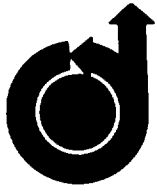


13000
79
in 2012
Paper 79-1456 only



AIAA 79-1456R

**Fast, Conservative Algorithm for Solving the
Transonic Full-Potential Equation**

T. L. Holst

Reprinted from

AIAA Journal

Volume 18, Number 12, December 1980, Page 1431

This paper is declared a work of the U.S. Government and therefore is in the public domain

Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation

NASA/TM-80-208091

Terry L. Holst*

NASA Ames Research Center, Moffett Field, Calif.

A fast, fully implicit approximate factorization algorithm designed to solve the conservative, transonic, full-potential equation in either two or three dimensions is described. The algorithm uses an upwind bias of the density coefficient for stability in supersonic regions. This provides an effective upwind difference of the streamwise terms for any orientation of the velocity vector (i.e., rotated differencing), thereby greatly enhancing the reliability of the present algorithm. A numerical transformation is used to establish an arbitrary body-fitted, finite-difference mesh. Computed results for both airfoils and simplified wings demonstrate substantial improvement in convergence speed for the new algorithm relative to standard successive-line over-relaxation algorithms.

Introduction

An implicit approximate factorization algorithm (AF2) for solving the low-frequency (unsteady) transonic small-disturbance equation in two and three dimensions was presented in Ref. 1. This algorithm has been subsequently applied to the solution of the transonic small-disturbance potential equation² and the conservative full-potential equation³⁻⁵ for steady flows in two space dimensions. For both steady formulations, significant improvement in convergence speed has been obtained relative to the standard transonic relaxation procedure, successive-line over-relaxation (SLOR). In the present study, the AF2 algorithm is applied to the conservative full-potential equation for steady three-dimensional transonic flowfields.

Several general guidelines for the construction of implicit approximate factorization (AF) schemes can be formulated by considering the two-level iteration procedure

$$NC^n + \omega L\phi^n = 0 \quad (1)$$

where C^n is the correction ($\phi^{n+1} - \phi^n$), $L\phi^n$ is the residual, which is a measure of how well the finite-difference equation is satisfied by the n th level velocity-potential solution (ϕ^n), and ω is a relaxation parameter. The iteration scheme given by Eq. (1) can be considered as an iteration in pseudotime, where the n superscript indicates the time step level of the solution; i.e., $(\)^{n+1} - (\)^n \sim \Delta t (\)_t$. The operator N determines the type of iterative procedure, and therefore determines the rate at which the solution procedure converges. In the AF approach, N is chosen as a product of several factors, usually two factors for two-dimensional algorithms and three factors for three-dimensional algorithms. These factors are chosen so that their product is an approximation to L , only simple matrix operations are required, and the overall scheme is stable.

Stability in the present full-potential formulation for supersonic regions of flow has been achieved by the addition of an artificial viscosity term similar to that introduced in Ref. 6. However, in the present formulation, addition of the artificial viscosity term is achieved by using an upwind bias of

the density coefficient. This strategy is significant because it simplifies the technique for including an artificial viscosity term into the residual operator. Other studies⁷⁻⁹ have used similar steady-state differencing procedures in a wide variety of problems to further substantiate this differencing procedure as being both reliable and flexible.

Governing Equations

The three-dimensional full-potential equation written in strong conservation-law form is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0 \quad (2a)$$

$$\rho = \left[1 - \frac{\gamma-1}{\gamma+1} (\phi_x^2 + \phi_y^2 + \phi_z^2) \right]^{1/\gamma-1} \quad (2b)$$

The density (ρ) and velocity components (ϕ_x , ϕ_y , and ϕ_z) are nondimensionalized by the stagnation density (ρ_s) and the critical sound speed (a_s), respectively; x , y , and z are Cartesian coordinates in the streamwise, spanwise, and vertical directions, respectively, and γ is the ratio of specific heats. The two-dimensional conservation-law form of the full-potential equation is simply obtained by dropping all y -derivative terms from Eq. (2).

Equation (2) expresses mass conservation for flows that are steady, isentropic, and irrotational. The corresponding shock-jump conditions are valid approximations to the Rankine-Hugoniot relations for many transonic flow applications. A comparison of isentropic and Rankine-Hugoniot shock polars is given in Ref. 10.

Equation (2) is transformed from the physical domain (Cartesian coordinates) to the computational domain by using a general independent variable transformation. This transformation, indicated by (see Fig. 1)

$$\xi = \xi(x, y, z) \quad \eta = \eta(x, y, z) \quad \zeta = \zeta(x, y, z) \quad (3)$$

maintains the strong conservation-law form of Eq. (2).¹¹ The full-potential equation written in the computational domain (ξ - η - ζ coordinate system) is given by

$$\left(\frac{\rho U}{J} \right)_\xi + \left(\frac{\rho V}{J} \right)_\eta + \left(\frac{\rho W}{J} \right)_\zeta = 0 \quad (4a)$$

$$\rho = \left[1 - \frac{\gamma-1}{\gamma+1} (U\phi_\xi + V\phi_\eta + W\phi_\zeta) \right]^{1/\gamma-1} \quad (4b)$$

Presented as Paper 79-1456 at the AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., July 23-26, 1979; submitted Aug. 24, 1979; revision received April 8, 1980. This paper is declared a work of the U.S. Government and therefore is in the public domain.

Index categories: Transonic Flow; Computational Methods.

*Research Scientist, Applied Computational Aerodynamics Branch, Member AIAA.

where

$$\begin{aligned} U &= A_1 \phi_\xi + A_4 \phi_\eta + A_5 \phi_\zeta & A_3 &= \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \\ V &= A_4 \phi_\xi + A_2 \phi_\eta + A_6 \phi_\zeta & A_4 &= \xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z \\ W &= A_5 \phi_\xi + A_6 \phi_\eta + A_3 \phi_\zeta & A_5 &= \xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z \\ A_1 &= \xi_x^2 + \xi_y^2 + \xi_z^2 & A_6 &= \eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z \\ A_2 &= \eta_x^2 + \eta_y^2 + \eta_z^2 \end{aligned} \quad (5a)$$

and

$$J = \xi_x \eta_y \zeta_z + \xi_y \eta_z \zeta_x + \xi_z \eta_x \zeta_y - \xi_z \eta_y \zeta_x - \xi_y \eta_x \zeta_z - \xi_x \eta_z \zeta_y \quad (5b)$$

$U, V,$ and W are contravariant velocity components along the $\xi, \eta,$ and ζ directions, respectively, A_1 - A_6 are metric quantities, and J is the Jacobian of the transformation. To evaluate the expressions of Eq. (5), the following metric identities are necessary:

$$\begin{aligned} \xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta) & \eta_x &= J(y_\zeta z_\xi - y_\xi z_\zeta) & \zeta_x &= J(y_\xi z_\eta - y_\eta z_\xi) \\ \xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta) & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi) & \zeta_y &= J(x_\eta z_\xi - x_\xi z_\eta) \\ \xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta) & \eta_z &= J(x_\zeta y_\xi - x_\xi y_\zeta) & \zeta_z &= J(x_\xi y_\eta - x_\eta y_\xi) \end{aligned} \quad (6)$$

The two-dimensional form of the full-potential equation written in the computational domain (ξ - ζ coordinates) is obtained by dropping all y and η terms in Eqs. (4-6); i.e., all y and η derivatives as well as all derivatives of y and η are set equal to zero. An exception to this is that y_η and η_y must be set equal to one.

