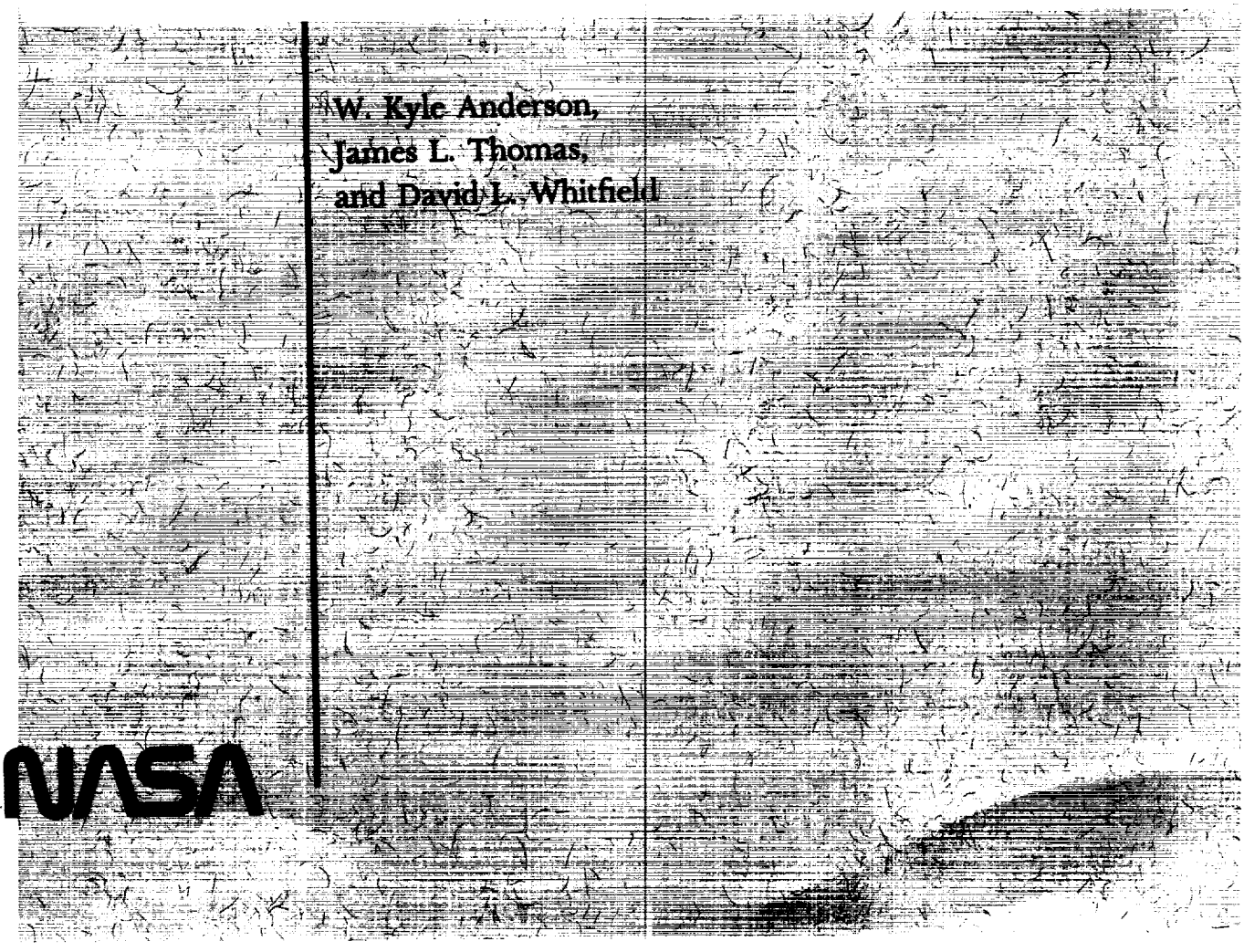
NASA
Technical
Paper
2829

November 1988

Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations



W. Kyle Anderson,
James L. Thomas,
and David L. Whitfield

NASA

NASA
Technical
Paper
2829

1988

Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations

W. Kyle Anderson
and James L. Thomas
Langley Research Center
Hampton, Virginia

David L. Whitfield
Mississippi State University
Mississippi State, Mississippi

National Aeronautics
and Space Administration

Scientific and Technical
Information Division

Summary

The full-approximation-scheme (FAS) multigrid method is applied to several implicit flux-split algorithms for solving the three-dimensional Euler equations in a body-fitted coordinate system. Each of the splitting algorithms uses a variation of approximate factorization and is implemented in a finite-volume formulation. The algorithms are all vectorizable with little or no scalar computation required. The flux vectors are split into upwind components using both the Steger-Warming and Van Leer splittings. Results comparing pressure distributions with experimental data using both splitting types are shown. The stability and smoothing rates of each of the schemes are examined using a Fourier analysis of the complete system of equations. Results are presented for three-dimensional subsonic, transonic, and supersonic flows that demonstrate substantially improved convergence rates with the multigrid algorithm. The influence of using both V-cycle and W-cycle strategies on the convergence is examined. By using the multigrid method in both subsonic and transonic wing calculations, the final lift coefficient is obtained to within 0.1 percent of its final value in as few as 15 multigrid cycles for a mesh with over 210 000 points. A spectral radius of 0.890 is achieved for both subsonic and transonic flows over the ONERA M6 wing, whereas a spectral radius of 0.830 is obtained for supersonic flow over an analytically defined forebody. Results compared with experiment show good agreement for all cases.

Introduction

Upwind difference methods for solving the Euler equations are becoming increasingly popular for several reasons. The time-dependent Euler equations form a system of hyperbolic equations, and upwind differencing models the characteristic nature of the equations in that information at each grid point is obtained from directions dictated by characteristic theory. Upwind methods have the advantage of being naturally dissipative. Separate spatial dissipation terms, such as those generally required in a central difference method to overcome oscillations or instabilities arising in regions of strongly varying gradients, need not be added. Some examples of upwind methods include the λ-method (ref. 1), the split-coefficient method (ref. 2), the flux-vector-splitting method (refs. 3, 4, and 5), and the flux-difference-splitting method (ref. 6).

Although the λ-method and the split-coefficient method closely mimic the method of characteristics, they are both applied to the nonconservative form of the equations and consequently require the use of shock-fitting techniques to obtain the correct location and strength of shocks in transition flows. Use of the conservation-law form of the Euler equations allows shock waves to be captured as weak solutions to the governing equations and circumvents the difficulty in applying shock-fitting techniques to arbitrary flows. Both the flux-difference-splitting and flux-vector-splitting methods can be applied to the conservation-law form.

The upwind method used in the current work is the flux-vector-splitting method in which the flux vectors are split into forward and backward contributions based on an eigenvalue decomposition and are differenced accordingly. The splittings investigated include those of Steger-Warming (refs. 3, 4, and 7) and Van Leer (refs. 5, 8, and 9). In comparison with unsplit methods, the advantages of flux splitting are obtained at the cost of increased computational work since two sets of fluxes are computed for each coordinate direction and implicit schemes require two sets of flux Jacobians for consistent linearization of the fluxes. In addition, the split fluxes and flux Jacobians are also generally more complicated than the unsplit terms because of the logic involved with eigenvalue sign changes.

In order to offset the additional computational work of the upwind methods, it is highly desirable to accelerate the convergence rate, especially when only steady-state solutions are sought; the objective is to reduce the computer time required while still maintaining the high level of robustness and accuracy attained from upwind differencing. Accelerating the convergence rate becomes increasingly important as the mesh is refined since the log of the spectral radius for single-grid methods generally increases linearly with the mesh size, thus making computations on very fine meshes impractical.

One method that has been successful in accelerating the convergence rate of elliptic problems, i.e., attaining a spectral radius independent of the mesh spacing, is the multigrid method (refs. 10 and 11). Although most of the existing theory on multigrid methods pertains specifically to elliptic equations, it has been shown in a number of references (refs. 12 to 22) that the multigrid method can greatly accelerate the convergence rate of numerical schemes used for solving the Euler equations.

One of the earliest applications of multiple grids in solving the Euler equations was presented by Ni (ref. 12) who used coarse grids to propagate corrections rapidly throughout the domain. His original idea was first incorporated into a one-step Lax-Wendroff method and was later extended for use into predictor-corrector types of methods by Johnson (ref. 13). Johnson and Chima (refs. 14, 15, and 16) subsequently used the

method to calculate both inviscid and viscous flows over several two-dimensional geometries. In 1984, Mulder applied a linear multigrid method to the Euler equations in two space dimensions by using upwind differencing to calculate flow over a circular arc and for a weakly barred galaxy (ref. 17). Jespersen also used upwind differencing in two spatial dimensions to calculate flow over airfoils (ref. 18). In this approach, the Euler equations were solved by Newton iteration where the linear system arising at each step was solved using the multigrid method. One of the first uses of the nonlinear multigrid method in accelerating the convergence rate for both the two- and three-dimensional Euler equations was reported by Jameson who used central differencing in a four-stage Runge-Kutta algorithm to advance the solution (refs. 19 and 20). In two dimensions, recent work by Jameson and Yoon also used central differencing and incorporated the multigrid algorithm into some implicit schemes with good success (refs. 21 and 22).

The purpose of the current investigation is to combine the full-approximation-scheme (FAS) multigrid method with flux-vector splitting to obtain efficient solutions to the Euler equations in three dimensions. The full-approximation scheme for a general nonlinear problem is discussed as well as its implementation for the Euler equations. Both V- and W-type cycling strategies are also investigated. Several smoothing algorithms are given involving mostly vectorizable computations on the Control Data Corporation VPS-32 supercomputer (a CYBER 205 with 32 million words of memory) at the NASA Langley Research Center. In addition, both the Steger-Warming and Van Leer splittings are considered for splitting the flux vectors into upwind components. Numerical pressure distributions are compared with available experiment for subsonic and transonic flows over the ONERA M6 wing and for supersonic flow over an analytically defined forebody.

Symbols

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$	flux Jacobians in Cartesian coordinates, $\partial\mathbf{F}/\partial\mathbf{Q}, \partial\mathbf{G}/\partial\mathbf{Q}$, and $\partial\mathbf{H}/\partial\mathbf{Q}$, respectively
$\hat{\mathbf{A}}^{\pm}, \hat{\mathbf{B}}^{\pm}, \hat{\mathbf{C}}^{\pm}$	flux Jacobians in generalized coordinates, $\partial\hat{\mathbf{F}}^{\pm}/\partial\hat{\mathbf{Q}}, \partial\hat{\mathbf{G}}^{\pm}/\partial\hat{\mathbf{Q}}$, and $\partial\hat{\mathbf{H}}^{\pm}/\partial\hat{\mathbf{Q}}$, respectively
$\bar{\mathbf{A}}$	matrix from similarity transformation
a	speed of sound
CFL	Courant-Friedrichs-Lewy number
C_p	pressure coefficient
c	airfoil chord
c_l	lift coefficient
e	total energy per unit volume
$\mathbf{F}, \mathbf{G}, \mathbf{H}$	flux vectors of mass, momentum, and energy
f_{energy}	energy component of flux vector
f_{mass}	mass component of flux vector
G_N	grid level N
grad( )	gradient operator
$\mathbf{I}$	identity matrix
$\mathbf{I}_i^{i-1}$	restriction operator used for transferring functions on grid i to grid $i-1$
$\mathbf{I}_{i-1}^{i}$	prolongation operator used for transferring functions on grid $i-1$ to grid i
$\hat{\mathbf{I}}_i^{i-1}$	restriction operator for the residual
i	$= \sqrt{-1}$
$\hat{i}, \hat{j}$	x and y components of a unit vector
J	transformation Jacobian

	denotes ξ, η, or ς
	generic nonlinear operator
	Mach number
	local Mach number in ξ-direction
	implicit operator
	unit normal vector
	forcing function added to residual
	pressure
	conserved variables representing mass, momentum, and total energy per unit volume
	change in dependent variables, $\mathbf{Q}^{n-1} - \mathbf{Q}^n$
	approximate value of conserved variables on grid i
	residual
	Riemann invariant
	components of a vector in x-, y-, and z-directions, respectively
	forcing function
	entropy
	length of cell face
	arc length, as used in appendix C
	similarity transformation matrix whose columns are the right eigenvectors of the flux Jacobians; also denotes a rotation matrix in appendix B
	similarity transformation matrix whose rows are the left eigenvectors of the flux Jacobians; also denotes inverse of rotation matrix in appendix B
	time
$\hat{t}_x, \hat{t}_y, \hat{t}_z$	components of a vector in x-, y-, and z-directions, respectively
Δt	time step
U, V, W	contravariant velocities; V also denotes the volume of a computational cell where indicated
$\mathbf{U}_0$	a constant vector
u, v, w	Cartesian velocities
$\mathbf{V}_i$	correction on grid level i
$\mathbf{v}$	eigenvector for Fourier analysis
x, y, z	Cartesian coordinates
α	angle of attack
$\alpha_s, \beta_s, \gamma_s$	coefficients for stability analysis
γ	ratio of specific heats
Δ	incremental change
$\delta_x, \delta_y, \delta_z$	difference operator in x-, y-, and z-directions

η_s	percent of spanwise location on wing, as used in figures 15, 16, 18, and 19
$\mathbf{\Lambda}$	diagonal matrix of eigenvalues
λ	amplification factor
$\lambda_1, ..., \lambda_5$	eigenvalues of Jacobian matrices
μ	smoothing factor
ξ, η, ς	general curvilinear coordinates
$\hat{\xi}_x, \hat{\xi}_y, \hat{\xi}_z$	components of a unit vector normal to a ξ = Constant face
ρ	density
τ	time
τ_i	relative truncation error
$\|\ \|$	magnitude of a vector
Subscripts:	
A	large mesh cell in appendix C
a, b, c, d	smaller mesh cells in appendix C; b also denotes values on boundary in equations (34) to (36)
i, j, k	cell indices
$N, N-1, ...$	discretization on grid $N, N-1, ...$
r	reference values
x, y, z	spatial derivatives
Superscripts:	
c	most current value of a quantity
n	time level
$\pm$	positive and negative flux and eigenvalue contributions; also denotes forward and backward spatial differencing or extrapolation where indicated
$(\bar{})$	quantities in locally orthogonal coordinates
$(\hat{})$	fluxes and flux Jacobians in generalized coordinates; also denotes Fourier symbol where indicated

Euler Solution Method

Euler Equations in Generalized Coordinates

The governing equations are the time-dependent equations of ideal gas dynamics, i.e., the Euler equations, that express the conservation of mass, momentum, and energy for an inviscid nonconducting gas in the absence of external forces. The conservation form of the equations in generalized coordinates is given by

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi} + \frac{\partial \hat{\mathbf{G}}}{\partial \eta} + \frac{\partial \hat{\mathbf{H}}}{\partial \varsigma} = 0 \tag{1}$$

where

$$\hat{\mathbf{Q}} = \frac{\mathbf{Q}}{J} = \frac{1}{J} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix} \tag{2}$$

$$\hat{\mathbf{F}} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ (e+p)U \end{Bmatrix} \tag{3}$$

$$\hat{\mathbf{G}} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ (e+p)V \end{Bmatrix} \tag{4}$$

$$\hat{\mathbf{H}} = \frac{1}{J}\begin{Bmatrix} \rho W \\ \rho W u + \varsigma_x p \\ \rho W v + \varsigma_y p \\ \rho W w + \varsigma_z p \\ (e+p)W \end{Bmatrix} \tag{5}$$

The pressure p is related to the conserved variables through the ideal gas law

$$p = (\gamma - 1)[e - \rho(u^2 + v^2 + w^2)/2] \tag{6}$$

The equations have been generalized from Cartesian coordinates using a steady transformation of the type

$$\xi = \xi(x, y, z) \qquad \eta = \eta(x, y, z) \qquad \varsigma = \varsigma(x, y, z) \qquad \tau = t \tag{7}$$

where the contravariant velocity components are

$$U = \xi_x u + \xi_y v + \xi_z w \tag{8a}$$

$$V = \eta_x u + \eta_y v + \eta_z w \tag{8b}$$

$$W = \varsigma_x u + \varsigma_y v + \varsigma_z w \tag{8c}$$

The transformation to generalized coordinates is given in appendix A.

The equations, although written in generalized coordinates, are used in a finite-volume formulation. Equation (1) can be interpreted as describing the balance of mass, momentum, and energy over an arbitrary control volume. In this connection, the vectors $\text{grad}(\xi)/J$, $\text{grad}(\eta)/J$, and $\text{grad}(\varsigma)/J$ represent directed areas of cell interfaces in the contravariant ξ-, η-, and ς-directions; i.e., in directions along normals to the ξ = Constant, η = Constant, and ς = Constant surfaces, respectively. The Jacobian J represents the inverse of the cell volume. Likewise, the quantities $\rho U/J, \rho V/J$, and $\rho W/J$ represent the contravariant mass flux crossing the cell interfaces in the contravariant ξ-, η-, and ς-directions.

