JOURNAL OF COMPUTATIONAL PHYSICS 35, 356–380 (1980)

A Computational Procedure for Supersonic Flows Governed by the Parabolic Navier–Stokes Equations

C. P. Li

Applied Mechanics Department, Lockheed Electronics Company, Inc., Houston, Texas 77058

Received November 22, 1977; revised July 26, 1979

An implicit finite-difference method is developed to integrate the parabolic Navier–Stokes equations along the main direction for flow over a finite-width plate at 0 and 10° angles of attack. This method utilizes the fractional steps (splitting) technique to seek the solution from a sequence of two-dimensional difference equations. A new linearization algorithm is devised to facilitate the calculation of Jacobian matrices in a Taylor series expansion and to perform successive iterations until the conservative-law equation is recovered. Both backward and centered implicit schemes are used in the splitting technique and the results are compared. Available numerical solutions and experimental data obtained at low-Reynolds-number conditions are also used for comparison. The backward implicit method provides a more successful solution, which ranges from the merged-layer to the strong-interaction regimes. The computed flowfield shows a shear layer around the side edge and small region of reversed lateral flow on the lee side of the plate at an angle of attack.

I. Introduction

The problem of analyzing viscous flow over simple configurations is of practical importance in fluid dynamics in connection with such applications as the design of high-speed flying vehicles and the investigation of flow phenomena in test facilities. The viscous and heat conducting effects are known not only to dominate the flow within a narrow layer adjacent to the solid boundary, but also to play an important role in shaping the flow structures near the leading edge, around the side edge, and behind the object. These flowfields, being far more complex and interesting than their two-dimensional counterparts, are not amenable to most of the existing methods of analysis due to some difficulties. In the first place, the governing equations, known as the Navier–Stokes equations, provide an excellent description of the flowfield, but they are of an elliptic type for which the numerical techniques are lengthy and cumbersome. Second, because of the occurrence of shocks and shear layers resulting in steep gradients of flow properties, sufficiently fine resolution is required to maintain acceptable accuracy as well as a stable solution. Thus, even in some situations where a numerical solution is feasible, the computation cost is prohibitively high for frequent applications. Consequently, there have been some atempts to solve the flow problems with slightly different governing equations which possess parabolicity in one of the independent variables, usually along the coordinate defining the main flow velocity.

0021-9991/80/060356–25$02.00/0

This set of equations, termed the parabolic Navier–Stokes equations, is integrated spatially and requires simpler and more efficient numerical techniques than those for the elliptic Navier–Stokes equations. The object of the present work is to develop a new algorithm to solve the parabolic equations and to evaluate its validity by using a problem of the flow passing a finite-width plate.

There exists strong evidence that the parabolic Navier–Stokes equations can be used successfully to describe a large class of flow problems in which the gradients of the physical dissipations and the pressure can be neglected in the direction of the main flow. Successful applications have been made by Rubin, Cresci, and their co-workers for the computation of supersonic leading-edge flowfields over three-dimensional geometries such as a finite-width plate by Nardo and Cresci [1], an axial corner by Rubin and Lin [2], and a sharp cone by Lin and Rubin [3] and by Helliwell and Lubard [4]. The subsonic or incompressible viscous flow within a square duct has been considered by Patankar and Spalding [5] and by Briley [6]. Although the nature of the problems studied is quite different, the governing equations employed have a close resemblance in the underlying assumption that the second-order derivatives with respect to the streamwise coordinate are much smaller than those with respect to the transverse and lateral coordinates. Furthermore, the numerical techniques used are typified by implicit integrations marching from one plane to another, both normal to the streamwise coordinate. The basic schemes may be grouped into three categories: a noniterative alternating-direction implicit (ADI) method [1, 6], an iterative predictor-corrector method [2–4], and an inexact double-sweep method [5]. The purpose of these schemes is to yield an equivalent solution of the sparse-banded system of equations in two or more steps, each concerning the solution of a simpler tridiagonal system of equations. The ADI method has unconditional stability that allows for a large integrative step, regardless of the fine space discretization on the normal plane which resolves steep gradients within a boundary layer or shock. The predictor-corrector method, which can be viewed as the line Jacobi or the line Gauss–Seidel method for solving a linear system of equations, has shown some appealing capabilities when coupled with the Newton technique for solving nonlinear equations. However, the step increment must satisfy convergence criteria, although they are less stringent than that for explicit methods.

Since the ADI method is in fact one version of the factorization of difference operators, the potentials and limitations of other versions should be explored in the continuing development of methods with superior stability, accuracy, and efficiency for practical computations of three-dimensional flows. The essential concept is based on the splitting up of a multidimensional operator into a sequence of two-dimensional operators and the use of one method of solution for each of the simpler operators. If the operator is linear, this idea can be worked out readily by means of factorization similar to algebraic equations [7]. Obviously, various schemes for approximating the time and space derivatives in the differential equations can be incorporated to exploit their maximum capability since the requirement for computing time and storage is more reasonable for two-dimensional operators than for three-dimensional operators.

