Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations

Antony Jameson* and Seokkwan Yoon†
Princeton University, Princeton, New Jersey

A lower-upper implicit scheme is developed for the unsteady Euler equations. It is proved that a symmetric successive overrelaxation method for solving the unfactored implicit scheme is equivalent to a modified form of the lower-upper implicit scheme. A multigrid method is combined with the lower-upper implicit scheme to produce a rapidly convergent algorithm for calculating steady-state solutions of the Euler equations.

I. Introduction

THE first Euler code to gain acceptance for the prediction of transonic flows was written by Magnus and Yoshihara.[1] They used the Lax-Wendroff scheme, which can be derived from a Taylor series expansion. The popular explicit scheme of MacCormack[2] is a variation of the two-step Lax-Wendroff scheme, which removes the necessity of computing unknowns at the midpoints. A disadvantage of these schemes for steady-state calculations is that the computed steady state is not unique, depending on the time step. This suggests the desirability of separating the space and time discretization procedures, using the so-called method of lines. This procedure was adopted in the scheme proposed by Jameson, Schmidt, and Turkel[3] in conjunction with the Runge-Kutta time-stepping scheme.

Since the time step of an explicit scheme is limited by the Courant-Friedrichs-Lewy (CFL) condition, which requires that the region of dependence of the difference scheme must at least include the region of dependence of the differential equation, it should be possible to reduce the number of time steps by the introduction of an implicit scheme. Implicit schemes using approximate factorization are attractive because they can easily be extended to viscous flow analysis.

The idea of using multigrid techniques for the acceleration of convergence has been known for decades. However, reports of success were rare. Recently, Jameson and Yoon[4] showed that the alternating direction implicit (ADI) scheme could be improved to achieve the expected efficiency of the multigrid method. Although the ADI scheme has been useful in two dimensions, it is well known that the corresponding scheme in delta form is unstable in three dimensions. An alternative implicit scheme that is stable in any number of space dimensions is based on lower-upper (LU) factorization.[5] In this paper, an LU implicit scheme in conjunction with a multigrid method is developed. It is also shown that a symmetric successive overrelaxation (SSOR) method for solving the unfactored implicit scheme is a variation of the LU implicit scheme.

Presented as Paper 86-0105 at the AIAA 24th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1986, received Feb. 19, 1986; revision received Nov. 23, 1986.

* Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.

† Graduate Research Assistant, Department of Mechanical and Aerospace Engineering; presently at Sverdrup Technology, Inc., Cleveland, OH. Member AIAA.

II. Governing Equations

The Euler equations are obtained from the Navier-Stokes equations by neglecting viscous terms. Let ρ, u, v, E, H, and p be the density, Cartesian velocity components, total energy, total enthalpy, and pressure, and let x and y be Cartesian coordinates. Then for a two-dimensional flow, these equations can be written as

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{1}$$

where W is the vector of dependent variables and F and G are convective flux vectors:

$$W = (\rho, \rho u, \rho v, \rho E)^T$$

$$F = (\rho u, \rho u^2 + p, \rho vu, (\rho E + p)u)^T$$

$$G = (\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v)^T \tag{2}$$

The pressure is obtained from the equation of state.

$$p = \rho(\gamma - 1)\{E - \tfrac{1}{2}(u^2 + v^2)\} \tag{3}$$

These equations are to be solved for a steady-state $\partial W/\partial t = 0$, where t denotes time.

III. Finite-Volume Scheme

A convenient way to assure a steady-state solution independent of the time step is to separate the space and time discretization procedures. The semidiscrete finite-volume scheme is derived by approximating the integral form of the equations.[3] The use of a finite-volume method for space discretization allows one to handle arbitrary geometries and helps to avoid problems with metric singularities that are usually associated with finite-difference methods. The scheme reduces to a central-difference scheme on a Cartesian grid and is second-order accurate in space, provided the mesh is smooth enough. It also has the property that uniform flow is an exact solution of the difference equations.

In a transonic flow calculation by a central-difference scheme, wiggles appear in the neighborhood of shock waves where pressure gradient is severe. In order to suppress the tendency for spurious odd and even point oscillations, and to prevent unsightly overshoots near shock waves, the scheme is augmented by artificial dissipative terms,[3] which introduce an upwind bias. These terms consist of blended second and fourth differences. The fourth differences provide background dissipation throughout the domain. In the neighborhood of a

shock wave, the second differences become the dominant dissipative terms through the use of an adaptive coefficient.

IV. LU Implicit Scheme

An obvious way to accelerate convergence to a steady state is to increase the size of the time step. When the time step limit imposed by an explicit stability bound is much less than that imposed by the accuracy bound, implicit schemes are to be preferred.

