An Implicit LU Scheme for the Euler Equations Applied to Arbitrary Cascades

Edward K. Buratynski* and David A. Caughey†
Cornell University, Ithaca, New York

A new implicit scheme for solving the Euler equations has been implemented and tested. Derivation of the scheme initially follows that of the popular alternating direction implicit scheme. The principal difference between the two schemes lies in the form of the approximate factoring of the implicit operator. The new scheme results in only two factors, independent of the spatial dimension of the problem. Each factor represents an algebraic system that is either a lower (L) or upper (U) block diagonal—hence, the name LU. Inversion of such systems is relatively simple and efficient. Implementation of the LU scheme is in a finite volume form that permits the calculation of realistic problems by allowing the use of arbitrary grids. Also, the conservation property of this form allows the capture of weak solutions. The present application of the scheme is for flows through two-dimensional cascades. Results are given for subsonic and transonic flows with varying degrees of geometrical complexity. Comparisons with other numerical and analytical solutions show good agreement. Execution times are compared with other methods and it is verified that the LU scheme is an attractive method for computing solutions to the Euler equations.

I. Introduction

A NEW scheme for solving the Euler equations has been implemented and tested for the problems of subsonic and transonic flow through two-dimensional cascades. Efficiency has been the motivation for the scheme and is achieved primarily in three ways:

1) The finite difference equations are in a finite volume formulation. This permits the calculation of realistic problems by allowing the use of arbitrary grids. These grids are usually generated numerically and can have the advantages of being boundary conforming and efficiently refined in regions of large flow gradients. Figure 1 shows an example of such a grid. The figure also shows the periodic "H-type" grid topology for which the code is written.

2) The numerical scheme is an implicit one, which remains stable as the Courant number (proportional to the ratio of the time step to the mesh spacing) becomes arbitrarily large. This allows the use of a larger time step than is permitted by an explicit method for problems that are slowly time varying or in which convergence to steady state is desired.

3) Numerical efficiency is achieved by the particular form of the factoring of the implicit operator.

Proposed by Jameson and Turkel,[1] the scheme always has two factors independent of the dimension of the problem. The use of one-sided difference operators in each factor results in matrices that are either lower (L) or upper (U) block diagonal; hence, the name LU. The blocks arise because of the system of coupled differential equations and are 4×4 for two-dimensional problems or 5×5 for three-dimensional problems. Inversion of such systems is very simple and efficient, since for each factor only a point-by-point sweep through the domain is required. At each point, the local diagonal block is inverted.

Presented as Paper 84-0167 at the AIAA 22nd Aerospace Sciences Meeting, Reno, NV, Jan. 9-12, 1984; received Feb. 7, 1984; revision received June 3, 1985. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1985. All rights reserved.

*Graduate Research Assistant, Theoretical and Applied Mechanics (presently Member, Research Staff, AT&T Technologies, Inc., Princeton, NJ).

†Professor, Sibley School of Mechanical and Aerospace Engineering. Associate Fellow AIAA.

This scheme is to be contrasted to the popular alternating direction implicit (ADI) factoring.[2-5] In this scheme, each factor requires the inversion of a block tridiagonal system along grid lines. Also, the ADI scheme has a factor for each spatial dimension, resulting in an extra factor for three-dimensional problems. For these reasons, it is expected that the LU factoring is more efficient than the ADI, especially in three dimensions.

There is also a long history of calculating compressor and turbine cascade flowfields using explicit, time-marching methods to solve the finite difference or finite volume/area equations.[6-10] For a review of these methods, see the survey by Habashi,[11] who also discusses use of the finite element technique.

II. Mathematical Model

If (x,y) are Cartesian coordinates and (ξ,η) general transformed coordinates, then the Jacobian of the transformation J may be defined as

$$J=\begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}$$

with

$$J^{-1}=\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}=\frac{1}{D}\begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix}$$

where D is the determinant of J. The divergence form of the Euler equations in the (ξ,η) coordinates may then be written as[12,13]

$$\frac{\partial W}{\partial t}+\frac{\partial F}{\partial \xi}+\frac{\partial G}{\partial \eta}=0 \tag{1}$$

where

$$W=\begin{bmatrix} \rho D \\ \rho Du \\ \rho Dv \\ eD \end{bmatrix},\quad F=\begin{bmatrix} \rho DU \\ \rho DUu+y_\eta p \\ \rho DUv-x_\eta p \\ DU(e+p) \end{bmatrix},\quad G=\begin{bmatrix} \rho DV \\ \rho DVu-y_\xi p \\ \rho DVv+x_\xi p \\ DV(e+p) \end{bmatrix}$$

and where t is the time, ρ the density, u and v the Cartesian velocity components, e the total energy per unit volume, and p the pressure given by

$$p=(\gamma-1)\,[e-(\rho/2)(u^2+v^2)]$$

and U and V are the contravariant velocity components given by

$$\begin{pmatrix} U \\ V \end{pmatrix} = \boldsymbol{J}^{-1} \begin{pmatrix} u \\ v \end{pmatrix}$$

To complete the definition of the mathematical problem, the initial and boundary conditions need to be specified. The initial conditions are trivial, since all of the dependent variables need to be specified and thus are set to some freestream condition.

The number of boundary conditions required at a given boundary may be obtained by considering the quasilinear form of the governing equation. This form is found by applying the chain rule to Eq. (1) to give

$$\frac{\partial \boldsymbol{W}}{\partial t}+\boldsymbol{A}\frac{\partial \boldsymbol{W}}{\partial \xi}+\boldsymbol{B}\frac{\partial \boldsymbol{W}}{\partial \eta}=0 \tag{2}$$

where

$$\boldsymbol{A}=\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{W}} \text{ and } \boldsymbol{B}=\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{W}}$$

are 4×4 matrices explicitly written out in Ref. 13. For a boundary condition analysis, ξ and η must be locally normal and tangent to the boundary. If ξ is increasing into the domain, then one boundary condition is required for each positive eigenvalue of $\boldsymbol{A}$[14]. This makes intuitive sense, since each positive eigenvalue corresponds to an incoming characteristic surface.

