Transonic Flow Calculations Using the Euler Equations

H. L. Atkins* and H. A. Hassan†
North Carolina State University, Raleigh, North Carolina

An implicit finite difference method with implicit boundary conditions is employed to solve the steady Euler equations for flows past arbitrary geometries. The resulting code is used to investigate in a systematic way various aspects of flow past airfoils at transonic speeds such as method of solution, boundary conditions, grid stretching and generation, shock and sonic point operators, the Kutta condition, and smoothing. Results obtained are in good agreement with results of other codes. Moreover, it appears that the method of solution employed is such that Kutta's condition need not be invoked. This statement appears to be valid for other existing schemes employed in the solution of Euler equations.

Introduction

THE primitive variable form of the system of partial-differential equations governing the steady motion of an inviscid fluid (Euler equations) are first order and of mixed elliptic and hyperbolic type. Because of this, two approaches are used in obtaining steady-state solutions for subcritical and supercritical flows past bodies. The first seeks temporally asymptotic solutions of the time-dependent equations[1-7] while the second exploits relaxation methods developed for the solution of second-order equations by embedding Euler equations in a second-order system[8] or by using a stream function formulation.[9]

The purpose of this investigation is to develop an efficient algorithm for the steady Euler equations in primitive variable form and use it to carry out numerical experiments designed to investigate in a systematic way the various aspects of flows past airfoils at transonic speeds. These include method of solution, i.e., Peaceman-Rachford[10] (PR) vs Douglas-Gunn[11] (DG); boundary conditions; grid generation and stretching; shock and sonic point operators; smoothing; and the Kutta condition. The resulting algorithm solves the strong conservation form of the Euler equations, with the energy equation replaced by the statement that the total enthalpy is constant, using an ADI finite difference scheme and implicit boundary conditions. To study arbitrary geometries, the algorithm is combined with the automatic grid solver (GRAPE) of Steger and Sorenson.[12] With this capability, the code provides a general and efficient method for the solution of flowfields past arbitrary geometries for all speed ranges. The achieved efficiency of this code can be traced to two main reasons: the resulting block tridiagonal matrix for two-dimensional flows is 3×3 and not 4×4; moreover, the code incorporates a procedure that selects an optimum relaxation factor for each iteration.

Formulation of the Problem

A. Governing Equations and Transformations

The steady conservative Euler equations in dimensionless form can be written as

$$F_x+G_y=0, \quad F_x=\partial F/\partial x, \quad \text{etc.} \tag{1}$$

where

$$F=\begin{bmatrix}\rho u^2+p\\ \rho uv\\ \rho u\end{bmatrix}, \quad G=\begin{bmatrix}\rho uv\\ \rho v^2+p\\ \rho v\end{bmatrix} \tag{2}$$

ρ is the density, p the pressure, and u and v the velocity components in the x and y directions. The dependent variables were normalized by ρ_∞, p_∞, and $(p_\infty/\rho_\infty)^{1/2}$, respectively, while the independent variables were normalized by the chord length. For adiabatic flows, the steady energy equation reduces to

$$T=1+\frac{\gamma-1}{2\gamma}(\gamma M_\infty^2-u^2-v^2) \tag{3}$$

where T is the temperature, γ the ratio of specific heats, and M_∞ the freestream Mach numbers. The equation of state

$$p=\rho T \tag{4}$$

and Eq. (3) were used to eliminate the pressure and temperature from the governing equations with the result that the dependent variables are the primitive variables u, v, and ρ.

Equation (1) is transformed into a rectangular computational plane (ξ,η), i.e.,

$$\xi=\xi(x,y), \quad \eta=\eta(x,y) \tag{5}$$

with the result that Eq. (1) takes the form

$$(1/J)(\bar{F}_\xi+\bar{G}_\eta)=0 \tag{6}$$

where J is the transformation Jacobian and is given by

$$J=x_\xi y_\eta-x_\eta y_\xi=1/(\xi_x\eta_y-\xi_y\eta_x) \tag{7}$$

and

$$\bar{F}=Fy_\eta-Gx_\eta, \quad \bar{G}=-Fy_\xi+Gx_\xi \tag{8}$$

B. Type-Dependent Operators

Equation (6) represents a system of nonlinear partial-differential equations. A first step in the solution of these equations is to linearize them using a Taylor expansion. The result can be written as

$$(L_\xi A+L_\eta B)\Delta W=-(L_\xi\bar{F}+L_\eta\bar{G}) \tag{9}$$

where L_ξ and L_η are the operators resulting from the ξ and η

Presented as Paper 82-0106 at the AIAA 20th Aerospace Sciences Meeting, Orlando, Fla., Jan. 11-14, 1982; submitted Jan. 22, 1982; revision received Sept. 28, 1982.

