Upwind Algorithm for the Parabolized Navier-Stokes Equations

Scott L. Lawrence* and John C. Tannehill†
Iowa State University, Ames, Iowa
and
Denny S. Chausee‡
NASA Ames Research Center, Moffett Field, California

A new upwind algorithm based on Roe's scheme has been developed to solve the two-dimensional parabolized Navier-Stokes equations. This method does not require the addition of user-specified smoothing terms for the capture of discontinuities such as shock waves. Thus, the method is easy to use and can be applied without modification to a wide variety of supersonic flowfields. The advantages and disadvantages of this adaptation are discussed in relation to those of the conventional Beam-Warming scheme in terms of accuracy, stability, computer time and storage requirements, and programming effort. The new algorithm has been validated by applying it to three laminar test cases, including flat-plate boundary-layer flow, hypersonic flow past a 15-deg compression corner, and hypersonic flow into a converging inlet. The computed results compare well with experiment and show a dramatic improvement in the resolution of flowfield details when compared with results obtained using the conventional Beam-Warming algorithm.

Introduction

THE parabolized Navier-Stokes (PNS) equations have been used extensively to compute complex steady, supersonic, viscous flowfields. The solution to these equations is marched in space rather than time and therefore is obtained much more efficiently than would be a solution of the complete Navier-Stokes equations. Unlike the boundary-layer equations, however, the PNS equations contain all of the terms of the Euler equations and, as a consequence, the interactions between the viscous and inviscid portions of the flowfield are automatically taken into account.

The PNS equations have been integrated using a variety of finite-difference schemes. Currently, refinements of the noniterative, implicit, approximate-factorization schemes developed by Vigneron et al.[1] and Schiff and Steger[2] represent the state-of-the-art methods for solving the PNS equations. These schemes are based on a class of alternating direction implicit schemes developed by Lindemuth and Killeen,[3] McDonald and Briley,[4] and Beam and Warming[5] to solve time-dependent equations such as the Navier-Stokes equations. One of the major drawbacks of the Beam-Warming type of algorithm is that the central differencing of fluxes across flowfield discontinuities tends to introduce errors into the solution in the form of local flow property oscillations. In order to control these oscillations, some type of artificial dissipation is required. The correct magnitude of this added "smoothing" is generally left for the user to specify through some sort of trial-and-error process.

Received July 21, 1986; revision received Oct. 6, 1988.

*Research Assistant, Department of Aerospace Engineeering.
†Manager, Computational Fluid Dynamics Center, and Professor, Department of Aerospace Engineering. Associate Fellow AIAA.
‡Research Scientist. Associate Fellow AIAA.

In recent years, upwind schemes have been the subject of extensive numerical investigation. The main reason for this attention is that these schemes have the ability to capture the flowfield discontinuities accurately without requiring user-specified smoothing parameters. Until now, the application of upwind schemes has been confined to problems in which the solution is either marched in the time direction[6-9] or relaxed through a pseudotime variable.[10-13] In either case, the function of the upwinding is to model the temporal dispersion of flowfield discontinuities locally. In the present study, where the PNS equations are solved, the spatial propagation of flowfield information is modeled locally using a steady version of Roe's approximate Riemann problem.[14] The algorithm is implicit as well as second-order accurate in the nonmarching direction. The resulting computer code has been used to compute three viscous tests cases, including flat-plate boundary-layer flow, hypersonic flow past a compression corner, and hypersonic flow into a converging inlet. Results of the new algorithm are compared with those of the Beam-Warming scheme as well as with other results, both numerical and experimental.

Governing Equations

The equations describing the planar flow of a Newtonian fluid are the two-dimensional, unsteady Navier-Stokes equations. These can be written in nondimensional strong conservation law form in Cartesian coordinates as

$$\frac{\partial U}{\partial t}+\frac{\partial (E_i-E_v)}{\partial x}+\frac{\partial (F_i-F_v)}{\partial y}=0 \tag{1}$$

where

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E_t\end{bmatrix},\quad E_i=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E_t+p)u\end{bmatrix},\quad F_i=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E_t+p)v\end{bmatrix}$$

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}, \qquad F_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{bmatrix}$$

$$\tau_{xx} = \frac{\mu}{Re_\infty}\frac{2}{3}\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \qquad q_x = \frac{\mu}{Re_\infty}\frac{1}{(\gamma-1)M_\infty^2 Pr}\frac{\partial T}{\partial x}$$

$$\tau_{yy} = \frac{\mu}{Re_\infty}\frac{2}{3}\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right), \qquad q_y = \frac{\mu}{Re_\infty}\frac{1}{(\gamma-1)M_\infty^2 Pr}\frac{\partial T}{\partial y}$$

$$\tau_{xy} = \frac{\mu}{Re_\infty}\frac{2}{3}\left(2\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \qquad E_t = \rho[e + \tfrac{1}{2}(u^2+v^2)]$$

The equations have been nondimensionalized (dimensional quantities are denoted by a tilde) in the following manner:

$$t = \tilde{t}/(\tilde{L}/\tilde{V}_\infty), \quad x = \tilde{x}/\tilde{L}, \quad y = \tilde{y}/\tilde{L}, \quad u = \tilde{u}/\tilde{V}_\infty$$

$$v = \tilde{v}/\tilde{V}_\infty, \quad e = \tilde{e}/\tilde{V}_\infty^2, \quad \rho = \tilde{\rho}/\tilde{\rho}_\infty, \quad T = \tilde{T}/\tilde{T}_\infty$$

$$\mu = \tilde{\mu}/\tilde{\mu}_\infty, \quad p = \tilde{p}/\tilde{\rho}\tilde{V}_\infty^2$$

where $\tilde{L}$ is the flowfield reference length.

The Reynolds number, Re_∞, appearing in the viscous terms is given by

$$Re_\infty = (\tilde{\rho}\tilde{V}_\infty\tilde{L})/\tilde{\mu}_\infty$$

The coefficient of thermal conductivity has been replaced by assuming a constant Prandtl number, and the coefficient of viscosity is calculated using Sutherland's equation

$$\mu = T^{3/2}\left(\frac{1 + 110.4/\tilde{T}_\infty}{T + 110.4/\tilde{T}_\infty}\right)$$

Finally, the system is closed using the perfect-gas equations of state, which in nondimensional form become

$$p = (\gamma - 1)\rho e, \qquad T = \gamma M_\infty^2 p/\rho$$

Coordinate Transformation

A transformation of the spatial coordinates of the form

$$\xi = x, \qquad \eta = \eta(x,y)$$

is applied to Eq. (1) to aid in differencing the equation over a nonuniform grid. The resulting equation written in strong conservation law form is

$$\frac{\partial U'}{\partial t} + \frac{\partial E'}{\partial \xi} + \frac{\partial F'}{\partial \eta} = 0 \tag{2}$$

where

$$U' = U/J$$

$$E' = (\xi_x/J)(E_i - E_v) = (1/J)(E_i - E_v)$$

$$F' = (\eta_x/J)(E_i - E_v) + (\eta_y/J)(F_i - F_v)$$

and J is the Jacobian of the transformation, given by

$$J = \frac{\partial(\xi,\eta)}{\partial(x,y)}$$

Parabolizing Assumptions

The PNS equations are obtained from Eq. (2) by eliminating the time derivative and by neglecting viscous derivatives along lines of constant η. The latter operation is considered a valid assumption for high-Reynolds-number flows. The resulting system is written as

$$\frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} = 0 \tag{3}$$

where

$$\bar{E} = (1/J)E_i, \qquad \bar{F} = [(\eta_x/J)(E_i - E_v^*)] + [(\eta_y/J)(F_i - F_v^*)]$$

The asterisks on E^* and F^* indicate that ξ-derivative terms have been eliminated. The PNS equations are a mixed set of hyperbolic-parabolic equations in the marching coordinate ξ, provided that the inviscid flow is supersonic, the streamwise velocity component is everywhere greater than zero, and the streamwise pressure gradient is either omitted or the "departure behavior" is suppressed by a suitable technique.

