Conservation Errors and Convergence Characteristics of Iterative Space-Marching Algorithms

D. S. Thompson*
University of Texas at Arlington, Arlington, Texas
and
R. J. Matus†
General Dynamics/Fort Worth Division, Fort Worth, Texas

The new iterative space-marching algorithms for numerical solution of the parabolized Navier-Stokes equations are critically examined. It is shown by analysis and numerical examples that employing parabolized flux difference splitting in the marching direction can lead to serious conservation errors even though a finite-volume formulation is employed. Schemes employing parabolized flux vector splitting in the marching direction are shown to be conservative. The structure of the iterative algorithm is examined to determine what factors play important roles in the convergence of the local iteration.

Introduction

UNTIL recently, most implicit parabolized Navier-Stokes (PNS) codes have employed essentially the same numerical technology.[1,2] The steady form of the thin-layer Navier-Stokes (TLNS) equations was solved using a space-marching scheme that employed central differences and, therefore, required the inclusion of user-specified artificial dissipation. A backward difference approximation was utilized for the derivative of a modified streamwise convective flux vector. The modification of the streamwise convective flux vector was required to eliminate the so-called departure solutions. The resulting algebraic system at each marching station was solved using a noniterative, approximate factorization algorithm. The key point is that in these algorithms only one sequence of calculations was performed at each marching station. This approach was first updated by Lawrence et al.[3] who developed an implicit algorithm employing Roe's scheme[4] in a finite-volume format. It should be noted that in this approach the basic philosophy was essentially the same as the methods just mentioned except that upwinding was employed to automatically include the artificial dissipation.

A different philosophy has been proposed by Newsome et al.,[5] Ota et al.,[6] and Chang and Merkle.[7] The common thread in each of these algorithms is that the unsteady form of the governing equations is solved at each marching station using an iterative procedure. It is well known (e.g., see Ref. 7) that for some cases significant mass conservation errors can occur when conventional PNS algorithms are employed. These results are due, in part, to linearization and factorization errors inherent in the implicit methods. The new PNS schemes have the advantage that these errors can be reduced to arbitrary levels by iteration. An additional advantage is that the infrastructure exists for utilizing the same algorithm as a multiple-pass upwind relaxation scheme for problems where the streamwise elliptic influence is significant. An obvious disadvantage is that multiple iterations are required at each marching station.

The objective of this paper is to address two numerical issues related to these algorithms: 1) conservation error and 2) convergence rate of the local iteration. The basic equations are presented along with the numerical algorithm. An analysis is performed to determine which methods of treatment for the streamwise convective flux vector are stable as well as conservative. Numerical calculations are used to verify the results of the analysis. Finally, the question of convergence of the local iteration is addressed with analysis and numerical examples. The overall objective is to determine whether the increased computational cost of these algorithms is offset by improvements in reliability and accuracy. The analysis and results presented here are for a two-dimensional algorithm. The trends shown here are expected to be observed for an analogous three-dimensional algorithm also.

Governing Equations

The governing equations are the so-called thin-layer Navier-Stokes equations, which are mathematical statements of the conservation of mass, linear momenta, and energy. The differential form of the two-dimensional TLNS equations in Cartesian (x,y) space is given by

$$\frac{\partial U}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=\frac{\partial F_v'}{\partial y} \tag{1}$$

where

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ e_t\end{bmatrix},\quad E=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (e_t+p)u\end{bmatrix},\quad F=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (e_t+p)v\end{bmatrix}$$

$$F_v'=\begin{bmatrix}0\\ \tau_{xy}'\\ \tau_{yy}'\\ u\tau_{xy}'+v\tau_{yy}'-q_y\end{bmatrix}$$

and

$$e_t=\rho[e+\tfrac{1}{2}(u^2+v^2)],\quad \tau_{xy}'=\mu\frac{\partial u}{\partial y}$$

$$\tau_{yy}'=\frac{4}{3}\mu\frac{\partial v}{\partial y},\quad q_y=-\kappa\frac{\partial T}{\partial y}$$

Presented as Paper 89-1935 at the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY, June 13-15, 1989; received Aug. 30, 1989; revision received Jan. 16, 1990; accepted for publication Jan. 23, 1990.

*Assistant Professor, Department of Aerospace Engineering. Member AIAA.
†Engineering Specialist, Computational Fluid Dynamics Group. Member AIAA.

In the preceding equations, ρ is the density, u and v the velocity components in the x and y directions, respectively, p the thermodynamic pressure, T the temperature, μ the coefficient of viscosity, and κ the thermal conductivity. The perfect gas equation of state and Sutherland's law for the viscosity are used to close the system. All streamwise diffusion terms have been neglected in Eq. (1). Although the formulation of the governing equations actually implemented is the generalized coordinate form, all discussion is referenced to the Cartesian form for simplicity.

Discussion of Method

It is now appropriate to describe the numerical method employed to solve the PNS equations. Detailed discussion will be presented only for those aspects of the algorithm that are considered crucial to the thrust of the work.

Finite-Volume Formulation

The governing equations are represented as integral conservation laws and converted to a semidiscrete approximation using a finite-volume approach. The finite-volume formulation was employed because the solution is guaranteed to be conservative in the telescoping sense, provided consistent flux definitions are employed. Integrating Eq. (1) over the cell designated i, j with dimensions $\Delta x \times \Delta y \times 1$ and

$$\frac{\partial}{\partial t}(U_{i,j})(\Delta x \Delta y) + (E_{i+1/2,j} - E_{i-1/2,j})\Delta y + (F_{i,j+1/2} - F_{i,j-1/2}) \times \Delta x - (F'_{v_{i,j+1/2}} - F'_{v_{i,j+1/2}})\Delta x = 0 \tag{2}$$

The semidiscrete approximation itself, i.e., Eq. (2), does not define the numerical algorithm. It is the specification of the numerical flux aproximations, $E_{i+1/2,j}$, $E_{i-1/2,j}$, etc., and the treatment of the temporal derivative that actually define the scheme.

Numerical Flux Approximation

Roe's scheme[4] was selected as the method employed to numerically specify the transverse inviscid fluxes based on the results of van Leer et al.[8] The evaluation of the streamwise fluxes requires more consideration. As will be discussed later, the conservation properties of the scheme are dependent on the method used to define the streamwise fluxes. At this point, it is sufficient to note that either a parabolized form of Roe's scheme or a parabolized flux vector split (FVS) scheme is employed to evaluate the streamwise fluxes.