*Research Assistant. Student Member AIAA.
†Professor. Associate Fellow AIAA.

differentiation and

$$\Delta W = W^{n+1} - W^n, \quad A = \frac{\partial F}{\partial W}, \quad B = \frac{\partial G}{\partial W}, \quad W = \begin{bmatrix} u \\ v \\ \rho \end{bmatrix} \tag{10}$$

The finite difference operators L_ξ and L_η are chosen in such a way that they are second-order accurate, conservative, and reflect the physics of the flow. To meet these objectives a central difference operator is used in the η direction and a type-dependent operator in the ξ direction, which is the mean flow direction. It was found that a type-dependent operator similar to that developed by Murman[13] for the potential equation is not suited for Euler equations when an ADI scheme is employed. This is because it involves four points and because it has undesirable features around the sonic line which led to severe stability problems. It was found that, for the problem under consideration, a number of three-point operators that maintain strict conservation[14] are satisfactory. The results presented here employ the form[1]

$$L_\xi Q_{i,j} = \frac{1}{\Delta\xi} \frac{\mu\delta - \tfrac{1}{2}\nabla(\epsilon_{ij}\Delta)}{1 - \tfrac{1}{2}\nabla\epsilon_{ij}} Q_{i,j} \tag{11}$$

where

$$\begin{aligned} Q_{i,j} &= Q(\xi_i, \eta_j), \quad \xi_i = (i-1)\Delta\xi + \xi_{\min} \\ \eta_j &= (j-1)\Delta\eta + \eta_{\min} \\ \mu\delta Q_{i,j} &= \tfrac{1}{2}(Q_{i+1,j} - Q_{i-1,j}) \\ \nabla Q_{i,j} &= Q_{i,j} - Q_{i-1,j}, \quad \Delta Q_{i,j} = Q_{i+1,j} - Q_{i,j} \end{aligned} \tag{12}$$

$$\begin{aligned} \epsilon_{i,j} &= 1, \quad M_{i,j} \geq 1 \\ &= 0, \quad M_{i,j} < 1 \end{aligned} \tag{13}$$

and M is the Mach number. When both the upper and lower halves of the plane are considered, the operator indicated in Eq. (10) is generalized to

$$L_\xi Q_{i,j} = \frac{1}{\Delta\xi} \frac{\mu\delta - (\alpha/2)\nabla(\epsilon_{ij}\Delta) + [(1-\alpha)/2]\Delta(\epsilon_{ij}\nabla)}{1 - (\alpha/2)\nabla\epsilon_{ij} + [(1-\alpha)/2]\Delta\epsilon_{ij}} Q_{i,j} \tag{14}$$

$$\begin{aligned} \alpha &= 1, \quad \text{upper half} \\ &= 0, \quad \text{lower half} \end{aligned} \tag{15}$$

Both operators indicated in Eqs. (11) and (14) are generalized to allow for one-sided differencing at a boundary by setting

$$\mu\delta = \lambda_\xi \nabla + (1 - \lambda_\xi)\Delta \tag{16}$$

where

$$\begin{aligned} \lambda_\xi &= 1 \text{ at } \xi_i = \xi_{\max} \text{ if } M_{i,j} < 1 \\ &= 0 \text{ at } \xi_i = \xi_{\min} \text{ if } M_{i,j} < 1 \\ &= \tfrac{1}{2} \text{ otherwise} \end{aligned} \tag{17}$$

The L_η operator, when generalized to allow for one-sided differencing at the boundary, takes the form

$$L_\eta Q_{i,j} = (1/\Delta\eta)[\lambda_\eta \nabla + (1 - \lambda_\eta)\Delta] Q_{i,j} \tag{18}$$

where

$$\begin{aligned} \nabla Q_{i,j} &= Q_{i,j} - Q_{i,j-1}, \quad \Delta Q_{i,j} = Q_{i,j+1} - Q_{i,j} \\ \lambda_\eta &= 1, \eta = \eta_{\max} \\ &= 0, \eta = \eta_{\min} \\ &= \tfrac{1}{2} \text{ otherwise} \end{aligned} \tag{19}$$

