78-12

Preliminary Study of the Use of the STAR-100 Computer for Transonic Flow Calculations

J.D. Keller, *NASA Langley Research Center, Hampton, Va.; and* A. Jameson, *Courant Institute of Mathematical Sciences, New York University, New York, N.Y.*

AIAA 16TH AEROSPACE SCIENCES MEETING

Huntsville, Alabama/January 16-18, 1978

PRELIMINARY STUDY OF THE USE OF THE STAR-100 COMPUTER FOR TRANSONIC FLOW CALCULATIONS

James D. Keller*
NASA Langley Research Center
Hampton, Virginia

and

Antony Jameson**
New York University
New York, New York

Abstract

A new explicit method for solving the transonic small-disturbance potential equation is presented. This algorithm, which is suitable for the new vector-processor computers such as the CDC STAR-100, is compared to successive line over-relaxation (SLOR) on a simple test problem. The convergence rate of the explicit scheme is slower than that of SLOR. However, the efficiency of the explicit scheme on the STAR-100 computer is sufficient to overcome the slower convergence rate and allow an overall speedup compared to SLOR on the CYBER 175 computer.

Introduction

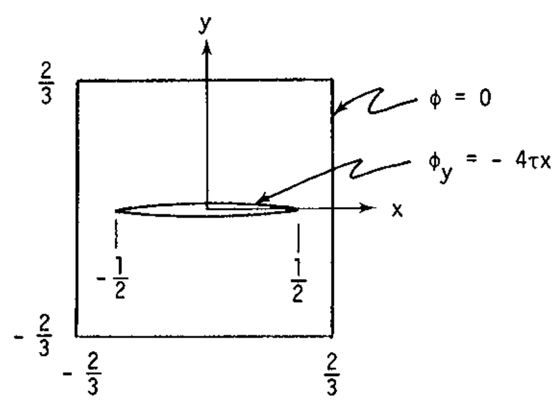
The state-of-the-art of transonic flow calculations has advanced to the point where two-dimensional flows, including the effects of viscosity, can be computed in a relatively short time on modern serial-type computers. For example, many people are using a program developed at the Courant Institute of New York University for the analysis of transonic flow past airfoils.[1] This program gives accurate solutions to the full-potential equation, including the effects of boundary-layer displacement, in about 2 or 3 minutes on a CDC CYBER 175 computer. Three-dimensional, transonic, finite-difference calculations, however, are expensive on this type of computer. The three-dimensional program developed at New York University[1] takes about half an hour for an inviscid calculation on a fairly crude grid.

It is hoped that the use of the STAR-100 computer will allow accurate, three-dimensional, transonic flow calculations to be done economically. One way to achieve this goal is through the development of algorithms which can make full use of the unique architecture of the STAR-100. The STAR computer has a "pipeline" type of processor which is very efficient in doing arithmetic operations on long vectors.[2] Unfortunately, the best available method for solving the transonic potential equation is successive line over-relaxation (SLOR), which is not amenable to vector arithmetic. The reason for this is the semi-implicit nature of the iterative method; that is, the calculations at a particular grid point require results from the current iteration at neighboring grid points and thus cannot be done in long vector operations.

This paper describes a new explicit algorithm which can be vectorized for use on the STAR-100. The new algorithm is applied to a simple test case and compared to SLOR on the CYBER 175 computer.

Test Problem

The test problem chosen for this preliminary study is to solve the transonic, nonlinear, small-disturbance potential equation for a nonlifting parabolic-arc airfoil in a finite box with uniform grid, as shown in the sketch below.



Although this is a very simple physical situation, it still has some of the most difficult features of transonic flow fields as far as programing for the STAR is concerned. The governing partial differential equation is

$$\left[1 - M_\infty^2 - (\gamma+1)\, M_\infty^2\, \phi_x\right]\phi_{xx} + \phi_{yy} = 0$$

The boundary conditions to be applied are

$\phi = 0$ on the outer boundary

and

$\phi_y = \mp 4\tau x$ on $y = \pm 0,\ -.5 \le x \le .5$

*Aerospace Engineer, Member AIAA
**Professor of Computer Science, Courant Institute of Mathematical Sciences, Member AIAA

where τ is the thickness of the parabolic-arc airfoil (which has a unit chord). For $\tau = 0.1$ and $M_\infty = 0.9$ the flow is supercritical. In regions where the coefficient of ϕ_{xx} is positive, the flow is subsonic and the equation is elliptic type. In regions where the coefficient of ϕ_{xx} is negative, the flow is supersonic and the equation is hyperbolic type. The general procedure for solving this equation is to replace the partial differential equation with a finite-difference equation at each grid point. These finite-difference equations are then solved iteratively.