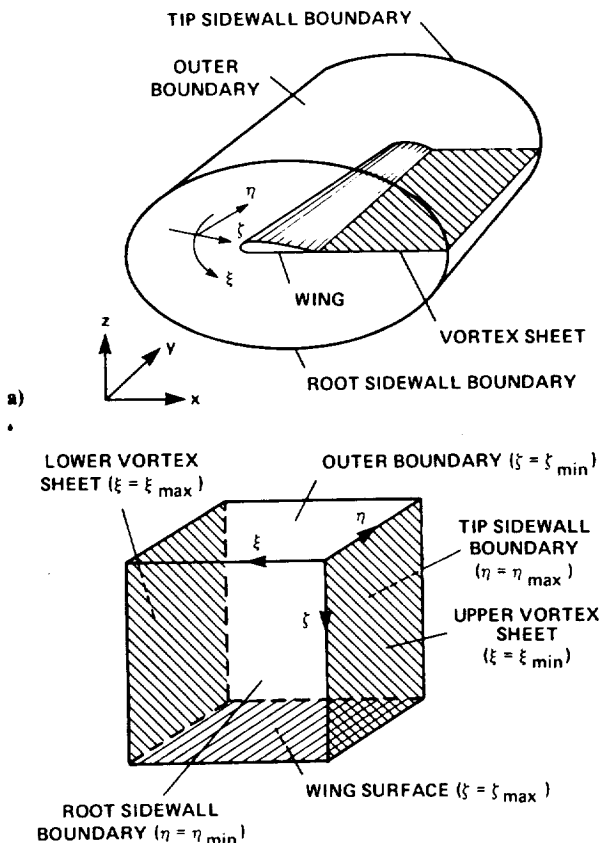


fig. 1 Schematic of general $(x, y, z) \rightarrow (\xi, \eta, \zeta)$ transformation: a) physical domain; b) computational domain.

Several significant advantages are offered by this very general form. The main advantage is that boundaries associated with the physical domain are transformed to boundaries of the computational domain. This aspect is illustrated in Fig. 1 where the physical and computational domains for a typical transformation are shown. The computational coordinates $\xi, \eta,$ and ζ are in the wraparound, spanwise, and radial-like directions, respectively. The inner wing boundary transforms to $\zeta = \zeta_{max}$, and the outer physical boundary transforms to $\zeta = \zeta_{min}$. Note that no restrictions have been placed on the shape of the outer boundary. Arbitrarily shaped outer boundaries, including wind-tunnel walls, may be used. The symmetry-plane boundary transforms to $\eta = \eta_{min}$, and the wing-tip boundary transforms to $\eta = \eta_{max}$. The last two sides of the computational domain are formed from the upper and lower cuts along the vortex sheet.

In the present study, the generality of Eqs. (4) and (5) is reduced somewhat by one simplification, namely, all $\eta = \text{constant}$ surfaces must coincide with $y = \text{constant}$ planes. This can be expressed mathematically by

$$y_\zeta = 0 \quad y_\xi = 0 \quad (7)$$

The metric quantities of Eqs. (5) and (6) reduce to the following:

$$\begin{aligned} A_1 &= \xi_x^2 + \xi_y^2 + \xi_z^2 & A_2 &= \eta_y^2 \\ A_3 &= \zeta_x^2 + \zeta_y^2 + \zeta_z^2 & A_4 &= \xi_y \eta_y \\ A_5 &= \xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z & A_6 &= \zeta_y \eta_y \end{aligned} \quad (8a)$$

$$J = (\xi_x \zeta_z - \xi_z \zeta_x) \eta_y = 1 / (x_\xi z_\zeta - x_\zeta z_\xi) y_\eta \quad (8b)$$

and

$$\begin{aligned} \xi_x &= J y_\eta z_\zeta, & \zeta_x &= -J y_\eta z_\xi \\ \xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta), & \zeta_y &= J(x_\eta z_\xi - x_\xi z_\eta) \\ \xi_z &= -J x_\zeta y_\eta, & \zeta_z &= J x_\xi y_\eta \\ \eta_x &= 0 & \eta_y &= 1 / y_\eta & \eta_z &= 0 \end{aligned} \quad (9)$$

Use of Eq. (7) permits the simplified treatment of wing geometries and does not affect the generality of the three-dimensional spatial-differencing scheme or the fully implicit AF2 iteration scheme.

Grid Generation

The grid-generation scheme used in the present three-dimensional formulation is a simple extension of the two-dimensional scheme presented in Ref. 4. The finite-difference mesh for each spanwise plane ($\eta = \text{constant}$ plane) is generated using the standard two-dimensional algorithm. This requires solution of the following two Laplace equations in each spanwise plane:

$$\xi_{xx} + \xi_{zz} = 0 \quad \zeta_{xx} + \zeta_{zz} = 0 \quad (10)$$

These equations are transformed to (and solved in) the computational domain; that is, ξ and ζ are the independent variables, and x and z are the dependent variables. A fast approximate factorization relaxation algorithm is used to solve the resulting transformed equations.⁴ This establishes values for x and z in each spanwise plane. Coordinate values in the spanwise direction (y values) are established by a simple stretching formula (usually $\Delta y = \text{const}$).

Next, the effects of wing sweep are built into the three-dimensional mesh by applying a simple translation to all x

values. This is accomplished by

$$x|_{\text{swept}} = x|_{\text{unswept}} + (j-1)\Delta y \tan \lambda_j \tag{11}$$

where λ_j is the sweep angle at the j th mesh node in the span direction. Given the values of x , y , and z at all mesh points, the values of the quantities x_ξ , x_η , etc., are easily computed using standard fourth-order-accurate finite-difference formulas. (It has been found that using fourth-order-accurate formulas, instead of second-order-accurate formulas, improves the accuracy of the resulting solutions.) Then, with Eqs. (8) and (9), the quantities $\xi_x, \xi_y, \dots, A_1, A_2, \dots$, and J are computed.

Full-Potential Equation Algorithm

Spatial Differencing

A second-order-accurate finite-difference approximation to Eq. (4a) is given by

$$\delta_\xi \left(\frac{\rho U}{J} \right)_{i+\frac{1}{2},j,k} + \delta_\eta \left(\frac{\rho V}{J} \right)_{i,j+\frac{1}{2},k} + \delta_\zeta \left(\frac{\rho W}{J} \right)_{i,j,k+\frac{1}{2}} = 0 \tag{12}$$

where the i , j , and k subscripts indicate position in the ξ , η , and ζ finite-difference mesh. The operators

$$\delta_\xi (), \delta_\eta (), \text{ and } \delta_\zeta () \tag{13}$$

are first-order accurate backward-difference operators in the ξ , η , and ζ directions, respectively. The density calculation is performed in a straightforward manner by using Eq. (4b). Values of density are computed and stored at half-points in the finite-difference mesh; i.e., $i+\frac{1}{2},j,k$, in a manner similar to that suggested by Ref. 12. Values of ϕ_ξ , ϕ_η , and ϕ_ζ required for computing the density at $i+\frac{1}{2},j,k$ are given by

$$\begin{aligned} \phi_{\xi_{i+\frac{1}{2},j,k}} &= \phi_{i+1,j,k} - \phi_{i,j,k} \\ \phi_{\eta_{i+\frac{1}{2},j,k}} &= \frac{1}{4} (\phi_{i+1,j+1,k} - \phi_{i+1,j-1,k} + \phi_{i,j+1,k} - \phi_{i,j-1,k}) \\ \phi_{\zeta_{i+\frac{1}{2},j,k}} &= \frac{1}{4} (\phi_{i+1,j,k+1} - \phi_{i+1,j,k-1} + \phi_{i,j,k+1} - \phi_{i,j,k-1}) \end{aligned} \tag{14}$$

