A Fast Semidirect Method for Computing Transonic Aerodynamic Flows

E. Dale Martin*
NASA Ames Research Center, Moffett Field, Calif.

A fast, semidirect, iterative computational method, previously introduced for finite-difference solution of subsonic and slightly supercritical flow over airfoils, is extended both to apply to strongly supercritical conditions and to include full second-order accuracy in computing inviscid flows over airfoils. The nonlinear small-disturbance equations are solved iteratively by a direct, linear, elliptic solver. General, fully conservative, type-dependent difference equations are formulated, including parabolic- and shock-point transition operators that provide consistency with the integral conservation laws. These equations specialize either to first-order (Murman's equations) or to fully second-order-accurate equations. The equations, derived for u and v, have equivalent velocity-potential expressions. Additional stabilizing terms (resulting in an off-center u_t term) produce the iterative convergence, even with large supersonic zones. Various free parameters are evaluated for rapid convergence of the first-order scheme. Resulting pressure distributions and computing times are compared with the improved Murman-Cole line-relaxation method. The second-order results are highly accurate on a very coarse mesh. The timing comparisons indicate that the present method, although not presently as flexible, is significantly faster than the conventional line-relaxation method.

I. Introduction

FOR the computation of two-dimensional steady, inviscid, transonic aerodynamic flows over airfoils, current computer programs[1] using line-relaxation methods[2-4] and based on transonic small-disturbance theory are fast and flexible, having been highly developed over the past few years. As such, they provide a valuable analysis tool that complements methods and programs based on the full potential equation.[5-7] (See Ref. 8 for a comprehensive review and further references.) Because a computer is limited by the program's speed and efficiency (which become very important in complex three-dimensional flows), we continually seek to develop methods that are as fast and efficient as possible.

The development of rapid computational techniques for solving both linear and nonlinear partial differential equations such as those governing fluid flow has been advanced recently by the use of "direct elliptic solvers" within iteration schemes.[9-11] A direct elliptic solver (e.g., Ref. 12) is a fast, efficient algorithm that takes maximum advantage of the sparseness and regular block structure of a separable, linear, elliptic-operator matrix in a system of finite-difference equations. In a "semidirect" iterative method, a direct elliptic solver determines the finite-difference solution at all points simultaneously, with possible nonlinear "source terms" that depend on the previous iteration. With this "fully implicit" approach, iterative convergence can be faster than in other relaxation methods because changes from one iteration to the next are felt simultaneously at all points without the limitation of a "propagation speed."

Reference 13 extended the semidirect method to a problem that is both nonlinear and of mixed type, i.e., nonelliptic in some regions: slightly supercritical, inviscid, transonic flow over an airfoil in a subsonic freestream. (An elliptic operator is solved at each iteration in such a way that, when the solution converges, the elliptic character of the left side of the equations is cancelled out at nonelliptic points by appropriate terms on the right side.) For use in Ref. 13, a fast, direct "Cauchy-Riemann solver" was developed[14] to treat the discretized elliptic operators in first-order systems of partial differential equations. The equations solved in that study were equivalent to the improved Murman-Cole transonic difference equations,[2-4] which are type-dependent, fully conservative representations of the nonlinear transonic small-disturbance equations. With that formulation, the iterations did not converge when the flow was more than slightly supercritical. Subsequently, Ref. 15 described progress made and preliminary results computed in extending the method of Ref. 13 to higher-Mach-number flows over airfoils. The Cauchy-Riemann solver of Ref. 14 also was extended[16] to allow stabilizing terms to be added to the iteration procedure.

The purposes of this paper are 1) to formulate general, fully conservative, type-dependent, nonlinear, transonic small-disturbance difference equations that can be specialized either to Murman's first-order-accurate equations or to second-order-accurate difference equations; 2) to present and analyze the second-order-accurate, fully conservative transonic difference equations, which allow use of a coarse mesh for accurate solutions; 3) to describe the extension of the fast semidirect method to obtain stable iterated solutions for strongly supercritical conditions with large embedded supersonic zones, and 4) to present results of computed pressure distributions, including the computation times, the values of free parameters in the formulation that produce rapid convergence, and comparisons with Murman's line-relaxation method. The computed results that illustrate the present developments are for a thin symmetrical biconvex airfoil at zero incidence, but it is anticipated that the method can be extended to more general flows.

II. Governing Equations

The two-dimensional, steady, inviscid, irrotational, transonic small-disturbance equations are

$$u_x + v_y = bu_x + a(u^2)_x \tag{1a}$$

$$u_y - v_x = 0 \tag{1b}$$

in terms of the perturbation-velocity components u and v, where

$$U = 1 + \tau u, \quad V = \tau v \tag{2}$$

and where τ is an airfoil thickness ratio. The Cartesian coordinates x and y are normalized by the airfoil chord length. The

Presented at the AIAA 2nd Computational Fluid Dynamics Conference, Hartford, Conn., June 19-20, 1975 (in bound volume of papers; no paper number); submitted Aug. 4, 1975; revision received Jan. 19, 1976.

Index category: Subsonic and Transonic Flow.

*Research Scientist, Computational Fluid Dynamics Branch. Associate Fellow AIAA.

constants a and b are defined by

$$a=(\tau/2)(\gamma+1)f(M_\infty), \quad b=M_\infty^2 \tag{3}$$

where γ is the ratio of specific heats, and $f(M_\infty)=O(1)$ as $M_\infty \to 1$. The function $f(M_\infty)$ commonly is taken to be M_∞^2, as recommended in a study by Spreiter,[17] although different powers of M_∞ ($M_\infty^{3/2}$ and $M_\infty^{7/4}$) have been used, e.g., by Murman and his collaborators,[1,3,4] to "tune up" the small-disturbance theory for better agreement with solutions to the full potential equation.

Two convenient transformations are used. The first includes two constant free parameters that affect the rate of iterative convergence in an iterative solution of the equations. If ν_1 is an arbitrary positive constant "relaxation parameter" and $\bar{b}$ is an arbitrary constant "shifting[10] parameter" used to shift all or part of the bu_x term in Eq. (1a) from one side to the other (see Ref. 15 for motivation and derivation), then the transformation and resulting equations are

$$\bar{u}(x,\bar{y})=\nu_1^{-1/2}\bar{\beta}u(x,y), \quad \bar{v}(x,\bar{y})=v(x,y), \quad \bar{y}=\nu_1^{1/2}\bar{\beta}y \tag{4}$$

$$\bar{a}=a\nu_1^{1/2}/\bar{\beta}^3, \quad \bar{\beta}=[(1-b)/(1-\bar{b})]^{1/2} \tag{5}$$

$$\bar{u}_x+\bar{v}_y=\bar{b}\bar{u}_x+\bar{a}(\bar{u}^2)_x \tag{6a}$$

$$\bar{u}_{\bar{y}}-\bar{v}_x=(1-\nu_1)\bar{u}_{\bar{y}} \tag{6b}$$

Special cases include 1) $\nu_1=1$ and $\bar{b}=b=M_\infty^2$, and 2) $\nu_1=1$ and $b=0$. Case 2 corresponds to a Prandtl-Glauert (P-G) transformation; Eqs. (6) then are the "P-G scaled" equations used in Ref. 13, and $\bar{b}$ is a transonic similarity parameter,

$$\bar{a}=\bar{a}_1\equiv(\tau/2)(\gamma+1)(1-M_\infty^2)^{-3/2}f(M_\infty)=(1/2)K^{-3/2} \tag{7}$$

Since Eqs. (1) are a special case of Eqs. (6), let us drop the overbars from all of the variables and parameters in Eqs. (6), with the understanding that u represents $\bar{u}$, etc.

The next transformation simplifies the development and analysis. Let

$$\tilde{u}=2au+b-1=2a(u-u_{CR}), \quad \tilde{v}=2av \tag{8}$$

where u_{CR} is sonic velocity. Then Eqs. (6) are simply

$$\tilde{q}_x+\tilde{v}_y=0 \tag{9a}$$

$$\nu_1\tilde{u}_y-\tilde{v}_x=0 \tag{9b}$$

where

$$\tilde{q}\equiv-\tilde{u}^2/2, \quad \tilde{q}_x=-\tilde{u}\tilde{u}_x \tag{10}$$

With $\tilde{u}=\phi_x$ and $\tilde{v}=\phi_y$, Eq. (9a) is equivalent to the model equation used by Murman[4] in his definitive development of fully conservative, first-order-accurate, type-dependent differencing.