The other portion of the numerical algorithm is equally important; it involves the

formulation of difference equations and the development of a solution procedure for the resulting nonlinear equations. A mesh of cells is used because of its recognized compatibility with the conservative laws both locally and globally, as discussed by Cheng [8]. This cell formulation is generally recommended for flow-containing shocks and boundary-layer separations, which are frequently seen in supersonic viscous flows. The nonlinear difference equation in two-dimensional form is solved by a rigorous iterative procedure using the primitive variables and their derivatives. The most significant feature is that the convergence leads to the satisfaction of the conservative-law equation. The linearization error is controlled by a prescribed tolerance parameter on the order of the truncation error associated with the difference scheme. These aspects have received considerable attention in this study because they are crucial to the success of any flowfield compution. This algorithm differs from the one developed by Beam and Warming [9] for the unsteady Navier–Stokes equations in the use of the factored conservative-law form, the Jacobian of the Taylor series expansion for linearizing the governing equations, and the iterative procedure for reducing the linearization error. Referrence [9] has taken a noniterative linearization approach similar to the one in Ref. [6], which has been generalized for an unsteady three-dimensional entrance flow in a duct by McDonald and Briley [10]. The present linearization procedure distinguishes itself from various other procedures in the consistency, which is comparable to the differencing scheme, in the generality for treating nonuniform cells in both directions on the normal plane, and in the simplicity of the coefficients for the block tridiagonal matrix.

As a test case for the new algorithm, the supersonic flow past a finite-width plate at both 0 and 10° angles of attack is considered. The results reported in Ref. [1] appear to have some difficulty since the computation is terminated prematurely at a streamwise location very close to the leading edge and the predicted inviscid–viscous interaction at the side edge is not as strong as anticipated. These difficulties may be caused in part by the fact that the ADI scheme is only marginally stable and therefore not suited to solutions containing sharp gradients, and in part by the use of uniform mesh points which would fail to detect any fine-scale phenomena around the side edge. Since experimental data are also available for comparison, this case is selected to validate the present algorithm and to explore its potential capabilities.

The discussion is divided into the following sections: finite-difference methods, solution to the nonlinear difference equations, governing equations and boundary conditions for flow over a finite-width plate, nonuniform mesh system and stability analysis, discussion of the results, and conclusions.

II. Finite-Difference Methods

Consider a model partial differential equation in the following conservative-law form:

$$\mathbf{F}_x + \mathbf{G}_y + \mathbf{H}_z = 0, \tag{1}$$

where the vectors **F**, **G**, and **H** are the functions of a simple vector **V** whose componnts are dependent variables. The x, y, and z are Cartesian coordinates. If the backward implicit scheme is applied directly to (1), the resulting difference equation is

$$\mathbf{F}_{j,k}^{i+1} = \mathbf{F}_{j,k}^{i} - \frac{\Delta x_i}{\Delta y_j}(\mathbf{G}_{j+1/2,k}^{i+1} - \mathbf{G}_{j-1/2,k}^{i+1}) - \frac{\Delta x_i}{\Delta z_k}(\mathbf{H}_{j,k+1/2}^{i+1} - \mathbf{H}_{j,k-1/2}^{i+1}) + O(\Delta x_i^2) \quad (2)$$

The subscripts j and k denote the locations of the cell point; $j \pm \frac{1}{2}$, k and j, $k \pm \frac{1}{2}$ denote the location of the cell line (Fig. 1). The superscripts i and $i + 1$ refer to the previous and present values. The computational domain defined in the y–z plane is discretized to have a mesh of cells, each having variable Δy_j and Δz_k. The integration step increment, Δx_i, is also determined prior to solving (2).

Equation (2) can be solved by the predictor-corrector method, or by the method of fractional steps in two steps:

$$\begin{aligned} \mathbf{F}_{j,k}^{*} &= \mathbf{F}_{j,k}^{i} - \frac{\Delta x_i}{\Delta y_j}(\mathbf{G}_{j+1/2,k}^{*} - \mathbf{G}_{j-1/2,k}^{*}), \\ \mathbf{F}_{j,k}^{i+1} &= \mathbf{F}_{j,k}^{*} - \frac{\Delta x_i}{\Delta z_k}(\mathbf{H}_{j,k+1/2}^{i+1} - \mathbf{H}_{j,k-1/2}^{i+1}). \end{aligned} \quad (3)$$

The two successive steps constitute a cycle of the calculation, which can be efficiently performed slince k and j are fixed in the first and second equations, respectively. A second-order scheme can be constructed by centering the spatial derivatives of **F** and **G**. The corresponding difference equation is

$$\mathbf{F}_{j,k}^{i+1} = \mathbf{F}_{j,k}^{i} - \frac{\Delta x_i}{\Delta y_j}(\mathbf{G}_{j+1/2,k}^{i+1/2} - \mathbf{G}_{j-1/2,k}^{i+1/2}) - \frac{\Delta x_i}{\Delta z_k}(\mathbf{H}_{j,k+1/2}^{i+1/2} - \mathbf{H}_{j,k-1/2}^{i+1/2}) + O(\Delta x_i^3). \quad (4)$$

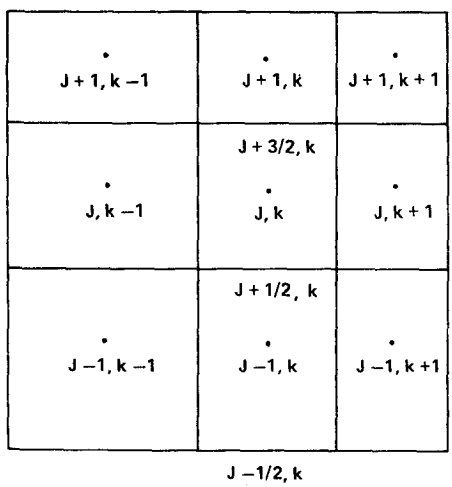


FIG. 1. Notation of the mesh of cells.