With the finite volume formulation of the present problem, all boundaries are aligned with either the ξ or η coordinate, as shown in Fig. 2. Thus, the number of boundary conditions are determined by the eigenvalues of either $\boldsymbol{A}$ or $\boldsymbol{B}$. For matrix $\boldsymbol{A}$, these are found to be[13]

$$\lambda_1=\lambda_2=U,\ \lambda_3=U+\frac{c}{D}\sqrt{y_\eta^2+x_\eta^2},\ \lambda_4=U-\frac{c}{D}\sqrt{y_\eta^2+x_\eta^2} \tag{3a}$$

and for matrix $\boldsymbol{B}$,

$$\lambda_1=\lambda_2=V,\ \lambda_3=V+\frac{c}{D}\sqrt{y_\xi^2+x_\xi^2},\ \lambda_4=V-\frac{c}{D}\sqrt{y_\xi^2+x_\xi^2} \tag{3b}$$

where c is the speed of sound given by $c^2=(\gamma p/\rho)$.

The boundary designated 1 in Fig. 2 corresponds to the airfoil blades. The perpendicular coordinate to this boundary is η and, therefore, the eigenvalues of concern are given by Eqs. (3b). Since the blades are no-flux boundaries, V is identically zero, resulting in only one positive eigenvalue. Thus, the only condition applied at the airfoil blade is the no-flux condition.

At the inflow and outflow boundaries discussed in this section (see also Sec. III), the relevant eigenvalues are given by Eqs. (3a). The number and type of conditions required at these boundaries is determined by both the sign of the normal component of the velocity and its magnitude relative to the local speed of sound. When the flow normal to the boundary is supersonic and directed into the domain (supersonic inflow), all of the eigenvalues are positive and, therefore, all of the dependent variables are set to the freestream condition. When the normal velocity is supersonic but directed out of the domain (supersonic outflow), all of the eigenvalues are positive and no boundary conditions are to be applied. At this point it should be noted that the specification of the conditions on the upstream and downstream boundaries in the present work is somewhat different than that usually found for internal flow problems and is suitable only when the upstream flow is subsonic and the casade not yet choked. The mass flow through the cascade is determined by specifying the freestream conditions far upstream of the blade row, rather than by specifying a downstream static pressure ratio. This latter procedure has recently been incorporated in the procedure, but will not be described here.

When the flow normal to the boundary is subsonic and directed into the domain (subsonic inflow), only three of the eigenvalues are positive. In this case, boundary conditions are applied in a fashion similar to that of Jameson et al.[15] Freestream conditions are applied on the three characteristic variables corresponding to the positive eigenvalues. These variables are computed by

$$\boldsymbol{V}=\boldsymbol{Q}^{-1}\boldsymbol{W}$$

$\boldsymbol{Q}^{-1}$ is defined by

$$\boldsymbol{Q}^{-1}\boldsymbol{A}\boldsymbol{Q}=\Lambda$$

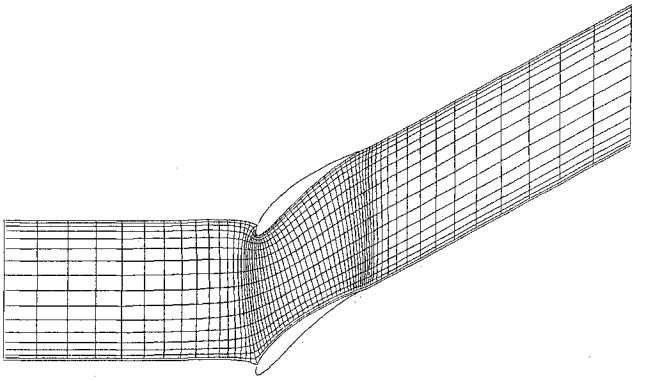

Fig. 1 Typical domain and grid topology.

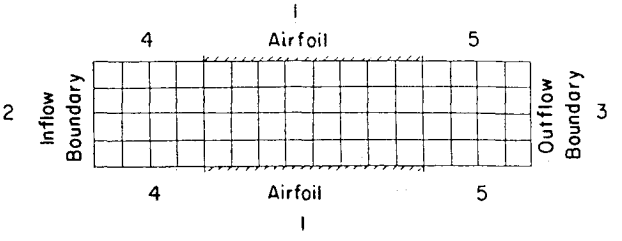

Fig. 2 Computational domain.

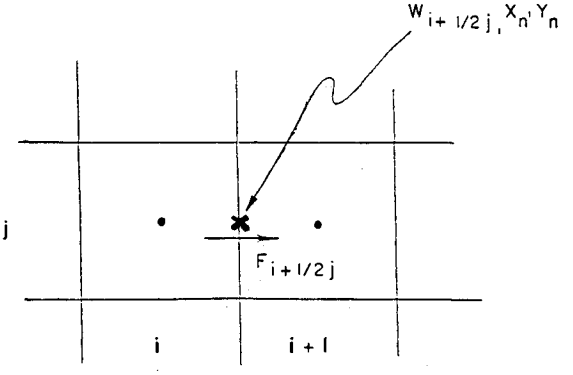

Fig. 3 Flux cells for residual calculation.

where Λ is the diagonal matrix of eigenvalues. This type of boundary treatment reduces the reflection of disturbances from the artificial inflow boundary and can greatly reduce the time required to reach steady state.[16,17] When the flow normal to the boundary is subsonic and directed out of the domain (subsonic outflow), only one boundary condition is appropriate. If the flow far downstream is known, then that condition could be applied on the incoming characteristic variable as in the inflow treatment. However, an infinite cascade of blades turns the flow by an amount that is unknown a priori, so the downstream conditions are unknown and some other method for specifying the boundary condition must be used. In the present work, since the total mass flux through the cascade is fixed by specifying the upstream conditions, this mass flux (together with the flow angle from the previous time step) is used to determine the downstream conditions to be specified.