The system of Eqs. (9) is of elliptic, parabolic, or hyperbolic type, depending on whether $\tilde{u}$ is negative, zero, or positive. In the solution of Eqs. (9), it is important to insure a proper transition from one type to another. For this purpose, following Murman,[4] note that the differential Eqs. (9), which are inconservative or conservation-law form, also may be represented by integral forms. With the flux vector $\boldsymbol{F}$ defined by

$$\boldsymbol{F}=\boldsymbol{e}_1\tilde{q}+\boldsymbol{e}_2\tilde{v} \tag{11}$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are unit vectors in the x and y directions, the divergence theorem applied to Eq. (9a) over an area becomes

$$\iint \nabla\cdot \boldsymbol{F}dA=\int \boldsymbol{F}\cdot \boldsymbol{n}ds=0 \tag{12}$$

Equations (9) are valid except across surfaces of discontinuity, where jump relations must be obtained from the integral form [Eq. (12)]. Furthermore, in a discretized treatment of Eqs. (9), the discretized form of Eq. (12) can be used for proper handling of any change of type, including smooth transitions between elliptic and hyperbolic forms.

III. General, Fully Conservative, Type-Dependent Difference Equations

General, fully conservative, type-dependent finite-difference equations that represent Eqs. (9), with transitions determined by Eq. (12), can be defined without regard to the order of accuracy. These equations may contain elliptic-point (E) or hyperbolic-point (H) operators, and, for the transition from elliptic to hyperbolic, a parabolic-point (P) operator, and, for hyperbolic to elliptic, a shock-point (S) operator (cf. Ref. 4). For this definition, it will be seen that one need define only 1) a suitable mesh, 2) an "elliptic-form" $\tilde{q}_E$ of the flux $\tilde{q}$, 3) a "hyperbolic-form" $\tilde{q}_H$ of the flux $\tilde{q}$, and 4) an appropriate test to determine if the representation of $\tilde{q}$ should be $\tilde{q}_E$ or $\tilde{q}_H$. Since $\tilde{q}_x$ in Eqs. (9) determines the type of point, the other terms can be central-differenced, denoted by subscript C. Thus the general, fully conservative, type-dependent difference equations representing Eqs. (9) may be written as

$$(\tilde{q}_x)_T+(\tilde{v}_y)_C=0 \tag{13a}$$

$$\nu_1(\tilde{u}_y)_C-(\tilde{v}_x)_C=0 \tag{13b}$$

where subscript T is replaced by E, H, P, or S, depending on the type of point at which Eq. (13a) is being solved.

For defining both the general and particular type-dependent operators, consider the mesh shown in Fig. 1. The indices j and k denote the x and y directions, respectively. The mesh is staggered in u and v, with velocity potential $\phi_{j,k}$ defined at points (dots in Fig. 1) horizontally between the $u_{j,k}$ points, and stream function $\psi_{j,k}$ defined at points (crosses on Fig. 1) horizontally between the $v_{j,k}$ points. The points $\phi_{j,k}$ and $\psi_{j,k}$ are, respectively, the points at which Eqs. (13a) and (13b) are satisfied. The mesh cell for the integral form of Eq. (13a) is the shaded area with $\phi_{j,k}$ at the center.

To define the type-dependent operators in terms of $\tilde{q}_E$ and $\tilde{q}_H$, consider now in Fig. 2a three mesh cells like the shaded area in Fig. 1. For whatever $\tilde{q}_E$ and $\tilde{q}_H$ are defined, and for whatever test determines the choice of $\tilde{q}_E$ or $\tilde{q}_H$, assume that an E point always has $\tilde{q}$ at both left and right boundaries represented by the form $\tilde{q}_E$ and that an H point always has $\tilde{q}$ at both left and right boundaries represented by the form

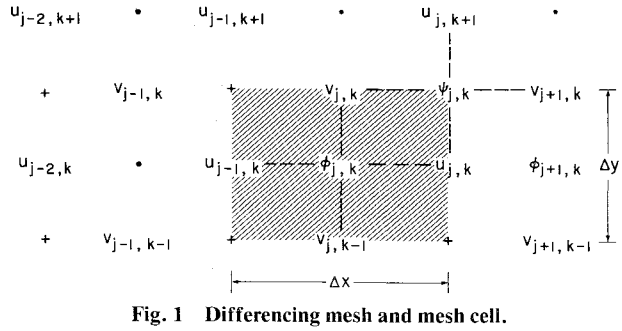

Fig. 1 Differencing mesh and mesh cell.

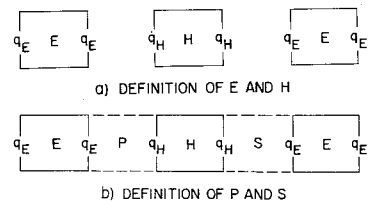

Fig. 2 Mesh cells and definition of operators.

$\tilde{q}_H$. Now define a P point as the center of a mesh cell that is a transition from E to H and an S point as the center of a mesh cell that is a transition from H to E [Fig. 2b]. Simply fill in the transition cells by the dashed lines on Fig. 2b, and require that $\tilde{q}$ on the upstream and downstream sides of a P point be represented, respectively, by $\tilde{q}_E$ and $\tilde{q}_H$, and for an S point, respectively, by $\tilde{q}_H$ and $\tilde{q}_E$. We thus obtain the general definitions of P and S points that satisfy the integral expression [Eq. (12)] for any collection of adjacent mesh cells, i.e.,

$$(\Delta\tilde{q})_E = (\tilde{q}_E)_{j,k} - (\tilde{q}_E)_{j-1,k} \tag{14a}$$

$$(\Delta\tilde{q})_H = (\tilde{q}_H)_{j,k} - (\tilde{q}_H)_{j-1,k} \tag{14b}$$

$$(\Delta\tilde{q})_P = (\tilde{q}_H)_{j,k} - (\tilde{q}_E)_{j-1,k} \tag{14c}$$

$$(\Delta\tilde{q})_S = (\tilde{q}_E)_{j,k} - (\tilde{q}_H)_{j-1,k} \tag{14d}$$

where $\tilde{q}$ on the right side of the mesh cell has subscript j and on the left $j-1$ (Fig. 1). The general type-dependent difference Eqs. (13) may be written simply as

$$(\Delta\tilde{q})_T/\Delta x + (\tilde{v}_{j,k} - \tilde{v}_{j,k-1})/\Delta y = 0 \tag{15a}$$

$$\nu_I(\tilde{u}_{j,k+1} - \tilde{u}_{j,k})/\Delta y - (\tilde{v}_{j+1,k} - \tilde{v}_{j,k})/\Delta x = 0 \tag{15b}$$

where

$$(\Delta\tilde{q})_T \equiv (\tilde{q}_G)_{j,k} - (\tilde{q}_G)_{j-1,k} \tag{16}$$

and where subscript G (for "generic," following Lax[18]) is either an H or an E, depending on a test to be specified at each j,k. Note that each $\tilde{v}_{j,k}$ and $\tilde{u}_{j,k}$ in Eqs. (15) could be replaced by some other symbols that represent those quantities and that would be conserved across cell boundaries and could be specified to represent the finite differences to higher order. The general method of conservative differencing was established by Lax and Wendroff,[19] and this method in which $\tilde{q}$ can change its form according to certain rules is an application of that method.

Summarizing at this point, 1) the definitions of $\tilde{q}_E$ and $\tilde{q}_H$, along with a test that specifies the choice between $\tilde{q}_E$ or $\tilde{q}_H$, determine all four operators, E, H, P, S [Eqs. (14) or (16)]; 2) the integral expression [Eq. (12)] is satisfied automatically for any collection of adjacent mesh cells; and 3) the order of accuracy of the operators will depend only on the specification of $\tilde{q}_E$ and $\tilde{q}_H$. It remains now to specify the test to choose between $\tilde{q}_E$ and $\tilde{q}_H$ at each mesh-cell boundary and to choose the forms of $\tilde{q}_E$ and $\tilde{q}_H$. The latter choice will depend on the order of accuracy desired and other considerations (Secs. IV and V).