A prototype implicit scheme for a system of nonlinear hyperbolic equations such as the Euler equations can be formulated as

$$W^{n+1} = W^n - \beta\Delta t\{ D_x F(W^{n+1}) + D_y G(W^{n+1})\}$$
$$-(1-\beta)\,\Delta t\{ D_x F(W^n) + D_y G(W^n)\} \tag{4}$$

where D_x and D_y are difference operators that approximate $\partial/\partial x$ and $\partial/\partial y$. Here n denotes the time level. In this form, the scheme is too expensive since it calls for the solution of coupled nonlinear equations at each time step. Let the Jacobian matrices be

$$A = \frac{\partial F}{\partial W} \qquad B = \frac{\partial G}{\partial W} \tag{5}$$

and let the correction be

$$\delta W = W^{n+1} - W^n$$

The scheme can be linearized by setting

$$F(W^{n+1}) = F(W^n) + A\,\delta W + \mathcal{O}(\|\delta W\|^2)$$
$$G(W^{n+1}) = G(W^n) + B\,\delta W + \mathcal{O}(\|\delta W\|^2)$$

and dropping terms of the second and higher order. This yields

$$\{ I + \beta\Delta t(D_x A + D_y B)\}\,\delta W + \Delta t R = 0 \tag{6}$$

where R is the residual,

$$R = D_x F(W^n) + D_y G(W^n)$$

If $\beta = \frac{1}{2}$, the scheme remains second-order accurate in time, while for other values of β, the time accuracy drops to first order.

The unfactored implicit scheme (6) produces a large block banded matrix, which is very costly to invert and requires huge storage. If $\beta = 1$, the scheme reduces to a Newton iteration in the limit $\Delta t \to \infty$.

$$(D_x A + D_y B)\,\delta W + R = 0 \tag{7}$$

Because of the rapid growth of the operation count with the number of mesh cells, the Newton method appears to be impractical. One can solve the system indirectly using a relaxation algorithm. If one uses a fixed number of iterations in each cycle, the scheme is not necessarily quadratically convergent and becomes an alternative to approximate factorization. It is desirable that the matrix should be diagonally dominant to assure the convergence of a relaxation method. This can be achieved by flux splitting at the expense of a substantial increase in the computational work per cycle.

The scheme is reduced to an ADI scheme by replacing the operator of Eq. (6) by a product of 2 one-dimensional operators.

$$(I + \beta\Delta t D_x A)(I + \beta\Delta t D_y B)\,\delta W + \Delta t R = 0 \tag{8}$$

This scheme requires relatively expensive tridiagonal or pentadiagonal inversions. When marching to a steady state using large time steps, one wants to ensure that the numerical solution is independent of the size of the time steps. A simple way to do this is to solve for the change in the solution at each time step. In the two-dimensional case, the ADI schemes that solve for either W^{n+1} or δW^n are equivalent. In the three-dimensional case, the ADI scheme is unconditionally stable if one solves for W^{n+1}, but the steady-state solution depends on Δt. However, if one solves for δW^n to produce a steady-state solution independent of Δt, then the ADI scheme is unstable. Although artificial dissipation has some stabilizing effect, large amounts of dissipation can impair the accuracy. In any event, the terms of order $(\Delta t)^3$ may reduce the rate of convergence. Perhaps the biggest drawback of the ADI scheme is the poor high-frequency damping characteristic in three dimensions.

Jameson and Turkel[5] proposed the idea of a lower-upper (LU) factored implicit scheme that is unconditionally stable in any number of space dimensions and also yields a steady-state solution that is independent of Δt. Since the LU implicit scheme needs only two factors, it has terms at most of order $(\Delta t)^2$, independent of the number of dimensions. A variation of this scheme was used by Buratynski and Caughey to calculate cascade flows on an H-mesh.[6] Steger and Warming arrived at a similar idea based on flux vector splitting.[7] An LU implicit scheme can be derived by the factorization

$$\{ I + \beta\Delta t(D_x^- A^+ + D_y^- B^+)\}$$
$$\{ I + \beta\Delta t(D_x^+ A^- + D_y^+ B^-)\}\,\delta W + \Delta t R = 0 \tag{9}$$

where D_x^- and D_y^- are backward-difference operators and D_x^+ and D_y^+ are forward-difference operators. Here, two-point operators are used for steady flow calculations. A^+, A^-, B^+, and B^- are constructed so that the eigenvalues of plus matrices are nonnegative and those of minus matrices are nonpositive.[8] The development of these matrices is extremely important for the success of LU-type schemes. One possibility, used in this work, is

$$A^+ = \tfrac{1}{2}(A + r_A I),\ A^- = \tfrac{1}{2}(A - r_A I)$$
$$B^+ = \tfrac{1}{2}(B + r_B I),\ B^- = \tfrac{1}{2}(B - r_B I) \tag{10}$$

where

$$r_A \geq \max(|\lambda_A|), \qquad r_B \geq \max(|\lambda_B|) \tag{11}$$

Here, λ_A and λ_B represent eigenvalues of Jacobian matrices. Equation (9) can be inverted in two steps as

$$\{ I + \beta\Delta t(D_x^- A^+ + D_y^- B^+)\}\,\delta W^* = -\Delta t R$$
$$\{ I + \beta\Delta t(D_x^+ A^- + D_y^+ B^-)\}\,\delta W = \delta W^* \tag{12}$$

The scheme is unconditionally stable for the linear scalar case if $\beta \geq \frac{1}{2}$. Equation (12) needs the inversion of sparse triangular matrices that can be done efficiently without using large storage.