The transformed equations are balanced exactly for a constant flowfield as long as the same operators are used to evaluate both metrics and flow variables. Thus, great care must be exercised if type-dependent operators are used. Even when the metrics are evaluated using the same operator, a grid error correction term must be introduced to insure the desired accuracy. This is because $L_\xi L_\eta \neq L_\eta L_\xi$ everywhere.

C. Method of Solution

The governing equations are solved by employing ADI or splitting techniques. Two procedures were examined here. In the first, the approximate factorization or Douglas-Gunn method was employed. In this method the time-dependent terms in the governing equations are reinstated and the algorithm reduces to a form similar to that of Ref. 2. The second procedure employed is that of Peaceman and Rachford.[10] The delta form of the equations can be written as

$$[\rho_n C^n + (1/J)(L_\xi A^n + L_\eta B^n)]\Delta W = -(1/J)[L_\xi \bar{F} + L_\eta \bar{G}] \tag{20}$$

where ρ_n is the relaxation factor for the nth iteration and C^n is a 3×3 matrix. C^n can be chosen as the unit matrix; however, improved convergence resulted when C was chosen as

$$C = \frac{\partial H}{\partial W}, \quad H = \begin{bmatrix} \rho u \\ \rho v \\ \rho \end{bmatrix} \tag{21}$$

An algorithm based on Eq. (20) was implemented using the two steps

$$[\rho_n C^n + (\beta/J) L_\xi A^n]\Delta W = -(1/J)[L_\xi \bar{F} + L_\eta \bar{G}] + D^n \tag{22}$$

$$[\rho_n C^* + (\beta/J) L_\xi B^*]\Delta W^* = -(1/J)[L_\xi F^* + L_\eta G^*] + D^* \tag{23}$$

with

$$\Delta W = W^* - W^n, \quad \Delta W^* = W^{n+1} - W^*, \quad C^* = C(W^*), \text{ etc.} \tag{24}$$

β is a relaxation parameter and D a smoothing function.[1,2] The algorithm was implemented using both implicit second-order smoothing and explicit fourth-order smoothing.

For the factored or Douglas-Gunn scheme n refers to the time step while for the Peaceman-Rachford method n refers to the iteration number. Briley and McDonald[15] have shown that the Peaceman-Rachford scheme with $\beta = 1$ is equivalent to the Douglas-Gunn splitting when a Crank-Nicholson scheme is applied to a linear problem. It is known, however, that both schemes are only neutrally stable in the mode and are never employed in this way. When employed in a stable manner ($\beta > 1$ for PR and backward Euler for DG) the characteristics of the two schemes are different.

D. Boundary Conditions

After intensive investigation, it was concluded that implicit boundary conditions should be used for all Mach numbers. This represents a departure from the procedure of Ref. 2 and, as such, an alternative formulation of boundary conditions.

The stated choice led to a rather robust algorithm. Thus, the same fourth-order explicit smoothing term was used for all cases presented. Because of this the code does not have adjustable parameters and does not require special "fixes" in certain situations.

Cartesian sheared grids and mapped C grids were employed. For the Cartesian grids, the boundary conditions on the body were obtained from setting the normal velocity equal to zero and from the one-sided differencing of the continuity and the x momentum equations. When C grids were employed, the x momentum was replaced by a linear combination of the ξ and η momentum equations. Satisfactory results are obtained by setting $\Delta W=0$ at the upstream boundary. For supercritical Mach numbers, the downstream boundary conditions must reflect the fact that certain streamlines are characterized by constant entropy while others are not. For these cases, the downstream boundary conditions were obtained from the one-sided differencing of the continuity and the y (or η) momentum equations and $s=s_\infty$ above the supersonic bubble (s is the entropy). Below the bubble, the one-sided differencing of the governing equations was employed. Other conditions employed included $\Delta W=0$, the one-sided differencing of two conservation equations, and one of the following conditions: $v=0$ for zero angle of attack or $p=p_\infty$.

