ARTICLE NO. 78-1113R

Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh

Terry L. Holst*
NASA Ames Research Center, Moffett Field, Calif.

A new, implicit approximate factorization (AF) algorithm designed to solve the conservative full-potential equation for the transonic flow past arbitrary airfoils has been developed. The new algorithm uses an upwind bias of the density coefficient to provide stability in supersonic regions. This allows the simple two- and three-banded matrix form of the AF scheme to be retained over the entire flowfield, even in regions of supersonic flow. A numerical transformation is used to establish an arbitrary body-fitted finite-difference mesh. Airfoil pressure distributions have been computed and are in good agreement with independent results.

I. Introduction

THE object of this investigation was to determine the feasibility of using implicit approximate factorization (AF) algorithms to solve the full-potential (FP) equation in conservation form for steady transonic flowfields. The primary motivation for using the AF approach is the increase in computational efficiency afforded by the fully implicit algorithm. The AF approach has been studied by Ballhaus et al.[1] for the numerical solution of the transonic small-disturbance (TSD) equation and by Holst and Ballhaus[2] for the numerical solution of the full-potential equation with small-disturbance boundary conditions. In both studies, AF schemes were found to be significantly faster than the standard transonic solution procedure successive line overrelaxation (SLOR). Two AF schemes have been considered: 1) alternating direction implicit (ADI), called AF1 in Ref. 1; and 2) AF2, a second type of AF technique first introduced by Ballhaus and Steger.[3] For transonic flows, the AF2 scheme has proved to be the fastest and most reliable and, therefore, is the only scheme considered in this study.

The spatial differencing used in the present algorithm is similar to that of Ref. 4. In Ref. 4, stable operation in supersonic regions is achieved by adding a first-order-accurate viscosity term to the standard second-order-accurate differencing scheme. The present algorithm, however, uses an upwind bias of the density coefficient to provide stability in supersonic regions without explicitly adding an artificial viscosity term.

Several guidelines for the construction of AF schemes can be formulated by considering the following general form for a two-level iteration procedure:

$$NC^n + \omega L\phi^n = 0 \tag{1}$$

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 nth level solution (ϕ^n), and ω is a relaxation parameter. The iteration scheme given by Eq. (1) can be regarded as an iteration in pseudo-time, where the n superscript indicates the time-step level of the solution. 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 two or more factors indicated by

$$N = N_1 N_2$$

The factors N_1 and N_2 are chosen so that 1) their product is an approximation to L, 2) only simple matrix operations are required, and 3) the overall scheme is stable.

Special attention has been placed on the treatment of "exact" airfoil surface boundary conditions (i.e., transformation of the airfoil surface to a coordinate line of the computational domain). Several mapping techniques are available for accomplishing this task, including conformal mapping techniques[5] and the mapping technique of Thompson et al.[6] The Thompson approach was chosen in this study because of its flexibility. This numerical technique is completely automatic and can be used to establish smooth finite-difference meshes about arbitrary airfoils with minimal user input. This technique can be extended to three-dimensional cases and can be used to directly model wind-tunnel walls.

It is essential that the finite-difference approximation to the FP equation be cast in conservation form.[7] Otherwise, the shock-capturing procedure will not necessarily conserve mass across the shock wave.[4] Nonconservative rather than conservative difference schemes have been used in many engineering applications. However, the nonconservative procedures introduce mass sources at shock waves, and the strength of these sources depends on the local grid spacing, a nonphysical consideration. Erroneous shock solutions can therefore result.

Section II begins with a discussion of the conservative full-potential equation and the general transformation from the physical to the computational domain. The numerical grid generation scheme (Sec. III) is then discussed including a new AF procedure for solving the grid generation equations. In Sec. IV the spatial differencing approximations are introduced, including the addition of artificial viscosity, by retarding the density. The AF2 algorithm and boundary conditions are considered next, followed by several numerical results in Sec. V.

II. Governing Equations

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

$$(\rho\phi_x)_x + (\rho\phi_y)_y = 0 \tag{2a}$$

Presented as Paper 78-1113 at the AIAA 11th Fluid and Plasma Dynamics Conference, Seattle, Wash., July 10-12, 1978; submitted July 31, 1978; revision received March 26, 1979. This paper is declared a work of the U.S. Government and therefore is in the public domain. Reprints of this article may be ordered from AIAA Special Publications, 1290 Avenue of the Americas, New York, N.Y. 10019. Order by Article No. at top of page. Member price $2.00 each, nonmember, $3.00 each. **Remittance must accompany order.**

Index categories: Transonic Flow; Computational Methods.

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

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

where the density (ρ) and velocity components (ϕ_x and ϕ_y) are nondimensionalized by the stagnation density (ρ_s) and the critical sound speed (a_*), respectively, x and y are Cartesian coordinates, and γ is the ratio of specific heats.

Equations (2) express mass conservation for flows that are isentropic and irrotational. The corresponding shock-jump conditions are valid approximations to the Rankine-Hugoniot relations for many transonic flow applications. A comparison of the isentropic and Rankine-Hugoniot shock polars is given in Ref. 8.

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

$$\xi=\xi(x,y) \qquad \eta=\eta(x,y) \tag{3}$$

maintains the strong conservation-law form of Eqs. (2) as discussed in Refs. 9-12. The full-potential equation written in the computational domain (ξ-η coordinate system) is given by

$$(\rho U/J)_\xi+(\rho V/J)_\eta=0 \tag{4a}$$

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

where

$$U=A_1\phi_\xi+A_2\phi_\eta, \quad V=A_2\phi_\xi+A_3\phi_\eta$$
$$A_1=\xi_x^2+\xi_y^2, \ A_2=\xi_x\eta_x+\xi_y\eta_y, \ A_3=\eta_x^2+\eta_y^2$$
$$J=\xi_x\eta_y-\xi_y\eta_x \tag{5}$$

U and V are the contravariant velocity components along the ξ and η directions, respectively, A_1, A_2, and A_3 are metric quantities, and J is the Jacobian of the transformation.

The transformed full-potential equation [Eqs. (4)] is only slightly more complicated than the original Cartesian form [Eqs. (2)] and offers several significant advantages. 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 inner airfoil boundary becomes the $\eta=\eta_{\max}$ computational boundary and the outer physical boundary becomes the $\eta=\eta_{\min}$ computational boundary. 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.