The remaining boundaries are designated 4 and 5. Since these are periodic boundaries, the above eigenvalue analysis does not apply. The difference formulas applied at these boundaries are identical to those applied at the interior points. Information required from beyond the grid is retrieved from the corresponding point near the opposite boundary. Also, the Kutta condition is enforced implicitly by the numerical viscosity of the scheme, since only cases with relatively sharp trailing edges are considered here.

III. Discretization

Basic Implicit Formulation

Discretization of Eq. (1) yields

$$\frac{W_{ij}^{n+1}-W_{ij}^{n}}{\Delta t}+\mu(\delta_\xi F+\delta_\eta G)_{ij}^{n+1}+(1-\mu)(\delta_\xi F+\delta_\eta G)_{ij}^{n}=0 \quad (4)$$

where the superscripts denote the time level, the subscripts denote the spatial location δ_ξ and δ_η are the standard central difference operators, and μ is a parameter weighting the evaluation of spatial differences between the two time levels. Since F and G are nonlinear functions of W, Eq. (4) represents a nonlinear system of equations for W^{n+1} if $\mu \neq 0$. Linearization may be achieved by approximating F^{n+1} and G^{n+1} by use of Taylor series expansions about time level n,

$$F_{ij}^{n+1}\approx F_{ij}^{n}+\left.\frac{\partial F}{\partial t}\right|_{ij}^{n}\Delta t\approx F_{ij}^{n}+\left.\frac{\partial F}{\partial W}\right|_{ij}^{n}\left.\frac{\partial W}{\partial t}\right|_{ij}^{n}\Delta t$$

$$\approx F_{ij}^{n}+A_{ij}^{n}(W_{ij}^{n+1}-W_{ij}^{n})\approx F_{ij}^{n}+A_{ij}^{n}\Delta W_{ij}^{n}$$

$$G_{ij}^{n+1}\approx G_{ij}^{n}+B_{ij}^{n}\Delta W_{ij}^{n}$$

Substituting into Eq. (4) gives

$$\Delta W_{ij}^{n}+\Delta t\mu[\delta_\xi(A\Delta W)_{ij}^{n}+\delta_\eta(B\Delta W)_{ij}^{n}]=-\Delta t(\delta_\xi F+\delta_\eta G)_{ij}^{n} \quad (5)$$

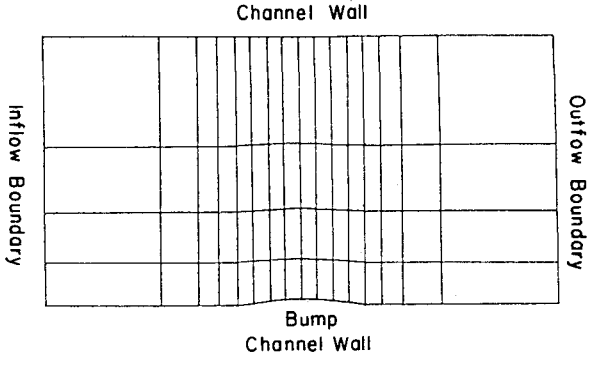

Fig. 4 Channel/bump geometry and grid.

By factoring ΔW from the left-hand side and identifying the spatial differences on the right-hand side as the "residual," the above equation may be rewritten as

$$[I+\Delta t\mu(\delta_\xi A+\delta_\eta B)_{ij}^{n}]\Delta W_{ij}^{n}=-\Delta t\cdot\text{residual} \quad (6)$$

Equation (6) is in "delta" form since ΔW^n is computed instead of W^{n+1}. The advantage of this form is that the steady-state solution (i.e., as $\Delta W\to 0$) satisfies the residual equation,

$$\delta_\xi F+\delta_\eta G=0$$

which has only spatial difference operators. The resulting steady-state solution is therefore independent of the time step used to reach it. This is not necessarily a property of the solution for a scheme in nondelta form.

Note that one should be careful that the meaning of Eq. (6) is explicitly shown in Eq. (5), i.e., the difference operators apply on products such as $A\Delta W$.

Equation (6) represents a linear sytem of equations for ΔW_{ij}^{n}. However, the size of this block system is equal to the total number of grid points (i,j) in the domain. Inversion of such large systems is costly and impractical and therefore an approximate factorization of the implicit operator is usually performed.

ADI Factoring

A popular factoring proposed independently by Briley and McDonald[2] and Beam and Warming[3-5] provides the basis for the alternating direction implicit (ADI) scheme. This approximate factoring is given by

$$[I+\Delta t\mu\delta_\xi A_{ij}^{n}][I+\Delta t\mu\delta_\eta B_{ij}^{n}]\Delta W_{ij}^{n}=-\Delta t\cdot\text{residual}$$

Application of each factor corresponds to a sweep through the domain, inverting block tridiagonal systems along horizontal or vertical lines. This still represents a considerable amount of work, especially for three-dimensional problems where a third factor arises and creates the need for three sweeps through the domain. Also, a linear von Neumann stability analysis shows that the ADI scheme is unconditionally unstable in delta form in three dimensions.

LU Factoring

To avoid the above problems, Jameson and Turkel[1] have proposed the LU factoring given by

$$[I+(\delta_\xi^- A_1+\delta_\eta^- B_1)_{ij}^{n}][I+(\delta_\xi^+ A_2+\delta_\eta^+ B_2)_{ij}^{n}]\Delta W_{ij}^{n}$$
$$=-\Delta t\cdot\text{residual} \quad (7)$$

where δ_ξ^-, δ_ξ^+, δ_η^-, and δ_η^+ are one-sided difference operators and where

$$A_1+A_2=\Delta t\mu A \quad (8a)$$

$$B_1+B_2=\Delta t\mu B \quad (8b)$$

Extension to three dimensions is straightforward by adding the terms $\delta_\zeta^- C_1$ to the first factor and $\delta_\zeta^+ C_2$ to the second factor. Note that only two factors remain, even for the three-dimensional case.

