Implicit Conservative Schemes for the Euler Equations

Stephen F. Wornom*
NASA Langley Research Center, Hampton, Virginia
and
Mahamed M. Hafez†
University of California, Davis, California

A new approach to a characteristic modeling scheme is presented that does not require the governing equations to be written in characteristic variables or the flux terms to be split into positive and negative parts. The method is based on the observation that, for certain finite volume schemes, the upwind influence can be accounted for through the conditions applied at the boundaries of individual cells instead of flux differencing. The method is developed and applied to the one-dimensional nozzle flow equations for subsonic, transonic, and supersonic flows and is then extended to two dimensions using an approximate factorization. For the problem of a shock wave reflecting from a flat plate, the present solution algorithm reduces the computational work per time step by 40% of that required by standard, central difference, implicit, calculations.

I. Introduction

NUMERICAL methods that attempt to model the physics of the differential equations being approximated by properly accounting for the directions in which signals are propagated can be called characteristic modeling schemes. The standard approach to characteristic modeling schemes is through the implementation of upwind schemes. (See Refs. 1-7.) To devise an upwind scheme, the primative fluxes are split into contributions whose Jacobians have eigenvalues that are either ≥ 0 or ≤ 0, respectively. At every point on the numerical mesh these fluxes are then differenced either upwind or downwind of that point.

One disadvantage of standard conservative upwind schemes is that the computational work per time step is on the order of 2-3 times that of a standard central difference scheme.

The purpose of this paper is to examine a new approach to characteristic modeling schemes. The present approach is based on the observation that for certain finite volume schemes, the upwind influence can be modeled, element by element, through the local conditions at the interface boundaries. This idea is illustrated as follows.

Consider the case where the flow is subsonic and examine the cell shown in Fig. 1a. From the characteristic speeds u, $u+c$, and $u-c$, one can deduce that two conditions can be applied to the inflow boundary and one at the outflow boundary.

In order for the present approach to be possible, the number of boundary conditions needed to close the system of difference equations must be equal to the number required by the differential equations. For natural first-order differential systems such as the Euler equations, a two-point spatial central difference scheme for the flux terms satisfies this criterion. Moreover, as will be seen later, two-point schemes possess the telescoping property that, once the external boundary conditions have been applied, the resulting global algorithm will naturally give the proper upwind influence for all cells throughout the field (subsonic or supersonic). Special points in mixed flows, such as sonic and shock points, are treated separately in a manner consistent with characteristic theory.

The present approach has some positive properties that are given here for one-dimensional subsonic flows. As will be seen later, many of these will carry over to two dimensions.

P1: The solution algorithm is well-suited for calculations using large time steps (no CFL number restriction).

P2: No dissipative terms are required for stability.

P3: The spatial accuracy is $\mathcal{O}(\Delta x^2)$.

P4: The solution algorithm accounts for the proper flow of information as indicated by the characteristic speeds.

P5: Only the analytical boundary conditions are required.

P6: The computational work per time step is always less than that of a central difference scheme.

The method is developed for one-dimensional flows and extended to two-dimensional calculations using an ADI factored scheme, where the two-dimensional problem is reduced to two one-dimensional-like problems and the same approach can then be applied to each factored sweep.

II. One-Dimensional Flows

The one-dimensional flow equations can be written as

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=0 \tag{1}$$

where

$$U\equiv\begin{bmatrix}\rho\\ m\\ E\end{bmatrix} \quad \text{and} \quad F(U)\equiv\begin{bmatrix}m\\ P+m^2/\rho\\ (P+E)m/\rho\end{bmatrix} \tag{2}$$

Here, U is the vector of conservative variables, F the flux vector, and $m\equiv\rho u$. The primitive variables are the density ρ, velocity u, and pressure P. The total energy per unit volume E is related to the pressure through the equation of state

$$P=(\gamma-1)[E-\tfrac{1}{2}m^2/\rho] \tag{3}$$

where γ is the ratio of specific heats.

The number of external boundary conditions that may be applied are determined from the directions in which the characteristics reach the boundaries. Characteristics coming into the computational domain are replaced by boundary conditions. The boundary conditions are discussed in Sec. V.

Presented as Paper 83-1939 at the AIAA Sixth Computational Fluid Dynamics Conference, Danvers, MA, July 13-15, 1983; received Sept. 13, 1983; revision received March 25, 1985.

*Research Scientist, Theoretical Aerodynamics Branch. Member AIAA.

†Professor, Dept. of Mechanical Engineering. Member AIAA.

Difference Scheme

Equation (1) is approximated by the following two-point (spatial), central difference scheme:

$$\frac{1}{2}\left[\frac{\Delta U_i}{\Delta t_i}+\frac{\Delta U_{i-1}}{\Delta t_{i-1}}\right]+\frac{F_i^\ell - F_{i-1}^\ell}{\Delta x}=0 \tag{4a}$$

where

$$\Delta U = U^\ell - U^{\ell-1},\ x_i=(i-1)\Delta x,\ \text{and}\ t^\ell=(\ell-1)\Delta t \tag{4b}$$

and

$$\Delta t_i = \frac{\text{CFLN}\Delta x}{u_i + c_i}\ \text{(local time step)} \tag{4c}$$

with the value of CFLN (Courant-Fredericks-Lewy number) being specified. All quantities in Eqs. (4) are evaluated at the new time level ℓ. The time derivative terms in Eq. (1) have been replaced by a backward-Euler time differencing; hence, the truncation errors are of $\mathcal{O}(\Delta t, \Delta x^2)$.

Equations (4) are linearized by Newton's method and written as

$$B_i \Delta U_i - A_i \Delta U_{i-1} = -\Delta t_i R_i \tag{5}$$

where

$$B_i = \frac{I}{2} + \lambda J_i, \qquad \lambda = \Delta t_i / \Delta x \tag{6a}$$

$$A_i = -\frac{I}{2r} + \lambda J_{i-1}, \qquad r = \Delta t_{i-i}/\Delta t_i \tag{6b}$$

with the Jacobian $\partial F/\partial U = J$ given by

$$J = \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{(\gamma-3)}{2}u^2 & -(\gamma-3)u & (\gamma-1) \\ -\dfrac{\gamma E u}{\rho}+(\gamma-1)u^3 & \dfrac{\gamma E}{\rho}-3\dfrac{(\gamma-1)}{2}u^2 & \gamma u \end{bmatrix} \tag{6c}$$

The steady-state terms in Eq. (4a) are denoted by R_i.

Solution Algorithm (Subsonic Flow)

The solution algorithm for subsonic flow is deduced by applying the number of conditions applicable to the boundaries of each cell. These are determined by the signs of u, $u+c$, and $u-c$. Note that the present method uses only the fact that the directions from which characteristics intersect the boundaries of each cell are related to the characteristic speed. Thus, by choosing the number of conditions based on the sign of these speeds, the characteristic behavior is properly modeled while retaining the primitive fluxes.

