INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 8, 1–16 (1988)

IMPROVED RELAXATION SCHEMES FOR TRANSONIC POTENTIAL CALCULATIONS

M. HAFEZ

Department of Mechanical Engineering, University of California, Davis, CA, 95616 U.S.A.

AND

D. LOVELL

Flow Research, Inc., Kent, WA, U.S.A.

SUMMARY

A block relaxation scheme, grouped in a red–black ordering, is applied to transonic aerofoil calculations using body-fitted co-ordinates. The scheme is simple and easily vectorizable. Detailed comparisons with the approximate factorization method (AF2) are presented and it is shown that the new scheme is competitive in all cases considered. Transonic results, of engineering accuracy, on an 0-type grid of 149 × 30 points, are usually obtained within 200 iterations ($\simeq$ 40 s on a Cyber 175).

KEY WORDS Transonic flows Potential equations Relaxation methods

INTRODUCTION

Until recently, most transonic codes[1] were based on the successive over-relaxation method. Although generally slow, it is simple and reliable. The rate of convergence depends on a single parameter which can be found by numerical experiments. Attempts to accelerate the convergence by extrapolation[2,3] (power method) succeed only when the iterative matrix has dominant eigenvalues.

Fast direct solvers[4,5] were once proposed for transonic problems. A uniform grid in one direction is required and three-dimensional calculation is not as efficient. Later, ADI-type methods[6,7] were used successively, where a parameter is cyclically varied to reduce different frequency components of the error. A similar method, where the difference equations are approximately factored, is the strongly implicit procedure (SIP).[8] More operations of a recursive nature and more storage are needed in the LU decomposition and no proof of convergence exists even for idealized problems. Unlike ADI and SIP, the preconditioned conjugate gradient method (PCG)[9] does not require an estimation of any iteration parameters. Theoretically it is applicable, but only for symmetric positive definite systems. Finally, a breakthrough came with multigrid.[10,11] Because of programming complexity, it is not easy to implement and to achieve the theoretical rate of convergence.

It should be mentioned, however, that most of these methods are not easily vectorizable. On a vector machine, or if external array processors are used, the rate of computation becomes as important as the rate of convergence in determining the efficiency of the method.

In this paper, simple vectorizable relaxation methods are re-examined. Modifications to

0271–2091/88/010001–16$08·00

Received 9 April 1986
Revised 15 February 1987

accelerate the rate of convergence are studied using a model problem. Results for practical problems are compared with those obtained by other methods.

POTENTIAL EQUATION

The governing differential equation of transonic potential flows, in Cartesian co-ordinates, is given by the mass conservation law:

$$(\rho \mathbf{u})_x + (\rho \mathbf{v})_y = 0,$$

where

$$\mathbf{u} = \phi_x, \quad \mathbf{v} = \phi_y,$$

$$\rho = \left(1 - \frac{\gamma - 1}{\gamma + 1}(\Phi_x^2 + \Phi_y^2)\right)^{1/(\gamma-1)} \tag{1}$$

Here, the density ρ and the velocity components ϕ_x and ϕ_y are non-dimensionalized by the stagnation density ρ_s and the critical sound speed a^* respectively.

Following Holst,[7] equation (1) is transformed from Cartesian co-ordinates into the computational domain. The full potential equation written in terms of the transformed co-ordinates, $\xi(x, y)$ and $\eta(x, y)$, is

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

where

$$\mathbf{U} = A_1 \Phi_\xi + A_2 \Phi_\eta,$$

$$\mathbf{V} = A_2 \Phi_\xi + A_3 \Phi_\eta,$$

$$A_1 = \xi_x^2 + \xi_y^2, \quad A_2 = \xi_x \eta_x + \xi_y \eta_y, \quad A_3 = \eta_x^2 + \eta_y^2,$$

$$J = \xi_x \eta_y - \xi_y \eta_x,$$

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

$\mathbf{U}$ and $\mathbf{V}$ 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. Usually the grid is stretched in the lateral direction, but the aspect ratio in the physical domain is almost uniform. For most practical problems, a 149×28 grid is sufficient to obtain engineering accuracy solution. Holst introduced an implicit approximate factorization algorithm (AF2) (using artificial densities in supersonic regions) to solve equation (2). A fast, reliable code (TAIR) is well documented in Reference 12.

In this work, TAIR is modified by replacing the iterative algorithm with an over-relaxation scheme. The grid generation and residual calculations, including the boundary conditions and circulation updating, are not changed. The spatial finite difference approximations are given by

$$\overleftarrow{\delta}_\xi \left(\frac{\tilde{\rho}\mathbf{U}}{J}\right)_{i+1/2,j} + \overleftarrow{\delta}_\eta \left(\frac{\tilde{\rho}\mathbf{V}}{J}\right)_{i,j+1/2} = 0, \tag{3}$$

where $\overleftarrow{\delta}_\xi$ and $\overleftarrow{\delta}_\eta$ are the usual backward difference operators and the quantities $(\mathbf{U}/J)_{i+1/2,j}$ and $(\mathbf{V}/J)_{i,j+1/2}$ are computed using standard second-order-accurate finite difference formulae. The densities $\tilde{\rho}_{i+1/2,j}$ and $\tilde{\rho}_{i,j+1/2}$ are defined by

$$\tilde{\rho}_{i+1/2,j} = [(1 - \nu)\rho]_{i+1/2,j} + \nu_{i+1/2,j}\,\rho_{i+k+1/2,j},$$

$$\bar{\rho}_{i,j+1/2} = [(1 - \nu)\rho]_{i,j+1/2} + \nu_{i,j+1/2}\,\rho_{i,j+l+1/2},$$

where

$$k = \pm 1 \quad \text{when} \quad U_{i+1/2,j} \gtrless 0, \tag{4}$$

$$l = \pm 1 \quad \text{when} \quad V_{i,j+1/2} \gtrless 0,$$

and

$$v_{i+1/2,j} = \begin{cases} \max\,[(M_{i,j}^2 - 1)\,\mathrm{CON}, 0] & \text{for} \quad U_{i+1/2,j} > 0, \\ \max\,[(M_{i+1,j}^2 - 1)\,\mathrm{CON}, 0] & \text{for} \quad U_{i+1/2,j} < 0, \end{cases} \tag{5}$$

where $M_{i,j}$ is the local Mach number and CON is a user-specified constant.

The aerofoil boundary condition is simply $\mathbf{V} = \mathbf{0}$. This is implemented at the aerofoil surface using reflection. To facilitate the circulation calculation, the velocity potential is written as

$$\Phi = \phi + \Gamma \xi / \xi_{\max}, \tag{6}$$

where Γ is the jump of the potential ϕ at the trailing edge. Hence $\Phi_\xi = \phi_\xi + \Gamma$ and Φ_η are, of course, not affected. At the outer boundary, ϕ is held fixed at the initial free stream value.