Results and Discussion

The numerical experiments carried out in this work are intended to answer a number of questions fundamental to the development of an efficient code for the solution of Euler equations. These included method of solution, type and amount of smoothing, shock and sonic point operators, grid size and location of outer boundary, and the Kutta condition. Although calculations were carried out for different airfoils, angles of attack, and Mach numbers, the figures presented here are for a NACA 0012 airfoil, zero angle of attack, and a freestream Mach number of 0.85. Unless indicated otherwise, freestream conditions are assumed at the upstream boundary while the downstream boundary employs $s=s_\infty$, the one-sided differencing of two of the governing equations above the supersonic bubble, and the one-sided differencing of the governing equations below the bubble. To keep the computational cost to a minimum, some calculations were carried out using sheared Cartesian grids, the others employed C grids generated by GRAPE[12]; figures using these grids are labeled as such.

Figure 1 compares the convergence history of Douglas-Gunn vs Peaceman-Rachford for $M_\infty=0.5$ with and without fourth-order smoothing. Without smoothing, both schemes have similar convergence history. On the other hand, explicit fourth-order smoothing accelerates Peaceman-Rachford much more than Douglas-Gunn. Similar results were obtained when implicit second-order smoothing was employed. Figure 2 shows that similar conclusions hold for a supercritical Mach number of 0.85. The figure shows also that marked improvement in the convergence history of the Peaceman-Rachford scheme results when a variable relaxation factor is employed. This factor is doubled when the spectral radius (ratio of successive residuals) increases beyond a predetermined number; otherwise, it is reduced by a certain percentage. As a result of this, the relaxation factor is almost the optimum factor throughout the calculations. When a variable time step was used with the Douglas-Gunn scheme, little improvement in the convergence history was noted. Similar results were obtained for subcritical Mach numbers. Because of this, subsequent results employ the Peaceman-Rachford method with variable ρ_n.

The question whether to use an implicit second-order smoothing or an explicit fourth-order smoothing or both is discussed next. Figure 3 compares a C_p plot for both types of smoothing. Implicit second-order smoothing resulted in the formation of a spike behind the shock. Because of this,

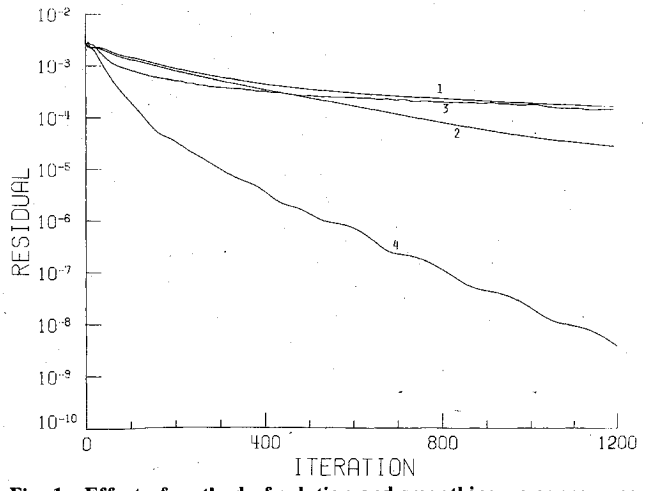

Fig. 1 Effect of method of solution and smoothing on convergence rate ($M_\infty=0.5$): 1) Douglas-Gunn (DG), no smoothing; 2) DG, fourth-order smoothing; 3) Peaceman-Rachford (PR), no smoothing; and 4) PR, fourth-order smoothing.

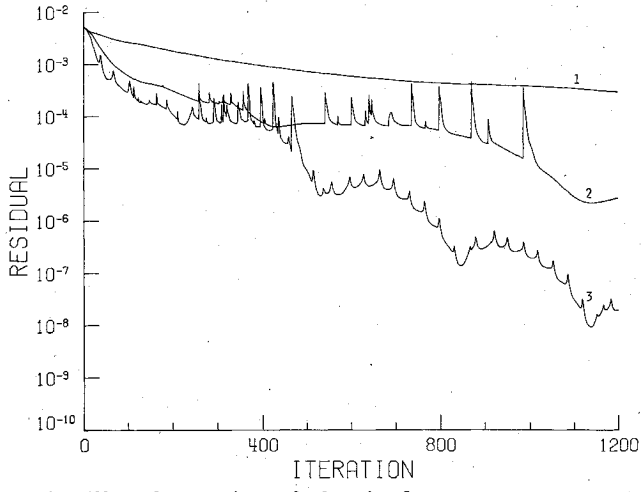

Fig. 2 Effect of smoothing and relaxation factor on convergence rate ($M_\infty=0.85$): 1) DG, fourth-order smoothing, 2) PR, fourth-order smoothing, and 3) PR, fourth-order smoothing and variable relaxation factor.