The split-up equations may have a form as follows:

$$\begin{aligned}\mathbf{F}^{*}_{j,k} &= \mathbf{F}^{i}_{j,k} - \frac{\Delta x_i}{2\Delta y_j}(\mathbf{G}^{i}_{j+1/2,k} - \mathbf{G}^{i}_{j-1/2,k} + \mathbf{G}^{*}_{j+1/2,k} - \mathbf{G}^{*}_{j-1/2,k}),\\ \mathbf{F}^{i+1}_{j,k} &= \mathbf{F}^{*}_{j,k} - \frac{\Delta x_i}{2\Delta z_k}(\mathbf{H}^{*}_{j,k+1/2} - \mathbf{H}^{*}_{j,k-1/2} + \mathbf{H}^{i+1}_{j,k+1/2} - \mathbf{H}^{i+1}_{j,k-1/2}).\end{aligned} \tag{5}$$

A simple analysis can be carried out to demonstrate the equivalence between (2) and (3) on the basis of linear operator notation. If $d\mathbf{F} = d\mathbf{V}$, $d\mathbf{G} = Ad\mathbf{V}$, and $d\mathbf{H} = Bd\mathbf{V}$, then (1) becomes

$$L\mathbf{V}^{i+1}_{j,k} = \mathbf{V}^{i}_{j,k} \tag{6}$$

whereas (2) becomes

$$L_y\mathbf{V}^{*}_{j,k} = \mathbf{V}^{i}_{j,k} \qquad \text{and} \qquad L_z\mathbf{V}^{i+1}_{j,k} = \mathbf{V}^{*}_{j,k}\,. \tag{7}$$

The operators are defined by

$$\begin{aligned}L\mathbf{V}_{j,k} &= -\alpha\mathbf{V}_{j-1/2,k} + \alpha\mathbf{V}_{j+1/2,k} + \mathbf{V}_{j,k} - \beta\mathbf{V}_{j,k-1/2} + \beta\mathbf{V}_{j,k+1/2}\,,\\ L_y\mathbf{V}_{j,k} &= -\alpha\mathbf{V}_{j-1/2,k} + \mathbf{V}_{j,k} + \alpha\mathbf{V}_{j+1/2,k}\,,\\ L_z\mathbf{V}_{j,k} &= -\beta\mathbf{V}_{j,k-1/2} + \mathbf{V}_{j,k} + \beta\mathbf{V}_{j,k+1/2}\,,\end{aligned}$$

where $\alpha = A_{j,k}\,\Delta x_i/\Delta y_j$ and $\beta = B_{j,k}\,\Delta x_i/\Delta z_k$. Upon substituting the second equation of (7) into the first equation, the following relation results:

$$L_yL_z\mathbf{V}_{j,k} = L_zL_y\mathbf{V}_{j,k} = L\mathbf{V}_{j,k} + O(\Delta x_i{}^2).$$

Likewise, the equivalence between the second-order schemes (4) and (5) can be established by replacing (5) with

$$L_y\mathbf{V}^{*}_{j,k} = \bar{L}_y\mathbf{V}^{i}_{j,k} \qquad \text{and} \qquad L_z\mathbf{V}^{i+1}_{j,k} = \bar{L}_z\mathbf{V}^{*}_{j,k}\,, \tag{8}$$

where

$$\begin{aligned}L_yV_{j,k} &= -\alpha V_{j-1/2,k} + 2V_{j,k} + \alpha V_{j+1/2,k}\,,\\ \bar{L}_yV_{j,k} &= \alpha V_{j-1/2,k} + 2V_{j,k} - \alpha V_{j+1/2,k}\,,\\ L_zV_{j,k} &= -\beta V_{j,k-1/2} + 2V_{j,k} + \beta V_{j,k+1/2}\,,\\ \bar{L}_zV_{j,k} &= \beta V_{j,k-1/2} + 2V_{j,k} - \beta V_{j,k+1/2}\,.\end{aligned}$$

The ADI scheme is also represented in the same notation:

$$L_y\mathbf{V}^{*}_{j,k} = \bar{L}_z\mathbf{V}^{i}_{j,k} \qquad \text{and} \qquad L_z\mathbf{V}^{i+1}_{j,k} = \bar{L}_y\mathbf{V}^{*}_{j,k}\,. \tag{9}$$

The solution of V obtained from (8) and (9) is more accurate than that from (7) for smooth solutions. The first-order scheme of (7), however, is generally preferable over

the second-order schemes when $\mathbf{V}$ has sharp gradients embedded in the flowfield because it provides a lower-order numerical dissipation to damp small-wavelength oscillations due to numerical dispersion (Douglas [11]). Finally, since the operators may not commute with each other in general, the following equations are used instead. Equations (2) and (4) in the actual applications are replaced by

$$L_z(L_y)^2 L_z \mathbf{V}_{j,k}^{i+2} = \mathbf{V}_{j,k}^{i},$$

$$L_z(L_y)^2 L_z \mathbf{V}_{j,k}^{i+2} = \bar{L}_z(\bar{L}_y)^2 \bar{L}_z \mathbf{V}_{j,k}^{i},$$

to maintain the symmetry and the formal order of accuracy.

The validity of approximating a multidimensional difference operator by a sequence of two-dimensional operators has not received full theoretical justification if the operators are nonlinear. Nevertheless, this method has been successfully applied for numerical solutions to various nonlinear flow problems and has recently been generalized by Li [12] to include both explicit and implicit operators for the computation of shock and boundary-layer interaction. Other versions have been developed by Beam and Warming [9] to include three-time-level differencing and a fourth-order quotient for space derivatives.

III. Solution to the Nonlinear Difference Equations

Equations (3) and (5) are strongly nonlinear; hence, a method of linearization is needed to reduce these equations into a solvable form. The following shows an iterative algorithm devised to treat this type of equation such that the conservative-law formulation can be retained throughout the iterations.

