the freely rotatable splitter plate case as well. Namely, as the flow visualizations indicate, the vortex formation length scale increases with L/D and, hence, shedding frequency decreases with L/D. As with the flow visualizations of Fig. 3, for $L/D > 2$, the fixed and freely rotatable splitter plates behave similarly, as indicated by the overlap of data in Fig. 4.

Conclusions

When allowed to rotate freely, a splitter plate shorter than five cylinder diameters does not align itself with the freestream. Instead, it migrates to a stable position on one side of the wake or the other, with the splitter plate angle a strong function of splitter plate length, but independent of Reynolds number in the range tested. The wake of a freely rotatable splitter plate smaller than about two cylinder diameters is not significantly altered by the presence of the plate. When comparing vortex formation length and Karman vortex street shedding frequency in the near wake, a cylinder with a free splitter plate behaves much differently than does a cylinder with a rigidly mounted splitter plate of the same length. On the other hand, a freely rotatable splitter plate larger than about two cylinder diameters, although positioning itself at a nonzero angle, affects the wake in nearly the same manner as does a rigid plate of the same length. The inverse relationship between vortex formation length and vortex shedding frequency, first suggested by Gerrard,[6] is apparently valid for both fixed and free splitter plates.

References

[1]Roshko, A., "On the Development of Turbulent Wakes from Vortex Streets," NACA TN 2913, March 1953.

[2]Bearman, P. W., "Investigation of the Flow Behind a Two-Dimensional Model with a Blunt Trailing Edge and Fitted with Splitter Plates," *Journal of Fluid Mechanics*, Vol. 21, Part 2, Feb. 1965, pp. 241–255.

[3]Cimbala, J. M., Garg, S., and Park, W. J., "The Effect of a Non-Rigidly Mounted Splitter Plate on the Flow over a Circular Cylinder," *Bulletin of the American Physics Society*, Vol. 33, No. 10, Nov. 1988, p. 2249.

[4]Xu, J. C., Sen, M., and Gad-el-Hak, M., "Low-Reynolds Number Flow Over a Rotatable Cylinder-Splitter Plate Body," *Physics of Fluids A*, Vol. 2, No. 11, 1990, pp. 1925–1927.

[5]Garg, S., "An Experimental Investigation of the Effects of Non-Rigidly-Mounted Splitter Plates on the Wakes of Circular Cylinders," M.S. Thesis, Pennsylvania State University, University Park, PA, May 1990.

[6]Gerrard, J. H. "The Mechanics of the Formation Region of Vortices Behind Bluff Bodies," *Journal of Fluid Mechanics*, Vol. 25, Part 2, June, 1966, pp. 401–413.

[7]Apelt, C. J., West, G. S. and Szewczyk, A. A., "The Effects of Wake Splitter Plates on the Flow Past a Circular Cylinder in the Range $10^4 < R < 5 \times 10^4$," *Journal of Fluid Mechanics*, Vol. 61, Part 1, Oct. 1973, pp. 187–198.

Solver for Unfactored Implicit Schemes

Marcello Vitaletti*
IBM European Center for Scientific and Engineering Computing, Rome, Italy

Introduction

CODES based on implicit schemes for Euler and Navier-Stokes computations of transonic flows are very demanding in terms of computing time and storage, because the flowfield must be expressed at each time step as the solution of a large system of nonlinear equations.

The linearization of an implicit time-stepping scheme for solving the flow equations produces a system of linear equations whose coefficients matrix, hereafter referred to as the "implicit-step operator," has a sparse structure. In alternating direction implicit (ADI) schemes[1] this matrix operator is approximately factored as the product of block-tridiagonal matrices, in order to maintain the computational costs of a direct numerical-solving procedure within acceptable limits. However, the ADI factorization error prevents the use of large time steps as a method for producing a fast elimination of the transient solution in steady-state computations,[2] and it may be the main source of inaccuracy in the simulation of unsteady flows. This paper illustrates the use of the conjugate gradient squared (CGS) iterative algorithm[3] for solving the implicit-step operator in unfactored form. In the present work the flow equations are linearized at each time step, following the Beam-Warming approach, and the ADI splitting of the implicit-step operator is used to build an efficient preconditioner of the unfactored system.