Roe's scheme belongs to a class of techniques called approximate Riemann solvers. In general, a Riemann solver treats the determination of the flux at a cell face as a locally one-dimensional Riemann problem. In Roe's scheme, the solution of a linearized Riemann problem is used to determine the flux at the cell face. Using this approach, the first-order flux on the cell face at $i,j + 1/2$ is defined to be

$$F_{i,j+1/2} = \tfrac{1}{2}(F_{i,j} + F_{i,j+1}) - \tfrac{1}{2}(\delta F^{+}_{i,j+1/2} - \delta F^{-}_{i,j+1/2}) \tag{3}$$

Particular to Roe's scheme is the specification of the flux increments $\delta F^{-}_{i,j+1/2}$ and $\delta F^{+}_{i,j+1/2}$. First, the flux difference is expressed as

$$\delta F_{i,j+1/2} = \hat{B}_{i,j+1/2}(U_{i,j+1} - U_{i,j}) \tag{4}$$

where

$$\hat{B}_{i,j+1/2} = \hat{T}_{i,j+1/2}\hat{\Lambda}_{i,j+1/2}\hat{T}^{-1}_{i,j+1/2} \tag{5}$$

The elements of the Jacobian matrix $\hat{B}$ have been computed using the averaging suggested by Roe; $\hat{\Lambda}$ is the diagonal matrix of the eigenvalues of $\hat{B}$, and $\hat{T}^{-1}$ is the matrix with rows that are the left eigenvectors of $\hat{B}$. This decomposition exists provided there is a complete set of linearly independent eigenvectors for $\hat{B}$. The $\hat{\Lambda}$ matrix can be split according to the signs of the eigenvalues to obtain

$$\hat{\Lambda} = \hat{\Lambda}^{+} + \hat{\Lambda}^{-} \tag{6}$$

where the elements of $\hat{\Lambda}^{+}$ and $\hat{\Lambda}^{-}$ are given by

$$\hat{\Lambda}^{\pm} = \tfrac{1}{2}(\hat{\Lambda} \pm |\hat{\Lambda}|) \tag{7}$$

The flux increments can now be defined as

$$\delta F^{\pm}_{i,j+1/2} = \hat{B}^{\pm}_{i,j+1/2}(U_{i,j+1} - U_{i,j}) \tag{8}$$

where

$$\hat{B}^{\pm}_{i,j+1/2} = \hat{T}_{i,j+1/2}\hat{\Lambda}^{\pm}_{i,j+1/2}\hat{T}^{-1}_{i,j+1/2} \tag{9}$$

One difficulty associated with Roe's scheme is the fact that the solution of the linearized Riemann problem does not satisfy an entropy condition. That is, solutions obtained using this approach are subject to nonphysical expansion "shocks." No entropy fix is employed here since the viscosity present in the PNS equations should provide sufficient dissipation to break down the nonphysical expansion process.

Treatment of Viscous Terms

The partial derivatives appearing in the viscous terms were evaluated using standard differencing techniques. The net effect is roughly equivalent to employing central differences.

Higher-Order Spatial Accuracy

Higher-order spatial accuracy is obtained using the method suggested by Chakravarthy and Osher.[9] Flux increments from adjacent cells are included in the definition of the flux at the cell face, i.e., Eq. (3), thereby improving the accuracy of the approximation. For example, the numerical flux approximation at $i,j + 1/2$ is given by

$$F_{i,j+1/2} = F^{\text{Eq. (3)}}_{i,j+1/2} - [(1-\phi)/4]\delta\tilde{F}^{-}_{i,j+3/2} - [(1+\phi)/4]\delta\tilde{\tilde{F}}^{-}_{i,j+1/2} + [(1+\phi)/4]\delta\tilde{F}^{+}_{i,j+1/2} + [(1-\phi)/4]\delta\tilde{\tilde{F}}^{+}_{i,j-1/2} \tag{10}$$

where the single tilde and the double tilde indicate the quantities that have been limited as discussed below, ϕ is a parameter that allows specification of schemes of different spatial accuracy, and the flux increments are defined as in Eq. (8). It should be noted that this approach is not unique. Several other methods have been given in the literature, i.e., Refs. 6 and 10. It is not known a priori which of these methods would be the most accurate or robust. An additional point that should be made is that, in the generalized coordinate form of the equations, the metrics used in the calculation of the flux at $i,j + 1/2$ should be the metrics for that cell face. Here, the metrics for each cell face are used in the computation of the associated flux increment.

As just mentioned, the flux differences from adjacent cells are limited in an attempt to control the oscillations that are inherent in results predicted using schemes of accuracy higher than first order. The approach used is that suggested by Chakravarthy and Osher.[9] They choose to limit the flux increment due to each wave family. In the notation used above, this flux increment vector can be written as

$$\sigma^{\pm}_{i,j+1/2} = \hat{\Lambda}^{\pm}_{i,j+1/2}\hat{T}^{-1}_{i,j+1/2}(U_{i,j+1} - U_{i,j}) \tag{11}$$

so that

$$\delta F^{\pm}_{i,j+1/2} = \hat{T}_{i,j+1/2}\sigma^{\pm}_{i,j+1/2} \tag{12}$$

The limited flux increments indicated in Eq. (10) are defined as

$$\delta\tilde{F}^{\pm}_{i,j+1/2} = \hat{T}_{i,j+1/2}\tilde{\sigma}^{\pm}_{i,j+1/2} \tag{13a}$$

$$\delta\tilde{\tilde{F}}^{\pm}_{i,j+1/2} = \hat{T}_{i,j+1/2}\tilde{\tilde{\sigma}}^{\pm}_{i,j+1/2} \tag{13b}$$

where

$$\tilde{\sigma}^{\pm}_{i,j+1/2} = \text{minmod}(\sigma^{\pm}_{i,j+1/2}, \beta\sigma^{\pm}_{i,j-1/2}) \tag{14a}$$

$$\tilde{\tilde{\sigma}}^{\pm}_{i,j+1/2} = \text{minmod}(\sigma^{\pm}_{i,j+1/2}, \beta\sigma^{\pm}_{i,j+3/2}) \tag{14b}$$

The limiter modifies the flux increments to ensure a monotone interpolation. The compression parameter β is included in the limiter to include the effects of curvature in the interpolating polynomial. The value of β is restricted so that the inequality

$$1 \le \beta \le (3-\phi)/(1-\phi) \tag{15}$$

is satisfied yielding a scheme that will provide nonoscillatory results in the steady state.