For a given boundary-conforming mesh and explicit boundary conditions, simple sweeps for system (9) are likely to give inaccurate solutions because of the asymmetrical boundary conditions. A useful strategy to avoid this problem is to enforce symmetry by splitting the computational domain into upper and lower half-planes. This gives slightly better results than alternating sweeps. While system (9) is used for the upper half-plane, the system for the lower half-plane is given by

$$\{ I + \beta\Delta t(D_x^+ A^- + D_y^- B^+)\}$$
$$\{ I + \beta\Delta t(D_x^- A^+ + D_y^+ B^-)\}\,\delta W + \Delta t R = 0 \tag{13}$$

V. LU-SSOR Implicit Scheme

The linearized implicit scheme

$$\left\{ I + \beta\Delta t\left(D_x^- A^+ + D_x^+ A^- + D_y^- B^+ + D_y^+ B^- \right) \right\} \delta W + \Delta t R = 0 \tag{14}$$

can be written as

$$\begin{aligned} &\delta W_{ij} + \beta\Delta t\left(A_{ij}^+ \delta W_{ij} - A_{i-1,j}^+ \delta W_{i-1,j} \right. \\ &\quad \left. + A_{i+1,j}^- \delta W_{i+1,j} - A_{ij}^- \delta W_{ij} \right) \\ &\quad + \beta\Delta t\left(B_{ij}^+ \delta W_{ij} - B_{i,j-1}^+ \delta W_{i,j-1} \right. \\ &\quad \left. + B_{i,j+1}^- \delta W_{i,j+1} - B_{ij}^- \delta W_{ij} \right) + \Delta t R_{ij} = 0 \end{aligned} \tag{15}$$

By simulating it with backward and forward relaxation sweeps, we obtain the symmetric successive overrelaxation (SSOR) method, which can be written in two steps as

$$\begin{aligned} &\delta W_{ij}^* + \beta\Delta t\left(A_{ij}^+ - A_{ij}^- \right) \delta W_{ij}^* + \beta\Delta t A_{i+1,j}^- \delta W_{i+1,j}^* \\ &\quad + \beta\Delta t\left(B_{ij}^+ - B_{ij}^- \right) \delta W_{ij}^* + \beta\Delta t B_{i,j+1}^- \delta W_{i,j+1}^* \\ &\quad + \Delta t R_{ij} = 0 \end{aligned} \tag{16}$$

followed by

$$\begin{aligned} &\delta W_{ij} + \beta\Delta t\left(A_{ij}^+ - A_{ij}^- \right) \delta W_{ij} - \beta\Delta t A_{i-1,j}^+ \delta W_{i-1,j} \\ &\quad + \beta\Delta t A_{i+1,j}^- \delta W_{i+1,j}^* + \beta\Delta t\left(B_{ij}^+ - B_{ij}^- \right) \delta W_{ij} \\ &\quad - \beta\Delta t B_{i,j-1}^+ \delta W_{i,j-1} \\ &\quad + \beta\Delta t B_{i,j+1}^- \delta W_{i,j+1}^* + \Delta t R_{ij} = 0 \end{aligned} \tag{17}$$

Subtract Eq. (16) from Eq. (17) to get

$$\begin{aligned} &\delta W_{ij} + \beta\Delta t\left(A_{ij}^+ - A_{ij}^- \right) \delta W_{ij} - \beta\Delta t A_{i-1,j}^+ \delta W_{i-1,j} \\ &\quad + \beta\Delta t\left(B_{ij}^+ - B_{ij}^- \right) \delta W_{ij} - \beta\Delta t B_{i,j-1}^+ \delta W_{i,j-1} \\ &\quad = \delta W_{ij}^* + \beta\Delta t\left(A_{ij}^+ - A_{ij}^- \right) \delta W_{ij}^* + \beta\Delta t\left(B_{ij}^+ - B_{ij}^- \right) \delta W_{ij}^* \end{aligned} \tag{18}$$

This may be written as

$$\left\{ I + \beta\Delta t\left(D_x^- A^+ + D_y^- B^+ - A^- - B^- \right) \right\} \delta W = \left\{ I + \beta\Delta t\left(A^+ + B^+ - A^- - B^- \right) \right\} \delta W^* \tag{19}$$

where

$$\delta W^* = \left\{ I + \beta\Delta t\left(D_x^+ A^- + D_y^+ B^- + A^+ + B^+ \right) \right\}^{-1} \left(-\Delta t R \right) \tag{20}$$

If we take plus and minus matrices as given in Eq. (10), then,

$$A^+ - A^- = r_A I,\ B^+ - B^- = r_B I$$

Thus Eq. (19) becomes the LU-SSOR implicit scheme

$$\begin{aligned} &\left\{ I + \beta\Delta t\left(D_x^- A^+ + D_y^- B^+ - A^- - B^- \right) \right\} \\ &\left\{ I + \beta\Delta t\left(D_x^+ A^- + D_y^+ B^- + A^+ + B^+ \right) \right\} \delta W \\ &\quad = -\left\{ 1 + \beta\Delta t\left(r_A + r_B \right) \right\} \Delta t R \end{aligned} \tag{21}$$