The contravariant velocity components, $U_{i+\frac{1}{2},j,k}$, $V_{i,j+\frac{1}{2},k}$, and $W_{i,j,k+\frac{1}{2}}$, used both in Eq. (12) as well as in the density calculation, are computed by standard second-order accurate finite-difference formulas, an example of which is given by

$$\begin{aligned} U_{i+\frac{1}{2},j,k} &= A_{i+\frac{1}{2},j,k} (\phi_{i+1,j,k} - \phi_{i,j,k}) + \frac{1}{4} A_{i+\frac{1}{2},j,k} (\phi_{i+1,j+1,k} \\ &\quad - \phi_{i+1,j-1,k} + \phi_{i,j+1,k} - \phi_{i,j-1,k}) + \frac{1}{4} A_{i+\frac{1}{2},j,k} (\phi_{i+1,j,k+1} \\ &\quad - \phi_{i+1,j,k-1} + \phi_{i,j,k+1} - \phi_{i,j,k-1}) \end{aligned} \tag{15}$$

Calculation of the density at half-points in the finite-difference mesh produces better shock-wave resolution, first verified in two dimensions by South.¹³ This is because the computational module in the streamwise ξ direction extends over fewer grid points. Special formulas replace Eqs. (14) and (15) at boundaries and will be discussed in a subsequent section.

Equation (12) is a suitable finite-difference scheme for subsonic flow regions. However, for supersonic regions, a properly chosen artificial viscosity term which usually provides an upwind bias to the differencing scheme, must be introduced to maintain stable convergence.⁶ With the present formulation, stable supersonic regions can be maintained by adding an artificial viscosity term of the form

$$-\Delta \xi \left(\nu \rho_\xi \frac{|U|}{J} \right)_\xi - \Delta \eta \left(\nu \rho_\eta \frac{|V|}{J} \right)_\eta - \Delta \zeta \left(\nu \rho_\zeta \frac{|W|}{J} \right)_\zeta \tag{16}$$

where ν is an artificial viscosity coefficient to be defined subsequently. The complete finite-difference approximation to Eq. (4a), including the addition of an upwind-differenced artificial viscosity term, is given by

$$\begin{aligned} &\delta_\xi \left(\frac{\rho U}{J} \right)_{i+\frac{1}{2},j,k} + \delta_\eta \left(\frac{\rho V}{J} \right)_{i,j+\frac{1}{2},k} + \delta_\zeta \left(\frac{\rho W}{J} \right)_{i,j,k+\frac{1}{2}} \\ &\quad - \delta_\xi \left[\left(\frac{\nu U}{J} \right)_{i+\frac{1}{2},j,k} (\rho_{i+\frac{1}{2},j,k} - \rho_{i+r+\frac{1}{2},j,k}) \right] \\ &\quad - \delta_\eta \left[\left(\frac{\nu V}{J} \right)_{i,j+\frac{1}{2},k} (\rho_{i,j+\frac{1}{2},k} - \rho_{i,j+s+\frac{1}{2},k}) \right] \\ &\quad - \delta_\zeta \left[\left(\frac{\nu W}{J} \right)_{i,j,k+\frac{1}{2}} (\rho_{i,j,k+\frac{1}{2}} - \rho_{i,j,k+t+\frac{1}{2}}) \right] = 0 \end{aligned} \tag{17}$$

The ρ_ξ , ρ_η , and ρ_ζ derivatives of Eq. (16) have been evaluated with a backward (forward) difference when $U_{i+\frac{1}{2},j,k}$, $V_{i,j+\frac{1}{2},k}$, and $W_{i,j,k+\frac{1}{2}}$ are positive (negative), respectively. The r , s , and t indices control the difference direction on the density derivatives and are defined by

$$\begin{aligned} r &= \pm 1 \quad \text{when } U_{i+\frac{1}{2},j,k} \lessgtr 0 \\ s &= \pm 1 \quad \text{when } V_{i,j+\frac{1}{2},k} \lessgtr 0 \\ t &= \pm 1 \quad \text{when } W_{i,j,k+\frac{1}{2}} \lessgtr 0 \end{aligned} \tag{18}$$

This maintains an upwind influence in the differencing scheme for supersonic regions anywhere in the finite-difference mesh for any orientation of the velocity vector. Thus, use of the differencing scheme given by Eqs. (17) and (18) closely approximates the effects of a rotated differencing scheme.⁵ This aspect greatly contributes to the stability and reliability of the present algorithm for many difficult test cases.

The scheme given by Eqs. (17) and (18) is centrally differenced, hence, second-order accurate in subsonic regions. In supersonic regions, the differencing is a combination of the second-order accurate central differencing used in subsonic regions and the first-order accurate upwind differencing resulting from the addition of artificial viscosity. As the flow becomes increasingly supersonic, the scheme is increasingly retarded in the upwind direction.

As shown in Refs. 3 and 4, Eq. (17) can be written in a simplified form given by

$$\delta_\xi \left(\frac{\bar{\rho} U}{J} \right)_{i+\frac{1}{2},j,k} + \delta_\eta \left(\frac{\bar{\rho} V}{J} \right)_{i,j+\frac{1}{2},k} + \delta_\zeta \left(\frac{\bar{\rho} W}{J} \right)_{i,j,k+\frac{1}{2}} = 0 \tag{19a}$$

where

$$\bar{\rho}_{i+\frac{1}{2},j,k} = [(1-\nu)\rho]_{i+\frac{1}{2},j,k} + \nu_{i+\frac{1}{2},j,k} [\rho_{i+r+\frac{1}{2},j,k}] \tag{19b}$$

$$\bar{\rho}_{i,j+\frac{1}{2},k} = [(1-\nu)\rho]_{i,j+\frac{1}{2},k} + \nu_{i,j+\frac{1}{2},k} [\rho_{i,j+s+\frac{1}{2},k}] \tag{19c}$$

$$\bar{\rho}_{i,j,k+\frac{1}{2}} = [(1-\nu)\rho]_{i,j,k+\frac{1}{2}} + \nu_{i,j,k+\frac{1}{2}} [\rho_{i,j,k+t+\frac{1}{2}}] \tag{19d}$$

The addition of the artificial viscosity given by Eq. (16) is thus equivalent to retarding the density in the original centrally differenced scheme [Eq. (12)]. Artificial viscosity is not added explicitly as in Ref. 6. The present approach is significant because it simplifies the technique for including an artificial viscosity term into the residual operator.

The artificial viscosity coefficient, ν , strongly affects the stability of the present scheme and is computed as follows:

$$\nu_{i+\frac{1}{2},j,k} = \begin{cases} \max[(M_{i,j,k}^2 - 1)C, 0] & \text{for } U_{i+\frac{1}{2},j,k} > 0 \\ \max[(M_{i+1,j,k}^2 - 1)C, 0] & \text{for } U_{i+\frac{1}{2},j,k} < 0 \end{cases} \tag{20}$$

The parameter C is a user-specified constant and is usually set between 1.0 and 2.0. Use of larger values of C increases the amount of artificial viscosity and, therefore, the amount of upwinding in the difference scheme. An additional constraint is $\nu \leq 1$ which improves the stability and in some cases, improves the convergence rate. Expressions of ν at $i, j + 1/2, k$ and $i, j, k + 1/2$ are required and are written similarly to $\nu_{i+1/2, j, k}$.

