Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets

Sukumar R. Chakravarthy* and Kuo-Yen Szema*
Rockwell International Science Center, Thousand Oaks, California

A new finite-difference scheme has been developed to solve efficiently the unsteady Euler equations for three-dimensional inviscid supersonic flows with subsonic pockets. The technique utilizes planar Gauss-Seidel relaxation in the marching direction and approximate factorization in the crossflow plane. An "infinitely large" time step is used in parts of the flowfield where the component of velocity in the marching direction is supersonic—here the Gauss-Seidel sweeps are restricted to the forward direction only, and the procedure reduces to simple space-marching; a finite time step is used in parts of the flowfield where the marching component of velocity is subsonic—here, backward and forward Gauss-Seidel sweeps are employed to allow for upstream and downstream propagation of signals, and a time-asymptotic steady state is obtained. The discretization formulas are based on finite-volume implementations of high accuracy (up to third-order) total variation diminishing formulations. Numerical solutions are obtained for an analytically defined forebody, a realistic fighter configuration, and the Space Shuttle. The results are in very good agreement with available experimental data and numerical solutions of the full-potential equation.

Nomenclature

A,B,C	= Jacobian matrices
a,b,c,d	= cell surface points
c	= speed of sound
E,F,G	= fluxes (Eq. 2)
f	= numerical flux
h	= first-order-accurate numerical flux
h	= enthalpy
I	= identity matrix
J	= Jacobian of transformation
j,k,l	= computational coordinate indices
L,R	= left and right eigenvector matrices
l,r	= left and right eigenvectors
M_∞	= freestream Mach number
N	= set of normals: n_x, n_y, n_z
$n_{x,y,z}$	= x, y, z components of cell-face normals
p	= pressure
Q	= dependent variable vector (Eq. 2)
q	= dependent variable vector at a known state
t	= time
$\bar{U}$	= contravariant velocity
u,v,w	= Cartesian velocity components
V	= cell volume
x,y,z	= Cartesian coordinates
α	= angle of attack
α	= pseudocharacteristic variables
γ	= ratio of specific heats for perfect gas
θ	= $(u^2+v^2+w^2)/2$
λ	= eigenvalue
Λ	= diagonal matrix whose diagonal elements are eigenvalues
ξ,η,ζ	= transformed coordinates
ρ	= density
τ	= transformed time coordinate
ϕ	= accuracy parameter

Superscripts

i	= index associated with ith eigenvalue
$(\bar{\ })$	= fluxes in the transformed domain
$(\hat{\ })$	= numerical flux at cell face
$(\tilde{\ }),(\tilde{\tilde{\ }})$	= slope-limited limited values [Eq. (23)]

Introduction

FOR fully supersonic flows, an efficient strategy for obtaining numerical solutions is to employ space-marching techniques. Realistic high-speed flight vehicle configurations often give rise to subsonic pockets, even though they fly at supersonic speeds. For such predominantly supersonic flows, a hybrid approach is suitable: a space-marching technique for the supersonic parts and a relaxation technique for the subsonic parts. Such a hybrid approach has been developed for potential flows by Shankar and Szema and their coworkers.[1-3] For the Euler equations, however, the hybridization is conventionally achieved by coupling separate space-marching and time-marching codes, each with disparate grid systems, etc. Here, a unified approach is presented for efficiently solving the Euler equations for three-dimensional supersonic flows with subsonic pockets. The aim is to develop an Euler solver as versatile as the potential flow solvers[1-3] in treating complex and realistic aircraft, Space Shuttle, and other types of flight vehicle configurations. By solving the Euler equations, however, it is hoped that a wider range of flows with stronger shocks, rotational slipstreams, etc., can be computed. Such flows would be outside the scope of the irrotationality assumptions built into the potential flow simulations.

The new approach utilizes finite-volume implementations of high accuracy (up to third-order) total variation diminishing (TVD) discretizations and is thus expected to be more accurate and reliable than other Euler space-marching and time-marching techniques based on central-difference approximations. In contrast to these latter methods, there are no parameters in the present approach for fine-tuning numerical dissipation for every case. Numerical oscillations are, for the most part, eliminated by using TVD-scheme-based discretizations.

The new approach is based on the unsteady Euler equations. However, in the supersonic parts of the flow (where the veloc-

Presented as Paper 85-1703 at the AIAA 18th Fluid Dynamics and Plasmadynamics and Lasers Conference, Cincinnati, OH, July 16-18, 1985; received Nov. 19, 1985; revision received Oct. 17, 1986.

*Member Technical Staff, Computational Fluid Dynamics Department. Member AIAA.

ity component normal to the cross-flow plane that identifies the local marching direction is supersonic), an "infinitely large" time step (which implies an "infinitely small" reciprocal of time step) is employed. This makes the transient terms of the discretized unsteady equations vanish. In subsonic parts of the flow, a finite time step is employed, and the steady state is approached as a time asymptote.

The new solution approach is based on a planar Gauss-Seidel relaxation method coupled to approximate factorization in the crossflow plane. In supersonic parts of the flowfield, the Gauss-Seidel method is restricted to forward sweeps, and thus the solution procedure reduces to a simple marching technique. In subsonic parts, both forward and backward sweeps are used, along with the finite time steps mentioned earlier. Stability of such an approach is guaranteed by the diagonal dominance resulting from using TVD discretizations in the marching dirction in the transonic parts of the flowfield. This is a crucial difference between conventional hybrid Euler solvers and the new approach. In conventional approaches, space-marching and time-marching techniques must be applied in overlapping regions for stability. In the new unified approach, there is no need for overlap.

In the following sections, the new method is described in detail. The equations are first cast in finite-volume discrete conservation law form. Then the evaluation of volume and metrics is explained. This essentially completes the treatment of geometry, and details of the algorithm are presented next. TVD discretizations are explained first. Then the marching/relaxation procedure is described. This covers the use of approximate factorization in the crossflow plane, the reduction of the Gauss-Seidel procedure to a marching procedure in supersonic zones, etc. The boundary-point treatment is also explained briefly.

In the results section, calculations for an analytically defined forebody are presented first to illustrate some features of the new algorithm. Results for many conical flow cases have also been obtained but are presented elsewhere.[4] Next, results are presented for a realistic fighter-aircraft configuration with fuselage, canopy, wing, nacelle, and vertical tail. Finally, results for the Space Shuttle Orbiter configuration are presented.

Finite-Volume Framework

In this section, the finite-volume framework chosen to implement the algorithm is described. The semidiscrete conservation law form is introduced and associated with a finite-volume formulation of the geometry. Then detailed formulas for the evaluation of the cell-volume and cell-face normals are provided.

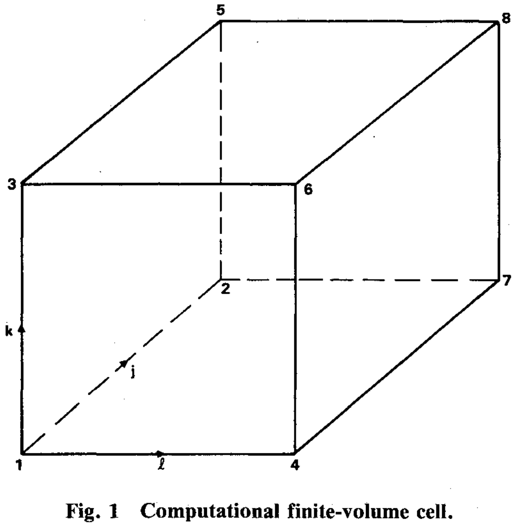

Fig. 1 Computational finite-volume cell.

Semidiscrete Conservation Law

The conservation law form of the unsteady Euler equations in the Cartesian coordinates x, y, z and time t is given by

$$Q_t + E_x + F_y + G_z = 0 \qquad (1)$$

where the dependent variable vector Q and the fluxes E, F, and G are given by

$$Q = \begin{bmatrix} e \\ \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix}, \quad E = \begin{bmatrix} (e+p)u \\ \rho u \\ \rho u^2+p \\ \rho vu \\ \rho wu \end{bmatrix}$$

$$F = \begin{bmatrix} (e+p)v \\ \rho v \\ \rho uv \\ \rho v^2+p \\ \rho wv \end{bmatrix}, \quad G = \begin{bmatrix} (e+p)w \\ \rho w \\ \rho uw \\ \rho vw \\ \rho w^2+p \end{bmatrix} \qquad (2)$$

In the above, pressure is p, density is ρ, Cartesian x, y, z velocity components are u, v, w, and the total energy per unit volume is e, computed from $e=p/(\gamma-1)+\rho(u^2+v^2+w^2)/2$.