SUCCESSIVE OVER-RELAXATION

For subsonic flows, equation (1) is of elliptic type. Undeniably, successive over-relaxation (SOR) is the simplest iterative method to solve elliptic equations.[13–16] For transonic flows, equation (1) is of mixed elliptic–hyberbolic type and again SOR is very suitable (the artificial time-dependent equation describing the development of iterations of line SOR, marching with the flow, is consistent with the unsteady transonic equation[1]).

For an 0-type grid such as the one used in TAIR, there are two possibilities: solving for the unknowns on a radial line or for the unknowns on a ring around the aerofoil. Most of the existing relaxation codes use radial lines, marching with the flow direction (hence a $\phi_{\xi t}$ term is implicitly introduced). On the other hand, using rings preserves the circulation and its effect will be felt instantaneously around the aerofoil; the price, however, is the extra work needed for the periodic tridiagonal solver which is a small penalty indeed. Normally, the grid is stretched in the lateral direction and thus marching the rings out is preferable to marching in. In either case, a $\phi_{\eta t}$ term is implicitly introduced. For transonic flows, a $\phi_{\xi t}$ term is needed and can be added explicitly; the $\phi_{\eta t}$ term, however, may lead to deterioration of convergence or even divergence. A simple remedy is to use red and black ordering of the rings. The asymptotic rate of convergence is the same as SOR on a uniform mesh, marching in or out, and for a stretched grid it lies in between. Such a scheme is called 'Zebra'. It is easily vectorizable; all the black rings can be solved at the same time, followed by the red ones.

The Zebra scheme was first tested on Cartesian co-ordinates for 2D problems[17] and on cylindrical co-ordinates for 3D calculations.[18] Doria and South[19] used it with a nearly orthogonal mesh generated by a sequence of Schwarz–Christoffel and shearing transformations. It was also used successfully with stream function calculations.[20]

In the following, some variants of the Zebra scheme are discussed and compared with standard relaxation methods. The SOR schemes are given by

Marching in (increasing j)

$$B_{i-1}C_{i-1,j} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)]C_{i,j} + B_i C_{i+1,j} = -L\Phi_{i,j}^n - B_{j-1}C_{i,j-1}, \tag{7}$$

Marching out (decreasing j)

$$B_{i-1}C_{i-1} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)]C_{i,j} + B_i C_{i+1,j} = -L\Phi_{i,j}^n - B_j C_{i,j+1}, \tag{8}$$

with

$$B_i = \left(\frac{\tilde{\rho}^n A_1}{J}\right)_{i+1/2,j}, \quad B_j = \left(\frac{\bar{\rho}^n A_3}{J}\right)_{i,j+1/2};$$

while the Zebra scheme is describe by

$$B_{i-1}C_{i-1,j} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)]C_{i,j} + B_i C_{i+1,j}$$
$$= -L\Phi^n_{i,j} - B_j C_{i,j+1} - B_{j-1} C_{i,j-1}; \tag{9}$$

and in all cases

$$\phi^{n+1}_{i,j} = \phi^n_{i,j} + \omega C_{i,j}.$$

It was found that it is faster to over-relax the terms corresponding to lateral derivatives only, namely

Marching in

$$B_{i-1}C_{i-1,j} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)/\omega]C_{i,j} + B_i C_{i+1,j} = -L\Phi^n_{i,j} - B_{j-1}C_{i,j-1}, \tag{7'}$$

Marching out

$$B_{i-1}C_{i-1,j} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)/\omega]C_{i,j} + B_i C_{i+1,j} = -L\Phi^n_{i,j} - B_j C_{i,j+1}, \tag{8'}$$

Zebra scheme

$$B_{i-1}C_{i-1,j} - [(B_{i-1} + B_i) + (B_{j-1} + B_j)/\omega]C_{i,j} + B_i C_{i+1,j}$$

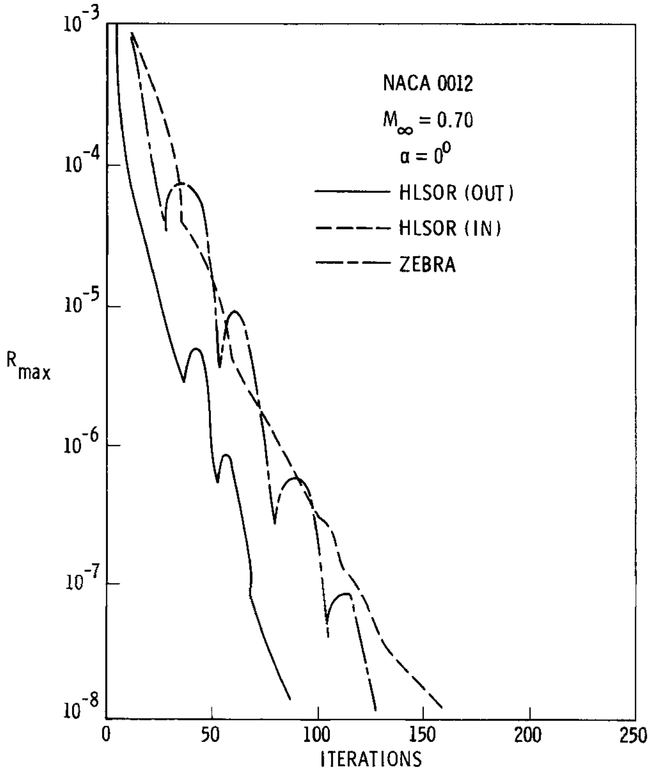

Figure 1. Convergence of SOR and Zebra schemes

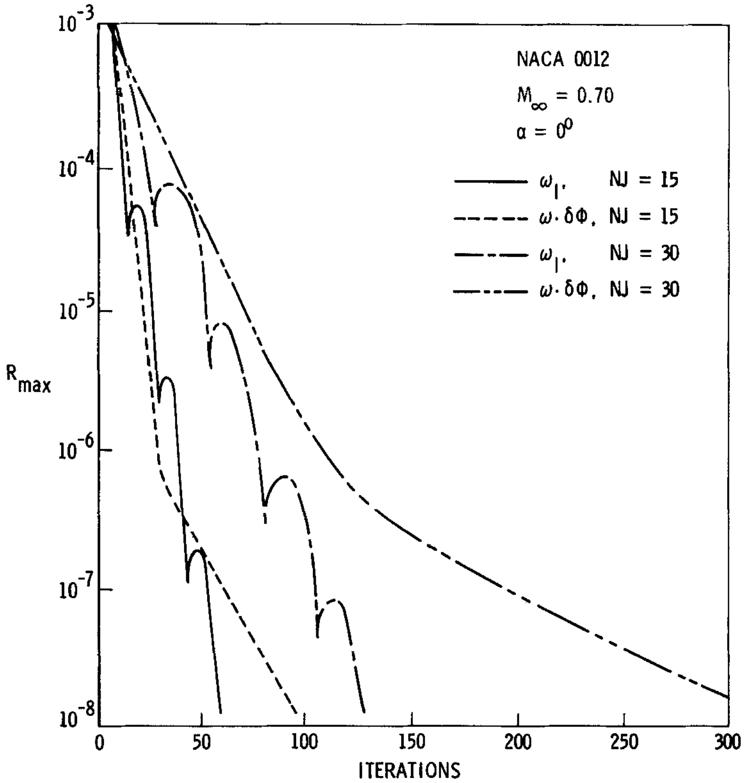

Figure 2. Convergence of Zebra schemes, relaxing certain terms only versus relaxing all terms