Another advantage of this approach is the ability to adjust arbitrarily the mesh spacing on the airfoil surface or in the mesh interior, with the provision that the smoothness of the mesh is not disrupted. This, in theory, could be used to cluster mesh points around any gradients in the flowfield (e.g., shock waves or the leading-edge stagnation point). Several investigations have experimented with various aspects of this option.[13,14] The next section introduces the method of generating the finite-difference meshes used in the present study.

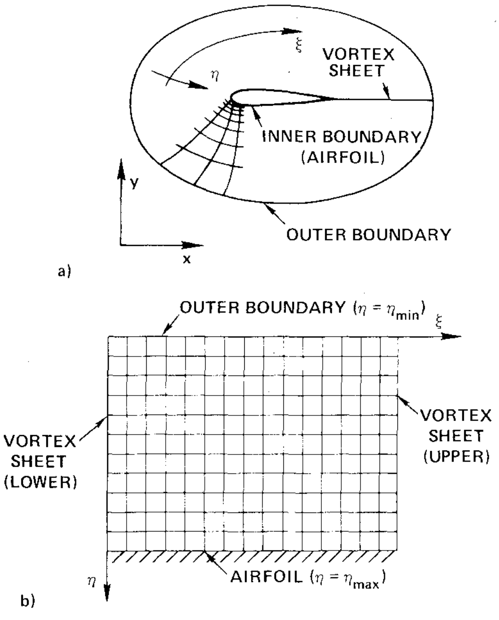

Fig. 1 Numerically generated finite-difference mesh: a) Physical domain; b) Computational domain.

III. Grid Generation

The automatic grid generation scheme introduced by Thompson et al.[6] has been adapted for use in this study. Basically, this grid generation scheme uses numerically generated solutions of Laplace's equation (or in some cases Poisson's equation) to establish regular and smooth finite-difference meshes around arbitrary bodies. These equations are transformed to (and solved in) the computational domain (i.e., ξ and η are the independent variables and x and y are the dependent variables). The transformed equations are given by

$$Ax_{\xi\xi}-2Bx_{\xi\eta}+Cx_{\eta\eta}=0$$
$$Ay_{\xi\xi}-2By_{\xi\eta}+Cy_{\eta\eta}=0 \tag{6}$$

where

$$A=x_\eta^2+y_\eta^2, \ B=x_\xi x_\eta+y_\xi y_\eta, \ C=x_\xi^2+y_\xi^2 \tag{7}$$

The numerical solution of Eq. (6) is achieved by first replacing all derivatives in Eqs. (6) and (7) by standard second-order-accurate finite differences. A residual operator can be defined and is given by

$$L(\ )_{i,j}=[A_{i,j}\delta_{\xi\xi}-2B_{i,j}\delta_{\xi\eta}+C_{i,j}\delta_{\eta\eta}](\ )_{i,j} \tag{8}$$

where the i and j subscripts indicate position in the finite-difference mesh. The operators used in Eq. (8) are defined by

$$\delta_{\xi\xi}(\ )_{i,j}=(\ )_{i+1,j}-2(\ )_{i,j}+(\ )_{i-1,j}$$
$$\delta_{\xi\eta}(\ )_{i,j}=\tfrac{1}{4}[(\ )_{i+1,j+1}-(\ )_{i+1,j-1}-(\ )_{i-1,j+1}+(\ )_{i-1,j-1}]$$
$$\delta_{\eta\eta}(\ )_{i,j}=(\ )_{i,j+1}-2(\ )_{i,j}+(\ )_{i,j-1} \tag{9}$$

The spatial increments ($\Delta\xi$ and $\Delta\eta$) are equal to 1 and, therefore, have been omitted. Once boundary values and an initial solution for $x_{i,j}$ and $y_{i,j}$ have been established, the final interior values can be computed by relaxation. Usually successive overrelaxation (SOR) or successive line overrelaxation (SLOR) is used for this purpose. However, in the present study, because of speed motivations, a faster relaxation scheme is desired. As a result an alternating-direction-implicit (ADI) relaxation algorithm was developed. This algorithm, which is of the AF type, can be expressed by choosing the N operator of Eq. (1) to be

$$\alpha N(\ )_{i,j}=-(\alpha-A_{i,j}^n\delta_{\xi\xi})(\alpha-C_{i,j}^n\delta_{\eta\eta})(\ )_{i,j} \tag{10}$$

where α is a relaxation parameter which will be discussed shortly. Multiplying out the two factors of Eq. (10) yields an approximation to the L operator [Eq. (8)] plus two error terms. This approximate L operator does not have the mixed-derivative term ($\phi_{\xi\eta}$) contained in the exact L operator and has time-linearized coefficients (A and C). In spite of these approximations, unconditional linear stability exists for this scheme and has been proven in Ref. 15. Because of the implicit construction of this scheme, each point in the finite-difference mesh influences every other point during each iteration. As a result, evolution of the solution proceeds at a much faster rate.

The scheme can be restated in practical terms using two steps as follows:

Step 1:

$$(\alpha - A^n_{i,j}\delta_{\xi\xi})f^n_{i,j} = \alpha\omega L x^n_{i,j} \tag{11a}$$

$$(\alpha - A^n_{i,j}\delta_{\xi\xi})g^n_{i,j} = \alpha\omega L y^n_{i,j} \tag{11b}$$

Step 2:

$$(\alpha - C^n_{i,j}\delta_{\eta\eta})(x^{n+1}_{i,j} - x^n_{i,j}) = f^n_{i,j} \tag{12a}$$

$$(\alpha - C^n_{i,j}\delta_{\eta\eta})(y^{n+1}_{i,j} - y^n_{i,j}) = g^n_{i,j} \tag{12b}$$

where $f^n_{i,j}$ and $g^n_{i,j}$ are intermediate results stored at each point in the finite-difference mesh. In step 1, the f and g arrays are obtained by solving two tridiagonal matrix equations for each η = constant line. The corrected values of x and y are then obtained in the second step from the f and g arrays, respectively, by solving two tridiagonal matrix equations for each ξ = constant line.

