Newton-Like Minimal Residual Methods Applied to Transonic Flow Calculations

Yau Shu Wong*
Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia
and
McGill University, Montreal, Canada

A new computational technique for the solution of the full potential equation is presented. The method consists of outer and inner iterations. The outer iteration is based on a Newton-like algorithm; a preconditioned minimal residual method is used to seek an approximate solution of the system of linear equations arising at each inner iteration. The present iterative scheme is formulated so that the uncertainties and difficulties associated with many iterative techniques are eliminated—namely, the requirements of acceleration parameters and the treatment of additional boundary conditions for the intermediate variables. Numerical experiments based on the new method for transonic potential flows around an NACA 0012 airfoil at different Mach numbers and different angles of attack are presented and the results are compared with those obtained by the approximate factorization technique. Extension of the present method to three-dimensional flow calculations and its application in finite element methods for fluid dynamics problems are also discussed.

I. Introduction

THE ability to compute transonic flowfields around airfoils or wings is an important aid in the design of efficient modern transport aircraft since they operate predominantly in transonic ranges. In recent years, considerable effort has been spent on the construction of fast and accurate numerical procedures for the solution of the full potential equation. To be useful as a design and analysis tool, the success of a computational procedure should not be problem dependent. For example, some numerical procedures yield rapidly converged solutions if optimal values of the acceleration parameters are provided and if other special conditions are given. However, it should be pointed out that optimal values of these parameters are often unobtainable for practical calculations.

The standard iterative procedure for transonic small-perturbation and full potential calculations was based on the successive line overrelaxation (SLOR) method. Because of its slow rates of convergence in many practical problems, the method has been replaced by many new iterative procedures. One of the most successful numerical techniques is based on the approximate factorization (AF) scheme. There are many variants of the AF method,[1-6] including those based on the alternating direction implicit (ADI) procedure developed by Ballhaus and Steger,[1] AF2 by Ballhaus et al.,[2] AF3 by Baker,[6] and strongly implicit procedure (SIP) by Sankar et al.[5] These computational procedures provide substantial improvements in the convergence rates compared to the SLOR method. However, they all require one or more iteration parameters in order to accelerate the convergence and an intermediate variable is introduced into the iterative process for a two-dimensional calculation. Consequently, the uncertainty as to what values should be used for the iteration parameters and the uncertainty about how to select the boundary conditions for the intermediate variable may affect the convergence rates as well as the stability of the iterative process. It is our aim here to present an efficient iterative procedure that yields the rapid rate of convergence of the AF scheme, while eliminating the difficulties associated with it. The present method consists of outer and inner iterations. The outer iterate is based on a Newton-like iterative process in which the Jacobian matrix is not required and a preconditioned minimal residual algorithm is applied only to seek an approximate solution of the system of linear equations arising at each inner iterate. Therefore, this method can be regarded as a Newton-like minimal residual algorithm or an inexact Newton-like (IN) iterative procedure.

The idea of the IN iterative scheme was first proposed by the author in Ref. 7. Although this paper indicated that the method can be used to compute transonic flowfields around airfoils, it was not competitive with the AF scheme implemented by Dougherty et al.[8] The computational results showed that more iterations were needed for a converged solution compared to the AF scheme and that the CPU time per iteration for the IN method was about three times that required for the AF scheme. More recently, the IN method has been modified to include a better preconditioning operator, resulting in a substantial improvement: the number of iterations is now about half of that required by the AF scheme and the CPU time per iteration is about twice that required by the AF scheme. The present paper is mainly concerned with the numerical solutions of a two-dimensional full potential equation. Particular attention is focused on the improved version of the IN iterative method. Comparisons of numerical results for lifting and nonlifting airfoil calculations between the IN and AF schemes are given and extension to three-dimensional calculations will be discussed. The problem formulation for the transonic flowfields around an airfoil is described in Sec. II and the solution of nonlinear discrete potential equations by the IN method in Sec. III; the results of computational experiments are discussed in Sec. IV and our conclusions in Sec. V.

II. Problem Formulation

For a two-dimensional problem in Cartesian coordinates, the governing partial differential equation for an inviscid isentropic fluid flow expressed in the conservation form is

$$(\rho\phi_x)_x + (\rho\phi_y)_y = 0 \tag{1}$$

where

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

Received June 2, 1983; revision received March 14, 1984.

*Department of Mathematics and Statistics, McGill University (presently with Department of Mathematics, University of Alberta, Edmonton, Canada).

Equation (1) is known as the full potential equation, where ϕ is the velocity potential, ρ the density of the fluid flow, and γ the ratio of specific heats. Equation (1) is a nonlinear equation since ρ is a function of ϕ_x and ϕ_y. The numerical solution of this equation for transonic flow is more delicate and more interesting than those for purely subsonic or purely supersonic flow because the governing equation changes its type from elliptic in subsonic regions to hyperbolic in supersonic regions and the boundary between these regions is unknown. Moreover, the equation also admits discontinuous solutions, such as shocks that may exist in the flowfields.