If the flow is entirely subsonic, each point at the new time level will have two characteristics coming from upstream and one characteristic coming from downstream. Thus, if we consider a cell where $i-1$ is the inflow boundary and i the outflow boundary (Fig. 1a), two conditions can be applied at the $i-1$ boundary and one at the i boundary.

We begin by applying Eq. (5) in the cell next to the external inflow boundary where $i=1$ locates the inflow boundary and $i=2$ the outflow boundary. After applying the two inflow boundary conditions permissible (take the simple example $\Delta m_1 = \Delta E_1 = 0$), the three difference equations (mass, momentum, and energy) can be solved for three of the four unknowns, say, $\Delta\rho_1$, Δm_2, and ΔE_2, in terms of the fourth unknown $\Delta\rho_2$, to obtain

$$\Delta\rho_1 = d_2^{(1)} + e_2^{(1)}\Delta\rho_2 \tag{7a}$$

$$\Delta m_2 = d_2^{(2)} + e_2^{(2)}\Delta\rho_2 \tag{7b}$$

$$\Delta E_2 = d_2^{(3)} + e_2^{(3)}\Delta\rho_2 \tag{7c}$$

Note that $\Delta\rho_2$, which is unspecified at this point, is simply the one local outflow condition that can be applied in this cell. At this point, the equations for $d_2^{(j)}$ and $e_2^{(j)}$, $j=1,2,3$, are not given; they are included in the general expressions to follow.

If this procedure is repeated for the second cell and Eqs. (7b) and (7c) are used to express the local inflow conditions Δm_2 and ΔE_2 in terms of $\Delta\rho_2$, the following general algorithm is deduced:

$$\Delta\rho_{i-1} = d_i^{(1)} + e_i^{(1)}\Delta\rho_i \tag{8a}$$

$$\Delta m_i = d_i^{(2)} + e_i^{(2)}\Delta\rho_i \tag{8b}$$

$$\Delta E_i = d_i^{(3)} + e_i^{(3)}\Delta\rho_i \tag{8c}$$

where

$$(e_i^{(1)}, e_i^{(2)}, e_i^{(3)})^T = -z^{-1}(b_{11}, b_{21}, b_{31})^T \tag{9a}$$

and

$$\begin{bmatrix} d_i^{(1)} \\ d_i^{(2)} \\ d_i^{(3)} \end{bmatrix} = Z^{-1}\begin{bmatrix} \Delta t_i R_i^{(1)} + a_{12}d_{i-1}^{(2)} + a_{13}d_{i-1}^{(3)} \\ \Delta t_i R_i^{(2)} + a_{22}d_{i-1}^{(2)} + a_{23}d_{i-1}^{(3)} \\ \Delta t_i R_i^{(3)} + a_{32}d_{i-1}^{(2)} + a_{33}d_{i-1}^{(3)} \end{bmatrix} \tag{9b}$$

The elements of A_i and B_i [Eq. (6)] are denoted by a_{jk} and b_{jk} and $R_i = (R_i^{(1)}, R_i^{(2)}, R_i^{(3)})^T$. The matrix Z is

$$Z = \begin{bmatrix} (-a_{11} - a_{12}e_{i-1}^{(2)} - a_{13}e_{i-1}^{(3)}) & b_{12} & b_{13} \\ (-a_{21} - a_{22}e_{i-1}^{(2)} - a_{23}e_{i-1}^{(3)}) & b_{22} & b_{23} \\ (-a_{31} - a_{32}e_{i-1}^{(2)} - a_{33}e_{i-1}^{(3)}) & b_{32} & b_{33} \end{bmatrix} \tag{9c}$$

(Other possible choices for the algorithm are given in Ref. 8.)

The subsonic solution algorithm is applied in two steps. The first step consists of an upstream-to-downstream sweep ($i=2,3,\ldots,I$) in which the coefficients $d_i^{(m)}$ and $e_i^{(m)}$ ($m=1,2,3$) are computed in a recursive manner using Eqs. (9).

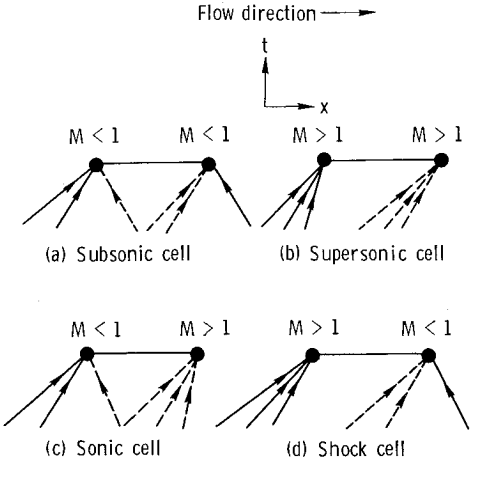


Fig. 1 Types of one-dimensional cells.

This sweep is initiated once the values $d_1^{(2)}$, $d_1^{(3)}$, $e_1^{(2)}$, and $e_1^{(3)}$ are known. These are obtained from the inflow boundary conditions which, for this solution algorithm, are written in the following form:

$$\Delta m_1 = d_1^{(2)} + e_1^{(2)} \Delta \rho_1 \tag{10a}$$

$$\Delta E_1 = d_1^{(3)} + e_1^{(3)} \Delta \rho_1 \tag{10b}$$

Several examples are worked out in Sec. V.

When the downstream boundary is reached ($i = I$), the one outflow boundary condition permissible, say pressure, allows the value of $\Delta \rho_I$ to be computed as follows. The pressure, Eq. (3), is linearized by Newton's method to give

$$\Delta E_I - u_I \Delta m_I + u_I^2 \Delta \rho_I / 2 = (P_{\text{exit}} - P_I^{\ell-1})/(\gamma - 1) \tag{11}$$

After eliminating ΔE_I and Δm_I with Eqs. (8b) and (8c), $\Delta \rho_I$ is obtained. Once $\Delta \rho_I$ is known, a downstream-to-upstream sweep ($i = I, I-1, \ldots, 2$) gives the remaining values of $\Delta \rho_i$, Δm_i, and ΔE_i. [See Eqs. (8).] This two-sweep solution procedure for subsonic flows can be shown to be a block-tridiagonal solver for the following system:

$$\underset{\text{Lower}}{\hat{A}(\Delta \rho_{i-2}, \Delta m_{i-1}, \Delta E_{i-1})^T} + \underset{\text{Middle}}{\hat{B}(\Delta \rho_{i-1}, \Delta m_i, \Delta E_i)^T}$$

$$+ \underset{\text{Upper}}{\hat{C}(\Delta \rho_i, \Delta m_{i+1}, \Delta E_{i+1})^T} = -\Delta t_i R_i \tag{12a}$$

where

$$\hat{A} = \begin{bmatrix} 0 & -a_{12} & -a_{13} \\ 0 & -a_{22} & -a_{23} \\ 0 & -a_{32} & -a_{33} \end{bmatrix} \tag{12b}$$

$$\hat{B} = \begin{bmatrix} -a_{11} & b_{12} & b_{13} \\ -a_{21} & b_{22} & b_{23} \\ -a_{31} & b_{32} & b_{33} \end{bmatrix} \tag{12c}$$

and

$$\hat{C} = \begin{bmatrix} b_{11} & 0 & 0 \\ b_{21} & 0 & 0 \\ b_{31} & 0 & 0 \end{bmatrix} \tag{12d}$$

To permit calculations of flows with supersonic regions, two procedures are possible. The first procedure (the one used herein) is to continue applying, at each cell boundary, the correct number of conditions indicated by the characteristics. At supersonic points the solution algorithm switches to a downstream-marching one. (All of the characteristic speeds are positive.)