Remesh Option

In addition to the standard technique of grid generation just discussed, an option for redistributing the mesh points along ξ = constant lines has been used. This option was first introduced in Ref. 14 and is useful in controlling the first η direction grid spacing away from the airfoil surface ($\Delta\eta_1$). Basically, the scheme involves retaining the lines of constant ξ and discarding lines of constant η. The points on the constant ξ lines are then redistributed according to simple exponential stretching formulas.

One additional feature, which has been added in the present formulation, is the ability to vary the $\Delta\eta_1$ spacing around the airfoil surface. In Ref. 14, $\Delta\eta_1$ is fixed to a single value. In the present formulation, values at the leading edge, trailing edge, and one other arbitrary position (usually near half-chord) can be assigned independent values of $\Delta\eta_1$. The values of $\Delta\eta_1$ in between are connected by smooth transition formulas. With this option, the values of $\Delta\eta_1$ can be controlled even more explicitly than in the original scheme.

IV. Full-Potential Equation Algorithm

The fully implicit, approximate factorization algorithm used in the present study is similar to the AF2 algorithm presented in Ref. 2. In Ref. 2 the AF2 algorithm was used to solve the conservative full-potential equation for transonic flows about circular-arc airfoils. Only nonlifting cases using Cartesian finite-difference meshes with small-disturbance boundary conditions were considered. In this study, the AF2 algorithm is extended to arbitrary airfoils with exact boundary conditions. Topics considered in this section include spatial differencing, the AF2 algorithm, and boundary conditions.

Spatial Differencing

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

$$\overleftarrow{\delta}_\xi\left(\frac{\rho U}{J}\right)_{i+\frac{1}{2},j} + \overleftarrow{\delta}_\eta\left(\frac{\rho V}{J}\right)_{i,j+\frac{1}{2}} = 0 \tag{13a}$$

where

$$U_{i+\frac{1}{2},j} = A_{1_{i+\frac{1}{2},j}}(\phi_{i+1,j} - \phi_{i,j}) + \tfrac{1}{4}A_{2_{i+\frac{1}{2},j}}(\phi_{i+1,j+1} - \phi_{i+1,j-1} + \phi_{i,j+1} - \phi_{i,j-1}) \tag{13b}$$

$$V_{i,j+\frac{1}{2}} = \tfrac{1}{4}A_{2_{i,j+\frac{1}{2}}}(\phi_{i+1,j+1} - \phi_{i-1,j+1} + \phi_{i+1,j} - \phi_{i-1,j}) + A_{3_{i,j+\frac{1}{2}}}(\phi_{i,j+1} - \phi_{i,j}) \tag{13c}$$

The quantities ρ, A_1, A_2, A_3, and J are all stored at integer points in the finite-difference mesh (i.e., at i,j). Values needed at half points (i.e., $i+\frac{1}{2},j$ or $i,j+\frac{1}{2}$) are obtained by using simple averages. The operators,

$$\overleftarrow{\delta}_\xi(\ ) \text{ and } \overleftarrow{\delta}_\eta(\ )$$

are first-order-accurate backward-difference operators in the ξ and η directions, respectively, and are defined by

$$\overleftarrow{\delta}_\xi(\ )_{i,j} = (\ )_{i,j} - (\ )_{i-1,j}$$

$$\overleftarrow{\delta}_\eta(\ )_{i,j} = (\ )_{i,j} - (\ )_{i,j-1} \tag{14}$$

The density calculation is performed in a straightforward manner by using Eq. (4b). Values of U, V, ϕ_ξ, and ϕ_η required for the density calculation are given by

$$\phi_{\xi_{i,j}} = \tfrac{1}{2}(\phi_{i+1,j} - \phi_{i-1,j})$$

$$\phi_{\eta_{i,j}} = \tfrac{1}{2}(\phi_{i,j+1} - \phi_{i,j-1})$$

$$U_{i,j} = (A_1\phi_\xi + A_2\phi_\eta)_{i,j}$$

$$V_{i,j} = (A_2\phi_\xi + A_3\phi_\eta)_{i,j} \tag{15}$$

Special formulas replace Eqs. (13) and (15) at boundaries and will be discussed in a subsequent section.

Equation (13) is a suitable finite-difference scheme for subsonic flow regions. However, for supersonic regions, a properly chosen artificial viscosity term must be added. For example, Jameson[4] adds the following viscosity term to the one-dimensional Cartesian form of the full-potential equation:

$$-\Delta x(\mu\phi_{xx})_x \tag{16}$$

where $\mu = \min[0, \rho(1-M^2)]$ (M is the local Mach number). This is analogous to the switching used in the Murman mixed-difference procedure.[16] It can be shown that this is equivalent to adding[4]

$$-\Delta x(\nu\rho_x\phi_x)_x \tag{17}$$

where $\nu = \max[0, 1-(1/M^2)]$. Exact implementation of Eq. (17) in the present case involving the general ξ-η coordinate system is difficult. Instead, approximate implementation is achieved by using an artificial viscosity of the following form:

$$-\Delta\xi\left(\nu\rho_\xi\frac{|U|}{J}\right)_\xi \tag{18}$$

where $\nu = \max[0, 1-(1/M^2)]$. The absolute value of the U velocity component has been included as a consequence of the wrap-around coordinate system. Equation (18) is a good approximation to Eq. (17) on the upper and lower surfaces of an airfoil where the general ξ-η coordinate system is approximately Cartesian.