$$= -L\Phi^n_{i,j} - B_j C_{i,j+1} - B_{j-1} C_{i,j-1}; \tag{9'}$$

and in all cases

$$\phi^{n+1}_{i,j} = \phi^n_{i,j} + C_{i,j}.$$

At a field point, the damping term ϕ_t has a coefficient proportional to $(2/\omega - 1)$. At the ring on the aerofoil surface, equation (8′) has to be slightly modified to avoid excessive damping there. In Figure 1, the convergence histories of SOR (7′) and (8′) and of Zebra (9′) are plotted for a subsonic flow ($M_\infty = 0{\cdot}7$) around a NACA 0012 aerofoil. In Figure 2, the convergence of Zebra schemes based on equation (9) and equation (9′) for two grids, 149 × 30 and 149 × 15, are compared.

For transonic flows, a $\phi_{\xi t}$ term is added and it is implemented as follows: On the upper surface of the aerofoil, B_{i-1} is replaced by

$$\tilde{B}_{i-1} = B_{i-1}(1 - \beta), \tag{10}$$

while on the lower surface, B_i becomes

$$\tilde{B}_i = B_i(1 - \beta), \tag{10'}$$

where β is a parameter to control the coefficient of the $\phi_{\xi t}$ term. Notice that the diagonal element is always augmented by this modification.

The number of supersonic points, the shock position, as well as the circulation approach their steady values in a relatively faster way than in radial SOR schemes.

CONVERGENCE ACCELERATION: A MODEL PROBLEM

The Zebra scheme is implicit in the ξ direction but explicit in the η direction. For most practical problems the number of rings in the η direction is relatively small and the performance of Zebra is satisfactory. To study the characteristics of the scheme, a model problem, Poisson's equation on a square of uniform mesh, is considered. The spectral radius of a general class of iterative methods is given by

$$\rho = 1 - \alpha h^{\sigma}, \tag{11}$$

where h is the mesh size. For example, Jacobi and Gauss–Seidel methods have $\sigma = 2$. Successive over-relaxation, using the optimum relaxation parameter, has $\sigma = 1$. Preconditioning techniques lead to $\sigma < 1$, while multi grid convergence is independent of $h (\sigma = 0)$.

In general, SOR cannot compete with the more powerful methods (smaller σ), since a finer grid can always be chosen, where the convergence of SOR is painfully slow. But if the grid is fixed, based on the accuracy requirement, the competition will depend on the parameter α.

It is well known that block SOR is faster than point SOR. For example, line SOR is $\sqrt{2}$ faster than point SOR. It will be shown that multi-line SOR is $\sqrt{m}$ faster, where m is the number of lines in each block.

Consider a line Jacobi method. The iterative algorithm is given by

$$C_{xx} - \frac{2C}{h^2} = -L\phi^n + f, \tag{12}$$

$$(\phi_{i-1,j}^{n+1} - 4\phi_{i,j}^{n+1} + \phi_{i+1,j}^{n+1}) + \phi_{i,j+1}^{n+1} = h^2 f_{i,j} - \phi_{i,j-1}^{n}, \tag{13}$$

$$\phi_{i,j}^{n+1} + (\phi_{i-1,j+1}^{n+1} - 4\phi_{i,j+1}^{n+1} + \phi_{i+1,j+1}^{n+1}) = h^2 f_{i,j+1} - \phi_{i,j+2}^{n}. \tag{13'}$$

The system of equations (13), (13′) is solved simultaneously for the unknowns on the two lines in terms of the previous values of ϕ at the surrounding lines. The solution can be written in the form

$$\frac{C_{i-1,j} - 2C_{i,j} + C_{i+1,j}}{h^2} - \frac{2C_{i,j}}{h^2} + \frac{C_{i,j+1}}{h^2} = f_{i,j} - L\phi_{i,j}^{n}, \tag{14}$$

$$\frac{(C_{i-1,j+1} - 2C_{i,j+1} + C_{i+1,j+1}}{h^2} - \frac{2C_{i,j+1}}{h^2} + \frac{C_{i,j}}{h^2} = f_{i,j+1} - L\phi_{i,j+1}^{n}. \tag{14'}$$

Either equation (14) or (14′) is identical to that obtained by a line Gauss–Seidel iteration, which is known to be twice as fast as line Jacobi iteration.

The three-line Jacobi iteration is given by

$$\begin{aligned}
&(\phi_{i-1,j-1}^{n+1} - 4\phi_{i,j-1}^{n+1} + \phi_{i+1,j-1}^{n+1}) + \phi_{i,j}^{n+1} = h^2 f_{i,j-1} - \phi_{i,j-2}^{n},\\
&\phi_{i,j-1}^{n+1} + (\phi_{i-1,j}^{n+1} - 4\phi_{i,j}^{n+1} + \phi_{i+1,j}^{n+1}) + \phi_{i,j+1}^{n+1} = h^2 f_{i,j},\\
&\phi_{i,j}^{n+1} + (\phi_{i-1,j+1}^{n+1} - 4\phi_{i,j+1}^{n+1} + \phi_{i+1,j+1}^{n+1}) = h^2 f_{i,j+1} - \phi_{i,j+2}^{n}.
\end{aligned} \tag{15}$$

The unknowns on the three lines are solved simultaneously. To see the behaviour of the convergence, the three equations are added and written in terms of the correction:

$$\frac{\tilde{C}_{i-1,j} - 2\tilde{C}_{i,j} + \tilde{C}_{i+1,j}}{h^2} - \frac{4}{3}\frac{\tilde{C}_{i,j}}{h^2} + \frac{1}{3}\frac{C_{i,j-1}}{h^2} + \frac{1}{3}\frac{C_{i,j+1}}{h^2} \simeq \tilde{f}_{i,j} - L\tilde{\phi}_{i,j}^{n}, \tag{16}$$

where

$$\tilde{C}_{i,j} = \tfrac{1}{3}(C_{i,j-1} + C_{i,j} + C_{i,j+1}).$$

Using Garabedian analysis,[21] equations (16), (14) and (14′) are approximations of artificial time-dependent equations describing the different processes. Hence the three-line Jacobi iteration is $\frac{3}{2}$ faster than the two-line method. In general, the m-line Jacobi iteration is m times faster than the one-line method. Algebraic proofs are available in the literature.[22,23]

For the model problem, the spectural radius of the multi-line Jacobi method is

$$\rho_J \simeq 1 - m\alpha h^2. \tag{17}$$

Hence the Gauss–Seidel spectral radius is

$$\rho_{GS} = \rho_J^2 \simeq 1 - 2m\alpha h^2. \tag{18}$$

The optimum over-relaxation parameter can be calculated in terms of ρ_J as

$$\omega_{opt} = \frac{2}{1 + \sqrt{(1 - \rho_J^2)}} \tag{19}$$

and the spectral radius of SOR using ω_{opt} is

$$\rho_{SOR,\omega_{opt}} = \omega_{opt} - 1 \simeq 1 - \sqrt{m\tilde{\alpha}}h. \tag{20}$$

Finally, the Zebra scheme has the same asymptotic rate of convergence as SOR.