Assuming a time-invariant grid, under the transformation of coordinates implied by

$$\tau = t, \qquad \xi = \xi(x,y,z), \qquad \eta = \eta(x,y,z), \qquad \zeta = \zeta(x,y,z) \qquad (3)$$

Eq. (1) can be recast into the conservation form given by

$$\bar{Q}_\tau + \bar{E}_\xi + \bar{F}_\eta + \bar{G}_\zeta = 0 \qquad (4)$$

where

$$\begin{aligned} \bar{Q} &= Q/J \\ \bar{E} &= (\xi_x/J)E + (\xi_y/J)F + (\xi_z/J)G \\ \bar{F} &= (\eta_x/J)E + (\eta_y/J)F + (\eta_z/J)G \\ \bar{G} &= (\zeta_x/J)E + (\zeta_y/J)F + (\zeta_z/J)G \end{aligned} \qquad (5)$$

where, in turn, J is the Jacobian of the transformation

$$J = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} \qquad (6)$$

Associating the subscripts j, k, l with the ξ, η, ζ directions, a numerical approximation to Eq. (4) may be expressed in the semidiscrete conservation law form given by

$$\begin{aligned} &(\hat{Q}_{j,k,l})_\tau + (\hat{E}_{j+1/2,k,l} - \hat{E}_{j-1/2,k,l}) \\ &\quad + (\hat{F}_{j,k+1/2,l} - \hat{F}_{j,k-1/2,l}) + (\hat{G}_{j,k,l+1/2} - \hat{G}_{j,k,l-1/2}) = 0 \end{aligned} \qquad (7)$$

where $\hat{E}$, $\hat{F}$, $\hat{G}$ are numerical or representative fluxes at the bounding sides of the cell for which discrete conservation is considered and $\hat{Q}_{j,k,l}$ is the representative conserved quantity (the numerical approximation to $\bar{Q}$), considered conveniently to be the centroidal or cell-average value. The half-integer subscripts denote cell sides, and the integer subscripts the cell itself or its centroid. In Fig. 1, the eight vertices of one computational hexahedral cell are identified by numerals 1–8. These must be associated with the appropriate j, k, l triplets:

$$1 \equiv j-½, k-½, l-½$$
$$2 \equiv j+½, k-½, l-½$$
$$3 \equiv j-½, k+½, l-½$$
$$4 \equiv j-½, k-½, l+½$$
$$5 \equiv j+½, k+½, l-½$$
$$6 \equiv j-½, k+½, l+½$$
$$7 \equiv j+½, k-½, l+½$$
$$8 \equiv j+½, k+½, l+½ \tag{8}$$

In the following, subscripts easily understood by implication will be dropped for brevity.

The semidiscrete conservation law given by Eq. (7) may be regarded as representing a finite-volume discretization if the following associations are made:

$$\hat{Q}_{j,k,l} = QV_{j,k,l} \tag{9}$$

where V is the volume of the cell under consideration;

$$(\xi_{x,y,z}/J)_{j\pm½} = n_{x,y,z}\{(k-½, l-½),\ (k+½, l-½),$$
$$(k+½, l+½), (k-½, l+½)\}_{j\pm½}$$
$$(\eta_{x,y,z}/J)_{k\pm½} = n_{x,y,z}\{(j-½, l-½),\ (j-½, l+½),$$
$$(j+½, l+½), (j+½, l-½)\}_{k\pm½}$$
$$(\zeta_{x,y,z}/J)_{l\pm½} = n_{x,y,z}\{(j-½, k-½),\ (j+½, k-½),$$
$$(j+½, k+½), (j-½, k+½)\}_{l\pm½} \tag{10}$$

In the above, $n_{x,y,z}$ are the components of the representative normals to the surface formed by the four points a, b, c, d implied in $n_{x,y,z}$ (a, b, c, d). Four points do not necessarily lie in one plane, and therefore the components $n_{x,y,z}$ refer to representative values for an equivalent single plane.

The evaluation of the volume and metrics (cell-face normals) are now presented in the following subsections. The evaluation of the representative flux is presented in the next major section.

Computation of Cell Volume

First, the volume of a tetrahedron denoted by its vertices a, b, c, d is evaluated from

$$V^{\text{tet}}(a,b,c,d) = |x_a[y_b(z_c-z_d) - y_c(z_b-z_d) + y_d(z_d-z_c)]$$
$$-x_b[y_a(z_c-z_d) - y_c(z_a-z_d) + y_d(z_a-z_c)]$$
$$+x_c[y_a(z_b-z_d) - y_b(z_a-z_d) + y_d(z_a-z_b)]$$
$$-x_d[y_a(z_b-z_c) - y_b(z_a-z_c) + y_c(z_a-z_b)]|/6.0 \tag{11}$$

Then, referring to Fig. 1 again, the volume of the hexahedron is computed as a sum of the six individual tetrahedrons that constitute it.

$$V = V^{\text{tet}}(1,2,5,8) + V^{\text{tet}}(1,2,8,7)$$
$$+ V^{\text{tet}}(1,3,8,5) + V^{\text{tet}}(1,4,7,8)$$
$$+ V^{\text{tet}}(1,4,8,6) + V^{\text{tet}}(1,3,6,8) \tag{12}$$

It is of interest to note that such a formula will result in the proper evaluation of volume even when some of the faces of the hexahedron collapse to a line or a point.

Computation of Cell-Face Metrics (Normals)

In Eq. (10), cell-face normals were introduced. Each cell face is identified by four vertices, not all of which are necessarily on a single plane (three points being sufficient for defining a plane). The present approach allows for this and also for some of the faces to collapse to an edge or even a point. Computationally, a face is always identified by its four vertices a, b, c, d expressed in the j, k, l subscript system. Physically, some or all of the four vertices may lie at the same x, y, z location.

The cell-face normals are evaluated as

$$n_x(a,b,c,d) = (\mathrm{d}y_{ba}\mathrm{d}z_{cb} - \mathrm{d}y_{cb}\mathrm{d}z_{ba})/2$$
$$+ (\mathrm{d}y_{dc}\mathrm{d}z_{ad} - \mathrm{d}y_{ad}\mathrm{d}z_{dc})/2$$
$$n_y(a,b,c,d) = (\mathrm{d}z_{ba}\mathrm{d}x_{cb} - \mathrm{d}z_{cb}\mathrm{d}x_{ba})/2$$
$$+ (\mathrm{d}z_{dc}\mathrm{d}x_{ad} - \mathrm{d}x_{ad}\mathrm{d}x_{dc})/2$$
$$n_z(a,b,c,d) = (\mathrm{d}x_{ba}\mathrm{d}y_{cb} - \mathrm{d}x_{cb}\mathrm{d}y_{ba})/2$$
$$+ (\mathrm{d}x_{dc}\mathrm{d}y_{ad} - \mathrm{d}x_{ad}\mathrm{d}y_{dc})/2 \tag{13}$$

where

$$\mathrm{d}s_{12} = s_1 - s_2 \tag{14}$$

where, in turn, s corresponds to x, y, or z and 1 and 2 correspond to a, b, c, or d. The first term in each of the definitions is, respectively, one-half of the x, y, or z component of the cross product of the vector from a to b with the vector from b to c. The second term in each definition is correspondingly one-half of the x,y, or z component of the cross product of the vector from c to d with the vector from d to a. A cross product of two vectors lies along the direction of the normal to the two vectors. In the present situation, the two vectors are connected. Therefore, such a normal also defines the direction of the normal to the plane containing the two vectors. One-half of the cross product of connected vectors also has a magnitude equal to the area of the three-dimensional triangular planar shape defined by the two vectors. Thus, while $\xi_{x,y,z}/J$, $\eta_{x,y,z}/J$, $\zeta_{x,y,z}/J$ define the x, y, z components of the normals (not unit normals) to the local tangent plane to the constant ξ, η, ζ surfaces, respectively, the associated quantities $(n_{x,y,z})_{j,k,l}$ define the components of the normals to the local constant j, k, l planes.

TVD Discretization

In the last section, the numerical or representative fluxes $\hat{E}$, $\hat{F}$, $\hat{G}$ were introduced. These fluxes are so named because they approximate the real fluxes $\bar{E}$, $\bar{F}$, $\bar{G}$ to the required order of accuracy. The actual fluxes appearing in the governing partial differential equations depend on the metrics $\xi_{x,y,z}/J$, $\eta_{x,y,z}/J$, $\zeta_{x,y,z}/J$ and, correspondingly, the numerical fluxes are allowed to depend on the numerical metrics (the cell-face normals). In the last section, the link between the metrics and the components of the cell normals was pointed out, but the numerical flux was not defined there. The latter task is the subject of this section.