To handle a general flow problem with complex geometries, it is advantageous to transform Eq. (1) from the physical domain in the Cartesian coordinates into the computational domain in a rectangle.[8] The full potential equation written in the computational coordinates ξ and η is given by

$$(\rho U/J)_\xi + (\rho V/J)_\eta = 0 \tag{2}$$

where

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

$$U = A_1\phi_\xi + A_2\phi_\eta, \qquad V = A_2\phi_\xi + A_3\phi_\eta$$

$$J = \xi_x\eta_y - \xi_y\eta_x, \qquad A_1 = \xi_x^2 + \xi_y^2$$

$$A_2 = \xi_x\eta_x - \xi_y\eta_y, \qquad A_3 = \eta_x^2 + \eta_y^2$$

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

One of the difficulties in the numerical solution of transonic flow calculations is that both compression and expansion shocks are admitted by Eq. (1). However, the expansion shocks are physically meaningless. Thus, in order to eliminate the expansion shocks from the flowfields, an artificial viscosity term is introduced, via an upwind bias, into the full potential equation. In this paper, the method of artificial density[8,9] is implemented, where the fluid density is modified in such a way that

$$\rho \leftarrow (\rho - \mu\rho_\xi \Delta\xi) \tag{3}$$

where

$$\mu = \max[0,\ 1 - (1/M^2)]$$

Here $s \leftarrow (t)$ indicates that s is replaced by the expression given in t. In Eq. (3) μ is a switching function that is zero in subsonic flowfields and nonzero in supersonic flowfields, M the local Mach number, and ρ_ξ the density gradient in the upwind direction. An important advantage of using the artificial density method is that a central difference approximation can be employed to discretize the full potential equation in the entire flowfield regardless of whether it is in a subsonic or a supersonic region.

III. Solution Procedure

By the application of the finite difference method, the solution of the full potential equation (2) is transformed to the solution of a large set of nonlinear equations

$$L(\phi) = 0 \tag{4}$$

where ϕ is a vector of the velocity potential at the grid points and L the nonlinear full potential operator.

Newton-Like Algorithm

Our iterative scheme for the solution of Eq. (4) can be described as follows.

Let ϕ° be an initial guess for the velocity potential vector and compute the residual vector $r^\circ = L(\phi^\circ)$. Then for $n = 0,1,2,\ldots$, until $\|r^n\|_2 < \epsilon$,

solve

$$M_n \delta\phi^n = -r^n$$

set

$$\phi^{n+1} = \phi^n + \delta\phi^n$$

compute

$$r^{n+1} = L(\phi^{n+1}) \tag{5}$$

where n is an iteration number, $\delta\phi$ the correction vector, and M_n a matrix operator varying from iteration to iteration. It should be noted that, if $M_n = L'(\phi^n)$, which is the Jacobian matrix of $L(\phi)$, then Eqs. (5) are a Newton's iterative process for the solution of the nonlinear equation (4). Although Newton's method yields a rapid convergence rate, the method requires the initial guess ϕ° to be inside a domain of attraction, that is, one must have a good initial vector to ensure for convergence. Furthermore, even if the linearized full potential operator is a sparse matrix, the Jacobian matrix $L'(\phi)$ will likely be a full matrix. For many practical problems in the aerospace industry, the order of the nonlinear equations is large (5000 or more); consequently, it is very difficult and expensive to compute the Jacobian matrix for each iteration n.

In order to implement the iterative scheme in Eqs. (5) efficiently, it would then be natural to consider another operator for M_n. Axelsson[10] has shown that if M_n is a linear operator and in some sense makes $\|L(\phi^n) - M_n\phi^n\|$ almost insensitive to ϕ^n, then the iterative procedure of Eqs. (5) converges. In this paper, we choose M to be an approximation to the full potential operator and, with this particular choice, the iterative process defines a Newton-like algorithm. The construction for M_n is given in the following paragraphs.

Consider that, at the nth iterate, the fluid density has been calculated from values of the velocity potential at the $(n-1)$th iteration. The result of the application of a central difference approximation to L then leads to a nine-point formula, where

$$\begin{aligned}(L\phi)_{i,j} = {} & C_{i,j}\phi_{i,j} + W_{i,j}\phi_{i-1,j} + E_{i,j}\phi_{i+1,j} \\ & + N_{i,j}\phi_{i,j+1} + S_{i,j}\phi_{i,j-1} + NW_{i,j}\phi_{i-1,j+1} \\ & + NE_{i,j}\phi_{i+1,j+1} + SW_{i,j}\phi_{i-1,j-1} + SE_{i,j}\phi_{i+1,j-1}\end{aligned} \tag{6}$$

Note that, the values at the NW, NE, SE, and SW positions are usually much smaller than those at N, W, C, E, and S positions, since they are due to the skewness effect of the coordinate transformation. The operator M is now chosen by setting the values at NW, NE, SW, and SE to zero. Hence, M will be a five-point formula, which implies that the skewness effect has been ignored. We point out that neglecting the nonzero values at the NW, NE, SW, and SE positions for the full potential equation in nonconservation form no longer guarantees that M will be a good approximation to the L operator. Thus, M should retain the nine-point formula structure for this situation. The results for the nonconservative full potential equation will be reported later.