Numerical Scheme

In general curvilinear coordinates, in two space dimensions, the Euler equations, describing the flow of an inviscid gas in thermodynamic equilibrium, may be written in the following strong conservation law form:

$$\partial_t \boldsymbol{q} + \partial_\xi \boldsymbol{F} + \partial_\eta \boldsymbol{G} = 0 \tag{1}$$

where $\xi = \xi(x,y,t)$ and $\eta = \eta(x,y,t)$ are the body-fitted curvilinear coordinates, and $\boldsymbol{q}$ is a vector whose four components are proportional to the conserved physical quantities of the flow. The flux vectors $\boldsymbol{F}$ and $\boldsymbol{G}$ are nonlinear functions of the flow variables $\boldsymbol{q}$ and also depend on geometric terms related to the mapping between the Cartesian and the curvilinear coordinates.[2]

By use of the Euler first-order implicit time-differencing scheme, the solution incremental change $\Delta \boldsymbol{q}^n = (\boldsymbol{q}^{n+1} - \boldsymbol{q}^n)$ at time $n\Delta t[\boldsymbol{q}^n = \boldsymbol{q}(n\Delta t)]$ is written as a funciton of the nonlinear flux vectors at time $(n+1)\Delta t$. A system of linear algebraic equations is then obtained by a linear expansion of the flux vectors $\boldsymbol{F}^{n+1}$ and $\boldsymbol{G}^{n+1}$ about the solution vector $\boldsymbol{q}^n$ and by a spatial discretization of the linearized probability density errors. On a numerical grid with uniform grid spacings $\Delta\xi = 1$ and $\Delta\eta = 1$, the partial derivatives are approximated by second-order centered finite-difference operators δ_ξ and δ_η such the $\delta_\xi f_{i,j} = (f_{i+1,j} - f_{i-1,j})/2$ and $\delta_\eta f_{i,j} = (f_{i,j+1} - f_{i,j+1})/2$, and the nonlinear artificial dissipation method developed by Jameson et al.[4] is used for damping high-frequency spurious oscillation modes triggered by discontinuities in the flow solution. The resulting linear algebraic system then takes the following form:

$$\{1 + \Delta t[\delta_\xi \boldsymbol{A}^n + \tilde{\delta}_\xi^2 + \tilde{\delta}_\xi^4) + (\delta_\eta \boldsymbol{B}^n + \tilde{\delta}_\eta^2 + \tilde{\delta}_\xi^4) +]\}\Delta \boldsymbol{q}^n$$

$$= -\Delta t[(\delta_\xi \boldsymbol{F}^n + \tilde{\delta}_\xi^2 \boldsymbol{q}^n + \tilde{\delta}_\xi^4 \boldsymbol{q}^n) + (\delta_\eta \boldsymbol{G}^n + \tilde{\delta}_\eta^2 \boldsymbol{q}^n + \tilde{\delta}_\eta^4 \boldsymbol{q}^n)] \tag{2}$$

where $\boldsymbol{A}^n = \partial \boldsymbol{F}(\boldsymbol{q}^n)/\partial \boldsymbol{q}$ and $\boldsymbol{B}^n = \partial \boldsymbol{G}(\boldsymbol{q}^n)/\partial \boldsymbol{q}$ are the Jacobians of the flux vectors. The three-point second-order terms $\tilde{\delta}^2 \boldsymbol{q}$ and the five-point fourth-order terms $\tilde{\delta}^2 \boldsymbol{q}$, appearing on both the implicit and the explicit side of the equations, are the artificial dissipation corrections.

The convergence of the time-marching scheme to a steady-state solution is accelerated by using a spatially varying time step $\Delta t_{i,j} = \Delta t_0/(1 + \sqrt{J_{i,j}})$, where $J_{i,j}$ is an estimate of the cell surface and the constant value Δt_0 is usually chosen in the interval between 1 and 10 for typical inviscid simulations.

Received Jan. 8, 1990; revision received June 5, 1990; accepted for publication June 26, 1990.

*Research Scientist, Via Giorgione, 159 I-00147.

CGS-ADI Solver

The implicit-step operator on the left-hand side of Eq. (2) would be approximated, in ADI schemes, as the product of two block-pentadiagonal factors, due to the presence of the fourth-order terms. The extra cost of solving block-pentadiagonal systems, instead of block-tridiagonal systems, usually offsets the advantage of an implicit treatment of the fourth-order terms in ordinary ADI schemes.[2] On the other hand, the fact that three out of the five blocks associated to a grid node have zero off-diagonal elements suggests that the sparsity of the coefficient matrix can be exploited by an iterative solver. Only 21 nonzero elements for each matrix row need to be stored, in this case, to represent the full unfactored operator (the nonzero elements are 17, instead of 21, when fourth-order terms are absent).