Consider now the test to determine if $(\tilde{q}_G)_{j,k}$ at a mesh-cell boundary should be represented by $\tilde{q}_E$ or $\tilde{q}_H$. The simplest and possibly the most natural choice would be to assume that sonic velocity should occur within every P mesh cell and within every S mesh cell and not in E or H cells. In the shaded area of Fig. 1, then, a P point would require $\tilde{u}_{j-1,k} < 0$ and $\tilde{u}_{j,k} > 0$, and, for an S point, $\tilde{u}_{j-1,k} > 0$ and $\tilde{u}_{j,k} < 0$. The preceding definitions of E, H, P, and S points then apply only if

$$(\tilde{q}_G)_{j,k} = (\tilde{q}_E)_{j,k} \text{ for } \tilde{u}_{j,k} + \delta\tilde{u}_{j-1,k} < 0 \tag{17a}$$

$$= (\tilde{q}_H)_{j,k} \text{ for } \tilde{u}_{j,k} + \delta\tilde{u}_{j-1,k} > 0 \tag{17b}$$

(with $\delta = 0$) in Eq. (16). However, Murman's[4] tests are Eqs. (17) with $\delta = 1$, which were derived by considering the first-order-accurate, finite-difference equations and the stability of the difference operators in his relaxation method. A reviewer has pointed out (see also Ref. 20) that, at least for a normal shock, δ should be 1.0 in order to be compatible with the proper shock-jump relation at the S point. This can be seen by writing the inequality in Eq. (17a) on the right side of the mesh cell and the inequality in Eq. (17b) on the left side (where j is replaced by $j-1$). The results are

$$-(1/\delta)\tilde{u}_{j,k} > \tilde{u}_{j-1,k} > -\delta\tilde{u}_{j-2,k} \tag{18}$$

for an S point. As shown later in Sec. IV E, either for first-order accuracy or in the limit as $\Delta x \to 0$ for second-order accuracy, the correct normal shock jump occurs from $\tilde{u}_{j-2,k}$ to $\tilde{u}_{j,k}$ and is given by

$$\tilde{u}_{j,k} = -\tilde{u}_{j-2,k} \tag{19}$$

Thus, $\tilde{u}_{j-1,k}$ should be limited by

$$\tilde{u}_{j-2,k} \geq \tilde{u}_{j-1,k} \geq \tilde{u}_{j,k}$$

but this condition, along with Eq. (19), requires $\delta = 1$ in Eq. (18) for a normal shock. Nevertheless, it was decided to let δ be arbitrary in the test Eqs. (17) and investigate the effects of varying δ in the computations.

As done analogously by Bailey and Ballhaus[21] in their formulation of the improved Murman-Cole equations for first-order-accurate operators, one could, if desired, replace each $(\tilde{q}_G)_{j,k}$ in Eqs. (16) by

$$(\tilde{q}_G)_{j,k} = (1-\sigma_{j,k})(\tilde{q}_E)_{j,k} + \sigma_{j,k}(\tilde{q}_H)_{j,k} \tag{20}$$

where

$$\sigma_{j,k} = 0 \text{ for } \tilde{u}_{j,k} + \delta\tilde{u}_{j-1,k} < 0 \tag{21a}$$

$$= 1 \text{ for } \tilde{u}_{j,k} + \delta\tilde{u}_{j-1,k} > 0 \tag{21b}$$

IV. First- and Second-Order-Accurate Difference Equations

The fully conservative difference equations will be defined completely when the forms of $\tilde{q}_E$ and $\tilde{q}_H$ are defined. For the following developments, finite-difference formulas for derivatives of $\tilde{q} = -\tilde{u}^2/2$ at the point designated $\phi_{j,k}$ in Fig. 1 are considered. Note for this purpose that $\tilde{u}_{j,k}$, $\tilde{u}_{j-1,k}$, $\tilde{u}_{j-2,k}$, and $\tilde{u}_{j-3,k}$ are, respectively, at distances of $1/2$, $-1/2$, $-3/2$, and $-5/2$ times Δx from point $\phi_{j,k}$. All first- and second-order-accurate formulas to follow can be put in terms of $\phi_{j,k}$ on a conventional mesh (dots in Fig. 1) by replacing each $\tilde{u}_{j,k}$ and $\tilde{v}_{j,k}$ by the second-order-accurate relations:

$$\tilde{u}_{j,k} = (\phi_{j+1,k} - \phi_{j,k})/\Delta x, \quad \tilde{v}_{j,k} = (\phi_{j,k+1} - \phi_{j,k})/\Delta y \tag{22}$$

In choosing $\tilde{q}_E$ and $\tilde{q}_H$ as representations of $\tilde{q}$, we take the point of view that the order of accuracy to which $(\tilde{q}_E)_{j,k}$ or $(\tilde{q}_H)_{j,k}$ represents $\tilde{q}_{j,k}$ itself is not important (illustrated in Ref. 22 for an E point). The only requirements in defining $\tilde{q}_E$ and $\tilde{q}_H$ are 1) that the differencing be conservative, which is already assured by the foregoing definitions; 2) that the differences [Eq. (14)] in Eq. (15) give the order of accuracy desired; and 3) that the operators be consistent and stable in the proposed applications (Sec. V).

A. Elliptic Point (E)

For an elliptic point, the second-order-accurate central difference approximation is

$$(\tilde{q}_x)_E = (1/\Delta x)[-(\tfrac{1}{2})\tilde{u}^2_{j,k} + (\tfrac{1}{2})\tilde{u}^2_{j-1,k}] + O(\Delta x^2) \tag{23}$$

Comparison of $\Delta x(\tilde{q}_x)_E$ from Eq. (23) with $(\Delta\tilde{q})_E$ from Eq. (14a) leads to the choice

$$(\tilde{q}_{E2})_{j,k} = -\tfrac{1}{2}\tilde{u}^2_{j,k} \tag{24}$$

where the 2 following the subscript E denotes second-order

accuracy. This, of course, gives the second-order-accurate E -point difference equation from Eq. (15a) with Eq. (14a) which customarily is used.

B. First-Order Formulas (*H1, P1, S1*)

In the hyperbolic region, upwind differencing is appropriate. Following Murman and Cole,[2] one obtains $(\tilde{q}_x)_H$ by just shifting the elliptic differencing one mesh interval upstream so that

$$(\tilde{q}_x)_H = (1/\Delta x)\,(-\tfrac{1}{2}\tilde{u}^2_{j-1,k} + \tfrac{1}{2}\tilde{u}^2_{j-2,k}) + 0(\Delta x) \tag{25}$$

Comparison of $\Delta x(\tilde{q}_x)_H$ from Eq. (25) with $(\Delta\tilde{q})_H$ from Eq. (14b) leads to

$$(\tilde{q}_{H1})_{j,k} = -\tfrac{1}{2}\tilde{u}^2_{j-1,k} \tag{26}$$

Equation (15a) with this relation is equivalent to the Murman-Cole first-order-accurate, hyperbolic-point difference equation.

Since $\tilde{q}_E$ and $\tilde{q}_H$ now are defined for the first-order-accurate method, the parabolic-point and shock-point operators follow directly from Eq. (15a) with Eqs. (14c) and (14d):

$$(\Delta\tilde{q})_{P1} = -\tfrac{1}{2}\tilde{u}^2_{j-1,k} + \tfrac{1}{2}\tilde{u}^2_{j-1,k} = 0 \tag{27a}$$

$$(\Delta\tilde{q})_{S1} = -\tfrac{1}{2}\tilde{u}^2_{j,k} + \tfrac{1}{2}\tilde{u}^2_{j-2,k} \tag{27b}$$

which yield Krupp and Murman's parabolic-[3] and shock-point[4] operators. Thus, the four forms given by Eqs. (15) with Eqs. (14, 24, and 26) are equivalent to those of Murman in Ref. 4.

C. Second-Order Formulas (*H2a, P2a, S2a*)

The first-order-accurate, upwind difference formula in the foregoing uses values at the points in Fig. 1 designated $u_{j-1,k}$ and $u_{j-2,k}$. A second-order-accurate, upwind difference formula needs to use $u_{j-1,k}$, $u_{j-2,k}$, and $u_{j-3,k}$. However, the formula is not unique,[22] and so two forms, *H2a* and *H2b*, are considered. Each of these forms has its corresponding P and S forms.

Consider first the second-order-accurate, upwind difference expression:

$$(\tilde{q}_x)_H = -\tfrac{1}{2}(2\tilde{u}_{j-1,k} + \tilde{u}_{j-2,k} - \tilde{u}_{j-3,k})(2\tilde{u}_{j-1,k} - 3\tilde{u}_{j-2,k} + \tilde{u}_{j-3,k})/\Delta x + O(\Delta x^2) = -\tfrac{1}{2}[(2\tilde{u}_{j-1,k} - \tilde{u}_{j-2,k})^2 - (2\tilde{u}_{j-2,k} - \tilde{u}_{j-3,k})^2]/\Delta x + O(\Delta x^2) \tag{28}$$

Comparison of $\Delta x(\tilde{q}_x)_H$ from Eq. (28) with $(\Delta\tilde{q})_H$ from Eq. (14b) leads to

$$(\tilde{q}_{H2a})_{j,k} = -\tfrac{1}{2}(2\tilde{u}_{j-1,k} - \tilde{u}_{j-2,k})^2 \tag{29}$$

Equation (28), with substitution of Eqs. (22), is equivalent to the second-order-accurate, hyperbolic formula proposed by Murman and Cole.[2]

With Eqs. (24) and (29), the corresponding parabolic- and shock-point operators are obtained from Eqs. (14c) and (14d):

$$(\Delta\tilde{q})_{P2a} = -\tfrac{1}{2}[(2\tilde{u}_{j-1,k} - \tilde{u}_{j-2,k})^2 - \tilde{u}^2_{j-1,k}] \tag{30a}$$

$$(\Delta\tilde{q})_{S2a} = -\tfrac{1}{2}[\tilde{u}^2_{j,k} - (2\tilde{u}_{j-2,k} - \tilde{u}_{j-3,k})^2] \tag{30b}$$

We thus have a complete system (E, H, P, and S) of second-order-accurate difference equations given by Eqs. (15) with (14, 24, and 29). Properties of this scheme are discussed in subsections below and in Secs. V and VI.