The complete finite-difference approximation to Eq. (4a) is given by:

$$\overleftarrow{\delta}_\xi\left(\frac{\rho U}{J}\right)_{i+\frac{1}{2},j}+\overleftarrow{\delta}_\eta\left(\frac{\rho V}{J}\right)_{i,j+\frac{1}{2}}$$

$$-\overleftarrow{\delta}_\xi\left\{\left(\frac{U}{J}\right)_{i+\frac{1}{2},j}\nu_{i+k,j}\left(\rho_{i+\frac{1}{2},j}-\rho_{i+2k-\frac{1}{2},j}\right)\right\}=0 \tag{19}$$

where

$$k=\begin{cases}0 \text{ when } U_{i+\frac{1}{2},j}>0\\ 1 \text{ when } U_{i+\frac{1}{2},j}<0\end{cases} \tag{20}$$

ρ_ξ, in the artificial viscosity term, is evaluated with a backward difference when $U_{i+\frac{1}{2},j}>0$ and with a forward difference when $U_{i+\frac{1}{2},j}<0$. This maintains an upwind influence in the artificial viscosity term for both upper and lower airfoil surfaces. The scheme given by Eq. (19) is centrally differenced and 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 Ref. 2, Eq. (19) can be rearranged to give:

$$\overleftarrow{\delta}_\xi\left[\tilde{\rho}_i\left(\frac{U}{J}\right)_{i+\frac{1}{2},j}\right]+\overleftarrow{\delta}_\eta\left(\frac{\rho V}{J}\right)_{i,j+\frac{1}{2}}=0 \tag{21a}$$

where

$$\tilde{\rho}_i=(1-\nu_{i+k,j})\rho_{i+\frac{1}{2},j}+\nu_{i+k,j}\rho_{i+2k-\frac{1}{2},j} \tag{21b}$$

The addition of the artificial viscosity given by Eq. (18) is thus equivalent to retarding the density in Eq. (13a). Artificial viscosity is not added explicitly as in the Jameson procedure. This will be discussed in more detail in the next section on the iteration scheme.

An artificial viscosity term similar to Eq. (18), but written for the η direction, could be added to the differencing approximation given by Eq. (19). This would have the effect of providing an upwind bias on the density coefficient of the η-direction term of Eq. (21a). Such a scheme would effectively achieve a form of rotated differencing and, perhaps, provide more reliable convergence for some cases. This type of artificial viscosity has not as yet been investigated.

Another formulation of the artificial viscosity term is given by:

$$-\Delta\xi\left(\nu\rho_\xi\frac{A_1}{J}\phi_\xi\right)_\xi \tag{22}$$

where again $\nu=\max[0,(1-1/M^2)]$. The complete finite-difference approximation to Eq. (4a) using this version of artificial viscosity is obtained similarly to Eq. (21a) and will not be presented. Both versions of artificial viscosity have been tried and, in the limited number of cases in which they were compared, produced the same results.

The choice of ν strongly affects accuracy and stability. The particular choice $\nu=\max[0,(1-1/M^2)]$ results in an adequate artificial viscosity for only small regions of supersonic flow. For larger regions of supersonic flow (i.e., regions terminated by a strong shock wave), the vlaue of ν must be increased to maintain stability. For these cases, ν has been chosen to be

$$\nu=\max[0,(M^2-1)c] \tag{23}$$

where c is a user-specified constant, usually set equal to 1.0-2.0. Use of Eq. (23), instead of the standard definition for ν, increases the amount of upwinding or, equivalently, the amount of artificial viscosity in supersonic flow regions.

AF2 Iteration Scheme

The AF2 fully implicit AF scheme used in the present study can be expressed by choosing the N operator of Eq. (1) as follows:

$$\alpha NC^n_{i,j}=-\left[\alpha-\overrightarrow{\delta}_\eta\left(\frac{\rho A_3}{J}\right)_{i,j-\frac{1}{2}}\right]\left[\alpha\overleftarrow{\delta}_\eta-\overleftarrow{\delta}_\xi\tilde{\rho}_i\left(\frac{A_1}{J}\right)_{i+\frac{1}{2},j}\overrightarrow{\delta}_\xi\right]C^n_{i,j} \tag{24}$$

where $\tilde{\rho}_i$ is defined by Eq. (21b) and α is a free parameter to be defined subsequently. Implementation of the AF2 scheme is achieved by writing it in a two-step form given by:

Step 1:

$$\left[\alpha-\overrightarrow{\delta}_\eta\left(\frac{\rho A_3}{J}\right)_{i,j-\frac{1}{2}}\right]f^n_{i,j}=\alpha\omega L\phi^n_{i,j} \tag{25}$$

Step 2:

$$\left[\alpha\overleftarrow{\delta}_\eta-\overleftarrow{\delta}_\xi\tilde{\rho}_i\left(\frac{A_1}{J}\right)_{i+\frac{1}{2},j}\overrightarrow{\delta}_\xi\right]C^n_{i,j}=f^n_{i,j} \tag{26}$$

where $f^n_{i,j}$ is an intermediate result stored at each mesh point. In step 1, the f array is obtained by solving a simple bidiagonal matrix equation for each $\xi=$ constant line. The correction array $(C^n_{i,j})$ is then obtained in the second step from the f array by solving a tridiagonal matrix equation for each $\eta=$ constant line. The nature of this AF2 factorization places a sweep-direction restriction on both steps, namely, outward (away from the airfoil) for the first step and inward (toward the airfoil) for the second step. There are no limitations placed on either of the two sweeps due to flow direction.

The N operator, as presented by Eq. (24) has been modified relative to the original algorithm of Ref. 2. For the AF2 factorization, the N operator must be written so that either the ξ- or η-difference approximation is split between the two factors. This construction generates either a $\phi_{\xi t}$- or a $\phi_{\eta t}$-type term, and if it is differenced properly,[17] provides time-dependent dissipation to the convergence process. If the ξ direction is split, corresponding to the Ref. 2 version, the resulting $\phi_{\xi t}$ term will be properly upwind differenced on either the upper or lower airfoil surface but not both. Instead, the η direction is split. This allows control of the $\phi_{\xi t}$ term because it is added explicitly and is not part of the factorization construction. The $\phi_{\xi t}$ term is included by adding

$$\mp\alpha\beta\overleftrightarrow{\delta}_\xi$$

inside the brackets of the second factor. The parameter β is a user-specified constant which can be adjusted as needed. The double arrow notation on the ξ-difference operator indicates that the difference is always upwind, which on the upper surface is a backward difference, and on the lower surface is a forward difference. The sign is chosen so that the addition of $\phi_{\xi t}$ increases the magnitude of the second-sweep diagonal.