To preserve symmetry, the system for the lower half-plane can be modified to

$$\begin{aligned} &\left\{ I + \beta\Delta t\left(D_x^+ A^- + D_y^- B^+ + A^+ - B^- \right) \right\} \\ &\left\{ I + \beta\Delta t\left(D_x^- A^+ + D_y^+ B^- - A^- + B^+ \right) \right\} \delta W \\ &\quad = -\left\{ 1 + \beta\Delta t\left(r_A + r_B \right) \right\} \Delta t R \end{aligned} \tag{22}$$

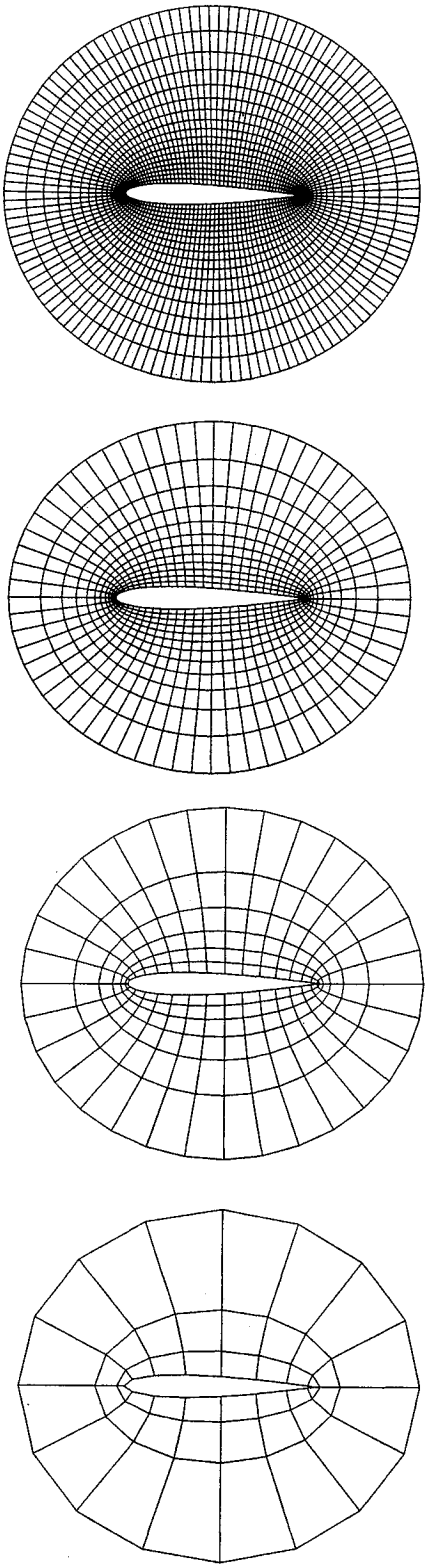


Fig. 1 Four-level multiple grids: a) 128 × 32 O-mesh; b) 64 × 16 O-mesh; c) 32 × 8 O-mesh; d) 16 × 4 O-mesh.

The derivation of the scheme (21) shows that a symmetric Gauss-Seidel relaxation method to solve the unfactored implicit scheme (6) is equivalent to a modified form of the LU implicit scheme.

VI. Multigrid Method

In order to adapt the implicit schemes for a multigrid algorithm, auxiliary meshes are introduced by doubling the mesh spacing. Figures 1a–1d show a typical sequence of four O-type meshes. A typical C-mesh is shown in Fig. 2. The solution vector on a coarser grid is initialized as

$$W_{2h}^{(0)} = \Sigma S_h W_h / S_{2h} \tag{23}$$

where the subscripts denote values of the mesh spacing parameter, S is the cell area, and the sum is over the four cells on the fine grid composing each cell on the coarser grid. The rule conserves mass, momentum, and energy. The solution on a coarse grid is updated as follows.

1) Calculate the correction, and update the solution on the fine grid.

2) Transfer the values of the variables to the coarse grid.

3) Collect the residual on the fine grid for the coarse grid. A forcing function is then defined as

$$P_{2h} = \Sigma R_h(W_h) - R_{2h}\left(W_{2h}^{(0)}\right) \tag{24}$$

where R is the residual. The residual on the coarse grid is given by

$$R_{2h}^{*} = R_{2h}(W_{2h}) + P_{2h} \tag{25}$$

4) Calculate the correction, and update the solution on the coarse grid. For the next coarser grid, the residual is recalculated as

$$R_{4h}^{*} = R_{4h}(W_{4h}) + P_{4h} \tag{26}$$

where

$$P_{4h} = \Sigma R_{2h}^{*} - R_{4h}\left(W_{4h}^{(0)}\right) \tag{27}$$

The process is repeated on successively coarser grids.

5) Finally, pass the correction calculated on each grid back to the next finer grid by bilinear interpolation.

Let W_{2h}^{+} be the final value of W_{2h} resulting from both the correction calculated on grid $2h$ and the correction transferred from grid $4h$. Then

$$W_h^{+} = W_h + I_{2h}^{h}\left(W_{2h}^{+} - W_{2h}^{(0)}\right) \tag{28}$$

where W_h is the solution on grid h before the transfer from grid $2h$, and I is an interpolation operator.