Streamwise Pressure Gradient

The presence of the pressure-gradient term in the streamwise momentum equation permits information to be propagated upstream through the subsonic portion of the flowfield, such as that within the boundary layer. As a consequence, a space-marching method of solution is not well posed, and in some cases exponentially growing solutions (departure solutions) are encountered. A number of different techniques have been proposed to eliminate this difficulty. For this study, the method proposed by Vigneron et al.[1] is used.

The Vigneron technique involves splitting the $\bar{E}$ vector into two parts,

$$\bar{E} = \bar{E}^* + \bar{P}$$

where

$$\bar{E}^* = (1/J)[\rho u, \rho u^2 + \omega p, \rho uv, (E_t + p)u]^T \tag{4a}$$

$$\bar{P} = (1/J)[0,(1-\omega)p, 0, 0]^T \tag{4b}$$

The $\bar{E}^*$ vector now replaces $\bar{E}$ in the numerical scheme, and $\bar{P}$ is treated as a source term that is either neglected or is evaluated in the supersonic region. In obtaining the results presented in this paper, the derivative of $\bar{P}$ is neglected. The final form of the governing equations becomes

$$\frac{\partial \bar{E}^*}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} + \frac{\partial \bar{P}}{\partial \xi} = 0 \tag{5}$$

An eigenvalue analysis shows that this system will be hyperbolic-parabolic if

$$\omega = \min\left[1, \frac{\sigma\gamma M_\xi^2}{1 + (\gamma-1)M_\xi^2}\right]$$

where M_ξ is the local streamwise Mach number, and σ is a safety factor included to provide for nonlinearities not accounted for in the analysis.

Numerical Solution of the Equations

First-Order Scheme

The PNS equations can be written in discrete conservation law form as

$$[(1/\Delta\xi)(\hat{E}_j^{n+1} - \hat{E}_j^n)] + [(1/\Delta\eta)(\hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}})] = 0 \tag{6}$$

where n and j are the indices in the ξ and η directions, respectively, and the numerical fluxes, $\hat{E}$ and $\hat{F}$, are approximations to the true fluxes at the sides of the finite volume over which conservation is desired. The preceding discretization results in either an explicit or implicit algorithm, depending on whether

the flux $\hat{F}$ is evaluated at the n or $n+1$ marching step, respectively. The numerical fluxes, of course, will be composed of metric quantities as well as physical flow variables, with the specific definitions of $\hat{E}$ and $\hat{F}$ determined by the choice of algorithm.

The computer code developed in this study employs a flow-solving algorithm that defines

$$\hat{E}_j = \bar{E}_j^* + \bar{P}_s$$

where $\bar{P}_s$ is defined as in Eqs. (4) except that the pressure is lagged in ξ to prevent the occurrence of departure solutions. In order to avoid the difficulty of extracting the required flow properties from the flux vector $\bar{E}^*$, a change is made in the dependent variable from $\bar{E}^*$ to $\bar{U}$ ($\bar{U} = U/J$) through the following linearization:

$$(\bar{E}^*)^{n+1} - (\bar{E}^*)^n = A^{*n}\delta^{n+1}\bar{U}$$

where

$$A^{*n} = \frac{\partial \bar{E}^*}{\partial \bar{U}} \quad \text{and} \quad \delta^{n+1}\bar{U} = \bar{U}^{n+1} - \bar{U}^n$$

The discretized conservation law, Eq. (6), then takes the form

$$A^*\delta^{n+1}\bar{U}_j = -\frac{\Delta\xi}{\Delta\eta}(\hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}}) - (\bar{P}_s^{n+1} - \bar{P}_s^n) \tag{7}$$

At this level, the algorithm is no different than the standard (i.e., Beam-Warming) PNS solver. Where the new algorithm distinguishes itself is the manner that the η numerical flux $\hat{F}$ is specified. In the definition of the inviscid portion of this flux, the algorithm is analogous to the scheme proposed by Roe[14] for solving the unsteady Euler equations. Roe's scheme belongs to the class of upwind schemes referred to as Riemann solvers, which, as the name implies, depend on solutions to Riemann problems in the specification of the numerical fluxes. Rather than solving true Riemann problems, however, Roe's scheme uses solutions to approximate (linearized) Riemann problems and this considerably simplifies the numerical algorithm. The version of the linearized Riemann problem associated with the PNS equations is given by

$$\frac{\partial \bar{E}^*}{\partial \xi} + C_{j+\frac{1}{2}}\frac{\partial \bar{E}^*}{\partial \eta} = 0 \tag{8}$$

with initial conditions

$$\bar{E}^{*n}(\eta) = \begin{cases} (1/J)_{j+\frac{1}{2}}E^*_{j+1} & \text{where } \eta > \eta_{j+\frac{1}{2}} \\ (1/J)_{j+\frac{1}{2}}E^*_j & \text{where } \eta < \eta_{j+\frac{1}{2}} \end{cases}$$

for the $j+\frac{1}{2}$ cell interface. The matrix $C_{j+\frac{1}{2}}$ is of the form

$$C_{j+\frac{1}{2}} = (\eta_x)_{j+\frac{1}{2}}\left(\frac{\partial E_i}{\partial E^*}\right)_{j+\frac{1}{2}} + (\eta_y)_{j+\frac{1}{2}}\left(\frac{\partial F_i}{\partial E^*}\right)_{j+\frac{1}{2}}$$

The calculation of the cell-wall metric terms is explained in the Computational Grid subsection to follow. In spite of the nonconservative form of Eq. (8), the local shock-capturing capabilities of the algorithm can be retained if the flow properties making up $C_{j+\frac{1}{2}}$ are carefully averaged between the grid points j and $j+1$ to satisfy the relation

$$C_{j+\frac{1}{2}}(1/J)_{j+\frac{1}{2}}\Delta E^* = (\eta_x/J)_{j+\frac{1}{2}}\Delta E_i + (\eta_y/J)_{j+\frac{1}{2}}\Delta F_i \tag{9}$$

When the flow is supersonic, Roe's averaging of the variables u, v, and h_t (see Ref. 14) yields flow properties that satisfy this equation. The treatment of the subsonic region will be discussed in a later subsection. The solution to the preceding approximate Riemann problem consists of four constant property regions separated by three lines of discontinuity emanating from the point $(\xi^n, \eta_{j+\frac{1}{2}})$ and having slopes given by the eigenvalues of $C_{j+\frac{1}{2}}$. Of particular interest to the numerical algorithm is the resulting flux across the ray corresponding to the $j+\frac{1}{2}$ grid-cell interface. This flux is given by

$$\begin{aligned} H_{j+\frac{1}{2}} &= (\eta_x/J)_{j+\frac{1}{2}}\,\tfrac{1}{2}[(E_i)_j + (E_i)_{j+1}] \\ &\quad + (\eta_y/J)_{j+\frac{1}{2}}\,\tfrac{1}{2}[(F_i)_j + (F_i)_{j+1}] - \tfrac{1}{2}(\mathrm{sgn}C)_{j+\frac{1}{2}} \\ &\quad \times [(\eta_x/J)_{j+\frac{1}{2}}\Delta E_i + (\eta_y/J)_{j+\frac{1}{2}}\Delta F_i] \end{aligned} \tag{10}$$

