COO-3077-82

Courant Institute of

Mathematical Sciences

ERDA Mathematics and Computing Laboratory

Accelerated Iteration Schemes for Transonic Flow Calculations Using Fast Poisson Solvers

Antony Jameson

ERDA Research and Development Report

Mathematics and Computers

March 1975

New York University

ERDA Mathematics and Computing Laboratory
Courant Institute of Mathematical Sciences
New York University

Mathematics and Computers

COO-3077-82

ACCELERATED ITERATION SCHEMES FOR TRANSONIC FLOW CALCULATIONS USING FAST POISSON SOLVERS

Antony Jameson

Contract No. ~~E(11-1)-3077~~

Table of Contents

1. Formulation of a Fast Iterative Method for Transonic Flow Calculations

Reliable but slow methods for calculating transonic flows have been developed in recent years [1,2,3]. These use central difference formulas in the subsonic zone and upwind difference formulas in the supersonic zone to ensure the proper region of dependence and jump conditions. The resulting difference equations are then solved by an iteration procedure derived from the method of successive overrelaxation. This note describes results obtained by using a fast Poisson solver to accelerate the rate of convergence of the iterative scheme.

Note: This work was also partially supported by NASA Grants NGR-33016-167 and -201.

NGR-33-016-201

It was proposed by Martin and Lomax [4] that a fast elliptic solver could be used to generate an iterative scheme for solving the difference equations appearing in compressible flow calculations. In the simplest case consider the small disturbance equation

$$(1-M^2)\phi_{xx} + \phi_{yy} = 0$$

where ϕ is the velocity potential, and M is the local Mach number, which is related to the free stream Mach number M_∞ by the formula

$$M^2 = M_\infty^2(1 + (\gamma+1)\phi_x) \ .$$

An iterative scheme can be constructed by putting the Laplacian on the left and the nonlinear terms on the right. Let v_n be the solution for ϕ at the nth iteration. Then

$$\Delta v_{n+1} = M^2 \frac{\partial^2}{\partial x^2} v_n \ .$$

To see that this can be expected to converge consider the linearized equation where M^2 is replaced by M_∞^2. If P and Q are nonnegative finite difference operators representing $-\frac{\partial^2}{\partial x^2}$ and $-\frac{\partial^2}{\partial y^2}$ we have

$$(P+Q)v_{n+1} = M^2 P v_n$$

or

$$P^{1/2} v_{n+1} = M^2 \ K \ P^{1/2} \ v_n$$

where

$$K = P^{1/2}(P+Q)^{-1} P^{1/2} \ .$$

Thus

$$\| P^{1/2} v_{n+1} \| \leq M^2 \| K \| \| P^{1/2} v_n \|$$

and since K is Hermitian

$$\| K \| = \lambda_{max}(K)$$

$$= \max \frac{(x, Kx)}{(x, x)}$$

$$= \max \frac{(y, Py)}{(y, Py) + (y, Qy)}$$

where

$$P^{1/2} y = K^{1/2} x .$$

Thus

$$\| P^{1/2} v_{n+1} \| \leq M^2 \| P^{1/2} v_n \| .$$

This estimate serves to indicate that for subsonic flows the scheme should converge at a rate independent of the mesh size.

The above analysis also suggests that it is doubtful whether such a scheme would converge for a supersonic flow with $M > 1$. The argument presented in the Appendix in fact indicates that the scheme would definitely not converge for a linearized supersonic flow. It thus appears that the fast elliptic solver used on its own is not likely to lead to good conver ence when the supersonic zone is large. If, however, it could be supplemented with another scheme which give fast convergence in the supersonic zone, the two in combination might produce an

effective iterative scheme. In fact the standard line relaxation method for transonic flow calculations is such a scheme. For the small disturbance equation, or in the case of the full potential equation with the flow aligned with one coordinate direction, the method consists of freezing the nonlinear coefficients at values determined from the previous iteration and solving the resulting wave equation in the supersonic zone by a marching procedure. Thus an exact solution of the supersonic zone could be obtained in one step if the correct coefficients and data at the sonic line could be inserted.