The approach chosen is to employ an upwind-biased scheme in such a fashion as to essentially eliminate numerical or spurious (unphysical) oscillations while, at the same time, achieving high accuracy. In order to describe this type of discretization, the underlying upwind scheme used is first described in terms of the corresponding approximate Riemann solver, and then it is shown how high accuracy and the TVD property are built in. More details on these and related topics may be found in Refs. 5 and 6 and in references cited therein.

Roe's Approximate Riemann Solver

The Riemann solver is a mechanism to divide the flux difference between neighboring states (between Q_m and Q_{m+1}, e.g.) into component parts associated with each wave field. These, in turn, can be divided into those that correspond to positive and negative wave speeds. When the numerical flux is computed at the cell face at $m+½$ using various combinations of fluxes and positive and negative flux differences in the finite-volume formulation, only the cell-face normals defined at $m+½$ will be used in all the terms contributing to that representative flux. The actual fluxes $\bar{E}$, $\bar{F}$, $\bar{G}$, when evaluated with the metrics equated to cell-face normals, can all be written in the same functional form given by

$$\bar{E}, \bar{F}, \bar{G} = f(Q, n_x, n_y, n_z) = f(Q, N) \tag{15}$$

where the appropriate values of n_x, n_y, n_z are used and N denotes the set of those normals. Using such notation, it is possible to present the necessary algebra very concisely.

Let the Jacobian matrix of the flux f with respect to the dependent variables Q be denoted by $\partial f/\partial Q$. This Jacobian can also be called the coefficient matrix. Let the eigenvalues of the coefficient matrix be denoted by λ^i and the corresponding left and right eigenvectors by ℓ^i and r^i, respectively. The matrix formed by the left eigenvectors as its rows is then called the left eigenvector matrix L and the matrix of right eigenvectors comprising right eigenvectors as its columns is R. For present purposes, an orthonormal set of left and right eigenvectors is chosen. This implies that $LR = RL = I$, the identity matrix. In the above, the superscript i has been used to denote the association of the ith eigenvalue with its corresponding eigenvector. Each eigenvalue is also associated with its own wave field.

The underlying upwind scheme is based on Roe's approximate Riemann solver.[7] In this approach, cell-interface values of density, velocities, and enthalpy $\{h = \gamma p/[(\gamma-1)\rho] + (u^2+v^2+w^2)/2\}$ are computed using a special averaging procedure:

$$\rho_{m+½} = \sqrt{\rho_m}\,\sqrt{\rho_m+1}$$

$$(u,v,w)_{m+½} = \frac{(u,v,w)_{m+1}\sqrt{\rho_{m+1}} + (u,v,w)_m\sqrt{\rho_m}}{\sqrt{\rho_{m+1}} + \sqrt{\rho_m}}$$

$$h_{m+½} = \frac{h_{m+1}\sqrt{\rho_{m+1}} + h_m\sqrt{\rho_m}}{\sqrt{\rho_{m+1}} + \sqrt{\rho_m}} \tag{16}$$

where $m = j$ or k or l. From the above, the speed of sound can be computed from

$$c_{m+½} = \sqrt{\{h_{m+½} - (u^2+v^2+w^2)/2\}(\gamma-1)} \tag{17}$$

Knowing $(u, v, w, c)_{m+½}$, the eigenvalues and orthonormal set of left and right eigenvectors corresponding to a cell face can be computed. These may be denoted by

$$\lambda^i_{m+½} = \lambda^i_{m+½}(Q_{m+½}, N_{m+½})$$

$$\ell^i_{m+½} = \ell^i_{m+½}(Q_{m+½}, N_{m+½})$$

$$r^i_{m+½} = r^i_{m+½}(Q_{m+½}, N_{m+½}) \tag{18}$$

At each cell face, the positive and negative projections of the eigenvalues may be defined by

$$\lambda^{i\pm} = \frac{(\lambda^i_{m+½} \pm |\lambda^i_{m+½}|)}{2}, \qquad i = 1,\ldots,5 \tag{19}$$

In order to help Roe's Riemann solver avoid expansion shocks, only at sonic rarefactions $[\lambda^i(Q_m, N_{m+½}) < 0 < \lambda^i(Q_{m+1}, N_{m+½})]$, the corresponding positive and negative projections are redefined as

$$\lambda^{i\pm}_{m+½} = \lambda^{i\pm}_{m+½} \pm \frac{[\lambda^i(Q_{m+1}, N_{m+½}) - \lambda^i(Q_m, N_{m+½})]}{4} \tag{20}$$

For the sake of completeness, detailed formulas for the eigenvalues and the eigenvector matrices are now presented. Defining the contravariant velocity by

$$\bar{U} = n_x u + n_y v + n_z w \tag{21}$$

the eigenvalues are given by

$$\lambda^1 = \bar{U} - c\sqrt{n_x^2+n_y^2+n_z^2}$$

$$\lambda^{2,3,4} = \bar{U}$$

$$\lambda^5 = \bar{U} + c\sqrt{n_x^2+n_y^2+n_z^2} \tag{22}$$

Defining

$$\hat{n}_{x,y,z} = n_{x,y,z}/\sqrt{n_x^2+n_y^2+n_z^2} \tag{23}$$

and

$$\theta = (u^2+v^2+w^2)/2 \tag{24}$$

the left and the right eigenvector matrices are given in Tables 1 and 2, respectively.

High-Accuracy TVD Schemes

Upwind-biased schemes of varying accuracies can be constructed using the basic ingredients given in the last subsection. Here, a family of schemes is presented based on the *preprocessing* approach.[6] Some convenient variables are now defined as an intermediate step before defining the numerical flux corresponding to a high-accuracy TVD scheme. First, parameters denoted as α are defined. These provide a measure of the change in dependent variables across the corresponding wave family and therefore measure the slope between neighboring states. In the following, the superscript i corresponds, as usual, to the ith eigenvalue and ith eigenvector. The subscripts 1–3 are just labels to differentiate between the three different types of α parameters.

$$\alpha^i_{1,m+½} = \ell^i_{m+½}(Q_m - Q_{m-1})$$

$$\alpha^i_{2,m+½} = \ell^i_{m+½}(Q_{m+1} - Q_m)$$

$$\alpha^i_{3,m+½} = \ell^i_{m+½}(Q_{m+2} - Q_{m+1}) \tag{25}$$

Next, the slope-limited values are defined by

$$\bar{\tilde{\alpha}}^i_{1,m+½} = \text{minmod}\,[\alpha^i_{1,m+½}, b\alpha^i_{2,m+½}]$$

$$\tilde{\alpha}^i_{2,m+½} = \text{minmod}\,[\alpha^i_{2,m+½}, b\alpha^i_{1,m+½}]$$

$$\tilde{\tilde{\alpha}}^i_{2,m+½} = \text{minmod}\,[\alpha^i_{2,m+½}, b\alpha^i_{3,m+½}]$$

$$\bar{\tilde{\alpha}}^i_{3,m+½} = \text{minmod}\,[\alpha^i_{3,m+½}, b\alpha^i_{2,m+½}] \tag{26}$$

In the above, the compression parameter b is to be taken as the following function of the accuracy parameter ϕ, which is explained shortly.

$$b = (3-\phi)/(1-\phi) \tag{27}$$

Table 1 The left eigenvector matrix L

$\left[\frac{\gamma-1}{c}\right]\Big/\sqrt{2}$	$\left[\frac{\gamma-1}{c}\theta+\hat{n}_x u + \hat{n}_y v+\hat{n}_z w\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}u - \hat{n}_x\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}v - \hat{n}_y\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}w - \hat{n}_z\right]\Big/\sqrt{2}$
$-\frac{\gamma-1}{c}\hat{n}_y$	$-\frac{\gamma-1}{c}\hat{n}_y\theta+\hat{n}_y c - \hat{n}_z u+\hat{n}_x w$	$\frac{\gamma-1}{c}\hat{n}_y u + \hat{n}_z$	$\frac{\gamma-1}{c}\hat{n}_y v$	$\frac{\gamma-1}{c}\hat{n}_y w - \hat{n}_x$
$-\frac{\gamma-1}{c}\hat{n}_z$	$-\frac{\gamma-1}{c}\hat{n}_z\theta+\hat{n}_z c + \hat{n}_y u-\hat{n}_x v$	$\frac{\gamma-1}{c}\hat{n}_z u - \hat{n}_y$	$\frac{\gamma-1}{c}\hat{n}_z v + \hat{n}_x$	$\frac{\gamma-1}{c}\hat{n}_z w$
$-\frac{\gamma-1}{c}\hat{n}_x$	$-\frac{\gamma-1}{c}\hat{n}_x\theta+\hat{n}_x c + \hat{n}_z v-\hat{n}_y w$	$\frac{\gamma-1}{c}\hat{n}_x u$	$\frac{\gamma-1}{c}\hat{n}_x v - \hat{n}_z$	$\frac{\gamma-1}{c}\hat{n}_x w + \hat{n}_y$
$\left[\frac{\gamma-1}{c}\right]\Big/\sqrt{2}$	$\left[\frac{\gamma-1}{c}\theta+\hat{n}_x u - \hat{n}_y v-\hat{n}_z w\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}u + \hat{n}_x\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}v + \hat{n}_y\right]\Big/\sqrt{2}$	$\left[-\frac{\gamma-1}{c}w + \hat{n}_z\right]\Big/\sqrt{2}$