In the following, solution procedures for multi-line Zebra schemes are described.

SOLUTION PROCEDURES OF MULTI-LINE SCHEMES

In general, a block relaxation scheme is efficient provided that the extra work involved in the algebraic manipulations of the blocks is not large. In line relaxation methods, a tridiagonal solver can be used and the computational effort per point is almost the same for line as for point relaxation. Periodic tridiagonal solvers require almost twice as much work.[24,25] For two-line schemes, a pentadiagonal solvers is needed. This is clear from Figure 3, where the unknowns of the two lines are ordered in a zigzag manner, since a point is directly related, at most, to $s \pm 2$ points. An algorithm for a pentadiagonal system of equations[26] is given in Appendix I. Varga[27] suggested

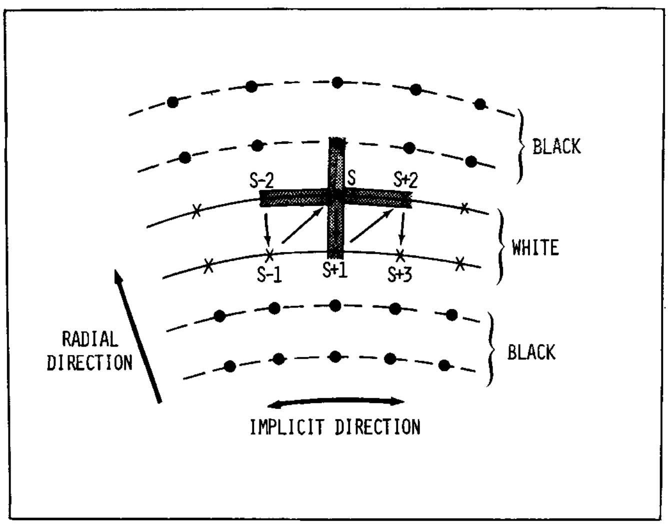

Figure 3. Zebra scheme with blocks of two lines

a normalized LU decomposition (with unit diagonal entries). The operations per mesh point are about 20% more than the corresponding tridiagonal procedure. For periodic pentadiagonal systems, the work is three times as much, as shown in Appendix II. Therefore, unless the pentadiagonal solver is optimized, the improvement in the rate of convergence of the two-line schemes will be offset by the extra algebra involved in the solvers.

Since the solvers are used iteratively, an approximation of the two-line equations may be useful, as long as it does not affect the rate of convergence. This idea is implemented as follows: the coefficients B_i for line j and line $j+1$ are averaged; hence the Zebra two-line scheme becomes

$$\hat{B}_{i-1}C_{i-1,j}-[(\hat{B}_{i-1}+\hat{B}_i)+(B_j+B_j)/\omega]C_{i,j}+\hat{B}_iC_{i+1,j}+\frac{B_j}{\omega}C_{i,j+1}$$

$$=-L\phi^n_{i,j}-B_jC_{i,j-1}, \tag{21}$$

$$\hat{B}_{i-1}C_{i-1,j+1}-[(\hat{B}_{i-1}+\hat{B}_i)+(B_j+B_j)/\omega]C_{i,j+1}+\hat{B}_iC_{i+1,j+1}+\frac{B_j}{\omega}C_{i,j}$$

$$=-L\phi^n_{i,j+1}-B_jC_{i,j+2}, \tag{21'}$$

where

$$\hat{B}_i=\tfrac{1}{2}\left[\left(\frac{\tilde{\rho}^nA_1}{J}\right)_{i+1/2,j}+\left(\frac{\tilde{\rho}^nA_1}{J}\right)_{i+1/2,j+1}\right].$$

(For simplicity, in all our calculations, the density is scaled out of the equations for the corrections.)

Adding and subtracting equations (21) and (21′), we have

$$\hat{B}_{i-1}\hat{C}_{i-1}-[(\hat{B}_{i-1}+\hat{B}_i)+B_j/\omega]\hat{C}_i+\hat{B}_i\hat{C}_{i+1}=-L\hat{\phi}_i-\tfrac{1}{2}(B_jC_{i,j-1}+B_jC_{i,j+2}), \tag{22}$$

$$\hat{B}_{i-1}\check{C}_{i-1}-[(\hat{B}_{i-1}+\hat{B}_i)+(B_j+2B_j)/\omega]\check{C}_i+\hat{B}_i\check{C}_{i+1}$$

$$=-L\check{\phi}_i-\tfrac{1}{2}(B_jC_{i,j-1}-B_jC_{i,j+2}), \tag{22'}$$

where

$$\hat{C}_i=\tfrac{1}{2}(C_{i,j}+C_{i,j+1}),\quad \check{C}_i=\tfrac{1}{2}(C_{i,j}-C_{i,j+1}).$$

Hence

$$C_{i,j}=\hat{C}_i+\check{C}_i, \tag{23}$$

$$C_{i,j+1}=\hat{C}_i-\check{C}_i. \tag{23'}$$

We found that, at least for subsonic flows, the rate of convergence of the Zebra scheme based on equations (22) and (22′) is almost the same as that obtained by the pentadiagonal solver, where no approximations of the coefficients were made. Notice that only tridiagonal solvers are used in equations (22) and (22′).

Using Garabedian analysis, the coefficients of the damping term ϕ_t for both equations (22) and (22′) are proportional to $(2/\omega-1)$.

A triple-line Zebra scheme

Black and white blocks of three lines are also considered. To solve for the unknowns on three lines simultaneously, a seven-diagonal solver is needed. Instead, the following approximate equations are used:

$$TC_{i,j-1}+(A/\omega)C_{i,j}=-L\phi_{i,j-1}-AC_{i,j-2}, \tag{24a}$$

$$(A/\omega)C_{i,j-1}+TC_{i,j}+(A/\omega)C_{i,j+1}=-L\phi_{i,j}, \tag{24b}$$

$$(A/\omega)C_{i,j}+TC_{i,j+1}=-L\phi_{i,j+1}-AC_{i,j+2}, \tag{24c}$$

where

$$A=\tfrac{1}{2}(B_{j-1}+B_j),\quad T=-2A/\omega+\overleftarrow{\partial}_\xi B_i \overrightarrow{\partial}_\xi.$$

