Implicit Approximate-Factorization Schemes for Steady Transonic Flow Problems

W. F. Ballhaus*
NASA Ames Research Center and Aeromechanics Lab., U.S. Army Aviation R&D Command, Moffett Field, Calif.
A. Jameson†
Courant Institute of Mathematical Sciences, New York University, New York, N.Y.
and
J. Albert‡
NASA Ames Research Center, Moffett Field, Calif.

Implicit appproximate-factorization (AF) algorithms are developed for the solution of steady-state transonic flow problems. The performance of the AF solution method is evaluated relative to that of the standard solution method for transonic flow problems, successive line over-relaxation (SLOR). Both methods are applied to the solution of the nonlinear, two-dimensional transonic small-disturbance equation for flows about representative transonic airfoils. Results indicate that the AF method requires substantially less computer time than SLOR to solve the nonlinear finite-difference matrix equation for the flowfield. This increase in computational efficiency is achieved with virtually no increase in computer storage or coding complexity.

Introduction

THE objective is to determine the feasibility of using implicit approximate-factorization (AF) algorithms to solve steady-state transonic flow problems. In this investigation, the AF procedure is compared in terms of efficiency, reliability, and flexibility with successive line over-relaxation (SLOR), the standard steady-state transonic flow solution method.

Murman and Cole[1] developed the first computer-programmable SLOR algorithm for the solution of transonic flow problems. That solution procedure proved to be about an order-of-magnitude more efficient computationally than the first computer-programmed transonic flow algorithm —the explicit, time-accurate procedure of Magnus and Yoshihara.[2] Since that time, the Murman-Cole procedure has been improved, extended, and applied to a variety of aerodynamic problems (reviews are given in Refs. 3 and 4). SLOR algorithms have generally proved to give reliable, but slow, convergence and to be readily adaptable to a wide range of applications.

The main limitation on the class of problems that can be treated by SLOR is the computer time required to obtain a converged solution. Hence, a number of potentially more efficient iterative solution procedures have been proposed as alternatives to SLOR. These methods, which have been used successfully to accelerate convergence for purely elliptic problems, are: 1) semidirect methods using fast Poisson solvers; 2) extrapolation, and 3) the multigrid approach. However, they all seem to have some adverse features restricting their use as general solution procedures for practical transonic (mixed elliptic-hyperbolic) problems.

In recent years, fast direct methods have been developed for solving matrix equations for the discrete Laplacian.[5-7] For the transonic, small-disturbance potential equation

$$(k-\phi_x)\phi_{xx}+\phi_{yy}=0 \tag{1}$$

this suggests an iteration procedure of the form

$$\phi_{xx}^{n+1}+\phi_{yy}^{n+1}=(1+\phi_x^n-k)\phi_{xx}^n \tag{2}$$

such that a discrete Poisson equation is solved for each iteration n. Such a scheme has been proposed by Martin and Lomax[8] and Martin.[9] Schemes of this type produce substantial increases in computational efficiency, especially for cases with embedded supersonic regions that are small. However, their application has been limited, because the most efficient Poisson solver algorithms require uniform grid spacing in all but one coordinate direction, and they require special, complex treatment of internal boundaries. A hybrid Poisson solver/SLOR scheme, proposed by Jameson,[10] avoids these difficulties for some special cases of two-dimensional flows by introducing a coordinate mapping in a simple domain.

Extrapolation has proved useful for the acceleration of SLOR convergence in those special cases in which dominant eigenvalue exists and can be identified.[11,12] It has also been used, in a somewhat different form, to accelerate convergence in the fast Poisson solver approach.[8,9]

The multigrid method was first proposed by Federenko,[13] developed by Brandt,[14] and recently applied to the solution of transonic flowfields by South and Brandt.[15] Preliminary results indicate a substantial increase in efficiency over SLOR for computations on uniform meshes. However, the method in its present form is sensitive to the aspect ratio of the mesh cells, leading to complications in the treatment of nonuniform grids. Moreover, difficulties have been encountered in the treatment of transonic flows with the appearance of limit-cycle oscillations which may prevent the calculation from returning to the finest grid, and which are sometimes accompanied by fluctuations in the size of the supersonic zone (unpublished work by South and also by Jameson).

An alternative procedure that has also been used successfully to accelerate convergence for purely elliptic problems is the approximate-factorization approach. Here, we construct AF schemes for the solution of transonic flows. We then test them to determine if they are capable of

Presented as Paper 77-634 at the AIAA Computational Fluid Dynamics Conference, Albuquerque, N. Mex., June 27-28, 1977; submitted June 27, 1977; revision received Feb. 13, 1978.

Index categories: Transonic Flow; Computational Methods.

*Research Scientist. Member AIAA.

†Professor. Member AIAA.

‡Summer employee; presently a student at the University of Santa Clara.

providing accelerated convergence and whether there are any restrictions limiting their use for practical transonic flowfield computations.

Approximate Factorizations

General Requirements

We seek the solution to the model linear difference equation

$$L\phi = (A\delta_{xx} + \delta_{yy})\phi = 0 \tag{3}$$

where δ_{xx} and δ_{yy} are second central difference operators, ϕ is the solution vector, and A is a constant. The analysis that follows is applicable only to the linear model equation. However, it provides some insight into the development of efficient AF solution procedures for nonlinear transonic flows. An iterative solution procedure to solve Eq. (3) can be written

$$NC^n + \sigma L\phi^n = 0 \tag{4}$$

where C^n is the correction vector $\phi^{n+1} - \phi^n$ computed for cycle $n+1$. We will call $R \equiv L\phi^n$ the residual at the nth iteration and σ is a parameter to be determined. N remains to be defined.