subsequent results employed fourth-order smoothing in the form

$$D=\omega[(\nabla_\xi\Delta_\xi)^2+(\nabla_\eta\Delta_\eta)^2]W_{i,j} \tag{25}$$

All computations reported here assume $\omega=1$. Contrary to the experience of Refs. 1 and 2, no oscillations were observed when the preceding term was kept throughout the computational field. Conservative suppression of this term (with $\omega=1$) in the supersonic region resulted in oscillations.

Comparison of central differencing throughout the flowfield with type-dependent differencing is shown in Fig. 4. When central differencing is used throughout, a spike develops before the shock; otherwise, it is in good agreement with the type-dependent operator. Nonconservative type-dependent operators led to instabilities. This suggests that, as long as the scheme is conservative, good results can be expected.

When carrying out calculations of the type under consideration, questions arise concerning how fine a grid should be employed and how far from the body the outer boundary should be placed. Stretching functions of the type

$$x=a\xi f(\xi^2)/(1-\xi^2)^\alpha$$

$$f(\xi^2)=1-C_1\alpha\xi^2+C_2\alpha(1-\alpha)\xi^2/2 \tag{26}$$

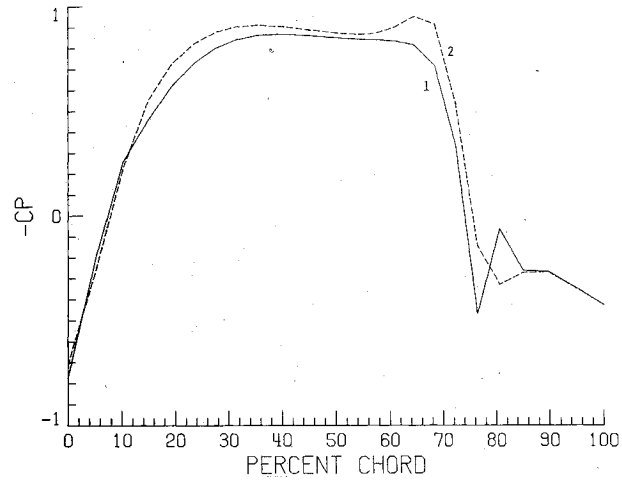

Fig. 3 Effect of smoothing on pressure coefficient: 1) second-order smoothing and 2) fourth-order smoothing.

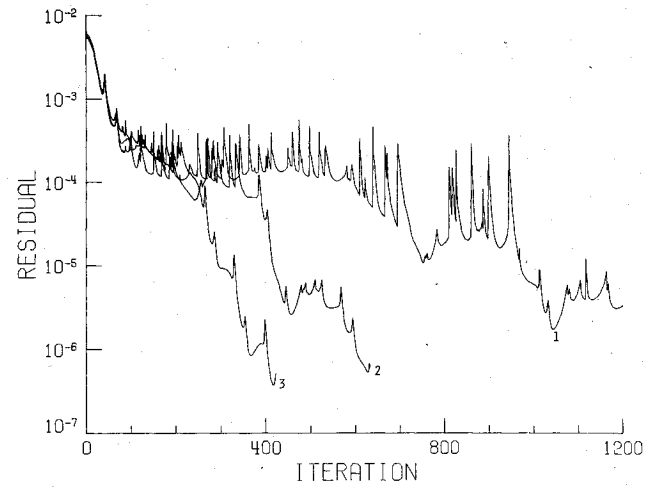

Fig. 6 Influence of outer boundary location on convergence rate: 1) $y=5.02$, 2) $y=3.45$, and 3) $y=2.36$.