For the conservative full potential equation, it should be noted that M will be identical to the linearized operator L, provided an orthogonal transformation is used. For other transformations, M will be only an approximation of L. Thus, the operator M takes the following form for orthogonal transformations:

$$M\delta\phi_{i,j} = \left[\bar{\delta}_\xi\left(\frac{\rho A_1}{J}\right)_{i+\frac{1}{2},j}\vec{\delta}_\xi + \bar{\delta}_\eta\left(\frac{\rho A_3}{J}\right)_{i,j+\frac{1}{2}}\vec{\delta}_\eta\right]\delta\phi_{i,j} \tag{7}$$

in purely subsonic flow calculations. For mixed subsonic/supersonic flow problems, the density ρ in M has been modified according to Eq. (3) so that an artificial viscosity term is introduced. However, for large supersonic regions (i.e., strong shock calculations), it is necessary to introduce an additional upwind directional bias in the supersonic flowfields to ensure smooth convergence. This can be achieved by modifying the operator M so that a $\phi_{\xi t}$ type of term is explicitly included and M will take the form

$$M\delta\phi_{i,j} = \left[\pm\mu\beta\vec{\vec{\delta}}_{\xi} + \overleftarrow{\delta}_{\xi}\left(\frac{\rho A_1}{J}\right)_{i+\frac{1}{2},j}\vec{\delta}_{\xi} + \overleftarrow{\delta}_{\eta}\left(\frac{\rho A_3}{J}\right)_{i,j+\frac{1}{2}}\vec{\delta}_{\eta}\right]\delta\phi_{i,j} \tag{8}$$

for transonic flow calculations, where μ is a switching function that is zero in subsonic regions and nonzero in supersonic regions, β a constant controlling the amount of the $\phi_{\xi t}$ term introduced.

It should be noted that Eqs. (7) and (8) are valid only for orthogonal transformations. In this paper, a nonorthogonal grid transformation is used and the operator M is only an approximation of L. Consequently, instead of Eq. (8), we have

$$M\delta\phi_{i,j} = [\pm\mu\beta\vec{\vec{\delta}}_{\xi} + L + E]\delta\phi_{i,j} \tag{9}$$

where E is the error matrix due to ignoring the skewness effect in the grid transformation. Now considering $E = E_1 + \alpha I$, the Newton-like iterative scheme in Eq. (5) becomes

$$(\pm\mu\beta\vec{\vec{\delta}}_{\xi} + L + \alpha I + E_1)(\phi^{n+1} - \phi^n) = -L\phi^n \tag{10}$$

Since $\|E_1\| \ll \|L\|$ for the full potential equation in conservation form, it is not hard to observe that Eq. (10) in fact simulates a time-dependent problem,

$$\alpha\phi_t \pm \mu\beta\phi_{\xi t} + L\phi^{n+1} = 0 \tag{11}$$

We would also like to remark that our iterative scheme is fully implicit and that, moreover, the boundary conditions for the M operator are the same as those imposed for the full potential operator. Although the iterative scheme given in Eqs. (5) appears to be similar to that based on the AF technique, there is an important difference in the choice of the operator for M. In the AF scheme, M is taken to be a product of two simple factors [$N_1(\alpha)$ and $N_2(\alpha)$] and the basic iterative scheme can be expressed as

$$N_1(\alpha)N_2(\alpha)\delta\phi^n = \alpha\omega L\phi^n \tag{12}$$

where α is a sequence of acceleration parameters, ω a relaxation parameter, and $N_1(\alpha)$ and $N_2(\alpha)$ functions of α (which are easy to invert). The solution of Eq. (12) is then obtained in two steps through an intermediate value F, that is,

Step 1:

$$N_1(\alpha)F^n = \alpha\omega L\phi^n$$

Step 2:

$$N_2(\alpha)\delta\phi^n = F^n \tag{13}$$

This, in turn, requires and additional boundary condition for F in order to solve step 1. Slow convergence or even divergence may occur if the values of acceleration parameters α and the boundary conditions for F are not carefully chosen. Although the effect of the intermediate boundary conditions in the AF scheme has been studied via the von Neumann and Gustafsson-Kreiss-Sundstrom theory by South and Hafez,[11] the stability analysis is valid only for purely subsonic flow calculations. There is still no rigorous analysis available for mixed subsonic-supersonic problems. Consequently, the performance of the AF scheme for transonic flow calculations may strongly depend upon the experience of an individual user. That is, a fast convergence rate can be achieved if optimal values of the acceleration parameters and suitable intermediate boundary conditions are provided, but not otherwise.