D. Second-Order Formulas (*H2b, P2b, S2b*)

The most straightforward derivation of the upwind difference for $(\tilde{u}^2)_x$ to second-order accuracy yields

$$(\tilde{q}_x)_H = -\tfrac{1}{2}[(2\tilde{u}^2_{j-1,k} - \tilde{u}^2_{j-2,k}) - (2\tilde{u}^2_{j-2,k} - \tilde{u}^2_{j-3,k})]/\Delta x + O(\Delta x^2) \tag{31}$$

Comparison of $\Delta x(\tilde{q}_x)_H$ from Eq. (31) with $(\Delta\tilde{q})_H$ from Eq. (14b) leads to

$$(\tilde{q}_{H2b})_{j,k} = -\tfrac{1}{2}(2\tilde{u}^2_{j-1,k} - \tilde{u}^2_{j-2,k}) \tag{32}$$

$$(\Delta\tilde{q})_{P2b} = -\tfrac{1}{2}(\tilde{u}^2_{j-1,k} - \tilde{u}^2_{j-2,k}) \tag{33}$$

$$(\Delta\tilde{q})_{S2b} = -\tfrac{1}{2}(\tilde{u}^2_{j,k} - 2\tilde{u}^2_{j-2,k} + \tilde{u}^2_{j-3,k}) \tag{34}$$

The complete set of equations for E, H, P, and S points for this scheme thus is given by Eqs. (15) with (14, 24, and 32). This scheme is discussed further in the following and in Secs. V and VI.†

E. Results for Normal Shock Waves

The correct normal-shock-wave solution[4] between two uniform flows is $\tilde{u}_2 = -\tilde{u}_1$, where subscripts 1 and 2 are before and after the shock, respectively. By setting $\Delta\tilde{v} = 0$ in (15a) to obtain $(\Delta\tilde{q})_S = 0$, or, from Eq. (14d),

$$(\tilde{q}_E)_{j,k} = (\tilde{q}_H)_{j-1,k} \tag{35}$$

at S point, we can, following Murman,[4] check the shock-point operators to insure the correct normal-shock jumps as $\Delta x \to 0$. Murman's operator *S1* from Eq. (27b) then gives $\tilde{u}^2_{j,k} = \tilde{u}^2_{j-2,k}$, from which follows Eq. (19). Thus, for an *S1* point, the correct jump across a normal shock occurs over two mesh intervals. For an *S2a* point, Eq. (30b) gives $\tilde{u}_{j,k} = -(2\tilde{u}_{j-2,k} - \tilde{u}_{j-3,k})$. As $\Delta x \to 0$, $\tilde{u}_{j-3,k} \sim \tilde{u}_{j-2,k}$, so Eq. (19) is satisfied also for an *S2a* point. For an *S2b* point, Eq. (34) gives $\tilde{u}_{j,k} = -(2\tilde{u}^2_{j-2,k} - \tilde{u}^2_{j-3,k})^{1/2}$. Again, as $\Delta x \to 0$, $\tilde{u}_{j-3,k} \sim \tilde{u}_{j-2,k}$ so Eq. (19) is satisfied for an *S2b* point.

F. Difference Equations in Terms of u and v

The finite-difference equations derived previously are converted back to u, v by the transformation of Eqs. (8) and summarized as follows. If we let

$$\mu \equiv \Delta y/\Delta x \tag{36}$$

Eqs. (15a) and (15b) are

$$(\Delta q)_T + \mu^{-1}(v_{j,k} - v_{j,k-1}) = 0 \tag{37a}$$

$$v_1(u_{j,k+1} - u_{j,k}) - \mu(v_{j+1,k} - v_{j,k}) = 0 \tag{37b}$$

where

$$(\Delta q)_T = (q_G)_{j,k} - (q_G)_{j-1,k} \tag{38a}$$

$$(q_G)_{j,k} = [(1-b)u_G - a(u^2_G)]_{j,k} \tag{38b}$$

and where if

$$\frac{u_{j,k} + \delta u_{j-1,k}}{1+\delta} < u_{CR}: (u_G)_{j,k} = (u_E)_{j,k}, (u^2_G)_{j,k} = (u^2_E)_{j,k} \tag{39a}$$

$$\frac{u_{j,k} + \delta u_{j-1,k}}{1+\delta} > u_{CR}: (u_G)_{j,k} = (u_H)_{j,k}, (u^2_G)_{j,k} = (u^2_H)_{j,k} \tag{39b}$$

with

$$(u_{E2})_{j,k} = u_{j,k}, \quad (u^2_{E2})_{j,k} = (u_{j,k})^2 \tag{40a}$$

†A second-order scheme equivalent to the 2*b* formulation has been derived independently in Ref. 23.

$$(u_{H1})_{j,k}=u_{j-1,k},\quad (u^2_{H1})_{j,k}=(u_{j-1,k})^2 \tag{40b}$$

$$(u_{H2a})_{j,k}=2u_{j-1,k}-u_{j-2k},$$

$$(u^2_{H2a})_{j,k}=(2u_{j-1,k}-u_{j-2,k})^2 \tag{40c}$$

$$(u_{H2b})_{j,k}=2u_{j-1,k}-u_{j-2,k}$$

$$(u^2_{H2b})_{j,k}=2(u_{j-1,k})^2-(u_{j-2,k})^2 \tag{40d}$$

Note that $(u^2_{H2b})_{j,k}\neq(u_{H2b})^2_{j,k}$ in Eq. (40d)

V. Semidirect Method with Stabilizing Terms

A. Iteration Method 1

A basic feature of the type-dependent, semidirect iterative method used in Ref. 13 (where $\bar{b}$ was not arbitrary, $\nu_1=1$, and only the first order H-point operator was used) was the shifting of $(\Delta q)_\tau$ to the right side of (37a) and the insertion of $\Delta x(u_x)_C=u_{j,k}-u_{j,k}-u_{j-1,k}$ on both the left and the right sides, so that the difference operator on the left was always elliptic. Then subscript n, denoting the nth iteration, was put on the left side and $n-1$ on the right side. The iteration was performed by solving the elliptic difference operator on the left using the direct Cauchy-Riemann solver,[14] with the type-dependent right side, known from iteration $n-1$, as a source term. A relaxation parameter was introduced (an additional relaxation parameter in the present treatment) by replacing each u_n on the left side by u_n^* and, after each iteraction, determining the nth iterate u_n from

$$u_n=(1-\nu_2)u_{n-1}+\nu_2 u_n^* \tag{41}$$

To stabilize the iterations by this method at strongly supercritical conditions, Ref. 15 introduced extra terms into both sides of the mass-conservation Eq. (37a), which made the iterations timelike. An extended form[16] of the Cauchy-Riemann solver is used to treat the elliptic difference operator with the extra terms.

In the present formulation, including the fully conservative, second-order-accurate operators as options, Eqs. (37) representing Eqs. (6) with the just-described iteration scheme become

$$(1-\alpha_1)(u^*_{j,k})_n-(1+\alpha_2)(u^*_{j-1,k})_n+\mu^{-1}(v_{j,k}-v_{j,k-1})_n$$
$$=(1-\alpha_1)(u_{j,k})_{n-1}-(1+\alpha_2)(u_{j-1,k})_{n-1}-[(\Delta q)_T]_{n-1} \tag{42a}$$

$$-(u^*_{j,k})_n+(u^*_{j,k+1})_n+\mu(v_{j,k}-v_{j+1,k})_n$$
$$=(1-\nu_1)(u_{j,k+1}-u_{j,k})_{n-1} \tag{42b}$$

where $\alpha_1 u_{j,k}$ and $\alpha_2 u_{j-1,k}$ are the extra added terms that can stabilize the iteration and that cancel out at convergence. If α_1 is different from α_2, then the added terms that make the iteration timelike, when combined, are "off center." The parameters α_1 and α_2 are arbitrary and are to be chosen for best iterative convergence. The term $[(\Delta q)_T]_{n-1}$ is given by Eqs. (38-40), with either of *H1, H2a,* or *H2b* in Eqs. (40) being selected for H in Eqs. (39). The additional relaxation parameter ν_2, with Eq. (41), has been introduced because it can be varied from point to point and from one iteration to the next, whereas ν_1, as part of the transformation, is constant. If only this iteration method (method 1) were used, one then could set $\nu_1=1$ everywhere and just use ν_2 and $\bar{b}$ (as well as α_1 and α_2) to obtain best convergence. However, ν_1 is included for switching to an extrapolation technique, method 2 (see Ref. 15).