Thus the following scheme is proposed: use a two stage iteration, in which the first stage is a step using the Laplacian on the left-hand side, and the second stage consists of a fixed number of relaxation steps to stabilize the supersonic zone. For the case of flow aligned with the coordinate system a single relaxation step should be sufficient. The full potential equation in a curvilinear coordinate system with an arbitrary flow direction requires the use of a rotated upwind difference scheme [3]. Then a simple marching scheme can no longer be used in the supersonic zone, and several relaxation steps may be required in the second stage.

2. Application to the Transonic Potential Flow Equation in a Mapped Domain

The use of a fast Poisson solver requires a simple domain such as a rectangle. This leads to a difficulty in applying the proposed method to an exterior flow problem with an infinite domain. This can be circumvented by using the full potential equation and mapping the exterior of the profile onto the interior of a circle. If $2\pi E$ is the circulation it is convenient to use a reduced potential G defined by

$$\phi = G + \frac{\cos\theta}{r} - E\,\theta .$$

Then G is finite and single valued. Now a fast solver for Poisson's equation in polar coordinates can be used in the first stage of the iteration. For this purpose a scheme using the Buneman algorithm in the θ direction has been programmed.

Two variants of this approach have been tried. The first treats the potential equation in quasilinear form. The residual at each point is evaluated as

$$R = (a^2-v^2)\, r \frac{\partial}{\partial r}(r\, G_r) + (a^2-u^2)\, G_{\theta\theta} - 2uv\, r(G_{r\theta}+G_\theta - E)$$
$$+ (u^2-v^2) r\, G_r + (u^2+v^2)\left(\frac{u}{r} H_\theta + v\, H_r\right) ,$$

where H is the modulus of the transformation to the exterior of the circle, u and v are the velocity components

$$u = \frac{r(G_\theta - E) - \sin\theta}{H} , \qquad v = \frac{r^2 G_r - \cos\theta}{H}$$

and a is the speed of sound. If γ is the ratio of

specific heats, a is determined from the stagnation speed of sound a_0 by the relation

$$a^2 = a_0^2 - \frac{\gamma-1}{2} (u^2+v^2) .$$

In evaluating R upwind differencing is used in the usual manner at supersonic points. In the first stage of the iteration the correction C is determined by solving

$$r \frac{\partial}{\partial r} (r C_r) + C_{\theta\theta} = \frac{R}{a^2}$$

and then G is updated by the rule

$$G^+ = G + \omega C$$

where the superscript $+$ denotes the new value, and ω is an overrelaxation factor. In the second stage of the iteration an ordinary relaxation step is used.

Results with this approach have been quite promising. Numerical tests have confirmed that the scheme sometimes diverges when the relaxation step is not included. When it is included fast convergence has been obtained. Figures 1 and 2 show typical results. In each calculation the calculation was performed first on a mesh with 64 cells in the θ direction and 16 cells in the r direction, and then on a mesh with 128 $\times$ 32 cells. The interpolated coarse mesh solution was used as the starting point for the fine mesh calculation. The largest absolute value of the residual anywhere in the field was used as a measure of convergence.

The first example is the 64A410 airfoil at Mach .720. In this case the residual was reduced from $\sim 10^{-1}$ to 10^{-9} in 26 cycles on the coarse mesh, and then from $\sim 10^{-3}$ to 10^{-9} in 21 cycles on the fine mesh, each cycle consisting of one Poisson step plus one relaxation step. The Poisson step takes about the same time as 2 relaxation steps, so each complete cycle requires about the same time as 3 relaxation steps. On the CDC 6600 at the ERDA Computing Facility at New York University one complete cycle on the fine mesh takes about 1.5 seconds. The entire calculation for the 64A410 took 48 seconds. The second example shows a shock-free supercritical airfoil designed by Garabedian [6]. In this case 27 cycles were required to reduce the largest residual to 10^{-9} on the coarse mesh, and another 27 cycles to reduce it to 10^{-9} on the fine mesh. In corresponding calculations using relaxation steps without the Poisson steps the largest residual was still $\sim 10^{-6}$ after 1000 cycles on the fine mesh.