Since the evolution on a coarse grid is driven by residuals collected from the next-finer grid, the final solution on the fine grid is independent of the choice of boundary conditions on the coarse grids. The surface boundary condition is treated in the same way on every grid, by using the normal pressure gradient to extrapolate the surface pressure from the pressure in the cells adjacent to the wall. Values are extrapolated to the fictitious cells inside the body surface for the second difference dissipation on the coarse grids. The far-field conditions can either be transferred from the fine grid or recalculated.

Several time steps might be used at each level of the multigrid cycle, but it appears that an effective multigrid strategy is to use a simple sawtooth cycle, in which a transfer is made from each grid to the next-coarser grid after a single time step. After reaching the coarsest grid, the corrections are then successively interpolated back from each grid to the next-finer grid without any intermediate Euler calculations. On each grid, the time step is varied locally to yield a fixed CFL number, and the same CFL number is generally used on all grids, so that progressively larger time steps are used after each transfer to a coarser grid. In comparison to a single time step on the fine grid, the total computational effort in one multigrid cycle is

$$1 + \frac{1}{4} + \frac{1}{16} + \cdots < \frac{4}{3}$$

plus the additional work of collecting residuals and interpolating the corrections.

The interpolation of corrections back to the fine grid will introduce errors that cannot be rapidly expelled from the fine grid and ought to be locally damped if a fast rate of convergence is to be attained. Thus, it is important that the driving scheme should have the property of rapidly damping out high-frequency modes. The success of a multigrid method is critically dependent on the shape of the amplification factor[4] as a function of the wave numbers ξ and η in the two coordinate directions. Using the von Neumann stability test, the amplification factor $G(\xi, \eta)$ of the LU implicit scheme

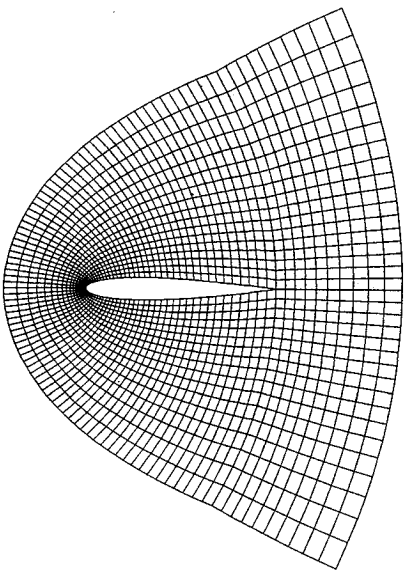


Fig. 2 128×32 C-mesh.

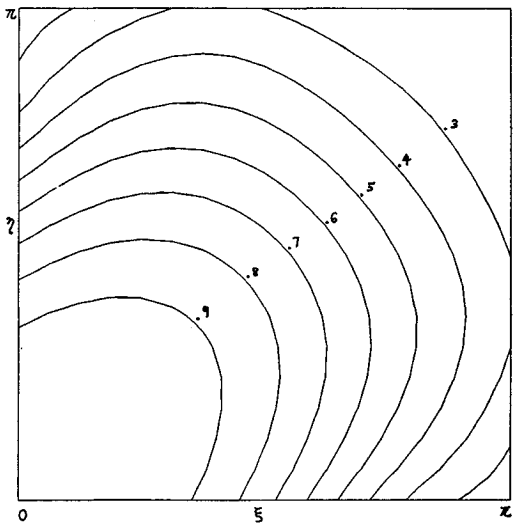


Fig. 3 Growth factor of the LU scheme.

may be calculated for the simple model problem $U_t + aU_x + bU_y = 0$. Figure 3 shows good damping characteristics of the LU implicit scheme with $\beta = \frac{1}{2}$.

VII. Boundary Conditions

Improper treatment of the boundary conditions can lead to serious errors and perhaps instability. At a solid boundary, the only contribution to the flux balance comes from the pressure. The normal pressure gradient $\partial p/\partial n$ at the wall can be estimated from the condition that $\partial/\partial t\ (\rho q_n) = 0$, where q_n is the normal velocity component. The pressure at the wall is then estimated by extrapolation from the pressure at the adjacent cell centers, using the known value of $\partial p/\partial n$ (Ref. 3).

In order to treat the flow exterior to a profile, one must introduce an artificial outer boundary to produce a bounded domain. If the flow is subsonic at infinity, there will be three incoming characteristics where there is inflow across the boundary and one outgoing characteristic, corresponding to the possibility of escaping acoustic waves. Where there is outflow, on the other hand, there will be three outgoing characteristics and one incoming characteristic. According to the theory of Kreiss,[9] three conditions may therefore be specified at the inflow boundary and one at the outflow boundary, while the remaining conditions are determined by the solution of the differential equation. It is not correct to specify freestream conditions at the outer boundary.