The second approach (the one used in Ref. 9) is to locally add dissipation to the difference equations at supersonic points. This was done via the retarded density and retarded pressure concepts. Here, the first approach is followed in order to demonstrate how the characteristic modeling feature can be incorporated at supersonic as well as subsonic points.

Solution Algorithm (Supersonic Flow)

For supersonic flow, the characteristics appear as shown in Fig. 1b. At the cell outflow boundary (index i) no conditions may be specified. At the cell inflow boundary (index $i-1$) all of the characteristics will be coming from upstream and can be replaced by three conditions. Thus, Eq. (5) can be solved for all quantities at the outflow boundary to obtain

$$\Delta U_i = -B_i^{-1}(\Delta t_i R_i + A_i \Delta U_{i-1}), \qquad i = 2,3,..,I \tag{13a}$$

$$U_i^n = U_i^{n-1} + \Delta U_i, \qquad i = 2,3,..,I \tag{13b}$$

Equation (13b) is applied after all of the changes have been computed with Eq. (13a).

Types of Cells

For transonic shocked flow, four types of cells are present. The preceding two sections showed how to solve flows consisting entirely of *subsonic* or *supersonic* cells. The remaining two types occur when 1) the flow changes from subsonic to supersonic (*sonic cell*) and 2) the flow changes from supersonic to subsonic (*shock cell*).

For the sonic cell, let k denote the value of index i at the cell outflow boundary. After applying the two conditions on the subsonic side, Δm_{k-1} and ΔE_{k-1} using Eqs. (8b) and (8c), and three difference equations contain four unknowns. Since no conditions can be specified on the supersonic side of this cell, it is obvious that one extra piece of information is needed to switch from the subsonic to the supersonic algorithm. The additional information is obtained by extrapolating the entropy at the first supersonic point from upstream.

$$(P/\rho^\gamma)_k^\ell = 2(P/\rho^\gamma)_{k-1}^\ell - (P/\rho^\gamma)_{k-2}^\ell \tag{14a}$$

After linearizing Eq. (14a), all boundary values for the sonic cell can be computed. The subsonic and supersonic algorithms can then be applied in their respective regions.

An alternative approach is to require that there be no jump in entropy across the sonic cell. That is,

$$(P/\rho^\gamma)_k - (P/\rho^\gamma)_{k-1} = 0 \tag{14b}$$

In practice, Eq. (14a) gave better results. This is due to the fact that Eq. (14b) can be viewed as a first-order extrapolation, whereas Eq. (14a) is a second-order extrapolation.

In the shock cell the opposite situation occurs—namely, too much information is specified if one attempts to satisfy all three conservation laws. After applying the three conditions on the supersonic inflow side (Fig. 1d) and the single condition on the subsonic outflow side, two unknowns remain to be obtained from the three conservative difference equations. As such, the system is overspecified. Solutions obtained by omitting any one of the three equations in the shock cell produced converged solutions with the incorrect strength and location.

To correct this problem, the following procedure was used. In the shock cell the continuity equation was omitted. In order to retain conservation of mass, the cells before and after the shock were modified as follows ($m^* = \rho_{\text{sonic}} u_{\text{sonic}}$):

$$\tfrac{1}{2}[\rho_t|_{i-1} + \rho_t|_{i-2}] + \frac{m_{i-1} - m_{i-2}}{\Delta x} = -\frac{(m - m^*)_i}{\Delta x} \tag{15a}$$

$$\tfrac{1}{2}[\rho_t|_{i+1} + \rho_t|_i] + \frac{m_{i+1} - m_i}{\Delta x} = \frac{(m - m^*)_{i-1}}{\Delta x} \tag{15b}$$

where i is the outflow boundary to the shock cell. Thus global conservation of mass is guaranteed although the continuity equation is not satisfied in the shock cell. (A similar idea was proposed by Hafez et al.[10] for potential calculations.) The solution algorithm for point $i-1$ then becomes

$$\Delta \rho_{i-1} = d_{i-1}^{(1)} + e_{i-1}^{(1)} \Delta m_i \tag{16a}$$

$$\Delta m_{i-1} = d_{i-1}^{(2)} + e_{i-1}^{(2)} \Delta m_i \tag{16b}$$

$$\Delta E_{i-1} = d_{i-1}^{(3)} + e_{i-1}^{(3)} \Delta m_i \tag{16c}$$

The shock-cell equations (momentum, energy) are then solved for Δm_i and ΔE_i in the form of Eqs. (10) which serve as the boundary conditions for the subsonic region downstream of the shock.

Initial and Boundary Conditions

The initial conditions for ρ, m, and E for the one-dimensional calculations were obtained by linear interpolation between the exact entrance and exit values.

Subsonic inflow. Three examples are given to illustrate the manner in which the coefficients in the boundary condition equations (10) are obtained for subsonic inflow.

Example 1—Mass flux and energy specified ($\Delta m_1 = \Delta E_1 = 0$).

$$d_1^{(2)} = d_1^{(3)} = e_1^{(2)} = e_1^{(3)} = 0 \tag{17}$$

Example 2—Pressure and velocity specified ($\Delta P_1 = \Delta u_1 = 0$).

$$d_1^{(2)} = 0, \quad e_1^{(2)} = u_1 \tag{18a}$$

$$d_1^{(3)} = 0, \quad e_1^{(3)} = u_1^2/2 \tag{18b}$$

Example 3—Mass and entropy ($\Delta m_1 = \Delta S_1 = 0$, $S = S_0 P/\rho^\gamma$).

$$d_1^{(2)} = e_1^{(2)} = d_1^{(3)} = 0 \tag{19a}$$

$$e_1^{(3)} = -\frac{u_1^2}{2} + \frac{c_1^2}{\gamma - 1} \tag{19b}$$

Most types of boundary conditions can be put into the form implied by Eqs. (10); and, they are relatively simple to work out.

Subsonic outflow:

$$P = P_{\text{exit}} \text{ is specified} \tag{20a}$$

Supersonic inflow:

$$\rho_1, m_1, \text{ and } E_1 \text{ are specified} \tag{20b}$$

Supersonic outflow:

No conditions specified

Note that with the exception of the entropy extrapolation or jump condition used in the sonic-cell calculations [see Eqs. (14)], no other numerical boundary conditions are required for either the one- or two-dimensional cases reported herein.