Let $e^n = \phi^n - \phi$ be the error after the nth cycle. Then $C^n = e^{n+1} - e^n$, and Eqs. (3) and (4) can be combined to give

$$e^{n+1} = Me^n = M^2 e^{n-1} = \ldots M^{n+1} e^0, \quad M = I - \sigma N^{-1} L \tag{5}$$

where e^0 is the error in the initial guess of the solution. For the iteration procedure to converge, the modulus of every eigenvalue of M must be less than unity. The degree to which the modulus of every eigenvalue of M can be reduced below unity determines the rate at which an iterative solution procedure converges.

The type of an iterative solution procedure is determined by the choice of the operator N. For rapid convergence, N should be chosen to resemble the operator L as closely as possible and, furthermore, $N^{-1}L\phi^n$ must be easily computable. If N is identical to L, for example, then $M = I - \sigma N^{-1} L$ vanishes (for $\sigma = 1$) and an exact solution to $L\phi = 0$ is obtained in a single iteration. In the special case of the discrete Laplacian in a rectangular, uniform Cartesian grid, this can be accomplished by the use of one of the fast Poisson algorithms. It is usually impractical to invert L, however, and one must use an iteration scheme in which N is an approximation to L. For example, for SLOR

$$NC^n_{i,j} = \left[\delta_{yy} - \frac{2A}{\Delta x^2} + \frac{\sigma A}{\Delta x^2} E_x^{-1}\right] C^n_{i,j} = -\sigma L\phi^n_{i,j} \tag{6}$$

where i,j are grid point indices in the x,y directions, $E_x^{-1} C_{i,j} = C_{i-1,j}$, and σ is the relaxation parameter. Note that N contains the δ_{yy} operator in L but contains only the lower diagonal part of the δ_{xx} operator. Hence, SLOR will require more than one cycle to converge. Actually, the required number of cycles will increase as the number of x grid points increases, because a grid point (i,j) is only influenced by a single grid point to the right of it $(i+1,j)$ in the x direction during one cycle.

The underlying idea of the approximate factorization methods is to construct N as a product of two or more factors

$$N = N_1 N_2 \ldots N_k$$

each of which is restricted to a form leading to equations that can be easily solved. For example, each factor may have a block tridiagonal structure. On the other hand, the added flexibility resulting from the use of several factors allows L to be more closely approximated by N. Moreover, algorithms can be constructed along these lines in such a way that the final N operator is fully implicit, so that every grid point is influenced by every other grid point during each cycle.

AF Scheme 1

Considering Eq. (3) for $A > 0$ so that the equation is elliptic, an approximate factorization can be realized as

$$(\alpha - A\delta_{xx})(\alpha - \delta_{yy}) C^n = \sigma\alpha L\phi^n \tag{7}$$

where α is a parameter to be chosen and σ is an overrelaxation factor. Multiplying the two factors on the left, the operator N, which should now be an approximation to αL, has the form

$$N = \alpha L - \alpha^2 I - A\delta_{xx}\delta_{yy}$$

The required equations can easily be solved by letting $f^n = (\alpha - \delta_{yy}) C^n$ and solving directly for f the tridiagonal matrix equation

$$(\alpha - A\delta_{xx}) f^n = \sigma\alpha L\phi^n \tag{8}$$

followed by a direct solution for C^n of the tridiagonal matrix equation

$$(\alpha - \delta_{yy}) C^n = f^n \tag{9}$$

To estimate the rate of convergence for this scheme (which is a reformulation of the Peaceman-Rachford scheme), note that Eq. (7) can be written in the form

$$(\alpha - A\delta_{xx})(\alpha - \delta_{yy})(e^{n+1} - e^n) = \sigma\alpha L e^n \tag{10}$$

Now, assuming periodic boundary conditions and a uniform Cartesian grid, let

$$e^n(x,y) = \sum_{p,q=1}^{\infty} G^n(p,q) e^{ipx} e^{iqy} \tag{11}$$

Because Eq. (10) is linear, we need consider only a single arbitrary Fourier component. Substituting in Eq. (10) and

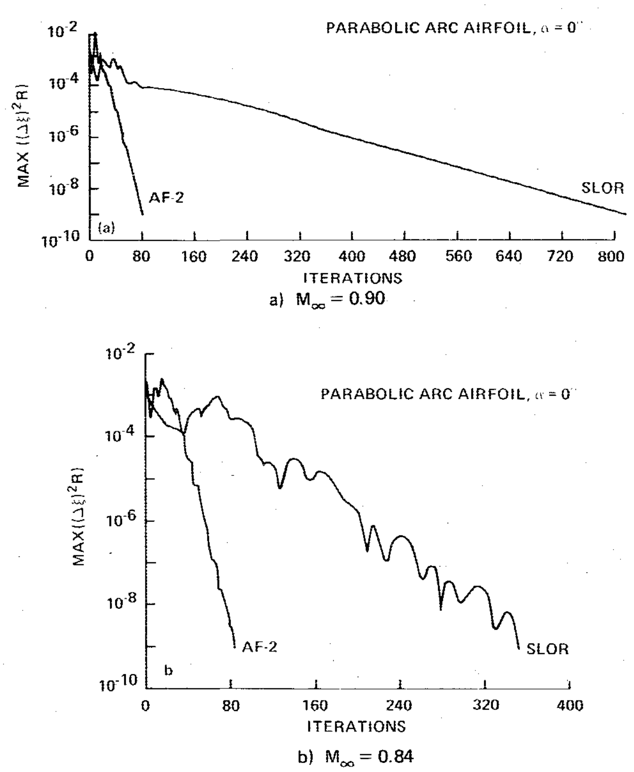

Fig. 1 Optimum convergence histories for a computation on a single fine grid.