Table 2 The right eigenvector matrix R

$\left[\frac{\theta}{c}+\frac{c}{\gamma-1}-\hat{n}_x u - \hat{n}_y v-\hat{n}_z w\right]\Big/\sqrt{2}$	$\frac{\theta}{c}\hat{n}_y + \hat{n}_z u - \hat{n}_x w$	$\frac{\theta}{c}\hat{n}_z - \hat{n}_y u + \hat{n}_x v$	$\frac{\theta}{c}\hat{n}_x - \hat{n}_z v + \hat{n}_y w$	$\left[\frac{\theta}{c}+\frac{c}{\gamma-1}+\hat{n}_x u + \hat{n}_y v+\hat{n}_z w\right]\Big/\sqrt{2}$
$\left[\frac{1}{c}\right]\Big/\sqrt{2}$	$\frac{\hat{n}_y}{c}$	$\frac{\hat{n}_z}{c}$	$\frac{\hat{n}_x}{c}$	$\left[\frac{1}{c}\right]\Big/\sqrt{2}$
$\left[\frac{u}{c}-\hat{n}_x\right]\Big/\sqrt{2}$	$\frac{u}{c}\hat{n}_y+\hat{n}_z$	$\frac{u}{c}\hat{n}_z-\hat{n}_y$	$\frac{u}{c}\hat{n}_x$	$\left[\frac{u}{c}+\hat{n}_x\right]\Big/\sqrt{2}$
$\left[\frac{v}{c}-\hat{n}_y\right]\Big/\sqrt{2}$	$\frac{v}{c}\hat{n}_y$	$\frac{v}{c}\hat{n}_z+\hat{n}_x$	$\frac{v}{c}\hat{n}_x-\hat{n}_z$	$\left[\frac{v}{c}+\hat{n}_y\right]\Big/\sqrt{2}$
$\left[\frac{w}{c}-\hat{n}_z\right]\Big/\sqrt{2}$	$\frac{w}{c}\hat{n}_y-\hat{n}_x$	$\frac{w}{c}\hat{n}_z$	$\frac{w}{c}\hat{n}_x+\hat{n}_y$	$\left[\frac{w}{c}+\hat{n}_z\right]\Big/\sqrt{2}$

The "minmod" slope-limiter operator is

$$\text{minmod}\,[x,y] = \text{sign}\,(x)\,\max\,[0, \min\,\{\,|x|,\, y\,\text{sign}\,(x)\,\}] \tag{28}$$

In Eq. (7), numerical fluxes $\hat{E}$, $\hat{F}$, $\hat{G}$ were introduced. Based on the concise notation of using f to represent either $\bar{E}$, $\bar{F}$, or $\bar{G}$, let us use $\hat{f}$ to denote the numerical fluxes $\hat{E}$, $\hat{F}$, or $\hat{G}$. A family of TVD schemes can be written down as follows in terms of the previously defined α parameters (with the subscript $m+½$ dropped from these for convenience):

$$\begin{aligned}\hat{f}_{m+½} &= h_{m+½}\\ &+ \sum_i \left(\frac{1+\phi}{4}\tilde{\alpha}_2^i + \frac{1-\phi}{4}\tilde{\tilde{\alpha}}_1^i\right)\lambda_{m+½}^{i+} r_{m+½}^i\\ &- \sum_i \left(\frac{1+\phi}{4}\tilde{\tilde{\alpha}}_2^i + \frac{1-\phi}{4}\tilde{\alpha}_3^i\right)\lambda_{m+½}^{i-} r_{m+½}^i\end{aligned} \tag{29}$$

The first term on the right-hand side of Eq. (29) defines a first-order numerical flux and is constructed from

$$\begin{aligned}h_{m+½} &= ½[f(Q_{m+1}, N_{m+½}) + f(Q_m, N_{m+½})]\\ &- ½\left[\sum_i \left(\lambda_{m+½}^{i+} - \lambda_{m+½}^{i-}\right)\alpha_2^i r_{m+½}^i\right]\\ &= f(Q_m, N_{m+½}) + \sum_i \lambda_{m+½}^{i-}\alpha_2^i r_{m+½}^i\\ &= f(Q_{m+1}, N_{m+½}) - \sum_i \lambda_{m+½}^{i+}\alpha_2^i r_{m+½}^i\end{aligned} \tag{30}$$

The remaining terms on the right-hand side of Eq. (29) define correction terms that upgrade the accuracy. For use in the next subsection, define

$$\mathrm{d}f_{m+½}^{\,i\pm} = \lambda_{m+½}^{i\pm}\alpha_{2,m+½}^i r_{m+½}^i \tag{31}$$

It is interesting to note that in all the above formulas used to define the numerical flux at $m+½$, the eigenvectors and eigenvalues are only necessary at the corresponding cell interface. Therefore, the only geometry information used corresponds to the cell-face normals at $m+½$. The solution variables Q are sampled between the centroidal points $m-1$, m, $m+1$, $m+2$ when the various α parameters are defined.

The parameter ϕ defines schemes of varying accuracy. The notations $\tilde{\alpha}^i$ and $\tilde{\tilde{\alpha}}^i$ have been used to define slope-limited values of the α parameters. If these are replaced by their unlimited values, the truncation error of the resulting schemes (in one-dimensional steady-state problems on uniform grids) is given by

$$TE = -\left(\frac{\phi-⅓}{4}\right)(\Delta x)^2 \frac{\partial f}{\partial Q}\frac{\partial^3 Q}{\partial x^3} \tag{32}$$

Here, the truncation error refers to the difference between the centroidal value of the numerical solution and the average value of the exact solution in that cell. The choice of $\phi = ⅓$ results in a TVD scheme based on an underlying third-order scheme. The choice of $\phi = -1$ results in a TVD scheme based on the fully upwind second-order-accurate formulation. Fromm's scheme arises when $\phi = 0$.

TVD Schemes and Diagonal Dominance

In the next section, a procedure is presented to solve the finite-difference equations resulting from the TVD discretization of the space-differencing terms. In supersonic zones, the method reduces to a simple marching scheme, while in the subsonic zones, it becomes a relaxation approach, and both forward and backward sweeps are employed along the marching direction. In order for such a relaxation approach to be stable, a sufficient condition is the diagonal dominance of the underlying finite-difference scheme. This diagonal dominance can be shown to exist for TVD discretizations. For more details, the reader is referred to Ref. 8.

The Solution Procedure

This section begins by considering an implicit time discretization coupled with the TVD space discretization discussed earlier in terms of the corresponding numerical flux terms.

$$\frac{\hat{Q}^{n+1}-\hat{Q}^{n}}{\Delta\tau}+(\hat{E}_{j+\frac{1}{2}}-\hat{E}_{j-\frac{1}{2}})^{n+1}$$

$$+(\hat{F}_{k+\frac{1}{2}}-\hat{F}_{k-\frac{1}{2}})^{n+1}+(\hat{G}_{l+\frac{1}{2}}-\hat{G}_{l-\frac{1}{2}})^{n+1}=0 \qquad (33)$$

Here, n is the index in time and $\Delta\tau$ is the time step. In what follows, the linearization of the above nonlinear set of finite-difference equations will be considered. Then the algebraic solution procedure will be simplified by approximately factorizing the implicit operator in the crossflow plane (which is a plane in only computational coordinates—constant j plane). The marching direction is along j. The scheme will be further specialized for the two cases of supersonic and subsonic velocity components in the marching direction.

