM Formulation and Parabolic Grid Generation," AIAA Paper 85-1682, 1985.

[8]Hodge, J. K., Leone, S. A., and McCarty, R. L., "Non-Iterative Parabolic Grid Generation for Parabolized Equations," *AIAA Journal*, Vol. 25, No. 4, 1987, pp. 542–549; also AIAA paper 85-1528, 1985.

[9]Noack, R. W., "Upwind Inviscid Computations of Complex Geometries in Hypersonic Flow," AIAA Paper 87-0281, Jan. 1987.

[10]Anderson, D. A., and Steinbrenner, J., "Generating Adaptive Grids with a Conventional Grid Scheme," AIAA Paper 86-0427, Jan. 1986.

[11]Holcomb, J. E. and Hindman, R. G., "Development of a Dynamically Adaptive Grid Method for Multidimensional Problems," AIAA Paper 84-1668, June 1984.

[12]Thomas, P. D. and Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[13]Hindman, R. G., "Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors," *AIAA Journal,* Vol. 20, No. 10, 1978, pp. 1359–1367.

[14]Thomas, J. L., Van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA Paper 85-1680, July 1985.

[15]Matus, R. J., "Numerical Solution of the Steady Gasdynamic Equations," Ph.D. Dissertation, Dept. of Aerospace Engineering, University of Texas at Arlington, Aug. 1986.

[16]Van Leer, B., "Flux Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, Springer-Verlag, New York, 1982, pp. 507–512.

[17]Anderson, A. K., Thomas, J. L., and Van Leer, B., "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal,* Vol. 24, No. 9, 1986, pp. 1453–1460.

[18]Steger, J. L. and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite Difference Methods," *Journal of Computational Physics,* Vol. 40, April 1981, pp. 263–293.

[19]Reklis, R. P. and Thomas, P. D., "Shock Capturing Algorithm for the Navier-Stokes Equations," *AIAA Journal*, Vol. 20, No. 9, 1982, pp. 1212–1218.

[20]Warming, R. F. and Beam, R. M., "Upwind Second-Order Difference Schemes and Applications in Aerodynamic Flows," *AIAA Journal*, Vol. 14, No. 9, 1976, pp. 1241–1249.

[21]MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, 1969.

[22]Whitfield, D. L. and Janus, J. M., "Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting," AIAA Paper 84-1552, June 1984.