This test problem represents a simple physical situation which is of little practical interest. A more useful program should allow for lifting flows and should extend the outer boundary farther away from the airfoil. This could be done either by using a stretched grid or by some type of grid nesting using additional coarse grids around the small region considered here. The test case does, however, include the major difficulties to be overcome in using the STAR-100 computer for transonic flows. For example, it has supercritical flow which requires a change from one type of difference equation at subsonic (elliptic) points to another type of difference equation at supersonic (hyperbolic) points. It also requires the use of an explicit iterative scheme to solve the difference equations if the arithmetic operations are to be done using long vector instructions.

Semi-Implicit Solution Method

The most common method used to solve the finite-difference equations is successive line over-relaxation (SLOR). This iterative scheme is implemented as follows:

Compute $U \equiv 1 - M_\infty^2 - (\gamma+1) M_\infty^2 \phi_x$

using

$$\phi_x = \frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2\Delta x}$$

If $U > 0$ (subsonic points), central differences are used together with over-relaxation ($\omega > 1$) to give:

$$U \frac{\phi_{i+1,j}^n - \frac{2}{\omega}\phi_{i,j}^{n+1} - 2\left(1 - \frac{1}{\omega}\right)\phi_{i,j}^n + \phi_{i-1,j}^{n+1}}{\Delta x^2}$$

$$+ \frac{\phi_{i,j+1}^{n+1} - 2\phi_{i,j}^{n+1} + \phi_{i,j-1}^{n+1}}{\Delta y^2} = 0$$

where the superscript indicates the iteration number.

If $U < 0$ (supersonic points), an upwind difference is used for ϕ_{xx} to give:

$$U \frac{\phi_{i,j}^{n+1} - 2\phi_{i-1,j}^{n+1} + \phi_{i-2,j}^{n+1}}{\Delta x^2}$$

$$+ \frac{\phi_{i,j+1}^{n+1} - 2\phi_{i,j}^{n+1} + \phi_{i,j-1}^{n+1}}{\Delta y^2} = 0$$

At the airfoil boundary the ϕ_{yy} term is replaced by

$$\phi_{yy} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta y^2} - \frac{2\phi_y\big|_{y=0}}{\Delta y}$$

which takes into account the fact that the nonlifting flow is symmetric (i.e., $\phi_{i,j-1} = \phi_{i,j+1}$ at $y = 0$). Each iteration is generated one column at a time going from left to right, by solving a tridiagonal system of equations at each column. This nonconservative scheme is similar to that originally proposed by Murman and Cole.[3] See references 4 and 5 for a discussion of related conservative schemes.

Explicit Solution Method

The explicit solution method uses values of the potential function from the two previous iterations in order to update the potential function for the present iteration (thus termed a three-level scheme). The method is implemented as follows:

Compute $U \equiv 1 - M_\infty^2 - (\gamma+1) M_\infty^2 \phi_x$

using

$$\phi_x = \frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2\Delta x}$$

If $U > 0$ (subsonic points), central differences are used at iteration, or "time," level n and the potential function is updated using:

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + (P_1 - 1)\left(\phi_{i,j}^n - \phi_{i,j}^{n-1}\right)$$

$$+ D_1 R_1 \qquad (1)$$

where

$$R_1 = U\left(\phi_{i+1,j}^n - 2\phi_{i,j}^n + \phi_{i-1,j}^n\right)$$

$$+ \frac{\Delta x^2}{\Delta y^2}\left(\phi_{i,j+1}^n - 2\phi_{i,j}^n + \phi_{i,j-1}^n\right)$$

$$D_1 = \frac{P_1 P_2^{\;2}}{2\left[U + \left(\frac{\Delta x}{\Delta y}\right)^2\right]}$$

and P_1 and P_2 are parameters of the algorithm. It is interesting to rearrange equation (1) to give

$$\phi^{n+1}_{i,j} - 2\phi^{n}_{i,j} + \phi^{n-1}_{i,j} + (2 - P_1)\left(\phi^{n}_{i,j} - \phi^{n-1}_{i,j}\right) = D_1 R_1$$