Numerical Solution Algorithm

The solution is obtained at each marching station via a sequence of calculations, each of which is denoted as a local iteration. The algorithm can be written in delta form as

$$\begin{aligned}(\Delta x\Delta y/\Delta t)\Delta Q^l_{i,j} &+ (\Delta E^l_{i+1/2,j} - \Delta E^l_{i-1/2,j})\Delta y \\ &+ (\Delta F^l_{i,j+1/2} - \Delta F^l_{i,j-1/2})\Delta x - (\Delta F'^l_{v_{i,j+1/2}} - \Delta F'^l_{v_{i,j-1/2}})\Delta x \\ &= \omega[(E^l_{i+1/2,j} - E^l_{i-1/2,j})\Delta y + (F^l_{i,j+1/2} - F^l_{i,j-1/2})\Delta x \\ &- (F'^l_{v_{i,j+1/2}} - F'^l_{v_{i,j-1/2}})\Delta x] \end{aligned} \tag{16}$$

where ω is a relaxation factor, $Q^l_{i,j}$ is the current estimate for the solution $U_{i,j}$, and $\Delta Q^l_{i,j} = Q^{l+1}_{i,j} - Q^l_{i,j}$, etc. The terms appearing on the right side of Eq. (16) may be the higher-order spatial approximations, i.e., Eq. (10), and are evaluated using the most recent values of Q. Appearing on the left side of Eq. (16) is a first-order spatial approximation to the steady-state discrete approximation that appears on the right side and the unsteady term. At this point, it is sufficient to say that the left side must be linearized if an efficient solution algorithm is to be developed. The precise form of the linearization is left unspecified until further consideration. Equation (16) is solved by providing an initial guess $Q^l_{i,j}$ and iterating at the given marching station until the L_2 norm of the local residual divided by the local area is less than a specified tolerance. The initial guess for the solution at the current marching station is provided by the converged solution at the previous marching station. The construction of the algorithm illustrates the primary advantage of the iterative approach. Iteration gives the ability to obtain the desired level of accuracy since the discrete *nonlinear* conservation law is identically satisfied when the right side, i.e., the residual, equals zero. In conventional algorithms, a *linearized* version of the discrete conservation law is satisfied.

An additional consideration is the implementation of the boundary conditions. The upstream boundary is correctly viewed as the initial data line, and the solution at this boundary must be specified in the single-pass philosophy. Similarly, no downstream boundary condition is allowed because of the marching nature of the solution algorithm. The wall boundary condition is partially given by the no-slip condition and either an isothermal or adiabatic wall assumption. The final required condition is the specification of the wall pressure. The pressure is the only term that appears in the inviscid flux normal to the wall since the flow tangency condition must be satisfied. One method for accomplishing this is to set up an approximate Riemann problem involving the point above the wall j and a ghost point below the wall $j-1$ with the ghost point values defined to enforce the no-slip condition at the wall. The resulting flux is formally of first-order accuracy. An additional ghost point at $j-2$ could easily be included for higher-order formal accuracy. However, since the procedure is somewhat contrived, it seems unproductive to pursue this approach. The main advantage of this approach is that it lends itself readily to implicit treatment of the boundaries. The viscous boundary conditions are imposed by satisfying the no-slip condition as well as the isothermal/adiabatic wall boundary condition. These conditions are introduced through evaluation of F'_v at the wall.

A final point to be made is that the solution is usually underrelaxed. As stated by Thomas et al.,[11] algorithms with first-order implicit differencing and higher-order differencing on the right side are only conditionally stable. Typical values for the relaxation factor are presented in the Results section.

Streamwise Conservation Errors

One of the difficulties associated with developing the algorithm is that a finite-volume discretization is not conservative in the telescoping sense unless the parabolized flux is consistently defined. This occurs naturally in the transverse direction. The discussion below will focus on the treatment of the interface fluxes $E_{i+1/2,j}$ and $E_{i-1/2,j}$. In this development, the j subscript is omitted for simplicity.

Parabolized Flux Vector Splitting

In the approaches of Newsome et al.[5] and Chang and Merkle,[7] the FVS suggested by Vigneron et al.[1] is used to define the interface fluxes. Following Vigneron et al.,[1] the flux is split into two parts:

$$E_{i+1/2} = E^d(U^L_{i+1/2}) + E^u(U^R_{i+1/2}) \tag{17}$$

one part associated with convection in the streamwise direction $E^d(U^L_{i+1/2})$ and one part associated with convection in the upstream direction $E^u(U^R_{i+1/2})$. The U^L is the vector of dependent variables obtained by extrapolation to the $i+1/2$ interface from the left and U^R is obtained by extrapolation from the right. The only requirements placed on this splitting are that

$$E(U) = E^d(U) + E^u(U) \tag{18}$$

and the eigenvalues of $\partial E^d/\partial U$ are >0 in subsonic regions and the eigenvalues of $\partial E^u/\partial U$ are <0 in subsonic regions. Following Eq. (18), the split fluxes at $i+1/2$ are given by

$$E^d(U) = T\Lambda^d T^{-1}U, \qquad E^u(U) = T\Lambda^u T^{-1}U \tag{19}$$

where T^{-1} is the matrix with rows that are the left eigenvectors of the Jacobian $A = \partial E/\partial U$, and Λ^d and Λ^u are the diagonal matrices of the eigenvalues used to define the splitting. It should be noted that any flux vector splitting scheme that satisfies the constraints just given can be employed. For example, the Steger-Warming[12] splitting can be used by setting $\Lambda^d = \Lambda^+$, the diagonal matrix of positive eigenvalues, and $\Lambda^u = \Lambda^-$, the diagonal matrix of negative eigenvalues. The eigenvalue splitting employed to obtain the Vigneron split flux can be found in Ref. 7.