The second variant treats the potential equation in conservation form using a rotated difference scheme in the supersonic zone [5]. In this case the residual is evaluated as

$$R = r \frac{\partial}{\partial x} (r\rho V) + \frac{\partial}{\partial \theta} (\rho U)$$

where

$$U = G_\theta - E - \frac{\sin \theta}{r}, \quad V = r G_r - \frac{\cos \theta}{r},$$

and ρ is the density. If M_∞ is the free stream Mach

number ρ is determined from the speed of sound a by the relation

$$\rho^{\gamma-1} = M_\infty^2 \, a^2 \ .$$

Now in the first stage of the iteration the correction C is determined by solving

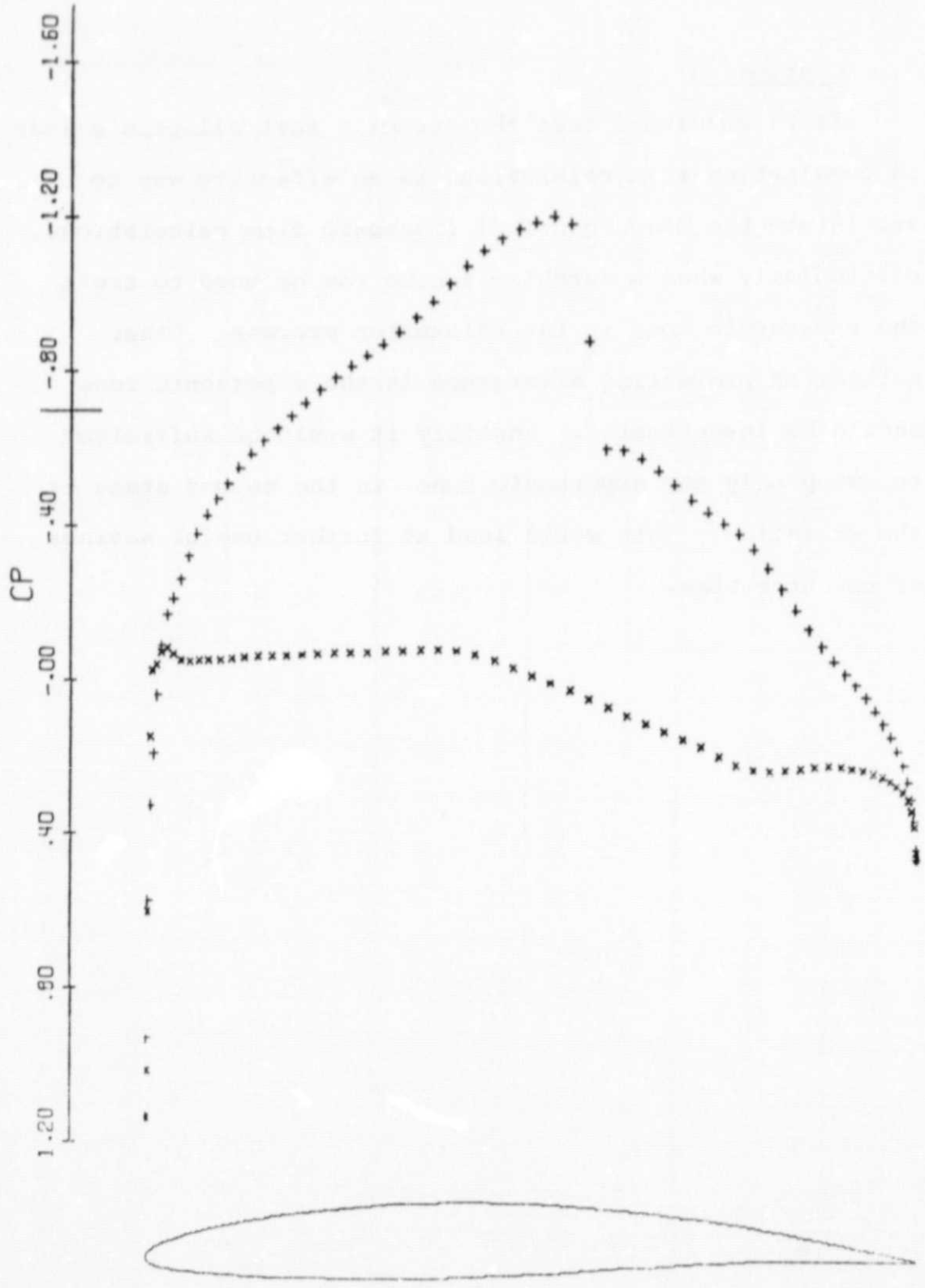
$$r \frac{\partial}{\partial r} (r \, C_r) + C_{\theta\theta} = \frac{R}{\rho} \ .$$