Linearization

Let Eq. (33) be linearized about a known state $Q-q^s$, using a Newton procedure to obtain a better approximation q^{s+1} to Q^{n+1}. Here, s is a subiteration index. Defining

$$\Delta^s q=q^{s+1}-q^s, \qquad \Delta_j\hat{E}=\hat{E}_{j+\frac{1}{2}}-\hat{E}_{j-\frac{1}{2}}$$

$$\Delta_k\hat{F}=\hat{F}_{k+\frac{1}{2}}-\hat{F}_{k-\frac{1}{2}}, \qquad \Delta_l\hat{G}=\hat{G}_{l+\frac{1}{2}}-\hat{G}_{l-\frac{1}{2}} \qquad (34)$$

the Newton procedure can be described by

$$\left[\frac{V}{\Delta\tau}I+\frac{\partial}{\partial q}(\Delta_j\hat{E}+\Delta_k\hat{F}+\Delta_l\hat{G})\right]\Delta^s q$$

$$=-\left[\frac{V}{\Delta\tau}(q^s-Q^n)+\Delta_j\hat{E}(q^s)+\Delta_k\hat{F}(q^s)+\Delta_l\hat{G}(q^s)\right] \qquad (35)$$

Next, the left-hand side is simplified by defining an approximation to the true linearization. Toward this goal, only a first-order-accurate scheme (used on the first-order numerical flux h) is used for the left-hand side, while the full high-accuracy scheme is included on the right-hand side. Even so, when the subiterations converge, the right-hand side is satisfied to the desired degree. Next, it is assumed that the eigenvalues and eigenvectors are not functions of q. Finally, it is observed that

$$\sum_{m=j,k,l} h_{m+\frac{1}{2}}-h_{m-\frac{1}{2}}$$

$$=\sum_{m=j,k,l}\sum_i \mathrm{d}f^{\,i+}_{m-\frac{1}{2}}+\sum_{m=j,k,l}\sum_i \mathrm{d}f^{\,i-}_{m+\frac{1}{2}} \qquad (36)$$

because, in expanding Eq. (36) using Eq. (30), it is found that

$$\sum_{m=j,k,l}[(n_x)_{m+\frac{1}{2}}-(n_x)_{m-\frac{1}{2}}]=0$$

$$\sum_{m=j,k,l}[(n_y)_{m+\frac{1}{2}}-(n_y)_{m-\frac{1}{2}}]=0$$

$$\sum_{m=j,k,l}[(n_z)_{m+\frac{1}{2}}-(n_z)_{m-\frac{1}{2}}]=0 \qquad (37)$$

when the cell-face normals are evaluated using the formulas given in Eq. (13). Using the above, Eq. (35) is simplified to

$$\frac{V}{\Delta\tau}\Delta^s q$$

$$+A^+_{j-\frac{1}{2}}(\Delta^s q_j-\Delta^s q_{j-1})+A^-_{j+\frac{1}{2}}(\Delta^s q_{j+1}-\Delta^s q_j)$$

$$+B^+_{k-\frac{1}{2}}(\Delta^s q_k-\Delta^s q_{k-1})+B^-_{k+\frac{1}{2}}(\Delta^s q_{k+1}-\Delta^s q_k)$$

$$+C^+_{l-\frac{1}{2}}(\Delta^s q_l-\Delta^s q_{l-1})+C^-_{l+\frac{1}{2}}(\Delta^s q_{l+1}-\Delta^s q_l)$$

$$=\text{right-hand side of Eq. (35)} \qquad (38)$$

where

$$A^{\pm}_{j\pm\frac{1}{2}}=R_{j\pm\frac{1}{2}}\Lambda^{\pm}_{j\pm\frac{1}{2}}L_{j\pm\frac{1}{2}}$$

$$B^{\pm}_{k\pm\frac{1}{2}}=R_{k\pm\frac{1}{2}}\Lambda^{\pm}_{k\pm\frac{1}{2}}L_{k\pm\frac{1}{2}}$$

$$C^{\pm}_{l\pm\frac{1}{2}}=R_{l\pm\frac{1}{2}}\Lambda^{\pm}_{l\pm\frac{1}{2}}L_{l\pm\frac{1}{2}} \qquad (39)$$

Here,

$$\Lambda^{\pm}=(\Lambda+|\Lambda|)/2 \qquad (40)$$

in which Λ is the diagonal matrix whose diagonal elements are λ^i, and $|\Lambda|$ is the diagonal matrix whose diagonal elements are $|\lambda^i|$.

Planar Gauss-Seidel Relaxation

Even after the many simplifications leading to Eq. (38), it is obvious that more algebraic simplification is needed before a computationally feasible and efficient solution procedure is obtained. This is because Eq. (38) signifies a system of equations that links every point j, k, l with its six neighbors $j+1$, $j-1$, $k+1$, $k-1$, $l+1$, and $l-1$ in such a fashion that the left-hand side of Eq. (38), when considered for all grid points, is a huge (even though sparse) matrix whose bandwidth is also very large. Of course, for supersonic flows, a fully upwind difference approximation arises in the j direction, and the dimensionality is reduced because the left-hand side does not link j with $j+1$. However, with the present aim of developing a method for subsonic pockets also, it is necessary to consider the case in which j is linked with both its neighbors, $j-1$ and $j+1$. In such a case, a direct Gaussian elimination procedure for the matrix system of equations would be unacceptably expensive. Therefore, instead of a direct elimination procedure, an efficient relaxation solution to Eq. (38) is sought. A planar Gauss-Seidel procedure is chosen by retaining all terms of the left-hand side except the off-diagonal terms in j (those terms that multiply $\Delta^s q_{j\pm1}$). That such a procedure will be stable for TVD discretizations was discussed in a previous section and in the references cited there.

The planar Gauss-Seidel procedure can be written as

$$\begin{aligned}&\left[\frac{I}{\Delta\tau}+\frac{1}{V}A^{+}_{j-1/2}+\frac{1}{V}A^{-}_{j+1/2}\right]\Delta^{s}q_{j}\\&+\frac{1}{V}B^{+}_{k-1/2}(\Delta^{s}q_{k}-\Delta^{s}q_{k-1})\\&+\frac{1}{V}B^{-}_{k+1/2}(\Delta^{s}q_{k+1}-\Delta^{s}q_{k})\\&+\frac{1}{V}C^{+}_{l-1/2}(\Delta^{s}q_{l}-\Delta^{s}q_{l-1})\\&+\frac{1}{V}C^{-}_{l+1/2}(\Delta^{s}q_{l+1}-\Delta^{s}q_{l})\\&=\frac{1}{V}[\text{right-hand side of Eq. (35)}]\end{aligned}\tag{41}$$

Denoting

$$\hat{A}=\frac{I}{\Delta\tau}+\frac{1}{V}A^{+}_{j-1/2}+\frac{1}{V}A^{-}_{j+1/2}\tag{42}$$

Eq. (41) can be rewritten as

$$\begin{aligned}&\left[I+\frac{1}{V}\hat{A}^{-1}\left\{B^{+}_{k-1/2}\Delta_{k-1/2}+B^{-}_{k+1/2}\Delta_{k+1/2}\right.\right.\\&\left.\left.+C^{+}_{l-1/2}\Delta_{l-1/2}+C^{-}_{l+1/2}\Delta_{l+1/2}\right\}\right]\Delta^{s}q\\&=\frac{1}{V}\hat{A}^{-1}[\text{right-hand side of Eq. (35)}]\end{aligned}\tag{43}$$

Of course, when the relaxation cycles denoted by superscript s converge to the desired extent, $\Delta^{s}q=0$, and the full accurate formulas of the right-hand side will be satisfied to a corresponding degree.

Approximate Factorization in the Plane

While Eq. (43) defines an algebraic set of equations whose dimensionality is one order less than that of Eq. (41), it is still too huge to be tackled by an elimination algorithm. Therefore, the dimensionality will now be further reduced by approximately factorizing the left-hand side of Eq. (43) to result in

$$\begin{aligned}&[I+(1/V)\hat{A}^{-1}\{B^{+}_{k-1/2}\Delta_{k-1/2}+B^{-}_{k+1/2}\Delta_{k+1/2}\}]\\&[I+(1/V)\hat{A}^{-1}\{C^{+}_{l-1/2}\Delta_{l-1/2}+C^{-}_{l+1/2}\Delta_{l+1/2}\}]\Delta^{s}q\\&=(1/V)\hat{A}^{-1}[\text{right-hand side of Eq. (35)}]\end{aligned}\tag{44}$$

The actual sequence of steps to solve Eq. (44) can be chosen so that $\hat{A}^{-1}$ need not actually be computed and only $\hat{A}$ is needed. For this purpose, the following equations are solved in the order given.

$$\begin{aligned}&[\hat{A}+(1/V)\{B^{+}_{k-1/2}\Delta_{k-1/2}+B^{-}_{k+1/2}\Delta_{k+1/2}\}]\tilde{q}\\&=(1/V)[\text{right-hand side of Eq. (35)}]\end{aligned}\tag{45}$$

$$\begin{aligned}&[\hat{A}+(1/V)\{C^{+}_{l-1/2}\Delta_{l-1/2}+C^{-}_{l+1/2}\Delta_{l+1/2}\}]\Delta^{s}q\\&=\hat{A}\tilde{q}\end{aligned}\tag{46}$$

with $\tilde{q}$ being a temporary storage variable.