In this equation, the matrix sgn$C$ is defined as

$$\mathrm{sgn}C = R(\mathrm{sgn}\Lambda)R^{-1}$$

where R is the matrix of right eigenvectors and sgnΛ is the diagonal matrix having $+1$ and -1 along the diagonal according to the signs of the corresponding eigenvalues (see Ref. 15 for the eigenvalues and eivenvectors). This flux, obtained by analytic solution of the approximate Riemann problem, consists of two parts: a central-differencing type flux plus a first-order upwinding term. Thus, a first-order upwind algorithm for the PNS equations is given by defining the numerical flux as

$$\hat{F}_{j+\frac{1}{2}} = H_{j+\frac{1}{2}} + (\hat{F}_v)_{j+\frac{1}{2}}$$

where $(\hat{F}_v)_{j+\frac{1}{2}}$ is the vector which includes the viscous effects at the cell wall.

Second-Order Scheme

The approach taken in this study to upgrade the algorithm's accuracy in the η direction is similar to that taken by Chakravarthy and Szema[16] for the integration of the unsteady Euler equations. This approach was chosen, after considerable experimentation, over the methods of van Leer[17] and Chakravarthy and Osher[18] because of its relative simplicity and for its observed accuracy and reliability. Also, because the chosen method involves extrapolations of only the flow variables and not of the metrics, it is expected to be better suited for cases involving irregular grids.

The numerical flux for the second-order upwind algorithm is specified in the following manner. First, intermediate variables $(ALF)_{j+\frac{1}{2}}$ are defined by

$$(ALF_1)_{j+\frac{1}{2}} = R^{-1}_{j+\frac{1}{2}}(E^*_j - E^*_{j-1})/J_{j+\frac{1}{2}}$$

$$(ALF_2)_{j+\frac{1}{2}} = R^{-1}_{j+\frac{1}{2}}(E^*_{j+1} - E^*_j)/J_{j+\frac{1}{2}}$$

$$(ALF_3)_{j+\frac{1}{2}} = R^{-1}_{j+\frac{1}{2}}(E^*_{j+2} - E^*_{j+1})/J_{j+\frac{1}{2}}$$

where, as before, $R^{-1}_{j+\frac{1}{2}}$ is the matrix of left eigenvectors. Each element of $(ALF_2)_{j+\frac{1}{2}}$ is a coefficient quantity that, when multiplying its associated right eigenvector, yields the change in $\bar{E}^*$ across the corresponding wave of the approximate Riemann problem at $j+\frac{1}{2}$. The other two ALF quantities are required for increasing the accuracy of the scheme but are less easily physically interpreted.

The preceding vectors are now limited relative to one another in order to achieve essentially oscillation-free shock capturing. The elements of the new vectors,

$(\widetilde{ALF}_1)_{j+\frac{1}{2}}$, $(\widetilde{ALF}_2)_{j+\frac{1}{2}}$, $(\widetilde{ALF}_2)_{j+\frac{1}{2}}$, and $(\widetilde{ALF}_3)_{j+\frac{1}{2}}$,

are given by

$$(\tilde{\alpha}^i_1)_{j+\frac{1}{2}} = \mathrm{minmod}[(\alpha^i_1)_{j+\frac{1}{2}}, b(\alpha^i_2)_{j+\frac{1}{2}}]$$

$$(\tilde{\alpha}^i_2)_{j+\frac{1}{2}} = \mathrm{minmod}[(\alpha^i_2)_{j+\frac{1}{2}}, b(\alpha^i_1)_{j+\frac{1}{2}}]$$

$$(\tilde{\tilde{\alpha}}^i_2)_{j+\frac{1}{2}} = \mathrm{minmod}[(\alpha^i_2)_{j+\frac{1}{2}}, b(\alpha^i_3)_{j+\frac{1}{2}}]$$

$$(\tilde{\alpha}^i_3)_{j+\frac{1}{2}} = \mathrm{minmod}[(\alpha^i_3)_{j+\frac{1}{2}}, b(\alpha^i_2)_{j+\frac{1}{2}}]$$

respectively, where the limiting function is defined by

$$\text{minmod}[x,y] = \text{sgn}(x)\max[0, \min\{|x|, y\,\text{sgn}(x)\}]$$

The parameter b is a compression parameter, which is usually determined by the accuracy parameter ϕ (discussed in a subsequent paragraph), according to the function

$$b = (3-\phi)/(1-\phi)$$

The flux limiting performs the function of reducing the accuracy of the scheme in the immediate vicinity of flowfield discontinuities so that the overshoots and undershoots characteristic of second-order methods are eliminated.

The limited intermediate vectors given earlier are transformed into changes in the numerical flux $\hat{F}$ through multiplication by the eigenvalues and right eigenvectors as follows:

$$(\widetilde{\widetilde{dF}}_1^+)_{j+\frac{1}{2}} = R_{j+\frac{1}{2}}(\Lambda_{j+\frac{1}{2}})^+(\widetilde{\widetilde{ALF}}_1)_{j+\frac{1}{2}}$$
$$(\widetilde{dF}_2^+)_{j+\frac{1}{2}} = R_{j+\frac{1}{2}}(\Lambda_{j+\frac{1}{2}})^+(\widetilde{ALF}_2)_{j+\frac{1}{2}}$$
$$(\widetilde{\widetilde{dF}}_2^-)_{j+\frac{1}{2}} = R_{j+\frac{1}{2}}(\Lambda_{j+\frac{1}{2}})^-(\widetilde{\widetilde{ALF}}_2)_{j+\frac{1}{2}}$$
$$(\widetilde{dF}_3^-)_{j+\frac{1}{2}} = R_{j+\frac{1}{2}}(\Lambda_{j+\frac{1}{2}})^-(\widetilde{ALF}_3)_{j+\frac{1}{2}}$$

where $(\Lambda_{j+\frac{1}{2}})^+$ and $(\Lambda_{j+\frac{1}{2}})^-$ are the diagonal matrices consisting of the elements

$$\lambda^{i+}_{j+\frac{1}{2}} = \tfrac{1}{2}(\lambda^i_{j+\frac{1}{2}} + |\lambda^i_{j+\frac{1}{2}}|) \text{ and } \lambda^{i-}_{j+\frac{1}{2}} = \tfrac{1}{2}(\lambda^i_{j+\frac{1}{2}} - |\lambda^i_{j+\frac{1}{2}}|)$$

respectively.