B. Accuracy, Consistency, and Stability

Although the order of accuracy and the consistency of the E and H operators is known from the preceding derivations, the P and S operators should be checked for order of accuracy and consistency. A simplified heuristic stability analysis also can be made to obtain a rough indication of the stability of the iteration scheme for each set of operators.

For this study, one first can define an artificial time t, its increment Δt, and a parameter α that relates the artificial time scale to the mesh interval Δx by

$$t=n\Delta t,\quad \alpha=(\alpha_1+\alpha_2)\Delta t/\Delta x \tag{43}$$

It is most convenient to treat the difference Eqs. (42) with the stabilizing terms added, but transformed into $\tilde{u}$, $\tilde{v}$ by Eqs. (8). One then can expand formally the terms of the difference equations in Taylor series in both Δx and Δt. The stabilizing terms added to the difference equations resulting in Eqs. (42) contribute to terms involving u_t, u_{tt}, u_{xt}, etc. Truncation errors in the difference equations also contribute to the higher derivatives, and Taylor-series expansions of the difference equations can be used to study the resulting accuracy, consistency, and stability. The accuracy and consistency of the difference operators derived in Secs. III and IV, without regard to the particular iteration scheme, can be obtained from the following analysis simply by setting all time derivatives equal to zero.

For simplicity in this discussion, assume that $\nu_1=\nu_2=1$. Then the differential equations resulting from the Taylor series expansions in Δx and Δt of the terms in Eq. (42a), corresponding to the different choices for $(\Delta q)_T$ given by Eqs. (38), are written conveniently in terms of the following differential operator:

$$D_2(\tilde{u})\equiv\alpha\tilde{u}_t+\tilde{u}\tilde{u}_x-\tilde{v}_y+(\Delta t/2)[-\alpha\tilde{u}_{tt}$$
$$+(\alpha_1-\alpha_2-2)\tilde{u}_{xt}-2(\tilde{u}\tilde{u}_x)_t] \tag{44}$$

The results are [assuming $\Delta t=O(\Delta x)$ as $\Delta x\to 0$]

E2: $$D_2(\tilde{u})=O(\Delta x^2) \tag{45a}$$

H1: $$D_2(\tilde{u})=\Delta x(\tilde{u}\tilde{u}_x)_x+O(\Delta x^2) \tag{45b}$$

P1: $$D_2(\tilde{u})=\tilde{u}\tilde{u}_x-\Delta t(\tilde{u}\tilde{u}_x)_t+O(\Delta x^2) \tag{45c}$$

S1: $$D_2(\tilde{u})=-\tilde{u}\tilde{u}_x+\Delta t(\tilde{u}\tilde{u}_x)_t+\Delta x(\tilde{u}\tilde{u}_x)_x+O(\Delta x^2) \tag{45d}$$

H2a: $$D_2(\tilde{u})=O(\Delta x^2) \tag{45e}$$

P2a: $$D_2(\tilde{u})=\Delta x(\tilde{u}\tilde{u}_{xx})+O(\Delta x^2) \tag{45f}$$

S2a: $$D_2(\tilde{u})=-\Delta x(\tilde{u}\tilde{u}_{xx})+O(\Delta x^2) \tag{45g}$$

H2b: $$D_2(\tilde{u})=O(\Delta x^2) \tag{45h}$$

P2b: $$D_2(\tilde{u})=\Delta x(\tilde{u}\tilde{u}_x)_x+O(\Delta x^2) \tag{45i}$$

S2b: $$D_2(\tilde{u})=-\Delta x(\tilde{u}\tilde{u}_x)_x+O(\Delta x^2) \tag{45j}$$

Consistency simply requires that, as $\Delta x\to 0$ and as time derivatives vanish, Eqs. (45) reduce to Eq. (9a): $\tilde{u}\tilde{u}_x-\tilde{v}_y=0$. From Eqs. (45a) and (45b), both the *E2* and the Murman-Cole *H1* operators are consistent with Eq. (9a). Furthermore, as explained by Murman,[4] both the Krupp-Murman parabolic (*P1*) and Murman shock-point (*S1*) operators are consistent for smooth acceleration through sonic and smooth recompressions through sonic velocity because $\tilde{u}=0+O(\Delta x)$ at points very near there. At a shock point in a sharp recompression, the consistency of Eq. (45d) is not pertinent; only the correct shock jump must be guaranteed, as just described. We thus see that *E2* is second-order accurate and *H1, P1,* and *S1* are first-order accurate at the points of their intended use.

For the second-order-accurate, 2*a*, formulation, Eqs. (45e-45g) show that the *H2a, P2a,* and *S2a* operators are not only consistent with Eq. (9a) but that, as $\Delta x\to 0$ and for vanishing

time derivatives, *H2a* is second-order accurate in hyperbolic regions, *P2a* is second-order accurate at a parabolic point, and *S2a* also is second-order accurate at a shock point in a smooth recompression. For the second-order-accurate, 2*b*, formulation, Eqs. (45h-45j) show that the *H2b*, *P2b*, and *S2b* operators are consistent with Eq. (9a) as $\Delta x \to 0$. However, although *H2b* is clearly second-order accurate as $\Delta x \to 0$ and time derivatives vanish, both *P2b* and *S2b* contain $\Delta x(\tilde{u}\tilde{u}_x)_x = \Delta x(\tilde{u}\tilde{u}_{xx} + \tilde{u}_x^2)$, which is only $O(\Delta x)$ near points where $\tilde{u} = 0 + O(\Delta x)$. Therefore, the *P2b* and *S2b* operators are only first-order accurate. This will be evident also in the computed results of Sec. VI.

Heuristic stability considerations are discussed in Ref. 22. It is indicated that the iteration method just described should be stable, and thus converge, for some range of values of α_1 and α_2, for all of the differencing schemes represented by Eqs. (45).

C. Artificial Viscosity

The *H2* operators are nondissipative to first order, and experience shows that the iterative convergence of the second-order-accurate methods is very slow. Therefore, two methods of adding artificial viscosity to the hyperbolic regions (retaining formal second-order accuracy) are considered. Both methods add artificial viscosity in such a way that the method is still fully conservative; i.e., the components of F in Eq. (11) still cancel out at adjacent mesh-cell boundaries, and the transition is smooth at P and S points.

The first method is obtained by observing that, since the *H1* operator is dissipative to first order, one could replace $\tilde{q}_{H2}$ given by either Eq. (29) or (32) by (cf. Ref. 6)

$$(\tilde{q}_{H2V})_{j,k} = (1 - \kappa\Delta x)(\tilde{q}_{H2})_{j,k} + \kappa\Delta x(\tilde{q}_{H1})_{j,k} \tag{46a}$$

The results are that both u_{H2} and u_{H2}^2 in Eqs. (40) are replaced by

$$(u_{H2V})_{j,k} = (1 - \kappa\Delta x)(u_{H2})_{j,k} + \kappa\Delta x(u_{H1})_{j,k} \tag{46b}$$

$$(u_{H2V}^2)_{j,k} = (1 - \kappa\Delta x)(u_{H2}^2)_{j,k} + \kappa\Delta x(u_{H1}^2)_{j,k} \tag{46c}$$

and the effects are that $(\tilde{q}_x)_H$ contains a truncation error of order Δx higher than that of $(\tilde{q}_x)_{H1}$ in Eq. (25), but that the truncation error in Eq. (45e) or (45h) contains a term like the first term in Eq. (45b) but multiplied by $\kappa\Delta x$. (This method is not adaptable to the first-order-accurate formulation.)

The second method is obtained by noting that the dissipation in the first-order method (cf. Ref. 24) from Eq. (45b) takes the form

$$\tilde{q}_x + \tilde{v}_y = \Delta x\tilde{q}_{xx} + O(\Delta x^2)$$

Therefore, this can be put into a higher-order dissipative form by multiplying the right side by $\kappa\Delta x$ so that

$$[\tilde{q} - \kappa(\Delta x)^2\tilde{q}_x]_x + \tilde{v}_y = O(\Delta x^2)$$

This, is equivalent to replacing $\tilde{q}$ by $\tilde{q} - \kappa(\Delta x^2)\tilde{q}_x$ or replacing u_H and u_H^2 in Eqs. (40) by

$$(u_{HV})_{j,k} = (u_H)_{j,k} - (\kappa\Delta x)(u_{j-1,k} - u_{j-2,k}) \tag{47a}$$

$$(u_{HV}^2)_{j,k} = (u_H^2)_{j,k} - (\kappa\Delta x)(u_{j-1,k}^2 - u_{j-2,k}^2) \tag{47b}$$

This method can be applied to the second-order Eqs. (40c) or (40d) using $\kappa = O(1)$, or it can be adapted to the first-order formulation by making κ larger. This method does not appear to be as effective when used with the present iteration technique as the method of Eqs. (46) for the second-order-accurate formulas.