Minimal Residual Algorithm

In order to obtain a better approximation ϕ^{n+1} in the Newton-like iterative scheme, we need to solve a system of linear equations

$$M\delta\phi = -r \tag{14}$$

where M is a large sparse matrix operator. It is important to have an efficient solution method, since the linear system has to be solved for each step in the Newton-like procedure. A direct method is not possible because it requires a large amount of storage and arithmetic operations. In this paper an iterative method based on a minimal residual (MR) algorithm is used. Although the MR method has a slower convergence rate than the conjugate gradient algorithm,[12] it can be applied to both symmetric and unsymmetric problems as long as all the eigenvalues of the matrix operator have positive (or negative) real parts. The number of iterations (NI) required to attain a given accuracy ϵ using the MR method is given[13] by

$$\mathrm{NI} = 0.5K\ell n(1/\epsilon) \tag{15}$$

where $K = \|M\|\|M^{-1}\|$ is the condition number of the matrix operator M. Clearly, the rate of convergence depends upon the value of K in the sense that the smaller the value for K, the faster the convergence rate. In order to accelerate the iterative process, a nonsingular matrix C is introduced and the linear system [Eq. (14)] is rewritten as

$$MC^{-1}\overline{\delta\phi} = -r \tag{16}$$

where $\delta\phi = C\delta\phi$. Equation (16) is known as the preconditioned system and C is the preconditioning operator. Suppose C is chosen so that C^{-1} is a good approximation of M^{-1}, then the condition number of MC^{-1} would be much smaller than that for M itself. Consequently, solving the preconditioned system [Eq. (16)] should yield a faster convergence rate than that of the originial system [Eq. (14)]. A detailed account of the construction for C and its relationship to the matrix M will be given below. The preconditioned MR algorithm can now be described as follows.

Let $\delta\phi^\circ$ be an initial guess correction vector; compute the residual vector, $p^\circ = -r^\circ - M\delta\phi^\circ$; and solve $Cz^\circ = p^\circ$. Then for $k = 0,1,2,\ldots k$, compute

$$\delta\phi^{k+1} = \delta\phi^k + \alpha_k z^k \quad \text{and} \quad p^{k+1} = p^k - \alpha_k Mz^k$$

and solve

$$Cz^{k+1} = p^{k+1} \tag{17}$$

where $\alpha_k = (p^k, Mz^k)/(Mz^k, Mz^k)$. Here (x,y) denotes the usual inner product, i.e., $(x,y) = x^T y$. The main computational work per iteration in the preconditioned MR algorithm is one matrix-vector multiplication for Mz and the solution for $Cz = p$.

Since we are interested in the overall convergence for the nonlinear problem [Eq. (2)], it may not be necessary to solve the linear system in Eq. (14) to excessively high accuracy for each Newton-like iteration. In our implementation, only an

approximate solution is sought for each iteration. This can be achieved by using a small fixed number of iterations (i.e., $\bar{k}=4$) in the preconditioned MR algorithm. The iterative procedure described in this section thus consists of outer and inner iterations: the outer is based on a Newton-like algorithm [Eq. (5)] and the preconditioned MR method [Eqs. (17)] is applied to find an approximate solution for the inner iteration. Therefore, this method is regarded as a Newton-like minimal residual algorithm or simply an inexact Newton-like (IN) procedure.

Preconditioning Matrix

Having presented the preconditioned MR algorithm [Eqs. (17)], we here construct the preconditioning matrix C. Remember that the extra computational work for each iteration in the preconditioned algorithm is to solve the linear system $Cz=p$. Note that, if $C=I$ (the identity matrix), then Eqs. (17) become the regular MR algorithm. For a good preconditioned algorithm, C should be chosen so that the condition number of MC^{-1} is much smaller than that of M itself. In particular, if $C=M$ the condition number of MC^{-1} is 1, the minimal value. Although a converged solution can be achieved in one iteration, we need to solve $Mz=p$, which is as difficult as solving the original problem $M\delta\phi=-r$. Thus, another important consideration is that C^{-1} should be easily invertible; otherwise, the preconditioned algorithm will not be efficient. To satisfy these two criteria, C is taken to be an approximation to the matrix operator M. C is also a product of the sparse triangular matrices

$$C=LU=M+E \tag{18}$$

where L and U are sparse lower and upper triangular matrices and E the error matrix that measures how good the approximation between C and M is. The matrix C is based on an incomplete factorization technique.[14,15] The algorithm for constructing the sparse matrices L and U can be described as follows. Recall that M is a sparse matrix consisting of five nonzero diagonals, where

$$(M\phi)_{i,j}=E_{i,j}\phi_{i,j}+D_{i,j}\phi_{i-1,j}+F_{i,j}\phi_{i+1,j}+H_{i,j}\phi_{i,j+1}+B_{i,j}\phi_{i,j-1} \tag{19}$$

Now L and U are constructed so that they have four nonzero diagonals respectively, in which the three nonzero diagonals are in the same positions to those in the lower and upper triangular part of M, where

$$(L\phi)_{i,j}=v_{i,j}\phi_{i,j}+t_{i,j}\phi_{i-1,j}+g_{i,j}\phi_{i,j-1}+x_{i,j}\phi_{i+1,j-1}$$

$$(U\phi)_{i,j}=\phi_{i,j}+e_{i,j}\phi_{i+1,j}+f_{i,j}\phi_{i,j+1}+y_{i,j}\phi_{i-1,j+1} \tag{20}$$