Among the various generalizations of the CG algorithm that have been proposed for solving systems that are not symmetric and not positive definite, the CGS algorithm, recently proposed by Sonnenveld,[3] has been preferred to other methods because of the fast convergence attained in solving the linear system of Eq. (2). Indeed, although closely related to the so-called biconjugate gradients (Bi-CG) algorithm,[5] CGS is usually more efficient than Bi-CG and does not require the adjoint A^T of the coefficient matrix. The method is simpler to implement than the generalized minimal residual (GMRES) algorithm,[6] in which a set of orthogonal "search directions" must be computed by a direct orthogonalization procedure, and the number of search directions needs to be tailored to the specific problem to attain optimal efficiency. (However, it may be useful for the reader who is interested in the use of GMRES to compare this work with that of Wigton et al.,[7] who have proposed a modified GMRES algorithm employing approximate Newton's iterations.) The CGS algorithm applied to the system $\boldsymbol{Ax} = \boldsymbol{b}$ reads as follows:

$$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$$

$$\boldsymbol{q}_0 = \boldsymbol{p}_{-1} = 0; \qquad \rho_{-1} = 1$$

$$n = 0$$

while residual > tolerance **do**
begin

$$\rho_n = \tilde{\boldsymbol{r}}_0^T \boldsymbol{r}_n; \qquad \beta_n = \frac{\rho_n}{\rho_{n-1}}$$

$$\boldsymbol{u}_n = \boldsymbol{r}_n + \beta_n \boldsymbol{q}_n$$

$$\boldsymbol{p}_n = \boldsymbol{u}_n + \beta_n(\boldsymbol{q}_n + \beta_n \boldsymbol{p}_{n-1})$$

$$\boldsymbol{v}_n = \boldsymbol{A}\boldsymbol{p}_n$$

$$\sigma_n = \tilde{\boldsymbol{r}}_0^T \boldsymbol{v}_n; \qquad \alpha_n = \frac{\rho_n}{\sigma_n}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{u}_n - \alpha_n \boldsymbol{v}_n$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \alpha_n \boldsymbol{A}(\boldsymbol{u}_n + \boldsymbol{q}_{n+1})$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n(\boldsymbol{u}_n + \boldsymbol{q}_{n+1})$$

$$n = n + 1 \qquad (3)$$

end

where $\boldsymbol{x}_0$ is the starting estimate of the solution, and the vector $\tilde{\boldsymbol{r}}_0$ is arbitrary and must be suitably chosen. In practice, $\tilde{\boldsymbol{r}}_0$ is usually chosen to be equal to $\mathbf{r}_0$ and $\boldsymbol{x}_0 = 0$ is a natural choice when applying CGS to solve Eq. (2). The identity $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$, following from Eqs. (3), says that $\boldsymbol{r}_n$ is the residual associated with the approximate solution vector $\boldsymbol{x}_n$ at the nth iterate. The convergence rate of CG-type algorithms depends on the spectral properties of the coefficient matrix and can be effectively accelerated by *preconditioning* the system, i.e, transforming the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ into an equivalent one whose coefficient matrix has a lower spectral condition number. This is obtained by using an approximation $\boldsymbol{P}$ of the matrix $\boldsymbol{A}$, which should be economically solvable by a direct method, and applying the CGS algorithm to the equivalent system $\tilde{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{P}^{-1}\boldsymbol{b} = \tilde{\boldsymbol{b}}$. In practice, each time in a CGS iteration one has to compute a matrix-vector product of the form $\boldsymbol{y} = \tilde{\boldsymbol{A}}\boldsymbol{x}$, with $\tilde{\boldsymbol{A}} = \boldsymbol{P}^{-1}\boldsymbol{A}$, the vector $\boldsymbol{y}$ is actually computed by solving a linear system of the form $\boldsymbol{Py} = (\boldsymbol{Ax})$.