AF2 Iteration Scheme

The AF2 fully implicit approximate factorization scheme^{1,4} is extended in the present study to the three-dimensional full-potential equation. This new scheme can be expressed by choosing the N operator of Eq. (1) as follows:

$$\alpha N C_{i,j,k}^n = - \left[\left(\alpha - \frac{1}{A_k} \bar{\delta}_\eta A_j \bar{\delta}_\eta \right) \left(A_k - \frac{1}{\alpha} \bar{\delta}_\xi A_i \bar{\delta}_\xi \right) - \alpha E_\xi^{+1} A_k \right] (\alpha + \bar{\delta}_\zeta) C_{i,j,k}^n \quad (21a)$$

where

$$A_i = \left(\frac{\bar{\rho} A_j}{J} \right)_{i-1/2, j, k} \quad A_j = \left(\frac{\bar{\rho} A_2}{J} \right)_{i, j-1/2, k} \quad A_k = \left(\frac{\bar{\rho} A_3}{J} \right)_{i, j, k-1/2} \quad (21b)$$

The $\bar{\rho}$, $\bar{\rho}$, and $\hat{\rho}$ coefficients are defined by Eqs. (19b-d), α is a free parameter defined subsequently, and the operator E_ξ^{+1} is a shift operator given by

$$E_\xi^{+1} ()_{i,j,k} = ()_{i,j,k+1} \quad (22)$$

Note that one form of the two-dimensional AF2 iteration scheme is obtained from Eq. (21a) by simply setting the η difference equal to zero.⁴

Inclusion of the retarded density coefficients, $\bar{\rho}$, $\bar{\rho}$, and $\hat{\rho}$, in the N operator is not necessary for stability. This situation was first discussed in Ref. 7 for an implicit ADI-type iteration scheme. Recent testing of the present AF2 iteration scheme in two dimensions, with the retarded density coefficients in the N operator replaced by just the density, yielded stable results with essentially no reduction in the convergence rate. Additional tests with the density coefficients in the N operator completely removed (i.e., replaced by one), also yielded stable results but slowed convergence by a factor of 2 to 3. Therefore, for the present formulation, existence of at least the density in the N operator, but not necessarily the upwinded or retarded density coefficients, is very important for achieving fast convergence but not always necessary for stability.

Implementation of the AF2 scheme is achieved by writing it in a three-step form given by

• Step 1:

$$\left(\alpha - \frac{1}{A_k} \bar{\delta}_\eta A_j \bar{\delta}_\eta \right) g_{i,j}^n = \alpha \omega L \phi_{i,j,k}^n + \alpha A_{k+1} f_{i,j,k+1}^n \quad (23)$$

Step 2:

$$\left(A_k - \frac{1}{\alpha} \bar{\delta}_\xi A_i \bar{\delta}_\xi \right) f_{i,j,k}^n = g_{i,j}^n \quad (24)$$

Step 3:

$$(\alpha + \bar{\delta}_\zeta) C_{i,j,k}^n = f_{i,j,k}^n \quad (25)$$

Here, ω is a relaxation factor equal to 1.8 for all cases presented; $g_{i,j}^n$ is an intermediate result stored at each grid point in a given k plane, i.e., g requires only a two-dimensional array of storage; and $f_{i,j,k}^n$ is an intermediate result stored at each point in the finite-difference mesh. In

step 1, the g array is obtained by solving a tridiagonal matrix equation for each $\xi = \text{constant}$ line in the k th plane. In step 2, the f array is obtained from g by solving a tridiagonal matrix equation for each $\eta = \text{constant}$ line, again for just the k th plane. Next, step 1 is used to obtain the g array for the $k+1$ plane, and then step 2 is used to obtain the f array for the $k+1$ plane, etc. This process continues until all values of f in the three-dimensional mesh are established. Then, by using step 3, the correction array is obtained from the f array by solving a simple bidiagonal matrix equation for each $\zeta = \text{constant}$ line in the entire finite-difference mesh. The nature of this AF2 factorization places a sweep-direction restriction on the step 1-2 combination and on step 3. The step 1-2 combination must be swept in the direction of the decreasing k subscript; that is, from the wing boundary toward the outer boundary (see Fig. 1). The step 3 sweep must proceed in just the opposite direction; that is, from the outer boundary toward the wing. There are no sweep-direction limitations placed on any of the three sweeps due to flow direction.

Initiation of step 1 at the wing boundary ($k = NK$) requires knowledge of f at $NK+1$, which is generally unobtainable. A simple solution is to set f at $NK+1$ equal to zero. Because the present iteration scheme is written in the correction form, f must approach zero as the solution converges. This boundary condition is therefore consistent with the steady-state solution. Other approaches using extrapolations of old known values of f (e.g., f_{NK}^{n-1} and f_{NK-1}^{n-1}), could perhaps provide faster convergence, but have not yet been investigated. A similar boundary condition is required for g at $\eta = \eta_{\min}$ ($j = 1$) and $\eta = \eta_{\max}$ ($j = NJ$) and is implemented by imposing $(g_\eta)_{i,1} = (g_\eta)_{i,NJ} = 0$.

Temporal Damping

For the AF2 factorization, the N operator must be written so that either the ξ -, η -, or ζ -difference approximation to the full-potential equation is split between two factors. This construction generates either a $\phi_{\xi t}$ -, $\phi_{\eta t}$ -, or $\phi_{\zeta t}$ -type term, and if it is properly upwind differenced,¹⁴ provides time-dependent dissipation to the convergence process. When a particular coordinate direction is split (e.g., the ξ direction), the resulting $\phi_{\xi t}$ difference direction is fixed by the construction of the AF2 algorithm; that is, the $\phi_{\xi t}$ term is either backward or forward differenced over the entire mesh. Due to the wraparound ξ coordinate, a backward- (forward-) differenced $\phi_{\xi t}$ is upwind (downwind) differenced below the wing and downwind (upwind) differenced above. Therefore, a problem with $\phi_{\xi t}$ arises either above or below the wing. Following the two-dimensional algorithm development,⁴ the ζ -difference approximation is split between two factors. This allows control over the other more important coordinate directions (ξ and η) because the $\phi_{\xi t}$ and $\phi_{\eta t}$ terms are added to the iteration scheme explicitly and are not part of the factorization construction. The $\phi_{\eta t}$ and $\phi_{\xi t}$ terms are included by adding

$$\mp \beta_\eta |V_{i,j,k}| \bar{\delta}_\eta \quad \text{and} \quad \mp \alpha \beta_\xi \bar{\delta}_\xi \quad (26)$$

inside the brackets of the first and second sweeps, Eqs. (23) and (24), respectively. The parameter β_ξ is determined as follows:

$$\beta_\xi = \beta_{\xi, M > 1} \quad \text{if} \quad \begin{cases} M_{i+1,j,k} > 1 & \text{upper surface} \\ M_{i-1,j,k} > 1 & \text{lower surface} \end{cases} \quad (27)$$

$$\beta_\xi = 0 \quad \text{if} \quad \begin{cases} M_{i+1,j,k} < 1 & \text{upper surface} \\ M_{i-1,j,k} < 1 & \text{lower surface} \end{cases}$$

where $\beta_{\xi, M > 1}$ is a user-specified constant which can be adjusted as needed. The parameter β_η is a user-specified constant fixed over the entire mesh. For all cases presented in the present study, $\beta_\eta = 0$. The double arrow notation on the ξ - and

η -difference operators indicates that the difference is always upwind. For the η direction, a backward difference is used when the η contravariant velocity component ($V_{i,j,k}$) is positive, and a forward difference is used when V is negative. The sign is chosen in each case so that the addition of ϕ_{η} and ϕ_{ξ} increases the magnitude of the first- and second-sweep diagonal coefficients, respectively.