Let the solution procedure developed in Eqs. (45-46) be summarized for just one constant j plane. Equation (45) must be solved for all k-varying lines (for all l). Then Eq. (46) must be solved for all l-varying lines (for all values of k). However, each k-varying or l-varying line is associated with only a one-dimensional block-tridiagonal system of algebraic equations whose block matrices are 5 × 5. These two steps constitute only one cycle of the Gauss-Seidel iterations and that for only one constant j plane. The planar Gauss-Seidel procedure requires that one constant j plane be updated at a time. When the neighboring j plane is updated next, the latest available values of the update variables q are used in the right- and left-hand sides. The j sweep strategy will be specialized for supersonic and subsonic flow regions in what follows.

Programming Notes

Grid information is stored at two planes (grid planes 2 and 3), which describe the j boundaries ($j-1/2, j+1/2$) of one plane of cells. Denote the centroids of these cells as solution plane 3. Array storage is provided for dependent variable planes (solution planes) 1-5. As the solution is marched, the contents of the grid-plane and solution-plane arrays are updated by replacing them with neighboring values or by the planar Gauss-Seidel algorithm.

The very first marching sweep is considered now. First, the two grid planes and the dependent variables at solution planes 1 and 2 are initialized. The focus is on updating the solution at plane 3. The solution values at planes 3-5 are set equal to the values at solution plane 2. After one or more subiterations for solution plane 3, attention is shifted to the next j plane. Grid plane 2 is replaced with the contents of grid plane 3. Grid-plane 3 nodal values are stored on auxiliary storage for later use. New values for grid plane 3 are generated by grid-generation procedures or read in from auxiliary storage initialized previously. Similarly, solution plane 3 is saved on auxiliary storage for subsequent processing. Solution plane 1 is replaced by contents of solution plane 2, and plane 2 is then replaced by contents of plane 3. Solution planes 4 and 5 are set to the values at plane 3, and the marching proceeds.

If more than one subiteration is to be performed in the first marching sweep, the grid information is not updated for the subsequent subiterations. Solution planes 4 and 5 are reset to values at solution plane 3 after the previous subiteration, and the next subiteration is processed. Solution-plane 3 values are not set to solution-plane 2 values for the second and subsequent subiterations.

For fully supersonic flows, a fully upwind, not flux-limited, differencing scheme is used. Thus, the values set for solution planes 4 and 5 are not actually used at all. Forward marching is enough. Even a first-order upwind scheme in the j direction and one subiteration per marching plane are also often enough. A small value is input for the reciprocal of time step. Accuracy of approximate factorization for any time-step size is maintained due to reasonable marching-step size (distance between j grid planes).

Subsonic regions could develop as a result of gradual compression (e.g., around canopies) or abrupt transition through a shock wave (e.g., in front of a blunt-nosed object in an oncoming supersonic flow). In such regions, a larger value is chosen for the reciprocal of time step. The solution is marched forward using one or more (usually a maximum of two) subiterations by conforming to the procedure outlined above for the first marching sweep. Then, a backward-marching sweep (or even another forward-marching sweep) is performed. For all sweeps (forward or backward) after the first, solution planes are filled with previous sweep solution values before they are updated using subiterations. Shifted replacements of solution-plane values of dependent variables are not used. For subsonic regions (subsonic pockets in supersonic flow), a TVD formulation of the desired accuracy is used, enabling even strong shocks to be captured routinely.

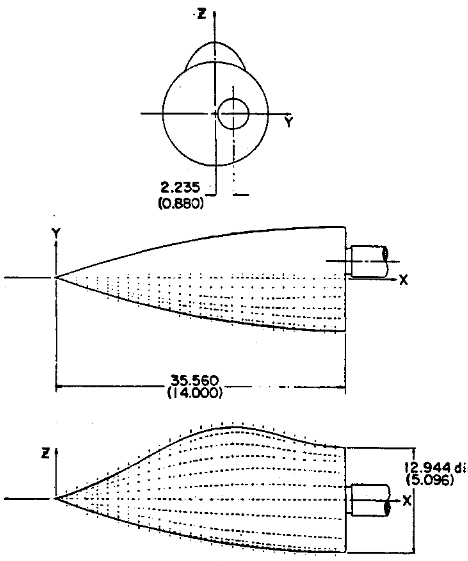

Fig. 2a Forebody geometry.

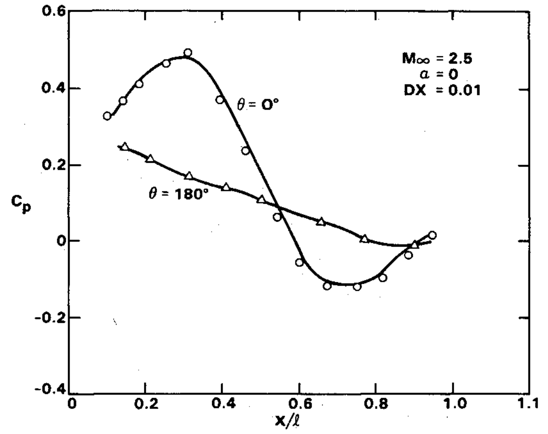

Fig. 2b Axial pressure distribution.

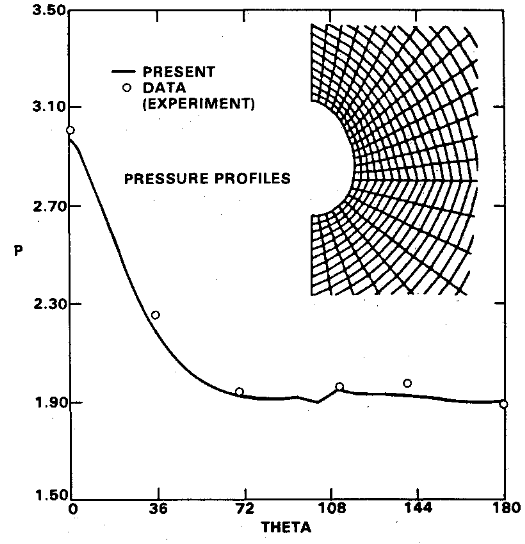

Fig. 2c Circumferential pressure distribution at $x/\ell = 0.22$.

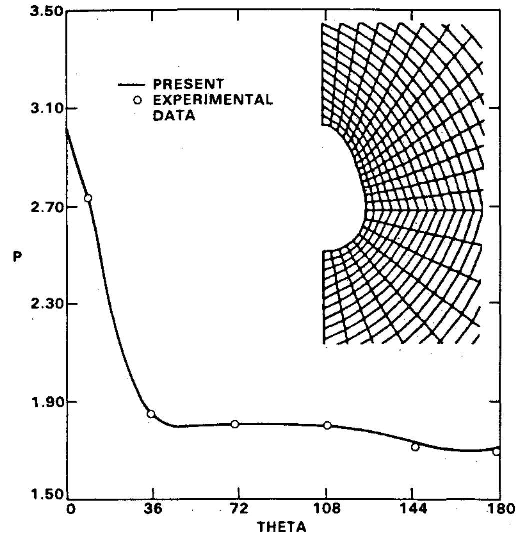

Fig. 2d Circumferential pressure distribution at $x/\ell = 0.34$.

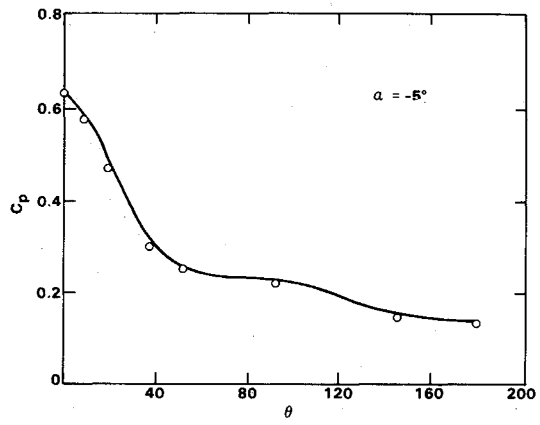

Fig. 2e Circumferential pressure distribution at $M_\infty = 1.7$, $x/\ell = 0.278$.