rearranging gives

$$\beta=\frac{G^{n+1}}{G^n}=\frac{\alpha-\dfrac{2}{\Delta y^2}(1-\cos\eta)}{\alpha+\dfrac{2}{\Delta y^2}(1-\cos\eta)}\cdot\frac{\alpha-\dfrac{2A}{\Delta x^2}(1-\cos\xi)}{\alpha+\dfrac{2A}{\Delta x^2}(1-\cos\xi)} \qquad (12)$$

where $\eta=q\Delta y$, $\xi=p\Delta x$, and $\sigma=2$. Note that, in the special case $\sigma=2$, the ξ and η dependence in β is separable. This has an interesting consequence that is discussed subsequently. For stability, that is, for the error to approach zero in the iteration procedure,

$$|\beta|<1 \qquad (13)$$

and this condition is satisfied for all ξ and η eigenvalues in Eq. (12).

Now $e^n \sim |\beta|^n e^o$, so that the number of iterations required to reduce the error to a predetermined amount depends on $|\beta|$; that is, fewer iterations are required to achieve a specified degree of convergence as $|\beta|$ decreases. The convergence rate, therefore, depends strongly on the choice of α. The choice

$$\alpha=(2/\Delta y^2)(1-\cos\eta) \qquad (14)$$

gives $\beta=0$ for the particular eigenvalue η. Hence, for a problem with k grid points in the y direction, corresponding to k eigenvalues, the solution process will converge to zero error after k cycles (for the $\sigma=2$ case only).

Precise estimation of the eigenvalues is generally not practical. Instead, a repeating sequence of α's can be used with each element of the sequence chosen to maintain small values of $|\beta|$ in a given range of eigenvalues.[16] Maximum and minimum values of α are estimated and then used to form a geometric sequence to cover the entire eigenvalue spectrum. The lowest eigenvalue is given approximately by $\eta\sim\Delta y$, which, from Eq. (14), gives $\alpha_\ell\sim 1$. For the high-frequency error components, $\eta\sim\pi$ corresponding to $\alpha_h\sim 4/\Delta y^2$. The geometric sequence

$$\alpha_k=\alpha_h(\alpha_\ell/\alpha_h)^{k-1/M-1} \quad k=1,2,3,\ldots M \qquad (15)$$

is then used repetitively during the course of the computation. Recall that the analysis here applies only to the model linear problem. There is no guarantee that either the estimates of α_h and α_ℓ or the form of the sequence, Eq. (15), are optimum in the general nonlinear case.

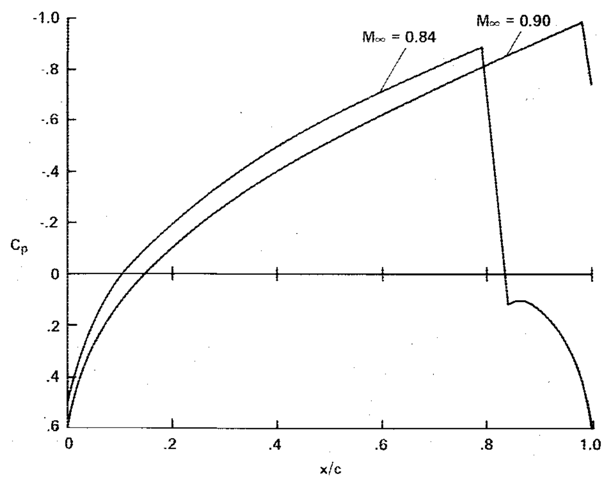

Fig. 2 Surface pressure coefficients on a 10% thick parabolic-arc airfoil for the two test case Mach numbers.

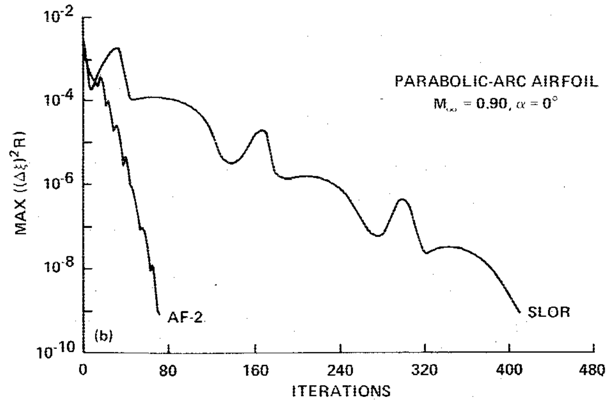

Fig. 3 Optimum fine grid convergence histories for a computation with initial conditions given by a medium grid solution.