Finally, the flux changes defined earlier are carefully combined using the accuracy parameter ϕ to yield the second-order-accurate numerical flux

$$\hat{F}_{j+\frac{1}{2}} = H_{j+\frac{1}{2}} + (\hat{F}_v)_{j+\frac{1}{2}} + [(1-\phi)/4](\widetilde{\widetilde{dF}}_1^+)_{j+\frac{1}{2}} + [(1+\phi)/4](\widetilde{dF}_2^+)_{j+\frac{1}{2}} - [(1+\phi)/4](\widetilde{\widetilde{dF}}_2^-)_{j+\frac{1}{2}} - [(1-\phi)/4](\widetilde{dF}_3^-)_{j+\frac{1}{2}} \quad (11)$$

where the first two terms are just the first-order numerical flux of the preceding subsection. It can be shown that when the unlimited, inviscid, second-order numerical flux is substituted into the discretized conservation law, Eq. (7), the resulting algorithm has a leading truncation error term of the form

$$T.E. = \tfrac{1}{4}(\phi - \tfrac{1}{3})(\Delta\eta)^2\left[\frac{\eta_x}{J}\frac{\partial^3 E}{\partial\eta^3} + \frac{\eta_y}{J}\frac{\partial^3 F}{\partial\eta^3}\right]$$

Thus, schemes of varying accuracy can be obtained simply by altering the value of the accuracy parameter, ϕ. However, in the test cases investigated in this study, the results did not show a strong dependence on ϕ. Possible reasons for this are that the flux limiting invalidates the analysis in regions near shock waves and that the viscosity and the truncation error associated with the viscous terms dominate the inviscid truncation error in the boundary-layer region.

Implicit Algorithm

In the previous subsections, the streamwise location of the numerical flux calculation was not specified, and the resulting algorithm, therefore, could be considered either explicit or implicit. The stability restrictions associated with an explicit scheme are relaxed when the numerical fluxes are evaluated at the $n+1$ marching station; however, this results in a nonlinear system of algebraic equations. Linearization of these nonlinear equations allows the solution to be computed without iteration, but another difficulty becomes apparent when the second-order upwind scheme is considered: linearization yields a pentadiagonal system of linear algebraic equations. For this reason, the present algorithm uses first-order numerical fluxes evaluated at the $n+1$ marching station with the second-order correction terms [the last four terms of Eq. (11)] lagged at the nth station. The first-order terms then are linearized producing a block tridiagonal system of equations that can be solved in a relatively efficient manner.

Locally linearizing the fluxes and neglecting changes in the Jacobian J gives the expressions for the fluxes at the new marching station,

$$E_i^{n+1} = E_i^n + J^n A_i^n \delta^{n+1}\bar{U} \text{ and } F_i^{n+1} = F_i^n + J^n B_i^n \delta^{n+1}\bar{U}$$

where

$$A_i = \frac{\partial E_i}{\partial U} \text{ and } B_i = \frac{\partial F_i}{\partial U}$$

These linearized fluxes are substituted into Eq. (10) to produce the inviscid first-order upwind numerical flux at the $n+1$ marching station

$$H^{n+1}_{j+\frac{1}{2}} = H^n_{j+\frac{1}{2}} + \tfrac{1}{2}\bar{B}_i(k_{j+\frac{1}{2}},Q_j)\delta^{n+1}\bar{U}_j + \tfrac{1}{2}\bar{B}_i(k_{j+\frac{1}{2}},Q_{j+1})\delta^{n+1}\bar{U}_{j+1} - \tfrac{1}{2}(\text{sgn}C)^n_{j+\frac{1}{2}}[\bar{B}_i(k_{j+\frac{1}{2}},Q_{j+1})\delta^{n+1}\bar{U}_{j+1} - \bar{B}_i(k_{j+\frac{1}{2}},Q_j)\delta^{n+1}\bar{U}_j]$$

where the matrix sgn$C$ has been lagged at the nth marching step and the Jacobian matrix $\bar{B}_i(k_j,Q_{j2})$ is defined by

$$\bar{B}_i(k_{j1},Q_{j2}) = [(\eta_x/J)_{j1}(A_i)_{j2} + (\eta_y/J)_{j1}(B_i)_{j2}]J_{j2}$$

The indices $j1$ and $j2$ then represent the locations of the metrics and flow variables, respectively.

The viscous flux vector is evaluated at the $n+1$ marching station using a similar linearization strategy. That is

$$(\hat{F}_v^{n+1})_{j+\frac{1}{2}} = (\hat{F}_v^n)_{j+\frac{1}{2}} + \frac{\partial(\hat{F}_v)_{j+\frac{1}{2}}}{\partial\bar{U}_{j+1}}\delta^{n+1}\bar{U}_{j+1} + \frac{\partial(\hat{F}_v)_{j+\frac{1}{2}}}{\partial\bar{U}_j}\delta^{n+1}\bar{U}_j$$

Denoting the viscous Jacobians by

$$\frac{\partial(\hat{F}_v)_{j+\frac{1}{2}}}{\partial\bar{U}_{j+1}} = \bar{B}_v(k_{j+\frac{1}{2}},Q_{j+1}) \text{ and } \frac{\partial(\hat{F}_v)_{j+\frac{1}{2}}}{\partial\bar{U}_j} = -\bar{B}_v(k_{j+\frac{1}{2}},Q_j)$$

the resultant numerical flux at the $n+1$ marching station can be written as

$$\hat{F}^{n+1}_{j+\frac{1}{2}} = \hat{F}^n_{j+\frac{1}{2}} + \{\tfrac{1}{2}[I + (\text{sgn}C)^n_{j+\frac{1}{2}}]\bar{B}_i(k_{j+\frac{1}{2}},Q_j) - \bar{B}_v(k_{j+\frac{1}{2}},Q_j)\}\delta^{n+1}\bar{U}_j + \{\tfrac{1}{2}[I - (\text{sgn}C)^n_{j+\frac{1}{2}}] \times \bar{B}_i(k_{j+\frac{1}{2}},Q_{j+1}) + \bar{B}_v(k_{j+\frac{1}{2}},Q_{j+1})\}\delta^{n+1}\bar{U}_{j+1}$$

The first term on the right-hand side is obtained by evaluating Eq. (11) at the n^{th} marching level. The tridiagonal form of the algorithm is displayed when this numerical flux is substituted into the discretized conservation law, Eq. (7),

$$d\delta^{n+1}\bar{U}_{j-1} + e\delta^{n+1}\bar{U}_j + f\delta^{n+1}\bar{U}_{j+1} = -\frac{\Delta\xi}{\Delta\eta}[\hat{F}^n_{j+\frac{1}{2}} - \hat{F}^n_{j-\frac{1}{2}}] - (\bar{P}^{n+1}_s - \bar{P}^n_s)$$

with

$$d = -\frac{\Delta\xi}{\Delta\eta}\{\tfrac{1}{2}[I + (\text{sgn}C)^n_{j-\frac{1}{2}}]\bar{B}_i(k_{j-\frac{1}{2}},Q_{j-1}) - \bar{B}_v(k_{j-\frac{1}{2}},Q_{j-1})\}$$

$$e = (A_j^*)^n - \frac{\Delta\xi}{\Delta\eta}\{\tfrac{1}{2}[I - (\text{sgn}C)^n_{j-\frac{1}{2}}]\bar{B}_i(k_{j-\frac{1}{2}},Q_j) - \tfrac{1}{2}[I + (\text{sgn}C)^n_{j+\frac{1}{2}}]\bar{B}_i(k_{j+\frac{1}{2}},Q_j) + \bar{B}_v(k_{j+\frac{1}{2}},Q_j) + \bar{B}_v(k_{j-\frac{1}{2}},Q_j)\}$$

$$f = +\frac{\Delta\xi}{\Delta\eta}\{\tfrac{1}{2}[I - (\text{sgn}C)^n_{j+\frac{1}{2}}]\bar{B}_i(k_{j+\frac{1}{2}},Q_{j+1}) + \bar{B}_v(k_{j+\frac{1}{2}},Q_{j+1})\}$$

For the detailed forms of the Jacobian matrices, see Ref. 15.