Of course, with this N operator construction, the $\phi_{\eta t}$-type term is properly differenced in only half the mesh, either the forward or aft half. Both versions have been tested and seem to yield equivalent results. However, most experimentation has been conducted on the present form in which $\phi_{\eta t}$ is properly differenced in the forward half of the mesh only. Adverse effects for cases with supersonic flow at the trailing edge may be anticipated with this form, but none have been experienced. In fact, cases with freestream Mach numbers near unity, as well as supersonic freestream cases, have been computed, in which the trailing edge is entirely imbedded in supersonic flow with no adverse effects.[18]

Normally, flowfield, type-dependent differencing is used to achieve stability in transonic flow calculations. Incorporating these different operators into iteration procedures, such as the AF2 scheme presented here, would be cumbersome if not impossible. Using the upwind bias of the density coefficient, which is always evaluated at the nth iteration level, allows the simple two- and three-banded matrix form of the AF2 scheme to be retained over the entire flowfield, even in regions of supersonic flow. In fact, use of upwinded density coefficients in any general iteration scheme (i.e., for any arbitrary N operator) could be used to remove the difficulties introduced by type-dependent differencing. The resulting general scheme would retain the same basic differencing (at the $n+1$ iteration level) throughout the entire flowfield, relying on the upwind bias of the density (at the nth iteration level) to provide the artificial viscosity in supersonic flow regions. This represents a significant simplification in the handling of supersonic flow regions for transonic flow calculations.

The quantity α introduced in Eq. (24) (α is also used in the grid generation section) 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 Ref. 1, 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. A suitable α sequence with analytically estimated endpoints is given in Ref. 2.

An AF2 scheme stability analysis using the standard linearized equation has been presented in Ref. 3. As expected, the fully implicit AF2 scheme is unconditionally stable. However, for the present case, the AF2 scheme has been modified in one important way. The mixed-derivative term, which appears as a result of the general transformation (see Sec. II), is not represented on the left-hand side of the iteration scheme (i.e., the mixed-derivative term is not represented at the $n+1$ iteration level in the N operator). Despite this modification, unconditional stability is still maintained and is discussed in more detail in Ref. 19.

Boundary Conditions

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

$$(\rho V/J)_{i,NJ-\frac{1}{2}} = -(\rho V/J)_{i,NJ+\frac{1}{2}} \tag{27}$$

where $j=NJ$ is the airfoil surface. In other expressions [Eqs. (13) and (15)] where ϕ_η is required at the airfoil surface, the $V=0$ boundary condition is used again to obtain

$$\phi_\eta \Big|_{\text{surface}} = -\frac{A_2}{A_3}\phi_\xi \Big|_{\text{surface}} \tag{28}$$

At the outer boundary of the computational mesh the velocity potential and density are held fixed at the initial freestream values. For lifting cases, the velocity potential distribution on the outermost ring ($\eta=\eta_{\min}$) is updated by the usual compressible vortex solution with circulation strength Γ. At the end of each iteration the circulation is recomputed from the trailing-edge velocity potential jump

$$\Gamma = \phi_{U_{TE}} - \phi_{L_{TE}}$$

At the beginning of the next iteration, during the residual calculation, this jump in velocity potential is imposed along the entire vortex sheet. When differencing the correction across the vortex sheet, the jump condition becomes

$$\Gamma^{n+1} - \Gamma^{n} = C_U^n - C_L^n \tag{29}$$

which is difficult to impost since Γ^{n+1} is unknown. A suitable alternative is to estimate the value of Γ^{n+1} from

$$\Gamma^{n+1} = 3(\Gamma^{n} - \Gamma^{n-1}) + \Gamma^{n-2} \tag{30}$$

and to use that value in Eq. (29). Since the evolution of the circulation is a smooth and continuous process, Eq. (30) provides suitable accuracy for predicting Γ^{n+1}. To maintain stability, the new value of the circulation must be slightly underrelaxed. A relaxation factor of about 0.7-0.8 seems to achieve the necessary critical damping.

Special boundary conditions inherent in the AF2 iteration algorithm arise at the airfoil surface. During each iteration, at the beginning of the first sweep, a boundary condition on f [see Eq. (25)] must be applied at the airfoil surface. Because f is a complicated function with little physical meaning, specification of its value is difficult. Since f approaches zero as a steady-state solution is approached, this boundary condition cannot affect solution accuracy. However, a poor choice can slow convergence or even cause instability. For lack of a better boundary condition, $f_\eta = 0$ is used and seems to produce acceptable results.

V. Computed Results

The implicit algorithm introduced in the previous section is evaluated in this section by presenting a range of numerically computed examples, including subsonic, transonic, and transonic shock-free cases. In all cases, the numerically generated finite-difference mesh was converged until the maximum residual dropped by three orders of magnitude. For a typical mesh with about 4000 grid points, this generally required 30-40 iterations and about 5 s of CPU time on the Ames 7600 computer. An example of a numerically generated finite-difference mesh for the NACA 0012 airfoil is shown in Fig. 2. There are 149 points in the ξ direction and 28 in the η direction. Figure 3 shows the same view as that of Fig. 2 but with the effects of the remesh option added. The $\Delta\eta_1$ spacings for this mesh were chosen to be 1.5%, 4.8%, and 0.8% of chord at the leading-edge, half-chord, and trailing-edge positions, respectively. It has been found to provide a more accurate solution especially in cases with a sharp leading-edge expansion.

The mesh presented in Fig. 3 has been used to compute a series of calculations for the NACA 0012 airfoil. A range of freestream conditions has been chosen for these calculations in an attempt to evaluate the new algorithm in terms of convergence speed and accuracy. Freestream conditions for these cases include 1) $M_\infty = 0.72$, $\alpha = 0$ deg, 2) $M_\infty = 0.63$, $\alpha = 2$ deg, 3) $M_\infty = 0.75$, $\alpha = 1$ deg, and 4) $M_\infty = 0.75$, $\alpha = 2$ deg. Maximum residual convergence histories for all of these cases are displayed in Fig. 4. In each case the maximum residual has been normalized by the maximum residual from the first iteration. The convergence rate for all cases has been approximately optimized by choosing the α-sequence endpoints using a trial-and-error process. As is clearly evident, the first case, which is a simple subsonic and nonlifting calculation, is quite fast, taking only 66 iterations to drop the residual six orders of magnitude. The second case is a subsonic lifting case and is approximately twice as slow as the previous nonlifting case. This degradation in convergence speed is caused by the addition of lift into the problem and specifically by the approximate technique for updating the circulation in the vortex solution. Although the present method has proved to be much more efficient than several