III. Two-Dimensional Euler Calculations

The two-dimensional Euler equations can be written as

$$U_t + F_x + G_y = 0 \tag{21}$$

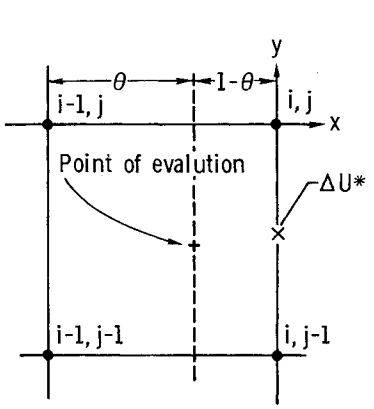


Fig. 2 Two-dimensional cell.

where

$$U = \begin{bmatrix} \rho \\ m \\ n \\ E \end{bmatrix}, \quad F = \begin{bmatrix} m \\ P + m^2/\rho \\ mn/\rho \\ (P+E)m/\rho \end{bmatrix}, \quad G = \begin{bmatrix} n \\ mn/\rho \\ P + n^2/\rho \\ (P+E)n/\rho \end{bmatrix}$$

and $n \equiv \rho v$.

Let $\bar{A} = \partial F/\partial U$ and $\bar{B} = \partial G/\partial U$ be the Jacobian matrices. After replacing the time derivatives by a backward difference in time, Eq. (21) can be written in the following factored form:

$$(I + \Delta t L_x \bar{A})(I + \Delta t L_y \bar{B}) \Delta U = -\Delta t R \tag{22}$$

where I is the identity matrix, L_x and L_y the spatial difference operators, and R the steady-state residual. A fixed time step Δt was used for the two-dimensional implementation.

Equation (22) is solved with the following ADI procedure:

x sweep:

$$(I + \Delta t L_x \bar{A}) \Delta U^* = -\Delta t R \tag{23a}$$

y sweep:

$$(I + \Delta t L_y \bar{B}) \Delta U = \Delta U^* \tag{23b}$$

Update:

$$U^n = U^{n-1} + \Delta U \tag{23c}$$

Implementation of the x Sweep

Using the two-point scheme approximation, Eq. (23a) is written like Eq. (5)

$$B_{i,j-\frac{1}{2}} \Delta U^*_{i,j-\frac{1}{2}} - A_{i,j-\frac{1}{2}} \Delta U^*_{i-1,j-\frac{1}{2}} = -\Delta t R_{k,j-\frac{1}{2}} \tag{24}$$

where A and B are similar to Eqs. (6a) and (6b). Note that the left-hand side of Eq. (24) is evaluated at the midcell. (See Fig. 2.) The flux terms are evaluated as follows:

$$R_{k,j-\frac{1}{2}} = (F_{i,j-\frac{1}{2}} - F_{i-1,j-\frac{1}{2}})/\Delta x + (\bar{G}_{k,j} - \bar{G}_{k,j-1})/\Delta y \tag{25a}$$

where

$$F_{i,j-\frac{1}{2}} = (F_{i,j} + F_{i,j-1})/2 \tag{25b}$$

$$\bar{G}_{k,j} = \theta G_{i,j} + (1-\theta) G_{i-1,j} \tag{25c}$$

$$k = i - 1 + \theta \tag{25d}$$

The i index parameter, θ ($\frac{1}{2} \le \theta \le 1$), is used to add the artificial dissipation at supersonic points. Note that for $\theta = 1$ the scheme has $\mathcal{O}(\Delta x)$ spatial accuracy similar to usual upwind difference schemes.

In general, it is convenient to increase θ near shocks. This was implemented by letting

$$\theta = \min[1, \bar{\theta} + 10\alpha] \tag{25e}$$

where

$$\alpha = |P_{i+1} - 2P_i + P_{i-1}|/(P_{i+1} + 2P_i + P_{i-1}) \tag{25f}$$

A value of $\bar{\theta} = 0.8$ was used.

The two-dimensional problem studied herein is a shock wave reflecting from a flat plate and is illustrated in Fig. 3. For this case, the characteristic speeds relative to the x sweep (u, u, $u+c$, and $u-c$) are always positive since the flow re-

mains supersonic in the streamwise direction. A Cartesian mesh was used. Therefore, all quantities are specified at the external inflow boundary; hence, $\Delta U^*_{1,j-\frac{1}{2}}=0$, and the solution to Eq. (24) reduces to the following one-dimensional supersonic algorithm:

$$\Delta U^*_{i,j-\frac{1}{2}} = B^{-1}_{i,j-\frac{1}{2}}(\Delta t R_{k,j-\frac{1}{2}} + A_{i,j-\frac{1}{2}}\Delta U^*_{i-1,j-\frac{1}{2}}), \qquad i=2,3,\ldots,I \tag{26}$$

Implementation of the y Sweep

Using the two-point approximation, Eq. (23b) is written as

$$C_{i,j}\Delta U_{i,j} - D_{i,j}\Delta U_{i,j-1} = \Delta U^*_{i,j-\frac{1}{2}} \tag{27a}$$

where

$$C_{i,j} \equiv \frac{1}{2}I + \frac{\Delta t}{\Delta y}\bar{B}_{i,j} \tag{27b}$$

$$D_{i,j} \equiv -\frac{1}{2}I + \frac{\Delta t}{\Delta y}\bar{B}_{i,j-1} \tag{27c}$$

Note that the y sweep operator, Eq. (23b), computes values of ΔU at the mesh points.

The y sweep solution algorithm is deduced starting at the cell adjacent to the top boundary. The actual physical variables are updated after the y sweep, therefore, the number of boundary conditions applied to this sweep must be consistent with the boundary conditions dictated by characteristic theory. Since the v component of velocity is subsonic, a portion of the Mach cone brings information from the interior of the computational domain to a boundary point i,J. Therefore only three boundary conditions can be specified along the top boundary downstream of the shock. The remaining variable at the top boundary must be determined as part of the solution. A method of choosing the local cell boundary conditions for the y sweep, consistent with the Mach cone analysis, is by assigning the local cell boundary conditions according to the sign of the characteristic speeds v, v, $v+c$, and $v-c$.