Flux-Vector Splitting

The upwind differencing in the present work is effected through the technique of flux-vector splitting. The generalized fluxes $\hat{\mathbf{F}}, \hat{\mathbf{G}}$, and $\hat{\mathbf{H}}$ are split into forward and backward contributions according to the signs of the eigenvalues of the Jacobian matrices and are differenced accordingly. For example, the flux in the ξ-direction can be differenced as

$$\delta_\xi \hat{\mathbf{F}} = \delta_\xi^- \hat{\mathbf{F}}^+ + \delta_\xi^+ \hat{\mathbf{F}}^- \tag{9}$$

since $\hat{\mathbf{F}}^+$ has all nonnegative eigenvalues and $\hat{\mathbf{F}}^-$ has all nonpositive eigenvalues. For the current study, two methods are considered for splitting the flux vectors into upwind components. Although the details of each method can be found in references 3, 4, 5, 7, and 9, both methods are briefly discussed below.

The first method presented is the technique outlined by Steger and Warming in reference 3. Since the flux vectors are homogeneous functions of degree one in $\hat{\mathbf{Q}}$, they can be expressed in terms of their Jacobian matrices. For example, considering the flux vector in the ξ-direction allows $\hat{\mathbf{F}}$ to be written as

$$\hat{\mathbf{F}} = \hat{\mathbf{A}}\hat{\mathbf{Q}} = \frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{Q}}}\hat{\mathbf{Q}} \tag{10}$$

Using a similarity transformation allows equation (10) to be rewritten as

$$\hat{\mathbf{F}} = \hat{\mathbf{A}}\hat{\mathbf{Q}} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\hat{\mathbf{Q}} \tag{11}$$

The matrix $\mathbf{\Lambda}$ is a diagonal matrix composed of the eigenvalues of $\hat{\mathbf{A}}$ and is given by

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 \\ 0 & 0 & 0 & \lambda_4 & 0 \\ 0 & 0 & 0 & 0 & \lambda_5 \end{bmatrix} \tag{12}$$

where

$$\left.\begin{aligned} \lambda_{1,2,3} &= U = \xi_x u + \xi_y v + \xi_z w \\ \lambda_4 &= U + |\mathrm{grad}(\xi)|a \\ \lambda_5 &= U - |\mathrm{grad}(\xi)|a \end{aligned}\right\} \tag{13}$$

The eigenvalues can then be decomposed into nonnegative and nonpositive components

$$\lambda_i = \lambda_i^+ + \lambda_i^- \tag{14}$$

where

$$\lambda_i^{\pm} = \frac{\lambda_i \pm |\lambda_i|}{2} \tag{15}$$

Similarly, the eigenvalue matrix $\mathbf{\Lambda}$ can be decomposed into

$$\mathbf{\Lambda} = \mathbf{\Lambda}^+ + \mathbf{\Lambda}^- \tag{16}$$

where $\mathbf{\Lambda}^+$ is made up of the nonnegative contributions of λ_i^+ and $\mathbf{\Lambda}^-$ is constructed of the nonpositive contributions of λ_i^-. This splitting of the eigenvalue matrix, combined with equation (11), allows the flux vector $\hat{\mathbf{F}}$ to be rewritten as

$$\hat{\mathbf{F}} = \mathbf{T}(\mathbf{\Lambda}^+ + \mathbf{\Lambda}^-)\mathbf{T}^{-1}\hat{\mathbf{Q}} = (\bar{\mathbf{A}}^+ + \bar{\mathbf{A}}^-)\hat{\mathbf{Q}} = \hat{\mathbf{F}}^+ + \hat{\mathbf{F}}^- \tag{17}$$

The flux vector $\hat{\mathbf{F}}$ has three distinct eigenvalues given by equation (13) and can therefore be written as a sum of three subvectors, each of which has a distinct eigenvalue as a coefficient (ref. 7):

$$\hat{\mathbf{F}} = \hat{\mathbf{F}}_1 + \hat{\mathbf{F}}_2 + \hat{\mathbf{F}}_3 \tag{18}$$

where

$$\hat{\mathbf{F}}_1 = \lambda_1 \frac{\gamma - 1}{J\gamma} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho}{2}(u^2 + v^2 + w^2) \end{Bmatrix} \tag{19}$$

$$\hat{\mathbf{F}}_{2,3} = \lambda_{4,5} \frac{1}{J2\gamma} \begin{Bmatrix} \rho \\ \rho u \pm \rho a \hat{\xi}_x \\ \rho v \pm \rho a \hat{\xi}_y \\ \rho w \pm \rho a \hat{\xi}_z \\ e + p \pm \frac{\rho a U}{|\mathrm{grad}(\xi)|} \end{Bmatrix} \tag{20}$$

and the direction cosines of the directed interface in the ξ-direction are

$$\hat{\xi}_x = \xi_x / |\mathrm{grad}(\xi)| \tag{21a}$$

$$\hat{\xi}_y = \xi_y / |\mathrm{grad}(\xi)| \tag{21b}$$

$$\hat{\xi}_z = \xi_z / |\mathrm{grad}(\xi)| \tag{21c}$$

The forward and backward flux vectors $\hat{\mathbf{F}}^+$ and $\hat{\mathbf{F}}^-$ are formed from equations (18), (19), and (20) by inserting $\lambda_i = \lambda_i^+$ and $\lambda_i = \lambda_i^-$, respectively.

For supersonic and sonic flow in the ξ-direction, i.e., $|M_\xi| = |\bar{u}/a| \geq 1$, where $\bar{u} = U/|\mathrm{grad}(\xi)|$ represents the velocity normal to a ξ = Constant face, it should be noted that the fluxes in this direction become

$$\hat{\mathbf{F}}^+ = \hat{\mathbf{F}} \qquad \hat{\mathbf{F}}^- = 0 \qquad (M_\xi \geq 1) \tag{22a}$$

$$\hat{\mathbf{F}}^+ = 0 \qquad \hat{\mathbf{F}}^- = \hat{\mathbf{F}} \qquad (M_\xi \leq -1) \tag{22b}$$

The split fluxes in the other two directions are easily obtained by interchanging η or ς in place of ξ.

The fluxes split in the aforementioned manner are not continuously differentiable at zeros of the eigenvalues (i.e., sonic and stagnation points). (See ref. 23.) This is illustrated in figure 1 where the split mass flux contributions for the one-dimensional Euler equations, nondimensionalized by ρa, are shown as a function of

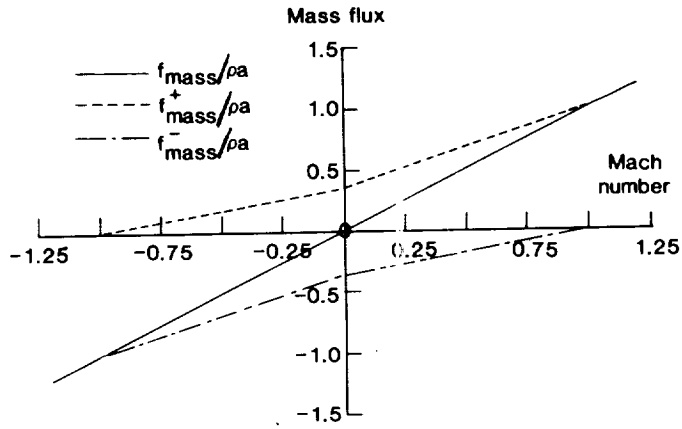


Figure 1. Variation of Steger and Warming split mass flux with Mach number.

Mach number. The gradient discontinuities in the split fluxes are evident as the eigenvalues pass through zero. The lack of differentiability of the split fluxes has been shown in some cases to cause small oscillations at sonic points that are rarely noticeable for most aerodynamic applications.

It should also be noted that the Jacobian matrices of $\hat{\mathbf{F}}^+$ and $\hat{\mathbf{F}}^-$ that are required for proper linearization for an implicit scheme do not have the same eigenvalues as $\bar{\mathbf{A}}^+$ and $\bar{\mathbf{A}}^-$ defined in equation (17). (See ref. 23.) However, the Jacobian matrices of $\hat{\mathbf{F}}^+$ and $\hat{\mathbf{F}}^-$ do have the same sign as $\bar{\mathbf{A}}^+$ and $\bar{\mathbf{A}}^-$ so that upwind differencing the spatial derivatives remains appropriate (ref. 23). Although $\bar{\mathbf{A}}^\pm$ are easier to form, their use in implicit schemes, instead of the correct linearizations $\hat{\mathbf{A}}^\pm$, has been shown in many cases to cause severe time-step limitations (refs. 3, 24, and 25).

In 1982 a new method of splitting the flux vector was proposed by Van Leer (ref. 5). Here, the fluxes were split so that the forward and backward flux contributions blended smoothly at eigenvalue sign changes, i.e., near sonic and stagnation points. Just as for the Steger-Warming splitting, it was required that the Jacobian matrices $\partial\hat{\mathbf{F}}^+/\partial\hat{\mathbf{Q}}$ have nonnegative eigenvalues and $\partial\hat{\mathbf{F}}^-/\partial\hat{\mathbf{Q}}$ have nonpositive eigenvalues so that upwind differencing could be used for the spatial derivatives. In addition, it was required that both Jacobians have one zero eigenvalue for subsonic Mach numbers which leads to steady transonic shock structures with only two transition zones (ref. 5). In practice, when second-order spatial differencing is used, shocks with only one interior zone are usually obtained (ref. 9). This feature is not observed with the Steger-Warming flux splitting.

The three-dimensional splittings of Van Leer were originally given for Cartesian coordinates. The extension to generalized coordinates is given in appendix B with the resulting split fluxes given below. Only the splitting for the flux in the ξ-direction is given, as the others can be obtained similarly. The flux vector $\hat{\mathbf{F}}$ is split according to the contravariant Mach number in the ξ-direction, defined previously as $M_\xi = \bar{u}/a$. For supersonic flow $(|M_\xi| \geq 1)$,

$$\hat{\mathbf{F}}^+ = \hat{\mathbf{F}} \qquad \hat{\mathbf{F}}^- = 0 \qquad (M_\xi \geq +1) \tag{23a}$$

$$\hat{\mathbf{F}}^- = \hat{\mathbf{F}} \qquad \hat{\mathbf{F}}^+ = 0 \qquad (M_\xi \leq -1) \tag{23b}$$

and for subsonic flow $(|M_\xi| < 1)$,

$$\hat{\mathbf{F}}^\pm = \frac{|\text{grad}(\xi)|}{J} \begin{Bmatrix} f^\pm_{\text{mass}} \\ f^\pm_{\text{mass}} \left\{ [\hat{\xi}_x(-\bar{u} \pm 2a)/\gamma] + u \right\} \\ f^\pm_{\text{mass}} \left\{ [\hat{\xi}_y(-\bar{u} \pm 2a)/\gamma] + v \right\} \\ f^\pm_{\text{mass}} \left\{ [\hat{\xi}_z(-\bar{u} \pm 2a)/\gamma] + w \right\} \\ f^\pm_{\text{energy}} \end{Bmatrix} \tag{24a}$$

where

$$f^\pm_{\text{mass}} = \pm\rho a(M_\xi \pm 1)^2/4 \tag{24b}$$

$$f^\pm_{\text{energy}} = f^\pm_{\text{mass}} \left\{ [-(\gamma-1)\bar{u}^2 \pm 2(\gamma-1)\bar{u}a + 2a^2]/(\gamma^2-1) + (u^2+v^2+w^2)/2 \right\} \tag{24c}$$

For forming $\hat{\mathbf{F}}^\pm$, the direction cosines $\hat{\xi}_x, \hat{\xi}_y$, and $\hat{\xi}_z$ are given by equations (21) and $\bar{u}$ is the velocity normal to a ξ = Constant face. The fluxes in the other two directions are easily formed by interchanging ξ with η or ς. In figure 2 the nondimensionalized mass flux using the Van Leer splitting is shown as a function of Mach number for the one-dimensional Euler equations. The split fluxes are continuously differentiable at sonic and stagnation points; the improvement over the Steger-Warming splitting is apparent.

Baseline Solution Algorithm

The baseline algorithm for updating the steady Euler equations stems from a backward Euler time integration of the unsteady equations, which yields (ref. 8)

$$\left[\mathbf{I} + \Delta t(\delta_\xi^- \hat{\mathbf{A}}^+ + \delta_\xi^+ \hat{\mathbf{A}}^-) + \Delta t(\delta_\eta^- \hat{\mathbf{B}}^+ + \delta_\eta^+ \hat{\mathbf{B}}^-) + \Delta t(\delta_\varsigma^- \hat{\mathbf{C}}^+ + \delta_\varsigma^+ \hat{\mathbf{C}}^-)\right] \Delta\hat{\mathbf{Q}} = -\Delta t\, \mathbf{R}^n \tag{25a}$$

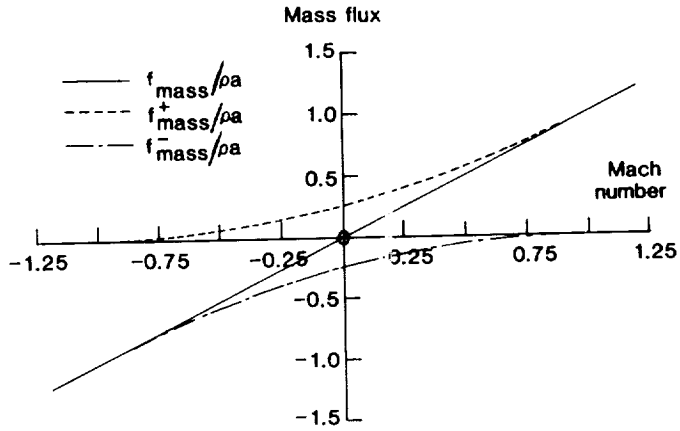


Figure 2. Variation of Van Leer split mass flux with Mach number.

where the residual at time level n is given by

$$\mathbf{R}^n = \delta_\xi^- \hat{\mathbf{F}}^+ + \delta_\xi^+ \hat{\mathbf{F}}^- + \delta_\eta^- \hat{\mathbf{G}}^+ + \delta_\eta^+ \hat{\mathbf{G}}^- + \delta_\varsigma^- \hat{\mathbf{H}}^+ + \delta_\varsigma^+ \hat{\mathbf{H}}^- \tag{25b}$$

The split-flux differences in equations (25) are implemented as a flux balance across a cell, corresponding to MUSCL-type differencing (Monotone Upstream-Centered Schemes for Conservation Laws). (See ref. 26.) For example, the flux balance in the ξ-direction across a cell centered at point (i, j, k) can be written as

$$\delta_\xi^- \hat{\mathbf{F}}^+ + \delta_\xi^+ \hat{\mathbf{F}}^- = [\hat{\mathbf{F}}^+(\mathbf{Q}^-) + \hat{\mathbf{F}}^-(\mathbf{Q}^+)]_{i+1/2)} - [\hat{\mathbf{F}}^+(\mathbf{Q}^-) + \hat{\mathbf{F}}^-(\mathbf{Q}^+)]_{i-(1/2)} \tag{26}$$

The notation $\hat{\mathbf{F}}^+(\mathbf{Q}^-)_{i+(1/2)}$ denotes the forward flux evaluated using the metric terms at the cell interface $i + (1/2)$ and the conserved state variables on the upwind side of the interface, obtained by a fully upwind second-order state variable interpolation:

$$\mathbf{Q}^-_{i+(1/2)} = 1.5\mathbf{Q}_i - 0.5\mathbf{Q}_{i-1} \tag{27a}$$

$$\mathbf{Q}^+_{i+(1/2)} = 1.5\mathbf{Q}_{i+1} - 0.5\mathbf{Q}_{i+2} \tag{27b}$$

As seen in figure 3, $\mathbf{Q}^n_{i,j,k}$ denotes the average value of $\mathbf{Q}$ in the cell centered on $(\xi_i, \eta_j, \varsigma_k)$ at time t^n; for simplicity, wherever the index notation is (i, j, k) or n, it is most often dropped.

In equations (25), if second-order differencing is used on both sides of the equation, Newton iteration for the steady Euler equations is obtained as Δt approaches infinity. The solution however requires a large banded block matrix to be solved at each step, a procedure that is generally not feasible because of the large number of operations required to invert the system. Even if the differencing on the left-hand side of the equation is reduced to first order, which would not affect the second-order accuracy of the final solution, the resulting system of equations usually remains uneconomical to solve. Therefore, the solution is obtained using approximate factorization, which splits the implicit operator into a sequence of easily invertible equations.