The elements of L and U are computed from the coefficients of M according to the relations

$$g_{i,j}=B_{i,j}$$

$$x_{i,j}=-g_{i,j}e_{i,j-1}$$

$$t_{i,j}=D_{i,j}-g_{i,j}y_{i,j-1}$$

$$v_{i,j}=(1+\alpha)E_{i,j}-g_{i,j}f_{i,j-1}-t_{i,j}(e_{i-1,j}+y_{i-1,j})-x_{i,j}(y_{i+1,j-1}+e_{i+1,j-1})$$

$$e_{i,j}=(F_{i,j}-f_{i+1,j-1}x_{i,j})/v_{i,j}$$

$$y_{i,j}=-t_{i,j}f_{i-1,j}/v_{i,j}$$

$$f_{i,j}=H_{i,j}/v_{i,j} \tag{21}$$

A small value α is added to the main diagonal elements of M to ensure the stability of the iterative scheme. However, the convergence rate is not sensitive to α in the range of 0.025-0.1 and $\alpha=0.05$ is used for all test problems in this paper. It should be pointed out that the algorithm given in Eqs. (21) satisfies the following condition[15]:

Nonzero elements (except the main diagonal) of M

= nonzero elements of C in the same locations as M

Consequently, although the preconditioning matrix C is factorized to LU decomposition, the product of LU will be symmetric as long as M is symmetric. Thus, LU may be a symmetric matrix for purely subsonic flows or during the early iterations in mixed subsonic-supersonic calculations. However, when supersonic regions are developed, a $\phi_{\xi t}$ term will appear in the M matrix operator and LU will then become asymmetric.

The solution of $Cz=p$ can now be obtained efficiently by the following simple procedure. Since $C=LU$, the linear system $Cz=p$ can be rewritten as

$$Ls=p, \qquad Uz=s \tag{22}$$

where s is a dummy vector. The solution of $Ls=p$ is obtained through a forward substitution, where

$$s_{i,j}=(p_{i,j}-g_{i,j}s_{i,j-1}-t_{i,j}s_{i-1,j}-x_{i,j}s_{i+1,j-1})/v_{i,j} \tag{23}$$

Note that, unlike the AF scheme, no boundary condition is required in order to solve for the dummy vector s. The solution of $Uz=s$ is obtained by a backward substitution, where

$$z_{i,j}=s_{i,j}-e_{i,j}z_{i+1,j}-f_{i,j}z_{i,j+1}-y_{i,j}z_{i-1,j+1} \tag{24}$$

To end this section we would like to mention that one of the differences in the implementation of the present method compared to that given in Ref. 7 is that a more accurate preconditioning matrix is used here. The computational results given in Ref. 7 were based on a different LU decomposition, in which $x_{i,j}=y_{i,j}=0$ for all i,j in Eqs. (20), i.e., L and U had only three nonzero diagonals. With an additional diagonal for both L and U, the norm of the error matrix E is smaller than that in Ref. 7. Consequently, a faster convergence rate is achieved in the present iterative scheme.

IV. Numerical Results

In this section, results of numerical experiments using the AF and IN iterative schemes are presented. The computer program is based on the TAIR code[8] and they have all been carried out on the CDC CYBER 203 computer at the NASA Langley Research Center. The problems to be considered are the transonic potential flowfields around an NACA 0012 airfoil at different Mach numbers and different angles of attack. The grid system used for both schemes is the same, where the total grid is $149\times30=4470$ points. Furthermore, the boundary conditions, initial starting vectors, and criteria for the convergence test are the same for both schemes. For all figures presented in this section, the solid lines are the results based on the AF scheme of Holst[8] and the dotted lines are those for the IN iterative scheme.

The surface pressure coefficient distribution C_p are identical to those reported by Holst's experiments[8] and hence will not be presented here. In this paper, we shall focus mainly on the comparison of convergence rates and efficiency for the AF and IN iterative schemes. Figures 1-3 compare the rates of convergence of the two methods for the following cases: 1) $M_\infty=0.85$, $\alpha=0$ deg; 2) $M_\infty=0.8$, $\alpha=0.5$ deg; and 3) $M_\infty=0.75$, $\alpha=2$ deg. Case 1 is for zero angle of attack,

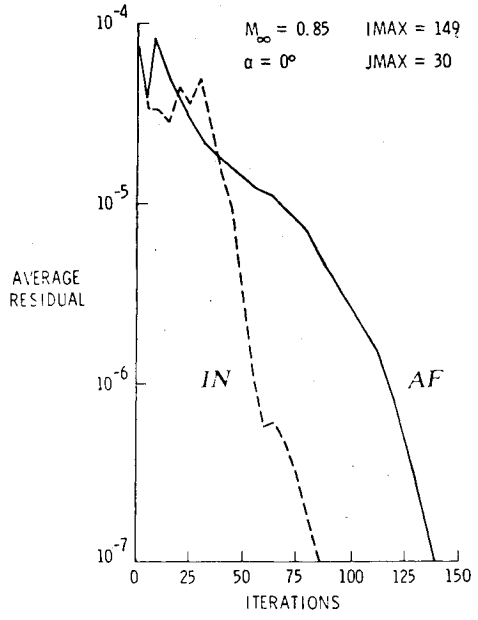

Fig. 1 Comparison of convergence histories (NACA 0012 airfoil, $M_\infty = 0.85$, $\alpha = 0$ deg).