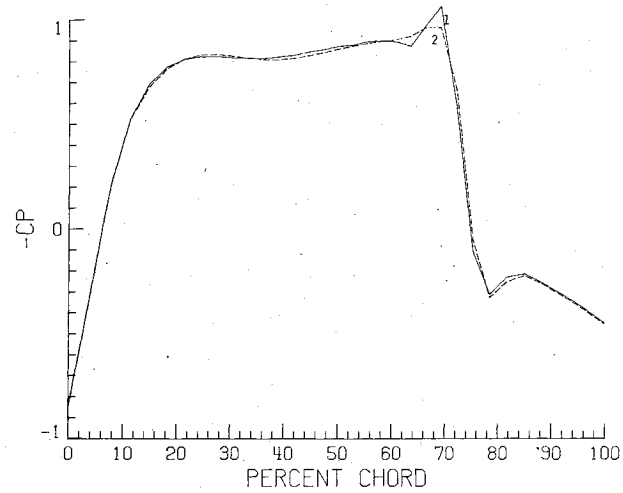

Fig. 4 Influence of L_ξ operator on pressure coefficient: 1) central difference and 2) type dependent.

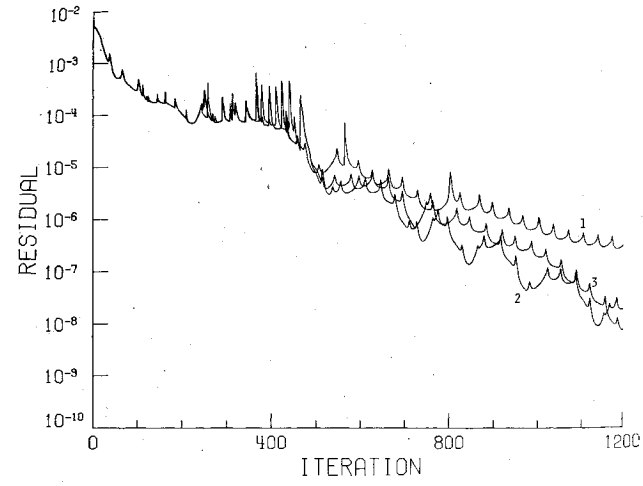

Fig. 7 Effect of downstream boundary conditions on convergence rate: 1) $v=0$, 2) $p=p_\infty$, and 3) $s=s_\infty$ above supersonic bubble.

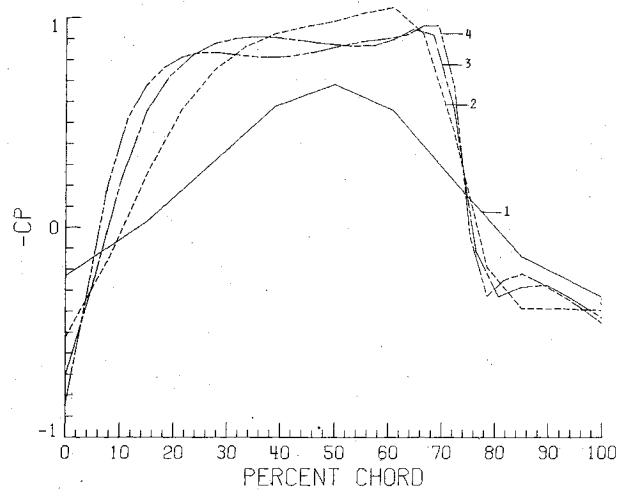

Fig. 5 Effect of grid size on pressure distribution: 1) 16×6, 2) 32×12, 3) 48×16, and 4) 64×24.

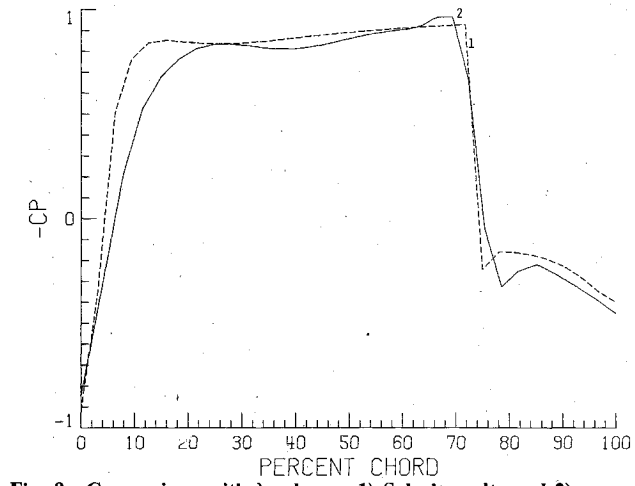

Fig. 8 Comparison with λ scheme: 1) Salas' results and 2) present method.