The numerical experiments showed that the following ADI approximate factorization of the implicit-step operator,

$$\boldsymbol{P} = [1 + \Delta t(\delta_\xi \boldsymbol{A}^n + \hat{\delta}_\xi^2)][1 + \Delta t(\delta_\eta \boldsymbol{B}^n + \hat{\delta}_\eta^2)] \qquad (4)$$

where $\hat{\delta}_\xi^2$ and $\hat{\delta}_\eta^2$ are three-point stencil approximations of the dissipative terms, acts as an effective preconditioner of the CGS algorithm. Indeed, the preconditioned solver, hereafter referred to as CGS-ADI, normally converges in about 3–10 times fewer iterations than CGS without preconditioning.

Results and Discussion

The unfactored implicit scheme has been applied to compute the transonic flow about the NACA 0012 wing profile for two common test cases. The values of the freestream Mach number M and of the angle of attack α for the first problem are $M = 0.8$ and $\alpha = 1.25$, and for the second problem they are $M = 0.85$ and $\alpha = 1$. The convergence rate and the computational cost of CGS-ADI were compared with those obtained by an ADI code employing block-tridiagonal factors. Boundary conditions were treated explicitly. The flows were computed on a 250×64 O-grid extending for about 30 chords away from the body, without any special clustering of the grid nodes in the vicinity of the shocks. A relatively high number of nodes in the direction normal to the body was used to achieve a good accuracy in the computation of the pressure gradients at the body surface, where the grid cells are slightly stretched along the direction of the flow stream. The relative performance of the two codes on a grid with such characteristics may give a preliminary indication of the behavior of the same codes in viscous simulations. In all of the experiments a locally variable time step was used to obtain a fast elimination of the transient solution. The maximum computational efficiency of the CGS-ADI code was obtained by requiring a moderate level of accuracy, about 10^{-2}, in the CGS iterative solution, so that only 2 iterations were approximately required for the converge of CGS. For the convergence to the steady state, two criteria were adopted in comparing the two codes. The first criterion is met when the lift coefficient C_L is accurate to the fourth decimal digit, whereas the second criterion requires that the L_2 norm of the residual is smaller than 10^{-8}.

On the first test case the C_L convergence was obtained in less than 300 time-steps with CGS-ADI, whereas it required 950 iterations with ADI. Respectively 600 and 1800 steps were required to meet the second convergence criterion. Figure 1 illustrates the improvement in the convergence rate obtained with the CGS-ADI code compared to the ADI code. The continuous line refers to the ADI code running with a locally constant time-step size, corresponding to a maximum Courant number (CFL) of about 13, the maximum that the ADI code can tolerate in the transient phase. The convergence history for the CGS-ADI code (dotted line) clearly demonstrates the ability of the unfactored scheme to exploit the use of larger time steps, corresponding to a CFL number of about 40, to accelerate the convergence to the steady-state solution. An adaptive procedure was optionally activated in the ADI code (dashed line) to increase the time-step size outside the transient phase, in order to accelerate the convergence. In this case, however, the advantages of using larger time steps are only marginal, due to the ADI factorization errors, and the code did not converge for values of the CFL number larger than 20.

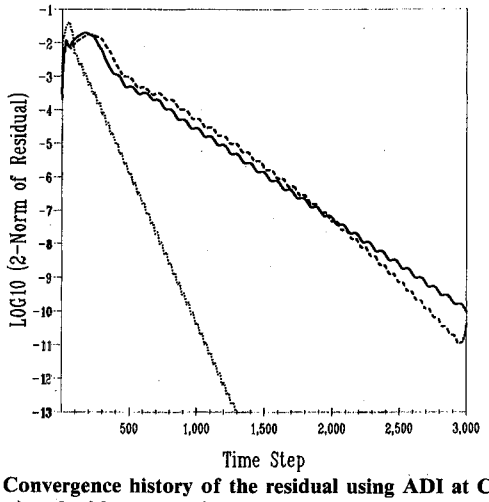

Fig. 1 Convergence history of the residual using ADI at CFL = 13 (solid line) and with an adaptive time step at CFL = 20 (dashed line) and CGS-ADI at CFL = 40 (dotted line).

Table 1 Computational efficiency of the CGS-ADI code and of the ADI code

Test case	Code	Number of steps	CPU time, s
1	CGS-ADI	600	1643
1	ADI	1800	2813
2	CGS-ADI	700	1955
2	ADI	2600	4105

The results on the second test cases show the same qualitative behavior. However, the convergence of the ADI code to the steady state is significantly slower than in the first case, probably due to the presence of greater shocks.