D. Method 2

In Refs. 13 and 15, two versions of an extrapolation technique are described. The reader is referred to Ref. 13 for the most detailed description and to Ref. 15 for a revised version that allows the method to be used after iteration method 1 described previously has nearly converged at some points in the computation field. Method 2 constitutes a new approach to applying the Aitken/Shanks formula[25,26] either for accelerating convergence or for extrapolating to a final solution using three successive iterates at each point. The formulation in Ref. 15 includes v_1 in the transformed equations [obtained from Eqs. (4) and (5)]. Details are not inculded here because they are available in Ref. 15, and, at the time of this writing, the second-order-accurate formulas have not been included yet in method 2.

VI. Results of Computations

A research computer program has been written to solve the transonic small-disturbance equations for a thin symmetrical parabolic-arc airfoil. The linearized boundary condition on v is applied at $y = 0$, and a far-field solution for u is applied on the upstream ($x = x_\ell$) and upper ($y = y_u$) boundaries and for v on the downstream ($x = x_r$) boundary. For present purposes, the condition applied at these outer boundaries was in all cases simply the Prandtl-Glauert solution for a biconvex airfoil as specified in Ref. 13. The program includes the options for first- or second-order accuracy in method 1 and the option of switching to method 2 after some iteration n for the first-order-accurate formulation. A conversational version of the program, for interacting with the program, was run on an IBM 360/67 computer; but the program was run on a Control Data 7600 computer with FTN (OPT = 2) compiler for measuring all quoted computing times. The program would require only two two-dimensional arrays (for u and v) if only method 1 with $v_2 = 1$ were used. Use of $v_2 \neq 1$ requires a third array, and use of method 2 requires a total of four two-dimensional arrays.

For the biconvex-airfoil problem, results described below were obtained at four transonic conditions which, with $f(M_\infty) = M_\infty^2$ in Eqs. (3) and with thickness ratio $\tau = 0.10$, correspond to $M_\infty = 0.800$, 0.825, 0.850, and 0.875. All cases were run with x_ℓ and x_r at about ½ chord length from the leading and trailing edges, respectively. The first two Mach numbers were run with $y_u = 3.5$ and the latter two with $y_u = 5.0$. In the comparisons below with the program of Ref. 1, which uses the far-field condition of Klunker,[27] the "variable mesh" had upstream and downstream boundaries, respectively, at 1 chord and 0.875 chord from the airfoil, with the lateral boundaries at 5.2 chords. The parameter option $k = 2$ in Ref. 1 was chosen corresponding to $f(M_\infty) = M_\infty^2$ for proper comparisons. For the timing comparison runs to be described, the program of Ref. 1 used a uniform mesh with the same boundaries as described previously for the present method.

A. Pressure Distributions

Pressure distributions on the biconvex airfoil computed using method 1 on a 39×32 mesh ($\Delta x = 0.05$) are shown plotted in Fig. 3. Both first- and second-order-accurate, 2*a*, computations are shown and are compared with results from the TSFOIL program of Murman et al.[1] using their refined variable mesh. On the rather coarse 39×32 mesh, the present first-order-accurate results compare well. The slight jump in the values behind the parabolic point in the first-order results is a property of the Krupp-Murman parabolic-point operator. The second-order-accurate results (using the 2*a* operators) tend less to smear over the shock than the first order. The transition at the *P2a* point is very smooth, since the operator is second-order accurate (Sec. V); for this reason, the supersonic results, which lie slightly below the variable-mesh results of Ref. 1, may be even more accurate than those results except right at the shock wave, where the finer mesh is better.

For the best illustration of the properties of the second-order schemes, pressure distributions on a very coarse (19×32) mesh are shown plotted in Fig. 4. As expected on

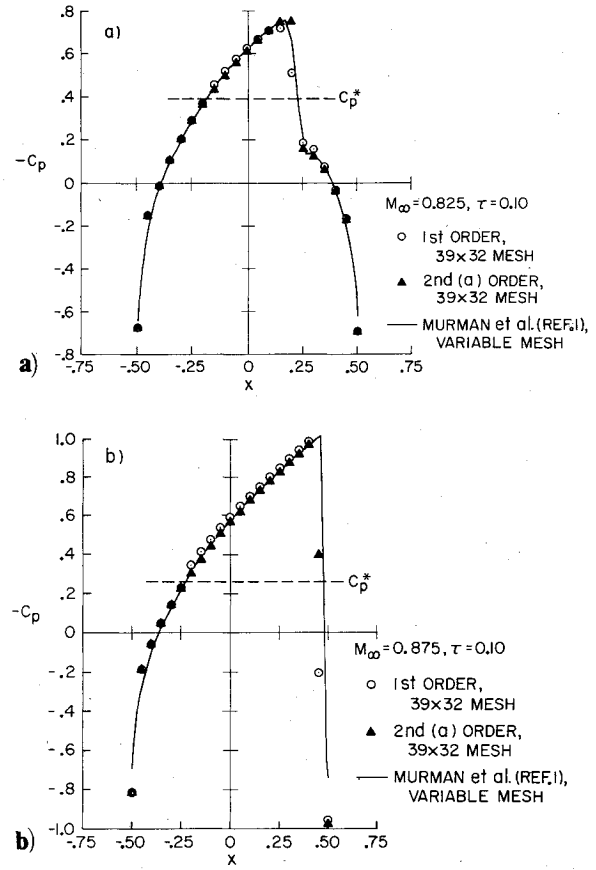

Fig. 3 Pressure on a thin biconvex airfoil.

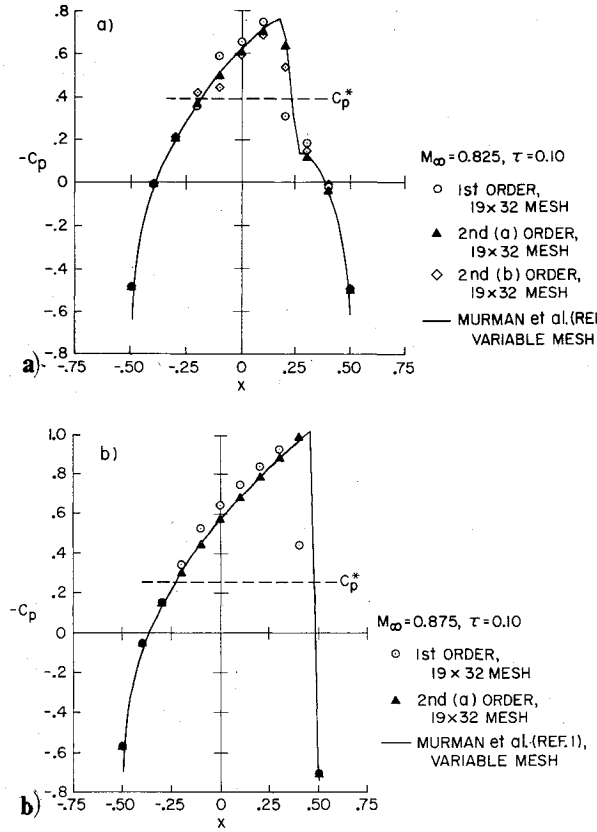

Fig. 4 Pressure on a thin biconvex airfoil (very coarse mesh).

such a coarse mesh, with $\Delta x = 0.10$, the first-order-accurate results shown by circles are very inaccurate in the supersonic region and smear over the shock very crudely. The anomalous jump behind the parabolic point is exaggerated by this coarse mesh, and the extremely smooth behavior of the *2a* results also is emphasized. The *2a* results are accurate, with the C_p values being nearly identical (except for some in the shock region) to the values computed on the finer (39×32) mesh of Fig. 3. Note that in this shock-capturing method the "shock jumps" are exact only in the limit as $\Delta x \to 0$, but the values of C_p at mesh points on both sides of the shock are computed surprisingly accurately. If for some reason, however, one needs the values of the shock jumps computed "exactly" (within the limits of the small-disturbance approximation), rather than just a reasonably accurate representation of the overall pressure distribution when using a coarse mesh, he may want to consider shock fitting.

Finally, Fig. 4a also illustrates the anomalous behavior of the *P2b* operator on a coarse mesh. It may be possible to remove this behavior, but the *P2b* operator remains first-order accurate (Sec. V). From the computations made, the *2b* operators also appear to be less stable than the *2a*.

B. Tentative Evaluation of Free Parameters

The conversational computer program was used for numerical experiments to determine the best values, for most rapid convergence, of the free parameters in the program: v_1, v_2, $\bar{b}$, α_1, α_2, and δ. The interaction of the various parameters is very complex. After considerable experimenting, the values are still tentative.