The rate of convergence to a steady state will be impaired if outgoing waves are reflected back into the flow from the outer boundaries. A perfect nonreflecting boundary condition, which absorbs all waves impinging on the boundary, requires global information in both space and time. Any boundary condition using local information will lead to some reflection. Approximations designed to absorb waves propagating in a direction normal to the boundary prove useful. Two variations have been tried in this work. The treatment of the far-field boundary condition for the O-mesh is based on the introduction of Riemann invariants for a one-dimensional flow normal to the boundary. The treatment of the outer boundary condition for the C-mesh follows the lines of Ref. 3. The equations normal to the boundary are linearized about values at the end of the previous time step, and the characteristic variables corresponding to outgoing characteristics are then determined by extrapolation from the interior. The remaining boundary conditions are specified in a manner consistent with the conditions imposed by the freestream.

VIII. Results

Two-dimensional calculations have been performed to test the multigrid method with the implicit schemes. Since the mesh generation has been separated from the Euler solver, any standard quadrilateral mesh can be used. In this work, an O-mesh generated by a conformal mapping to a near circle and a C-mesh generated by sheared parabolic coordinates are used. The O-mesh allows for denser spacing on the airfoil surface than the C-mesh, with the result that the O-mesh gives more accurate results than the C-mesh if the number of mesh points is the same. However, the C-mesh seems to be better suited for viscous calculations since it can provide better wake resolution with the aid of mesh stretching. Airfoil calculations were carried out on both meshes with 128 intervals in the direction around the airfoil and 32 intervals in the radial direction. These meshes are shown in Figs. 1a and 2.

The figures show both the computed solution and the convergence history. In each plot of the convergence history, two indicators of the convergence rate are presented. One is the decay of the logarithm of the error, where the error is measured by the root-mean-square rate of change of density on the fine grid. The other is the buildup of the number of points in the supersonic zone. For a transonic flow, this indicates how quickly the supersonic zone develops and is a useful measure of the global convergence of the flowfield.

In order to compare the different schemes both with and without multigrid, a number of examples are presented for the NACA 0012 airfoil at zero angle of attack and Mach 0.8. This nonlifting flow is a useful test case to check the accuracy and performance of a given scheme since results are available from a number of previous Euler codes. Inaccurate schemes may also fail to preserve symmetry and predict nonzero lift at zero angle of attack.

The solution by the multigrid LU scheme on the C-mesh is shown in Fig. 4. Figure 5 shows the Mach number contours. The dashed line indicates the sonic line. Figures 6 and 7 show the convergence histories of the LU implicit scheme on the C-mesh with and without multigrid. The supersonic zone was

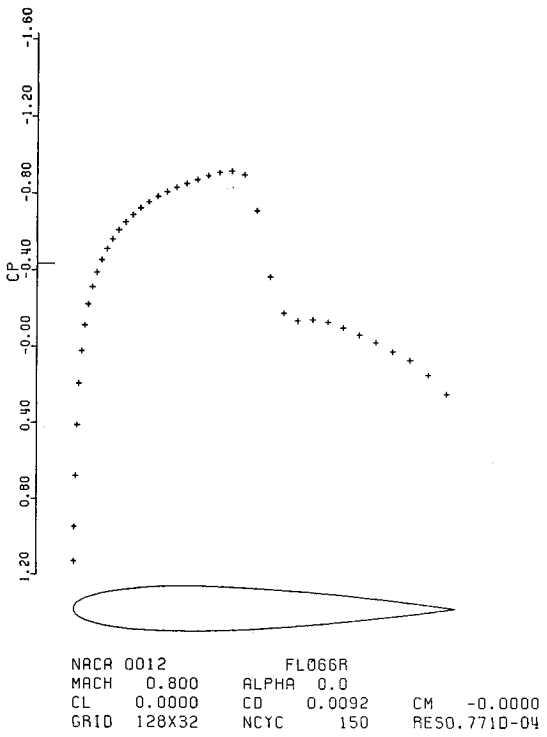


Fig. 4 Surface pressure distribution for transonic flow.