When using flux-vector splitting, there are numerous ways of factoring the implicit operator into a sequence of simpler operators (ref. 3). For the results shown below, three ways of factoring are considered. Each of the schemes uses first-order spatial differencing on the implicit side of the equation, whereas second-order differencing is maintained for the residual calculations. Since the steady state does not depend on the differencing of the left-hand side, the final steady-state result will have spatially second-order accuracy. The computational modules for each of the schemes is shown in figure 4. All the schemes employ simple explicit boundary conditions. Since only steady-state solutions are sought, each cell is advanced at its own time step corresponding to a given CFL number defined by

$$\mathrm{CFL} = \Delta t \{|U| + |V| + |W| + a\,[|\mathrm{grad}(\xi)| + |\mathrm{grad}(\eta)| + |\mathrm{grad}(\varsigma)|]\} \tag{28}$$

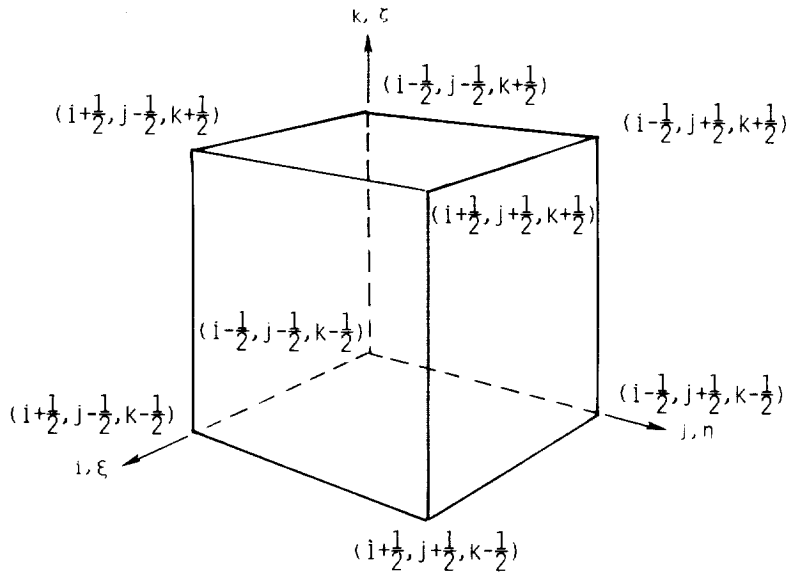
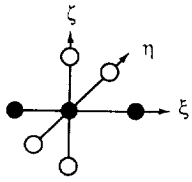
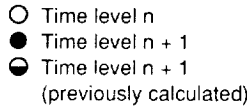
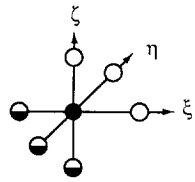


Figure 3. Computational cell indexing.

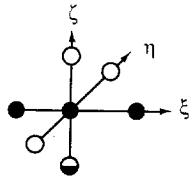


(a) Three-factor spatially split. ξ sweep.





(b) Two-factor eigenvalue split. Positive eigenvalue sweep.



(c) Two-factor combination split. $\xi/\hat{\mathbf{C}}^+$ sweep.

Figure 4. Computational modules.

The first scheme considered is a spatially split algorithm given by

$$[\mathbf{I} + \Delta t(\delta_\xi^- \hat{\mathbf{A}}^+ + \delta_\xi^+ \hat{\mathbf{A}}^-)][\mathbf{I} + \Delta t(\delta_\eta^- \hat{\mathbf{B}}^+ + \delta_\eta^+ \hat{\mathbf{B}}^-)][\mathbf{I} + \Delta t(\delta_\zeta^- \hat{\mathbf{C}}^+ + \delta_\zeta^+ \hat{\mathbf{C}}^-)]\Delta\hat{\mathbf{Q}} = -\Delta t\ \mathbf{R}^n \qquad (29)$$

The computational module for the left-hand side of equation (29) is shown in figure 4(a) for the ξ sweep. Since the solution at each point is directly coupled to the two neighboring points, the scheme requires the solution of a system of block tridiagonals. Similarly, the other two factors also require a block tridiagonal inversion. This scheme has the advantage however of being completely vectorizable, and viscous effects can be easily incorporated into the left-hand side. Since the VPS-32 computer is much faster for long vector lengths than for short ones, the computations in the present implementation take advantage of the large memory available on the VPS-32 and solve the block matrix equations over multiple planes simultaneously, thus yielding longer vector lengths and faster processing rates. As seen in figure 5, the block tridiagonal matrices can be solved with vector lengths corresponding to the number of lines in a plane times the number of planes taken. The residual calculations, on the other hand, can be made with vector lengths corresponding to the number of points in the grid. To decompose the implicit operator into lower (L) and upper (U) matrices (LU decomposition) and perform the back-substitutions requires 695 multiplications and additions per factor resulting in a total of 2085 operations for each sweep through the grid.

The second method considered to factor the left-hand side of equation (25a) is a two-factor method in which

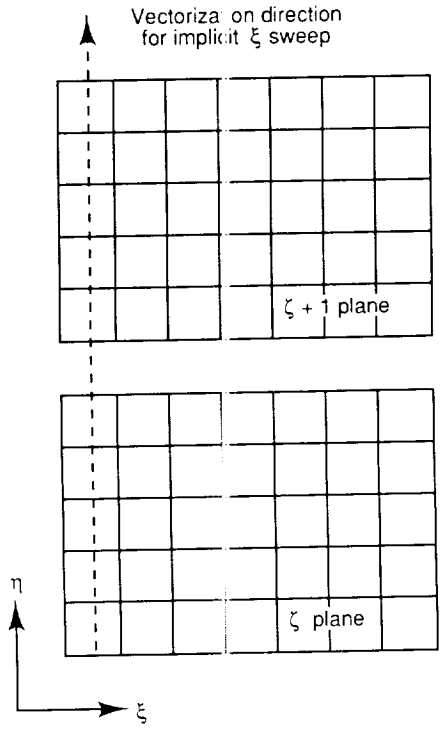


Figure 5. Vectorization of LU decomposition and back-substitution over multiple planes.

The second method considered to factor the left-hand side of equation (25a) is a two-factor method in which the implicit operator is split such that one operator contains the Jacobians with all positive eigenvalues and the other operator contains the Jacobians with all negative eigenvalues. The scheme can be written as

$$[\mathbf{I} + \Delta t(\delta_\xi^- \hat{\mathbf{A}}^+ + \delta_\eta^- \hat{\mathbf{B}}^+ + \delta_\zeta^- \hat{\mathbf{C}}^+)][\mathbf{I} + \Delta t(\delta_\xi^+ \hat{\mathbf{A}}^- + \delta_\eta^+ \hat{\mathbf{B}}^- + \delta_\zeta^+ \hat{\mathbf{C}}^-)]\Delta\hat{\mathbf{Q}} = -\Delta t\ \mathbf{R}^n \tag{30}$$

and requires only the solution of block lower triangular equations. Each factor is solved by starting at one corner of the grid and solving for each point by marching across the field to the opposite corner. From the computational module shown in figure 4(b) for the first factor in equation (30), the solution at the center node requires that the solution of each of the points behind it be previously obtained. These terms are taken to the right-hand side of the equation and added to the residual. A similar procedure is carried out for the second factor. For the present implementation of this scheme only 270 operations are required for each factor to invert the left-hand side and perform the back-substitutions, and most of the algorithm is vectorizable. This scheme requires that a 5×5 matrix be inverted at each point in the grid. These inversions can be carried out with vector lengths corresponding to the number of points in the grid. However, since the solution of each plane requires that the solution of the previous plane be known, the back-substitution cannot be performed over multiple planes. The maximum vector lengths in the back-substitution process correspond to the number of points in a plane. Scalar computations are used to perform the back-substitution along a line and comprise roughly one-third of the total operations. Although the scheme requires only about 25 percent of the operations required for the spatially split scheme, the scalar computations degrade the processing rate significantly so that the overall computer time per iteration on the VPS-32 is about twice as much as that for the spatially split scheme.

The last scheme considered is another two-factor scheme that is spatially split in two directions, with the third direction split according to the sign of its eigenvalues. The resulting scheme, which is referred to as combination splitting, is given by

$$[\mathbf{I} + \Delta t(\delta_\xi^- \hat{\mathbf{A}}^+ + \delta_\xi^+ \hat{\mathbf{A}}^- + \delta_\zeta^- \hat{\mathbf{C}}^+)][\mathbf{I} + \Delta t(\delta_\eta^- \hat{\mathbf{B}}^+ + \delta_\eta^+ \hat{\mathbf{B}}^- + \delta_\zeta^+ \hat{\mathbf{C}}^-)]\Delta\hat{\mathbf{Q}} = -\Delta t\ \mathbf{R}^n \tag{31}$$

and the computational module for the first factor is shown in figure 4(c). This scheme also requires the solution of block tridiagonal systems and requires about 695 multiplications and divisions for each factor. This scheme is completely vectorizable; however, as in the previous two-factor scheme, the solution of each plane requires that the solution of the previous plane be known, thereby eliminating the possibility of extending the vector operations over several planes. The result is that even though this scheme requires only two-thirds of the operations of the three-factor scheme, the computational rate is actually degraded by about 10 percent. A summary of operations required to solve the left-hand side for each of the three schemes is given in table I.

Table I. Operation Counts for Solving the Implicit Operators

Scheme	Operations required per point per factor with associated vector lengths[a] for—		Total operations per point
	LU decomposition	Back substitution	
Three-factor spatially split	550 ML	145 ML	695 × 3 = 2085
Two-factor combination split	{550 ML, 50 L}	145 L	745 × 2 = 1490
Two-factor eigenvalue split	{75 MP, 50 P, 50 L, 50 S}	45 S	270 × 2 = 540

[a]L—vector length corresponds to number of lines in a plane.
LU—lower and upper.
ML—vector length corresponds to number of lines in a plane times number of planes.
MP—vector length corresponds to number of points in a plane times number of planes.
P—vector length corresponds to number of points in a plane.
S—scalar computation.

Boundary Conditions

The boundary conditions for the solutions presented below are applied explicitly. On the body, the normal velocity is set to zero, whereas the pressure and density are determined by zeroeth-order extrapolation from the interior. For subsonic flow in the far field, the velocity normal to the boundary and the speed of sound are obtained from two locally one-dimensional Riemann invariants given by

$$R^{\pm} = \bar{u} \pm \frac{2a}{\gamma - 1} \tag{32}$$

These invariants are considered constant along characteristics defined normal to the outer boundary given by

$$\left(\frac{d\bar{x}}{dt}\right)^{\pm} = \bar{u} \pm a \tag{33}$$

For subsonic conditions at the boundary, R^- can be evaluated locally from free-stream conditions outside the computational domain and R^+ is evaluated locally from the interior of the domain. The local normal velocity and speed of sound on the boundary are calculated using the Riemann invariants as

$$\bar{u}_b = \frac{1}{2}(R^+ + R^-) \tag{34a}$$

$$a_b = \frac{\gamma - 1}{4}(R^+ - R^-) \tag{34b}$$

The Cartesian velocities are determined on the outer boundary by decomposing the normal and tangential velocity vectors into components yielding

$$\left.\begin{aligned} u_b &= u_r + \hat{k}_x(\bar{u}_b - \bar{u}_r) \\ v_b &= v_r + \hat{k}_y(\bar{u}_b - \bar{u}_r) \\ w_b &= w_r + \hat{k}_z(\bar{u}_b - \bar{u}_r) \end{aligned}\right\} \tag{35}$$

where the subscript r represents values obtained from one point outside the domain for inflow and from one point inside the domain for outflow.

The entropy S is determined using the value from outside or inside the domain, depending on whether the boundary is an inflow or outflow boundary. Once the entropy is known, the density on the far field boundary is calculated from the entropy and speed of sound as

$$\rho_b = \left(\frac{a_b^2}{\gamma S_b}\right)^{\frac{1}{\gamma-1}} \tag{36}$$

The energy is then calculated from the equation of state.

For supersonic free-stream conditions along inflow boundaries, quantities are extrapolated from the exterior; along outflow boundaries, quantities are extrapolated from the interior of the computational domain.

Stability Analysis

In order to examine the stability characteristics of the three-dimensional approximate-factorization algorithms considered previously, a Fourier analysis is conducted on the complete system of equations in Cartesian coordinates (refs. 8 and 25). Because of the mixed signs of the eigenvalues of the Euler equations and the fact that the three-dimensional Euler equations cannot be diagonalized to yield a system of convection equations, stability analysis of the scalar convection equation is not sufficient to determine stability properties of the three schemes. (For example, consider the two-factor eigenvalue split scheme that would reduce to only a one-factor scheme for scalar analysis.) The complete system of equations can be written as

$$\mathbf{N}\,\Delta\mathbf{Q} = -\mathbf{L} = -\Delta t\,\mathbf{R}^n \tag{37}$$

where, for Cartesian coordinates,

$$\mathbf{R}^n = \delta_x^-\mathbf{F}^+ + \delta_x^+\mathbf{F}^- + \delta_y^-\mathbf{G}^+ + \delta_y^+\mathbf{G}^- + \delta_z^-\mathbf{H}^+ + \delta_z^+\mathbf{H}^- \tag{38}$$

and $\mathbf{N}$ is an implicit operator corresponding to the scheme considered. Linearizing the residual $\mathbf{R}^n$ as

$$\mathbf{R}^n = \mathbf{A}^+\delta_x^-\mathbf{Q}^n + \mathbf{A}^-\delta_x^+\mathbf{Q}^n + \mathbf{B}^+\delta_y^-\mathbf{Q}^n + \mathbf{B}^-\delta_y^+\mathbf{Q}^n + \mathbf{C}^+\delta_z^-\mathbf{Q}^n + \mathbf{C}^-\delta_z^+\mathbf{Q}^n \tag{39}$$

and assuming that the Jacobians are locally constant allows the stability to be analyzed by letting

$$\mathbf{Q}^n = \lambda^n\mathbf{U}_0\,\exp(i\beta_s x)\,\exp(i\gamma_s y)\,\exp(i\alpha_s z) \tag{40}$$

where $\mathbf{U}_0$ is an initial constant vector. Upon substitution into equation (37) and dividing out the common factors, the generalized eigenvalue problem for λ, which is the vector of amplification factors, can be obtained as

$$\left(\hat{\mathbf{N}} - \hat{\mathbf{L}}\right)\mathbf{v} = \hat{\mathbf{N}}\lambda\mathbf{v} \tag{41}$$

where $\hat{\mathbf{N}}$ and $\hat{\mathbf{L}}$ are the Fourier symbols of $\mathbf{N}$ and $\mathbf{L}$, respectively. The stability characteristics are determined by cycling through a fixed number of each of the spatial frequencies, in this case 16 frequencies, in the range

$$0 \le \beta\,\Delta_x,\ \gamma\,\Delta_y,\ \alpha\,\Delta z \le 2\pi$$

for a series of CFL numbers between 0.1 and 50. Each time the generalized eigenvalue problem is solved using a routine from the International Mathematics and Statistics Library (IMSL). Each time the maximum eigenvalue, the average eigenvalue, and the smoothing factor are determined. The smoothing factor is defined as

$$\mu = \max(|\lambda|) \tag{42a}$$

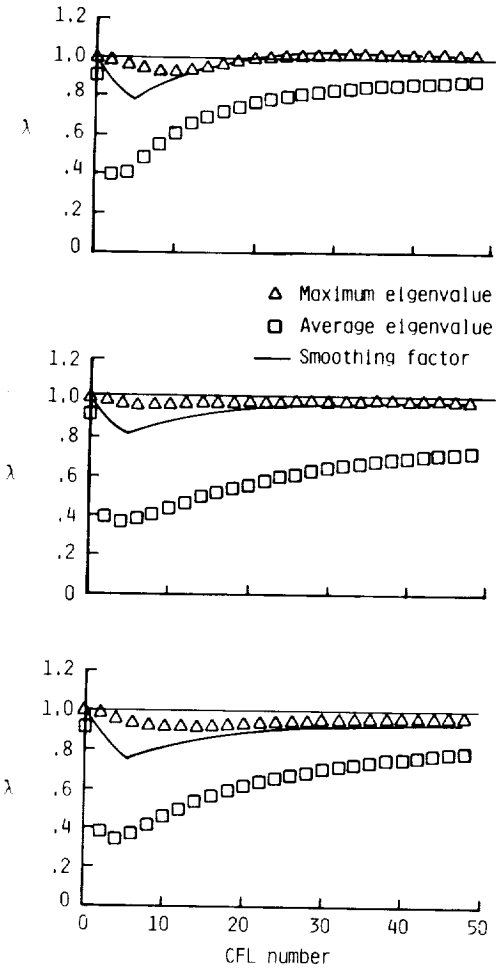
when

$$\pi/2 \leq \max(\beta_s \, \Delta x, \; \gamma_s \, \Delta y, \; \alpha_s \, \Delta z) \leq 3\pi/2 \tag{42b}$$

This corresponds to the damping of the high frequencies and serves as an indication of how effectively the multigrid procedure can accelerate convergence for a given scheme.

Results are shown using the Van Leer splitting for each of the factorization schemes given previously. Identical cases were run using the Steger-Warming splittings with little change in the results, and these are therefore not shown. Each result was obtained by using first-order differencing on the implicit side of the equation and fully upwind, second-order differencing for the residual computations. All the calculations assume Cartesian coordinates, a Mach number of 0.8, and 0° yaw and angle of attack.

The average eigenvalue, the smoothing factor, and the maximum eigenvalue are shown in figure 6 for the three schemes given previously. For the three-factor, spatially split scheme shown in figure 6(a), the maximum eigenvalue indicates that this scheme is conditionally stable with a maximum CFL number



(a) Three-factor spatially split.

(b) Two-factor eigenvalue split.

(c) Two-factor combination split.