For the space-marching algorithm to be well posed, the effects of the upstream propagating waves must be suppressed; therefore, the parabolization is achieved when the term associated with upstream propagation is neglected. It is clear that $E^u(U^R_{i+1/2})$ must be neglected to obtain a stable, well-posed space-marching algorithm. The scheme is obviously conservative in the telescoping sense since consistent flux evaluations are used for each cell face. That is, the flux at the $i+1/2$ face is the same whether the $i+1/2$ face is a left or

right face. This is a direct result of neglecting the contribution to the flux of the left running wave, i.e., neglecting $E^u(U^R_{i+1/2})$ in Eq. (17). In fact, any parabolized FVS method is conservative in the telescoping sense since consistent flux evaluations are used for each cell face. However, this does not address what quantities are conserved. The quantities that are conserved are those that constitute $E^d(U^L_{i+1/2})$. In this respect, it would seem that the Vigneron splitting is more appropriate than either the Steger-Warming[12] or van Leer[13] splittings since it entails the least modification of the flux vector, i.e., only the second term.

Parabolized Flux Difference Splitting

Ota et al.[6] employ Roe's scheme[4] (a form of flux difference splitting—FDS) in the streamwise direction. In this case, a problem arises because the computation of the inviscid flux at $i + 1/2$ for the ith cell requires the specification of the solution at $i + 1$, an undetermined quantity. The solution at $i + 1$ must be specified so that the resulting flux does not allow any upstream influence. The problem is best illustrated by considering the definition of the flux at $i + 1/2$. It can be shown that Roe's scheme applied at the $i + 1/2$ interface yields

$$E_{i+1/2} = E_i + \hat{A}^-_{i+1/2}(U_{i+1} - U_i) \tag{20}$$

where $\hat{A}^-_{i+1/2}$ is the Roe-averaged matrix associated with the negative eigenvalues of $\partial E/\partial U$. A corresponding expression can be obtained for the $i - 1/2$ interface by decrementing the index by 1. It is clear that the value of U_{i+1} is required in regions where the velocity component in the x direction is subsonic, i.e., $\hat{A}^-_{i+1/2}$ is not identically zero. In supersonic regions, the value of U_{i+1} is not required. It now remains to specify the value of U_{i+1} in subsonic regions. The remainder of this discussion focuses on the implications of the selection on the well posedness and the conservation properties of the scheme.

The development that follows can be facilitated more easily by considering the following divided difference

$$(1/\Delta x)(E_{i+1/2} - E_{i-1/2}) \tag{21}$$

which occurs in Eq. (2) after division by the area $\Delta x \Delta y$. Substituting Eq. (20) for $E_{i+1/2}$ and a similar expression for $E_{i-1/2}$ in terms of E_i, U_i, and U_{i-1} into Eq. (21) yields

$$(1/\Delta x)(E_{i+1/2} - E_{i-1/2}) = (1/\Delta x)(\hat{A}^-_{i+1/2}\Delta U_i + \hat{A}^+_{i-1/2}\nabla U_i) \tag{22}$$

The similarity between Eq. (22) and the split coefficient matrix (SCM) scheme of Chakravarthy et al.[14] is apparent. It should be noted, however, that Eq. (22) is spatially conservative because of the manner in which the $\hat{A}^-_{i+1/2}$ and $\hat{A}^+_{i-1/2}$ matrices are evaluated while the SCM scheme is nonconservative. Casting the equation in this form also makes it apparent that the space marching scheme is well posed only if the effects of the negative eigenvalues, i.e., those associated with $\hat{A}^-_{i+1/2}$, are suppressed. This is the only method by which a departure solution can be eliminated and a consistent differencing scheme obtained.

Assume that the provisional value of U^*_{i+1} will be obtained through a linear combination of U_i and U_{i-1}, values that are known or being computed at station i, given by

$$U^*_{i+1} = aU_i + (1 - a)U_{i-1} \tag{23}$$

Using this expression, it follows directly that

$$\Delta U^*_i = U^*_{i+1} - U_i = (a - 1)\nabla U_i \tag{24}$$

so that Eq. (22) can be written as

$$(1/\Delta x)(E_{i+1/2} - E_{i-1/2}) = (1/\Delta x)[\hat{A}^+{}_{i-1/2} + (a - 1)\hat{A}^-_{i+1/2}]\nabla U_i \tag{25}$$

To eliminate the destabilizing effects of the negative eigenvalues, the coefficient of $\hat{A}^-_{i+1/2}$ should be negative. A stable space-marching scheme is therefore obtained for $a \leq 1$. Some of the more rational choices for a are the following: 1) $a = 0$: changes sign on negative eigenvalues (stable); 2) $a = \frac{1}{2}$: average of U_i and U_{i-1} (stable); 3) $a = 1$: equivalent to neglecting Λ^- (stable); and 4) $a = 2$: standard second-order extrapolation (unstable). Numerical experiments with each of these choices verify these results. It is clear that there is no unique method for specifying the value of U^*_{i+1}. It is also clear from these considerations that the method employed to define U^*_{i+1} is not entirely arbitrary. A careful examination will show that selecting $a = 2$ is roughly equivalent to employing a backward difference for the unmodified streamwise convective flux vector, which, as is well known, produces an unstable scheme. Ota et al.[6] use $a = 1$.

An additional consideration is the fact that the parabolized scheme just discussed is not conservative in subsonic regions. This difficulty becomes apparent if the value of the flux when $i + 1/2$ is a right face, i.e., Eq. (20) combined with Eq. (24) when the current marching station is i,

$$(E_{i+1/2})^R = E_i + (a - 1)(\hat{A}^-_{i+1/2})^R\nabla U_i \tag{26}$$

is compared with the value when $i + 1/2$ is a left face, that is, when the current marching station is $i + 1$

$$(E_{i+1/2})^L = E_i + (\hat{A}^-_{i+1/2})^L\nabla U_i \tag{27}$$

When $i + 1/2$ interface is a left face, U^*_{i+1} is not required since U_{i+1} is being computed. The finite-volume formulation is conservative only if the flux when $i + 1/2$ is a right face is identical to the flux when $i + 1/2$ is a left face, i.e., the flux leaving cell i is exactly equal to the flux entering cell $i + 1$. It is clear that in supersonic flow the flux evaluations are consistent since every element of $\hat{A}^-_{i+1/2}$ is identically zero. In subsonic flow, the difference is given by

$$(E_{i+1/2})^R - (E_{i+1/2})^L = (a - 1)(\hat{A}^-_{i+1/2})^R\nabla U_i - (\hat{A}^-_{i+1/2})^L\Delta U_i \tag{28}$$

Therefore, it is not possible to develop a parabolized FDS scheme that is conservative. Assuming that the flow variation is smooth and monotonic, the two terms on the right side of Eq. (28) are approximately equal to $O(\Delta U^2)$. Therefore, the minimum conservation error would occur for $a = 2$ when the two terms aproximately cancel. Unfortunately, it was shown previously that the space marching approach is well posed only if $a \leq 1$. Given this constraint, it is apparent that the smallest error for a stable scheme occurs when $a = 1$, i.e., when the coefficient of the first term is zero. Any stable, nonzero contribution from the first term will only increase the error. That is, it is expected that the error introduced by selecting $a = 0$ is somewhat larger than the error incurred using $a = 1$.