If the column vectors $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are not only functions of $\mathbf{V}$, but of $\mathbf{V}_y$ and $\mathbf{V}_z$ as well, they can be related directly by

$$\begin{aligned} d\mathbf{F} &= PA\, d\mathbf{V}, \\ d\mathbf{G} &= PB\, d\mathbf{V} - PC\, d\mathbf{V}_y - PD\, dV_z, \\ d\mathbf{H} &= PE\, d\mathbf{V} - PF\, d\mathbf{V}_z - PG\, d\mathbf{V}_y, \end{aligned}$$

where P, A, B, C, D, E, F, and G are square matrices, PA can be interpreted as the Jacobian matrix of $\mathbf{F}$ with respect to $\mathbf{V}$, and so on. Making use of these relations, iteration formulas are obtained according to the Newton–Raphson technique:

$$\begin{aligned} \mathbf{V}^{l+1} &= \mathbf{V}^{l} + \delta\mathbf{V}, \\ \mathbf{F}^{l+1} &= \mathbf{F}^{l} + (PA)^{l}\, \delta\mathbf{V}, \\ \mathbf{G}^{l+1} &= \mathbf{G}^{l} + (PB)^{l}\, \delta\mathbf{V} - (PC)^{l}\, \delta\mathbf{V}_y - (PD)^{l}\, \delta\mathbf{V}_z, \\ \mathbf{H}^{l+1} &= \mathbf{H}^{l} + (PE)^{l}\, \delta\mathbf{V} - (PF)^{l}\, \delta\mathbf{V}_z - (PG)^{l}\, \delta\mathbf{V}_y. \end{aligned} \tag{10}$$

Here, l indicates the iteration count. Update functions are given on the left side of the equations, while previous functions are on the right side. Since new Jacobian matrices are computed at each iteration, the convergence of (10) is quadratic. For most applications, no more than five iterations are needed to ensure that the $\delta\mathbf{V}$'s are less than a reasonable magnitude of tolerance ϵ. The prescribed value should be greater than the round-off error, but smaller than the truncation error due to the finite-difference approximation of (1). A check is made to see if $|\ \delta\mathbf{V}\ | < \epsilon$ after each iteration. If every point on a fixed jth or kth line satisfies the condition, this line is dropped out of the iterative process. The iterations are terminated when every line has converged solutions. This procedure is also known as the Gauss–Seidel line iteration technique since the updated value is used immediately in the iteration procedure.

The correction vector $\delta\mathbf{V}$ is determined by solving a linear system of algebraic equations, which is obtained after substituting (10) into the first equation in (3) or (5). The system of equations has the form

$$\begin{aligned} b_2\delta\mathbf{V}_2 + c_2\delta\mathbf{V}_3 &= d_2\,, \\ a_j\delta\mathbf{V}_{j-1} + b_j\delta\mathbf{V}_j + c_j\delta\mathbf{V}_{j+1} &= d_j\,, \\ a_{JL}\delta\mathbf{V}_{JL-1} + b_{JL}\delta V_{JL} &= d_{JL}\,, \end{aligned} \tag{11}$$

with $j = 3, 4, \ldots, JL - 1$. The boundary conditions are incorporated into the coefficients in the first and last equations; hence, they are slightly different from the coefficients in the second equation. A general expression of these coefficients can be derived as a function of the Jacobian matrices and the mesh spacings:

$$\begin{aligned} a_j &= \alpha_1(PB)^l_{j-1} + (PC)^l_{j-1/2}/(y_j - y_{j-1}), \\ b_j &= -\frac{\theta\,\Delta y_j}{\Delta x_i}(PA)^l_j + (\alpha_2 - \beta_2)(PB)^l_j - (PC)^l_{j-1/2}/(y_j - y_{j-1}) \\ &\quad - (PC)^l_{j+1/2}/(y_{j+1} - y_j), \\ c_j &= -\beta_1(PB)^l_{j+1} + (PC)^l_{j+1/2}/(y_{j+1} - y_j), \\ d_j &= \frac{\theta\,\Delta y_j}{\Delta x_i}(\mathbf{F}_j^{\,l} - \mathbf{F}_j^{\,i}) + \mathbf{G}^i_{j+1/2} - \mathbf{G}^i_{j-1/2} + (1-\theta)(\mathbf{G}^l_{j+1/2} - \mathbf{G}^l_{j-1/2}). \end{aligned} \tag{12}$$

Note that these groups of coefficients are used to solve for $\delta\mathbf{V}_j$ at a fixed kth line. Hence, the subscript k is not shown for clarity. Equation (11) yields a second-order solution with $\theta = \frac{1}{2}$, and a first-order solution with $\theta = 1$. The weighting parameters are $\alpha_1 = 1/(1 + R)$ and $\alpha_2 = \alpha_1 R$, where $R = \Delta y_{j-1}/\Delta y_j$; likewise, β_1 and β_2 are obtained using $R = \Delta y_{j+1}/\Delta y_j$.

In order to derive (12), the derivatives of $\mathbf{V}$ are approximated by either forward or backward difference quotients in terms of $\mathbf{V}$; viz.,

$$\begin{aligned} (\delta\mathbf{V}_y)_{j\pm1/2,k} &= (\delta\mathbf{V}_{j\pm1,k} - \delta\mathbf{V}_{j,k})/(y_{j\pm1} - y_j), \\ (\delta\mathbf{V}_z)_{j\pm1/2,k\pm1/2} &= (\delta\mathbf{V}_{j\pm1/2,k\pm1} - \delta\mathbf{V}_{j\pm1/2,k\pm1})/(z_{k\pm1} - z_k). \end{aligned}$$