Of course, with this N operator construction, the ϕ_{τ} -type term is properly differenced in only the aft half of the mesh. In the two-dimensional algorithm presented in Ref. 4, the ϕ_{τ} -type term was properly differenced in the forward half of the mesh. Both versions seem to yield equivalent results. However, most experimentation has been conducted on the Ref. 4 version. Although adverse effects for cases with supersonic flow at the trailing edge may be anticipated with this form, none has been experienced. In fact, cases with freestream Mach numbers near unity have been computed (see the two-dimensional results section) in which the trailing edge is entirely imbedded in supersonic flow with no adverse effects.

The quantity α appearing in Eq. (21a) can be considered as Δt^{-1} . This direct analogy to time provides one strategy for obtaining fast convergence, namely, advance time as fast as possible with large time steps (i.e., small values of α). As pointed out in Refs. 2 and 3, this is effective for attacking the low-frequency errors but not the high-frequency errors. The best overall approach is to use an α sequence containing several values of α . The small values are particularly effective for reducing the low-frequency errors, and the large values are particularly effective for reducing the high-frequency errors. The α sequence given in Ref. 3 has been used for both two- and three-dimensional cases presented herein. The endpoints generally used in two dimensions are $\alpha_L = 0.07$ and $\alpha_H = 1.5$, and for three dimensions $\alpha_L = 0.4$ and $\alpha_H = 4.0$.

Boundary Conditions

The wing surface boundary condition is that of flow tangency (i.e., no flow through the wing surface), and requires the ζ contravariant velocity component at the wing surface be zero ($W=0$). This boundary condition is implemented by applying

$$\left(\frac{\rho W}{J}\right)_{i,j,NK+1/2} = -\left(\frac{\rho W}{J}\right)_{i,j,NK-1/2} \quad (28)$$

where $k=NK$ is the wing surface. In other expressions, where ϕ_{τ} is required at the wing surface [Eqs. (14) and (15)], the $W=0$ boundary condition is used again to obtain

$$\phi_{\tau}|_{\text{wing}} = -\frac{A_5}{A_3} \phi_{\xi} - \frac{A_6}{A_3} \phi_{\eta} \quad (29)$$

Thus, a value of ϕ_{τ} at the wing surface can be obtained without using a one-sided difference on ϕ .

In the present study, a special wing geometry has been chosen to evaluate the new three-dimensional AF2 algorithm, namely, flow past an arbitrary wing mounted between parallel walls. The purpose of this model problem is to simulate the flow past a wing in a wind tunnel. The parallel sidewalls are treated with the same tangency boundary condition used for the wing surface (i.e., $V=0$).

Computed Results

The implicit algorithm presented in the previous section has been coded into a transonic airfoil analysis computer code (TAIR) and a transonic wing analysis computer code (TWING). (For details of the TAIR code, see Ref. 4.) Each of these codes is evaluated in this section by presenting a range of computed examples. The two-dimensional results computed from TAIR were all computed in the default mode. This simply means that only three inputs are allowed to be changed

from case to case: freestream Mach number, angle of attack, and airfoil coordinates. Other parameters including all relaxation factors, acceleration parameters, and temporal damping coefficients are either held fixed or are adjusted automatically by internal computer code logic. This feature greatly simplifies operation of TAIR and, at the same time, improves the reliability, especially for inexperienced users. In the default mode, the timelike dissipation coefficient β_{ξ} and the artificial viscosity coefficient C are chosen very conservatively ($\beta_{\xi} = 5$, $C = 2$). Then as the solution progresses, both the number of supersonic points (NSP) and the circulation Γ are monitored. If NSP and Γ grow very rapidly indicating a difficult case, β_{ξ} and C are increased. Conversely, if NSP and Γ grow slowly, β_{ξ} and C are decreased. This philosophy generally produces converged results for most airfoil solutions ranging from subcritical cases to difficult strong shock cases; however, it reduces the convergence speed below optimum by about 10-50%, depending on the particular case.

Two-dimensional comparisons with the GRUMFOIL computer code¹⁵ are considered next. GRUMFOIL is similar to TAIR in that both codes solve the conservative full-potential equation, but different in that TAIR uses the AF2 iteration scheme and GRUMFOIL uses a hybrid direct-solver/SLOR iteration scheme.⁶ This hybrid iteration scheme is composed of one direct-solver iteration, which is very effective for reducing low-frequency errors, but is unstable for supersonic regions, followed by several (10 is the default) SLOR iterations. The purpose of the SLOR iterations is to smooth high-frequency errors generated by the direct-solver step in regions of supersonic flow. (It should be pointed out that stable operation from the direct-solver iteration scheme, without benefit of the SLOR scheme, can be obtained for solutions in which the supersonic region is small, providing the level of artificial viscosity is increased above the standard amount. This aspect has not been investigated in the present study.) The boundary-layer option, which is available in GRUMFOIL, has not been used in any of the results presented herein. All GRUMFOIL results are computed on a 148×32 mesh, while the TAIR results are computed on a 149×32 mesh.

The first two-dimensional case considered here is for the NACA 0012 airfoil at $M_{\infty} = 0.75$ and $\alpha = 1$ deg. The present surface pressure coefficient distribution for this calculation is compared in Fig. 2 with the GRUMFOIL computer code result. The agreement is excellent everywhere except at the shock wave where the disagreement in position is about one mesh-cell width. This error causes a $2\frac{1}{2}\%$ error in lift coefficient.

Convergence history curves for this calculation are presented in Fig. 3. As discussed in Ref. 3, use of the residual operator for comparing different iteration scheme convergence histories can produce quite misleading results. This is because the residual operator is heavily biased toward the high-frequency end of the error spectrum, and therefore, does not produce a fair comparison for algorithms which treat low-frequency errors differently. A much more suitable means of comparing convergence histories is to use error. The rms error in the airfoil surface pressure is used in the present study and is defined by

$$E_{\text{rms}}^n = \left\{ \left[\sum_{i=1}^{NI} (C_{p_i}^n - C_{p_i})^2 \right] / \sum_{i=1}^{NI} C_{p_i}^2 \right\}^{1/2} \quad (30)$$

where $C_{p_i}^n$ is the surface pressure coefficient at the i th grid point and the n th iteration, C_{p_i} is the surface pressure coefficient at the i th grid point taken from the converged solution, and NI is the total number of surface grid points.