Convergence of Local Iteration

The primary concern associated with these iterative algorithms is the increased computational effort at each marching station. The objective of this section is to attempt to understand what parameters are significant in determining the rate of convergence of the local iteration.

Linearization of the Implicit Operator

The left side of Eq. 16 is a nonlinear expression for $\Delta Q^l_{i,j}$. Development of an efficient numerical algorithm requires that the implicit operator be linearized. It has been clearly shown by Barth[15] that the form of the linearization plays an important role in the convergence characteristics of the resulting algorithm. In the following, three different linearization procedures will be discussed. The three methods are presented in decreasing order of anticipated effectiveness based on the analysis of Ref. 15.

Rather than giving a lengthy expression for the implicit operator, only one term will be considered. In the first-order spatial approximation, the functional relationship for $F^l_{i,j+1/2}$ from Eq. (3), is given by

$$F^l_{i,j+1/2} = F(Q^l_{i,j+1}, Q^l_{i,j}) \tag{29}$$

so that $\Delta F^l_{i,j+1/2}$ is linearized as

$$\Delta F^l_{i,j+1/2} = \frac{\partial F^l_{i,j+1/2}}{\partial Q^l_{i,j+1}} \Delta Q^l_{i,j+1} + \frac{\partial F^l_{i,j+1/2}}{\partial Q^l_{i,j}} \Delta Q^l_{i,j} \tag{30}$$

Exact evaluation of the Jacobian matrices appearing in Eq. (30) is not a trivial matter. The difficulty is introduced by the presence of $B^+_{i,j+1/2}$ and $B^-_{i,j+1/2}$ in Roe's definition of the flux. The Jacobian for Roe's flux definition is given in the appendix of Ref. 15. For future reference, this method is designated the exact Jacobian linearization. This method was not implemented but is included for completeness.

Considerable simplification can be obtained by treating $B^+_{i,j+1/2}$ and $B^-_{i,j+1/2}$ as locally constant in the linearization procedure. This approximate linearization can be written as

$$\Delta F^l_{i,j+1/2} = \tfrac{1}{2}[B^l_{i,j+1} - (\hat{B}^{+l}_{i,j+1/2} - \hat{B}^{-l}_{i,j+1/2})]\Delta Q^l_{i,j+1} + \tfrac{1}{2}[B^l_{i,j} + (\hat{B}^{+l}_{i,j+1/2} - \hat{B}^{-1}_{i,j+1/2})]\Delta Q^l_{i,j} \tag{31}$$

An analogous expression exists for $\Delta F^l_{i,j-1/2}$. As noted by Barth,[15] this simplification effectively introduces a first-order error in the evaluation of the Jacobian since the matrices are multiplied by a difference in Q. This method is designated the approximate Jacobian linearization.

Even more simplicity can be obtained by using the result of Eq. (22) for the terms in the y direction, i.e.,

$$\Delta F^l_{i,j+1/2} = \hat{B}^{-l}_{i,j+1/2}(\Delta Q^l_{i,j+1} - \Delta Q^l_{i,j}) \tag{32}$$

This linearization is nonconservative in time[15] and is designated the nonconservative linearization.

The expressions for $\Delta E^l_{i+1/2,j}$ and $\Delta E^l_{i-1/2,j}$ depend upon whether parabolized FVS or FDS is employed. If Roe's scheme is employed, one of the linearizations just mentioned is employed. If FVS, or more specifically the Vigneron splitting, is used, the exact Jacobian is employed.

The viscous Δ terms are linearized using standard techniques assuming that μ and κ are locally constant.

Effects of Cell Aspect Ratio and Time-Step Size

It is well known that the cell aspect ratio plays an important role in the convergence rate of virtually any algorithm. In general, for a relaxation scheme, diagonal dominance is a sufficient condition for convergence. The goal here is to attempt to determine the effects of the cell aspect ratio and the choice of the time-step size on the diagonal dominance of the resulting matrix system. For the development below, only the Euler equations will be considered. It would appear that the viscous terms would only enhance the diagonal dominance. The nonconservative linearization is used for illustrative purposes.

Neglecting the viscous terms from Eq. (16) and using Eqs. (22) and (32), the implicit portion of the line relaxation scheme can be written as

$$[(\Delta y/\Delta t)\, I + A^{\text{eff}}_{i,j}(\Delta y/\Delta x)]\Delta x \Delta Q^l_{i,j} + (\hat{B}^{+l}_{i,j-1/2} - \hat{B}^{-l}_{i,j+1/2})\Delta x \Delta Q^l_{i,j} + \hat{B}^{-l}_{i,j+1/2}\Delta x \Delta Q^l_{i,j+1} - \hat{B}^{+l}_{i,j-1/2}\Delta x \Delta Q^l_{i,j-1} \tag{33}$$

where Δx has been factored out of each term and $A^{\text{eff}}_{i,j}$ is the effective Jacobian matrix multiplying $\Delta Q_{i,j}$ due to the streamwise convection term. It is apparent that whatever excess there might be on the main diagonal is due to

$$[(\Delta y/\Delta t)\, I + A^{\text{eff}}_{i,j}\,(\Delta y/\Delta x)]\Delta x \tag{34}$$

Equation (34) shows that the diagonal dominance is a function of $\Delta y/\Delta t$ as well as $\Delta y/\Delta x$. That is, diagonal dominance is achieved only if these terms are sufficiently large. Not surprisingly, the marching-step size should be small for more rapid convergence. In addition, the role played by the unsteady term is also clear as it merely adds a term to the main diagonal.