For very small pockets of subsonic flow caused by gradual compression, one forward sweep followed by one reverse sweep is enough. Even the reverse sweep is usually redundant in this case. For larger subsonic zones, a few (tens) of sweeps usually suffice. Residues are monitored for convergence.

Boundary-Point Treatment

Only an outline of the boundary-point treatment will be presented here due to lack of space. The boundary method used is fully compatible with the interior point differencing. It is based on considering a Riemann initial and boundary-value problem at the boundary to construct the boundary-point discretization. In this, it is similar in spirit to the correspondence between interior point discretization and the Riemann initial-value problem. The implementation is specifically tailored to approximately factored implicit schemes. Linear boundary conditions (such as surface tangency) are exactly satisfied after every marching step. Corner points are also properly treated. A brief description of the type of boundary-condition techniques used here can also be found in Ref. 5. Another approach to boundary-condition procedures that can be applied to implicit schemes for the Euler equations is presented in Ref. 9.

Computational Examples

The preceding sections have described an Euler marching technique for accurate computations (EMTAC), and many computational results obtained using the EMTAC code are now presented. The first set of results is for an analytically defined forebody geometry, and these results are compared with experimental data. The second set of results is for a realistic fighter configuration, and the results are compared

Table 3 Comparison of potential, Euler, and experimental data for C_L, C_D, and C_L/C_D

	SIMP	EMTAC	DATA
C_L	0.30588	0.3017	0.303
C_D	0.032458 + 0.013 = 0.045458	0.03433 + 0.013 = 0.04733	0.0475
C_L/C_D	6.72	6.38	6.42

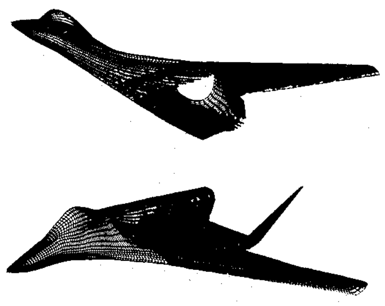

Fig. 3a Geometry and surface grid for realistic fighter-type configuration.

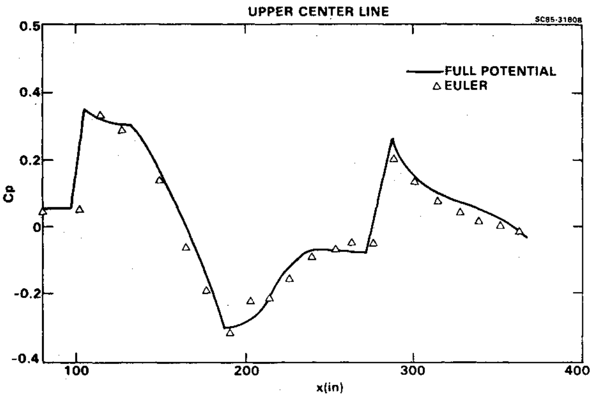

Fig. 3b Surface pressure distribution along upper symmetry plane.

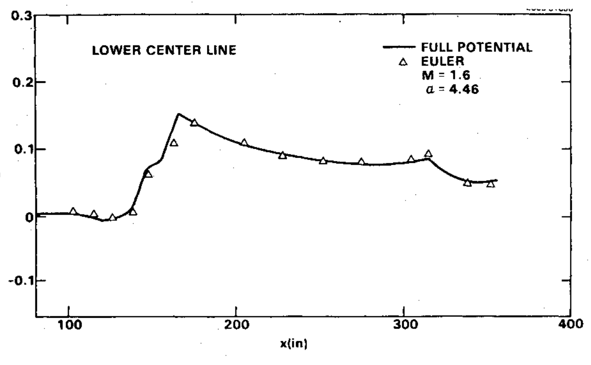

Fig. 3c Surface pressure distribution along lower symmetry plane.

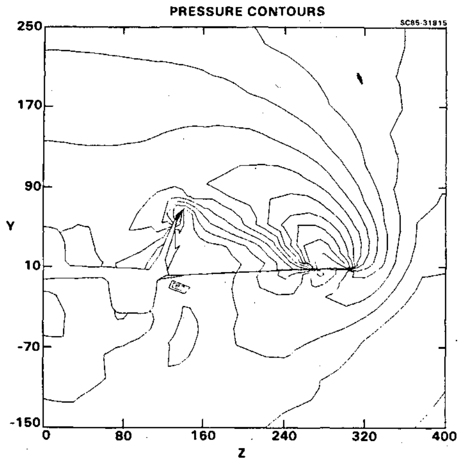

Fig. 3d Pressure contours at $x = 688$ in.

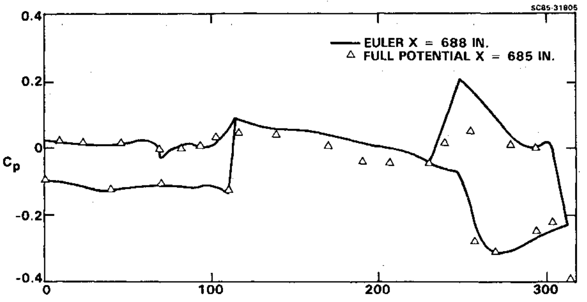

Fig. 3e Circumferential pressure distribution.

with results obtained using the full-potential marching code. The last set of results is for a Shuttle Orbiter configuration, and the numerical results are compared with experimental data for this case.

Analytic Forebody

Figure 2a shows the developed cross section of a forebody geometry reported in Ref. 10. The surface pressure distributions in the axial direction on the upper ($\theta = 0$ deg, leeward side) and lower ($\theta = 180$ deg, windward side) planes of symmetry at $M_\infty = 2.5$, $\alpha = 0$ deg are given in Fig. 2b. The grid and circumferential pressure distribution on the body surface at $x/\ell = 0.22$ and $x/\ell = 0.34$ for the same freestream conditions are presented in Figs. 2c and 2d, respectively. Figure 2e shows the circumferential pressure distribution on the same geometry for $M_\infty = 1.7$, $\alpha = -5$ deg at $x/\ell = 0.278$. It is noted that a small subsonic pocket develops, for this second case, on the lee side and that two global marching sweeps are enough for the present numerical method to give a very good converged solution. The experimental data[10] are also presented in these figures. The comparisons show that the present numerical predictions are in excellent agreement with experimental data.

Realistic Fighter Configuration

Figure 3a shows the geometry and surface gridding of a realistic fighter-type configuration, which includes a nacelle and a vertical tail. To illustrate the important features of the present analysis method, results have been obtained for the freestream condition $M_\infty = 1.6$, $\alpha = 4.94$ deg. The results are compared with those obtained using the SIMP full-potential solver. Figures 3b and 3c present the surface pressure at the upper and lower symmetry plane. The results show the excellent agreement between the predictions of these two codes. Circumferential pressure distributions and pressure contours at $x = 688$ in. are presented in Figs. 3d-3e. This section includes the nacelle, vertical tail, wake, and wing. The comparison shows very good agreement except at the lower surface of the wing in the vicinity of the wake region. A higher pressure is predicted by the Euler (EMTAC) code. It is also noted that the wake treatment in both methods provides the correct zero pressure jump across the wake.

Table 3 shows the comparison of overall forces in terms of C_L, C_D, and C_L/C_D. The drag calculation includes skin-friction drag estimated using a boundary-layer technique and an estimate of the base drag. Both the full-potential and Euler results agree very well with Rockwell experimental data, with the Euler results being closer to the data.

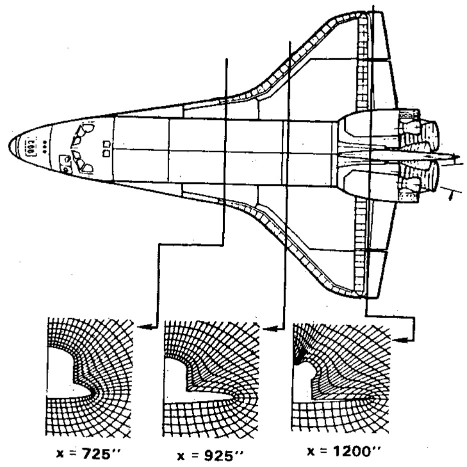

Fig. 4a Shuttle Orbiter configuration and sample grids.