Unless method 2 is to be used in a given case for extrapolation, it is simplest to set v_1 at unity. However, if method 2 is to be used, then $v_1 = 0.9$ appears to give best results. The most appropriate assignment of v_2 appears to be to set $v_2 = v_{2E}$ for all subsonic values of $u_{j,k}$ except just in front of *P* points and just behind *S* points. At those points, as well as for all supersonic $u_{j,k}$, v_2 is set at v_{2H}. The best value of v_{2H} appears to be 0.9, and v_{2E} is best left at unity, or set at 1.2 for several iterations (say, until the maximum normalized residual becomes less than 0.05) and then set at 1.0.

The best value for $\bar{b}$ appears to be zero. This is true at least for all subcritical and low supercritical Mach numbers. However, there has not been enough experimenting with this parameter at higher Mach numbers.

The evaluation of α_1 and α_2 appears to be the most critical to the stability of the iteration scheme and to the rate of convergence. From the heuristic stability analysis[15] and the preliminary numerical experiments, it appears best to keep α at zero and use α_2 as the primary free parameter to be varied. One needs α_2 large enough for stability of the iteration but not large enough to cause too much temporal damping.

For this and subsequent discussions, define a relative normalized error (or residual) at point *j,k* by

$$E_r = [(u_{j,k})_n - (u_{j,k})_{n-1}] / (|u|_{\max})_{n-1} \tag{48}$$

and the maximum value of this over all *j,k* at given *n* by $(E_r)_{\max}$.

The following described conditions all refer to $f(M_\infty) = M_\infty^2$ in Eq. (3) and $\tau = 0.10$. All values of α_2 are for the 39×32 mesh and for the first-order-accurate *H1* operator. Other meshes may have different best values of α_2. At $\tau = 0.10$, for all $M_\infty \le 0.800$ ($\bar{a}_1 \le 0.3556$), α_2 is kept at zero. At $M_\infty = 0.825$ ($\bar{a}_1 = 0.4525$), α_2 is started at 0.5; then, after $(E_r)_{\max}$ becomes less than 0.01, α_2 is changed to 1.0. At $M_\infty = 0.850$ ($\bar{a}_1 = 0.5931$), α_2 is started at 1.0; after $n = 16$, α_2 is changed to 2.0; after $(E_r)_{\max} < 0.01$, α_2 is changed to 3.0. At $M_\infty = 0.875$ ($\bar{a}_1 = 0.8097$), α_2 is started at 1.0; after $n = 10$, α_2 is changed to 2.0. When E_r becomes less than 0.001 at all but three or fewer points, α_2 is changed to 3.0. At conditions with other values of $\bar{a}_1$, one can interpolate between the stated variations of α_2 according to the value of $\bar{a}_1$.

For stability and iterative convergence, the value $\delta = 1$ in Eqs. (17, 21, and 39) gives either the same or better results

than $\delta = 0$. Since $\delta = 1$ is more reliable in producing a rapidly converging solution, and since it corresponds to arguments given by Murman[4] for the *H1* operator as well as arguments in Sec. III, $\delta = 1$ is recommended for all cases.

C. Iteration Counts and Computing Times

Computing times for the first-order-accurate cases are compared here with computing times from the line-relaxation program TSFOIL[1] using a mesh and other conditions that are as close to being "equivalent" to the present program as possible. The different capabilities and different properties of the two methods (and programs) require numerous qualifications to be made, and an attempt is made to do this fairly. Both programs were run on the Control Data 7600 computer with FTN (OPT = 2) compiler.

The program TSFOIL has the capability of using a three-stage mesh refinement, which is effective in decreasing the total computing time. Since this procedure also could be applied to the present method but is not included in the research program, only the single-mesh option of TSFOIL is used for the comparison. Similarly, the program TSFOIL has a variable-mesh capability, which also is effective in providing accurate results with fewer mesh points (and thus shorter time) than required by a uniform mesh. To some extent (and without qualification in one direction), the present method also could employ a variable mesh, and this is expected to be included in the future. However, since the present program is restricted to a uniform mesh, the variable-mesh option of TSFOIL is not used for the timing comparisons.

Table 1 shows the number of iterations n and the corresponding computing times t_n to obtain sufficiently converged solutions on a 39×32 mesh. Because of the different convergence properties of the two methods (TSFOIL being a line-relaxation method for the velocity potential, and the present method being a semidirect iteration in terms of u and v), different tests determine when an acceptable solution has been obtained. This makes the comparison difficult, but the most appropriate test for each method is used. The program TSFOIL stops after a maximum difference in (scaled but not normalized) velocity potential between successive iterations ($|\Delta\phi|_{\max}$) becomes less than some small number (their FORTRAN parameter CVERGE). For the cases run, their results for C_p changed significantly from $|\Delta\phi|_{\max}$ of 10^{-3} to 10^{-4}, but not between 10^{-4} and 10^{-5}. Therefore, CVERGE $= 10^{-4}$ was used (even though the authors of Ref. 1 recommend the more stringent 10^{-5}). On the right side of Table 1 are values of n for this test to be satisfied for the program TSFOIL on an "equivalent mesh," discussed below. In the present method, after E_r became less than 10^{-3} at all but three or fewer points, the solutions for C_p were graphically indistinguishable from fully converged solutions except for, at the highest Mach numbers, points in the shock wave which may be changing slightly. Therefore, this test is the one used for timing the iteration method 1. It is found that the slightly changing values in the shock do not have a significant effect on the rest of the solution. Such a condition in the semidirect method may be more acceptable than in line relaxation because the effect of any error or change at each iteration is felt at all points simultaneously in the next succeeding iteration, whereas a line-relaxation method requires a certain number of iterations for the errors to propagate, reflect, and evantually spread and damp out. In the present method, the solution, except within the shock, stops changing significantly rather quickly. (The plots in Fig. 3 are fully converged, however.) Corresponding iteration counts (n) obtained by switching to method 2 are listed in the middle of Table 1. For example, at $M_\infty = 0.825$, $\tau = 0.10$, after nine iterations by method 1, the next three iterations by method 2 were used to extrapolate to the final solution, as described in Ref. 15.

The determination of computing times listed in Table 1 needs some elaboration. The total computing time t_n for n iterations in each case is obtained by multiplying n by Δt_n, the time per iteration. To determine the appropriate Δt_n used by TSFOIL for comparison with the present method, one must recognize that TSFOIL is set up for an arbitrary airfoil at angle of attack, and so the airfoil is in the middle of the mesh, whereas, for the present method, the symmetrical airfoil at zero incidence is along one edge of the mesh. Therefore, the "equivalent mesh" used by TSFOIL in measuring n has the same physical dimensions and number of mesh points on one side of the airfoil as in the present method. However, the "equivalent mesh" used for measuring Δt_n in TSFOIL had the same total number of mesh points on both sides together as the present method had on one side. Thus, although the time per iteration in the cases for which n is listed in Table 1 for TSFOIL was 0.033 sec, the equivalent Δt_n for the "timing mesh" was $\Delta t_n = 0.017$ sec, which was used to obtain t_{nc}. The lifting-airfoil capability of TSFOIL requires some computing time to satisfy the Kutta condition at the trailing edge, and this capability and extra time are superfluous for the comparison.

Two columns of computing times are listed in Table 1 for both methods 1 and 2. These correspond to two different values for Δt_n in the present method and represent the uncertainty in a fair evaluation of the computing time of the method because of very inefficient coding in the research computer program. The inefficiency is due partly to inclusion of many options (some of little use) for research purposes; these involve many FORTRAN conditional control statements, some within loops. Several facts indicate the coding inefficiency: first, the program as presently written requires 0.040 sec/iteration on a 39×32 mesh (resulting in values of t_{na} in Table 1), but only 0.014 sec of this is used for the direct solver itself, indicating an excessive amount of "overhead." Second, it was noticed that, in switching to method 2, the Δt_n drops from 0.040 to 0.030 sec for no immediately apparent reason. Thus, it is estimated that streamlining the program would result in a reduction of Δt_n to 0.020 sec or less. For this reason, the corresponding values of t_{nb} are listed in Table 1.

Use of the semidirect method does not depend on use of any particular direct solver. Significantly faster direct solvers currently are being developed, and, when available, these could replace the cylic-reduction procedure used in the Cauchy-Riemann solver[14,16] for reducing the large sparse matrix that is involved. Thus, substantial improvements in the efficiency of computations by the present method are anticipated.