The preceding development showed that diagonal dominance is related to the time-step size. It is clear that for small time steps the system is diagonally dominant. It is also clear that diagonal dominance is not the only issue. The implicit operator should mimic the steady-state right side; for small time steps, this is not the case. As will be shown in a later section, specification of the time-step size does not appear particularly crucial to the success of the algorithm provided the value is sufficiently large. The approach used here is to compute a time step based on a specified approximate Courant-Friedrichs-Levy (CFL) number given by

$$\nu_{\text{app}} = \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + a\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}\right)\Delta t \tag{35}$$

Typically, the CFL number used in the calculations shown here was on the order of 10^{+06}. No difficulties have been

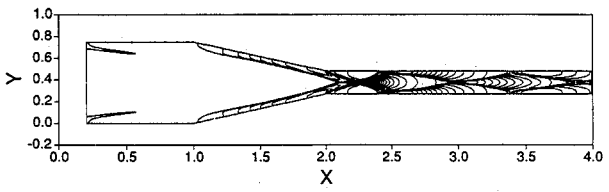

Fig. 1 Predicted pressure contours for hypersonic inlet.

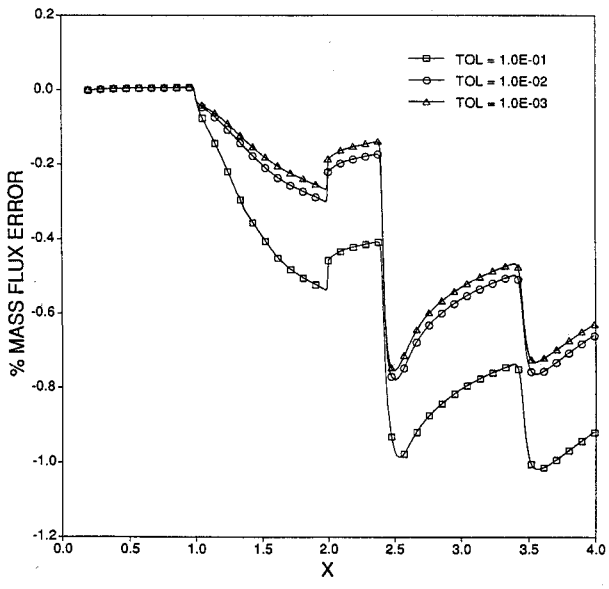

Fig. 2 Convergence of mass flux errors in parabolized FDS scheme for hypersonic inlet.

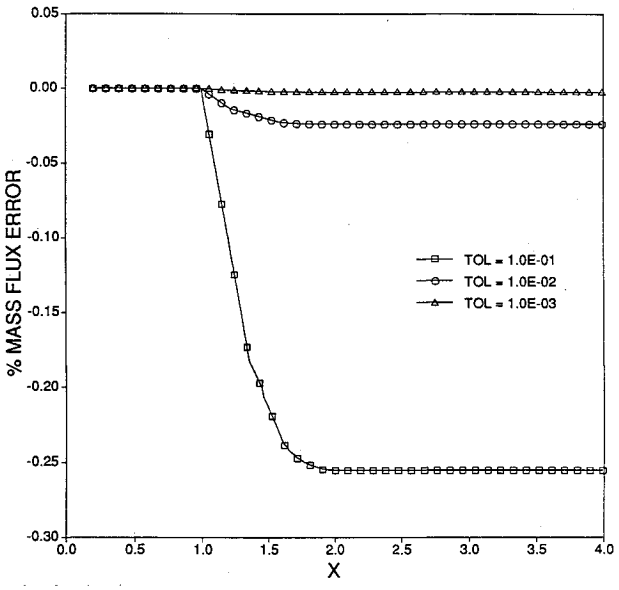

Fig. 3 Convergence of mass flux errors in parabolized FVS scheme for hypersonic inlet.

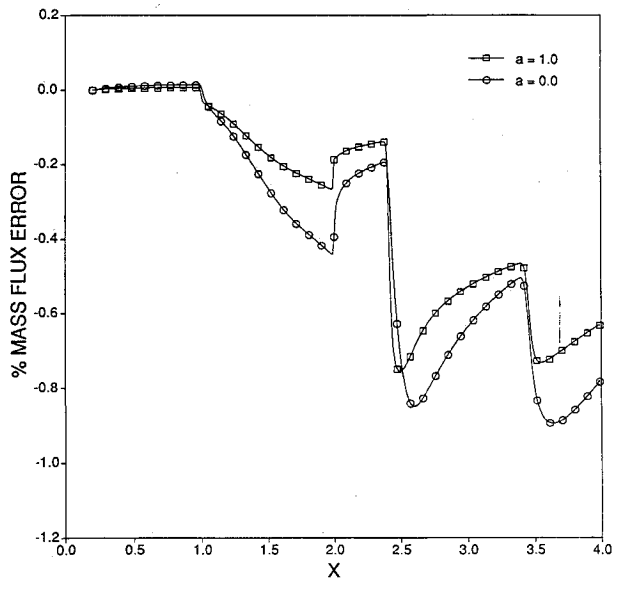

Fig. 4 Parabolized FDS mass flux errors for different values of a.

encountered running arbitrarily large CFL numbers except near the leading edge. Difficulties obtaining a converged solution near the leading edge can be eliminated by employing a smaller value of the CFL number near the leading edge.

Numerical Results

The topics addressed in the body of the paper were investigated using numerical experiments. The first calculations were made to answer two questions: 1) How significant are the conservation errors when parabolized FDS is used in the marching direction and 2) What local iteration tolerance is required to obtain acceptable levels of accuracy when parabolized FVS is used? The L_2 norm of the local residual divided by the local area was monitored to determine convergence. The mass flux was selected as the parameter used to assess the conservation properties of the algorithm because of the simplicity of the calculation. The specific test case used to evaluate the magnitude of the conservation errors is the hypersonic inlet described in Refs. 3 and 5. The lower boundary of the inlet consists of a flat plate, a 15 deg compression, and a 15 deg expansion. The upper boundary was treated as a symmetry plane. The flow conditions are given by $M_\infty = 15.0$, $Re_\infty = 8.0 \times 10^4$, $T_\infty = 100$ K, and $T_{wall} = 1000$ K. Re_∞ is based on the length from the leading edge of the flat plate to the corner. Figure 1 shows pressure contours predicted for the hypersonic inlet. This figure is presented to give the reader a general overview of the problem rather than to present computed results. In both cases, a starting solution was obtained near the leading edge of the plate $0.0 \le x \le 0.2$ and a discrete network of 400×60 cells was used in the lower half of the inlet. The results that will be discussed were obtained using first-order flux evaluations in the marching direction and third-order upwind biased flux evaluations, $\phi = 1/3$ in Eq. (10), in the transverse direction.