As long as Eqs. (8) are satisfied, A_1 through B_2 can be chosen arbitrarily since the product of the two factors in Eq. (7) is then consistent with the operator given in Eq. (6).[13] However, to ensure numerical stability, their exact form is carefully chosen to yield factors in Eq. (7) that remain diagonally dominant. The forms are based on those pro-

posed by Jameson and Turkel[1] and are given by

$$A_1 = \frac{\Delta t\mu}{2}(A+\rho_1 I) - \frac{I}{4},\quad A_2 = \frac{\Delta t\mu}{2}(A-\rho_1 I) + \frac{I}{4}$$

$$B_1 = \frac{\Delta t\mu}{2}(B+\rho_2 I) - \frac{I}{4},\quad B_2 = \frac{\Delta t\mu}{2}(B-\rho_2 I) + \frac{I}{4}$$

where I is the 4×4 identity matrix and ρ_1 and ρ_2 are, respectively, the maximum absolute values of the eigenvalues of A and B given by Eqs. (3).

Numerical Implementation

Implementation of the LU scheme is in three steps. The first two, residual calculation and L sweep, are carried out concurrently and represented by

$$[I + (\delta_\xi^- A_1 + \delta_\eta^- B_1)_{ij}^n]\Delta V_{ij}^n = -\Delta t \cdot \text{residual} \qquad (9)$$

If the dependent variables are stored at cell centers, the residual calculation may be considered as a flux balance over the mesh cells. Figure 3 shows that the dependent variables at the points (i,j) and $(i+1,j)$ are averaged to give W at $(i+\frac{1}{2},j)$. The metric terms x_η and y_η can also be calculated at that point by a central difference involving the given values of x and y at the points $(i+\frac{1}{2},j+\frac{1}{2})$ and $(i+\frac{1}{2},j-\frac{1}{2})$. With this information, the flux F may be computed at the flux face at $i+\frac{1}{2}$. The fluxes at the other three faces of the cell centered at (i,j) can be computed similarly. The residual calculation is then given by central differences.

Since the flux at a given face is always computed in the same fashion, this scheme has the property of numerical conservation. This is a desirable property for problems with shocks (discontinuities), since the proper weak solution is captured (i.e., correct shock strength and location).

Once the residual is calculated at a point, the matrices A_1 and B_1 must be computed. To save CPU time, it was found that these matrices need not be computed at each time step. Instead, they are recomputed and stored every 10 time steps. This treatment does not affect the steady-state solution, but does reduce the accuracy of the time evolution.

Since only one-sided difference operators are used in the left side of Eq. (9), it represents a system that is lower block diagonal. (Although this is true in general, the specific cascade problem with periodic boundaries results in cyclic block matrices. The solution of such a system is described in the Appendix.) The inversion of such systems is performed simply by marching point to point through the domain. At each point (i,j), only the local diagonal block is inverted to give ΔV_{ij}^n.

The third step is the U sweep given by

$$[I + (\delta_\xi^+ A_2 + \delta_\eta^+ B_2)_{ij}^n]\Delta W_{ij}^n = \Delta V_{ij}^n \qquad (10)$$

This step is very similar to the L sweep. The right-hand side is known from the previous calculation, while the left-hand side represents an upper diagonal block system. Inversion is again a point-by-point march through the domain, but now it is in the opposite direction than that taken by the L sweep. (For the periodic problem, this system is also cyclic and must be treated as shown in the Appendix.) The completion of the U sweep is also the completion of one time step.

Boundary Treatment

Since the dependent variables are stored at cell centers and the boundaries are aligned with grid lines, the boundary conditions are not applied directly on the computed values of W, but instead indirectly in the solution procedure. For the residual calculation, fluxes on the boundary faces must be defined since the regular evaluation procedure breaks down. These values are obtained as a linear combination of known boundary conditions and extrapolation of those variables for which there is no boundary condition. The treatment at a no-flux boundary is simple since most of the flux terms have the normal contravariant velocity as a factor that is indentically zero there.

After the residuals are calculated, the L sweep must be started. When applying the operator given in Eq. (9) at points adjacent to the left and bottom boundaries, ΔV^n at the boundary is needed. (For the cascade case, this is true only at the left boundary and the bottom boundary at the airfoil. The periodic bottom boundary does not require auxiliary values of ΔV since the needed values are those adjacent to the corresponding top boundary, thus resulting in the cyclic system.) These auxiliary values are obtained by a linear combination of ΔV^n at the adjacent cell centers and ΔW^{n-1} at the boundary. The above combination with ΔW is possible because ΔV and ΔW are consistent with each other.[12] Once the L sweep is initiated in this fashion, no further special treatment is required.

After the L sweep is finished, the U sweep must be started with auxiliary values for ΔW on the top and right boundaries (except at periodic boundaries). Again, these extra values are obtained by linear combinations of given boundary conditions and extrapolation of ΔW from the interior.

Explicit Dissipation

As is usually necessary, explicit dissipation terms have been added to the residual in Eq. (7) to damp wiggles or recouple solutions that have become decoupled. These terms are added in both the ξ and η directions, but only the term in the ξ direction is described here. It is composed of two parts, the first given by

$$-\Delta t\alpha_x \delta_\xi D \delta_\xi (W/D) \qquad (11)$$

where α_x is a parameter that controls the amount of this dissipation term. Equation (11) is a second difference term that uses values of W at the points $i-1$, i, and $i+1$ and therefore recouples the solution. Factoring out the D from W allows the freestream to be an exact solution of the difference equations. Keeping the D inside the first difference operator retains the conservative form.