Figure 6. Stability analysis of three-dimensional approximate factorization schemes. $M_\infty = 0.8$; $\alpha = 0°$.

of approximately 20. Note that this is contrary to the unconditional instability obtained when central differencing is used with the same algorithm (ref. 27). The minimum smoothing factor occurs at a CFL

of about 5 which is somewhat less than the CFL number where the maximum eigenvalue is lowest. In contrast to the spatially split algorithm, both of the two-factor schemes shown in figures 6(b) and 6(c) appear to be stable for all CFL numbers considered. The maximum eigenvalues and smoothing factors also exhibit less sensitivity to the CFL number than the three-factor scheme with minimum smoothing factors also occurring at a CFL number of about 5. However, the two-factor eigenvalue split scheme has a somewhat higher smoothing factor than the other two schemes. This indicates that the three-factor spatially split and the two-factor combination split algorithms are more appropriate for multigrid applications than the two-factor eigenvalue split scheme.

Multigrid Algorithms

General Algorithm

The multigrid method used in the current study is the full-approximation scheme (FAS) that appears in many references (refs. 10, 28, 29, and 30) and is summarized below. It is most easily understood by first considering the solution of a general nonlinear equation

$$\mathbf{L}(\mathbf{Q}) = \mathbf{S} \tag{43}$$

Equation (43) is to be solved numerically by dividing the domain into discrete cells yielding a system of equations to be solved simultaneously at each point as

$$\mathbf{L}_N(\mathbf{Q}_N) = \mathbf{S}_N \tag{44}$$

where $\mathbf{Q}_N$ is the exact solution to the discretized system and $\mathbf{L}_N$ is the discrete analog of the operator $\mathbf{L}$. If initial conditions are close enough to the final solution, equation (44) could be solved iteratively by using Newton iteration. This approach however may be prohibitively expensive if the number of unknowns is large as typically occurs in multidimensional problems. Many other iterative schemes have therefore been devised that require significantly fewer operations. After a few iterations, however, these methods generally exhibit a slow convergence rate, thus reducing the residuals by a very small amount each time (ref. 29). The reason for the slow asymptotic convergence rate is inadequate damping of the low-frequency errors (ref. 11).

The multigrid method efficiently damps the low-frequency errors by using a sequence of grids $G_0, G_1, ..., G_N$, where G_N denotes the finest grid from which successively coarser grids can be formed by deleting every other mesh line in each direction. In this context, the high-frequency-error components on a given grid are those that cannot be resolved on the next coarser mesh because of the increased grid spacing. If an iterative method is chosen that quickly damps the high-frequency errors on a given grid, then the remaining errors will be only the low-frequency smooth components after a few iterations. A sequence of coarser grids can then be used to accelerate the convergence rate on the finest grid by reducing the remaining low-frequency errors since some of these same frequencies appear as high-frequency errors on a coarser grid. Therefore, the errors on the fine grid that are usually responsible for slow convergence are quickly damped when using the coarser grids where the computations are relatively inexpensive.

In order to use the coarser grids, it is necessary to obtain an equation on the fine mesh that can be accurately represented by the coarser mesh. It is important to realize first that neither the solution nor the high-frequency-error components on the fine grid can generally be resolved on a coarser grid. The high-frequency errors however can be sufficiently damped on a fine grid by using appropriate iterative schemes so that the remaining errors will be composed of only smooth, low-frequency components that can be adequately represented on coarser meshes. For this reason, it is necessary to obtain an equation on the fine mesh in terms of the errors.

When solving in an iterative fashion, equation (44) is solved approximately at each step as

$$\mathbf{L}_N(\mathbf{q}_N^c) = \mathbf{S}_N + \mathbf{R}_N \tag{45}$$

where $\mathbf{q}_N^c$ is the most current approximation to $\mathbf{Q}_N$ and $\mathbf{R}_N$ is the residual that will be zero only when $\mathbf{q}_N^c = \mathbf{Q}_N$ and, hence, the exact solution is obtained. Subtracting equation (45) from equation (44) yields

an equation on the finest grid in terms of the residual

$$\mathbf{L}_N(\mathbf{Q}_N) - \mathbf{L}_N(\mathbf{q}_N^c) = -\mathbf{R}_N \tag{46}$$

If the high-frequency errors have been previously smoothed, then the fine-grid residual equation (46) can be adequately approximated on a coarser mesh by

$$\mathbf{L}_{N-1}(\mathbf{Q}_{N-1}) = \hat{\mathbf{I}}_N^{N-1}(-\mathbf{R}_N) + \mathbf{L}_{N-1}(\mathbf{I}_N^{N-1}\mathbf{q}_N^c) \tag{47}$$

where $\mathbf{I}_N^{N-1}$ and $\hat{\mathbf{I}}_N^{N-1}$ are restriction operators for transferring the dependent variables and the residual from the fine grid to the coarse grid, respectively. Here, $\mathbf{I}_N^{N-1}\mathbf{q}_N^c$ serves as an initial approximation to the solution on the coarse mesh, whereas $\mathbf{Q}_{N-1}$ is the exact solution which is the sum of the initial approximation and a correction (ref. 30). Since the full solution is computed and stored on each grid level as opposed to only the corrections, this process is referred to as the full-approximation scheme (FAS).

On a sufficiently coarse grid, equation (47) can be solved exactly using a variety of numerical techniques to obtain $\mathbf{Q}_{N-1}$ from which the correction can be formed as

$$\mathbf{V}_{N-1} = \mathbf{Q}_{N-1} - \mathbf{I}_N^{N-1}\mathbf{q}_N^c \tag{48}$$

This can then be passed up to the fine grid and used as a correction to $\mathbf{q}_N^c$, which is replaced by its previous value plus the correction

$$\mathbf{q}_N^c \leftarrow \mathbf{q}_N^c + \hat{\mathbf{I}}_{N-1}^{N}\mathbf{V}_{N-1} \tag{49}$$

This process yields a simple FAS two-level algorithm where the operations on the coarse grid (eqs. (47) to (49)) that are used to update the fine-grid solution are termed the coarse-grid correction. Often, however, the exact solution of equation (47) can still be quite expensive to obtain. Also, since the correction on the coarse grid serves only as an approximation to the fine-grid correction, the exact solution of equation (47) is not required. Therefore, instead of solving equation (47) to completion, several iterations can be carried out to get a reasonable approximation to $\mathbf{Q}_{N-1}$. After each iteration of equation (47), the equation satisfied by $\mathbf{q}_{N-1}^c$ is given by

$$\mathbf{L}_{N-1}(\mathbf{q}_{N-1}^c) = \hat{\mathbf{I}}_N^{N-1}(-\mathbf{R}_N) + \mathbf{L}_{N-1}(\mathbf{I}_N^{N-1}\mathbf{q}_N^c) + \mathbf{R}_{N-1} \tag{50}$$

which differs from the solution of equation (47) only by the residual term $\mathbf{R}_{N-1}$ which will be zero when $\mathbf{q}_{N-1}^c = \mathbf{Q}_{N-1}$. If the errors are smooth, then subtraction of equation (50) from equation (47) yields an equation that can be well-represented on yet a coarser mesh, G_{N-2}. Writing this equation on G_{N-2} yields

$$\mathbf{L}_{N-2}(\mathbf{Q}_{N-2}) = \mathbf{L}_{N-2}(\mathbf{I}_{N-1}^{N-2}\mathbf{q}_{N-1}^c) + \hat{\mathbf{I}}_{N-1}^{N-2}(-\mathbf{R}_{N-1}) \tag{51}$$

where equation (50) is used to determine $\mathbf{R}_{N-1}$. The solution may be obtained by solving equation (51) exactly, by approximating by several iterations, or by introducing more coarse-grid levels. On all coarse grids, one or more FAS cycles (smoothing followed by coarse-grid correction) are completed. In this manner, each of the coarse meshes is used to obtain a correction for the solution on the next finer mesh. Since only the equations for smooth error components may be represented well on coarser grids, it is important to only pass corrections and not the full solution from a coarse grid up to the next finer one (ref. 28).

Using equation (45), note that equation (47) can be recast

$$\mathbf{L}_{N-1}(\mathbf{Q}_{N-1}) = \mathbf{S}_{N-1} + \tau_{N-1} = \mathbf{P}_{N-1} \tag{52}$$

where

$$\mathbf{S}_{N-1} = \hat{\mathbf{I}}_N^{N-1}\mathbf{S}_N \tag{53}$$

$$\tau_{N-1} = \mathbf{L}_{N-1}(\mathbf{I}_N^{N-1}\mathbf{q}_N^c) - \hat{\mathbf{I}}_N^{N-1}[\mathbf{L}_N(q_N^c)] \tag{54}$$

Here, τ_{N-1} is the relative truncation error (or defect correction) between the grids so that the solution on the coarse grid is driven by the fine grid, and the defect correction accounts for the difference in truncation error between the coarse and fine grids (ref. 28). The analogous equation for grid G_{N-2} is given by

$$\mathbf{L}_{N-2}(\mathbf{Q}_{N-2}) = \mathbf{S}_{N-2} + \tau_{N-2} \tag{55}$$

where

$$\mathbf{S}_{N-2} = \hat{\mathbf{I}}_{N-1}^{N-2}\mathbf{S}_{N-1} \tag{56}$$

and

$$\tau_{N-2} = \mathbf{L}_{N-2}\left(\mathbf{I}_{N-1}^{N-2}\mathbf{q}_{N-1}^{c}\right) - \hat{\mathbf{I}}_{N-1}^{N-2}\left[L_{N-1}(\mathbf{q}_{N-1}^{c})\right] + \hat{\mathbf{I}}_{N-1}^{N-2}\tau_{N-1} \tag{57}$$

Note that the relative truncation error on the $N-2$ grid is the sum of the relative truncation error between grids N and $N-1$, as well as $N-1$ and $N-2$.

Algorithm for Euler Equations

For the steady Euler equations in generalized coordinates, equation (44) can be written as

$$\mathbf{L}_N(\mathbf{Q}_N) = \delta_\xi^-\hat{\mathbf{F}}^+ + \delta_\xi^+\hat{\mathbf{F}}^- + \delta_\eta^-\hat{\mathbf{G}}^+ + \delta_\eta^+\hat{\mathbf{G}}^- + \delta_\varsigma^-\hat{\mathbf{H}}^+ + \delta_\varsigma^+\hat{\mathbf{H}}^- = 0 \tag{58}$$

In the multigrid-solution process, a forcing function arises on the coarse grids from restricting the residual equation on a fine mesh down to the coarser one. Since $\mathbf{S} = 0$ in equation (44) for the Euler equations, the forcing function is the relative truncation error between the two meshes. The resulting equation to be solved on any mesh $\mathbf{G}_i$ can be written as

$$\mathbf{L}_i(\mathbf{Q}_i) = \tau_i \tag{59}$$

where $\tau_i = 0$ on the finest mesh and is the relative truncation error on each of the coarser grids. The solution of equation (59) is updated by introducing a time derivative of the dependent variables to the left-hand side so that the solution can be advanced in time using the approximate-factorization methods previously described. The resulting scheme written on mesh G_i is given by

$$\mathbf{N}\,\Delta\mathbf{q}_i^c = -\Delta t\left[\mathbf{L}_i(\mathbf{q}_i^c) - \tau_i\right] = -\Delta t\,\mathbf{R}_i \tag{60}$$

where $\mathbf{N}$ is the implicit operator of the scheme considered and $\mathbf{L}_i(\mathbf{q}_i^c)$ on the right-hand side is due to the linearization of $\mathbf{L}_i(\mathbf{Q}_i)$ from the backward Euler time integration. Note that even on the coarse meshes where τ_i is nonzero, equation (60) maintains the same form as the equation on the fine mesh. The result of this is that the coarse meshes can be updated using the same scheme as that used on the fine mesh with only a slight modification to the right-hand side.

Several strategies exist for deciding when to switch from one grid level to another, and they generally fall under the categories of fixed- or adaptive-cycling algorithms. The strategy used in the present study is a fixed-cycling strategy in which each global cycle consists of a set number of FAS cycles on each of the coarser grids. Recall that one FAS cycle on any grid consists of a smoothing step followed by a coarse-grid correction. A predetermined number of iterations are performed on each grid level to smooth the errors.

The conserved variables are transferred to the next coarser grids each time by the rule

$$\mathbf{Q}_{i-1} = \mathbf{I}_i^{i-1}\mathbf{Q}_i \tag{61}$$

where $\mathbf{I}_i^{i-1}$ is a volume-weighted restriction operator that transfers values on the fine grid to the coarser one and is defined by

$$\mathbf{I}_{i-1}^{i}\mathbf{Q}_i = \sum V\mathbf{Q}\Big/\sum V \tag{62}$$

and the summations are taken over all the fine-grid cells that make up the coarse-grid cell. As shown in appendix C, restriction of the dependent variables in this manner conserves mass, momentum, and energy

in each of the cell volumes. The relative truncation error is calculated on the coarse grid as

$$\tau_{i-1} = \mathbf{L}_{i-1}(\mathbf{I}_i^{i-1}\mathbf{q}_i^c) - \hat{\mathbf{I}}_i^{i-1}\mathbf{R}_i \tag{63}$$

where $\hat{\mathbf{I}}_i^{i-1}$ is the restriction operator for the residual defined as

$$\hat{\mathbf{I}}_i^{i-1}\mathbf{R}_i = \sum \mathbf{R}_i \tag{64}$$

where, again, the summation is over the cells on the fine grid that make up the coarse-grid cell. By summing the residuals, the surface integral of the fluxes crossing the cell boundaries on the coarse grid is the same as would arise by integrating around all the fine-grid cells making up the coarse grid. (See appendix C.) Several iterations of the approximate factorization scheme can be conducted to get an approximation to the steady solution on G_{i-1} with the right-hand side modified to include the relative truncation error. If only one coarse grid is used to correct the finest grid, the result is the simple FAS two-level cycle. On the other hand, if more grid levels are introduced so that one or more FAS cycles can be recursively carried out on each subsequent coarse-grid level to get a better approximation to $\mathbf{Q}_{N-1}$, then a multilevel algorithm results. When only one FAS cycle is carried out for each of the coarser grids, the resulting global cycling strategy is termed a V-cycle and is depicted in figure 7. Another cycling strategy of interest, which is shown in figure 8, is termed a W-cycle and results when two FAS cycles are used on each of the coarser meshes. Results will be shown in the next section using both types of cycles where, on the coarsest mesh, three smoothing iterations are performed in lieu of solving the equations exactly on the coarsest mesh. The corrections on coarse meshes are passed to the next finer mesh using trilinear interpolation with no additional iteration steps between meshes. When a W-cycle is used, however, note that an iteration is carried out at the beginning of each FAS-cycle correction in order to smooth the high frequencies.

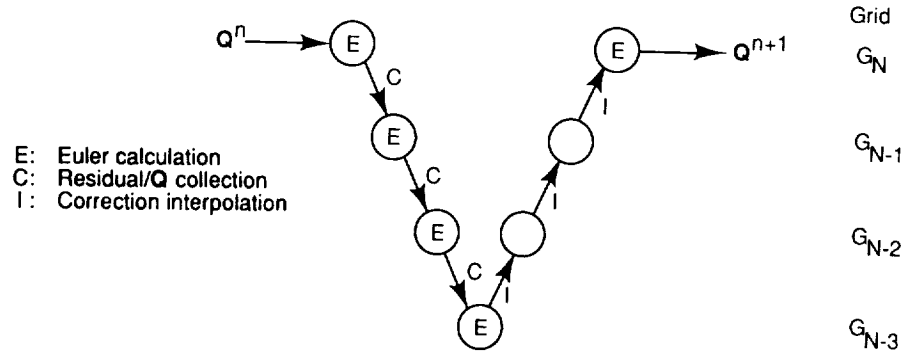


Figure 7. Multigrid V-cycle.

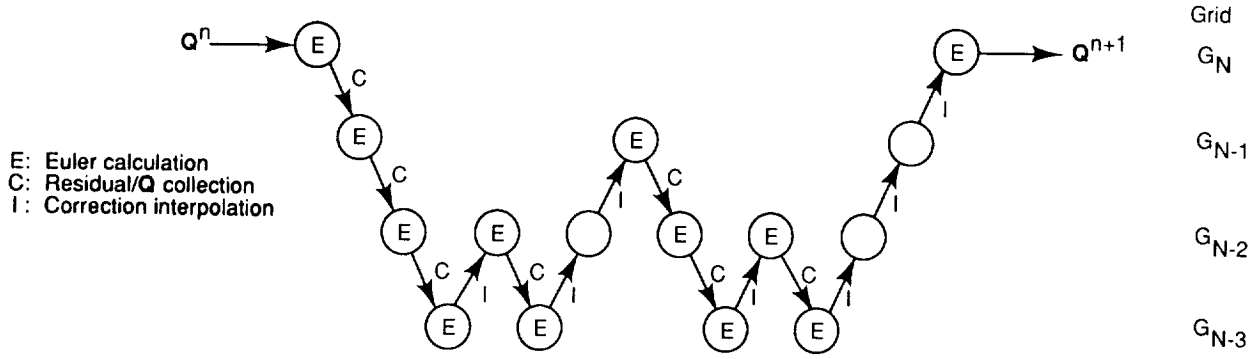


Figure 8. Multigrid W-cycle.