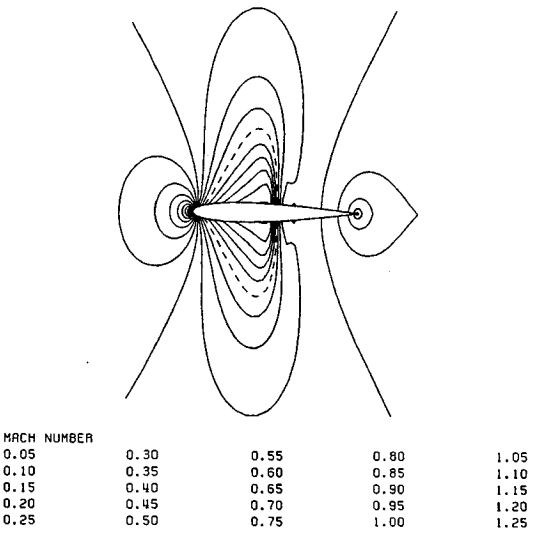
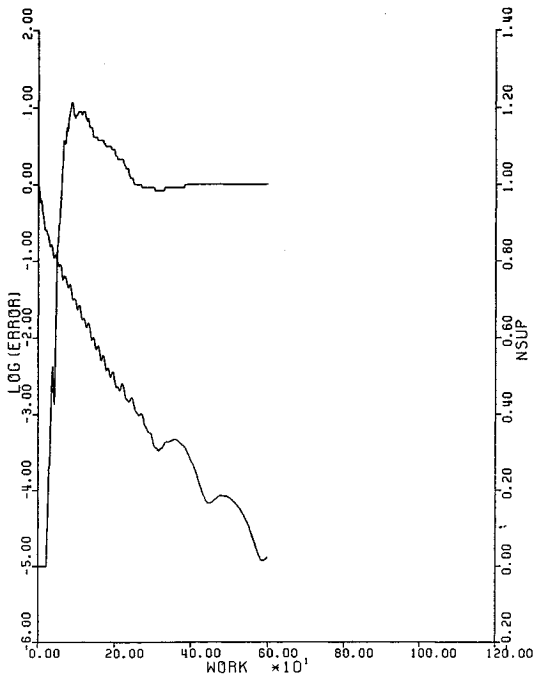
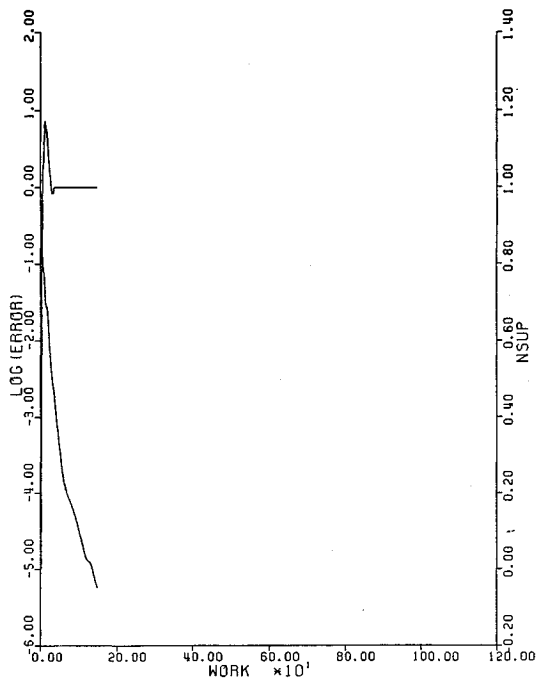


Fig. 5 Mach number contours for transonic flow.



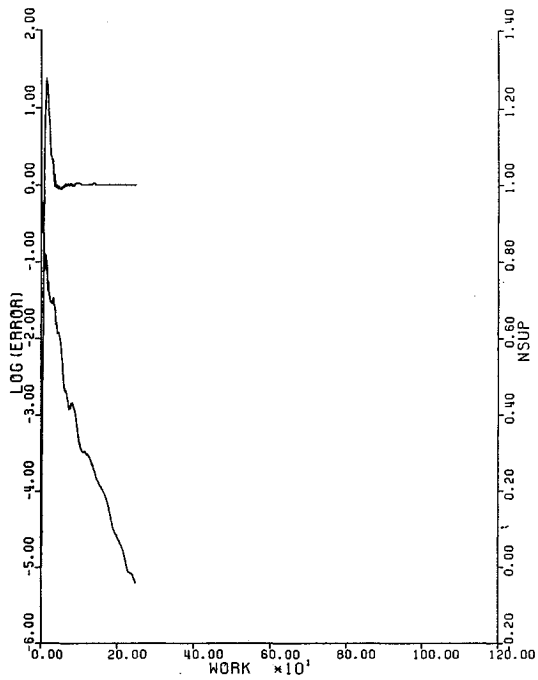
NACA 0012		FL066R	
MACH	0.800	ALPHA	0.0
RESID1	0.946D+01	RESID2	0.123D-03
WORK	598.00	RATE	0.9814
GRID	128X32		

Fig. 6 Convergence history of the LU scheme on C-mesh with single grid.



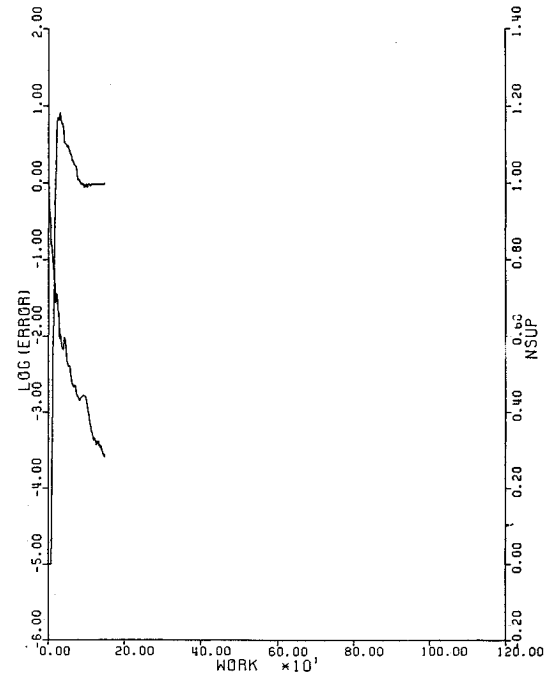
NACA 0012		FL066R	
MACH	0.800	ALPHA	0.0
RESID1	0.133D+02	RESID2	0.771D-04
WORK	149.00	RATE	0.9222
GRID	128X32		

Fig. 7 Convergence history of the LU scheme on C-mesh with multiple grids.