The difference quotient for the cross derivative of $\delta\mathbf{V}$, which requires updated values on the $k \pm 1$th lines, is excluded from the expressions of a_j, b_j, and c_j under the assumption that the magnitude of cross derivatives is much smaller than that of second derivatives. This approximation does not impede the accuracy of $\mathbf{V}$ as long as the residue vector, d_j, is evaluated rigorously. The rate of convergence undoubtedly will improve if $\delta\mathbf{V}_z$'s are included in the evaluation of the coefficients, but this calculation necessitates additional computer core for storing $\delta\mathbf{V}$ in a two-dimensional array and has not been implemented. The boundary conditions are imposed on the 3/2th line by a simple reflection technique to relate $\delta\mathbf{V}_1$ and $\delta\mathbf{V}_2$. It is easy to show that

$$b_2 = -(\Delta y_2/\Delta x_i)(PA)_2 - \beta_2(PD)_2 - 2(PC)_{3/2}/(y_2 - y_1).$$

The coefficient b_{JL} remains unchanged because, on the upper boundary, it is simply $\delta\mathbf{V}_{JL} = 0$. The solution to the second equation in (3) and (5) is obtained by the same algorithm. The resulting coefficients are

$$\begin{aligned}
a_k &= \alpha_1(PE)^l_{k-1} + (PF)^l_{k-1/2}/(z_k - z_{k-1}),\\
b_k &= -(\theta\,\Delta z_k/\Delta x_i)(PA)^l_k + (\alpha_2 - \beta_2)(PE)^l_k - (PF)^l_{k-1/2}/(z_k - z_{k-1})\\
&\quad - (PF)^l_{k+1/2}/(z_{k+1} - z_k) \qquad (13)\\
c_k &= -\beta_1(PE)^l_{k+1} + (PF)^l_{k+1/2}/(z_{k+1} - z_k),\\
d_k &= (\theta\,\Delta z_k/\Delta x_i)(\mathbf{F}_k{}^l - \mathbf{F}^*_k) + \mathbf{H}^*_{k+1/2} - \mathbf{H}^*_{k-1/2} + (1-\theta)(\mathbf{H}^l_{k+1/2} - \mathbf{H}^l_{k-1/2}).
\end{aligned}$$

The weighting parameters are determined from the ratios of $\Delta z_{k-1}/\Delta z_k$ and $\Delta z_{k+1}/\Delta z_k$.

An alternate linearizing procedure is to solve for $\mathbf{V}^{l+1}$ directly from $\mathbf{V}^l$. The Newton–Raphson technique leads to relations such as

$$\mathbf{F}^{l+1} = \mathbf{F}^l + (PA)^l(\mathbf{V}^{l+1} - \mathbf{V}^l)$$

The resultant coefficients closely resemble those in (12) and (13), but the accuracy may not be as good because the round-off error is larger due to $|\mathbf{V}| \gg |\delta\mathbf{V}|$. Furthermore, the derivation of coefficients at the boundaries becomes more complex [7]. ($\delta\mathbf{V}_1 = \delta\mathbf{V}_{JL} = 0$ is replaced by $\mathbf{V}_1 = \mathbf{V}_{JL} = \mathbf{V}_\infty$.)

Equations (12) and (13) are block tridiagonal systems of equations for which an efficient algorithm in Ref. [13] is readily adopted for solution. Note that as $|\delta\mathbf{V}_j|$ approaches ϵ, $|d_j|$ approaches ϵ, which brings about the recovery of the conservative-law equations in (3) and (5) at the end of the iterations. This linearization algorithm and a similar procedure in Ref. [4] advocate successive iterations until the linearization error in the form of $|\delta\mathbf{V}|$ is equal to or less than the truncation error caused by differencing. The overall accuracy of the solution has been found to rely on the satisfaction of the convergence tolerance parameter and could become unacceptable when a noniterative algorithm is used. Other than this, the present one differs from

those presented in Refs. [9, 10] in the form of Jacobian matrices and in the implicit differencing of the flux, stress, and heat conduction terms in **G** and **H**. Although this conservative algorithm is much less prone to nonlinear instability, it is not entirely obvious which will give more accurate results for the same computing time and storage.

IV. Governing Equations and Boundary Conditions for Flow over a Finite-Width Plate

The functions in (1),

$$\mathbf{F}_x + \mathbf{G}_y + \mathbf{H}_z = 0,$$

are defined by

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + \bar{p} \\ \rho uv \\ \rho uw \\ (\rho\epsilon + \bar{p})\, u \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho vu + \tau_{yx} \\ \rho v^2 + \sigma_y \\ \rho vw + \tau_{yz} \\ (\rho\epsilon + \sigma_y)\, v + \tau_{yx}u + \tau_{yz}w + q_y \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} \rho w \\ \rho wu + \tau_{zx} \\ \rho wv + \tau_{zy} \\ \rho w^2 + \sigma_z \\ (\rho\epsilon + \sigma_z)\, w + \tau_{zx}u + \tau_{zy}v + q_z \end{bmatrix}. \tag{14}$$

The computational space is defined by $0 \leqslant y \leqslant H$ and $0 \leqslant z \leqslant B$, in which the plate lies on the z-axis and $0 \leqslant z \leqslant W$ (Fig. 2a). The governing equations are integrated along the x-axis in the y–z plane, subject to the following conditions:

$0 \leqslant z \leqslant W$:

$$u(0, z) = \Lambda u_y(0, z), \qquad v(0, z) = w(0, z) = 0,$$

$$e(0, z) = e_w - \frac{2\gamma}{\gamma + 1}\frac{\Lambda}{Pr}\, e_y(0, z), \qquad p_y(0, z) = 0;$$