In order to clarify the multigrid procedure further, the overall process is summarized as follows for an exemplary case where three grid levels are used in a V-cycle:

1. Start on the finest grid and smooth the errors by doing one iteration of equation (60) with $\tau_i = 0$.
2. Calculate the residual on the fine grid from equation (45) where $\mathbf{L}_N(\mathbf{q}_N^c)$ is given on the right-hand side of equation (58) and $\mathbf{S}_N = 0$.
3. Restrict the dependent variables to the first coarse grid G_{N-1} by using equation (61).
4. Restrict the residual from the finest grid to G_{N-1} using equation (64) and calculate the relative truncation error using equation (63).
5. Calculate the right-hand side of equation (60) and update the solution on mesh G_{N-1}. (This serves to smooth the errors on this grid so that a coarser grid can be introduced.)
6. Calculate the residual on this mesh using equation (50). Note that this can be written as

$$\mathbf{R}_{N-1} = \mathbf{L}_{N-1}(\mathbf{q}_{N-1}^c) - \tau_{N-1} \tag{65}$$

Since τ_{N-1} has been previously calculated, the residual is easily calculated by simply calculating $\mathbf{L}_{N-1}(\mathbf{q}_{N-1}^c)$ from the most current values of the dependent variables on the mesh and subtracting τ_{N-1}.

7. Restrict the dependent variables on G_{N-1} to G_{N-2} by using equation (61).
8. Restrict the residual from equation (65) to the $N-2$ grid and calculate τ_{N-2} from equation (63).
9. Calculate the right-hand side of equation (60) and update the solution on this mesh. Since this is the coarsest mesh used in the present example, three iterations of equation (60) are undertaken to get an approximation to $\mathbf{Q}_{N-2}$. During each step, the right-hand side is updated to use the most current values of the dependent variables in $\mathbf{L}_{N-2}(\mathbf{q}_{N-2}^c)$. Note that τ_{N-2} will not change.
10. Calculate the correction on this mesh to give

$$\mathbf{V}_{N-2} = \mathbf{q}_{N-2}^c - \mathbf{I}_{N-1}^{N-2}\mathbf{q}_{N-1}^c \tag{66}$$

11. Pass the correction to the next finest mesh using trilinear interpolation and update the solution to give

$$\mathbf{q}_{N-1}^c \leftarrow \mathbf{q}_{N-1}^c + \mathbf{I}_{N-2}^{N-1}\mathbf{V}_{N-2} \tag{67}$$

Note that steps 5 to 11 make up one FAS cycle on grid $N-1$ where steps 6 to 11 constitute a coarse-grid correction. At this point, if a W-cycle were being employed, another FAS cycle (steps 5 to 11) would be repeated to update $\mathbf{q}_{N-1}^c$ further.

12. Calculate the correction on the $N-1$ mesh as

$$\mathbf{V}_{N-1} = \mathbf{q}_{N-1}^c - \mathbf{I}_N^{N-1}\mathbf{q}_N^c$$

13. Pass this correction to the finest mesh and update the solution to give

$$\mathbf{q}_N^c \leftarrow \mathbf{q}_N^c + \mathbf{I}_{N-1}^N\mathbf{V}_{N-1}$$

14. Perform one smoothing iteration using equation (60) to smooth the errors.

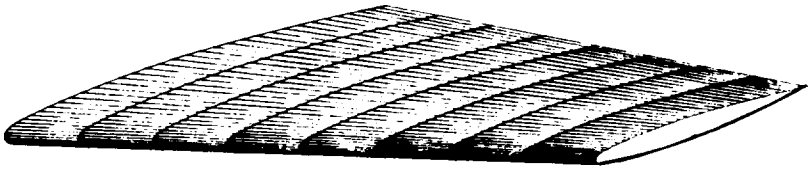
Presentation of Results

ONERA M6 Wing

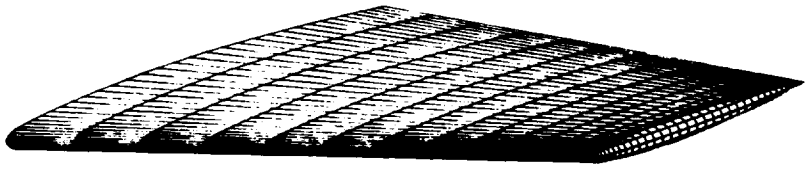
Three-dimensional subsonic- and transonic-flow computations over the ONERA M6 wing are now presented. Comparisons are made with experimental data at a Reynolds number of 11.7×10^6 (ref. 31), corresponding to conditions for which viscous effects are relatively small. The wing consists of symmetrical airfoil sections with a planform swept 30° along the leading edge, an aspect ratio of 3.8, and a taper ratio of 0.56. Solutions are obtained for two mesh types, C-H and C-O, both of which are C-type mesh topologies around the airfoil profile. The C-H mesh has uniform spacing in the spanwise direction, whereas the C-O mesh wraps around the wingtip, consequently leading to a more precise definition of the actual

rounded-tip geometry tested in the experiment. The C-O mesh has been generated with a transfinite interpolation procedure given in reference 32. The C-H mesh was obtained by simply stacking a series of two-dimensional cross sections along the span. The surface meshes for both are shown in figure 9.

The first computation is on the ONERA M6 wing at transonic conditions: a Mach number of 0.84 and an angle of attack of 3.06°. Figure 10 shows the effect of using multigrid on the residual and lift coefficient histories for a mesh with over 210 000 points. The mesh is a 193 × 33 × 33 C-H mesh. This corresponds to 193 points along the airfoil and wake (110 of which are on the airfoil), 33 points approximately normal to the airfoil, and 33 points in the spanwise direction (17 of which are on the wing planform). For this case, the Van Leer splittings are used with a V-cycle and four grid levels (a fine grid and three coarser ones). The multigrid method is very effective in accelerating convergence of both the residual and lift coefficients. The residual is reduced to "machine zero" in 400 cycles, whereas the single-grid method has reduced the residual only between one and two orders of magnitude. The benefit of multigrid is especially pronounced in the lift coefficient history where the lift coefficient value is obtained to within 0.1 percent of its final value in 41 cycles. This is a dramatic improvement over the single-grid result that required more than 400 iterations to converge to the same level of accuracy for the lift coefficient. It should be noted that for all the cases considered, several multigrid V-cycles (usually five) were run with first-order spatial differencing before switching to second order.



(a) 97 × 17 × 17 C-H mesh.



(b) 97 × 17 × 17 C-O mesh.

Figure 9. Surface mesh for ONERA M6 wing.

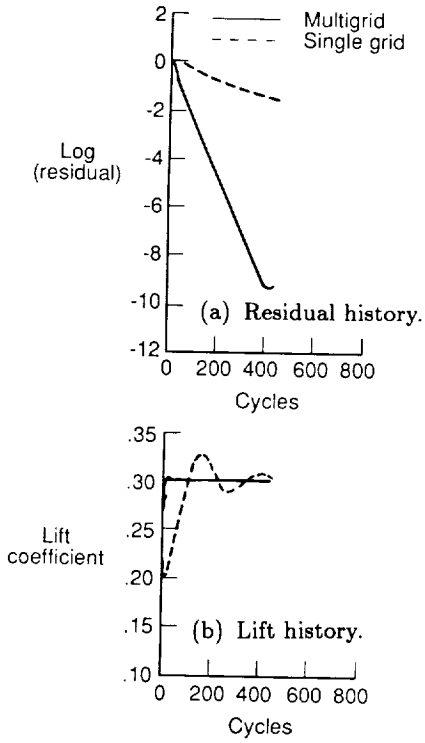


Figure 10. Effect of multigrid on convergence on ONERA M6 wing with 193 × 33 × 33 C-H mesh. $M_\infty = 0.84$; $\alpha = 3.06°$.

A comparison of convergence rates between the three schemes discussed earlier is shown in figure 11 for identical conditions as given above with the exception that only every other point from the $193 \times 33 \times 33$ mesh is used, thus resulting in a $97 \times 17 \times 17$ C-H mesh. For this mesh size, only two coarser grids are used. The three-factor, spatially split algorithm demonstrates a faster rate of convergence than either of the two-factor schemes. The spectral radius indicating the error reduction per cycle is 0.898. The two-factor scheme in which the implicit operator is split according to the sign of the eigenvalues displays the slowest convergence rate with a spectral radius of 0.93. It should be pointed out however that even though the spectral radius using this scheme is not as good as that for the spatially split scheme, this still represents a good improvement over a corresponding single-grid spectral radius of 0.98. All the runs on the $97 \times 17 \times 17$ meshes were made at a CFL number of 7. This was determined experimentally to be nearly optimum and agrees well with the CFL number for best smoothing predicted by the stability analysis.

In 64-bit precision on the VPS-32, the computational rate using a three-grid V-cycle for the three-factor scheme is about 75 μsec per grid per cycle, whereas the two-factor eigenvalue-split and combination-split schemes exhibit computational rates of 140 and 85 μsec per grid point per cycle, respectively. The computational rates obtained using one grid level are decreased about 15 percent from those using three grid levels. Computational times are decreased approximately 40 percent when the computations are done in 32-bit precision on the VPS-32. Because of the higher performance of the three-factor spatially split algorithm in both convergence rate and computational rate, it is used exclusively in the results that follow.

A comparison of the residual convergence histories for a W-cycle and the previously used V-cycle is shown in figure 12 for the $97 \times 17 \times 17$ C-H mesh. An improvement in the convergence rate using a W-cycle is apparent. In addition, the lift coefficient is obtained to within 0.3 percent of its final value in only 14 cycles and to under 0.1 percent in 24 cycles. This is an improvement over the V-cycle that took 37 cycles to get the error in lift below 0.1 percent. Although the work involved for a W-cycle is more than that for the V-cycle because of the extra smoothing iterations on the coarser grids, the time required per cycle increased only about 13 percent over a V-cycle. Therefore, even though more work is involved for each cycle, a net gain is achieved by employing the W-cycle. A summary of results for this case is given in table II for $193 \times 33 \times 33$ and $97 \times 17 \times 17$ grid points for both C-H and C-O types of grids. This table includes the spectral radius based on cycles, the number of cycles required to obtain the lift coefficient to within 0.3 and 0.1 percent of its final value, and the number of cycles required to obtain the lift to five significant digits. Note that the number of cycles required for the W-cycle to obtain the lift coefficient is relatively insensitive to the number of grid points, thus indicating that the multigrid algorithm is working successfully.

Figure 13 shows the upper-surface pressure distributions on the $193 \times 33 \times 33$ C-H mesh as well as on the $193 \times 33 \times 33$ C-O mesh. The wing under these conditions exhibits both a swept shock emanating from the root leading edge and a nearly normal shock emanating from the root. Both shocks coalesce at about 80 percent of the span to form a single shock. Figure 14 shows upper-surface pressure contours for the C-O mesh.

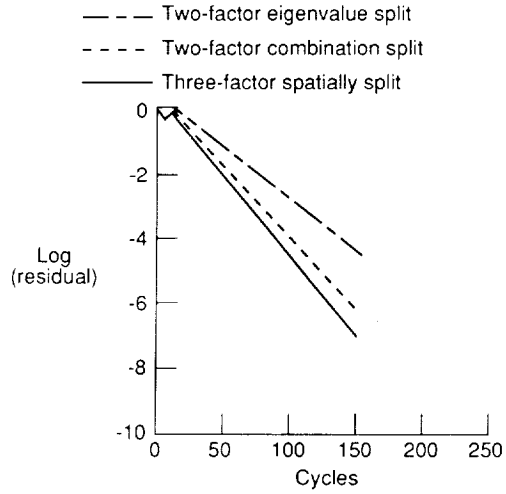


Figure 11. Comparison of convergence rate for the three schemes on ONERA M6 wing with 97 × 17 × 17 C-H mesh. $M_\infty = 0.84$; $\alpha = 3.06°$.

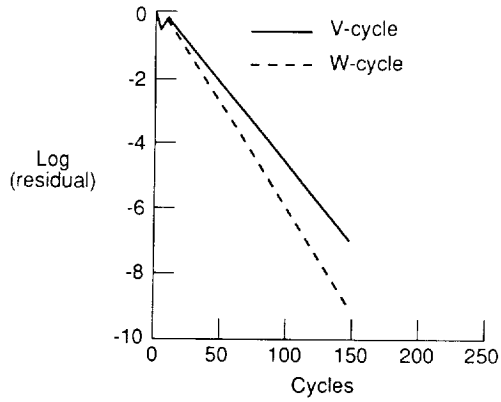
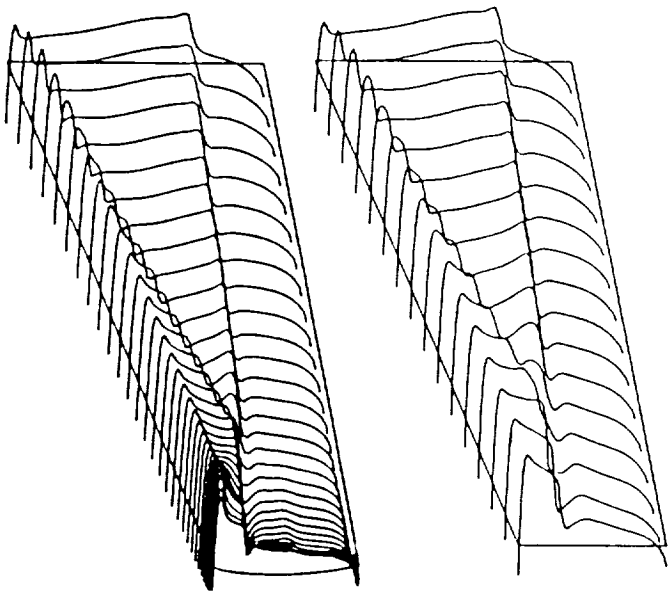


Figure 12. Comparison of convergence rate for V-cycle and W-cycle on ONERA M6 wing with 97 × 17 × 17 C-H mesh. $M_\infty = 0.84$; $\alpha = 3.06°$.



(a) 193 × 33 × 33 C-O mesh.

(b) 193 × 33 × 33 C-H mesh.

Figure 13. Upper-surface variation of pressure coefficient on ONERA M6 wing. $M_\infty = 0.84$; $\alpha = 3.06°$.

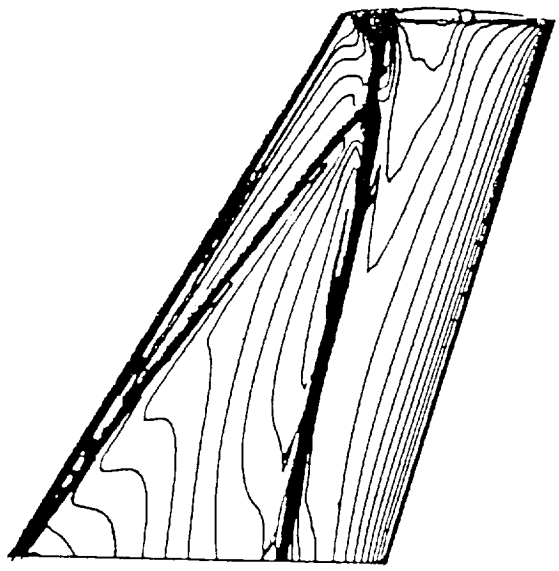


Figure 14. Pressure coefficient contours on upper surface on ONERA M6 wing. $M_\infty = 0.84$; $\alpha = 3.06°$.

Table II. Summary of Results for ONERA M6 Wing With $M_\infty = 0.84$ and $\alpha = 3.06°$

Type of cycle	Number of cycles required to obtain c_l within—			Spectral radius
	0.3 percent of final value	0.1 percent of final value	Five decimal places	
$97 \times 17 \times 17$ C-H mesh size				
V-cycle	20	37	75	0.898
W-cycle	14	24	42	0.871
$97 \times 17 \times 17$ C-O mesh size				
V-cycle	34	45	91	0.912
W-cycle	15	27	44	0.879
$193 \times 33 \times 33$ C-H mesh size				
V-cycle	37	41	153	0.948
W-cycle	12	23	47	0.923
$193 \times 33 \times 33$ C-O mesh size				
V-cycle	27	68	149	0.952
W-cycle	14	19	47	0.926

In figure 15, pressure coefficients obtained using the Van Leer splitting on both the $97 \times 17 \times 17$ and $193 \times 33 \times 33$ C-O meshes are compared with experimental data at six spanwise locations. The computations are obtained at the experimental spanwise locations by linear interpolation. The computations on both meshes agree reasonably well with experiment for each spanwise location; the effect of the finer mesh is to resolve the leading-edge suction pressures and shock positions and improve the agreement with experiment at the most outboard station. Results obtained with the Steger-Warming splitting on the same two meshes are compared with experimental data in figure 16. The computations are nearly identical to the previous ones with small differences occurring near the shock regions.