Equations (24a), (24b) and (24c) are solved as follows:

$$T\check{C}_i=-\tfrac{1}{2}(L\phi_{i,j-1}+AC_{i,j-2}-L\phi_{i,j+1}-AC_{i,j+2}),$$

$$[T^2-2(A/\omega)^2]\hat{C}_i=\tfrac{1}{2}T(L\phi_{i,j-1}+AC_{i,j-2}+L\phi_{i,j+1}+AC_{i,j+2})+(A/\omega)L\phi_{i,j},$$

$$C_{i,j}=[\tfrac{1}{2}(L\phi_{i,j-1}+AC_{i,j-2}+L\phi_{i,j+1}+AC_{i,j+2})-T\hat{C}_i]/(A/\omega),$$

where

$$\check{C}_i=\tfrac{1}{2}(C_{i,j-1}-C_{i,j+1}),\quad \hat{C}_i=\tfrac{1}{2}(C_{i,j-1}+C_{i,j+1}).$$

Notice that

Table I. CPU time (on Cyber 175) for 100 iterations of different schemes

Method	CPU time (s)
Zebra 3 lines	18
Zebra 2 lines	20
TAIR (AF2)	20
Zebra 2 lines (penta)	26

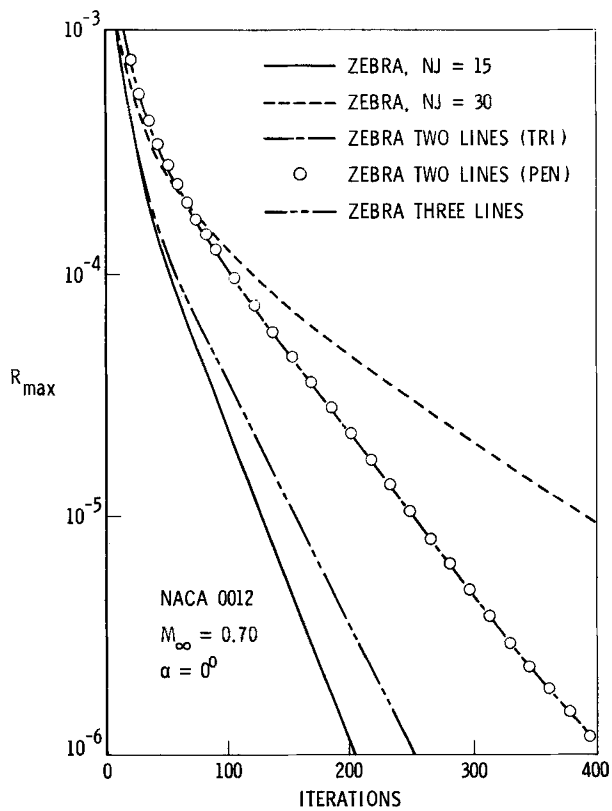

Figure 4. Convergence of one-line, two-line, and three-line Zebra schemes ($\omega=1$)

$$T^2 - 2(A/\omega)^2 \simeq (T - \sqrt{2}A/\omega)(T + \sqrt{2}A/\omega);$$

hence a tridiagonal solver is called three times. Because of the approximations, the CPU time is slightly less than used by the one-line Zebra scheme (see Table I). Again using Garabedian analysis, the coefficients of the ϕ_t term for equations (24a), (24b) and (24c) are all proportional to $(2/\omega - 1)$.

In Figure 4, convergence rates of a line Zebra, a two-line Zebra and a three-line Zebra, using $\omega = 1$ (Gauss–Seidel) in all cases, are shown. The rate of convergence of the line Zebra is $\simeq 0{\cdot}992$. According to equation (18), the two-line Zebra rate should be 0·984 and the three-line scheme rate should be 0·976. The fourth curve in this figure represents the rate of convergence for a 149×15 grid using a line Zebra scheme. The expected rate of convergence is 0·968. The numerical results confirm these predictions. It seems that the approximations do not affect the convergence for subsonic cases. The present method can be viewed as a local inversion (similar, for example, to odd and even reduction). However, the approximations based on the assumption of uniform coefficients will eventually affect the convergence rate if more and more lines are grouped together. In particular, for transonic flows with shocks, such approximations will have adverse effects, but multi-line schemes can be used at least in the subsonic far field.

NUMERICAL RESULTS

Subsonic and transonic flows around a NACA 0012 aerofoil at different Mach numbers and angles of attack are calculated. Results are presented in Figure 5. Similar results for a Korn aerofoil are shown in Figure 6. Except for Jameson's work,[11] it seems that existing applications of multigrid (Reference 29–31) are not much more efficient than the present method. Zebra is also competetive

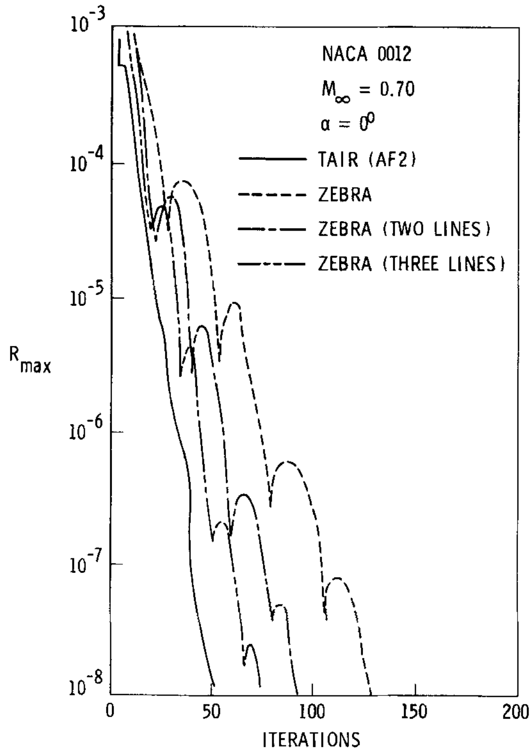

Figure 5(a). Convergence of Zebra schemes compared with TAIR results for a NACA 0012 aerofoil

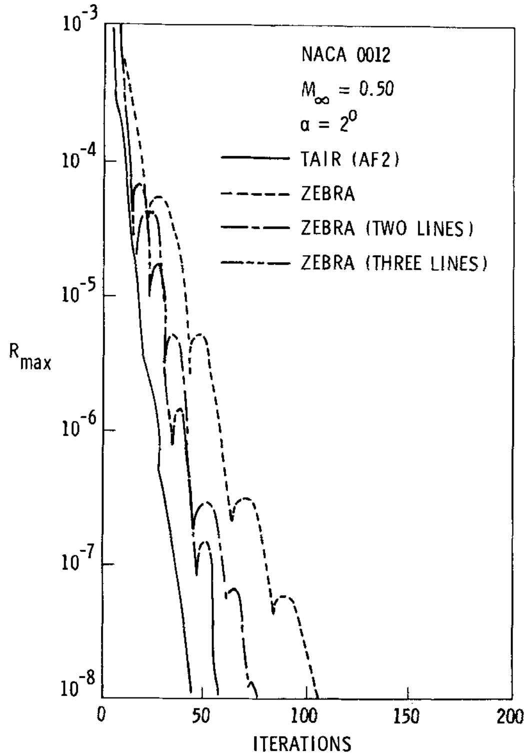

Figure 5(b). Convergence of Zebra schemes compared with TAIR results for a NACA 0012 aerofoil