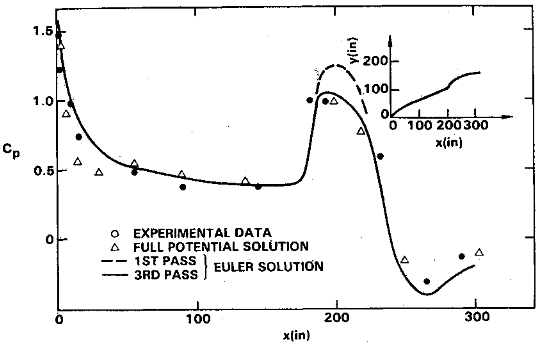

Fig. 4b Surface pressure distribution along upper symmetry plane.

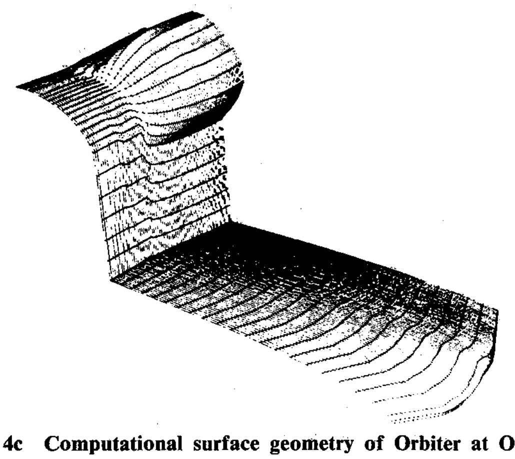

Fig. 4c Computational surface geometry of Orbiter at OMS pod region.

Space Shuttle Orbiter

Figures 4a–4f give the geometry, gridding, and corresponding flowfield solutions for an isolated Space Shuttle Orbiter flying at $M_\infty = 1.4$, $\alpha = 0$ deg. The EMTAC code is applied to compute the flowfield about the entire Orbiter, from nose to tail. Multiple uni- or bidirectional sweeps are used in the nose region to capture the detached bow shock and the subsonic region behind it. After this subsonic region transitions by expansion, over the shoulder region of the nose, into a supersonic flowfield, a simple forward-marching technique is employed. Multiple relaxation sweeps are also used in the canopy and OHMS pod regions to compute the locally subsonic regions.

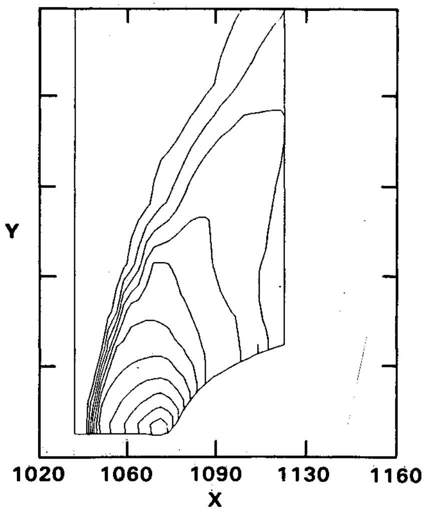

Fig. 4d Pressure and Mach number contours at OMS pod region: *x-y* section.

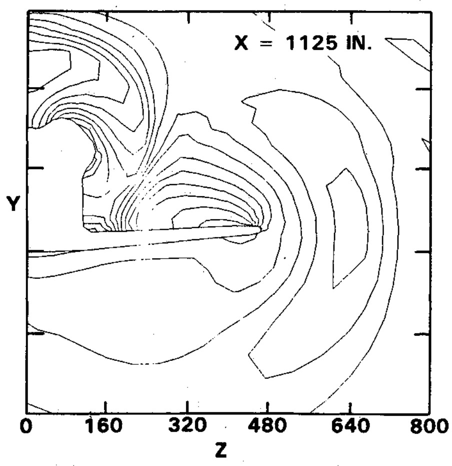

Fig. 4e Pressure contours at OMS pod region: *z-y* section.

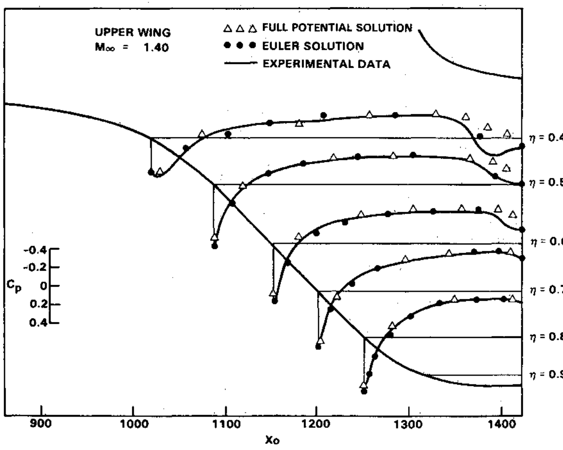

Fig. 4f Orbiter upper surface pressure distribution.

The surface pressure distribution along the leeward plane of symmetry in the nose region is presented in Fig. 4b. At $x = 170$ in., which is the beginning of the canopy, the pressure increases rapidly from $C_p = 0.3$ to ≈ 1.0. An embedded subsonic pocket is formed in the canopy region and required three relaxation marching sweeps to develop the solution. The results show that the present prediction is in excellent agreement with data.

Figure 4c shows the details of the orbiter geometry in the OMS pod region as modeled in this study. A detached OMS pod shock and a large subsonic pocket are formed in this region. Since the subsonic pocket is big and the Mach number is almost zero near the root of the OMS pod, a total of 30 relaxation marching sweeps (forward only) are required to give a good, converged result. Figure 4d presents the Mach number contours obtained in this region. The cross-sectional pressure contours at $x = 1125$ in. are given in Fig. 4e. The OMS pod shock is formed around $x = 1050$ in., then grows, and finally hits the upper wing surface at $x \approx 1095$ in. The chordwise pressure distributions on the upper surface of the wing at several span stations are compared with experimental data in Fig. 4f. It is seen that the present calculation agrees with the experimental data very well over the entire upper surface, including in the region where the OMS pod shock interacts with the wing surface.

Concluding Remarks

A new computational procedure has been devised to solve the Euler equations for three-dimensional supersonic inviscid flows with subsonic pockets. The method is akin to a simple marching procedure in portions of the flowfield where the component of velocity normal to the local marching plane is supersonic. When this local velocity is subsonic (in subsonic pockets, for example), a relaxation approach is used. The marching and relaxation strategies are both variations of a unified approach to the development of finite-difference methods for this class of problems. This approach is based on a planar Gauss-Seidel procedure coupled with approximate factorization in the plane. As an expository paper, detailed formulas are presented to aid the reader who would like to program the method independently.

The use of TVD discretizations results in a highly reliable method with no artificial parameters, such as coefficients of numerical smoothing, to be provided by the user. Spurious oscillations and expansion shocks are also eliminated.

Acknowledgments

This study was partially supported by NASA Langley Research Center under Contract NAS 1-17492 awarded to Rockwell International Science Center.

References

[1] Shankar, V., Szema, K. Y., and Osher, S., "A Conservative Type-Dependent Full Potential Method for the Treatment of Supersonic Flow with Imbedded Subsonic Region," AIAA Paper 83-1887, July 1983.

[2] Szema, K. Y. and Shankar, V., "Nonlinear Computation of Wing-Body-Vertical Tail-Wake Flows at Low Supersonic Speed," AIAA Paper 84-0427, Jan. 1984.

[3] Szema, K. Y., Riba, W. L., Shankar, V., and Gorski, J. J., "Full Potential Treatment of Flows Over 3-D Geometries Including Multibody Configurations," AIAA Paper 85-0272, Jan. 1985.

[4] Chakravarthy, S. R. and Ota, D. K., "Numerical Issues in Computing Inviscid Supersonic Flow Over Conical Delta Wings," AIAA Paper 86-0440, Jan. 1986.

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

[6] Chakravarthy, S. R. and Osher, S., "Computing With High-Resolution Upwind Schemes for Hyperbolic Equations," *Lectures in Applied Mathematics*, Vol. 22, American Mathematical Society, 1985, pp. 57–86.

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

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

[9] Chakravarthy, S. R., "Euler Equations—Implicit Schemes and Boundary Conditions," *AIAA Journal*, Vol. 21, May 1983, pp. 699–706.

[10] Townsend, J. C., Howell, D. T., Collins, I. K., and Hayes, C., "Surface Pressure Data on a Series of Analytic Forebodies at Mach Numbers from 1.7 to 4.5 and Combined Angles of Attack and Sideslip," NASA TM-80062, June 1979.