The second stage consists of k relaxation cycles. Typically k = ..

In this case also the combined iteration has proved to give faster convergence than the simple relaxation method. The improvement is not as great as with the first variant, however, because of the need to use more relaxation steps than Poisson steps. As an example of the application of the method, Figure 3 shows the pressure distribution for the 64A410 at Mach .720 recalculated using conservation form. The proper theoretical jump condition is now satisfied, as can be seen. In this calculation the number of cycles required to reduce the largest residual to 10^{-9} was 37 on the coarse mesh and 40 on the fine mesh. Each cycle consisted of 1 Poisson step plus 5 relaxation steps, and took about the same amount of time as 7 relaxation steps, so the fine mesh calculation is equivalent to a little under 300 relaxation steps, which would be enough to reduce the largest residual to $\sim 10^{-5}$ using relaxation alone.

3. Conclusion

It is concluded that the use of a fast elliptic solver in combination with relaxation is an effective way to accelerate the convergence of transonic flow calculations, particularly when a marching scheme can be used to treat the supersonic zone in the relaxation process. Other methods of preventing divergence in the supersonic zone should be investigated. Possibly it would be sufficient to sweep only the supersonic zone in the second stage of the iteration. This would lead to further useful savings of computer time.

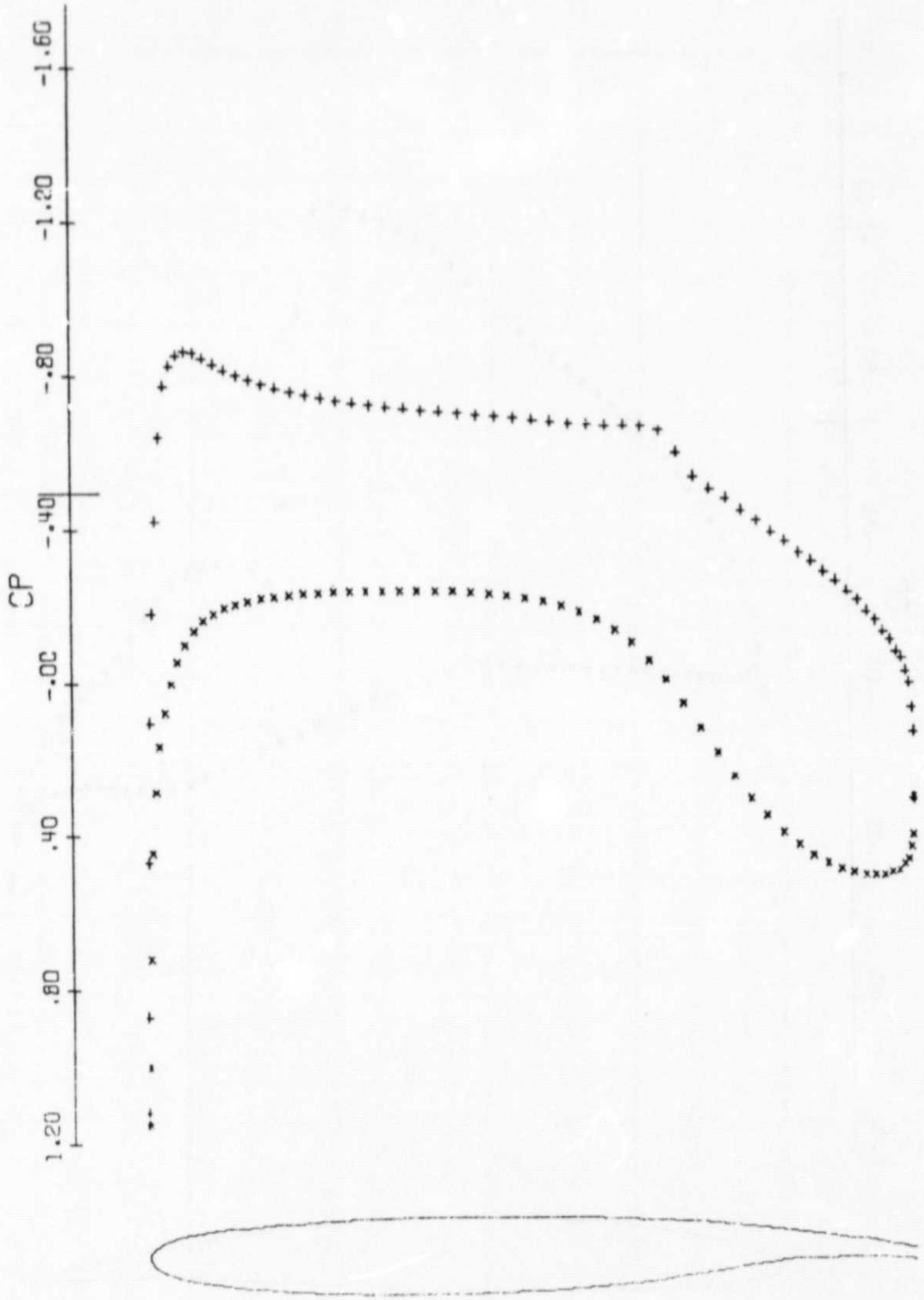


NACA 64A410 AIRFOIL

MACH NO. .720 ANGLE OF ATTACK 0°

LIFT COEFFICIENT .6289 DRAG COEFFICIENT .0040

Figure 1

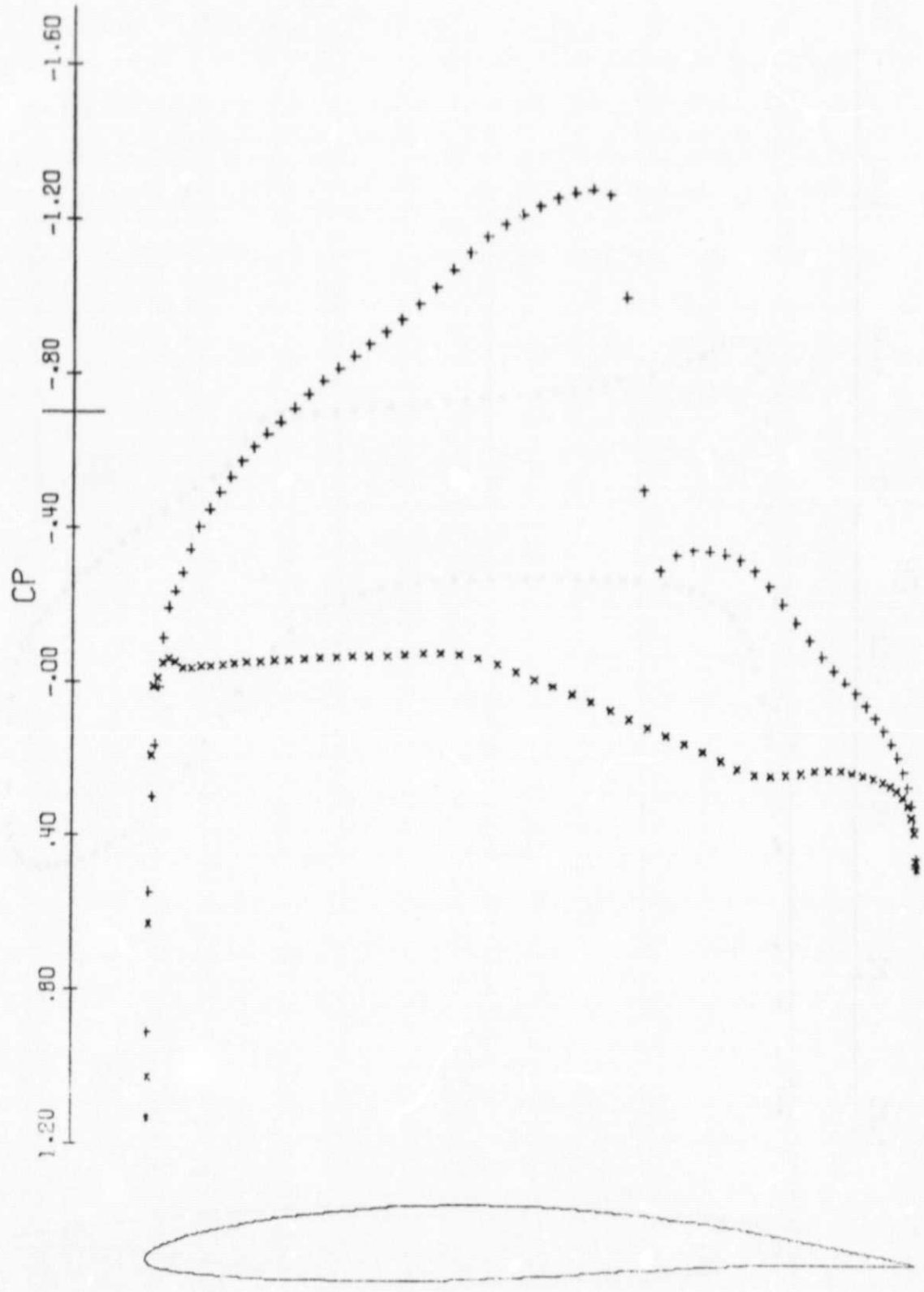


78-06 AIRFOIL

MACH NO. .780 ANGLE OF ATTACK 0°

LIFT COEFFICIENT .5867 DRAG COEFFICIENT .0006

Figure 2



NACA 64A410 AIRFOIL

MACH NO.	.720	ANGLE OF ATTACK	0°
LIFT LIFT COEFFICIENT	.6654	DRAG COEFFICIENT	.0029

Figure 3

References

1. Murman, E. M., and Cole, J. D., Calculation of plane steady transonic flows, AIAA Journal, Vol. 10, 1972, pp. 171-176.
2. Garabedian, P. R., and Korn, D., Analysis of transonic airfoils, Comm. Pure Appl. Math., Vol. 24, 1972, pp. 841-851.
3. Jameson, Antony, Iterative solution of transonic flows over airfoils and wings, Comm. Pure Appl. Math., Vol. 27, 1974, pp. 283-309
4. Martin, E. Dale, and Lomax, Harvard, Rapid finite difference computation of subsonic and transonic aerodynamic flows, AIAA Paper, Jan. 1974.
5. Jameson, Antony, Transonic potential flow calculations using conservation form, to be presented at AIAA Computational Fluid Dynamics Conference, June 1975.
6. Bauer, F., Garabedian, P., Korn, D., and Jameson, A., Handbook of Supercritical Wing Sections, Springer Verlag, 1975.

Appendix. Analysis of the Poisson Iteration for the Linearized Equation with $M > 1$.