Finally, a note about the subsonic region. In all of the calculations performed in this study, the subsonic region of the boundary layer is computed using only the central-differencing part of the algorithm. Extending the upwinding to the wall has been found to be detrimental to both the stability and the accuracy of the algorithm. The root cause of this behavior lies in the physically elliptic (with respect to ξ) nature of subsonic flow, which makes it incompatible with an upwind approach. However, this elliptic nature does not allow the existence of discontinuities and, thus, upwinding should not be necessary.

Computational Grid

The resolution of the boundary-layer region is enhanced as much as possible through the use of a computational mesh in which the grid points are clustered toward the wall according to the stretching function

$$z(\eta) = \frac{\beta + 1 - (\beta - 1)[(\beta + 1)/(\beta - 1)]^{1-\eta}}{[(\beta + 1)/(\beta - 1)]^{1-\eta} + 1}$$

In this equation, $z(\eta) = 1$ when $\eta = 1$ and $z(\eta) = 0$ when $\eta = 0$. The points become more tightly clustered as β approaches 1.

The physical distance y then is obtained from

$$y(\xi,\eta) = y_w(\xi) + \delta(\xi)z(\eta)$$

where $\delta(\xi)$ is the distance from the wall to the freestream edge of the grid, and $y_w(\xi)$ is the value of y at the wall.

The metric terms y_ξ and y_η are evaluated at grid points using central differences for the latter and backward differences for the former. The Jacobian of the transformation then can be calculated by

$$J = 1/y_\eta$$

Cell-wall geometries are given in terms of metrics calculated at the grid points as

$$(\eta_x/J)_{j+\frac{1}{2}} = -\tfrac{1}{2}[(y_\xi)_j + (y_\xi)_{j+1}]$$
$$(\eta_y/J)_{j+\frac{1}{2}} = 1$$

Simple averages are also used to compute the cell-wall metrics required for the calculation of viscous derivatives, i.e.,

$$(\eta_x)_{j+\frac{1}{2}} = -\tfrac{1}{2}[(Jy_\xi)_j + (Jy_\xi)_{j+1}]$$
$$(\eta_y)_{j+\frac{1}{2}} = \tfrac{1}{2}(J_j + J_{j+1})$$

Boundary Conditions

At the wall boundary, no-slip conditions are imposed on the velocities and the temperature is specified. In order to determine the wall pressure, zero-gradient extrapolation is explicitly applied to the pressure at the end of each marching step.

For the two external test cases, the freestream edge of the grid is specified explicitly with zero-gradient extrapolations of all of the dependent variables. In the converging inlet test case, the $J\max - 1$ grid line represents a line of symmetry and, therefore, the dependent variables are reflected about this line using

$$U_{J\max} = \bar{R}U_{J\max-1}$$

where

$$\bar{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Numerical Results

Validation of the upwind algorithm for the PNS equations was performed through the computation of three viscous test cases.

Test Case I

The first test, intended to determine the algorithm's ability to compute flowfields dominated by viscous effects, consisted of supersonic, laminar flow over a flat plate. The freestream flow conditions for this case are as follows: $M_\infty = 2.0$, $Re_\infty/\tilde{L} = 1.65 \times 10^6/\text{m}$, $\tilde{T}_\infty = \tilde{T}_w = 221.6\,\text{K}$, $Pr = 0.72$, and $\gamma = 1.4$.

For the calculation of this test case, two PNS codes were employed. In addition to the present upwind code, a two-dimensional version of the Beam-Warming code described in Ref. 1 was used. Results of both codes were compared with those obtained from the compressible boundary-layer code of Pletcher.[19]

Initial conditions were provided by the boundary-layer code at the streamwise location, $\tilde{x} = 0.305$ m. The PNS solutions proceeded from this point with a marching step size of $\Delta\tilde{x} = 0.1 \times 10^{-2}$ m. The grid normal to the wall for both PNS calculations was equally spaced with $\Delta\tilde{y} = 0.1524 \times 10^{-3}$ m, and the top of the grid was kept at a constant height of 0.61×10^{-2} m. The safety factor σ used in evaluating the Vigneron parameter ω was set to 0.8 in all three test cases presented here.

Profiles of tangential velocity and temperature at the marching station, $\tilde{x} = 0.9144$ m, are compared in Figs. 1 and 2, respectively. Figure 3 displays the streamwise variation of the derivative quantity of heat-transfer coefficient. The formula used to compute this quantity is

$$C_h = \frac{\mu_{\text{wall}}}{PrRe_\infty} \frac{1}{\frac{1}{2}(\gamma - 1)M_\infty^2 + 1 - T_w} \frac{\partial T}{\partial n}$$

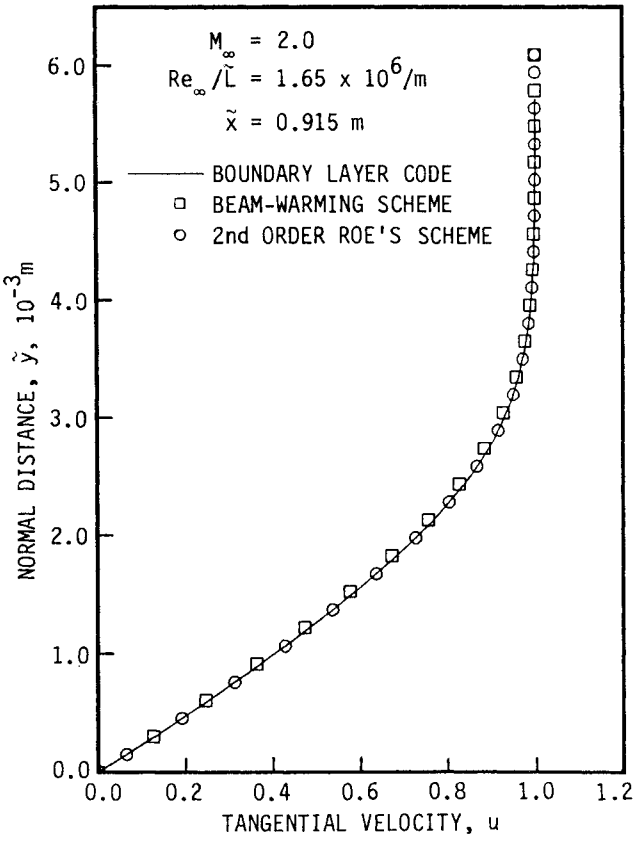


Fig. 1 Comparison of velocity profiles.

where n represents the distance normal to the wall. In all three of these figures, excellent agreement is observed between the PNS results and those of the boundary-layer code. The upwind code gives slightly better results in this case than does the Beam-Warming code, but this is believed to be due to a difference in treatment of the viscous terms and not the upwinding.

The step size used to produce these results corresponds to a maximum Courant number of approximately 50. As the step size is increased, the results of both codes gradually deteriorate, but this effect is slightly more pronounced with the upwind algorithm. This seems to indicate that, for flowfields with relatively gradual variations where no added smoothing is required (none was used in the Beam-Warming calculation of this case), central differencing may be the most appropriate approach. Nevertheless, in this test case, the new algorithm has exhibited a satisfactory ability to compute viscous regions without the upwind dissipation overwhelming the physical viscosity.