Figure 2 shows a plot of the axial distribution of the mass flux error for different local iteration tolerances when parabolized FDS is used in the streamwise direction. The mass flux error at each marching station is defined to be the difference between the computed mass flux and the exact mass flux divided by the exact mass flux. As such, it is a cumulative error. Clearly, the errors are not driven to zero through continued iteration. Not surprisingly, the error is quite small on the flat plate portion ($0.0 < x < 1.0$), where the axial gradients are small, and increases significantly on the ramp ($1.0 < x < 2.0$), where the axial gradients are larger. This figure also illustrates a trend exhibited by all results computed using parabolized FDS in the marching direction—a positive streamwise pressure gradient induces a negative mass flux error gradient. A similar relationship exists between a negative pressure gradient and a positive mass flux error gradient. Figure 3 shows the analogous plot when parabolized FVS is used in the streamwise direction. As the tolerance is decreased, the mass errors are driven to zero as expected. The tolerances 10^{-01}, 10^{-02}, and 10^{-03} correspond to two-, three-, and four-digit accuracy in the mass flow rate, respectively. To further verify the results of the analysis, Fig. 4 shows a comparison of the mass flow rate errors using two different values of a in Eq. (23). As predicted by the analysis, the error is larger for $a = 0$ than for $a = 1$.

Although not shown here, calculations using the second-order parabolized FDS scheme in the marching direction typically show larger errors than the first-order scheme. The second-order parabolized FVS scheme produces five-digit accuracy in the mass flow rate for a local iteration tolerance of 10^{-01}.

The next set of computations was made to determine the effects of different parameters on the rate of convergence of the local iteration. The case considered was the 15 deg compression corner of Holden and Moselle,[16] with PNS results given in Refs. 3 and 6 and Navier-Stokes results given in Refs. 6 and 17. The flow conditions are given by $M_\infty = 14.1$, $Re_\infty = 1.05 \times 10^5$, $T_\infty = 72.2$ K, and $T_{wall} = 297$ K. The computed pressure contours are shown in Fig. 5. Once again, the figure is shown more to orient the reader than to present computed results. The third-order flux evaluations are employed in the transverse direction, and the second-order parabolized FVS scheme is used in the marching direction. The approximate linearization, i.e., Eq. (31), was used in these calculations except where noted.

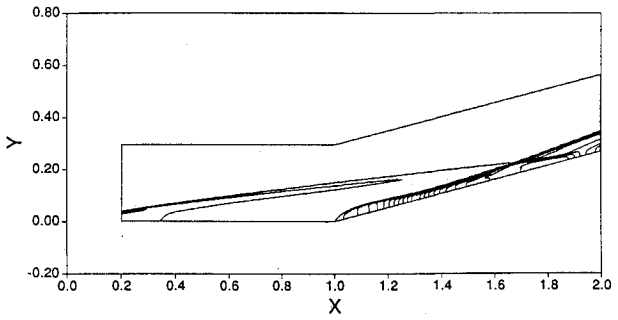

Fig. 5 Predicted pressure contours for hypersonic compression corner.

Figure 6 shows the number of work units (WU) required to obtain a converged solution as a function of the CFL number for several values of the relaxation factor ω. One WU is taken to be the computational effort required to perform one local iteration at each marching station, i.e., the amount of work required to compute a solution using a noniterative PNS algorithm on the same grid. The calculations were performed on a grid with 400×60 cells, and a convergence tolerance of 10^{-01} was used. This tolerance yields five-digit accuracy in the mass flow rate for the large CFL numbers and represents approximately a three to four order of magnitude reduction in the maximum residual at each marching station. In all cases, as the CFL number is increased, the number of work units required for convergence approaches a limiting value. It should be noted that even though the same convergence tolerance was used for each case, the mass flow rate errors vary widely over the CFL number range. For example, the mass flow rate error was two orders of magnitude larger for a CFL number of 10 than for a CFL number of 10^{+06}. This is a manifestation of the fact that the implicit operator does not mimic the steady-state right side for small CFL numbers. This

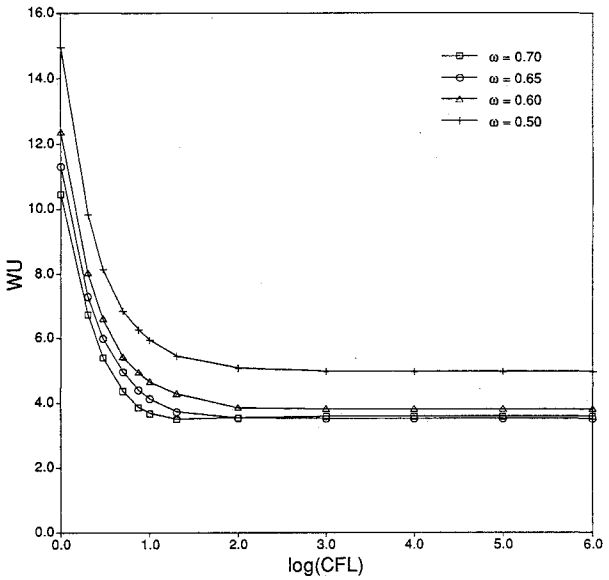

Fig. 6 Convergence as a function of CFL number and relaxation factor.

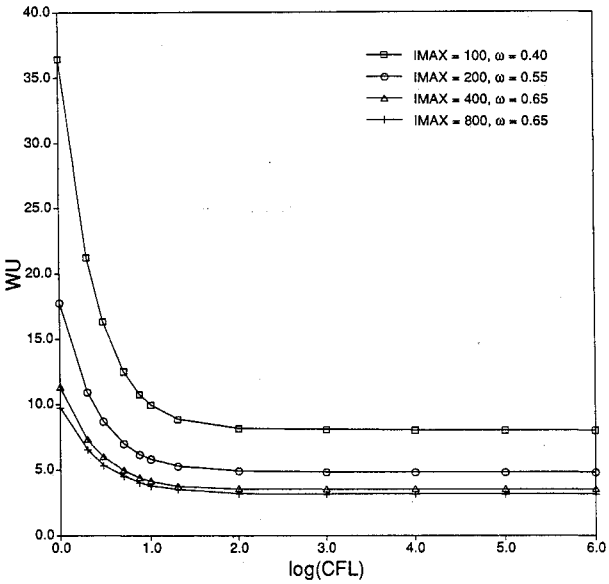

Fig. 7 Convergence as a function of CFL number and streamwise spacing.