Let the equation

$$(1-M^2)\phi_{xx} + \phi_{yy} = 0 ,$$

with M^2 a constant > 1, be approximated with equal mesh spacing in the x and y directions by the Murman difference scheme

$$(1-M^2)(\phi_{ij}-2\phi_{i-1,j}+\phi_{i-2,j}) + \phi_{i,j+1} - 2\phi_{ij} + \phi_{i,j-1} = 0$$

in which an upwind difference formula is used for ϕ_{xx}. Denoting updated values by the superscript +, consider the iteration

$$\phi^+_{i+1,j} - 2\phi^+_{ij} + \phi^+_{i-1,j} + \phi^+_{i,j+1} - 2\phi^+_{ij} + \phi^+_{i,j-1}$$

$$= \phi_{i+1,j} - 2\phi_{ij} + \phi_{i-1,j} + (M^2-1)(\phi_{ij}-2\phi_{i-1,j}+\phi_{i-2,j}).$$

Let i and j both run from 1 to n and define the n × n matrix

$$T = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & & \ddots \end{bmatrix}$$

Also define the n × n matrix

$$R = \begin{bmatrix} 1 & & & & \\ -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & & \cdots & \end{bmatrix}$$

and let Φ be the matrix with entries ϕ_{ij}. Then the iteration can be written as

$$T\,\Phi^{+} + \Phi^{+}T = (T + \alpha R)\,\Phi$$

where

$$\alpha = M^2 - 1 > 0 \ .$$

Also it is easily verified that

$$R = T\,S$$

where

$$S = \begin{bmatrix} 0 & & & & -1/(n+1) \\ 1 & 0 & & & -2/(n+1) \\ & 1 & 0 & & -3/(n+1) \\ & & 1 & 0 & -4/(n+1) \\ & & & \cdots & \cdots \end{bmatrix}$$

Thus

$$T\,\Phi^{+} + \Phi^{+}T = T(I + \alpha S)\,\Phi$$

If

$$\Phi^{+} = \lambda\,\Phi$$

then Φ (with its elements suitably ordered to form a vector) is an eigenvector of the iteration matrix. Consider the form

$$\Phi = uv^{T}$$

where v is an eigenvector of T

$$T\ v = \mu\ v$$

Then Φ is an eigenvector if

$$T(I + \alpha S)uv^T = \lambda(Tuv^T + uv^T T)$$
$$= \lambda(T + \mu I)\ uv^T$$

This is satisfied if

$$P_\mu u = \lambda\ u$$

where

$$P_\mu = (T+\mu I)^{-1}\ T(I+\alpha S)\ .$$

Thus the eigenvectors of the iteration matrix can be expressed as

$$u\ v^T$$

where v is an eigenvector of T with eigenvalue μ and u is an eigenvector of P_μ, and the eigenvalues of the iteration matrix are the eigenvalues of P_μ for $\mu = \mu_1, \mu_2, \ldots, \mu_n$. For large n the smallest eigenvalue μ of T is of order $\frac{1}{n^2}$. But as $\mu \to 0$ the eigenvalues of P_μ approach those of

$$P_0 = I + \alpha S$$

and correspondingly some of the eigenvalues of the iteration matrix approach

$$1 + \alpha\ \lambda_i$$

where λ_i are the eigenvalues of S. Now

$$\det(\lambda I - S) = \lambda^n + \frac{n}{n+1}\lambda^{n-1} \ldots + \frac{1}{n+1}$$

Thus if the polynomial

$$J_n = (n+1)\lambda^n + n\lambda^{n-1} \ldots + 1$$

has a root in the right half plane the corresponding eigenvalues of P_0 will lie outside the unit circle. Applying the Routh Hurwitz test, it can be verified that J_n has at least one root in the right half plane when $n \geq 4$.

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Administration, nor any person acting on behalf of the Administration:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Administration" includes any employee or contractor of the Administration, or employee of such contractor, to the extent that such employee or contractor of the Administration, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Administration, or his employment with such contractor.