Since the streamlines behind the shock are turned toward the plate, three of the four characteristic speeds (v, v, and $v-c$) will be coming from the exterior. Hence, three boundary conditions are specified there. These are $\rho_{i,J}$, $m_{i,J}$, and $n_{i,J}$. The initial conditions for the two-dimensional case are freestream values everywhere except at the top boundary, where the exact values corresponding to the shock-wave solution shown in Fig. 3 are specified. These boundary conditions are expressed in delta form as

$$\Delta\rho_{i,J} = \Delta m_{i,J} = \Delta n_{i,J} = 0 \tag{28}$$

At the local outflow boundary, index $i,J-1$—the one condition permissible there (see Fig. 4)—is taken to be $\Delta n_{i,J-1}$. The remaining four unknowns can then be solved for in terms of the local outflow boundary condition $\Delta n_{i,J-1}$ as follows:

$$\Delta\rho_{i,J-1} = d^{(1)}_{i,J-1} + e^{(1)}_{i,J-1}\Delta n_{i,J-1} \tag{29a}$$

$$\Delta m_{i,J-1} = d^{(2)}_{i,J-1} + e^{(2)}_{i,J-1}\Delta n_{i,J-1} \tag{29b}$$

$$\Delta E_{i,J} = d^{(3)}_{i,J-1} + e^{(3)}_{i,J-1}\Delta n_{i,J-1} \tag{29c}$$

$$\Delta E_{i,J-1} = d^{(4)}_{i,J-1} + e^{(4)}_{i,J-1}\Delta n_{i,J-1} \tag{29d}$$

If the procedure is repeated for the next cell and Eqs. (29) are used to replace the inflow conditions, the following general algorithm is obtained:

$$\Delta\rho_{i,j} = d^{(1)}_{i,j} + e^{(1)}_{i,j}\Delta n_{i,j} \tag{30a}$$

$$\Delta m_{i,j} = d^{(2)}_{i,j} + e^{(2)}_{i,j}\Delta n_{i,j} \tag{30b}$$

$$\Delta N_{i,j+1} = d^{(3)}_{i,j} + e^{(3)}_{i,j}\Delta n_{i,j} \tag{30c}$$

$$\Delta E_{i,j} = d^{(4)}_{i,j} + e^{(4)}_{i,j}\Delta n_{i,j} \tag{30d}$$

where $1\le j\le J-2$. The coefficients d and e are computed with a top-to-bottom sweep. At the plate, the one boundary condition permissible is $v_{i,1}=0$ or, equivalently, $\Delta n_{i,1}=0$. A bottom-to-top sweep then follows using Eqs. (30). Finally, Eq. (23c) is used to update all of the variables.

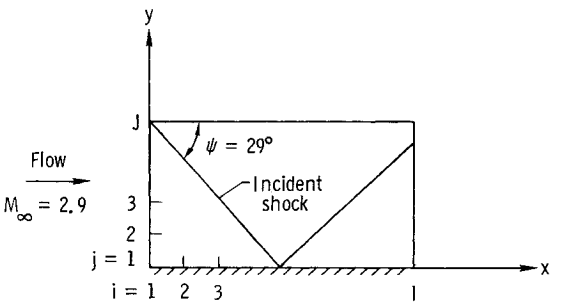


Fig. 3 Two-dimensional problem.

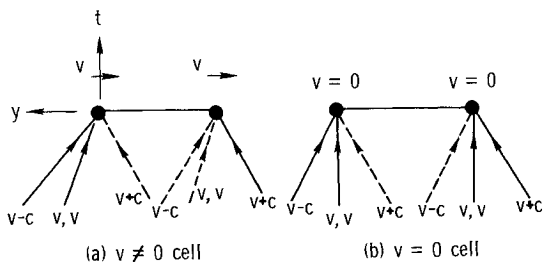


Fig. 4 Cell in y direction.

Table 1 One-dimensional test cases

Case No.	Nozzle type	Description	Area ratio[a]
1	Convergent-divergent	Subsonic inflow, subsonic outflow (no shock)	2:2
2	Divergent	Supersonic inflow, supersonic outflow	
3	Convergent-divergent	Subsonic inflow, supersonic outflow	2:2
4	Divergent	Supersonic inflow, subsonic outflow	
5	Convergent-divergent	Subsonic inflow, subsonic outflow (with shock)	2:1.16

[a] Area ratio = inflow area: outflow area.

Note that the algorithm implied by Eqs. (30) is valid as long as $v \leq 0$. For the test case this is always true.

The two-point solution procedure for the y sweep, Eq. (23b), can be shown to be a block-tridiagonal solver. However, because the matrices are sparse, similar to those of Eqs. (12), the computational effort required is less than a block solver for the three-point difference approximation. This particular algorithm requires 28 fewer multiplications per mesh node for the first step of the y sweep and 12 fewer multiplications per mesh point for the second step [Eqs. (30)] than a corresponding three-point central difference solution algorithm.

IV. Results and Discussion

One-Dimensional Results

The present method was applied to quasi-one-dimensional nozzle flows for subsonic, transonic, and supersonic cases. Each of these cases are discussed below. (See Table 1 for a description of each case.)

Subsonic Flow (Case 1)

Figure 5 shows the convergence history and computed Mach number distribution for a convergent-divergent nozzle with cross-sectional area distribution given by[11]

$$\kappa(x) = \begin{bmatrix} 1 + (\kappa_{\text{inlet}} - 1)(1 - x/5)^2, & x \leq 5 \\ 1 + (\kappa_{\text{exit}} - 1)\dfrac{(x-5)^2}{(x_{\text{exit}} - 5)}, & x > 5 \end{bmatrix} \tag{31}$$

where κ_{inlet} is the inlet area and κ_{exit} the exit area. In Fig. 5a, the error norm is defined as

$$||F||^n = \sqrt{\sum_{i=2}^{I} \sum_{m=1}^{4} (R_i^m)^2} \tag{32}$$

The results given in Fig. 5 were obtained with a CFL number of 10^6; accurate variable profiles were established in eight time steps. The rapid rate of convergence for the one-dimensional cases should not be surprising since, for large time steps, the present method reduces to Newton's method which exhibits quadratic convergence. The very large CFL number used in the subsonic case illustrates the suitability of the present two-point scheme for calculations with large CFL numbers.

Supersonic Expansion (Case 2)

The Mach number profile and residual convergence history for this case are given in Fig. 6. These results were obtained with a CFL number of 10^6; the residual was reduced to machine zero accuracy in five time steps. The comparison between the analytic and numerical Mach number profiles indicates excellent agreement. The nozzle geometry for this case is given by

$$\kappa(x) = 1.398 + 0.347\tanh(0.8x - 4) \tag{33}$$

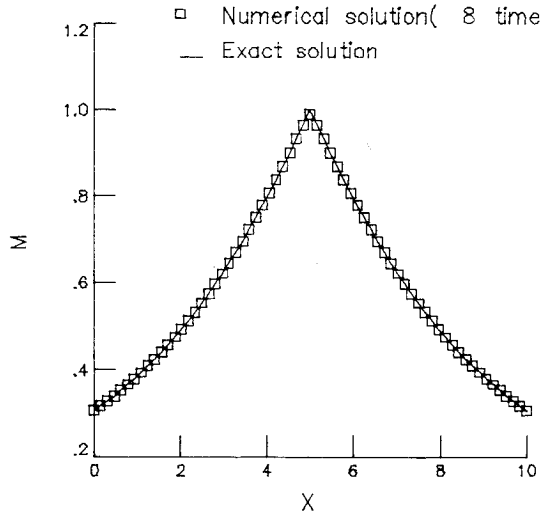
and was obtained from Ref. 12.