Table 1 Total iterations to convergence

IMAX	ω	ω_{max}	Total iterations	Work units
100	0.40	0.43	796	7.960
200	0.55	0.57	964	4.820
400	0.65	0.72	1409	3.523
800	0.65	0.98	2536	3.170

difference occurs only for a partially converged solution. The mass errors are driven to zero if the residual is reduced to zero regardless of the CFL number. An additional point is that, for this case, values of the relaxation factor greater than 0.72 yielded a relaxation scheme that was only conditionally stable. This is in qualitative agreement with the discussion of Thomas et al.[11] where stability limits were observed when a first-order left side was used with a higher-order right side unless underrelaxation was employed.

The sensitivity of the algorithm to the cell aspect ratio was studied by varying the number of equally spaced marching stations. Four cases were run for the hypersonic compression corner using 100, 200, 400, and 800 cells in the marching direction. In all cases, a local iteration tolerance of 10^{-01} was used. Figure 7 shows a plot of the number of work units for convergence vs the CFL number for each of the four cases. It is important to note that for each case, the relaxation factor ω indicated in the figure is the value that gave the most rapid convergence. It is clear that the benefits of decreasing the marching-step size obey a law of diminishing returns. An important point is that although the case with 100 streamwise cells required the most work units, it actually required fewer total iterations and, therefore, less CPU time. Table 1 shows the total number of iterations for each of the cases along with the maximum value of ω that yielded an unconditionally stable algorithm. The algorithm exhibited only conditional stability for values of the relaxation factor larger than ω_{max}, often requiring very small values of the CFL number for convergence. It is also clear that as the mesh spacing was increased, it was necessary to reduce the value of the relaxation factor to maintain unconditional stability. It appears, then, that a finite time step is not necessary for convergence of the algorithm provided sufficient underrelaxation is employed. These results indicate that $\omega = 0.65$ appears to be an optimum value provided that the streamwise spacing is sufficiently small or, more likely, the cell aspect ratio $\Delta y/\Delta x$ is sufficiently large.

The approximate and nonconservative linearization procedures just discussed were implemented to determine which one yielded the most efficient algorithm. Although not shown here, no significant differences in the convergence behavior were observed for the compression corner. In general, the results were too case dependent to draw firm conclusions as to the relative efficiencies of the different linearization procedures.

Conclusions

The new iterative space-marching algorithms for the PNS equations were critically examined. Ostensibly, these new algorithms have the advantage of eliminating linearization and factorization errors through iteration. However, it was shown that if parabolized flux difference splitting is employed in the streamwise direction, significant conservation errors can occur. On the other hand, parabolized flux vector split formulations were shown to be conservative. The effects of several factors on the rate of convergence of the local iteration were also investigated. Numerical results indicate that the stability of the algorithm is more strongly influenced by the magnitude of the relaxation factor than the time-step size. Also, the convergence is relatively insensitive to the CFL number, for sufficiently large CFL numbers, and a given relaxation factor. As far as constructing the implicit operator, results were inconclusive as to the benefits of using the approximate linearization vs the nonconservative linearization. In general, the

results presented here indicate that using very small marching-step sizes is probably not productive. The most effective strategy appears to be employing an infinite CFL number and underrelaxing the solution so that it is stable.

The basic question about these algorithms is whether the increased computational expense due to iteration outweighs the potential benefits. For the cases considered here, numerical solutions were obtained that satisfy the discrete nonlinear conservation law to sufficient accuracy for four to five times the cost of a traditional PNS solution using the same grid. For many problems, this may be an acceptable price to pay for the improvements in accuracy and stability obtained by using this approach.

Acknowledgment

This research was partially funded by the General Dynamics Internal Research and Development Program under Contract 1263915.

References

[1]Vigneron, Y. C., Rakich, J. V., and Tannehill, J. C., "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Leading Edges," NASA TM 78500, June 1978.

[2]Schiff, L. B., and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow," AIAA Paper 79-0130, Jan. 1979.

[3]Lawrence, S. L., Tannehill, J. C., and Chaussee, D. S., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," AIAA Paper 86-1117, May 1986.

[4]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Differencing Schemes," *Journal of Computational Physics,* Vol. 43, No. 2, 1981, pp. 357–372.

[5]Newsome, R. W., Walters, R. W., and Thomas, J. L., "An Efficient Iteration Strategy for Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," AIAA Paper 87-1113, June 1987.

[6]Ota, D. K., Chakravarthy, S. R., and Darling, J. C., "An Equilibrium Air Navier-Stokes Code for Hypersonic Flow," AIAA Paper 88-0419, Jan. 1988.

[7]Chang, C. L., and Merkle, C. L., "The Relation Between Flux Vector Splitting and Parabolized Schemes," *Journal of Computational Physics*, Vol. 80, No. 2, 1989, pp. 344–361.

[8]van Leer, B., Thomas, J. L., Roe, P. L., and Newsome, R. W., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," AIAA Paper 87-1104, June 1987.

[9]Chakravarthy, S. R., and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA Paper 85-0363, Jan. 1985.

[10]Chakravarthy, S. R., and Szema, K. Y., "Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," *Journal of Aircraft,* Vol. 24, No. 2, 1987, pp. 73–83.

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

[12]Steger, J. L., Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods," *Journal of Computational Physics,* Vol. 40, No. 2, 1981, pp. 263–293.

[13]van Leer, B., "Flux Vector Splitting for the Euler Equations," *Lecture Notes in Physics,* Vol. 170, 1982, pp. 507–512.

[14]Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "The Split-Coefficient Matrix Method for Hyperbolic Systems of Gasdynamics Equations," AIAA Paper 80-0286, Jan. 1980.

[15]Barth, T. J., "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," AIAA Paper 87-0595, Jan. 1987.

[16]Holden, M. S., and Moselle, J. R., "Theoretical and Experimental Studies of the Shock Wave-Boundary Layer Interaction on Compression Surfaces in Hypersonic Flow," Calspan, Buffalo, NY, CALSPAN Rept. AF-2410-A-1, 1969.

[17]Hung, C. M., and MacCormack, R. W., "Numerical Solutions of Supersonic and Hypersonic Laminar Compression Corner Flows," *AIAA Journal,* Vol. 14, No. 4, 1976, pp. 475–481.