An indication of the relative computer effort required by the two PNS codes is given by a comparison of the CPU times and storage involved. The time spent in CPU by the Beam-Warming code is 0.92×10^{-4} s/step/grid point on a Cray-XMP computer. As would be expected, the upwind algorithm is slower, requiring 0.25×10^{-3} s/step/grid point. The additional time is spent computing and multiplying the eigenvalues and eigenvectors for the evaluation of the upwind dissipation terms. It should be noted that vectorization was not a high priority in the development of either of these codes. The storage requirements of both codes are very mild since storage is required in only one dimension.

Also of some interest is the programming effort involved in developing the upwind code. The new code contains aproximately 30% more FORTRAN statements than does the Beam-Warming code. These additional statements are distributed fairly evenly among the calculation of eigenvalues and eigenvectors, the evaluation of the dF's, and the upwinding of the left-hand side.

Test Case II

The second test case computed was that of hypersonic laminar flow over a 15-deg compression corner. The flow conditions, chosen to correspond with one of the cases studied experimentally by Holden and Moselle,[20] are as follows: $M_\infty = 14.1$, $\tilde{T}_\infty = 72.2$ K, $Re_{\tilde{l}} = 1.04 \times 10^5$, $\gamma = 1.4$, $\tilde{l} = 0.439$ m, $\tilde{T}_w = 297$ K, and $Pr = 0.72$ where $Re_{\tilde{l}}$ is the freestream Reynolds number based on the distance from the leading edge to the beginning of the ramp. This flow is supersonic in the inviscid region and exhibits no separation of the boundary layer. Thus, the space-marching procedure is stable. Also, since the flowfield contains an extremely strong shock wave, this case provides a good test of the shock-capturing capabilities of the new algorithm. The flow is illustrated schematically in Fig. 4.

The initial conditions for this case were specified using the second-order Roe's scheme code marched with $\Delta\tilde{x} = 0.5 \times 10^{-3}$ m from freestream conditions to the downstream station at $\tilde{x} = 0.1$ m. Both PNS codes then were restarted from these results and marched further downstream. Forty-five grid points were distributed in the normal direction with a stretching parameter of 1.08. The marching step size downstream of $\tilde{x} = 0.1$ m was kept constant at 0.2×10^{-2} m, and the calculations were terminated at $\tilde{x} = 0.9$ m.

The streamwise distribution of the wall pressure coefficient, defined by

$$C_p = \tilde{p}_w / (\tilde{\rho}_\infty \tilde{V}_\infty^2)$$

is shown in Fig. 5, and the heat-transfer coefficient distribution is given in Fig. 6. Results of the first- and second-order Roe's schemes and the Beam-Warming scheme are compared with the experimental results of Holden and Moselle. Smoothing terms of the form suggested by Hung and MacCormack[21]

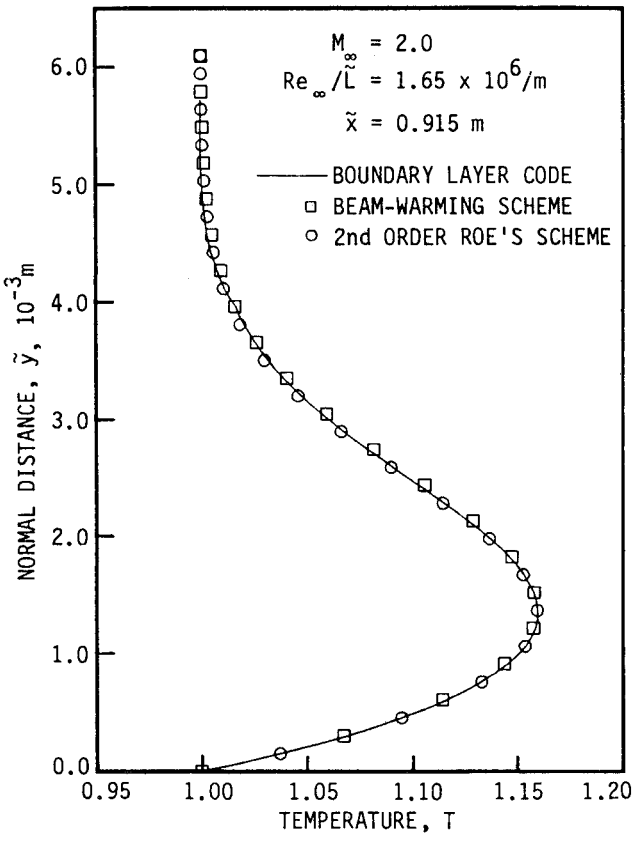


Fig. 2 Comparison of temperature profiles.

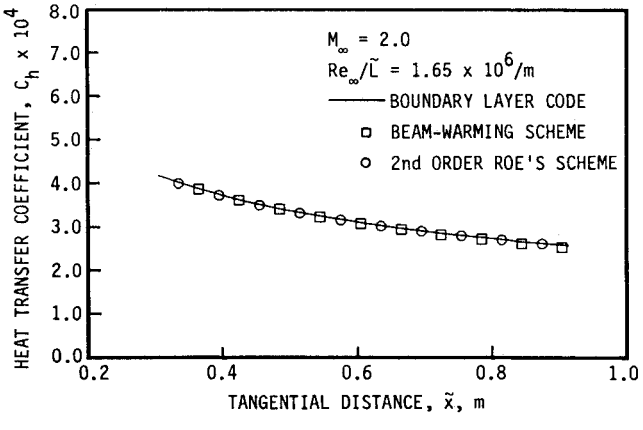


Fig. 3 Comparison of flat plate heat-transfer coefficients.

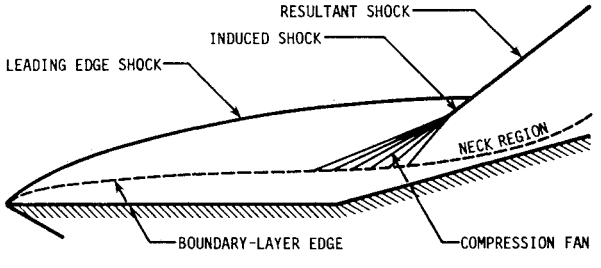


Fig. 4 Hypersonic compression corner test case.

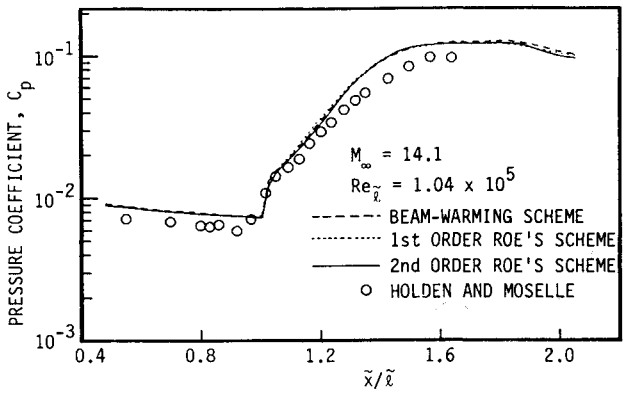


Fig. 5 Comparison of wall pressure coefficients.