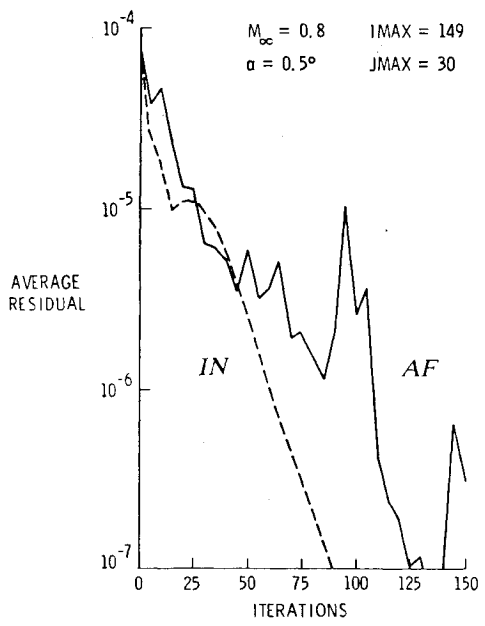

Fig. 2 Comparison of convergence histories (NACA 0012 airfoil, $M_\infty = 0.8$, $\alpha = 0.5$ deg).

i.e., a nonlifting condition, while cases 2 and 3 correspond to lifting airfoil calculations.

From the comparison of the convergence histories we observe that the IN iterative scheme generally produces a smoother reduction in the residual norm, especially for lifting airfoil calculations. Another useful criterion for comparing the efficiency of each method is to study the development of supersonic points and the circulations as the number of iterations is increased. Figures 4 and 5 show the development of the number of supersonic points for $M_\infty = 0.85$, $\alpha = 0$ deg and $M_\infty = 0.8$, $\alpha = 0.5$ deg, respectively, while Fig. 6 gives the development of circulation for $M_\infty = 0.8$, $\alpha = 0.5$ deg. The results in Figs. 4-6 clearly indicate that the number of supersonic points and the circulations are rapidly established for the IN iterative scheme. From Figs. 4 and 5 we observe that at the second iteration the supersonic point has already been established in the IN scheme and moreover, it reaches almost 50% of its final value at the fifth iteration, whereas the AF scheme attained its first supersonic points in the sixth and seventh iterations for $M_\infty = 0.85$ and $M_\infty = 0.8$, respectively.

It should be noted that the convergence histories alone do not reveal the complete picture for a comparison of the AF and IN schemes. Another important point of concern is the computing time required for each method to obtain a given accuracy. For a detailed comparison, one should study the exact numbers of arithmetic operations in each computer code. However, this is obviously very difficult to achieve for practical problems such as transonic flowfields calculations. To provide a reasonable indication, Table 1 gives the total CPU time in seconds required to reduce an average residual (i.e., $||r||_2$) to less than 10^{-7}. The CPU time per iterate is 0.235 s for the AF scheme and 0.524 s for the IN scheme.

It should be pointed out that a considerable amount of computational work is needed to evaluate the residual vector at each iteration, since it is necessary to update the fluid density at each grid point, modify the densities in the supersonic regions, and calculate the velocity potentials, etc. In fact, evaluating these residuals takes more computing time than solving the discrete potential equation using the AF scheme for each iteration. Consequently, although the work per iteration for the IN scheme takes twice the CPU time required by the AF scheme, the total number of iterations is reduced so that the overall computing time needed to attain the same accuracy for the IN scheme is not significantly larger than that based on the AF scheme.

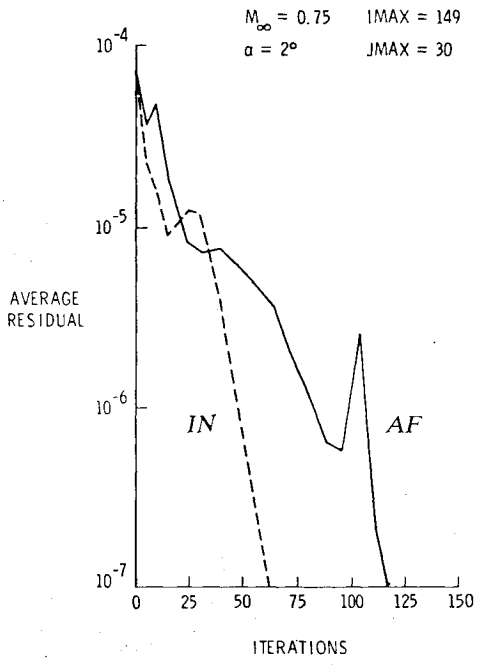

Fig. 3 Comparison of convergence histories (NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 2$ deg).

A further improvement in computing time for the IN iterative scheme is possible, since no effort to optimize the present computer code was attempted. One could expect that a larger number of inner iterations (i.e., for the value of k in the preconditioned minimal residual algorithm) per outer iteration (i.e., for the value of n in the Newton-like iterative process) might result in a smaller number of outer iterations. Similarly, a smaller number of inner iterations per outer iteration might result in a larger number of outer iterations. It is, of course, not clear what values are the best possible for a particular problem. However, a variable for the inner iteration, so that k is gradually increased as the outer iterations proceed, will probably be a better strategy than a constant value for k as used in the present implementation. A criterion for achieving this objective is being investigated.