In solving Eq. (1) for transonic flow computations, scheme (7) is suitable for the treatment of the subsonic part of the flow, where $A=(k-\phi_x)>0$. To extend the scheme to treat flows with embedded supersonic zones, the factorization is modified at supersonic points, where $A=(k-\phi_x)<0$, by upwind differencing $A\delta_{xx}$ and letting $\alpha=-A\delta_{xx}$ in Eq. (7). On cancelling out a common factor $A\delta_{xx}$ from both sides, the equation becomes

$$(-A\overleftarrow{\delta}_{xx}-\delta_{yy})C^n=L\phi^n \qquad (16)$$

where $\overleftarrow{\delta}_{xx}$ is the upwind second difference operator. Here, because of the use of upwind differencing, it is feasible to shift the term $A\overleftarrow{\delta}_{xx}$ to the second factor while retaining an easily invertible form, and the first factor is essentially eliminated. The result is a transition to the fully implicit marching scheme of Murman and Cole[1] in the supersonic zone. Transition operators are used at the sonic and shock points to maintain conservation form.[17] A term $\beta\overleftarrow{\delta}_x$ (where $\overleftarrow{\delta}_x$ is a first-order, first upwind difference operator) is added to the first factor as a stabilizer with $\beta\sim\Delta x$. At the shock point, $D\overleftarrow{\delta}_x$ is added to the second factor for the first few cycles to attempt to control instabilities associated with shock motion.

In the nonlinear case, the two factors in Eq. (7) do not commute. Consequently, the ordering of the factors can be expected to have some effect on the convergence rate of the method. Taking the factor with δ_{yy} first was found to work best in the cases tested.

Some difficulty was encountered in the application of AF scheme 1 to flowfields with large regions of supersonic flow. In these cases, the initial rapid motion of the shock wave in seeking its final steady-state location frequently rendered the convergence procedure unstable. It was found that the procedure could be stabilized by the addition of $D\overleftarrow{\delta}_x$, as previously mentioned, to retard the shock motion. However, this remedy has two disadvantages: 1) it reduces the convergence speed, and 2) it introduces an uncertainty in the choice of the constant D. Consequently, a more attractive scheme, AF scheme 2, was developed—it maintains stability without explicitly retarding the shock motion.

AF Scheme 2

The type of "moving-shock" instability resulting from the use of AF scheme 1 has been encountered before and it is associated with certain types of implicit schemes. It was studied in some detail by Ballhaus and Steger[18] in the course of their investigation of implicit AF schemes for the low-frequency transonic equation

$$\phi_{xt}=(k-\phi_x)\phi_{xx}+\phi_{yy} \qquad (17)$$

For the conservative schemes tested in Ref. 18, an instability appeared whenever the following two conditions were satisfied: 1) the differencing was shifted from upwind to central across a shock, and 2) the shock propagated at a rate greater than one spatial grid point per time step. However, a nonconservative-in-time (but conservative in the steady state) scheme, described below, was found to be virtually insensitive to this instability. Because the scheme is nonconservative in time, the shock-wave motion depends on the time step and spatial grid spacing, which are nonphysical quantities. Hence, the scheme was rejected as a method for solving unsteady transonic flows. However, it was mentioned as an attractive candidate for a steady-flow solution method because of its insensitivity to the moving shock instability. Note, incidentally, that this instability does not occur with SLOR, a "semi-implicit" scheme, because the flowfield cannot adjust fast enough to violate the second condition previously listed (for relaxation parameters within the linear stability bounds).

AF scheme 2, which is a factorization of Eq. (17), can be written

$$[\alpha-(1-\epsilon_j)A_j\vec{\delta}_x-\epsilon_{j-1}A_{j-1}\overleftarrow{\delta}_x][\alpha\overleftarrow{\delta}_x-\delta_{yy}]C^n$$
$$=\alpha\{[(1-\epsilon_j)A_j\vec{\delta}_x+\epsilon_{j-1}A_{j-1}\overleftarrow{\delta}_x]\overleftarrow{\delta}_x+\delta_{yy}\}\Phi^n \tag{18}$$

where

$$\epsilon_j=\begin{bmatrix}0\\1\end{bmatrix}\text{ for }A_j\begin{bmatrix}>\\<\end{bmatrix}0$$

$A_j=k-(\phi^n_{j+1}-\phi^n_{j-1})/2\Delta x$, $\alpha=(\Delta t)^{-1}$, and $\vec{\delta}_x$, $\overleftarrow{\delta}_x$ are first-order, first forward, and backward difference operators. Note that the algorithm can be coded so that the ϕ array need be stored for only a single time level.[18] [For subsonic flow ($A_j>0$), AF scheme 2 can be derived from AF scheme 1, Eq. (7), by replacing α by $\alpha\overleftarrow{\delta}_x$ and them removing $\overleftarrow{\delta}_x$ from the first factor and the right-hand side. This was noted by the second author, who, working independently, also discovered the subsonic factorization in Eq. (18), originally reported in Ref. 18.]

Substitution of Eq. (11) into Eq. (18) leads to an error growth factor β that is not real. Choosing α to minimize β is therefore more difficult, and, in fact, there is no choice of α that can cause β to be zero for a given eigenvalue. Values of α_ℓ and α_h for Eq. (15) that approximately minimize $|\beta|$ are 1 and $(\Delta y)^{-1}$. Of course, there is no guarantee that Eq. (15) is the optimal functional form for the acceleration parameter sequence.

From the preceding discussion it would appear at first glance that AF scheme 2 cannot be so highly tuned for rapid convergence as AF scheme 1, but it shows potential for greater convergence reliability because of its treatment of shock waves.

Optimized Acceleration Parameters

Determination of the optimum relaxation parameter for SLOR, or the optimum acceleration parameter sequence for approximate factorizations, for a practical computation is a formidable task to achieve analytically. However, comparisons of both optimum and nonoptimum SLOR and approximate factorizations would be useful in assessing the relative merits of these solution procedures. Therefore, we have selected particular representative transonic flow problems for which "optimum" acceleration parameters can be determined by numerical optimization.