$W < z \leqslant B$:

$$u_y(0, z) = w_y(0, z) = e_y(0, z) = p_y(0, z) = \rho_y(0, z) = 0,$$
$$v(0, z) = 0 \text{ if } \alpha = 0 \quad \text{and} \quad v_y(0, z) = 0 \text{ if } \alpha \neq 0;$$

$z > 0$:

$$u(H, z) = U_\infty \cos\alpha, \qquad v(H, z) = -U_\infty \sin\alpha, \qquad w(H, z) = 0,$$
$$e(H, z) = e_\infty, \qquad p(H, z) = p_\infty, \qquad \rho(H, z) = \rho_\infty; \tag{15}$$

$y > 0$:

$$u(y, B) = U_\infty \cos \alpha, \qquad v(y, B) = -U_\infty \sin \alpha, \qquad w(y, B) = 0,$$
$$e(y, B) = e_\infty, \qquad p(y, B) = p_\infty, \qquad \rho(y, B) = \rho_\infty,$$
$$u_z(y, 0) = v_z(y, 0) = e_z(y, 0) = p_z(y, 0) = \rho_z(y, 0) = 0,$$
$$w(y, 0) = 0;$$

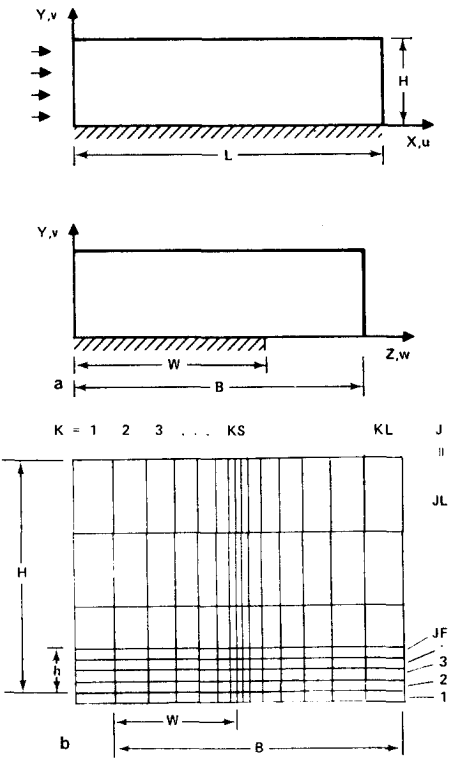


FIG. 2. Schematic of the computational domain. (a) The coordinate system; (b) the cell system.

where ρ, u, v, w, p, and e are, respectively, the density, velocity components in (x, y, z), pressure, and internal energy. The total internal energy is $\epsilon = e + 0.5\,(u^2 + v^2 + w^2)$. The stress components are defined as follows:

$$\tau_{yx} = -\mu u_y, \qquad \tau_{yz} = \tau_{zy} = -\mu(w_y + v_z), \qquad \tau_{zx} = -\mu u_z,$$
$$\sigma_y = p - (\lambda + 2\mu)\, v_y - \lambda w_z, \tag{16}$$
$$\sigma_z = p - (\lambda + 2\mu)\, w_z - \lambda v_y.$$

The heat fluxes are obtained from the expressions

$$q_y = -\frac{\kappa}{C_v} e_y \qquad \text{and} \qquad q_z = -\frac{\kappa}{C_v} e_z, \tag{17}$$

where $2\lambda + 3\mu = 0$, $\kappa = C_p\mu/Pr$, $e = C_vT$, and $\gamma = C_p/C_v$. The μ is the viscosity coefficient given for air by

$$\mu = 2.27 \times 10^{-8} \frac{T^{3/2}}{T + 198.6} \left(\frac{\text{lb/sec}}{\text{ft}^2}\right). \tag{18}$$

The coefficient of thermal conductivity, κ, is related to the Prandtl number, $Pr = 0.71$. The specific heats at constant pressure and volume are C_p and C_v; their ratio, γ, is equal to 1.4 for perfect air. The temperature is T; the angle of attack is α.

Equation (16) is supplemented by the equation of state

$$p = (\gamma - 1)\rho e. \tag{19}$$

Note p is different from $\bar{p}$ used in (14), which represents a known streamwise pressure gradient because of the stability consideration in the numerical integration to be discussed in Section V.

The boundary conditions have taken into account the slip in velocity parallel to the wall and the jump in temperature. The mean free path of air, Λ, can be estimated by $\Lambda = (\pi\gamma/2)^{1/2}(M/Re) = (\pi/2p\rho)^{1/2}\mu$. The Reynolds number, Re, and the viscous–inviscid interaction parameter, χ, are defined as $Re_x = \rho_\infty U_\infty x/\mu_\infty$ and $\chi_\infty = M_\infty{}^3(c/Re_x)^{1/2}$, where M_∞ is the free-stream Mach number, and c is the Chapman–Rubesin constant ($c = \mu T_\infty/\mu_\infty T$). Equation (15) also includes extraneous boundary conditions off the plate on $y = 0$ and at the symmetry plane $z = 0$, which are needed to simplify the computation.

The Jacobian matrices used in (10) and (13) are given below:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ \frac{1}{2}q^2 & \rho u & \rho v & \rho w & 1/g \end{bmatrix}, \quad A = \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ gue & p + g\bar{p} & 0 & 0 & g\rho u \end{bmatrix}, \tag{20}$$

$$B = \begin{bmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ ge/\rho & 0 & v & 0 & g \\ 0 & 0 & 0 & v & 0 \\ gve & 0 & \gamma g\rho e & 0 & g\rho v \end{bmatrix}, \quad C = \frac{1}{\rho}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \lambda + 2\mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \gamma g u/Pr \end{bmatrix},$$

$$E = \begin{bmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ ge/\rho & 0 & 0 & w & g \\ gwe & 0 & 0 & \gamma g\rho e & g\rho w \end{bmatrix}, \quad F = \frac{1}{\rho}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \lambda + 2\mu & 0 \\ 0 & 0 & 0 & 0 & \gamma g\mu/Pr \end{bmatrix},$$

where $q^2 = u^2 + v^2 + w^2$ and $g = \gamma - 1$.