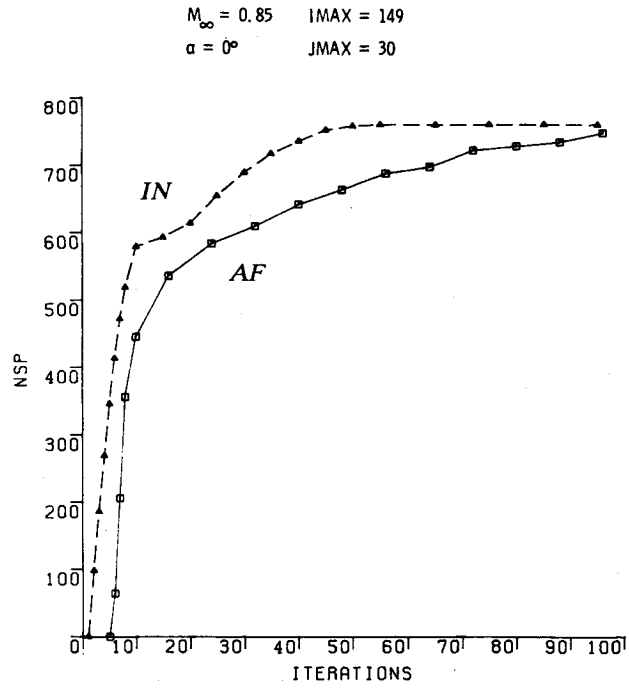

Fig. 4 Development of the number of supersonic points (NSP) as the number of iterations is increased (NACA 0012 airfoil, $M_\infty = 0.85$, $\alpha = 0$ deg).

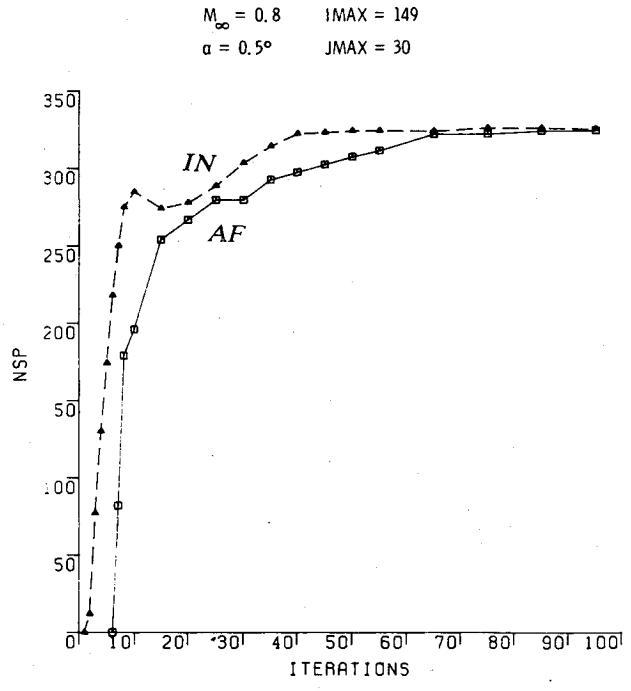

Fig. 5 Development of the number of supersonic points (NSP) as the number of iterations is increased (NACA 0012 airfoil, $M_\infty = 0.8$, $\alpha = 0.5$ deg).

Table 1 Comparison of CPU time for the AF and IN schemes, s

Scheme	$M_\infty = 0.85$, $\alpha = 0$ deg	$M_\infty = 0.8$, $\alpha = 0.5$ deg	$M_\infty = 0.75$, $\alpha = 2$ deg
AF	32.4	35.5	28.2
IN	41.8	44.5	31.3

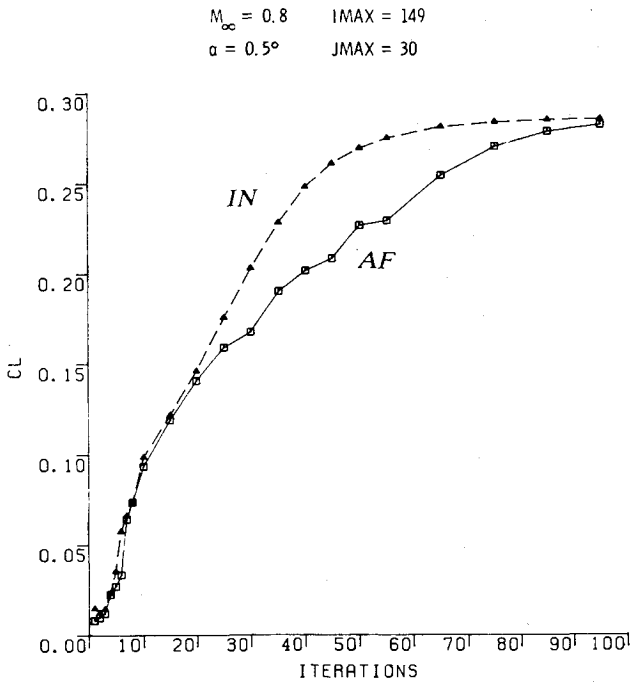

Fig. 6 Development of the circulations (*CL*) as the number of iterations is increased (NACA 0012 airfoil, $M_\infty = 0.8$, $\alpha = 0.5$ deg).