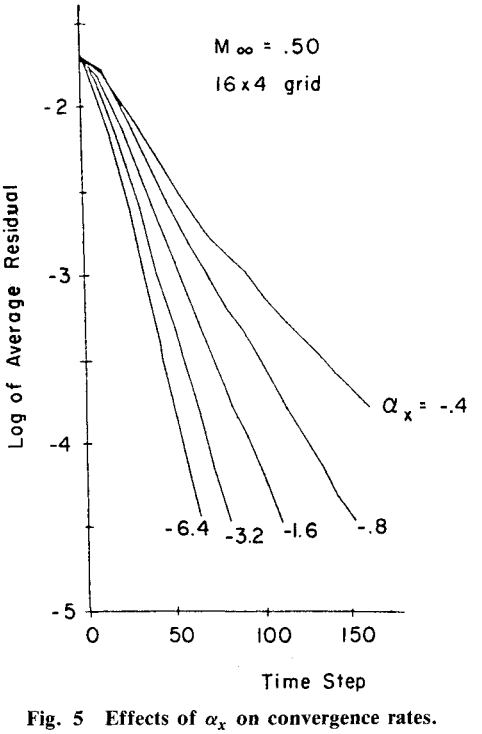

Fig. 5 Effects of α_x on convergence rates.

This dissipation term is of second order compared to the residual and therefore does not upset the formal accuracy of the scheme. However, the addition of this term does increase the truncation error and introduces an asymmetry into the numerical solutions. The results show this asymmetry and how it quickly tends to zero as the mesh is refined. Also shown in the results is the strong dependence of the convergence rate on the parameter α_x.

Although the second-order dissipation was found to be sufficient for simple problems, it was necessary to add additional dissipation for problems with stagnation points. This additional dissipation is given by

$$-\Delta t \delta_\xi k_x D \delta_{\xi\xi\xi}(W/D) \tag{12}$$

where

$$k_x = \beta_x (N/64)^2 |\delta_{\xi\xi} e| \tag{13}$$

and where β_x is a constant and N the number of cells in the ξ direction. Again, the special form of Eq. (12) allows the freestream to be an exact numerical solution. The parameter k_x, controlling the amount of dissipation, is now a function of the solution and must be kept within the first δ_ξ operator to retain a conservative form. Equation (13) shows that k_x is large in regions with high gradients of flow (for example, at the leading edge of the airfoil where the problems had appeared) and small in regions of smooth flow. The factor of N^2 is included to keep k_x of order one instead of order N^{-2} (due to the $\delta_{\xi\xi}$ term). The overall order of Eq. (12) compared to the residual is four. Thus, the effects on the solution of this dissipation term tend to zero much faster than the second order dissipation as the grid is refined.

Similar terms are added in the η direction and contain the parameters α_y and β_y.

IV. Results

Initial calculations were carried out for the relatively simple problem of flow through a channel with a bump on one wall. The geometry and a 16×4 grid are shown in Fig. 4. All results presented set the implicit parameter μ to 1 for maximum convergence. Figure 5 shows the effect of the dissipation parameter α_x [Eq. (11)] on the convergence rate, indicating that large absolute values are desirable. The vertical axis in the convergence plots is the logarithm of the average residual. This average is defined as the root mean square value over the four equations of the average absolute value of the local residuals. However, the dissipation term does add an asymmetrical error, which is shown in the pressure distribution plot of the coarse grid in Fig. 6. The theoretical solution is symmetric because of the symmetric geometry and completely subsonic flow. This is quickly approached by refining the grid, demonstrating the second-order property of the dissipation term. The vanishing asymmetry also suggests that total pressure losses due to the numerical dissipation will become small, although this was not computed explicitly in our results.

For the same channel flow, Fig. 7 shows the trend in convergence rate on three different grids, each having twice as many cells in each direction as the previous grid. The initial condition for the coarsest grid is the freestream flow, while the initial conditions for the other grids are interpolated solutions from the previous coarser grid. Results are given

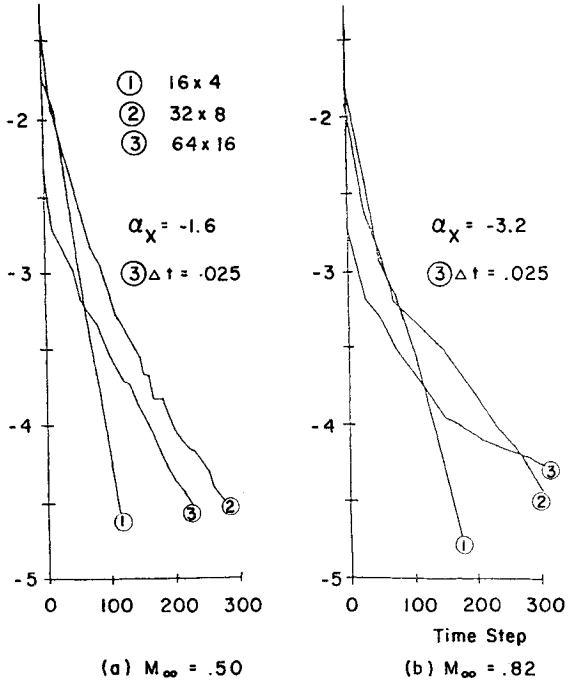

Fig. 7 Effects of grid refinement on convergence.

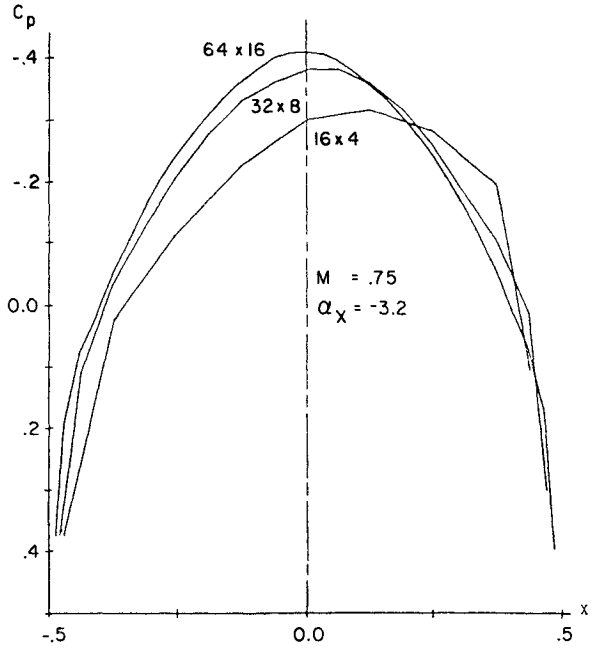

Fig. 6 The $\mathcal{O}(N^{-2})$ effect of dissipation.