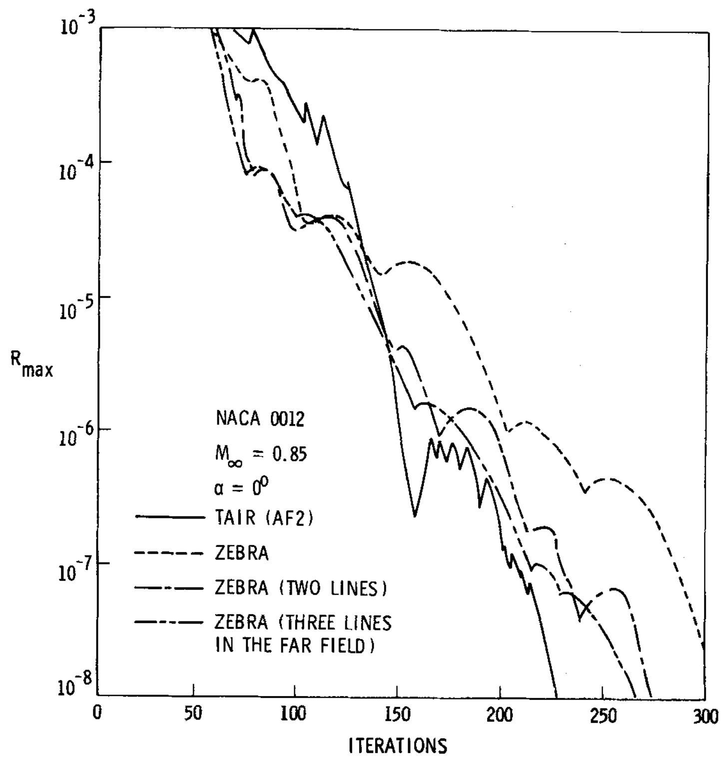

Figure 5(c). Convergence of Zebra schemes compared with TAIR results for a NACA 0012 aerofoil

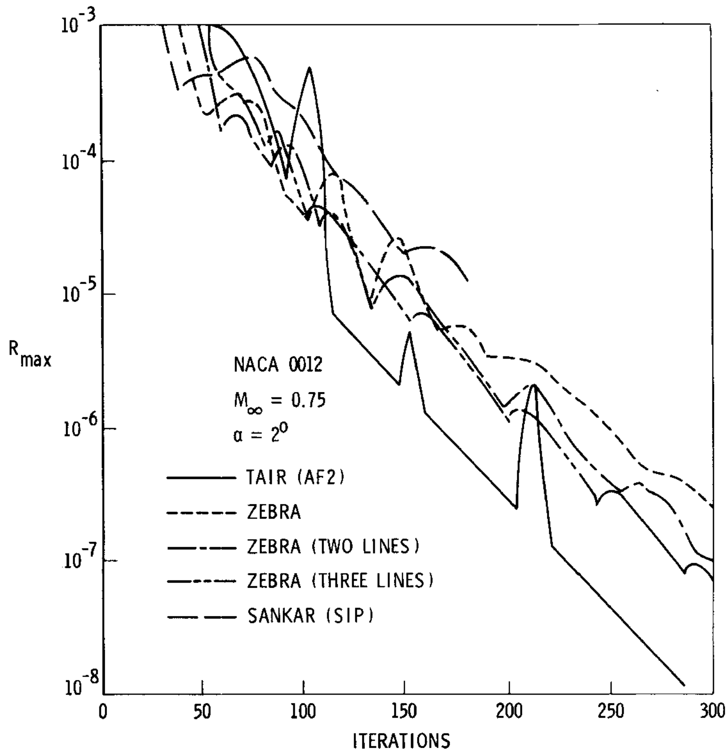

Figure 5(d). Convergence of Zebra schemes compared with TAIR results for a NACA 0012 aerofoil

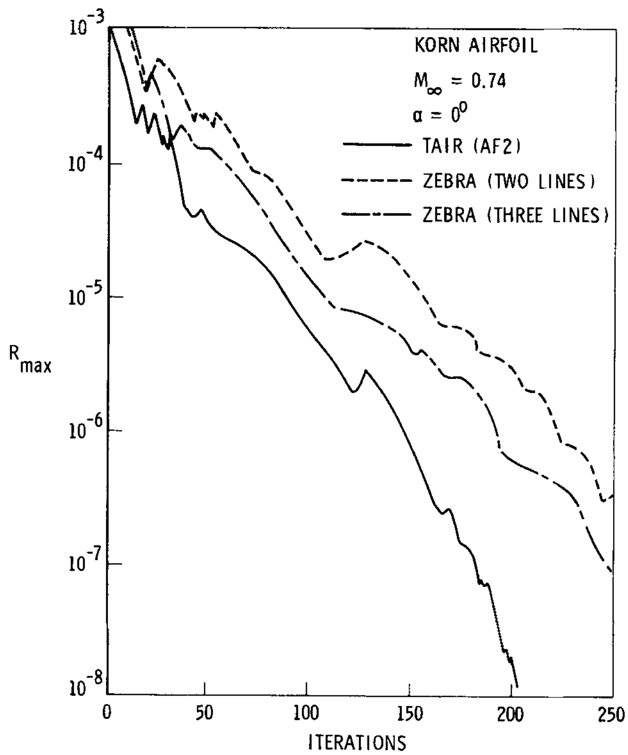

Figure 6(a). Convergence of Zebra schemes TAIR compared with results for a Korn aerofoil

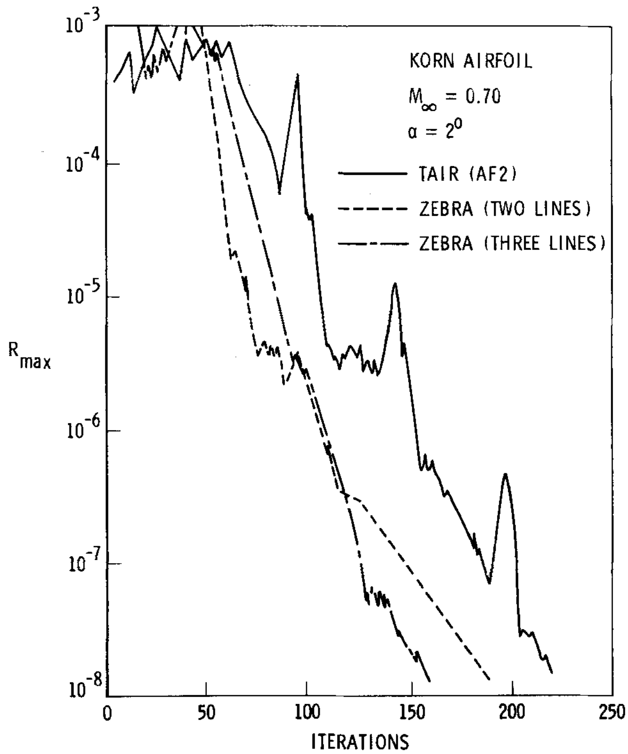

Figure 6(b). Convergence of Zebra schemes compared with TAIR results for a Korn aerofoil