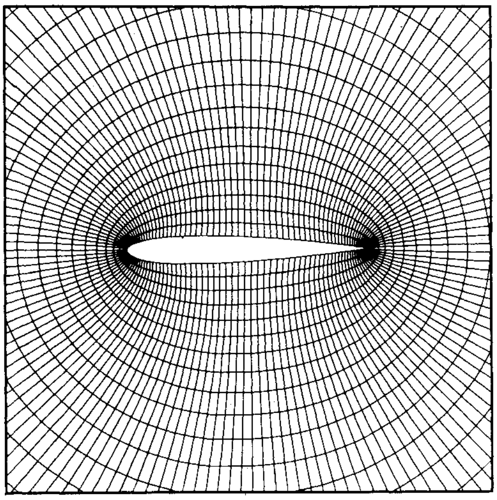

Fig. 2 Finite-difference mesh around NACA 0012 airfoil (149 × 28).

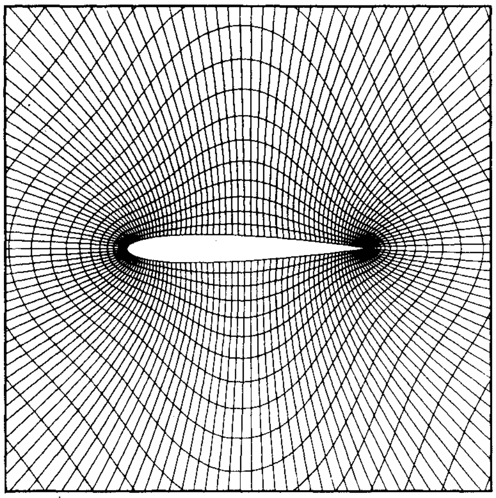

Fig. 3 Finite-difference mesh around NACA 0012 airfoil with remesh option (149 × 28).

other versions, it is felt that a significant increase in convergence speed could be realized with improvements in this area.

The third case presented in Fig. 4 is a lifting transonic case with a moderate strength shock wave. The maximum residual for this solution is reduced six orders of magnitude in just 155 iterations. The final number of supersonic points is achieved in 47 iterations, and 99% of the lift is achieved in 53 iterations. Generally speaking, plotable accuracy is achieved for this algorithm with a three-order-of-magnitude reduction in the maximum residual which, for this case, was achieved in 83 iterations.

The final case in this series of calculations is a transonic lifting case with a relatively strong shock wave. The strong supersonic region plus the existence of a relatively large amount of lift ($C_L = 0.57$) cause a reduction in the convergence speed. The primary factor causing this reduction is again the approximate technique for updating the circulation in the vortex solution. An additional cause (related to the existence of strong supersonic flow regions) is the addition of time-dependent dissipation, $\beta\phi_{\xi t}$, which is necessary in this case to maintain stability. The parameter β for this case was chosen to be 1. None of the previous cases required the use of $\phi_{\xi t}$ to maintain stability and, therefore, did not suffer in convergence speed in this respect. Cases with much stronger shock waves (and, therefore, much larger regions of supersonic flow) have been computed with the present technique. A

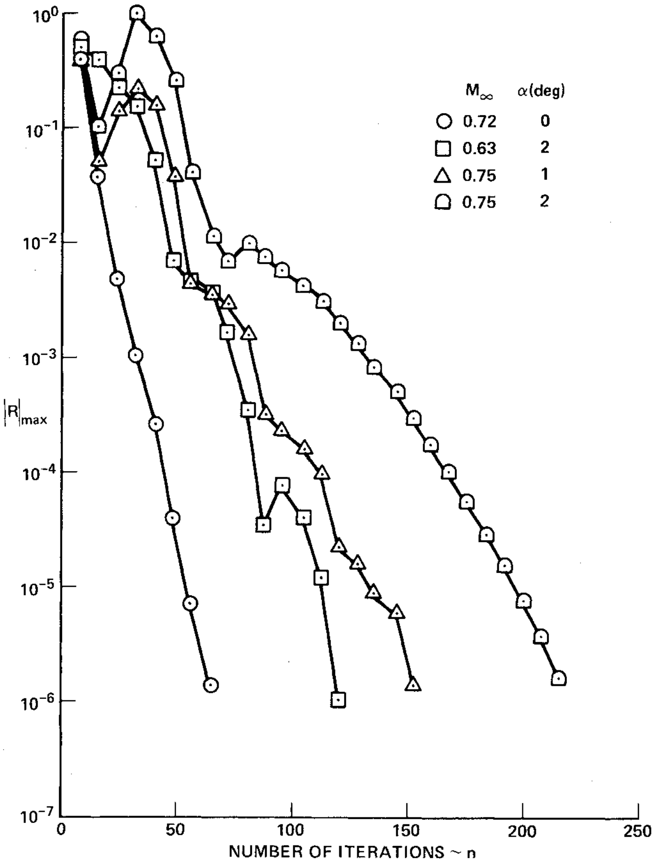

Fig. 4 Maximum residual convergence histories for a series of calculations using an NACA 0012 airfoil.

crucial ingredient for the successful convergence of these more difficult cases is the proper specification of β.[20] Values of β as high as 6 or 8 have been used to stabilize various strong shock calculations.[18]

Each iteration of this new implicit algorithm is only slightly more complicated than a single iteration of an SLOR algorithm. The new algorithm takes approximately 0.1 s of CPU time per iteration for the standard 149 × 28 mesh. This is only about 10% more computer time per iteration than the Jameson nonconservative FL06 computer program using an SLOR algorithm for an equivalent finite-difference mesh.

Pressure coefficient distributions for two of the four cases just presented are displayed in Figs. 5 and 6. The first pressure coefficient distribution is for case 2 ($M_\infty = 0.63$, $\alpha = 2$ deg), which is a lifting subsonic calculation. This result is compared with the result of Ref. 21. The results are in excellent agreement.