which shows that the iteration process is related to the time dependent equation

$$\phi_{tt} + a\phi_t = b(U\phi_{xx} + \phi_{yy})$$

If $U < 0$ (supersonic points), the upwind ϕ_{xx} difference is formed at "time" level $n-1$, while ϕ_{yy} is centrally differenced and averaged in time level n. The potential function is updated using

$$\phi^{n+1}_{i,j} = \phi^{n}_{i,j} + (1 - 2\sigma)\left(\phi^{n}_{i,j} - \phi^{n-1}_{i,j}\right)$$

$$+ 2\sigma\left(\phi^{n}_{i-1,j} - \phi^{n-1}_{i-1,j}\right) + D_2 R_2 \qquad (2)$$

where

$$R_2 = \overline{U}\left(\phi^{n-1}_{i,j} - 2\phi^{n-1}_{i-1,j} + \phi^{n-1}_{i-2,j}\right)$$

$$+ \left(\frac{\Delta x}{\Delta y}\right)^2 \left[(1-\sigma)\left(\phi^{n}_{i,j+1} - 2\phi^{n}_{i,j} + \phi^{n}_{i,j-1}\right)\right.$$

$$\left. + \sigma\left(\phi^{n}_{i-1,j+1} - 2\phi^{n}_{i-1,j} + \phi^{n}_{i-1,j-1}\right)\right]$$

$$\overline{U} = (1-\sigma)\,U + \sigma\widetilde{U}$$

$$\widetilde{U} = 1 - M_\infty^2 - (\gamma+1)\,M_\infty^2\,\frac{\phi^{n-1}_{i,j} - \phi^{n-1}_{i-2,j}}{2\Delta x}$$

$$\sigma = P_2 \min\left(1, \sqrt{|U|}\,\frac{\Delta y}{\Delta x}\right)$$

and

$$D_2 = \frac{P_2^{\;2}\left(\frac{\Delta y}{\Delta x}\right)^2}{\max\left[1,\ |U|\left(\frac{\Delta y}{\Delta x}\right)^2\right]}$$

Again it is interesting to rearrange equation (2) to give

$$\phi^{n+1}_{i,j} - 2\phi^{n}_{i,j} + \phi^{n-1}_{i,j} + 2\sigma\left(\phi^{n}_{i,j} - \phi^{n-1}_{i,j}\right.$$

$$\left. - \phi^{n}_{i-1,j} + \phi^{n-1}_{i-1,j}\right) = D_2 R_2 \qquad (3)$$

which is related to the time-dependent equation

$$\phi_{tt} + c\phi_{xt} = d(U\phi_{xx} + \phi_{yy})$$

A von Neumann stability analysis of this last scheme with the $\overline{U}$ replaced by U shows why the ϕ_{xx} derivative is evaluated at the $n-1$ iteration and why the ϕ_{yy} derivative is a weighted average between ϕ_{yy} at the i column and ϕ_{yy} at the $i-1$ column. To do the von Neumann analysis, let ϕ at the kth iteration be

$$\phi^{(k)} = g^k e^{imx} e^{iny}$$

Also let $m\Delta x = \xi$ and $n\Delta y = \eta$. Putting these definitions into equation (3) gives

$$g^2 - 2g + 1 + 2\sigma\,(g - 1 - ge^{-i\xi} + e^{-i\xi})$$

$$= D_2 \left\{U\,(1 - 2e^{-i\xi} + e^{-2i\xi})\right.$$

$$+ g\left(\frac{\Delta x}{\Delta y}\right)^2 \left[(1-\sigma)(e^{i\eta} - 2 + e^{-i\eta})\right.$$

$$\left.\left. + \sigma\,(e^{-i\xi}e^{i\eta} - 2e^{-i\xi}e^{-i\eta})\right]\right\} \qquad (4)$$

Note that

$$D_2 U = -\sigma^2$$

Also define

$$\rho \equiv 1 - 2\sigma^2 \frac{1}{|U|}\left(\frac{\Delta x}{\Delta y}\right)^2 \sin^2 \frac{\eta}{2}$$

so equation (4) becomes

$$g^2 - 2\rho g \left[1 - \sigma\,(k - e^{-i\xi})\right] + \left[1 - \sigma\,(1 - e^{i\xi})\right]^2 = 0$$

or

$$g = \left(\rho \pm i\sqrt{1 - \rho^2}\right)\left[1 - \sigma\,(1 - e^{-i\xi})\right]$$

Thus, in order to have $|g| \leq 1$, it is necessary and sufficient to have both $|\rho| \leq 1$ and $\sigma \leq 1$.