The numerical optimization problem is formulated in the following way. First, we select as the objective, that is, the quantity to be minimized, a combination of parameters that represents the computational efficiency associated with a given set of decision variables:

$$\text{OBJ}=(R_N/R_1)^{1/N} \tag{19}$$

OBJ is related to the average decrease in the residual per iteration, R_n is the maximum residual at the nth iteration, and N is the number of iterations required to reduce the maximum residual below some specified value.

Next, we define the decision variables, which are parameters that affect the convergence rate. For SLOR, the decision variable is the subsonic relaxation parameter. The supersonic relaxation parameter is always unity. For both AF schemes, the acceleration parameter sequence is

$$\alpha_k=\alpha_h\gamma^{k-1}\qquad k=1,2,3,\ldots M \tag{20}$$

and the decision variable used in the optimization is γ. The initial guess is $\gamma=(\alpha_\ell/\alpha_h)^{1/(M-1)}$, where α_ℓ and α_h are given in the previous section.

The finite difference AF and SLOR codes were coupled with an executive optimization code, CONMIN.[19] CONMIN is designed to minimize an objective for a set of decision variables, subject to specified constraints. Here, the only constraint imposed was a limit of 2 on the relaxation parameter for SLOR.

The present optimization formulation was designed to be as simple and economical as possible. Only single-parameter optimizations were performed for both the AF and SLOR schemes. There is certainly no guarantee that either the functional form, Eq. (20), or the arbitrarily chosen value of M used are optimum. For the computed solutions presented in the next section, $M=8$.

Computed Results

The convergence characteristics of the SLOR and AF schemes are compared here for two different airfoils.

10% Thick Parabolic-Arc Airfoil

Convergence histories of the flowfield about a nonlifting 10% thick parabolic-arc airfoil are shown in Fig. 1. For the two cases considered, the freestream Mach numbers were 0.84 and 0.90. The converged solutions, in terms of surface pressure coefficients, are shown in Fig. 2. The solutions were computed on a nonuniformly spaced grid with 128×32 x,y points. This grid was obtained by a mapping from a uniformly spaced ξ,η grid. The computations were started from uniform flow ($\phi=0$).

The convergence histories for these cases are given in terms of the maximum value of the residual, multiplied by $(\Delta\xi)^2$ as a function of iteration. This form of the residual was used for this case to be consistent with the code used to compute the SLOR convergence histories. The convergence histories in Fig. 1 were generated by using optimized acceleration and relaxation parameters determined from the simple numerical optimization procedure outlined in the previous section. Note that AF-2 is about ten times faster than SLOR for $M_\infty=0.9$ in terms of the number of iterations required to reduce the

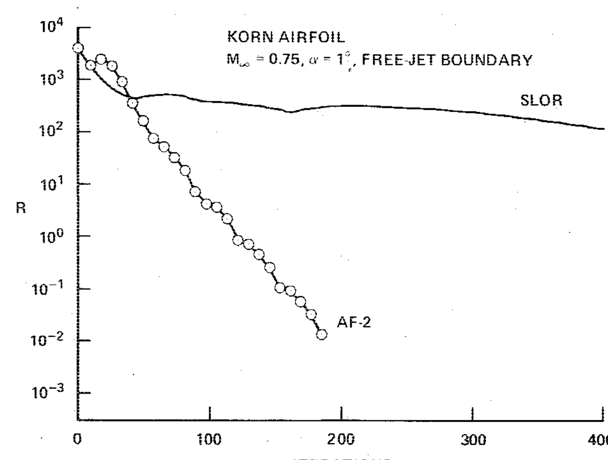

Fig. 4 Nonoptimum convergence histories for a computation on a single fine grid.

maximum residual by about six orders of magnitude. For $M_\infty = 0.84$, AF-2 is only about four times faster.

SLOR convergence can usually be accelerated by the use of coarse-to-fine mesh interpolations. For example, converged solutions on a 64×16 grid were interpolated onto the fine 128×32 grid to provide a good starting solution, and the resulting fine grid convergence history for $M_\infty = 0.90$ is shown in Fig. 3. There is a factor of 2 improvement in SLOR convergence, while the AF-2 scheme shows very little improvement.

The peaks in the AF-2 convergence history correspond to the smaller values of α (i.e., larger values of Δt) in the eight-element sequence. The peaks in the SLOR convergence history correspond to changes in the location of the maximum residual.

Difficulties with the moving shock instability were obtained with the AF-1 scheme in the cases shown in Figs. 1 and 3. Convergence histories for a lower value of M_∞ for AF-1, AF-2, and SLOR are reported in Ref. 20.

Korn Airfoil

Here, we investigate, in some detail, the convergence performance of AF-2 and SLOR for the solution of the flowfield about a lifting transonic airfoil designed by Korn.[21] The SLOR computations were obtained by using TSFOIL, a transonic airfoil analysis code written by Murman and Bailey.[22]

First, we consider the case $M_\infty = 0.75$, $\alpha = 1$ deg where α is the airfoil angle of attack. The far-field boundary condition corresponds to a free-jet wind-tunnel wall condition with a tunnel half-height-to-chord ratio of 8.4. The tunnel wall pressure is uniform and equal to the freestream pressure, implying $\phi = 0$. At the upstream boundary, $\phi = 0$, implying parallel flow ($\phi_y = 0$). At the downstream boundary, freestream pressure is enforced ($\phi_x = 0$). This simplified boundary condition does not depend explicitly on the circulation. It, therefore, allows us to compare AF and SLOR convergence performance for a lifting case without having to consider how that performance is affected by the particular method used to account for changing circulation in the far-field boundary condition. Results are shown for a free-air case, a case in which the circulation does affect the far-field boundary condition, at the end of this section.