V. Nonuniform Mesh System and Stability Analysis

The computational domain outlined in Section IV is divided into a mesh of rectangular cells, each of which has variable length and height. The smallest cell is located at the leading edge ($x = 0$), whereas the largest cell is at the opposite corner of the region ($x = L$ and $y = H$). Integers are used to designate the cell number; half-integers designate the line between two cells. The system of cells is determined prior to the computation by means of the following procedure. First an estimate is made of Δy_1 or Δy_2, using the smallest value of the boundary-layer thickness at $x = L$ and the mean free path of the free stream; i.e., $\Delta y_1 = \min(\delta, \Lambda)$, where $\delta = 0.3L/(Re_L)^{1/2}$ and Λ is given in Section IV. Then an exponential function is employed to determine Δy_j, where $j = 3, 4, \ldots, JF$. This function relates the transformed coordinate $\bar{y}$ to the physical coordinate by $y = (e^{c\bar{y}} - 1)\, h/(e^c - 1)$, $0 \leqslant \bar{y} \leqslant 1$ and $0 \leqslant y \leqslant h$. The parameter c is determined iteratively in order to satisfy the requirements that $y_1 = 0.5\Delta y_1$, $y_{j+1} = y_j + 0.5\Delta y_j$, and $y_{JF} = h$, where JF is the number of fine cells in the y direction. On top of this system of fine cells, there is a system of coarse cells. The height of these cells is uniform and is obtained from $\Delta y = (H - h)/(JL - JF)$ with $y_{j+1} = y_j + 0.5\Delta y$ and $y_{JL} = H$, where JL is the total number of cells along the j-axis. The width of these cells is nonuniform in the region near the side edge and uniform elsewhere. The procedure used to determine Δz_k and z_k is similar to that for Δy_j and y_j except that for this case $\Delta z_{KS} = \Delta z_{KS-1}$ is specified to ensure that fine resolution is provided at the side edge; KS designates the selected location of the side edge. Likewise, Δx_i and x_i are estimated using the requirement

[redacted]

The mesh of cells is constructed to properly resolve the gradients of flow properties that may occur near the leading edge and around the side edge. For a finite plate at an angle of attack, it should be complemented with formal second-order accuracy in Δy_j or Δz_k in the approximations of the flux, stress, and heat-conducting terms in **F**, **G**, and **H**. For simplicity, those terms appearing in **G** are considered. The conventional two-point formula is used to represent y-derivatives, while a weighted interpolation formula is used for fluxes; viz.,

$$(u_y)_{j\pm1/2} = (u_{j\pm1} - u_j)/(y_{j\pm1} - y_j),$$
$$(\rho u^2)_{j+1/2} = \alpha_1(\rho u^2)_{j+1} + \alpha_2(\rho u^2)_j .$$

Therefore, the accuracy of representation of flux terms is equal to the three-point formula applied to the nonuniform mesh of points. The mixed derivatives are evaluated as follows, using both the old and updated iteration values because the iteration sweeps the columns from $k = 2$ to $k = KL - 1$:

$$(u_{xy})_{j+1/2,k} = (u^l_{j+1,k+1} - u^l_{j+1,k-1} - u^{l+1}_{j-1,k+1} + u^{l+1}_{j-1,k-1})/(y_{j+1} - y_{j-1})/(z_{k+1} - z_{k-1}).$$

The boundary conditions in (15), when imposed on the cell lines, require that the dependent variables and their normal gradients be zero. A finite-difference representation of the derivatives is made consistent with those inside the computational domain. For example, $\mathbf{V}_{1,k} = -\mathbf{V}_{2,k}$ and $\mathbf{V}_{j,1} = -\mathbf{V}_{j,2}$ approximate, respectively, $(\mathbf{V})_{y=0} = 0$ and $(\mathbf{V})_{z=0} = 0$. In a similar manner, $\mathbf{V}_{1,k} = \mathbf{V}_{2,k}$ and $\mathbf{V}_{j,1} = \mathbf{V}_{j,2}$ approximate $(V_y)_{y=0} = 0$ and $(\mathbf{V}_z)_{z=0} = 0$. Additional conditions are required to ensure there are no fluxes across the wall. For example, this is accomplished in $\mathbf{G}$ as follows:

$$G_{3/2,k} = (0, \tau_{yx}, \sigma_y, \tau_{yz}, q_y)^T.$$

The computational efficiency and stability of an implicit technique are not dictated by the fine cells on the y–z plane. A simple analysis shows the present algorithm is unconditionally stable. A system of quasilinear equations corresponding to (1) is

$$A\mathbf{V}_x + B\mathbf{V}_y - C\mathbf{V}_{yy} = 0, \tag{21}$$

where A, B, and C are given in Section IV as functions of $\mathbf{V}$. Equation (21) is composed of the hyperbolic and the parabolic equations whose difference forms, obtained from the centered scheme for a mesh of uniform cells, are

$$A_j^i(\mathbf{V}_j^{i+1} - \mathbf{V}_j^i) = -\frac{\Delta x_i}{2\Delta y_j} B_j^i(\mathbf{V}_{j+1}^{i+1/2} - \mathbf{V}_{j-1}^{i+1/2}), \tag{22}$$

$$A_j^i(\mathbf{V}_j^{i+1} - \mathbf{V}_j^i) = \frac{\Delta x_i}{(\Delta y_j)^2} C_j^i(\mathbf{V}_{j+1}^{i+1/2} - \mathbf{V}_j^{i+1/2} + \mathbf{V}_{j-1}^{i+1/2}). \tag{23}$$