Table II. Dependence of Zebra two-line scheme on relaxation parameter for NACA 0012 ($R_{max} = 10^{-7}$)

$M_\infty = 0.5, \alpha = 2°$		$M_\infty = 0.63, \alpha = 2°$		$M_\infty = 0.7, \alpha = 0°$	
W	*N*	*W*	*N*	*W*	*N*
1·82	62	1·82	86	1·82	94
1·84	60	1·84	76	1·84	84
1·86	64	1·86	70	1·86	82
1·88	70	1·88	76	1·88	82
$M_\infty = 0.8, \alpha = 0°$		$M_\infty = 0.85, \alpha = 0°$		$M_\infty = 0.75, \alpha = 2°$	
W	*N*	*W*	*N*	*W*	*N*
1·87	148	—	—	—	—
1·88	120	—	—	—	—
1·89	128	—	—	—	—
1·90	132	1·90	288	—	—
1·92	148	1·92	260	1·92	348
—	—	1·94	232	1·94	324
—	—	—	—	1·95	314
—	—	—	—	1·96	296
—	—	—	—	1·97	320

with the minimal residual method.[28] Conjugate gradient–Zebra combined iterations have been proved to be useful.[9]

The dependence of the Zebra two-line scheme on the relaxation parameter is shown in Table II, where N is the number of iterations required to reach $R_{max} = 10^{-7}$. The results shown are for NACA 0012. TAIR is less sensitive to the acceleration parameter, which is cyclically varied between α_l and α_H. The rate of convergence is not really very sensitive to small variations (20%) of these parameters. All reported results are obtained, therefore, using default values, with no attempt at optimization. It is obvious that the performance of TAIR is very satisfactory. There are, however, some questions about the approximate factorization involved, and it is not clear how the error terms affect the convergence of transonic calculations. Another point of concern is the treatment of the boundary conditions for the intermediate variable used with the approximate factorization process, which is handled in the code in a rather empirical way.

CONCLUDING REMARKS

Within the limitations of the relaxation methods, a modest improvement can be achieved using blocks of two lines. The two-line scheme can be implemented with almost the same computational rate as the one-line scheme. For an 0-type grid around an aerofoil, the unknowns of two rings are solved simultaneously. It is found that black and red ordering of the blocks is suitable for transonic calculations and leads to easily vectorizable codes. Blocks of three lines are also considered.

To obtain transonic results of engineering accuracy, the present simple method is competitive with existing techniques in terms of efficiency. Extension to three-dimensional calculation is straightforward.

ACKNOWLEDGEMENTS

The authors would like to thank J. South of NASA Langley for many helpful discussions. They are also thankful to T. Holst of NASA Ames for providing his AF2 code (TAIR) for detailed comparisons, as well as to N. L. Snakar of Lockheed-Georgia for his SIP calculations.

APPENDIX I. SOLUTION OF A PENTADIAGONAL SYSTEM OF EQUATIONS

Consider

$$a_k\phi_{k-2} + b_k\phi_{k-1} + c_k\phi_k + d_k\phi_{k+1} + e_k\phi_{k+2} = f_k, \quad k = 1,2,\ldots,N-1, \tag{25}$$

$$a_1 = b_1 = a_2 = e_{N-1} = e_{N-2} = d_{N-1} = 0.$$

The solution of (25) can be written in the form

$$\phi_k = \alpha_k\phi_{k+1} + \beta_k\phi_{k+2} + \alpha_k. \tag{26}$$

The coefficients α_k, β_k and γ_k are then computed according to the formulae

$$\alpha_k = -\frac{d_k + \beta_{k-1}(a_k\alpha_{k-2} + b_k)}{c_k + \alpha_{k-1}(a_k\alpha_{k-2} + b_k) + a_k\beta_{k-2}},$$

$$\beta_k = -\frac{e_k}{c_k + \alpha_{k-1}(a_k\alpha_{k-2} + b_k) + a_k\beta_{k-2}},$$

$$\gamma_k = \frac{f_k - \gamma_{k-1}(a_k\alpha_{k-2} + b_k) - a_k\gamma_{k-2}}{c_k + \alpha_{k-1}(a_k\alpha_{k-2} + b_k) + a_k\beta_{k-2}}. \tag{27}$$

APPENDIX II. SOLUTION OF A PERIODIC PENTADIAGONAL SYSTEM OF EQUATIONS

Consider

$$\mathbf{A\Phi} = \mathbf{F},$$

$$\begin{bmatrix} c_1 & d_1 & e_1 & & & a_1 b_1 \\ b_2 & c_2 & d_2 & e_2 & & a_2 \\ & a_k & b_k & c_k & d_k & e_k \\ e_{N-2} & & & a_{N-2} b_{N-2} c_{N-2} d_{N-2} \\ d_{N-1} k_{N-1} & & & & a_{N-1} b_{N-1} c_{N-1} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_k \\ \phi_{N-2} \\ \phi_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_k \\ f_{N-2} \\ f_{N-1} \end{bmatrix}. \tag{28}$$

If we let $\bar{\mathbf{A}}$ be the pentadiagonal matrix obtained by deleting the last two rows and two columns of $\mathbf{A}$, and define

$$\bar{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ 0 \\ 0 \\ 0 \\ \phi_{N-3} \end{bmatrix}, \quad \bar{\mathbf{F}} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \\ 0 \\ 0 \\ f_{N-3} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} a_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ d_{N-3} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} b_1 \\ a_2 \\ 0 \\ 0 \\ e_{N-4} \\ e_{N-3} \end{bmatrix}, \tag{29}$$

then the system (28) can be rewritten as

$$\bar{\mathbf{A}}\bar{\mathbf{\Phi}} = \bar{\mathbf{F}} - \phi_{N-2}\mathbf{G} - \phi_{N-1}\mathbf{H} \tag{30}$$

or

$$\bar{\mathbf{\Phi}} = \bar{\mathbf{\Phi}}_1 - \phi_{N-2}\bar{\mathbf{\Phi}}_2 - \phi_{N-1}\bar{\mathbf{\Phi}}_3, \tag{31}$$

where $\bar{\mathbf{\Phi}}_1$, $\bar{\mathbf{\Phi}}_2$ and $\bar{\mathbf{\Phi}}_3$ are obtained by solving the following non-periodic pentadiagonal systems:

$$\bar{\mathbf{A}}\bar{\mathbf{\Phi}}_1 = \bar{\mathbf{F}}, \quad \bar{\mathbf{A}}\bar{\mathbf{\Phi}}_2 = \mathbf{G}, \quad \bar{\mathbf{A}}\bar{\mathbf{\Phi}}_3 = \mathbf{H}. \tag{32}$$