The second pressure coefficient distribution is for case 4 ($M_\infty = 0.75$, $\alpha = 2$ deg), which is a lifting transonic calculation involving a strong shock wave. Both conservative and nonconservative results of Ref. 22 are shown in Fig. 6 for comparative purposes. The difference between the conservative and the nonconservative shock positions and strengths is quite striking, indicating the need for a conservative technique. The present pressure coefficient distribution is in excellent agreement with the Ref. 22 conservative result. The largest disagreement is at the shock wave where the shock strengths are in good agreement, but the positions differ by about 1% of chord, causing an error in the lift coefficient of about 1%. These disagreements are probably caused by slight differences in the two finite-difference meshes. For instance, the Ref. 22 mesh is finer near the shock wave in the ξ direction than the present standard mesh.

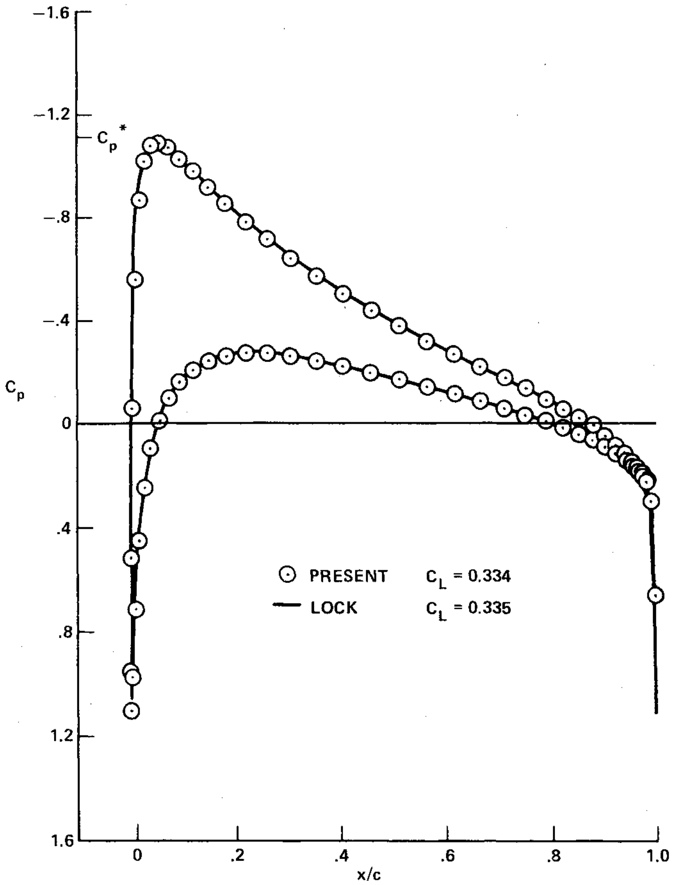

Fig. 5 Pressure coefficient comparison (NACA 0012, $M_\infty = 0.63$, $\alpha = 2$ deg).

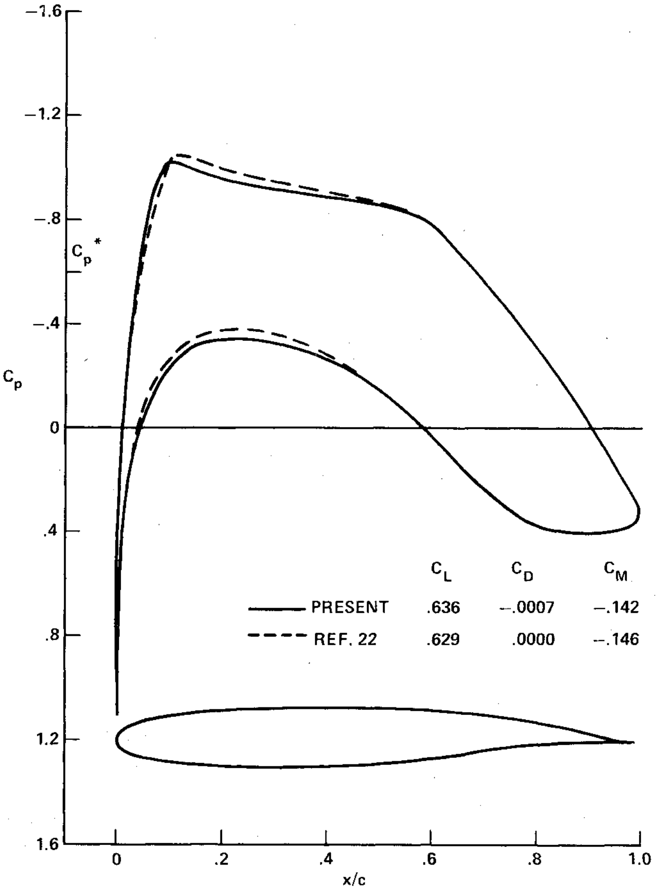

Fig. 7 Pressure coefficient comparison (airfoil 75-06-12, Ref. 22, $M_\infty = 0.75$, $\alpha = 0.12$).