The next three-dimensional test case is the ONERA M6 wing at a free-stream Mach number of 0.699 and an angle of attack of 3.06°. At these conditions, the flow remains subsonic over the entire wing. Results were obtained for this case on a $97 \times 17 \times 17$ C-O mesh, a $97 \times 17 \times 17$ C-H mesh, and both a $193 \times 33 \times 33$ C-H mesh and C-O mesh. Figure 17 shows the residual history for both the $97 \times 17 \times 17$ C-O and C-H meshes using the Van Leer splittings and a three-grid V-cycle. The convergence rate on the C-H mesh is slightly better than on the C-O mesh. Machine zero is reached in approximately 200 cycles for the C-H mesh and in approximately 300 cycles for the C-O mesh, corresponding to an asymptotic spectral radius of 0.891 and 0.926, respectively. For both meshes, the lift was obtained to less than 0.1 percent of its final value in less than 28 cycles and required only about 46 sec of computer time. On the $193 \times 33 \times 33$ C-H mesh, a spectral radius of 0.929 was obtained with the multigrid algorithm, whereas a spectral radius of 0.950 was obtained on the same size of C-O mesh. When using a W-cycle, a spectral radius of 0.866 is obtained for the $97 \times 17 \times 17$ C-H mesh and one of 0.891 is obtained for the C-O mesh. With the $193 \times 33 \times 33$ mesh, the spectral radius using the W-cycle is also about 0.890 for the C-H mesh and about 0.912 for the C-O mesh. A summary of results that is similar to those shown in table II is given in table III.

The pressure distributions on the $97 \times 17 \times 17$ C-O mesh and the $193 \times 33 \times 33$ C-H mesh are compared with experiment at six spanwise stations in figure 18 using the Van Leer splittings and in figure 19 using the Steger-Warming splittings. At the inboard stations, the results for both meshes are essentially identical and compare well with experiment. At the outboard station, however, the pressures computed on the C-O mesh agree much closer with experiment because of the increased resolution at the tip. Again, the Steger-Warming and Van Leer splittings give nearly identical results.

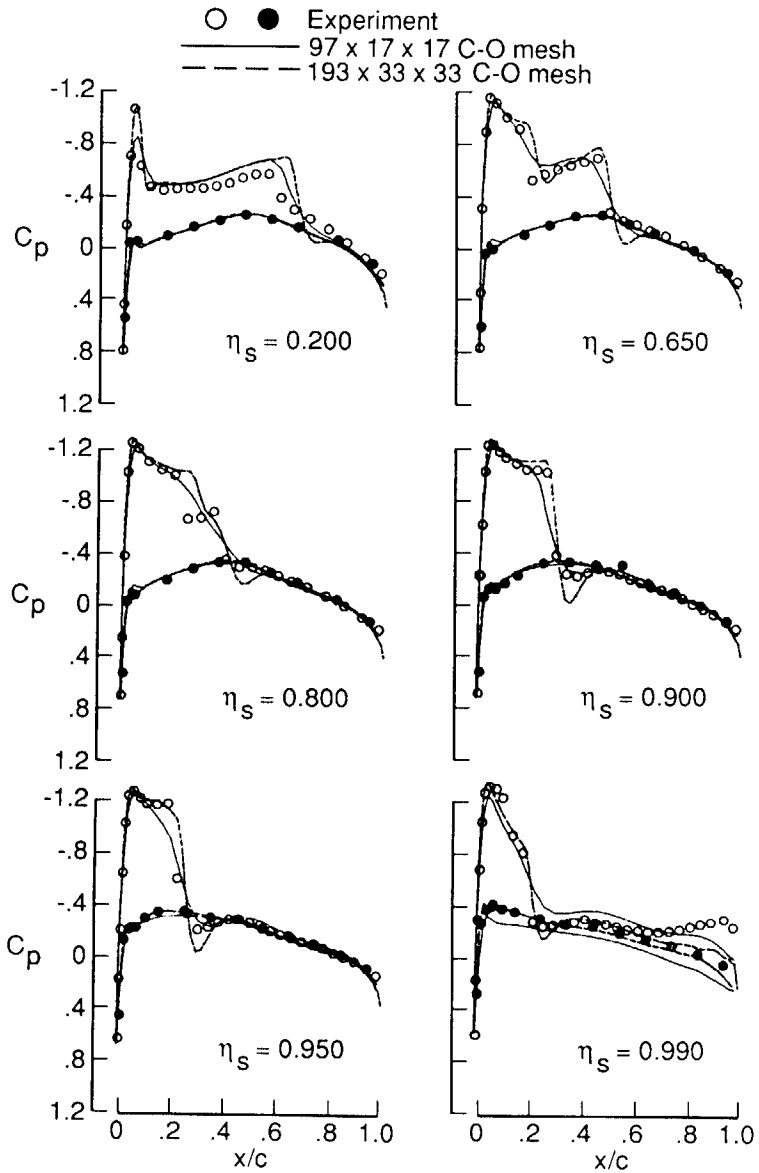


Figure 15. Comparison of experiment and inviscid calculations using Van Leer splittings on ONERA M6 wing for $M_\infty = 0.84$ and $\alpha = 3.06°$.

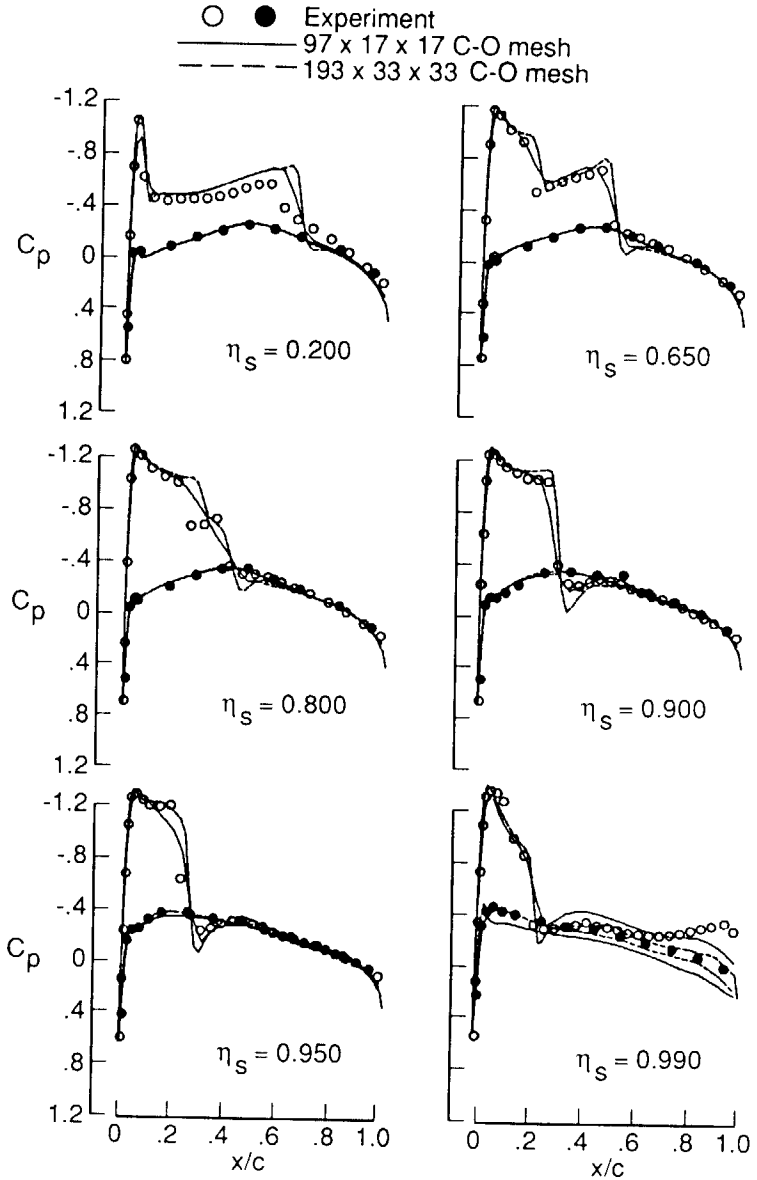


Figure 16. Comparison of experiment and inviscid calculations using Steger-Warming splittings on ONERA M6 wing for $M_\infty = 0.84$ and $\alpha = 3.06°$.

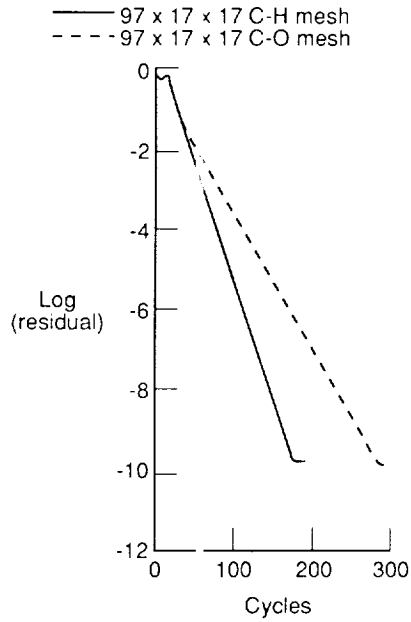


Figure 17. Multigrid convergence for C-H and C-O meshes on ONERA M6 wing for V-cycle. $M_\infty = 0.699$; $\alpha = 3.06°$.

Table III. Summary of Results for ONERA M6 Wing With $M_\infty = 0.699$ and $\alpha = 3.06°$

Type of cycle	Number of cycles required to obtain c_l within—			Spectral radius
	0.3 percent of final value	0.1 percent of final value	Five decimal places	
97 × 17 × 17 C-H mesh size				
V-cycle	19	28	38	0.891
W-cycle	11	19	33	0.866
97 × 17 × 17 C-O mesh size				
V-cycle	21	22	48	0.926
W-cycle	13	15	31	0.891
193 × 33 × 33 C-H mesh size				
V-cycle	29	37	71	0.929
W-cycle	11	21	37	0.891
193 × 33 × 33 C-O mesh size				
V-cycle	21	38	55	0.950
W-cycle	11	15	36	0.912

Analytic Forebody

The last test case considered is an analytically defined forebody for which experimental data are available at supersonic Mach numbers (ref. 33). The grid used was a 49 × 49 × 49 grid with a symmetry plane along the centerline. Figure 20(a) shows a cross section of the grid around the body, and figure 20(b) illustrates the body surface together with contours of the static density in the upper symmetry plane. The conditions correspond to a free-stream Mach number of 1.7 and an angle of attack of 0°; this leads to an oblique shock at the nose and supersonic flow over the entire length of the body. In this case, it is known that a pure axial marching scheme can obtain the solution quite efficiently, as the equations are hyperbolic

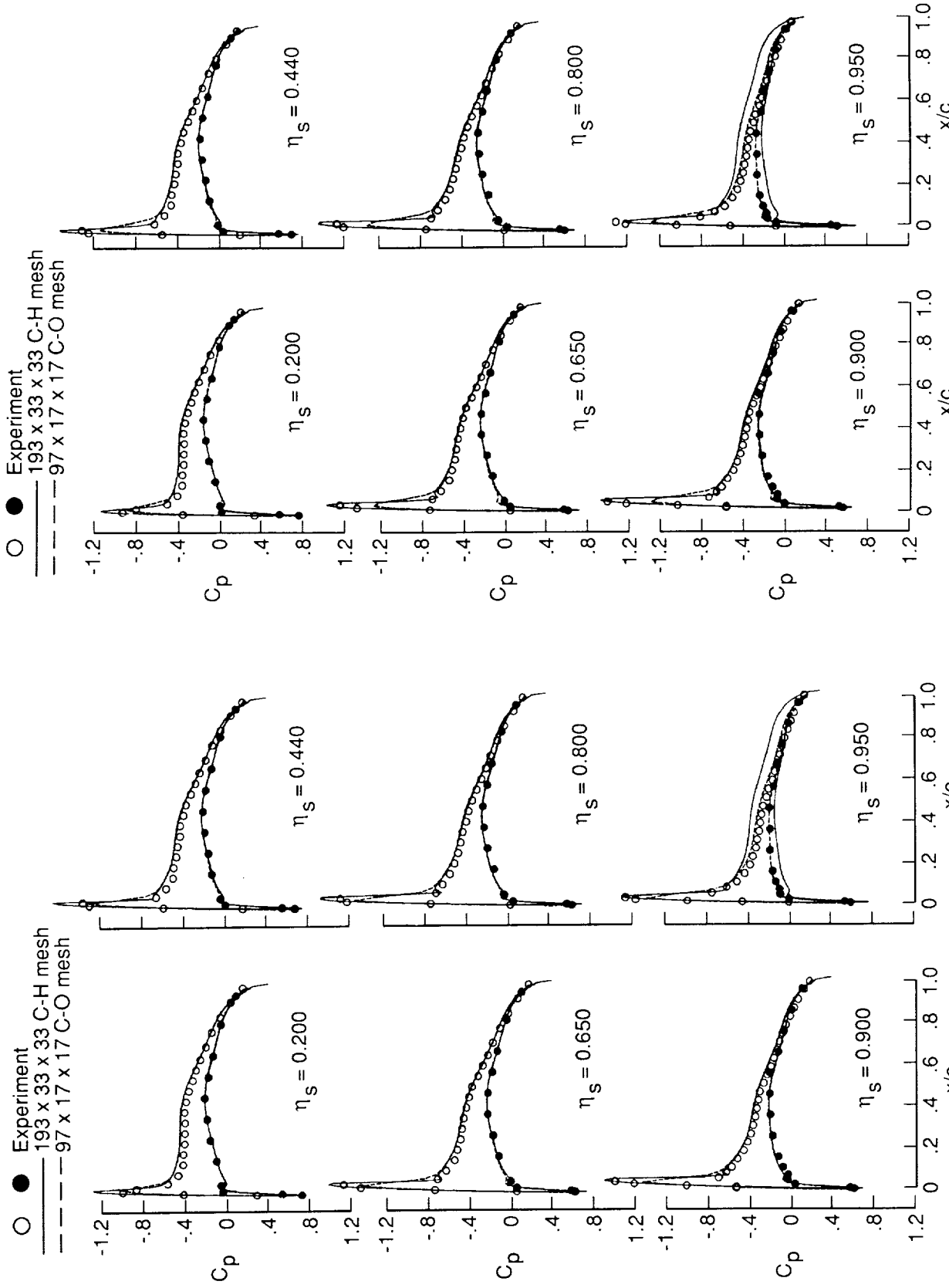
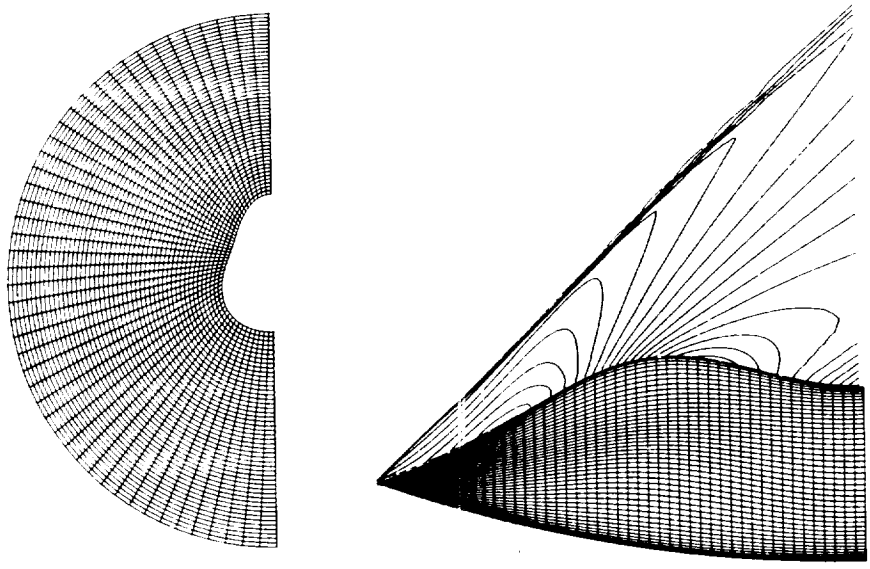


Figure 18. Comparison of experiment and inviscid calculations using Van Leer splittings on ONERA M6 wing for $M_\infty = 0.699$ and $\alpha = 3.06°$.

Figure 19. Comparison of experiment and inviscid calculations using Steger-Warming splittings on ONERA M6 wing for $M_\infty = 0.699$ and $\alpha = 3.06°$.

in the axial coordinate as long as the axial velocity is supersonic. However, if subsonic pockets of flow develop, an axial marching scheme will fail and thus must be modified to include some iterative scheme to solve the subsonic pockets. Hence, it is of interest to study the efficiency of the present multigrid algorithm in this fully supersonic flow. The occurrence of the subsonic pockets presents no difficulty to the present approach.