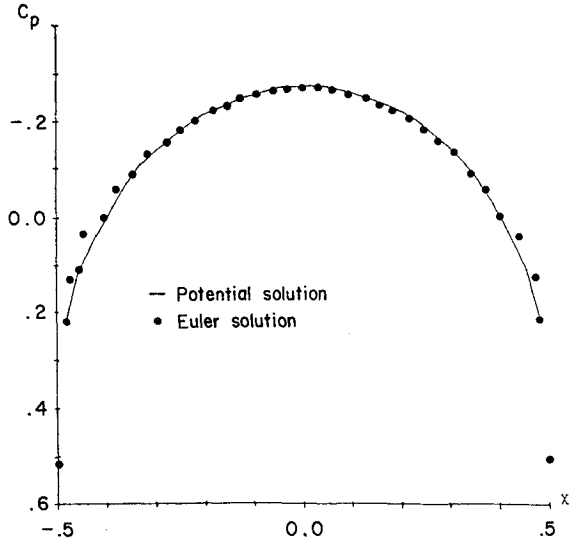

Fig. 8 Converged pressure distribution for $M_\infty = 0.5$.

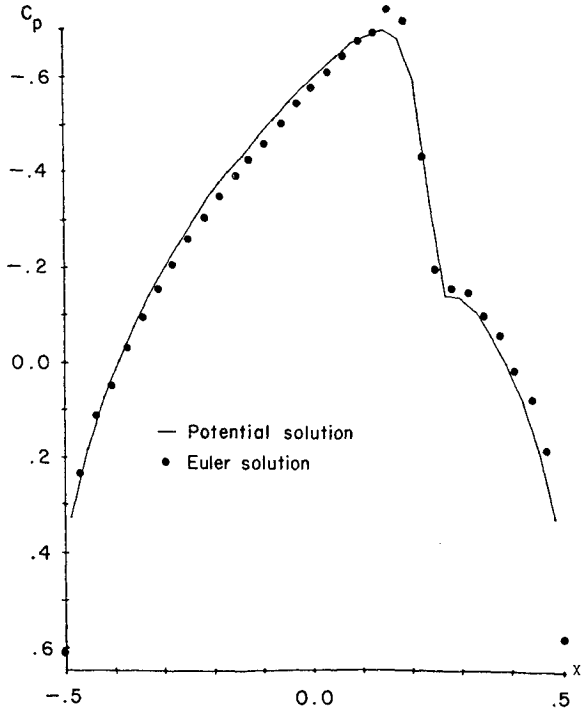

Fig. 9 Converged pressure distribution for $M_\infty = 0.82$.

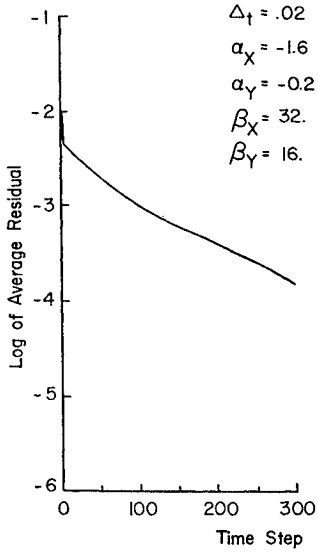

Fig. 10 Convergence history for the Gostelow case.

for both subsonic and transonic cases. The converged pressure distributions are plotted in Figs. 8 and 9 and good agreement is achieved with solutions of the potential problem.[19]

The computer code was then modified to compute cascade problems (i.e., to incorporate treatment of the periodic boundary) and tested for the Merchant and Collar airfoil given by Gostelow[20] and shown in Fig. 1. This represents a much more difficult problem due to the large stagger angle of 37.5 deg and large inflow angle of 53.5 deg. Also, due to the asymmetry, the flow is turned through an angle of approximately 23.5 deg. This case was run with an inflow Mach number of 0.5 where the flow remained completely subsonic. The Mach number at the outflow boundary was found to be approximately 0.33. Convergence on the 64×16 grid is shown in Fig. 10. As expected, the rate is poorer than for a nonstaggered subsonic case, but is only slightly slower than for symmetric transonic cases.

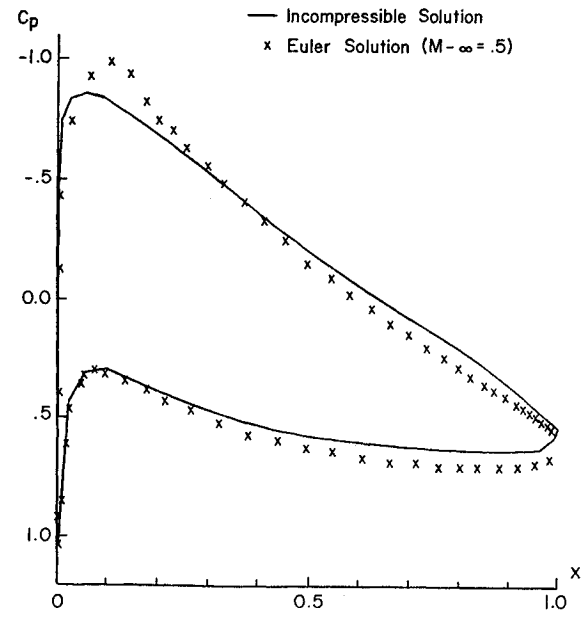

Fig. 11 Pressure distribution for the Gostelow case.