and a similar expression for y was employed when sheared Cartesian grids were employed. In Eq. (25), a, α, C_1, and C_2 are constants with C_1, C_2 being 0 or 1. The stretching function indicated in Eq. (26) was used to generate the results indicated in Fig. 5 for $|\xi|\leq 0.995$ and $0\leq\eta\leq 0.995$, $\alpha=\frac{1}{2}$ and $a=2$. Solutions are deemed convergent when the rms of the change in C_p is less than 10^{-6}. It is seen from the figure that the pressure coefficient is quite dependent on the grid size. However, the position of the shock seems to stabilize for the finer grids. Although the results are for a NACA 0012 airfoil, the foregoing problem is a result of the fact that the flow is "seeing" different airfoils.

The grid sizes indicated in Fig. 5 are for the upper half of the airfoil. When the angle of attack is different from zero both halves have to be considered with the result that computational costs are doubled for a given grid size. One

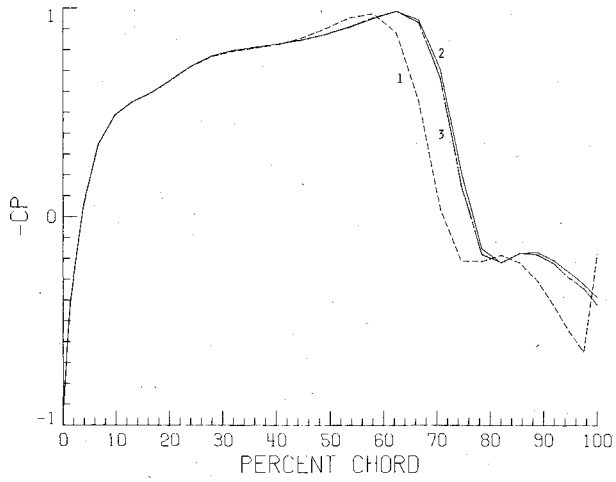

Fig. 9 Effect of conditions at trailing edge on pressure distribution (C grid): 1) stagnation point, 2) body point, and 3) wake point.

possibility for reducing the computational cost is to experiment with the location of the outer boundary without changing the cell size. Figure 6 shows the convergence history for the cells corresponding to the 48×16 grid when the outer boundary is placed at the positions indicated. This is equivalent to moving the outer boundary one, two, or four cells in the direction of the body. Significant reduction in computer time was achieved with little influence on the pressure distribution.

The effect of the downstream boundary condition is considered next. Because of the presence of a shock for supercritical flows, $\partial s/\partial n \neq 0$ where n is the normal to the streamline. Moreover, streamlines crossing the shock are characterized by an entropy jump while the others are not. Because of the preceding, the downstream boundary conditions are not obvious. A number of boundary conditions were examined and the convergence history for three of them is presented in Fig. 7. It is seen that all three downstream boundary conditions have the same convergence history and, thus, lead to the same pressure distribution. The reason for this is that for a freestream Mach number of 0.85 and zero angle of attack, the entropy gradient normal to the body is not severe. In all probability, higher supercritical Mach numbers and angles of attack would yield different results.

Comparison of the results of this work and those of Ref. 6, which were obtained using a time-dependent explicit method based on the λ scheme[16] is shown in Fig. 8. Good agreement is indicated.

All of the results presented so far were based on a sheared Cartesian grid with the leading and trailing edges placed between nodal points. To investigate the role of the Kutta condition, a C grid generated by GRAPE[12] is employed. When C grids are employed the cut from the trailing edge to the rear boundary is mapped into a boundary in the computational plane. The computations were carried out in such a way that points on that part of the boundary corresponding to the cut were treated as if they were interior points. It appears that the trailing edge may be handled in one of three ways: 1) assume velocity to be zero there; 2) treat it as part of the cut, this ensures continuity of properties at that point; and 3) treat it as a point on the airfoil. Because the slope is discontinuous when trailing-edge angle is finite, the flow properties are not equal at the upper and lower surfaces. However, because the next grid point lies in the wake, case 3 is essentially case 2 with the trailing edge shifted slightly to the right.