The AF and SLOR convergence histories are shown in Fig. 4. The solution was obtained on a variably spaced 85×64 (x,y) grid with uniform flow ($\phi = 0$) as the initial condition. Results are reported in terms of $R = L\phi^n$ to be consistent with the output of TSFOIL. The acceleration and relaxation parameters were not optimized. For AF-2, values of α_h and α_ℓ from the simple growth factor analysis discussed previously were used. In TSFOIL, a fine grid relaxation parameter of 1.95, the default value in the program, was used. Only every eighth iteration was plotted for AF-2 and every tenth for SLOR.

Figure 5 illustrates the improvement in SLOR convergence obtained with the use of mesh interpolation. In this computation, a solution was first obtained in 240 iterations on a coarse 22×16 grid. That solution was interpolated to a medium 43×32 grid, on which a solution was obtained in 200 iterations. The computations on the coarse and medium grids, which represent 65 equivalent fine grid iterations, provided a good initial guess for the fine grid SLOR computation. Multiple grid computations were not carried out in this case for AF-2.

Figure 6 shows the lift coefficient (C_ℓ) convergence histories that correspond to the residual histories shown in Fig. 4. Note that the AF-2 steady C_ℓ is very closely approximated after only 16 iterations, corresponding to two applications of the 8-element acceleration parameter sequence. With SLOR, the coarse- and medium-mesh computations produce a good estimate of the fine grid steady-state lift. Note that without this good initial guess, SLOR requires hundreds of iterations to establish the correct lift.

An attempt was made to optimize the convergence of both AF-2 and SLOR, and the results are shown in Fig. 7. Again, coarse, medium, and fine grids were used with SLOR, whereas only the fine grid was used with AF-2. For AF-2, only every eighth iteration was plotted. This corresponds to the residual after the last element in the sequence, the smallest α. The residual for this α is usually the largest one in the sequence. The optimization procedure improved convergence by increasing α_h by about a factor of 3. This increased the rate at which the residuals near the leading edge were reduced. The optimizer also lowered α_ℓ to maintain the rapid development of the lift, which would have been retarded by merely increasing α_h.

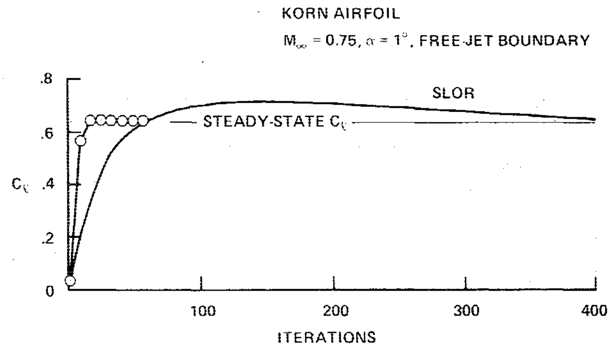

Fig. 6 Lift coefficient development for computations on a single fine grid.

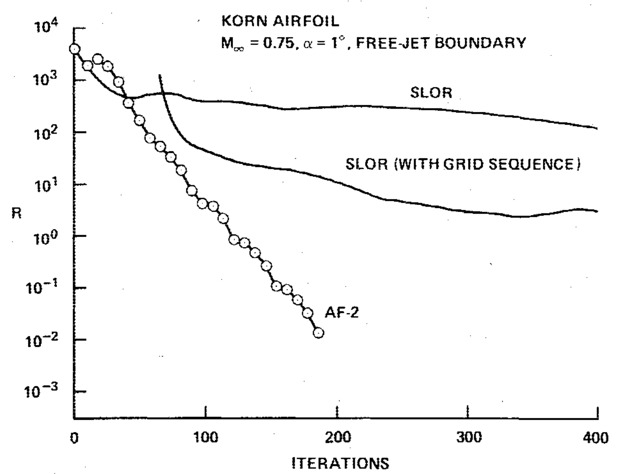

Fig. 5 Nonoptimum convergence histories.

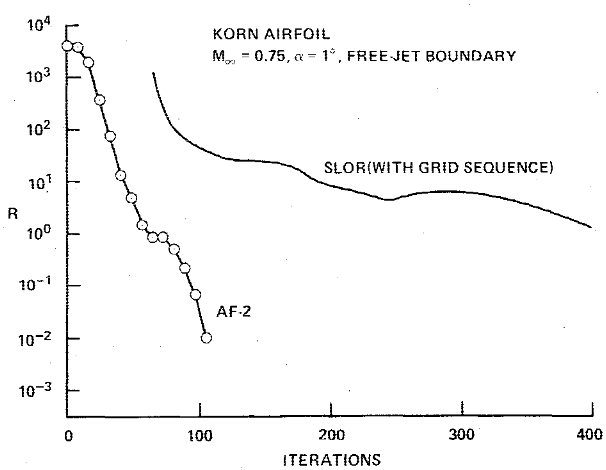

Fig. 7 Optimum convergence histories.

It is interesting to compare the airfoil surface pressure after 16 iterations with the final converged values. The comparison is shown in Fig. 8 and indicates that the shock-wave location and the pressures over much of the airfoil have come very close to reaching their final values.

Convergence histories for a free-air computation are shown in Fig. 9; the converged surface pressure coefficient solution is shown in Fig. 10. The SLOR code, TSFOIL, uses the far-field solution for a compressible vortex and doublet to specify ϕ at the outer edge of the computational domain. The AF-2 code uses only the vortex part, which dominates the far field and which varies as the lift changes during the computation. Presumably, convergence could be accelerated by optimizing the manner in which the lift is updated in the far field. However, no such attempt was made in this initial investigation of transonic AF schemes.