The three curves shown in Fig. 3 correspond to the following iteration schemes: 1) AF2, 2) hybrid, and 3) SLOR. Each convergence-history curve is constructed by plotting

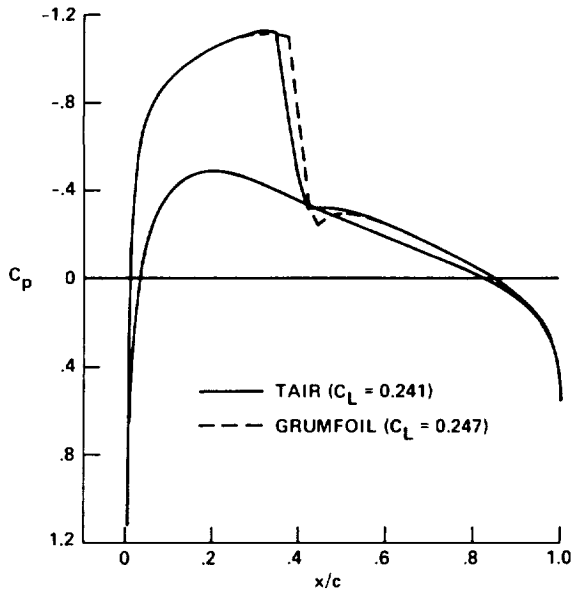


Fig. 2 Pressure coefficient comparison (NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 1$ deg).

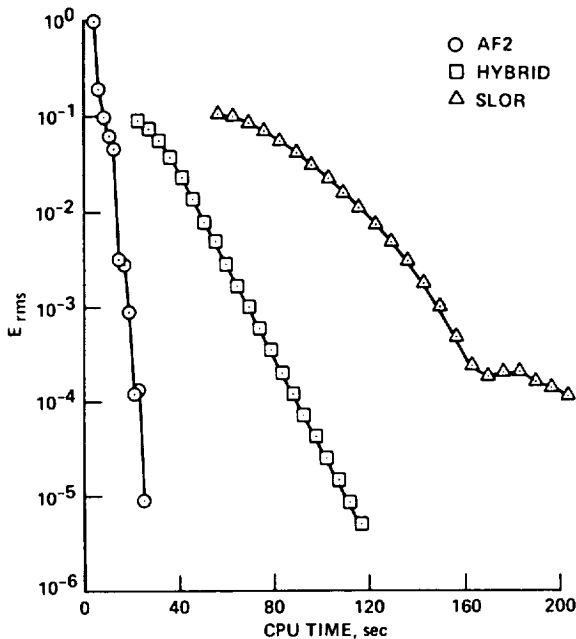


Fig. 3 Two-dimensional convergence histories (NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 1$ deg).

E_{rms} vs CPU time (Ames CDC 7600 computer). The SLOR scheme is simply the hybrid scheme without benefit of the direct-solver step, and has been approximately optimized by a trial-and-error adjustment of the relaxation parameters. The hybrid cases have been computed with default values for all relaxation parameters. Setup times, that is, the CPU time required for grid generation, initialization, and coarse- and medium-mesh calculations, are included in each convergence-history curve. The AF2 curve includes 6 s for grid generation and initialization. The hybrid and SLOR curves both use coarse-medium-fine mesh sequences. Converged results from the coarse mesh are interpolated onto the medium mesh, and then from the medium mesh onto the fine mesh, thus providing a good initial guess for the fine-mesh calculation. The setup times for these cases are 9 s for the hybrid scheme and 43 s for the SLOR scheme. For this calculation, AF2 is 3.5 times faster than the hybrid scheme and 7 times faster than SLOR.

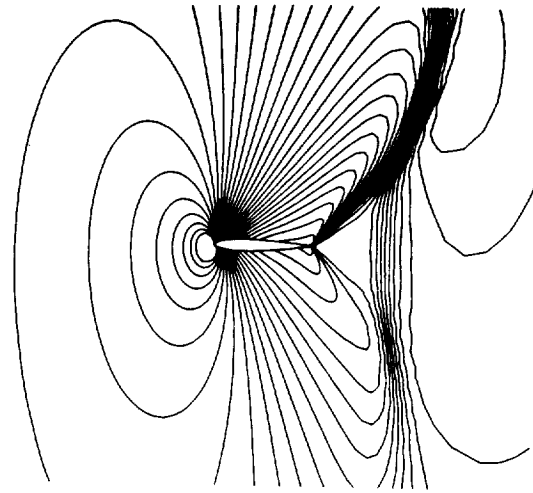


Fig. 4 Mach number contours (NACA 0012 airfoil, $M_\infty = 0.95$, $\alpha = 4$ deg).

Solutions with M_∞ Approaching Unity

As the freestream Mach number approaches one, interesting airfoil shock wave patterns develop. For instance, the shock wave pattern about a NACA 0012 airfoil at $M_\infty = 0.95$ and $\alpha = 4$ deg is shown by the Mach number contours in Fig. 4. A so-called "fishtail" shock system is formed. Relatively weak supersonic-to-supersonic oblique shocks emanate from the trailing edge and merge with a normal shock downstream of the airfoil. The oblique shock emanating from the trailing edge upper surface has been strengthened by the addition of circulation, while the oblique shock emanating from the trailing edge lower surface has been weakened and is almost nonexistent. The normal shock above the airfoil is much stronger than the normal shock below the airfoil. The triangular region between the oblique and normal shocks has a nearly constant Mach number which is approximately 1.1. This shock wave pattern is characteristic of solutions with freestream Mach numbers near unity and has been observed experimentally as well as computationally. It is generally considered to be the correct qualitative solution. For instance, a fishtail shock solution for a 10% circular arc airfoil at $M_\infty = 0.98$ and 0 deg angle of attack was presented in Ref. 7. This calculation was a solution to the conservative full-potential equation using a Cartesian mesh and small-disturbance boundary conditions. Because the flow in this solution is essentially aligned with the finite-difference mesh, "rotated differencing" is not necessary.³ However, with the present wraparound coordinate system, rotated differencing is essential for maintaining stability of difficult cases such as the present fishtail shock solution.⁵

As pointed out in Ref. 14, nonconservative full-potential equation solutions with freestream Mach numbers approaching unity are characterized by strong oblique shock waves at the trailing edge followed by subsonic flow. The fishtail shock structure for these cases is not predicted. Conservative vs nonconservative differencing was the subject of discussion in Ref. 16 where similar differences for the transonic small-disturbance potential equation were reported. It is generally understood that these differences are the result of effective mass creation at shock waves for the non-conservative differencing schemes. Therefore, to obtain the proper mass balance and the correspondingly correct solution conservative differencing is required.

Convergence history curves for the last calculation, including E_{rms} , $|R|_{max}$ [defined by Eq. (19a)], NSP, and C_L convergence histories, are presented in Fig. 5. Convergence is achieved in this case in approximately 20 s of CPU time (116 iterations), as indicated by constant values of both NSP and C_L . At this point, $|R|_{max}$ has dropped only slightly, while

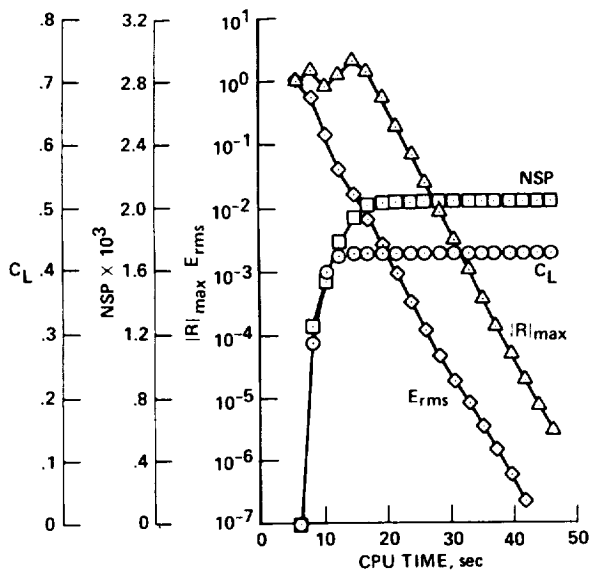


Fig. 5 Two-dimensional convergence histories (NACA 0012 airfoil, $M_\infty = 0.95$, $\alpha = 4$ deg).