The pressure coefficients for the three options just indicated are shown in Fig. 9 for $M_\infty = 0.85$ at zero angle of attack. It is seen that the first option is not realistic because of the presence of the spike. Moreover, when such a condition is employed with a coarse grid (65×33), a small dead-water region with pressure lower than freestream pressure appears behind the trailing edge. This region shrinks appreciably when the grid is refined. This behavior can be traced to the fact that specifying a stagnation point at the trailing edge overspecifies the problem. This, in turn, can be interpreted to mean Kutta's condition need not be enforced when Euler equations are employed. Computations presented in Ref. 7 confirm this conclusion. However, the explanation advanced there is not appropriate. It is known[17] that Euler equations have a mechanism for generating vorticity when density and pressure nonuniformities exist. Even for subcritical Mach numbers where a pressure-density relationship exists and where the ratio of vorticity to density is constant along a streamline, the form of boundary conditions employed here and the presence of artificial viscosity result in the generation of vorticity. Thus, use of options 2 or 3 should give the correct lift.

Concluding Remarks

The numerical experiments presented here have resulted in an efficient code for the solution of the steady Euler equations. Results obtained using this code are in good agreement with Salas' explicit time-dependent code. Requiring the trailing edge to be a stagnation point overspecifies the problem; thus, it appears that for the scheme employed here, Kutta condition is not required.

Acknowledgments

Most of this work was carried out while the authors were at the Theoretical Aerodynamics Branch of the NASA Langley Research Center and was supported in part by NASA NCCI-22. The authors would like to acknowledge many helpful discussions with all members of the Branch, especially with Jerry C. South Jr., Manuel D. Salas, and Dr. Frank C. Thames and Dr. F. R. DeJarnette of the Mechanical and Aerospace Engineering Department of North Carolina State University.

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, Sept. 1976, pp. 87-110.

[2] Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, July 1978, pp. 679-686.

[3] Pulliam, T. H. and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, Feb. 1981, pp. 374-383.

[4] Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, Palo Alto, Calif., June 1981.

[5] 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, Palo Alto, Calif., June 1981, pp. 257-264.

[6] Salas, M. D., private communication, NASA Langley Research Center, 1981.

[7] Rizzi, A. and Eriksson, L. E., "Transfinite Mesh Generation and Damped Euler Equation Algorithm for Transonic Flow Around Wing-Body Configurations," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, Palo Alto, Calif., June 1981, pp. 43-68.

[8] Johnson, G. M., "An Alternative Approach to the Numerical Simulation of Steady Inviscid Flow," *Lecture Notes in Physics*, Vol. 141, edited by W. C. Reynolds and R. W. MacCormack, Springer-Verlag, Berlin, 1981, pp. 236-241.

[9] Hafez, M. and Lovell, D., "Numerical Solution of Transonic Stream Function Equation," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, Palo Alto, Calif., June 1981, pp. 364-372.

[10] Peaceman, D. W. and Rachford, H. H. Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 3, March 1955, pp. 28-41.

[11] Douglas, J. Jr. and Gunn, J. E., "A General Formulation of Alternating Direction Methods, Part 1. Parabolic and Hyperbolic Problems," *Numerische Mathematik*, Vol. 6, Dec. 1964, pp. 428-453.

[12] Steger, J. L. and Sorenson, R. L., "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations," *Journal of Computational Physics*, Vol. 33, Dec. 1979, pp. 405-410; also, Sorenson, R. L., NASA TM 81198, May 1980.

[13] Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," *AIAA Journal*, Vol. 12, May 1974, pp. 626-633.

[14] Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation," *Communications on Pure and Applied Mathematics*, Vol. 7, Feb. 1954, pp. 159-193.

[15] Briley, W. R. and McDonald, H., "On the Structure and Use of Linearized Block Implicit Schemes," *Journal of Computational Physics*, Vol. 34, Jan. 1980, pp. 54-73.

[16] Moretti, G., "The λ-Scheme," *Computers and Fluids*, Vol. 7, Sept. 1979, pp. 191-205.

[17] Yih, C. S., *Fluid Dynamics*, McGraw-Hill Book Co., New York, 1969, pp. 62-64.