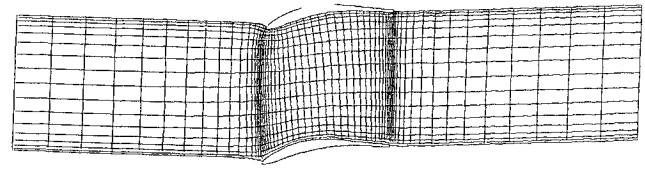

Fig. 12 Sanz shock-free cascade.

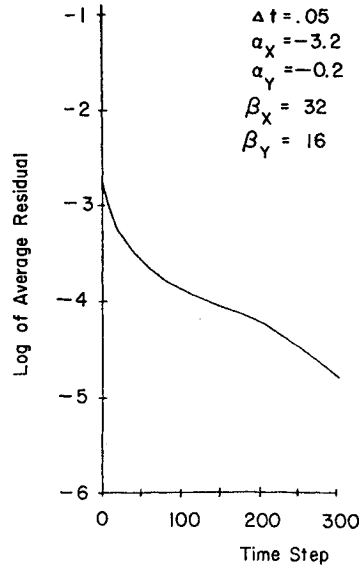

Fig. 13 Convergence history for the Sanz case.

Figure 11 shows the pressure distributions on the upper and lower surfaces of the airfoil. This solution is compared to the exact analytical solution of the incompressible problem given by Gostelow.[20] Agreement is close and the difference between the two solutions is qualitatively correct for a compressibility correction.

The final test case is that of a shock-free supercritical cascade derived from the Korn hodograph code by Sanz.[21,22] The geometry and 64×16 grid are shown in Fig. 12. A

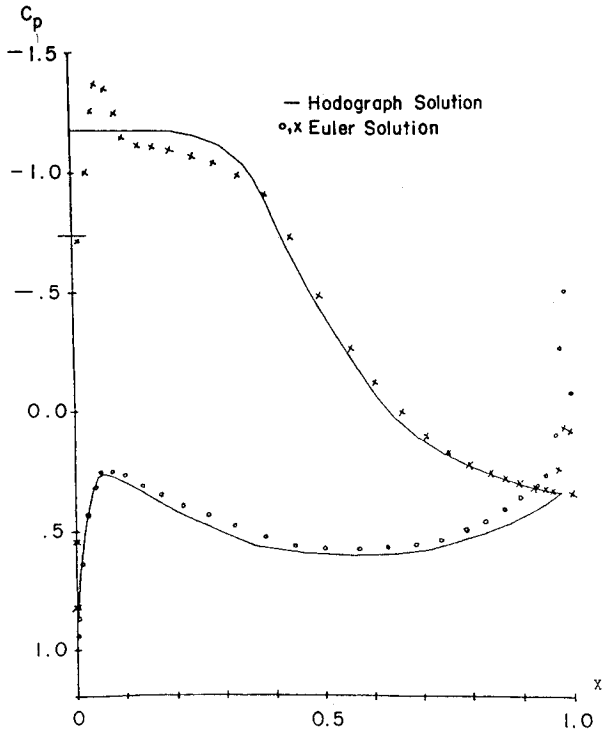

Fig. 14 Pressure distribution for the Sanz case.

modification of the geometry was performed at the trailing edge since the hodograph solution uses a viscous flow approximation and produces an open trailing edge. The present grid closes the airfoil by inserting a rounded trailing edge.

The upstream conditions for this problem are an inflow angle of 30.8 deg and a Mach number of 0.711. The convergence history, shown in Fig. 13, shows that the smaller stagger angle and moderate inflow angle do not affect convergence rate.

The pressure distribution is shown in Fig. 14 and is compared to the exact hodograph solution.[21,22] Agreement is quite good except for an overshoot on the upper surface near the leading edge and discrepancy at the trailing edge. The leading-edge problem is probably due to insufficient resolution there (since the exact solution shows an extremely large gradient), aggravated by the H-type grid topology. Different trailing-edge solutions are to be expected because of the geometrical changes. With the high curvature introduced there, an acceleration of the flow is expected and thus a decrease in the pressure. A fine grid resolution at the trailing edge has captured this detail. Even with these local problems, the global solution is accurate, yielding the shock-free flow.

V. Conclusions

It has been shown that the LU scheme is capable of computing accurate solutions to complex cascade problems. This is achieved with good convergence rates on nonuniform, nonorthogonal, and unaligned (to the flow direction) grids.

However, the ultimate measure of efficiency must be based on the required CPU time. The final form of the LU code (i.e., with all dissipation terms and cyclic solver) takes 0.00092 s/cell/time step on an IBM 370/168 using the FORTHX OPT(3) compiler. Depending on the particular two-dimensional code of the ADI scheme, it may take from 0.00037 s/cell/time step[23] to 0.00063 s/point/time step[24] on a CDC 7600 computer. Assuming the CDC 7600 is faster than the IBM 370/168 by a factor of five, the LU scheme is 2.0-3.4 times faster than the ADI scheme. These are promising results, since the real advantage of the LU scheme is realized for three-dimensional problems.

Appendix

Cyclic Solver

Application of the L operator [Eq. (9)] or U operator [Eq. (10)], respectively, at cell centers adjacent to the bottom or top periodic boundaries results in matrix systems that are cyclic. That is, the lower or upper block structure now contains a nonzero block in the opposite corner. This single nonzero block upsets the simple marching algorithm used to invert the truly lower or upper system. However, it is not necessary to solve the cyclic system as if it were dense. An efficient method has been developed which is based on the Ahlberg-Nilson-Walsh algorithm,[18] which was designed to solve scalar cyclic tridiagonal systems.