Finally, it should be mentioned that in assessing these numerical results, one should keep in mind that there is a major difference between these methods. Although the AF scheme requires less computing time, its formulation and application require the specialized knowledge and experience of an individual user. In that sense, the performance of the AF scheme may be greatly affected by the choice of the acceleration parameters and the treatment of the boundary conditions for the intermediate variable. For these reasons, many researchers have had a mixed experience witht the AF scheme, sometimes obtaining excellent results and sometimes finding them disappointing. Although more computing time is needed for the IN iterative scheme, it is easy to program and it does not suffer from these difficulties.

V. Conclusion

A Newton-like minimal residual iterative scheme is presented for the solution of the full potential equation in transonic ranges. The method described here exhibited an attractive property over the approximate factorization scheme, namely the uncertainties and difficulties in choosing the acceleration parameters and the treatment of boundary conditions for the intermediate variable are eliminated. Consequently, the present method is less problem dependent as well as less user dependent. Numerical results for transonic airfoil calculations are promising: the IN method generally produces a smoother reduction in the residual norm and the number of supersonic points and circulations are rapidly established as the number of iterations is increased. In addition, there is still room for improvement of the present method. Two potential areas of application are as follows:

1) Transonic wing calculations. It is technically straightforward to extend the present method to the numerical solution of a three-dimensional full potential equation. Moreover, the increase of the computational work over a two-dimensional flow calculation is smaller for the present method than for a corresponding increase in the approximate factorization scheme. For a three-dimensional problem, the matrix operator M in the Newton-like iterative procedure will be a seven-point formula instead of the five-point one used in a two-dimensional problem. However, the main com-

putational work for the inner iteration is comparable to that required for a two-dimensional calculation, since a sparse LU factorization can be obtained with no difficulty. On the other hand, the approximate factorization scheme consists of three-step calculations,[16] which can be expressed as

$$N_1(\alpha)N_2(\alpha)N_3(\alpha)\delta\phi^n = \alpha\omega L\phi^n \tag{25}$$

The solution of this scheme would now require two additional boundary conditions for the two intermediate variables.

2) Finite element method in fluid dynamics problems. Since the approximate factorization scheme is essentially based on alternating direction splitting methods, they will not be applicable, since in finite element formulations it is no longer possible to partition the matrix operator in terms of the usual directional derivatives. However, the present method does not suffer from this restriction, since an incomplete sparse LU factorization can still be derived for the finite element matrix.

These two areas of applications and other possibilities are under investigation.

Acknowledgments

This work was supported by the National Aeronautics and Space Administration under Contracts NASA-15810 and NASA-16394 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, Va. The author wishes to thank Dr. Terry Holst of NASA Ames Research Center for providing the TAIR code and Dr. Mohamed Hafez and the reviewers for many helpful comments leading to the improved presentation of this paper.

References

[1] Ballhaus, W. F. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equations," NASA TM X-73082, 1975.

[2] 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, 1977.

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

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

[5] Sankar, N. L., Malone, J. B., and Tassa, Y., "A Strongly Implicit Procedure for Steady Three-Dimensional Transonic Potential Flow," AIAA Paper 81-0385, Jan. 1981.

[6] Baker, T. J., "A Fast Implicit Algorithm for the Nonconservative Potential Equation," Open forum presentation at AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979.

[7] Wong, Y. S. and Hafez, M. M., "A Minimal Residual Method for Transonic Potential Flows," *Lecture Notes in Physics*, Vol. 170, Spring-Verlag, New York, 1982, pp. 526-532.

[8] Dougherty, F. C., Holst, T. L., Gundy, K. L., and Thomas, S. D., "TAIR-A Transonic Airfoil Analysis Computer Code," NASA TM 81296, 1981.

[9] Hafez, M. M., South, J. C., and Murman, E. M., "Artificial Compressibility Methods for Numerical Solution of Transonic Full Potential Equation," *AIAA Journal*, Vol. 17, 1979, pp. 838-844.

[10] Axelsson, O., "On Global Convergence of Iterative Methods," *Lecture Notes in Mathematics*, Vol. 953, Springer-Verlag, New York, 1982, pp. 1-19.

[11] South, J. C. Jr., and Hafez, M. M., "Stability Analysis of Intermediate Boundary Conditions in Approximate Factorization Schemes," AIAA Paper 83-1898, July 1983.

[12] Wong, Y. S. and Hafez, M. M., "Conjugate Gradient Methods Applied to Transonic Finite Difference and Finite Element Calculations," *AIAA Journal*, Vol. 20, 1982, pp. 1526-1533.

[13] Samarskii, A. A., *Theory of Difference Schemes*, Nauka, Moscow, 1977.

[14] Gustafsson, I., "On Modified Incomplete Factorization Methods," *Lecture Notes in Mathematics*, Vol. 968, Springer-Verlag, New York, 1980.

[15] Wong, Y. S., "Conjugate Gradient Type Methods for Unsymmetric Matrix Problems," Institute of Applied Mathematics and Statistics, University of British Columbia, Vancouver, Canada, TR 79-36, 1979.

[16] Holst, T. L. and Thomas, S. D., "Numerical Solution of Transonic Wing Flow Fields," AIAA Paper 82-105, Jan. 1982.