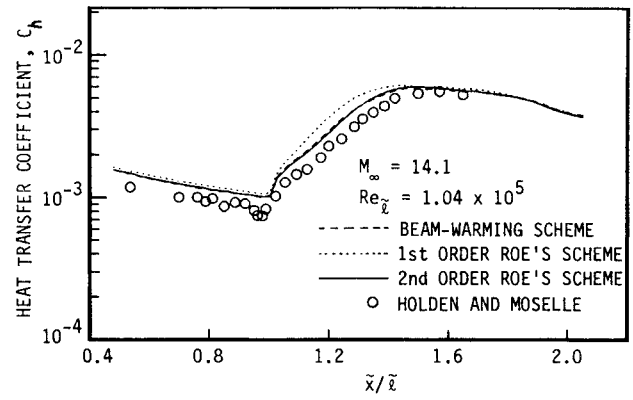
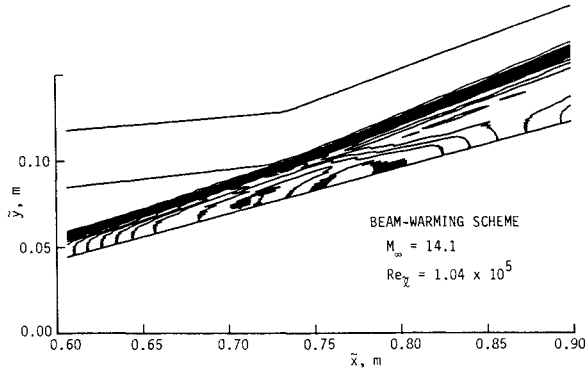
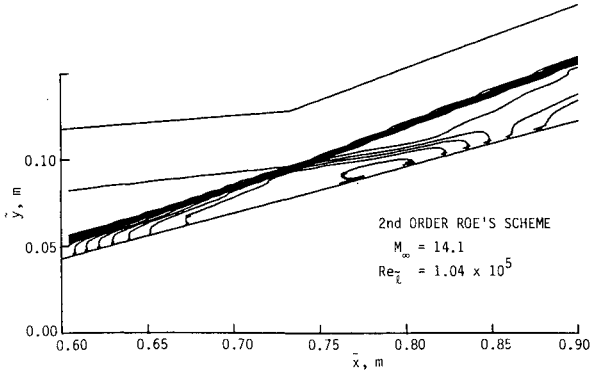


Fig. 6 Comparison of compression corner heat-transfer coefficients.

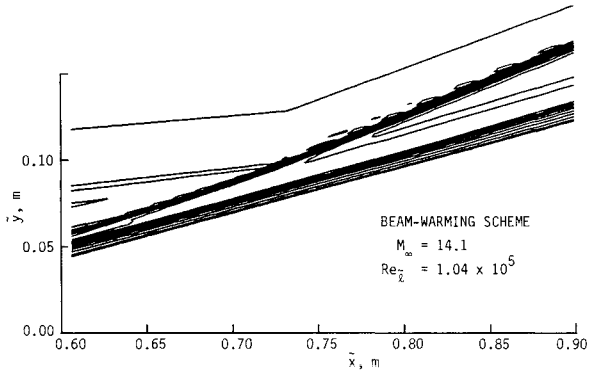


a) Beam-Warming scheme

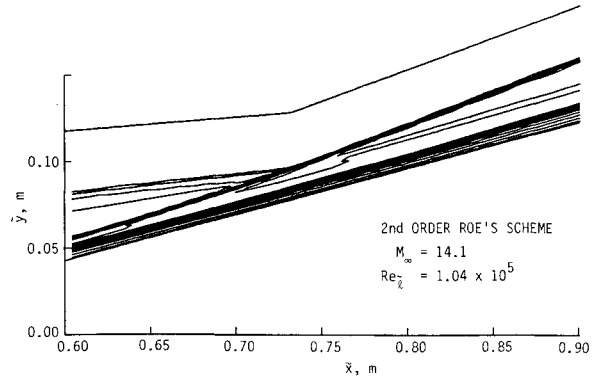


b) Present method

Fig. 7 Ramp case pressure contours.



a) Beam-Warming scheme



b) Present method

Fig. 8 Ramp case Mach contours.

were added to the right-hand side of the Beam-Warming algorithm to control the nonlinear instabilities associated with the strong shock wave of this test case. As anticipated, smoothing was not required for any of the calculations using the new algorithm. The results indicate that the computed wall pressures are relatively insensitive to changes in the algorithm; however, the derivative quantity of heat transfer is noticeably improved by the change form first- to second-order accuracy. The slight qualitative disagreement near the corner between the numerical results and the experiment is due to the single-sweep space-marching procedure; i.e., the flow upstream is not "warned" of the oncoming compression. The slight overprediction of both pressure and heat-transfer coefficients is not so easily explained. The same trend can be seen in the results of Hung and MacCormack[21] for the numerical integration of the full unsteady Navier-Stokes equations. Calculations were also performed using a real-gas Beam-Warming PNS code for equilibrium air. However, because of the very low freestream static temperature, virtually no difference in results was observed.

The details of the shock intersection region are illustrated in the contour maps of pressure and Mach number presented in Figs. 7 and 8, respectively. These figures reveal an interesting aspect of this test case in that the inviscid flowfield along the line, $\tilde{x} = 0.73$ m, represents a large-scale version of the steady Riemann problem considered in the development of the new upwind algorithm. The resulting shock wave and expansion fan are evident in the pressure contours, and the contact sur-

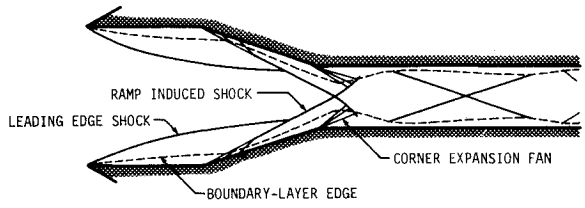


Fig. 9 Hypersonic inlet test case.

face appears in the Mach contours. The most noteworthy feature of these figures, however, is that the flowfield resolution by the upwind scheme is, in general, markedly superior to that achieved using the conventional algorithm. Using the new algorithm, the shock transition takes place over one or two grid points with no associated oscillations, whereas the Beam-Warming scheme generates oscillations that extend several grid points into the shock layer.

An effort was undertaken in this case also to determine the upper bound on marching step size. This study proved the new algorithm to be significantly more robust than the conventional scheme. At large Courant numbers, the smoothing required by the central-differencing scheme to maintain positive pressures at the shock undershoots tended to initiate departure behavior near the wall.

Test Case III

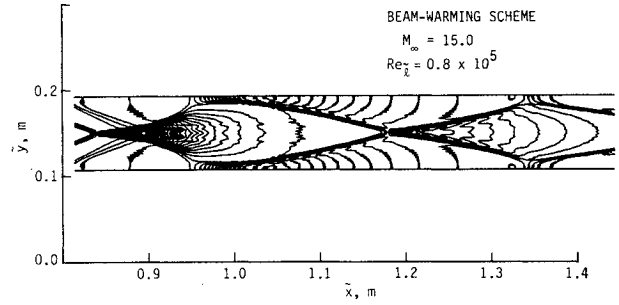
The final test case consists of the hypersonic flow into a two-dimensional converging inlet, which is basically an extension of the previous case. The rear of the ramp is turned back to horizontal, and a reflection condition is applied to the top of the grid. The ramp angle is, again, 15 deg and the horizontal distance from the compression corner to the top of the ramp is 0.4 m. A schematic illustration of this flowfield is shown in Fig. 9. The freestream conditions that were applied are as follows: $M_\infty = 15.0$, $\tilde{T}_\infty = 10.0$ K, $Re_{\tilde{l}} = 8.0 \times 10^4$, $\gamma = 1.4$, $\tilde{l} = 0.4$ m, $\tilde{T}_w = 1000$ K, and $Pr = 0.72$. Again, $Re_{\tilde{l}}$ is the Reynolds number based on $\tilde{l}$, the distance from the leading edge to the beginning of the ramp. This test case is challenging because it involves many interactions between shock waves and expansion fans and also requires the capturing of shock waves that are oblique to the grid, i.e., shock waves that appear to move as the computation proceeds downstream.