The inequality $|\rho| \leq 1$ implies that $\sigma \leq \sqrt{|U|}\,\frac{\Delta y}{\Delta x}$.

This convenient factoring of the expression for the amplification factor was made possible by choosing this particular representation for the ϕ_{xx} and ϕ_{yy} derivatives. In practice the coefficient of ϕ_{xx} is $\overline{U}$ rather than U; this averaging makes the scheme approach second-order accuracy as σ approaches 1.

It should be noted that this explicit scheme is not only a different iterative algorithm from the semi-implicit scheme, but it also has a different steady-state solution. This difference occurs because of the weighted averaging done on both the ϕ_{yy} term and on the coefficient of the ϕ_{xx} term for supersonic points.

Convergence and Timing Comparisons

Short computer programs have been written to solve the sample problem using the two methods described. The semi-implicit SLOR method was coded in standard FORTRAN IV and run on a CDC CYBER 175 computer (which is about 2½ times as fast as a CDC 6600 for this type of problem). The explicit method was coded in STAR FORTRAN[6] and run on a CDC STAR-100 computer. The STAR code includes the use of vector instructions in the iteration loop. It also includes the use of bit control vectors to distinguish between subsonic and supersonic points. The bit control vectors provide the capability of performing the complicated supersonic calculations only at supersonic points, which are collected into a vector through the "compress" and "expand" type of instructions available on the STAR.

The computations were done on three different grids. Each calculation was terminated when the value of the largest residual in the flow field was less than $\frac{1}{2}(\Delta x^2 + \Delta y^2)$. Each calculation was run with experimentally-determined optimum values of the parameters for that algorithm so that convergence was attained in a minimum number of cycles. The results in the table below show that the new three-level explicit scheme has a slower convergence rate than SLOR. However, the efficiency of this new scheme on the STAR computer is enough to make up for the slower convergence rate and still allow an overall reduction in computing time. Not surprisingly, the speedup is greater for the cases with finer grids than for the 40X40 case.

Concluding Remarks

This preliminary study has shown that a new explicit method for solving the transonic small-disturbance potential equation on the STAR-100 computer can almost halve the computer time required for this type of computation when compared to successive line over-relaxation on the CDC CYBER 175 computer. These results are limited to a relatively simple problem with a uniform Cartesian grid. Although the speedup is not as great as desired, it is enough to justify further study of this method. The effects of lift and of grid stretching on the convergence rates of the schemes should be investigated. Also, the new explicit scheme should be applied to the full potential equation.

There are several possibilities for obtaining further reductions in computer time. One is through the development of more efficient algorithms. Improvements might be made in the convergence rate of explicit algorithms, or other vectorizable algorithms might be developed. Another possibility is through programing techniques to get successive line over-relaxation to run as efficiently as possible on the STAR-100 computer. Although the method is semi-implicit, there are portions of it which can be written in vector instructions of short length.

Grid size	SLOR ON CYBER 175			EXPLICIT METHOD ON STAR-100		
	Cycles to converge	Time to converge	Average sec/cy.	Cycles to converge	Time to converge	Average sec/cy.
40X40	42	1.242	.0296	131	.879	.0067
80X80	98	12.118	.1237	300	6.385	.0213
160X160	244	116.095	.4758	655	59.834	.0913

References

1. Bauer, Frances; Garabedian, Paul; Korn, David; and Jameson, Antony: Supercritical Wing Sections II. Lecture Notes in Economics and Mathematical Systems, vol. 108, Springer-Verlag, 1975.

2. Control Data STAR-100 Computer Hardware Reference Manual. CDC Publication No. 60256000, 1975.

3. Murman, Earll M.; and Cole, Julian D.: Calculation of Plane Steady Transonic Flows. AIAA J., vol. 9, no. 1, 1971, pp. 114-121.

4. Murman, Earll M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. AIAA Computational Fluid Dynamics Conference, Palm Springs, Calif., 1973.

5. Jameson, Antony: Transonic Potential Flow Calculations Using Conservation Form. AIAA 2nd Computational Fluid Dynamics Conference, Hartford, Conn., 1975.

6. STAR FORTRAN Language Version 2 Reference Manual. CDC Publication No. 60386200, 1977.