The acceleration parameter limits α_ℓ and α_h for this case were again determined from a simple (linear) growth factor analysis similar to the one following Eq. (12) for AF-1. No optimization was performed. However, the geometric sequence was replaced by a sequence of the form

$$\alpha_I = \alpha_h$$

$$\alpha_k = \alpha_{k-1} - \left(\frac{\alpha_h - \alpha_\ell}{28}\right)(9-k) \quad k=2,3,\ldots 8 \tag{21}$$

This sequence tended to reduce the maximum residual faster than the geometric sequence in this computation for the following reason: Numerical experiments indicated that a shift in the sequence toward α_ℓ tended to speed up the development of lift but slow down the reduction of the maximum residual. (The maximum residual in this case was nearly always at the airfoil leading edge.) Similarly, prejudicing the sequence toward α_h retarded the lift development slightly but reduced the maximum residual at a faster rate. Sequence (21) is biased toward the high-frequency side and thus reduces the maximum residual at a faster rate than the geometric sequence for the same values of α_h and α_ℓ. (The sequence should be adjusted to reduce errors in the flow quantities of interest as rapidly as possible. Shifting the sequence toward α_h to maximize the residual reduction rate can actually decrease the rate of overall error reduction. This is because the residual is a summation of errors that heavily weights the high-frequency error components.[23])

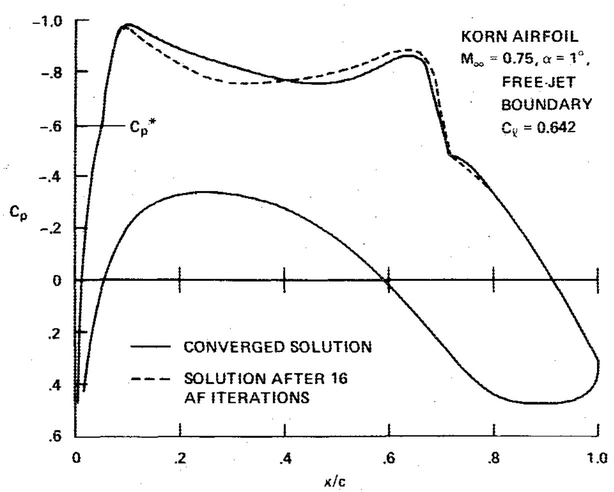

Fig. 8 Computed surface pressure coefficients.

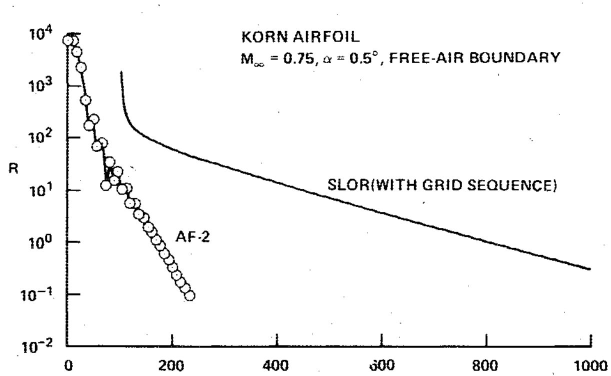

Fig. 9 Nonoptimum convergence histories.

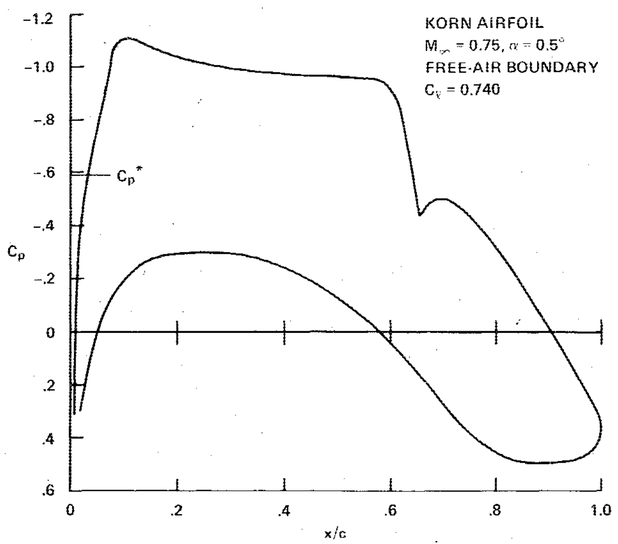

Fig. 10 Computed surface pressure coefficients.

Our experience in computing solutions with both AF-2 and SLOR indicates that they are comparable in convergence reliability. We have not found a case in which SLOR converged and AF-2 did not. Furthermore, we have found that, with a little experience, choosing α_ℓ and α_h to get rapid convergence with AF-2 is no more difficult than choosing relaxation parameters for SLOR. However, it may well be that the speed of the AF procedure could be increased substantially relative to the speed of the computations in this report. This could perhaps be accomplished by the use of acceleration parameter schedules other than the repeating sequences used here. For example, one could develop a strategy for dynamically selecting α at the end of each iteration to effect the largest reduction in residual at the next iteration. Such a strategy has been developed and applied successfully to nonfluid-dynamic problems by Miller and Doss at the University of California, Berkeley (unpublished work).