Consider the lower-block-system,

$$\begin{bmatrix} A_1 & & & & B_1 \\ B_2 & A_2 & & 0 & \\ & B_3 & A_3 & & \\ & & \ddots & \ddots & \\ & 0 & & \ddots & \\ & & & B_n & A_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \cdot \\ \cdot \\ \cdot \\ A_n \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \cdot \\ \cdot \\ \cdot \\ C_n \end{bmatrix} \tag{A1}$$

where the A_i and B_i are arbitrary blocks and the X_i and C_i arbitrary vectors. Let E be the lower block bidiagonal matrix that results when the last block column and last block row are deleted from the matrix in Eq. (A1). Also define,

$$f = \begin{bmatrix} B_1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \hat{x} = \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_{n-1} \end{bmatrix} \quad \hat{c} = \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ \cdot \\ C_{n-1} \end{bmatrix}$$

Then the system in Eq. (A1) may be rewritten as

$$E\hat{x} + fX_n = \hat{c} \tag{A2a}$$

$$B_n X_{n-1} + A_n X_n = C_n \tag{A2b}$$

The above system is solved in four steps. The first two are carried out simultaneously for computational efficiency and are given by the solution of

$$E\hat{y} = \hat{c} \text{ and } Ee^{-1} = h$$

where

$$h = \begin{bmatrix} I \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

and where I is the identity matrix of the same size as the blocks. It is clear that e^{-1} is the first block column of E^{-1}. Because of the lower bidiagonal form of E, the inversion is

carried out efficiently by the usual marching technique. The third step is to solve for X_n by noting that Eq. (A2a) implies

$$\hat{x} = \hat{y} - e^{-1} B_1 X_n \tag{A3}$$

which is used in Eq. (A2b) to give

$$[-B_n e_{n-1}^{-1} B_1 + A_n] X_n = C_n$$

Finally, once X_n is known, Eq. (A3) is used to evaluate $\hat{x}$. An operation count shows that this procedure takes roughly twice as much time as the lower block bidiagonal system.

Acknowledgment

This work was supported by the NASA Lewis Research Center under Grant NAG 3-19.

References

[1]Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, Oct. 1981, pp. 385-397.

[2]Briley, W. R. and McDonald, H., "Solution of the Three-Dimensional Compressible Navier Stokes Equations by an Implicit Technique," *Proceedings of 4th International Conference on Numerical Methods in Fluid Dynamics*, June 1974, *Lecture Notes in Physics*, Vol. 35, Springer Verlag, New York, 1975, pp. 105-110.

[3]Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic System in Conservation Law Form," *Journal of Computational Physics*, Vol. 22, 1976, pp. 87-110.

[4]Warming, R. F. and Beam, R. M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," *SIAM-AMS Proceedings*, Vol. 11, edited by H. B. Keller, 1978, pp. 85-129.

[5]Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

[6]Gopalakrishnan, S. and Bozzola, R., "A Numerical Technique for the Calculation of Transonic Flow in Turbomachinery," ASME Paper 71-GT-42, 1971.

[7]McDonald, P. W., "The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades," ASME Paper 71-GT-899, 1971.

[8]Delaney, R. A. and Kavanagh, P., "Transonic Flow Analysis in Axial Flow Turbomachinery Cascades by a Time-Dependent Method of Characteristics," *Journal of Engineering for Power, Transactions of ASME*, Vol. 98, 1976, pp. 356-364.

[9]Denton, D. J., "A Time-Marching Method for Two- and Three-Dimensional Blade to Blade Flows," ARC R&M 3775, 1974.

[10]Denton, D. J., "Throughflow Calculations for Transonic Axial Flow Turbines," *Journal of Engineering for Power, Transactions of ASME*, Vol. 100, 1978, pp. 212-218.

[11]Habashi, W. G., "Numerical Methods of Turbomachinery," *Recent Advances in Numerical Methods in Fluids*, Vol. 1, edited by C. Taylor and K. Morgan, Pineridge Press, Swansea, UK, 1980, pp. 245-286.

[12]Pulliam, T. H. and Steger, J. L., "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, Feb. 1980, pp. 159-167.

[13]Buratynski, E. K., "A Lower-Upper Factored Implicit Scheme for the Numerical Solution of the Euler Equations Applied to Arbitrary Cascades," Ph.D. Thesis, Cornell University, Ithaca, NY, May 1983.

[14]Kreiss, H. and Oliger, J., "Methods for the Approximate Solution of Time Dependent Problems," Global Atmospheric Research Programme, Pub. Ser. 10, Feb. 1973.

[15]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, June 1981.

[16]Hedstrom, G. W., "Nonreflecting Boundary Conditions for Nonlinear Hyperbolic Systems," *Journal of Computational Physics*, Vol. 30, 1979, pp. 222-237.

[17]Rudy, D. H. and Strikwerda, J. C., "A Nonreflecting Outflow Boundary Condition for Subsonic Navier-Stokes Calculations," *Journal of Computational Physics*, Vol. 36, 1980, pp. 55-70.

[18]Temperton, C., "Algorithm for the Solution of Cyclic Tridiagonal Systems," *Journal of Computational Physics*, Vol. 19, 1975, pp. 317-323.

[19]Jameson, A., Caughey, D. A., Jou, W. H., Steinhoff, J., and Pelz, R., "Accelerated Finite Volume Calculation of Transonic Potential Flows," *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves*, edited by A. Rizzi and H. Viviand, Vieweg & Sohn, Braunschweig, 1981, pp. 11-27.

[20]Gostelow, J. P., "Potential Flow Through Cascades—A Comparison Between Exact and Approximate Solutions," *Aeronautical Research Council Current Papers*, No. 807, 1965.

[21]Steger, J. L., Pulliam, T. H., and Chima, R. V., "An Implicit Finite Difference Code for Inviscid and Viscous Cascade Flow," AIAA Paper 80-1427, July 1980.

[22]Dulikravich, D. S. and Caughey, D. A., "Finite Volume Calculations of Transonic Potential Flow Through Rotors and Fans," Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, Rept. FDA-80-03, March 1980.

[23]Chaussee, D. S. and Pulliam, T. H., "Two-Dimensional Inlet Simulation Using a Diagonal Implicit Algorithm," *AIAA Journal*, Vol. 19, Feb. 1981, pp. 153-159.

[24]Chyu, W. J., Davis, S. S., and Chang, K. S., "Calculation of Unsteady Transonic Flow Over an Airfoil," *AIAA Journal*, Vol. 19, June 1981, pp. 684-690.