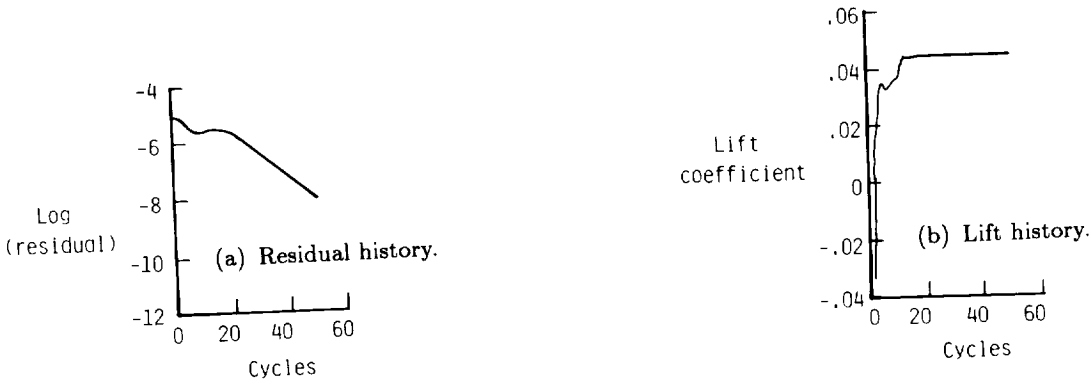


(a) Cross section of grid around body.

(b) Grid on body surface with contours of static density in upper symmetry plane.

Figure 20. Grid for analytic forebody.

The residual and lift histories obtained using a V cycle and the Van Leer splittings are shown in figure 21. As can be seen, the residual is reduced 3 orders of magnitude in only 50 cycles (the first 10 of which were first-order accurate), and an asymptotic spectral radius of 0.830 is achieved based on the last 30 cycles. The lift coefficient is obtained to within 0.3 percent of the final value in 22 cycles. The computed pressure distribution is compared with experimental data at the forebody symmetry plane in figure 22 for both the upper and lower sides of the body. The pressure coefficients compare well with the experimental data over both the lower and upper surfaces of the body.



(a) Residual history.

(b) Lift history.

Figure 21. Convergence history for analytic forebody. $M_\infty = 1.7$; $\alpha = 0°$.

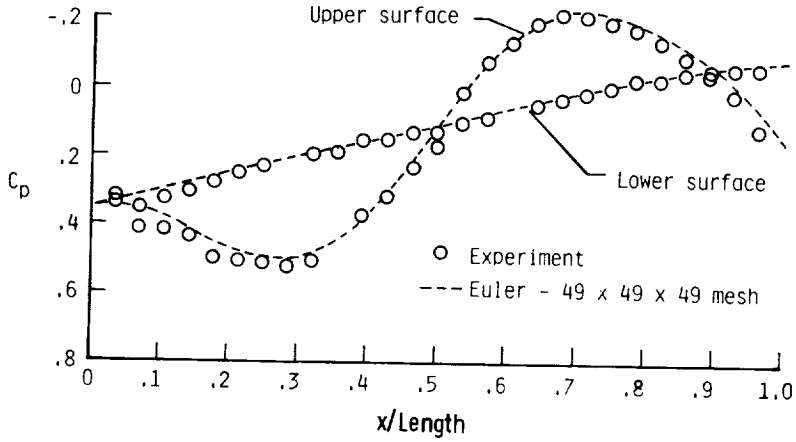


Figure 22. Comparison of calculated pressure coefficient with experiment at forebody symmetry plane. $M_\infty = 1.7$; $\alpha = 0°$.

Concluding Remarks

A multigrid algorithm has been developed, analyzed, and applied to the three-dimensional, flux-split Euler equations in generalized, body-fitted, curvilinear coordinates. Three implicit smoothing operators were studied, as well as two flux-splitting schemes. Applications were demonstrated for subsonic and transonic flows over a wing and for fully supersonic flow over an analytic forebody shape. Results were compared with experiment.

A linearized stability analysis of the three-dimensional system of difference equations representing the Euler equations was carried out numerically for a range of simulated free-stream conditions. Sixteen modes of discrete frequency in each spatial direction were examined. The results indicated that the three-factor, spatially split implicit scheme is conditionally stable for the upwind, flux-vector-splitting difference scheme. This is in contrast to the unconditional instability of the three-factor scheme with central differencing. The two-factor eigenvalue-split and combination-split schemes were found to be unconditionally stable. However, since extremely large time steps are not necessary for a multigrid smoothing algorithm, the conditional stability of the three-factor scheme is not a penalty.

Of the three implicit iteration schemes used for multigrid smoothing operators, the three-factor, spatially split operator was found to be superior to the two-factor operators. Its superiority is due to a combination of a better smoothing rate and more complete vectorization.

Two fixed-cycle multigrid methods were tested. The W-cycle was more efficient than the simpler V-cycle. Either algorithm was an order of magnitude faster than the single-grid iteration. The multigrid method proved efficient for subsonic, transonic, and fully supersonic flows.

Of the two flux-splitting schemes tested, the Steger-Warming and Van Leer splittings, both yielded similar results for pressure distributions. Although not shown, the convergence rates using the Steger-Warming splittings were very similar to those obtained with the Van Leer splittings.

NASA Langley Research Center
Hampton, Virginia 23665-5225
August 25, 1988

Appendix A

Transformation to Generalized Coordinates

The three-dimensional Euler equations in Cartesian coordinates and strong conservation-law form are given by

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0 \tag{A1}$$

where

$$\mathbf{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix} \quad \mathbf{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{Bmatrix} \quad \mathbf{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{Bmatrix} \quad \mathbf{H} = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{Bmatrix} \tag{A2}$$

and

$$p = (\gamma - 1)\left[e - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right]$$

When using the chain rule and the body-fitted coordinate system given by the steady transformation

$$\xi = \xi(x, y, z) \qquad \eta = \eta(x, y, z) \qquad \varsigma = \varsigma(x, y, z) \qquad \tau = t \tag{A3}$$

the Euler equations can be recast as

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \xi_x \frac{\partial \mathbf{F}}{\partial \xi} + \eta_x \frac{\partial \mathbf{F}}{\partial \eta} + \varsigma_x \frac{\partial \mathbf{F}}{\partial \varsigma} + \xi_y \frac{\partial \mathbf{G}}{\partial \xi} + \eta_y \frac{\partial \mathbf{G}}{\partial \eta} + \varsigma_y \frac{\partial \mathbf{G}}{\partial \varsigma} + \xi_z \frac{\partial \mathbf{H}}{\partial \xi} + \eta_z \frac{\partial \mathbf{H}}{\partial \eta} + \varsigma_z \frac{\partial \mathbf{H}}{\partial \varsigma} = 0 \tag{A4}$$

Now, again using the chain rule, the derivatives with respect to τ, ξ, η, and ς can be written in matrix form as

$$\begin{Bmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \varsigma} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x_\xi & y_\xi & z_\xi \\ 0 & x_\eta & y_\eta & z_\eta \\ 0 & x_\varsigma & y_\varsigma & z_\varsigma \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} \tag{A5}$$

from which Cramer's rule can be used to solve for the x, y, z, and t derivatives. Using these to evaluate the metric terms gives

$$\left.\begin{aligned} \xi_x &= J(y_\eta z_\varsigma - z_\eta y_\varsigma) & \eta_x &= J(z_\xi y_\varsigma - y_\xi z_\varsigma) & \varsigma_x &= J(y_\xi z_\eta - z_\xi y_\eta) \\ \xi_y &= J(z_\eta x_\varsigma - x_\eta z_\varsigma) & \eta_y &= J(x_\xi z_\varsigma - z_\xi x_\varsigma) & \varsigma_y &= J(z_\xi x_\eta - x_\xi z_\eta) \\ \xi_z &= J(x_\eta y_\varsigma - y_\eta x_\varsigma) & \eta_z &= J(x_\varsigma y_\xi - y_\varsigma x_\xi) & \varsigma_z &= J(x_\xi y_\eta - y_\xi x_\eta) \end{aligned}\right\} \tag{A6}$$

where

$$J^{-1} = x_\xi(y_\eta z_\varsigma - z_\eta y_\varsigma) - y_\xi(x_\eta z_\varsigma - z_\eta x_\varsigma) + z_\xi(x_\eta y_\varsigma - y_\eta x_\varsigma) \tag{A7}$$

In order to regain the strong conservation-law form, equation (A4) can now be multiplied through by J^{-1} and rearranged using the chain rule on certain terms. For example, the term $J^{-1}\xi_x \mathbf{F}_\xi$ can be rewritten as

$$J^{-1}\xi_x \mathbf{F}_\xi = \frac{\partial}{\partial \xi}\left[(J^{-1}\xi_x)\mathbf{F}\right] - \mathbf{F}\left[\frac{\partial}{\partial \xi}(J^{-1}\xi_x)\right] \tag{A8}$$

After rewriting all the appropriate terms and noting that many parts of the resulting equation can be shown to be zero by substituting equations (A6) for the metrics, the Euler equations can be written in generalized coordinates and maintain the strong conservation-law form as

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi} + \frac{\partial \hat{\mathbf{G}}}{\partial \eta} + \frac{\partial \hat{\mathbf{H}}}{\partial \varsigma} = 0 \tag{A9}$$

where

$$\left.\begin{aligned} \hat{\mathbf{Q}} &= J^{-1}\mathbf{Q} \\ \hat{\mathbf{F}} &= J^{-1}(\xi_x \mathbf{F} + \xi_y \mathbf{G} + \xi_z \mathbf{H}) \\ \hat{\mathbf{G}} &= J^{-1}(\eta_x \mathbf{F} + \eta_y \mathbf{G} + \eta_z \mathbf{H}) \\ \hat{\mathbf{H}} &= J^{-1}(\varsigma_x \mathbf{F} + \varsigma_y \mathbf{G} + \varsigma_z \mathbf{H}) \end{aligned}\right\} \tag{A10}$$

Substituting equations (A2) into (A10) allows the flux vectors to be further written in an alternate form as

$$\hat{\mathbf{F}} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U \end{Bmatrix} \qquad \hat{\mathbf{G}} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V \end{Bmatrix} \qquad \hat{\mathbf{H}} = \frac{1}{J}\begin{Bmatrix} \rho W \\ \rho u W + \varsigma_x p \\ \rho v W + \varsigma_y p \\ \rho w W + \varsigma_z p \\ (e+p)W \end{Bmatrix} \tag{A11}$$

where U, V, and W are the contravariant velocities defined as

$$\left.\begin{aligned} U &= \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_x u + \eta_y v + \eta_z w \\ W &= \varsigma_x u + \varsigma_y v + \varsigma_z w \end{aligned}\right\} \tag{A12}$$

Appendix B

Splitting Flux Vectors in Generalized Coordinates

The Van Leer method of splitting the flux vectors was originally given only for a Cartesian coordinate system (ref. 5). For example, the flux-split vectors in the x-direction were given in terms of the local one-dimensional Mach number $M_x = u/a$. For supersonic flow, i.e., $|M_x| \geq 1$, we have

$$\left.\begin{array}{lll} \mathbf{F}^+ = \mathbf{F} & \mathbf{F}^- = 0 & (M_x \geq 1) \\ \mathbf{F}^+ = 0 & \mathbf{F}^- = \mathbf{F} & (M_x \leq -1) \end{array}\right\} \tag{B1}$$

and for subsonic flow, i.e., $|M_x| < 1$, we have

$$\mathbf{F}^{\pm} = \left\{\begin{array}{c} \pm\rho a \left[\frac{1}{2}(M_x \pm 1)\right]^2 = f^{\pm}_{\text{mass}} \\ f^{\pm}_{\text{mass}} \left[(\gamma - 1)u \pm 2a\right]/\gamma \\ f^{\pm}_{\text{mass}} v \\ f^{\pm}_{\text{mass}} w \\ f^{\pm}_{\text{mass}} \left\{ \left[(\gamma - 1)u \pm 2a\right]^2 \Big/ \left[2(\gamma^2 - 1)\right] + (v^2 + w^2)/2 \right\} \end{array}\right\} \tag{B2}$$

For many applications, however, it is advantageous to construct generalized (body-fitted) coordinate systems of the type

$$\xi = \xi(x, y, z) \qquad \eta = \eta(x, y, z) \qquad \varsigma = \varsigma(x, y, z) \qquad \tau = t \tag{B3}$$

where, in the present work, the transformation is chosen so that the grid spacing in the computational domain is uniform and of unit length. In the discussion that follows, the superscript $(\hat{})$ indicates variables in generalized coordinates and the superscript $(^-)$ indicates variables in a localized Cartesian coordinate system. If no superscript is used, Cartesian coordinates are assumed. The strong conservation form of the Euler equations in generalized coordinates is given by

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi} + \frac{\partial \hat{\mathbf{G}}}{\partial \eta} + \frac{\partial \hat{\mathbf{H}}}{\partial \varsigma} = 0 \tag{B4}$$

For the purpose of determining a generalized splitting for $\hat{\mathbf{F}}$, only the derivatives in the ξ- and t-directions are considered, whereas the η and ς derivatives are treated as source terms. For determining the splitting of $\hat{\mathbf{F}}$, equation (B4) is transformed by a local rotation matrix in order to decompose the flux vector $\hat{\mathbf{F}}$ into components normal and tangential to a ξ = Constant cell face. The rotation matrix is given by

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \hat{\xi}_x & \hat{\xi}_y & \hat{\xi}_z & 0 \\ 0 & \hat{t}_x & \hat{t}_y & \hat{t}_z & 0 \\ 0 & \hat{r}_x & \hat{r}_y & \hat{r}_z & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{B5}$$

where $\hat{\xi}_x$, $\hat{\xi}_y$, and $\hat{\xi}_z$ are components of a unit vector $\hat{\mathbf{n}}$ normal to a ξ = Constant line. The $\hat{t}_i$ and $\hat{r}_i$ are components of vectors that are normal to $\hat{\mathbf{n}}$ and to each other so that the three vectors form a localized Cartesian coordinate system. Note that an infinite number of vectors normal to $\hat{\mathbf{n}}$ exist that form a localized

Cartesian coordinate system. These vectors however are arbitrary and their exact specification is unnecessary. Multiplication of equation (B4) with the matrix $\mathbf{T}$ then yields

$$\bar{\mathbf{Q}}_t + \bar{\mathbf{F}}_\xi = -\mathbf{T}\hat{\mathbf{G}}_\eta + \mathbf{T}_t\hat{\mathbf{Q}} + \mathbf{T}_\xi\hat{\mathbf{F}} - \mathbf{T}\hat{\mathbf{H}}_\varsigma \tag{B6}$$

where

$$\bar{\mathbf{Q}} = \mathbf{T}\hat{\mathbf{Q}} = \frac{1}{J}\begin{Bmatrix} \rho \\ \rho\bar{u} \\ \rho\bar{v} \\ \rho\bar{w} \\ e \end{Bmatrix} \tag{B7}$$

$$\bar{\mathbf{F}} = \mathbf{T}\hat{\mathbf{F}} = \frac{|\mathrm{grad}(\xi)|}{J}\begin{Bmatrix} \rho\bar{u} \\ \rho\bar{u}\bar{u} + p \\ \rho\bar{u}\bar{v} \\ \rho\bar{u}\bar{w} \\ (e+p)\bar{u} \end{Bmatrix} \tag{B8}$$

The rotated velocity component $\bar{u}$ is the velocity normal to a line of constant ξ, representing the scaled contravariant velocity component, whereas $\bar{v}$ and $\bar{w}$ are normal to $\bar{u}$ and to each other. Thus,

$$\bar{u} = \hat{\xi}_x u + \hat{\xi}_y v + \hat{\xi}_z w \tag{B9}$$

$$\bar{v} = \hat{t}_x u + \hat{t}_y v + \hat{t}_z w \tag{B10}$$

$$\bar{w} = \hat{r}_x u + \hat{r}_y v + \hat{r}_z w \tag{B11}$$

The transformed flux $\bar{\mathbf{F}}$ is of the same functional form as the Cartesian flux vector and thus can be split according to any splitting developed for Cartesian coordinates. Therefore, equations for both the Steger-Warming and Van Leer splittings can be used to split the flux vector $\bar{\mathbf{F}}$ after replacing the Cartesian velocity components u, v, and w by the rotated velocity components $\bar{u}, \bar{v}$, and $\bar{w}$, respectively. Applying the rotation $\mathbf{T}$ to equation (B4) simply allows the flux vector to be split in a one-dimensional fashion along a coordinate axis perpendicular to the cell interface. After splitting $\bar{\mathbf{F}}$, the appropriate splitting for $\hat{\mathbf{F}}$ is determined by applying the inverse transformation matrix $\mathbf{T}^{-1}$ to equation (B6), thus leading to

$$\hat{\mathbf{Q}}_t + (\hat{\mathbf{F}}^+ + \hat{\mathbf{F}}^-)_\xi + \hat{\mathbf{G}}_\eta + \hat{\mathbf{H}}_\varsigma = 0 \tag{B12}$$

with

$$\hat{\mathbf{F}}^\pm = \mathbf{T}^{-1}\bar{\mathbf{F}}^\pm = \frac{|\mathrm{grad}(\xi)|}{J}\begin{Bmatrix} f^\pm_{\mathrm{mass}} \\ f^\pm_{\mathrm{mass}}\left\{\left[\hat{\xi}_x(-\bar{u} \pm 2a)/\gamma\right] + u\right\} \\ f^\pm_{\mathrm{mass}}\left\{\left[\hat{\xi}_y(-\bar{u} \pm 2a)/\gamma\right] + v\right\} \\ f^\pm_{\mathrm{mass}}\left\{\left[\hat{\xi}_z(-\bar{u} \pm 2a)/\gamma\right] + w\right\} \\ f^\pm_{\mathrm{energy}} \end{Bmatrix}$$

where

$$f^\pm_{\mathrm{mass}} = \pm\rho a(M_\xi \pm 1)^2 \Big/ 4$$

$$f^\pm_{\mathrm{energy}} = f^\pm_{\mathrm{mass}}\left\{\left[-(\gamma-1)\bar{u}^2 \pm 2(\gamma-1)\bar{u}a + 2a^2\right] \Big/ (\gamma^2-1) + (u^2+v^2+w^2)/2\right\} \tag{B13}$$

Note that the inverse transformation restores the original form of the equations; i.e., no additional source terms arise and the form of $\hat{\mathbf{G}}$ and $\hat{\mathbf{H}}$ is unaffected. This allows a splitting of $\hat{\mathbf{G}}$ and $\hat{\mathbf{H}}$ similar to the splitting of $\hat{\mathbf{F}}$ shown above.