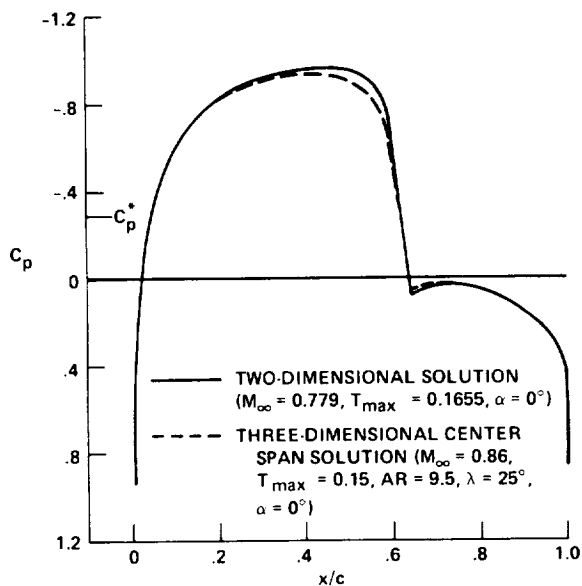


Fig. 6 Two- and three-dimensional pressure coefficient comparison (NACA 00XX airfoil).

E_{rms} has dropped by over $2\frac{1}{2}$ orders of magnitude. This discrepancy is partially due to the fact that the airfoil surface solution, which is effectively monitored by E_{rms} , converges before the downstream fishtail shock wave pattern. Another cause is that during the initial phase of convergence in which the residual does not drop, the position of the shock sonic line is rapidly being adjusted. This excites high-frequency errors, and therefore keeps the residual artificially high even though E_{rms} is being reduced. Establishment of convergence for such a small reduction in the maximum residual is a characteristic behavior of many strong shock calculations using the present algorithm. Use of a three order-of-magnitude reduction in $|R|_{max}$ as a convergence criterion for the present case would require E_{rms} to be reduced by five orders of magnitude, which represents a factor of two more iterations than necessary.

This is the first time in transonic flow computations that calculations such as the last one have been computed using the conservative full-potential equation with an exact airfoil mapping. Without the newly developed rotated difference scheme, these calculations would have been unstable. The

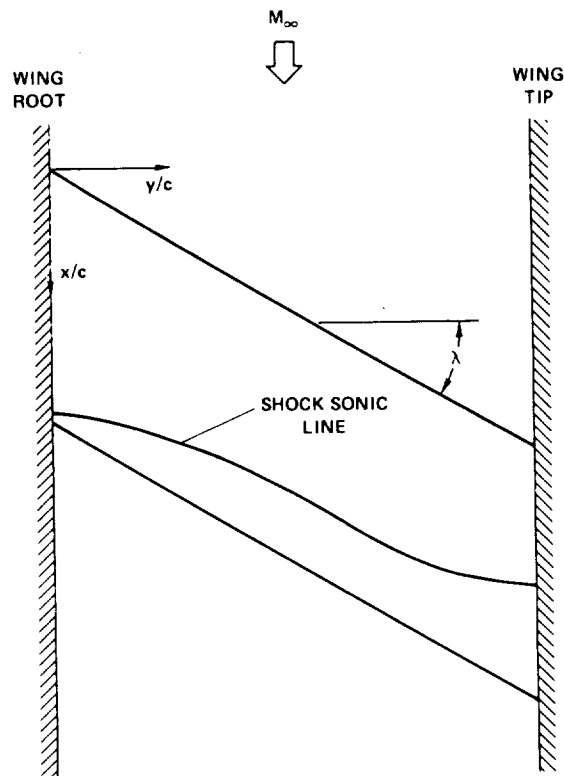


Fig. 7 Shock wave sonic line position relative to the wing planform (NACA 0015 wing, $M_\infty = 0.86$, $\lambda = 30$ deg, $AR = 1.9$, $\alpha = 0$ deg).

rapid convergence of this difficult case demonstrates the reliability and efficiency of the present transonic flow solution procedure.

Three-Dimensional Solutions

Results from the transonic wing analysis code (TWING) are presented in this section. All calculations have been computed with the density coefficients upwinded in only the ξ and η directions. For the cases presented herein, this was sufficient; but for cases with stronger shocks at the trailing edge, density upwinding along the ζ direction would probably be required. All results have been computed with β_ξ ranging from 0.1 to 0.5. The larger values of β_ξ were required for the larger aspect ratio cases.

To evaluate the three-dimensional code, several infinite-aspect-ratio results have been compared with two-dimensional results using the concept of simple sweep theory. For three-dimensional infinite-aspect-ratio calculations, the solution in the wing-normal plane is the same as a two-dimensional solution with appropriate scaling; that is, $M_{\infty,2D} = \cos \lambda \cdot M_{\infty,3D}$, $C_{p,3D} = \cos^2 \lambda \cdot C_{p,2D}$, and $T(x)_{2D} = T(x)_{3D} / \cos \lambda$, where $T(x)$ is the airfoil thickness distribution. Subcritical results using this concept have been compared and are in excellent agreement. A supercritical comparison is shown in Fig. 6. The three-dimensional calculation was computed with an aspect ratio (AR) of 9.5 and thus simulates an infinite-aspect-ratio condition at center span. Other conditions for this calculation were NACA 0015 wing, $M_\infty = 0.86$, $\lambda = 25$ deg, and $\alpha = 0$ deg. The two results are in close agreement but are not identical. The reason for the slight disagreement is numerical and is due to the increased level of artificial viscosity arising from larger values of the local Mach number in the three-dimensional case. If the artificial viscosity parameter (ν) in the two-dimensional case is increased by adding $\sin^2 \lambda M_{\infty,3D}^2$ to the M^2 term of Eq. (20), the two numerical solutions become identical.

Figures 7 and 8 display the results of a wing calculation with the following characteristics: NACA 0015 wing, $AR = 1.9$,

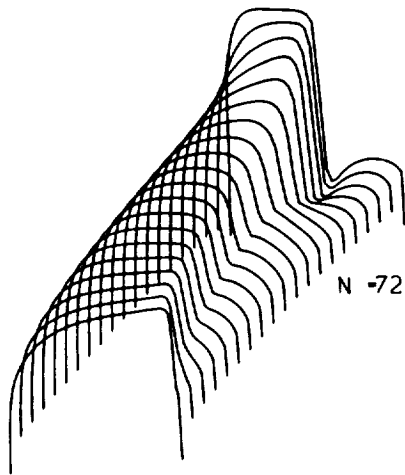


Fig. 8 Three-dimensional pressure coefficient distribution (NACA 0015 wing, $M_\infty = 0.86$, $\lambda = 30$ deg, $R = 1.9$, $\alpha = 0$ deg).