Subsonic-Supersonic Expansion (Case 3)

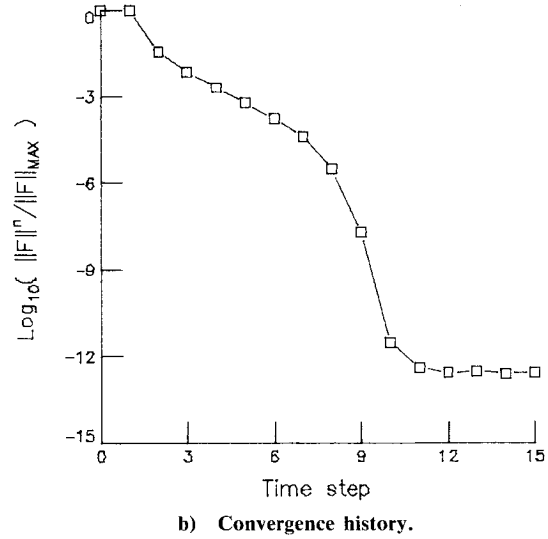
Figure 7 presents the Mach number profile and residual convergence history for this case. The extremely fast convergence rate for this implicit method applied to one-dimensional problems is also evident for this flow.

Supersonic-Subsonic Flow (Case 4)

The convergence history and Mach number profile for the supersonic-subsonic shock case of Shubin et al.[12] are shown in Fig. 8. For the shock cases it was necessary to begin the calculations with a small CFL number until the shock position was fixed, then to increase the time step. This approach was proposed by Dadone and Napolitano[13] and used by Mulder and van Leer.[14] A similar procedure[9] is used here.



a) Mach number.



b) Convergence history.

Fig. 5 Case 1—subsonic flow.

Transonic Flow (Case 5)

The convergence history and Mach number profile for the transonic shock flow case are shown in Fig 9. As can be seen, the agreement with the exact solution is very good. For both shock cases quadratic convergence was observed once the shock location was fixed.

Two-Dimensional Results

The results obtained with the current scheme are compared herein with those obtained with the upwind scheme of van Leer.[7,15]

The present scheme used a fixed time step corresponding to a freestream CFLN_x of 4.0 and a CFLN_y of 1.3. The results using the upwind scheme of van Leer[7,15] were obtained using a constant CFL number of 5. A uniform mesh with 61 horizontal points and 21 vertical points was used.

Figures 10 and 11 present comparisons of the results for the residual convergence history and the centerline static-pressure profile ($y = 0.5$). The normalized residuals are plotted vs the number of time steps and central processing unit (CPU) time in seconds.

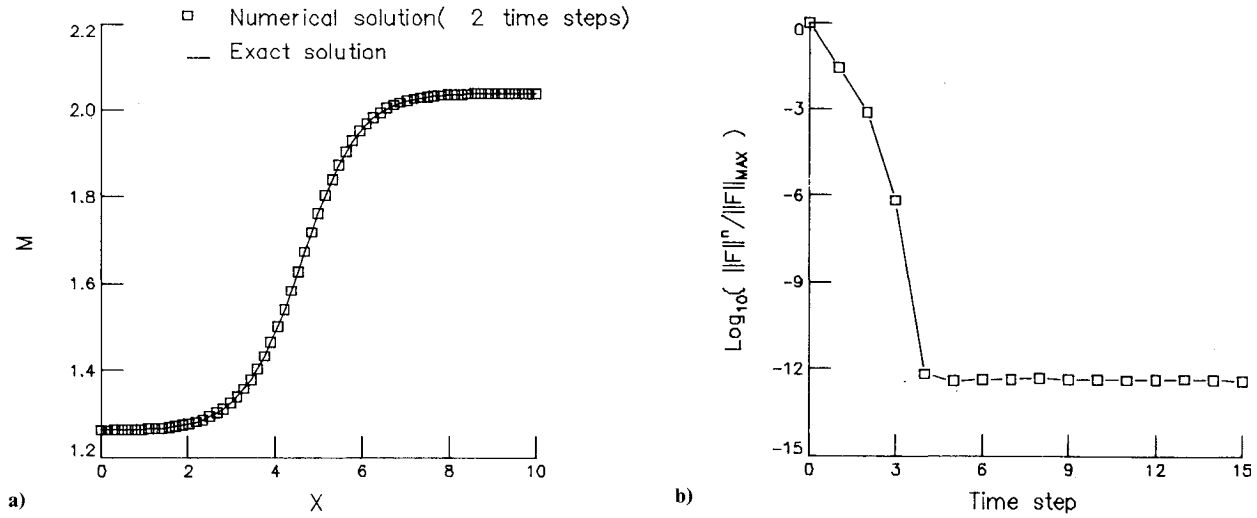


Fig. 6 Case 2—supersonic flow: a) Mach number; b) convergence history.

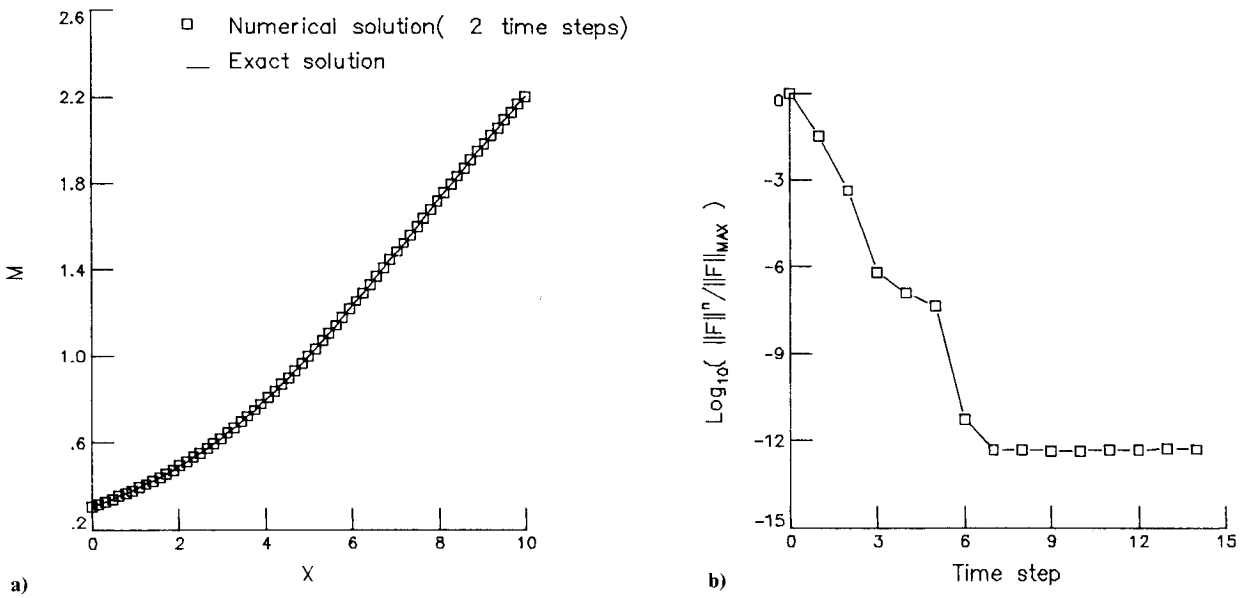


Fig. 7 Case 3—subsonic-supersonic flow: a) Mach number; b) convergence history.