Concluding Remarks

The present study indicates that implicit, approximate-factorization schemes can provide rapid and reliable convergence in finite-difference transonic flow computations. These schemes can be easily coded, require about the same storage as, and only about 50% more computational work per iteration than, present successive line over-relaxation algorithms. They can be used to obtain solutions on nonuniformly spaced grids, a feature not presently available in some of the other convergence acceleration schemes.

Of the two AF schemes investigated, AF-1 converged the most rapidly. However, it was found to be very susceptible to the "moving shock wave instability" discussed in Ref. 18. AF-2 was found to be virtually insensitive to this instability, and its convergence was found to be at least as reliable as that of SLOR and substantially faster.

In the present effort, the AF procedure was applied only to two-dimensional lifting, small-disturbance cases. Its success in accelerating convergence in these applications is sufficient motivation for the development of AF schemes for three-dimensional problems and for flows governed by the full potential equation. The problem is to devise stable, easily solvable, implicit factorizations for the more complex forms of governing equations involved. An AF solution procedure for the full potential equation has just recently been developed[24] for two-dimensional flows, and the method is presently being tested in three dimensions.

References

[1]Murman, E. M. and Cole, J. D., "Calculation of Plane Steady Transonic Flows," *AIAA Journal*, Vol. 9, Jan. 1971, pp. 114-121.

[2]Magnus, R. and Yoshihara, H., "Inviscid Transonic Flows over Airfoils," *AIAA Journal*, Vol. 8, Dec. 1970, pp. 2157-2162.

[3]Jameson, A., "Transonic Flow Calculations," *VKI Lecture Series: Computational Fluid Dynamics*, von Kármán Institute for Fluid Dynamics, Rhode-St-Genese, Belgium, March 1976.

[4]Ballhaus, W. F., "Some Recent Progress in Transonic Flow Computations," *VKI Lecture Series: Computational Fluid Dynamics*, von Kármán Institute for Fluid Dynamics, Rhode-St-Genese, Belgium, March 1976.

[5]Buneman, O., "A Compact Non-Iterative Poisson Solver," Rept. 294, Stanford University Institute for Plasma Research, 1969.

[6]Buzbee, B. L., Golub, G. H., and Nielsen, C. W., "On Direct Methods of Solving Poisson's Equation," *SIAM Journal of Numerical Analysis*, Vol. 7, Dec. 1970, pp. 627-656.

[7]Fischer, D., Golub, G., Hald, O., Leiva, C., and Widlund, O., "On Fourier Toeplitz Methods for Separable Elliptic Problems," *Mathematics of Computation*, Vol. 28, April 1974, pp. 349-368.

[8]Martin, E. D. and Lomax, H., "Rapid Finite Difference Computation of Subsonic and Transonic Aerodynamic Flows," AIAA Paper 74-11, Washington, D.C., Feb. 1974.

[9]Martin, E. D., "Advances in the Application of Fast Semi-Direct Computational Methods in Transonic Flow," *Symposium Transsonicum II*, Springer-Verlag, 1976, pp. 431-438.

[10]Jameson, A., "Accelerated Iterative Schemes for Transonic Flow Calculations Using Fast Poisson Solvers," ERDA Report C00-3077-82, March 1975, New York University.

[11]Hafez, M. and Cheng, H. K., "Convergence Acceleration and Shock Fitting for Transonic Flow Computations," AIAA Paper 75-51, Pasadena, Calif., Jan. 1975.

[12]Caughey, A. D. and Jameson, A., "Accelerated Iterative Calculation of Transonic Nacelle Flowfields," AIAA Paper 76-100, Washington, D.C., Jan. 1976.

[13]Fedorenko, R. P., "The Speed of Convergence of an Iterative Process," *USSR Computational Mathematics and Mathematical Physics*, Vol. 4, 1964, pp. 559-564 (Russian).

[14]Brandt, Achi, "Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems," *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, Vol. 1, Springer-Verlag, 1973, pp. 82-89.

[15]South, J. C, and Brandt, A., "The Multi-Grid Method: Fast Relaxation for Transonic Flows," *Advances in Engineering Science*, CP-2001, Vol. 4, NASA, 1976, pp. 1359-1369.

[16]Varga, Richard, S., *Matrix Iterative Analysis*, Prentice Hall, Inc., New York, 1962.

[17]Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," *Proceedings of AIAA Computational Fluid Dynamics Conference*, 1973, pp. 27-40.

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

[19]Vanderplaats, G. N., "CONMIN—A Fortran Program for Constrained Function Minimization," TM X-62,282, NASA, Aug. 1973.

[20]Ballhaus, W. F., Jameson, A., and Albert, J., "Implicit Approximate-Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," TM X-73,202, NASA, Jan. 1977.

[21]Kacprzynski, J. J., Ohman, L. H., Garabedian, P. R., and Korn, D. G., "Analysis of the Flow Past a Shockless Lifting Airfoil in Design and Off-Design Conditions," LR-554 (NRC No. 12315), National Research Council of Canada (Ottawa), Nov. 1971.

[22]Murman, E. M. and Bailey, F. R., "TSFOIL—A Computer Code for Two-Dimensional Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation," SP-347, Part II, NASA, March 1977, pp. 769-788.

[23]Ballhaus, W. F., "A Fast Implicit Solution Procedure for Transonic Flows," *Proceedings of the Third International Symposium on Computing Methods in Applied Sciences and Engineering*, Dec. 1977, to be published by Springer-Verlag, Berlin.

[24]Holst, T. L. and Ballhaus, W. F., "Conservative Implicit Schemes for the Full Potential Equation Applied to Transonic Flows," TM-78469, NASA, March 1978.