Table 1 summarizes the computational costs for obtaining a converged solution (residual L_2 norm smaller than 10^{-8}) with the two codes. All of the relevant computational kernels of both codes run at about 20 Mflops on one vector processor of the IBM 3090 model 600E. The greater cost per time step of the CGS-ADI code is largely compensated, on both problems, by its faster convergence rate. Indeed, for both convergence criteria CGS-ADI required about 1.7 times less CPU time than ADI on the first problem and about 2.1 less CPU time on the second problem.

Conclusion

The numerical stability bounds of implicit schemes based on the Beam-Warming linearization procedure are adversely affected by the factorization errors introduced by ADI splitting methods. In order to accelerate the convergence to the steady state, the use of large time steps may be exploited more effectively when the unfactored form of the implicit-step operator is retained. In two-dimensional problems the CGS-ADI iterative algorithm is very efficient for solving the unfactored implicit-step operator, and the implemented code was found to be significantly faster than an ADI code employing block-tridiagonal factors. On two transonic airfoil test cases the CGS-ADI code required, respectively, 1.7 and 2 times less CPU time than the ADI code to obtain a converged solution.

References

[1]Beam, R. M., and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," *Journal of Computational Physics*, Vol. 22, No. 1, 1976, pp. 87–110.

[2]Pulliam, T. H., "Euler and Thin Layer Navier Stokes Codes: ARC2D, ARC3D," *Notes for the Computational Fluid Dynamics User's Workshop*, Univ. of Tennessee Space Institute, Tullahoma, TN, 1984.

[3]Sonnenveld, P., "CGS, a Fast Lanczos-Type Solver for Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 10, No. 1, 1989, pp. 36–52.

[4]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[5]Fletcher, R., "Conjugate Gradient Methods for Indefinite Systems," *Lecture Notes in Mathematics 506*, Springer-Verlag, Berlin, Germany 1976, pp. 73–89.

[6]Saad, Y., Schultz, M.H. "GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[7]Wigton, L. B., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamics Codes," AIAA Paper 85-1494, 1985.

Study of Wake Optical Properties

N. Gerber* and R. Sedney†
U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland 21005

An experiment[1] and associated theoretical study[2] were performed to study the optical transmittal properties of the turbulent wake behind a supersonic projectile. The experiment consisted of taking nearly end-on shadowgraphs of the wake and projectile. A projectile was fired toward a photographic film oriented normal to the trajectory. A small turning mirror located near the trajectory reflected a pulsed, diverging beam of laser light (wavelength 6943 Å) upstream through the wake toward the projectile at a predetermined time. This short-duration pulse (< 100 ns) imprinted a shadow image on the film before the projectile pierced it. The projectile shadow was centered off the trajectory, and the piercing hole left enough of it so that more than half of the boundary appeared on the film, as seen in a typical shadowgraph, Fig. 1.

The projectile shadowgraphs were then measured. In every test, the shadow with wake was significantly larger than that computed by geometrical projection, indicating that the wake acted as a diverging lens.

Geometrical optics was the theoretical approach employed to determine the projectile shadow. The differential equations for propagation of light rays in a nonhomogeneous medium[3] were integrated for the light traveling upstream from the pulsed point source. Those rays that grazed the base of the projectile were then extended in straight lines from base to screen in the directions that they had at the base, thus forming the theoretical shadow. The equations to be integrated are

$$\frac{\partial F}{\partial x} - \frac{d(\partial F/\partial \dot{x})}{dz} = 0$$

$$\frac{\partial F}{\partial y} - \frac{d(\partial F/\partial \dot{y})}{dz} = 0$$

where

$$\dot{x} \equiv \frac{dx}{dz}, \qquad \dot{y} \equiv \frac{dy}{dz}$$

$$F(z,x,y,\dot{x},\dot{y}) \equiv n(x,y,z)(\dot{x}^2 + \dot{y}^2 + 1)^{1/2}$$

The x, y plane is normal to the trajectory, which coincides with the z axis; the light source is located on $y = 0$. The index

Received Feb. 20, 1990; revision received May 5, 1990; accepted for publication May 21, 1990. This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Aerospace Engineer, Launch and Flight Division. Member AIAA.
†Research Scientist, Launch and Flight Division. Member AIAA (deceased).