Duplicating the derivation with the Steger-Warming Cartesian splitting, starting with equation (B8), yields

$$\bar{\mathbf{F}} = \bar{\mathbf{F}}_1 + \bar{\mathbf{F}}_2 + \bar{\mathbf{F}}_3 \tag{B14}$$

where

$$\bar{\mathbf{F}}_1 = \frac{|\text{grad}(\xi)|}{J}\bar{\lambda}_1\frac{\gamma-1}{\gamma}\begin{Bmatrix} \rho \\ \rho\bar{u} \\ \rho\bar{v} \\ \rho\bar{w} \\ \frac{\rho}{2}(\bar{u}^2+\bar{v}^2+\bar{w}^2) \end{Bmatrix} \tag{B15}$$

$$\bar{\mathbf{F}}_{2,3} = \frac{|\text{grad}(\xi)|}{J}\frac{\bar{\lambda}_{4,5}}{2\gamma}\begin{Bmatrix} \rho \\ \rho\bar{u} \pm \rho a \\ \rho\bar{v} \\ \rho\bar{w} \\ e+p \pm \rho a\bar{U} \end{Bmatrix} \tag{B16}$$

$$\bar{\lambda}_1 = \frac{\lambda_1}{|\text{grad}(\xi)|} = \frac{U}{|\text{grad}(\xi)|} \tag{B17}$$

where

$$\bar{\lambda}_{4,5} = \frac{\lambda_{4,5}}{|\text{grad}(\xi)|} = \frac{U \pm a|\text{grad}(\xi)|}{|\text{grad}(\xi)|} \tag{B18}$$

Applying the inverse transformation to equation (B14) gives

$$\hat{\mathbf{F}} = \mathbf{T}^{-1}\bar{\mathbf{F}} = \hat{\mathbf{F}}_1 + \hat{\mathbf{F}}_2 + \hat{\mathbf{F}}_3 \tag{B19}$$

$$\hat{\mathbf{F}}_1 = \lambda_1\frac{\gamma-1}{J\gamma}\begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho}{2}(u^2+v^2+w^2) \end{Bmatrix} \tag{B20}$$

$$\hat{\mathbf{F}}_{2,3} = \lambda_{4,5}\frac{1}{J2\gamma}\begin{Bmatrix} \rho \\ \rho u \pm \rho a\hat{\xi}_x \\ \rho v \pm \rho a\hat{\xi}_y \\ \rho w \pm \rho a\hat{\xi}_z \\ e+p \pm \frac{\rho aU}{|\text{grad}(\xi)|} \end{Bmatrix} \tag{B21}$$

This result is identical to the generalized splitting given in references 3 and 7.

Appendix C

Restriction Operators

The dependent variables are transferred from a fine mesh to a coarse mesh so that the mass, momentum, and energy contained in the coarse-grid cell are the same as those contained in the fine-grid cells that compose the coarser grid. For a two-dimensional example, referring to figure 23 shows that the mass, momentum, and energy in any given cell are given by $\mathbf{QV}$, where $\mathbf{V}$ is the volume of the cell and $\mathbf{Q}$ is the column vector of dependent variables representing the conserved quantities per unit volume. Since cell A is comprised of the smaller cells a, b, c, and d, a relationship that conserves mass, momentum, and energy is easily established for transferring the dependent variables. Thus,

$$\mathbf{Q}_A = \frac{\mathbf{Q}_a\mathbf{V}_a + \mathbf{Q}_b\mathbf{V}_b + \mathbf{Q}_c\mathbf{V}_c + \mathbf{Q}_d\mathbf{V}_d}{\mathbf{V}_A} \tag{C1}$$

The restriction of the residuals is also guided by conservation laws. The steady Euler equations can be written in integral form as

$$\int \vec{\mathbf{F}} \cdot \hat{\mathbf{n}}\, ds = 0 \tag{C2}$$

Here, the integral is the surface integral over the volume considered, $\vec{\mathbf{F}}$ is the flux of mass, momentum, or energy across the boundaries, and $\hat{\mathbf{n}}$ is the outward pointing unit vector normal to the boundary.

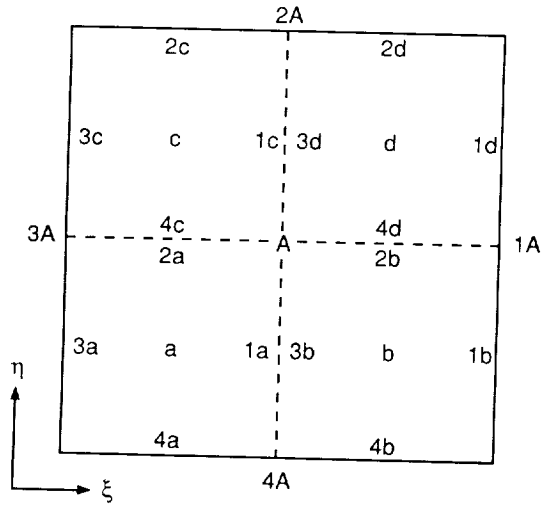


Figure 23. Control volumes for restriction of dependent variables and residual.

Considering the two-dimensional case given above for simplicity shows that the integral around the large volume shown in figure 23 is given by

$$\int \vec{\mathbf{F}} \cdot \hat{\mathbf{n}}\, ds = (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{1A} S_{1A} + (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{2A} S_{2A} + (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{3A} S_{3A} + (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{4A} S_{4A} \tag{C3}$$

Now, the integral along each of the larger faces is the sum of the two integrals along each of the smaller ones. For example,

$$(\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{1A} S_{1A} = (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{1b} S_{1b} + (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{1d} S_{1d} \tag{C4}$$

Also, note that since the outward pointing normals on adjacent cell boundaries point in opposite directions, several terms that share a common boundary will cancel. For instance,

$$(\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{1c} S_{1c} = -(\vec{\mathbf{F}} \cdot \hat{\mathbf{n}})_{3d} S_{3d} \tag{C5}$$

Therefore, by performing the integrations around each of the smaller cells and adding them together, it is seen that the integral around the larger cell is simply the sum of the integrals around each of the smaller ones:

$$\int_{\text{L.C.}} \vec{\mathbf{F}} \cdot \hat{\mathbf{n}}\, ds = \sum \int_{\text{S.C.}} \vec{\mathbf{F}} \cdot \hat{\mathbf{n}}\, ds \tag{C6}$$

where the limits L.C. and S.C. indicate large cells and small cells, respectively.

In order to better relate this specifically to the Euler equations, consider the steady continuity equation given by

$$\int \rho \vec{\mathbf{U}} \cdot \hat{\mathbf{n}}\, ds = 0 \tag{C7}$$

$$\vec{\mathbf{U}} = u\hat{i} + v\hat{j} \tag{C8}$$

$$\hat{\mathbf{n}} = \hat{k}_x\hat{i} + \hat{k}_y\hat{j} = \frac{\text{grad}(k)}{|\text{grad}(k)|} \tag{C9}$$

A normal to a ξ = Constant line is given by using $k = \xi$ in equation (C9), and the normal to an η = Constant line is obtained by using η in the same manner. Now, the length of each face is given by

$$S_i = \frac{|\text{grad}(k)|}{J} \tag{C10}$$

where k is again chosen to be ξ or η, depending on which face is desired, and J represents the inverse of the cell volume (i.e., the Jacobian). Using equations (C3) (with $\vec{\mathbf{F}} = \rho\vec{\mathbf{U}}$), (C7), (C8), and (C9) allows the integral around cell a in figure 23 to be written as

$$\int \rho \vec{\mathbf{U}} \cdot \hat{\mathbf{n}}\, ds = \left(\frac{\rho U}{J}\right)_{1a} - \left(\frac{\rho U}{J}\right)_{3a} + \left(\frac{\rho V}{J}\right)_{2a} - \left(\frac{\rho V}{J}\right)_{4a} \tag{C11}$$

where

$$\left.\begin{aligned} U &= \xi_x u + \xi_y v \\ V &= \eta_x u + \eta_y v \end{aligned}\right\} \tag{C12}$$

When calculated numerically, equation (C11) is simply the residual for the continuity equation. Similar results are obtained for the momentum and energy equations. Therefore, the residuals on a fine grid can be transferred to a coarser grid so that the integral of the fluxes over the cell boundary is conserved simply by summing the fine-grid residuals that make up the coarse grid.

References

1. Moretti, Gino: The λ-Scheme. *Comput. & Fluids*, vol. 7, no. 3, Sept. 1979, pp. 191–205.
2. Chakravarthy, Sukumar R.; Anderson, Dale A.; and Salas, Manuel D.: The Split Coefficient Matrix Method for Hyperbolic Systems of Gasdynamic Equations. AIAA-80-0268, Jan. 1980.
3. Steger, Joseph L.; and Warming, R. F.: Flux Vector Splitting of the Inviscid Gasdynamic Equations With Application to Finite-Difference Methods. *J. Comput. Phys.*, vol. 40, no. 2, Apr. 1981, pp. 263–293.
4. Steger, Joseph L.: *A Preliminary Study of Relaxation Methods for the Inviscid Conservative Gasdynamics Equations Using Flux Splitting.* NASA CR-3415, 1981.
5. Van Leer, Bram: Flux-Vector Splitting for the Euler Equations. *Eighth International Conference on Numerical Methods in Fluid Dynamics*, E. Krause, ed., *Volume 170 of Lecture Notes in Physics*, Springer-Verlag, 1982, pp. 507–512. (Available as ICASE Rep. No. 82-30.)
6. Lombard, C. K.; Oliger, Joseph; and Yang, J. Y.: A Natural Conservative Flux Difference Splitting for the Hyperbolic Systems of Gasdynamics. AIAA-82-0976, June 1982.
7. Janus, Jonathan Mark: *The Development of a Three-Dimensional Split Flux Vector Euler Solver With Dynamic Grid Applications.* M.S. Thesis, Mississippi State Univ., 1984.
8. Thomas, James L.; Van Leer, Bram; and Walters, Robert W.: Implicit Flux-Split Schemes for the Euler Equations. AIAA-85-1680, July 1985.
9. Anderson, W. Kyle; Thomas, James L.; and Van Leer, Bram: A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations. *AIAA J.*, vol. 24, no. 9, Sept. 1986, pp. 1453–1460. (Available as AIAA-85-0122.)
10. Hackbusch, Wolfgang: *Multi-Grid Methods and Applications.* Springer-Verlag, c.1985.
11. Stüben, Klaus; and Trottenberg, Ulrich: Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. *Multigrid Methods*, W. Hackbusch and U. Trottenberg, eds., *Volume 960 of Lecture Notes in Mathematics*, Springer-Verlag, c.1982, pp. 1–176.
12. Ni, Ron-Ho: A Multiple-Grid Scheme for Solving the Euler Equations. *AIAA J.*, vol. 20, no. 11, Nov. 1982, pp. 1565–1571.
13. Johnson, Gary M.: *Multiple-Grid Acceleration of Lax-Wendroff Algorithms.* NASA TM-82843, 1982.
14. Johnson, Gary M.: *Multiple-Grid Convergence Acceleration of Viscous and Inviscid Flow Computations.* NASA TM-83361, 1983.
15. Johnson, Gary M.: *Convergence Acceleration of Viscous Flow Computations.* NASA TM-83039, 1982.
16. Chima, Rodrick V.; and Johnson, Gary M.: *Efficient Solution of the Euler and Navier-Stokes Equations With a Vectorized Multiple-Grid Algorithm.* NASA TM-83376, 1983.
17. Mulder, Wim A.: Multigrid Relaxation for the Euler Equations. *J. Comput. Phys.*, vol. 60, no. 2, Sept. 15, 1985, pp. 235–252.
18. Jespersen, Dennis C.: A Multigrid Method for the Euler Equations. AIAA-83-0124, Jan. 1983.
19. Jameson, Antony: *Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method.* MAE Rep. No. 1613, Princeton Univ., June 1983.
20. Jameson, Antony; and Baker, Timothy J.: Multigrid Solution of the Euler Equations for Aircraft Configurations. AIAA-84-0093, Jan. 1984.
21. Jameson, Antony; and Yoon, Seokkwan: Multigrid Solution of the Euler Equations Using Implicit Schemes. AIAA-85-0293, Jan. 1985.
22. Jameson, Antony; and Yoon, Seokkwan: LU Implicit Schemes With Multiple Grids for the Euler Equations. AIAA-86-0105, Jan. 1986.
23. Buning, Pieter G.; and Steger, Joseph L.: Solution of the Two-Dimensional Euler Equations With Generalized Coordinate Transformation Using Flux Vector Splitting. AIAA-82-0971, June 1982.
24. Von Lavante, E.; Claes, D.; and Anderson, W. K.: The Effects of Various Implicit Operators on a Flux Vector Splitting Method. AIAA-86-0273, Jan. 1986.
25. Jespersen, Dennis C.; and Pulliam, Thomas H.: Flux Vector Splitting and Approximate Newton Methods. *A Collection of Technical Papers—AIAA 6th Computational Fluid Dynamics Conference,* July 1983, pp. 535–543. (Available as AIAA-83-1899.)
26. Van Leer, Bram: Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method. *J. Comput. Phys.*, vol. 32, no. 1, July 1979, pp. 101–136.
27. South, Jerry C., Jr.: Recent Advances in Computational Transonic Aerodynamics. AIAA-85-0366, Jan. 1985.
28. Brandt, Achi: Guide to Multigrid Development. *Multigrid Methods*, W. Hackbusch and U. Trottenberg, eds., *Volume 960 of Lecture Notes in Mathematics*, Springer-Verlag, c.1982, pp. 220–312.
29. Brandt, Achi: Multi-Level Adaptive Solutions to Boundary-Value Problems. *Math. Comput.*, vol. 31, no. 138, Apr. 1977, pp. 333–390.
30. Brandt, Achi: Multilevel Adaptive Computations in Fluid Dynamics. *AIAA J.*, vol. 18, no. 10, Oct. 1980, pp. 1165–1172.
31. Schmitt, V.; and Charpin, F.: Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers. *Experimental Data Base for Computer Program Assessment,* AGARD-AR-138, May 1979, pp. B1-1–B1-44.

32. Eriksson, Lars-Erik: *Transfinite Mesh Generation and Computer-Aided Analysis of Mesh Effects.* Ph.D. Diss., Uppsala Univ., 1984.
33. Townsend, James C.; Howell, Dorothy T.; Collins, Ida K.; and Hayes, Clyde: *Surface Pressure Data on a Series of Analytic Forebodies at Mach Numbers From 1.70 to 4.50 and Combined Angles of Attack and Sideslip.* NASA TM-80062, 1979.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2829		

4. Title and Subtitle	5. Report Date
Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations	November 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
W. Kyle Anderson, James L. Thomas, and David L. Whitfield	L-16416
9. Performing Organization Name and Address	10. Work Unit No.
NASA Langley Research Center Hampton, VA 23665-5225	505-60-01-03
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, DC 20546-0001	Technical Paper
	14. Sponsoring Agency Code

15. Supplementary Notes

W. Kyle Anderson and James L. Thomas: Langley Research Center, Hampton, Virginia.
David L. Whitfield: Mississippi State University, Mississippi State, Mississippi.

16. Abstract

The full-approximation-scheme (FAS) multigrid method is applied to several implicit flux-split algorithms for solving the three-dimensional Euler equations in a body-fitted coordinate system. Each of the splitting algorithms uses a variation of approximate factorization and is implemented in a finite-volume formulation. The algorithms are all vectorizable with little or no scalar computation required. The flux vectors are split into upwind components using both the Steger-Warming and Van Leer splittings. The stability and smoothing rates of each of the schemes are examined using a Fourier analysis of the complete system of equations. Results are presented for three-dimensional subsonic, transonic, and supersonic flows that demonstrate substantially improved convergence rates with the multigrid algorithm. The influence of using both V-cycle and W-cycle strategies on the convergence is examined.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Multigrid algorithms Upwind differencing Flux splitting Euler equations	Unclassified—Unlimited Subject Category 64

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	38	A03

NASA FORM 1626 OCT 86

NASA-Langley, 1988

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171