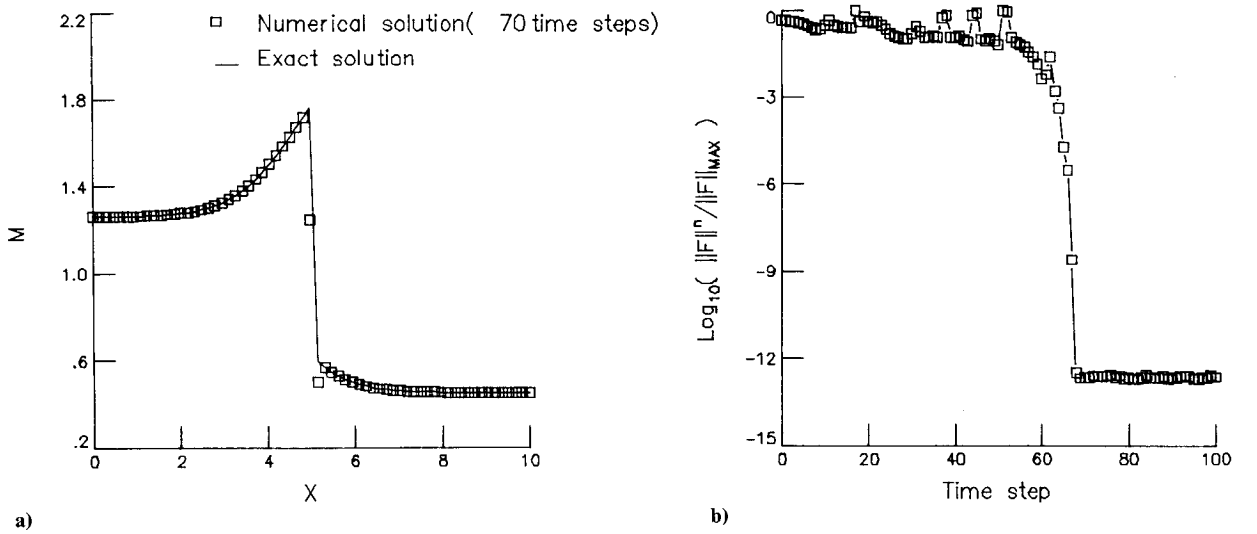


Fig. 8 Case 4—supersonic-subsonic flow: a) Mach number; b) convergence history.

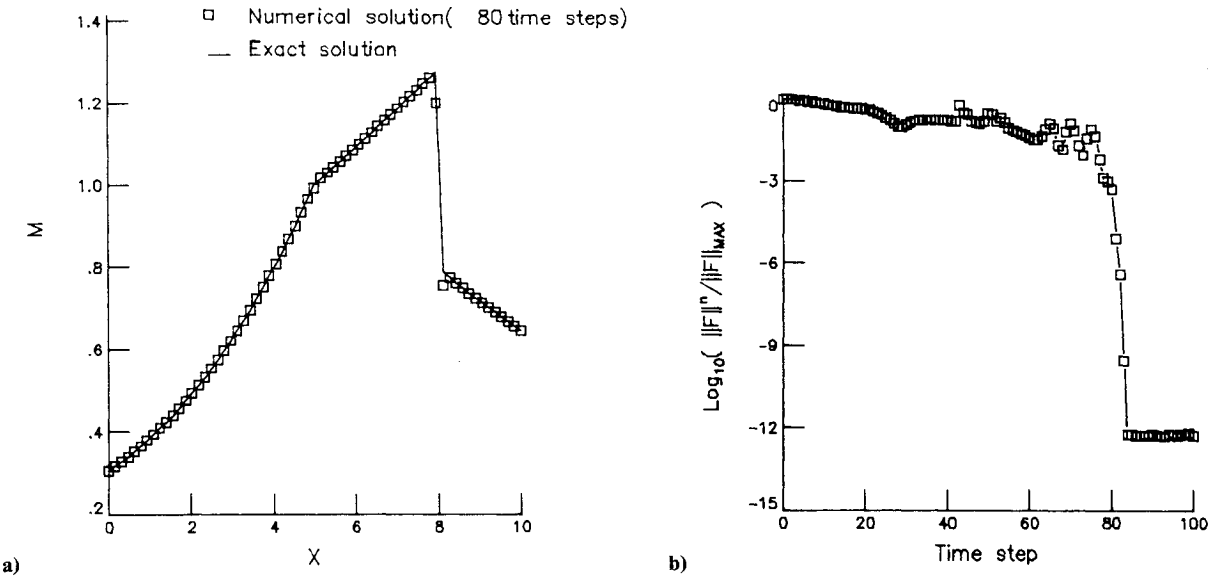


Fig. 9 Case 5—transonic flow: a) Mach number; b) convergence history.