It remains to solve the two scalar equations for the unknowns ϕ_{N-2} and ϕ_{N-1}, namely

$$e_{N-2}\phi_1 + a_{N-2}\phi_{N-4} + b_{N-2}\phi_{N-3} + C_{N-2}\phi_{N-2} + d_{N-2}\phi_{N-1} = f_{N-2},$$

$$d_{N-1}\phi_1 + e_{N-1}\phi_2 + a_{N-1}\phi_{N-3} + b_{N-1}\phi_{N-2} + C_{N-1}\phi_{N-1} = f_{N-1}. \tag{33}$$

Substituting (32) in (33), we obtain

$$a_{11}\phi_{N-2} + a_{12}\phi_{N-1} = b_1, \quad a_{21}\phi_{N-2} + a_{22}\phi_{N-1} = b_2$$

or

$$\phi_{N-2} = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11}a_{22} - a_{21}a_{12}}, \quad \phi_{N-1} = \frac{a_{11}b_2 - a_{21} - b_1}{a_{11}a_{22} - a_{21} - a_{12}}, \tag{34}$$

where

$$b_1 = f_{N-2} - e_{N-2}\bar{\Phi}_1(1) - b_{N-2}\bar{\Phi}_1(N-4) - C_{N-2}\bar{\Phi}_1(N-3),$$

$$b_2 = f_{N-1} - d_{N-1}\bar{\Phi}_1(1) - e_{N-1}\bar{\Phi}(2) - b_{N-1}\bar{\Phi}_1(N-3),$$

$$a_{11} = -e_{N-2}\bar{\Phi}_2(1) - b_{N-2}\bar{\Phi}_2(N-4) - C_{N-2}\bar{\Phi}_2(N-3) + d_{N-2},$$

$$a_{12} = -e_{N-2}\bar{\Phi}_3(1) - b_{N-2}\bar{\Phi}_3(N-4) - C_{N-2}\bar{\Phi}_3(N-3) + e_{N-2},$$

$$a_{21} = -d_{N-1}\bar{\Phi}_2(1) - e_{N-1}\bar{\Phi}_2(2) - b_{N-1}\bar{\Phi}_2(N-3) + C_{N-1},$$

$$a_{22} = -d_{N-1}\bar{\Phi}_3(1) - e_{N-1}\bar{\Phi}_3(2) - b_{N-1}\bar{\Phi}_3(N-3) + d_{N-1}.$$

Finally, $\bar{\Phi}$ is obtained from (31).

REFERENCES

1. A. Jameson, 'Transonic flow calculations', *VKI Lecture Series, Vol. 87*, March 1976.
2. M. Hafez and H. K. Cheng, 'Convergence acceleration of relaxation solution for transonic flow computations', *AIAA J.*, **15**, 329–336 (1977).
3. D. A. Caughey and A. Jameson, 'Accelerated iterative calculation of transonic nacelle flow fields', *AIAA J.*, **15**, 1474–1480 (1977).
4. E. D. Martin and H. Lomax, 'Rapid finite-difference computation of subsonic and slightly supercritical aerodynamic flow', *AIAA J.*, **13**, 579–586 (1975).
5. A. Jameson, 'Transonic potential flow calculations using conservation form', *AIAA Computational Fluid Dynamics Conf.*, Hartford, CT, June 1975.
6. W. F. Ballhaus, A. Jameson and J. Albert, 'Implicit approximate factorization schemes for the efficient solution of steady transonic flow problems', *AIAA J.*, **16**, 573–579 (1978).
7. T. L. Holst, 'Implicit algorithm for the conservative transonic full-potential equation using an arbitrary mesh', *AIAA J.*, **17**, 1038–1045 (1979).
8. N. L. Sankar, J. B. Malone and Y. Tassa, 'A strongly implicit procedure for steady three-dimensional transonic potential flows', *AIAA Paper 81-0385*, January 1981.
9. Y. S. Wong and M. Hafez, 'Application of conjugate gradient methods to transonic finite difference and finite element calculations', *AIAA Paper 81-1032*, June 1981.
10. J. C. South and A. Brandt, 'Application of a multilevel grid method to transonic flow calculations', in T. C. Adamson and M. C. Platzer (eds), *Transonic Flow Problems in Turbomachinery*, Hemisphere, 1977.
11. A. Jameson, 'Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method', *AIAA Computational Fluid Dynamics Conf.*, Williamsburg, VA, July 1979.
12. F. C. Dougherty, T. L. Holst, K. L. Gundy and S. D. Thomas, 'TAIR—A transonic aerofoil analysis computer code', *NASA Technical Memorandum 81296*, May 1981.
13. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
14. E. L. Wachspress, *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
15. D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.
16. L. A. Hageman and D. A. Young, *Applied Iterative Methods*, Academic Press, New York, 1981.
17. M. Hafez and J. C. South, 'Vectorization of relaxation methods for solving transonic full potential equation', in *Numerical Methods for the Computation of Inviscid Transonic Flow with Shock Waves; GAMM Workshop*, Sweden, 1979.
18. J. C. South, J. D. Keller and M. Hafez, 'Vector processor algorithms for transonic flow calculations', *AIAA J.*, **18**, 786–792 (1980).
19. M. L. Doria and J. C. South, 'Transonic potential flow and coordinate generation for bodies in a wind tunnel', *AIAA Paper 82-0223*, January 1982.
20. M. Hafez and D. Lovell, 'Numerical solution of transonic stream function equation', *AIAA Paper 81-1017*, June 1981.
21. P. Garabedian, 'Estimation of the relaxation factor for small mesh size', *Math. Tables Aids Comput.*, **10**, 183–185 (1956).
22. S. V. Parter, 'On two-line iterative methods for the Laplace and biharmonic difference equations', *Numerische Math.*, **1**, 240–252 (1959).
23. S. V. Parter, 'Multi-line iterative methods for elliptic difference equations and fundamental frequencies', *Numerische Math.*, **3**, 305–319 (1961).
24. F. C. Thames, Private communication.
25. H. H. Ahlberg, E. N. Nilson and J. L. Walsh, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.
26. S. D. Conte and R. T. Dames, 'An alternating direction method for solving the biharmonic equation', *Math. Comput.*, **12**, 198–205 (1958).
27. R. S. Varga, in R. E. Langer (ed.), *Factorization and Normalized Iteration Methods in Boundary Problems in Differential Equations*, The University of Wisconsin Press, 1960.
28. Y. S. Wong and M. Hafez, 'A minimal residual method for transonic potential flows', *8th Int. Conf. on Numerical Methods in Fluid Dynamics*, Germany, July 1982.
29. H. Deconinck and C. Hirsch, 'A multigrid method for the transonic full potential equation discretized with finite elements on an arbitrary body fitted mesh', in *Multigrid Methods; NASA Conference Publication 2202*, October 1981.
30. J. W. Boerstoel, 'A multigrid algorithm for steady transonic potential flows around airfoils using Newton iteration', in *Multigrid Methods, NASA Conference Publication 2202*, October 1981.
31. N. L. Sankar, 'A multigrid strongly implicit procedure for two-dimensional transonic potential problems', *AIAA Paper 82-0931*, June 1982.