VII. Concluding Remarks

The fast semidirect iterative method previously introduced has been extended both to apply to strongly supercritical conditions and to include full second-order accuracy in the tran-

Table 1 Iteration counts and computing times[a]

	Method 1			Method 2			TSFOIL	
M_∞ ($\tau = 0.10$)	n	t_{na}, sec	t_{nb}, sec	n	t_{na}, sec	t_{nb}, sec	n	t_{nc}, sec
0.800	12	0.48	(0.24)	6	0.24	(0.12)	90	1.53
0.825	20	0.80	(0.40)	12	0.45	(0.23)	100	1.70
0.850	35	1.40	(0.70)	...	...	...	130	2.21
0.875	35	1.40	(0.70)	...	...	...	450	7.65

[a] t_{na} = time with $\Delta t_n = 0.040$ sec; t_{nb} = time with $\Delta t_n = 0.020$ sec; t_{nc} = time with $\Delta t_n = 0.017$ sec.

sonic regime. For the extensions, general, fully conservative, type-dependent, finite-difference equations have been formulated to represent the nonlinear transonic small-disturbance equations. The preferred second-order scheme 1) maintains second-order accuracy through smooth transitions between elliptic and hyperbolic regions, 2) is consistent with the integral equation to give the correct shock locations and jumps in the limit $\Delta x \to 0$, and 3) is stable in the present method for the examples computed. The main contribution for stabilizing the semidirect iteration method is the noncentered addition of the terms that become effectively αu_t plus higher derivatives in x and t. The choice of free parameters determines not only α but also the degree to which the representation of u_t is off center for best convergence.

Computed pressure distributions, obtained for a range of transonic conditions on a biconvex airfoil, compare well with results from a line-relaxation program (TSFOIL) for the improved Murman-Cole method. With use of either first- or second-order-accurate operators, the shock jumps are correct to the specified order of accuracy (depending on Δx), and the shocks are located properly. The second-order-accurate scheme produces highly accurate results for C_p on a very coarse mesh, including both relatively sharp shock-wave transitions and completely smooth transitions at parabolic points.

Timing comparisons between the present method and the line-relaxation method, with carefully chosen "equivalent meshes" and conditions in the program TSFOIL for fair comparison and with numerous qualifications, indicate that the computations by the present method are significantly faster than by the conventional line-relaxation method. The computing times are expected to be shortened significantly by improving the presently inefficient coding, by possible improvements in specifying free parameters to get shorter iteration counts, and by using faster direct solvers. The very short computing times indicate a significant potential for use in highly efficient transonic flow computations.

The results that have been presented are for very simple cases and hence are not sufficient to validate the method thoroughly. For example, the capability of the second-order-accurate scheme in reproducing the singularity at the foot of the shock has not been explored yet (because a variable mesh that is very fine near the shock would be required). In addition, the method needs yet to be tested with blunt-nosed airfoils, including flows with a decelerating supersonic region such as occurs over a shock-free airfoil. Efforts in the near future will concentrate on extending the method, first to a variable grid for better accuracy with fewer mesh points and for applicability to more general airfoils, and then to lifting airfoils and to three dimensions.

References

[1] Murman, E.M., Bailey, F.R., and Johnson, M.H., "TSFOIL – A Computer Code for 2-D Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation," *Aerodynamic Analyses Requiring Advanced Computers*, Pt. II, SP-347, 1975, NASA, pp. 769-788.

[2] Murman, E.M. and Cole, J.D., "Calculation of Plane Transonic Flows," *AIAA Journal*, Vol. 9, Jan. 1971, pp. 114-121.

[3] Murman, E.M. and Krupp, J.A., "Solution of the Transonic Potential Equation Using a Mixed Finite Difference System," *Lecture Notes in Physics*, Vol. 8, *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics*, Sept. 15-19, 1970, edited by M. Holt, Springer-Verlag, Berlin, 1971, pp. 199-206.

[4] Murman, E.M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," presented at AIAA Computational Fluid Dynamics Conference (in bound book of papers; no paper number), July 19-20, 1973, Palm Springs, Calif.; also *AIAA Journal*, Vol. 12, May 1974, pp. 626-633.

[5] Bauer, F., Garabedian, P., and Korn, D., *"Supercritical Wing Sections; Lecture Notes in Economics and Mathematics*, Vol. 66, Springer-Verlag, Berlin, 1972.

[6] Garabedian, P.R. and Korn, D.G., "Analysis of Transonic Airfoils," *Communications in Pure and Applied Mathematics*, Vol. XXIV, 1971, pp. 841-851.

[7] Jameson, A., "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1," *Communications in Pure and Applied Mathematics*, Vol. XXVII, May 1974, pp. 283-309.

[8] Bailey, F.R., "On the Computation of Two- and Three-Dimensional Steady Transonic Flows by Relaxation Methods," *Lecture Notes in Physics*, Vol. 41, *Progress in Numerical Fluid Dynamics*, edited by H.J. Wirz, Springer-Verlag, Berlin, 1975, pp. 1-77.

[9] Widlund, O.B., "On the Use of Fast Mehods for Separable Finite Difference Equations for the Solution of General Elliptic Problems," *Sparse Matrices and Their Applications* edited by D.J. Rose and R.A. Willoughby, Plenum Press, New York, 1972. pp. 121-134.

[10] Concus, P. and Golub, G.H., "Use of Fast Direct Methods for the Efficient Numerical Solution of Nonseparable Elliptic Equations," *SIAM Journal on Numerical Analysis*, Vol. 10, Dec. 1973, pp. 1103-1120.

[11] Roache, P.J., "Finite-Difference Methods for the Steady-State Navier-Stokes Equations," Sandia Labs., Albuquerque, N. Mex., Rept. SC-RR-72-0419, 1972.

[12] Buzbee, B.L., Golub, G.H., and Nielson, C.W., "On Direct Methods for Solving Poisson's Equations," *SIAM Journal on Numerical Analysis*, Vol. 7, Dec. 1970, pp. 627-656.

[13] Martin, E.D. and Lomax, H., "Rapid Finite-Difference Computation of Subsonic and Slightly Supercritical Aerodynamic Flows," AIAA Paper 74-11 (portion of paper) Jan. 1974; also *AIAA Journal*, Vol. 13, May 1975, pp. 579-586.

[14] Lomax, H., and Martin, E.D., "Fast Direct Numerical Solution of the Nonhomogeneous Cauchy-Riemann Equations," *Journal of Computational Physics*, Vol. 15, May 1974, pp. 55-80.

[15] Martin, E.D., "Progress in Application of Direct Elliptic Solvers to Transonic Flow Computations," *Aerodynamic Analyses Requiring Advanced Computers*, Pt. II, SP-347, 1975, NASA, pp. 839-870.

[16] Lomax, H. and Martin, E.D., "Variants and Extensions of a Fast Direct Numerical Cauchy-Riemann Solver, With Illustrative Applications," TN D-7934, 1976, NASA.

[17] Spreiter, J.R., "On the Application of Transonic Similarity Rules to Wings of Finite Span," NACA Rept. 1153, 1953.

[18] Lax, P.D., *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1973.

[19] Lax, P.D. and Wendroff, B., "Systems of Conservation Laws," *Communications in Pure and Applied Mathematics*, Vol. XIII, May 1960, pp. 217-237.

[20] Jameson, A., "Numerical Solution of Nonlinear Partial Differential Equations of Mixed Type," *SYNSPADE 1975*, May 1975, Univ. of Maryland.

[21] Bailey, F.R. and Ballhaus, W.F., "Comparisons of Computed and Experimental Pressures for Transonic Flows About Wings and Wing-Fuselage Configurations," *Aerodynamic Analyses Requiring Advanced Computers*, Pt. II, SP-347, 1975, NASA, pp. 1213-1231.

[22] Martin, E.D., "A Fast Semidirect Method for Computing Transonic Aerodynamic Flows," presented at AIAA 2nd Computational Fluid Dynamics Conference (in bound book of papers; no paper number), June 1975, Hartford, Conn., pp. 162-174.

[23] Warming, R.F. and Beam, R.M. "Upwind Second-Order Difference Schemes and Applications in Unsteady Aerodynamic Flows," presented at AIAA 2nd Computational Fluid Dynamics Conference (in bound book of papers; no paper number), June 1975, Hartford, Conn., pp. 17-28.

[24] Jameson, A., "Three-Dimensional Flows Around Airfoils With Shocks," *Proceedings of the IFIP Symposium of Computing Methods in Applied Sciences and Engineering, IRIA*, Paris, France, Dec. 1973, *Lecture Notes in Computer Science*, Vol. 11, edited by R. Glowinski and J.L. Lions, Springer-Verlag, New York, pp. 554-571.

[25] Aitken, A.C., "On Bernoulli's Numerical Solution of Algebraic Equations," *Proceedings of the Royal Society of Edinburgh*, Vol. 46, 1926, pp. 289-305.

[26] Shanks, D., "Nonlinear Transformations of Divergent and Slowly Convergent Sequences," *Journal of Mathematics and Physics*, Vol. 34, April 1955, pp. 1-42.

[27] Klunker, E.B., "Contribution to Methods for Calculating the Flow About Thin Lifting Wings at Transonic Speeds – Analytical Expression for the Far Field," NASA TN D-6530, 1971.