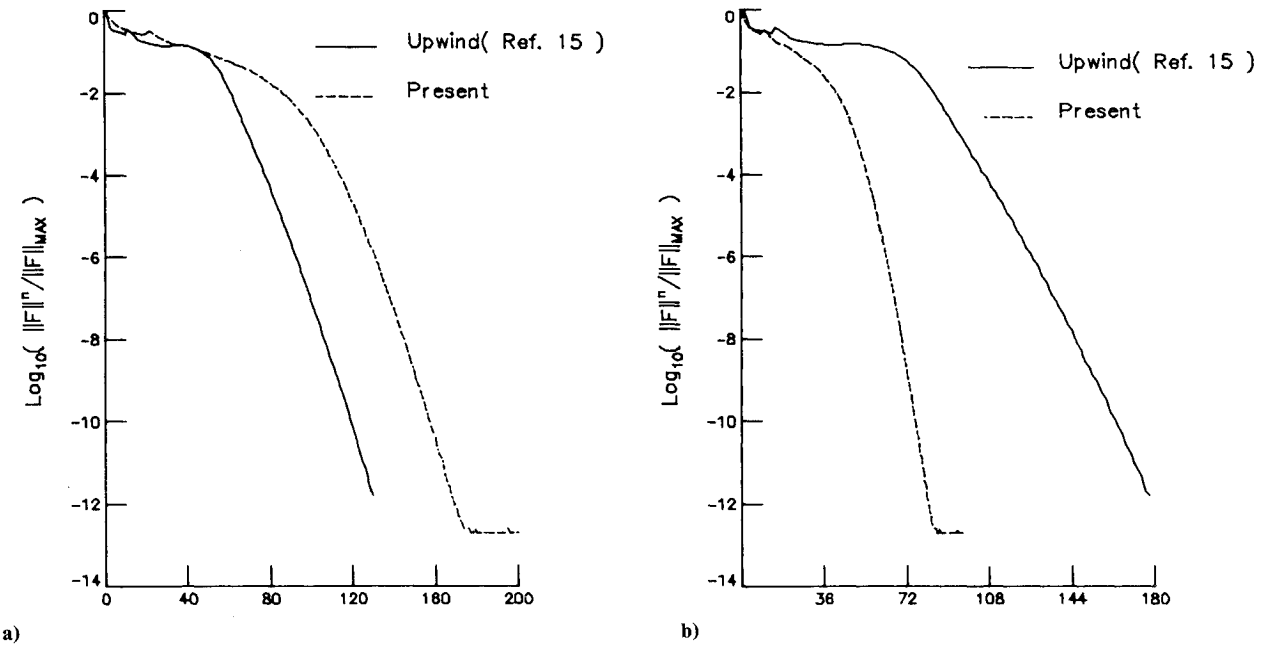


Fig. 10 Shock reflection case: a) time steps; b) CPU time, seconds.

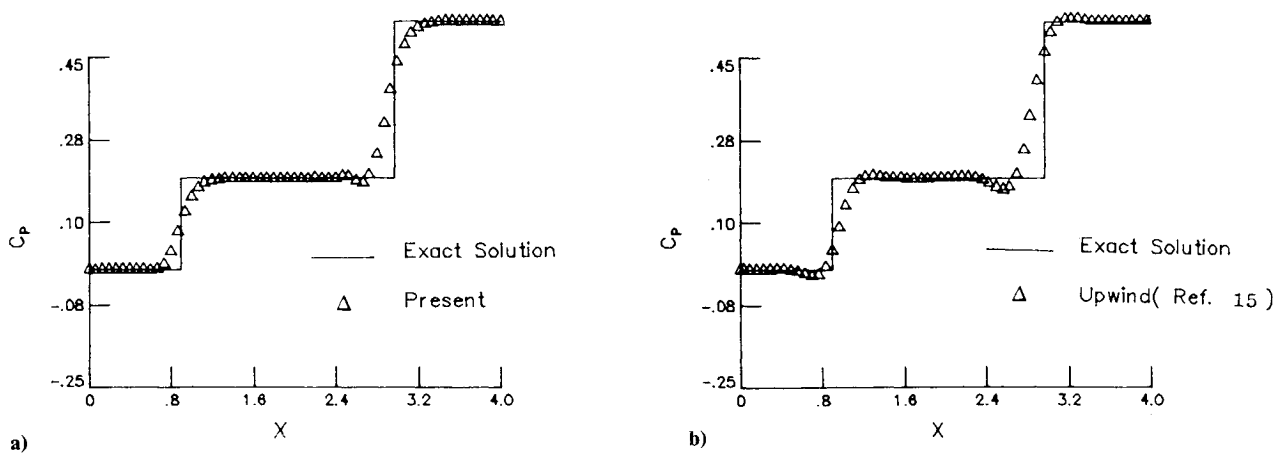


Fig. 11 Pressure coefficient: a) present method; b) upwind scheme.

Table 2 Comparison of work per time step

Code	Relative time
Present	1.00
Beam and Warming	1.67
Upwind	2.77

Figure 10a compares the convergence histories for the two methods vs the time steps. As can be seen, the upwind scheme appears much more efficient. However, Fig. 10b, which shows the convergence history vs CPU time, shows the present method to be more efficient. This is due to the fact that each upwind step requires 2.77 times more work than the present method. Table 2 compares the work per time step of these two methods with the standard Beam and Warming ADI scheme.

Comparisons of the computed midline ($y=0.5$) static-pressure profiles with the exact solution for both methods are shown in Fig. 11. Both methods tend to smear the shocks to the same degree.

V. Conclusions

A two-point (spatial differencing), implicit, characteristic modeling scheme has been presented for the one- and two-dimensional Euler equations. A key feature of the method is that the flux terms are not split into positive and negative parts.

In general, the computational work per time step is less than three-point central difference calculations. Moreover, no extra "numerical" boundary conditions are required. Further practical applications to two-dimensional problems are in progress.

References

[1]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.

[2]Moretti, G., "The λ-Scheme," *Computers and Fluids*, Vol. 7, Sept. 1979, pp. 191-205.

[3]Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "The Split Coefficient Matrix Method for Hyperbolic Systems of Gas-Dynamic Equations," AIAA Paper 80-0268, Jan. 1980.

[4]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, Oct. 1981, pp. 357-372.

[5]Engquist, B. and Osher, S., "One Sided Difference Equations for Nonlinear Conservation Laws," *Mathematics of Computation*, Vol. 36, April 1981, pp. 321-352.

[6]Lombard, C. K., Oliger, J., and Yang, J. Y., "A Natural Conservative Flux Difference Splitting for the Hyperbolic System of Gasdynamics," AIAA Paper 82-0976, 1982.

[7]van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Eighth International Conference on Numerical Methods in Fluid Dynamics*, Aachen, FRG, June 1982, pp. 507-512.

[8]Wornom, S. F., "Implicit Conservative Characteristic Modeling Schemes for the Euler Equations—A New Approach," AIAA Paper 83-1935, July 1983.

[9]Wornom, S. F. and Hafez, M. M., "Calculation of Quasi-One-Dimensional Flows with Shocks," AIAA Paper 84-1245, June 1984.

[10]Hafez, M., Osher, S., and Whitlow, W., "Improved Finite Difference Schemes for Transonic Potential Calculations," AIAA Paper 84-0092, Jan. 1984.

[11]Yee, H. C., Beam, R. M., and Warming, R. F., "Stable Boundary Approximations for a Class of Implicit Schemes for the One-Dimensional Inviscid Equations of Gas Dynamics," *Proceedings of the AIAA 5th Computational Fluid Dynamics Conference*, June 1981, pp. 125-135.

[12]Shubin, G. R., Stephen, A. B., and Glaz, H. M., "Steady Shock Tracking and Newton's Method Applied to One-Dimensional Duct Flow," *Journal of Computational Physics*, Vol. 39, 1981, pp. 364-374.

[13]Dadone, A. and Napolitano, M., "An Implicit Lambda Scheme," AIAA Paper 82-0972, June 1982.

[14]Mulder, B. and van Leer, B., "Implicit Upwind Methods for the Euler Equations," *Proceedings of the AIAA 6th AIAA Computational Fluid Dynamics Conference*, July 1983, pp. 303-310.

[15]Anderson, W. K., Thomas, J. L., and van Leer, B., "A Comparison Between Finite-Volume Flux-Vector Splitting for the Euler Equations," AIAA Paper 85-0122, Jan. 1985.