NACA 0012		FL065R	
MACH	0.800	ALPHA	0.0
RESID1	0.115D+01	RESID2	0.716D-05
WORK	249.00	RATE	0.9530
GRID	128X32		

Fig. 8 Convergence history of the LU scheme on O-mesh with multiple grids.



NACA 0012		FL056M	
MACH	0.800	ALPHA	0.0
RESID1	0.154D+01	RESID2	0.396D-03
WORK	149.00	RATE	0.9460
GRID	128X32		

Fig. 9 Convergence history of the SSOR scheme on O-mesh with multiple grids.

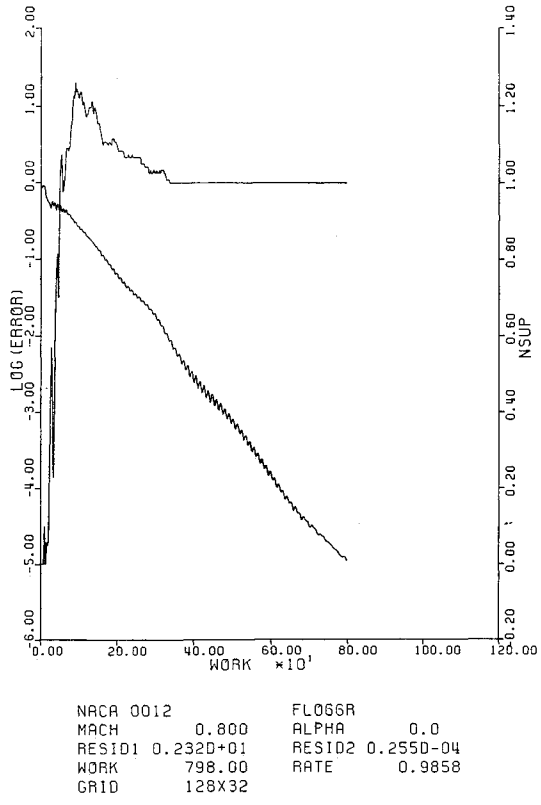


Fig. 10 Convergence history of the LU scheme with constant time stepping.

frozen in 35 cycles with four-level multigrid and in 400 cycles on the single grid. Figures 8 and 9 show O-mesh convergence histories of the LU implicit scheme and the SSOR scheme, respectively. The SSOR scheme is clearly stable and convergent although the convergence rate is worse than that of the LU implicit scheme.

All the previous examples have been run using enthalpy damping in order to enhance convergence to the steady state. In addition, convergence has been improved by using a variable time step. In these calculations, a CFL number of 5 was used with local time stepping. When calculating unsteady flows, these acceleration devices are no longer available and computations must be carried out with a constant time step. Implicit schemes have the advantage of allowing the use of a much larger time step than is normally possible with explicit schemes. As an illustration of this possibility, Fig. 10 shows the convergence history with a CFL number of 35, using a constant time step without enthalpy damping. Although local time stepping gives a little faster convergence, the use of a high CFL number suggests that the scheme would be a useful tool for unsteady flow analysis.

Conclusion

Although the ADI scheme is useful for two-dimensional problems, its inherent limitations in three dimensions suggest an alternative approach. The LU implicit scheme is shown to be well suited to a multigrid method. The number of cycles to reach a steady state was reduced by an order of magnitude by the introduction of multiple grids. The computational work per cycle for the implicit scheme is found to be comparable to that of an explicit multistage scheme. Extensions to three-dimensional, unsteady, and viscous flows, and the implementation of upwind schemes can be realized within this framework.

References

[1]Magnus, R. and Yoshihara, H., "Inviscid Transonic Flow over Airfoils," *AIAA Journal*, Vol. 8, Dec. 1970, pp. 2157–2162.

[2]MacCormack, R.W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-0354, 1969.

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

[4]Jameson, A. and Yoon, S., "Multigrid Solution of the Euler Equations using Implicit Schemes," *AIAA Journal*, Vol. 24, Nov. 1986, pp. 1737–1743.

[5]Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, No. 156, 1981, pp. 385–397.

[6]Buratynski, E.K. and Caughey, D.A., "An Implicit LU Scheme for the Euler Equations Applied to Arbitrary Cascades," AIAA Paper 84-0167, 1984.

[7]Steger, J.L. and Warming, R.F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods" *Journal of Computational Physics*, Vol. 40, 1981, pp. 263–293.

[8]Yoon, S., "Numerical Solution of the Euler Equations by Implicit Schemes with Multiple Grids," Ph.D. Thesis, MAE Rept. 1720-T, Princeton University, Princeton, NJ, Sept. 1985.

[9]Kreiss, H.O., "Initial Boundary Value Problems for Hyperbolic Systems," *Communications on Pure and Applied Mathematics*, Vol. 23, 1970, pp. 277–298.