Conditions for initializing the PNS calculations for this case were provided in a manner similar to that used in the previous test case. Forty-five grid points were again distributed in the η direction with a stretching parameter of 1.08 and the line of symmetry was located at $\tilde{y} = 0.15$ m. The calculations proceeded from initialization at $\tilde{x} = 0.1$ m to the final marching station at $\tilde{x} = 1.5$ m using the constant step size, $\Delta\tilde{x} = 0.2 \times 10^{-2}$ m.

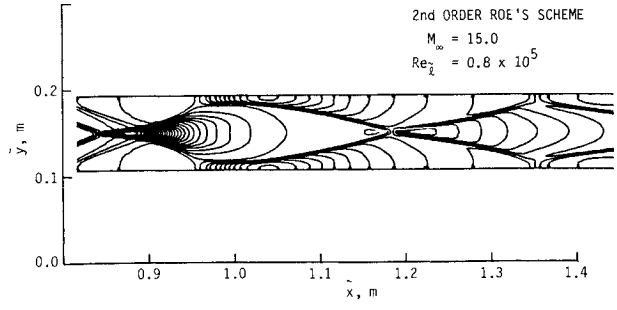
Computed pressure and Mach number contours for this case are presented in Figs. 10 and 11. The shock waves are sharply and smoothly captured with the upwind scheme, whereas oscillations persist in calculations performed with the Beam-Warming scheme despite the addition of artificial smoothing. Also, the results shown in Figs. 10b and 11b were obtained in virtually the first attempt. On the other hand, several runs with the conventional code using different smoothing coefficients were necessary before satisfactory results were obtained.

Concluding Remarks

A new algorithm has been applied to the two-dimensional parabolized Navier-Stokes (PNS) equations. The algorithm, based on Roe's approximate Riemann solver, is implicit, second-order in the nonmarching direction and does not require user specification of smoothing coefficients. In order to validate the method, three laminar test cases were investigated, including supersonic flow over a flat plate, hypersonic flow past a 15-deg compression corner, and, finally, hypersonic flow into a converging inlet. Results compared favorably with experimental or other numerical data in all three cases. The new algorithm requires more CPU time per grid point per run than does the conventional central-differencing approach. However, the improved shock-capturing capability of the new upwind algorithm produces a significant increase in reliability, which offsets the higher computer costs. Work is currently under way to extend the approach to three dimensions in order to produce a PNS code that will be more robust, more accurate, and easier to use than the three-dimensional codes presently in existence.

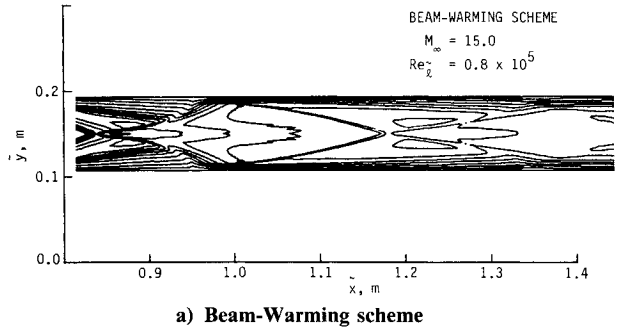


a) Beam-Warming scheme

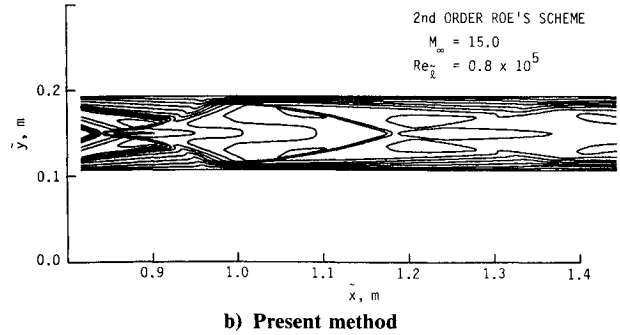


b) Present method

Fig. 10 Inlet case pressure contours



a) Beam-Warming scheme



b) Present method

Fig. 11 Inlet case Mach contours.

Acknowledgments

This work was partially supported by the U.S. Army through NASA Ames Research Center under Grant NCA2-1R340-501 and by the Computational Fluid Dynamics Center, Iowa State University, Ames, Iowa.

References

[1]Vigneron, Y. C., Rakich, J. V., and Tannehill, J. C., "Calculation of Supersonic Viscous Flow Over Delta Wings with Sharp Subsonic Leading Edges," AIAA Paper 78-1137, July 1978.

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

[3]Lindemuth, I. and Killeen, J., "Alternating Direction Implicit Techniques for Two-Dimensional Magnetohydrodynamics Calculations," *Journal of Computational Physics,* Vol. 13, Oct. 1973, pp. 181-208.

[4]McDonald, H. and Briley, W. R., "Three-Dimensional Supersonic Flow of a Viscous or Inviscid Gas," *Journal of Computational Physics,* Vol. 19, Oct. 1975, pp. 150-178.

[5]Beam, R. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal,* Vol. 16, April 1978, pp. 393-401.

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

[7]van Leer, B., "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," *Journal of Computational Physics,* Vol. 23, March 1977, pp. 276-299.

[8]Godunov, S. K., "A Finite-Difference for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Matematiceskii Sbornik,* Vol. 47, 1959, pp. 271-290; also, Cornell Aeronautical Laboratory translation.

[9]Chakravarthy, S. R. and Osher, S., "Numerical Experiments with the Osher Upwind Scheme for the Euler Equations," *AIAA Journal,* Vol. 21, Sept. 1983, pp. 1241-1248

[10]Lombard, C. K., Bardina, J., Venkatapathy, E., and Oliger, J., "Multi-Dimensional Formulation of CSCM—An Upwind Flux Difference Eigenvector Split Method for the Compressible Navier-Stokes Equations," AIAA Paper 83-1895, July 1983.

[11]Chakravarthy, S. R., "Relaxation Methods for Unfactored Implicit Upwind Schemes," AIAA Paper 84-0165, Jan. 1984.

[12]MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, Jan. 1985.

[13]Thomas, J. L. and Walters, W. W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," AIAA Paper 85-1501, July 1985.

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

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

[16]Chakravarthy, S. R. and Szema, K. Y., "An Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," AIAA Paper 85-1703, July 1985.

[17]van Leer, B., "On the Relation Between the Upwind-Differencing Schemes of Godunov, Enquist-Osher and Roe," *SIAM Journal on Scientific and Statistical Computing,* Vol. 5, March 1984, pp. 1-20.

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

[19]Pletcher, R. H., "On a Calculation Method for Compressible Turbulent Boundary Layer Flows with Heat Transfer," AIAA Paper 71-165, Jan. 1971.

[20]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, Rept. AF-2410-A-1, Oct. 1969.

[21]Hung, C. M. and MacCormack, R. W., "Numerical Solutions of Supersonic and Hypersonic Laminar Compression Corner Flows," *AIAA Journal,* Vol. 14, April 1976, pp. 475-481.