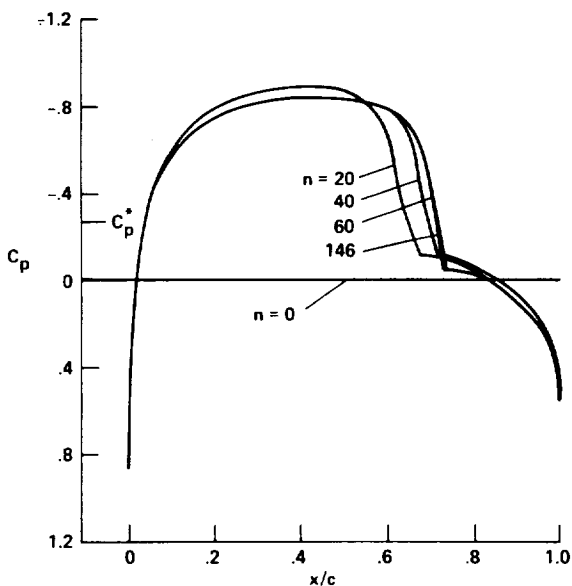


Fig. 9 Center span pressure coefficient evolution with iteration number (NACA 0015 wing, $R = 1.9$, $\alpha = 0$ deg, $M_\infty = 0.86$, $\lambda = 30$ deg).

$M_\infty = 0.86$, $\lambda = 30$ deg, and $\alpha = 0$ deg. The shock wave sonic-line position is plotted to scale on the wing planform in Fig. 7. As expected, the shock wave approaches both sidewalls in a nearly perpendicular fashion. Between the sidewalls near center span, the shock is swept and is approximately parallel to the wing leading and trailing edges. The shock-strength gradient along the span is quite large as indicated by the surface C_p distributions shown in Fig. 8. The maximum local Mach numbers in the root, center span, and tip planes are 1.26, 1.33, and 1.78, respectively. A large part of this spanwise shock-strength gradient is caused by the tip sidewall/wing interaction, which is essentially the opposite of a three-dimensional relief effect. The existence of the tip sidewall constrains the streamlines to remain in the tip plane. The wing sweep induces a spanwise component of velocity which in effect squeezes the streamlines toward the tip plane. This increases the Mach number and therefore the shock strength in the tip plane region.

Figure 9 displays the center span C_p distribution evolution with iteration number (n) for the case just presented. The $n=0$ solution corresponds to the initial condition, and the $n=146$ solution corresponds to the solidly converged solution

in which the maximum residual has been reduced by three orders of magnitude. In just 20 iterations, a good approximation is established in which the shock position is off by only 7%. By 60 iterations the difference in shock positions is less than 1%.

All TWING calculations presented herein have been computed using a mesh with 46,000 points ($115 \times 20 \times 20$). This places 58 points from leading to trailing edge on both the upper and lower surfaces for each spanwise plane. Each iteration requires about 4.2 s of CPU time on the Ames CDC 7600 computer. For the case of Fig. 8, this equates to a total run time of about 5 min to achieve plottable accuracy. For subcritical cases, the run time is much less, amounting to just over a minute to achieve plottable accuracy.

Conclusions

A fast, implicit algorithm for solving the conservative full-potential equation in both two and three dimensions is presented. Stability in supersonic regions is maintained by using an upwind evaluation of the density coefficient along all coordinate directions. This provides an effective upwind difference of the streamwise terms for any orientation of the velocity vector (i.e., rotated differencing), and thereby greatly enhances the reliability of the present algorithm. Use of the newly developed rotated differencing scheme has been instrumental in computing a number of difficult two-dimensional test cases including several cases with "fishtail" shock-wave patterns. This represents the first time such calculations have been computed using the conservative full-potential equation with an exact airfoil mapping.

The present fully implicit AF2 algorithm has been compared with both the standard transonic-solution procedure, successive-line over-relaxation (SLOR), and a hybrid (direct-solver/SLOR) scheme. The surface C_p distributions produced by these schemes are in good agreement. Based on CPU time, the rms error in the surface pressures is reduced from five to seven times faster by the AF2 algorithm relative to SLOR. The hybrid scheme displays a very wide range of convergence speeds, being very fast for subcritical cases but much slower for strong shock cases. The AF2 scheme displays a much more uniform convergence rate over a very wide range of cases. It is essentially comparable to the hybrid scheme for subcritical cases but as much as 3.5 times faster for cases with strong supersonic regions.

Three-dimensional results from the newly developed transonic-wing-analysis code (TWING) are presented. Infinite-aspect-ratio results are in good agreement with standard two-dimensional results. Other calculations for swept wings mounted between parallel walls are presented in which a strong shock wave extends across the entire wing span indicating a high degree of reliability. Convergence histories indicate that the substantial improvement in convergence rate experienced in two-dimensional cases carries over to the newly implemented three-dimensional version of the fully implicit AF2 iteration scheme.

Acknowledgments

The author expresses gratitude to W. F. Ballhaus for many helpful suggestions during the course of this study, and to J. Frick and A. Fernquist for writing the computer codes described herein.

References

- Ballhaus, W. F. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equation," NASA TM X-73,082, 1975.
- Ballhaus, W. F., Jameson, A., and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," *AIAA Journal*, Vol. 16, June 1978, pp. 573-579.
- Holst, T. L. and Ballhaus, W. F., "Fast Conservative Schemes for the Full Potential Equation Applied to Transonic Flows," NASA

TM-78469, 1978; also *AIAA Journal*, Vol. 17, Feb. 1979, pp. 145-152.

⁴Holst, T. L., "An Implicit Algorithm for the Conservative, Transonic Full Potential Equation Using an Arbitrary Mesh," *AIAA Paper* 78-1113, July 1978; also *AIAA Journal*, Vol. 17, Oct. 1979, pp. 1038-1045.

⁵Holst, T. L. and Albert, J., "An Implicit Algorithm for the Conservative, Transonic Full Potential Equation with Effective Rotated Differencing," *NASA TM* 78570, April 1979.

⁶Jameson, A., "Transonic Potential Flow Calculations Using Conservative Form," *AIAA Second Computational Fluid Dynamics Conference Proceedings*, June 1975, pp. 148-155.

⁷Hafez, M. M., Murman, E. M., and South, J. C., "Artificial Compressibility Methods for Numerical Solution of Transonic Full Potential Equation," *AIAA Paper* 78-1148, July 1978; also *AIAA Journal*, Vol. 17, Aug. 1979, pp. 838-844.

⁸Eberle, A., "A Finite Volume Method for Calculating Transonic Potential Flow Around Wings from the Pressure Minimum Integral," *Tech. Translation*, *NASA TM-75,324*, 1978.

⁹Eberle, A., "Transonic Potential Flow Computations by Finite Elements: Airfoil and Wing Analysis, Airfoil Optimization," Lecture

held at the DGLR/GARTEUR 6 Symposium, Transonic Configurations, Bad Harzburg, Germany, June 1978.

¹⁰Steger, J. L. and Baldwin, B. S., "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow," *NASA TN D-6997*, 1972.

¹¹Steger, J. L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils," *AIAA Paper* 77-665, June 1977; also *AIAA Journal*, Vol. 16, July 1978, pp. 679-686.

¹²Jameson, A., "Transonic Flow Calculations," *VKI Lecture Series* 87, March 1976.

¹³South, J. C., private communication, 1979.

¹⁴Jameson, A., "Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1," *Communications on Pure and Applied Mathematics*, Vol. 27, 1974, pp. 283-309.

¹⁵Melnik, R. E., "Wake Curvature and Trailing Edge Interaction Effects in Viscous Flow over Airfoils," *Advanced Technology Airfoil Research*, *NASA CP* 2045, March 1978, pp. 255-270.

¹⁶Newman, P. A. and South, J. C., "Conservative Versus Non-conservative Differencing: Transonic Streamline Shape Effects," *NASA TM X-72,827*, 1976.