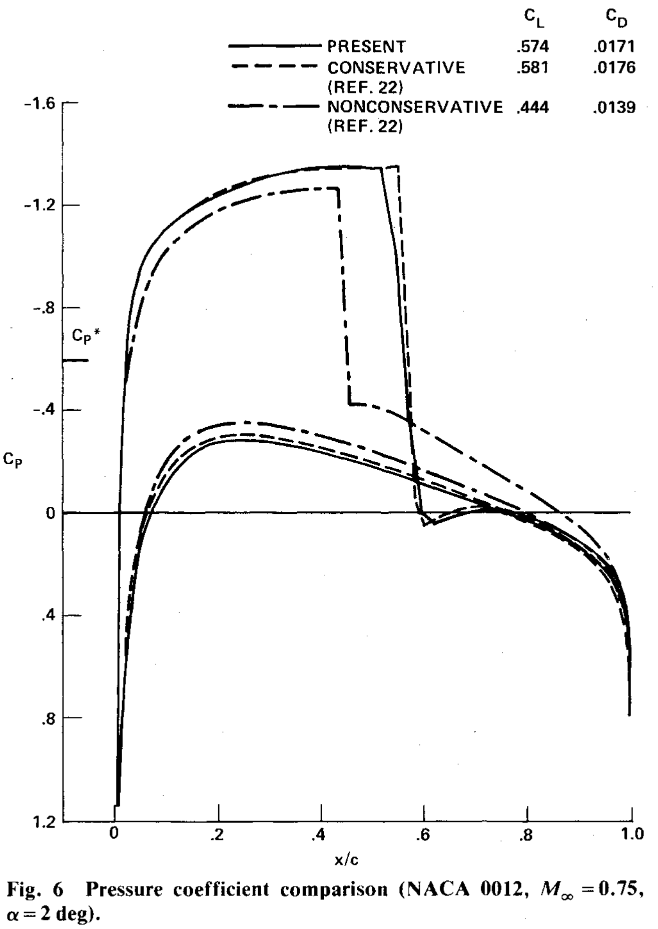

Fig. 6 Pressure coefficient comparison (NACA 0012, $M_\infty = 0.75$, $\alpha = 2$ deg).

The final pressure coefficient comparison, which was computed for the transonic shock-free airfoil of Ref. 22 (airfoil 75-06-12), is presented in Fig. 7. The freestream Mach number for this calculation is 0.75, and the angle of attack is 0.12 deg. The pressure coefficient distribution in Ref. 22 was computed using an isentropic hodograph design technique. The present result was computed without the remesh option. For cusped trailing-edge airfoils, the remesh option has a tendency to provide inappropriate clustering at the trailing edge. This can result in pressure oscillations at the trailing edge which may have an adverse effect on the solution's circulation level. To offset the loss of the remesh clustering, the number of points in the η direction of the standard mesh has been increased to 42. The results, presented in Fig. 8, are in good agreement. The small discrepancies which do exist can probably be attributed to the sensitive nature of this airfoil. It should be pointed out that an accuracy optimization of this solution by applying small perturbations in freestream Mach number or angle of attack was not required.

VI. Conclusions

A fast, fully-implicit algorithm for solving the conservative full-potential equation for the transonic flow past arbitrary airfoils has been developed. This new scheme is of the approximate factorization variety. The spatial differencing used to approximate the full-potential equation is similar to the differencing presented in Ref. 4. However, in the present approach, the artificial viscosity required to maintain stability in supersonic regions is not added explicitly. It is introduced by spatially retarding the density in the finite-difference equation. This strategy greatly simplifies the solution procedure and effectively allows the simple two- and three-banded matrix form of the AF scheme to be retained over the entire flowfield, even in regions of supersonic flow. Pressure coefficient distributions show good agreement with independent results and are computed in substantially less computer time than standard successive line overrelaxation algorithms.

In the present study, special attention has been placed on the treatment of exact airfoil surface boundary conditions. This is achieved by using the numerical transformation of Ref. 6 to establish an arbitrary, nonorthogonal finite-difference mesh. The efficiency of the numerical grid generation scheme has been improved in the present study by using a fully-implicit AF relaxation scheme, which is similar to the full-potential solution procedure.

The present full-potential equation AF algorithm coupled with the numerical grid generation routine represents an efficient, flexible tool for computing transonic airfoil flowfields. There are essentially no restrictions with regard to the airfoil geometry. Unlike techniques utilizing circle plane mappings, the present technique is extendable to three-dimensional cases and has the ability to directly model the effects of wind-tunnel walls. Future extensions, including both of these aspects, are currently in progress.

References

[1] Ballhaus, W. F., Jameson, A., and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," AIAA Paper 77-634, June 1977; also *AIAA Journal*, Vol. 16, June 1978, pp. 573-579.

[2] Holst, T. L. and Ballhaus, W. F., "Conservative Implicit Schemes for the Full Potential Equation Applied to Transonic Flows," NASA TM 78469, March 1978; also *AIAA Journal*, Vol. 17, Feb. 1979, pp.145-152.

[3] Ballhaus, W. F. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equation," NASA TM X-73,082, 1975.

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

[5] Sells, C.C.L., "Plane Subcritical Flow Past a Lifting Airfoil," *Proceedings of the Royal Society of London*, Vol. 308A, 1968, pp. 377-401.

[6] Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, 1974, pp. 299-319.

[7] Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Compuation," *Communications on Pure and Applied Mathematics*, Vol. 7, No. 1, 1954, pp. 159-193.

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

[9] Lapidus, A., "A Detached Shock Calculation by Second-Order Finite Differences," *Journal of Computational Physics*, Vol. 2, 1967, pp. 154-177.

[10] Viviand, H., "Conservative Forms of Gas Dynamic Equations," *La Recherche Aerospatiale*, No. 1, Jan.-Feb. 1974, pp. 65-68.

[11] Vinokur, M., "Conservative Equations of Gas Dynamics in Curvilinear Coordinate Systems," *Journal of Computational Physics*, Vol. 14, Feb. 1974, pp. 105-125.

[12] Steger, J. L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665, June 1977.

[13] Hodge, J. K. and Stone, A. L., "Numerical Solution for Airfoils Near Stall in Optimized Boundary-Fitted Curvilinear Coordinates," AIAA Paper 78-284, Jan. 1978.

[14] Sorenson, R. L. and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations," NASA TM 73252, Aug. 1977.

[15] McKee, S. and Mitchell, A. R., "Alternating Direction Methods for Parabolic Equations in Two Space Dimensions with a Mixed Derivative," *The Computer Journal*, Vol. 13, Feb. 1970, pp. 81-86.

[16] Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," *Proceedings of AIAA Computational Fluid Dynamics Conference*, Palm Springs, Calif., July 1973.

[17] Jameson, 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.

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

[19] Holst, T. L., "An Implicit Algorithm for the Conservative, Transonic Full Potential Equation Using an Arbitrary Mesh," AIAA Paper 78-1113, July 1978.

[20] South, J. C., private communication, 1978.

[21] Lock, R. C., "Test Cases for Numerical Methods in Two-Dimensional Transonic Flows," AGARD Rept. 575, 1970.

[22] Bauer, F., Garabedian, P., Korn, D., and Jameson, A., "Supercritical Wing Sections," *Lecture Notes in Economics and Mathematical Systems,* 108, Springer-Verlag, 1975.