The amplification matrices for these equations are obtained using the Von Neumann method [14]:

$$G_{\mathrm{h}} = \frac{I - (-1)^{1/2}\,\alpha A^{-1}B \sin^2\phi}{I + (-1)^{1/2}\,\alpha A^{-1}B \sin^2\phi}, \tag{24}$$

$$G_{\mathrm{p}} = \frac{I - (-1)^{1/2}\,\beta A^{-1}C \sin^2\phi/2}{I + (-1)^{1/2}\,\beta A^{-1}C \sin^2\phi/2}, \tag{25}$$

where $\alpha = (\Delta x_i/2\Delta y_j)^2$, $\beta = \Delta x_i/(\Delta y_j)^2$, $\phi = j\Delta y_j$, and I is the unit matrix. The eigenvectors are obtained by solving polynomials of the equations

$$|\, I\lambda_{\mathrm{h}} - A^{-1}B \,| = 0 \qquad \text{and} \qquad |\, I\lambda_{\mathrm{p}} - A^{-1}C \,| = 0.$$

Thus,

$$\lambda_{\mathrm{h}} = \left(\frac{v}{u}, \frac{v}{u}, \frac{v-a}{u}, \frac{v+a}{u}\right), \tag{26}$$

$$\lambda_{\mathrm{p}} = \left(0, \frac{\mu}{\rho}, \frac{\lambda + 2\mu}{\rho u}, \frac{\gamma\mu}{\kappa\rho u}\right). \tag{27}$$

Since they are real variables, $| G_h | \leqslant 1$ and $| G_p | \leqslant 1$ are implied. Thus (22) and (23) are always stable. If the backward scheme is used, the numerators in (24) and (25) are replaced by an identity matrix. Following the same procedure, the same conclusion is reached.

Numerical instability could arise if the streamwise pressure is treated on the same basis as other variables in (14). For this case, F and A become

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho\epsilon + p)\, u \end{bmatrix}, \qquad A = \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ ge/\rho & u & 0 & 0 & g \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ gue & \gamma g\rho e & 0 & 0 & g\rho u \end{bmatrix}.$$

The corresponding eigenvalues are

$$\lambda_h = \left(\frac{v}{u}, \frac{v}{u}, \frac{uv \pm a(u^2 + v^2 - a^2)^{1/2}}{u^2 - a^2}\right), \tag{28}$$

$$\lambda_p = \left(0, \frac{\lambda + \mu}{\rho u}, \lambda_3, \lambda_4\right). \tag{29}$$

They will have complex components if $u^2 + v^2 < a^2$, where a is the local sonic speed. Furthermore, λ_3 and λ_4 in (29) are complex; therefore, (22) and (23) become unstable. This stability problem has been recognized in earlier works solving the parabolic Navier–Stokes equations and p_x has been neglected or approximated by $\bar{p}_x$, the pressure gradient obtained at the immediate upstream station [1–4]. The second approach is adopted, because it is found in [15] that this approximation gives satisfactory results in comparison with an exact solution near the leading edge. For incompressible flows the pressure must be determined from Poisson's equation. Meaningful solutions can be obtained only after the ellipticity of the flow problem is correctly handled.

The computational efficiency of this algorithm is relatively difficult to assess. The advantage of selecting the larger step increments and extremely fine cell sizes is often penalized by a relatively large amount of computation time required for solving the nonlinear equations. In fact, more iterations are required in the procedure described in Section III for regions where properties change rapidly than for regions near the undisturbed free stream.

VI. Discussion of the Results

The numerical computations have focused on a case that has been studied and reported in Ref. [1]. A limited amount of data has also been available from a test facility consisting of a nozzle placed inside a Mach 12 blowdown tunnel. Due to the large drop in stagnation pressure, extremely low densities are achieved in the test section. The free stream conditions upstream of the plate have a Reynolds number

($Re_\infty = 300$ per inch) and a Mach number ($M_\infty = 5.15$). The free stream temperature is $T_\infty = 230°\ R$, while the wall temperature is $T_w = 460°\ R$. The numerical results have been obtained from an ADI scheme using a mesh of points with constant spacings in the transverse, lateral, and streamwise directions. Comparisons with the experimental data have been made at $\chi \cong 10$; this station displays strong merged-layer characteristics.

The computational domain and the mesh of cells have been selected in a different manner than those used previously in an attempt to predict the flowfield structures around the side edge. The size of the domain is given as follows: $L = 2$ in., $H = 2.4$ in., $h = 0.48$ in., $W = 2.16$ in., and $B = 3.03$ in. The cell system is designated by the notation $(JF)\ JL \times KL(KS)$ as (12) $32 \times 32(20)$; the nomenclature has been defined in Section IV. The resulting cell dimensions are $\Delta y_1 = \Delta y_2 = 0.024$ in. and $\Delta z_{20} = \Delta z_{19} = 0.0123$ in. A schematic of the cell system is shown in Fig. 2b. The streamwise increments are also nonuniform and are determined after assigning $\Delta x_1 = 0.024$ in. and $IL = 30$. Note that the smallest cell used is less than the mean free path, which is $\Lambda_\infty = 0.03$ in. for this case.

Comparisons are first made among the theoretical predictions of the pressure distributions on the plate in the lateral direction. Figure 3a shows that there are significant differences in pressure values near the side edge, but a good agreement toward the plane of symmetry and the opposite side in the free stream. The centered solution of (4) indicates a very sharp drop of pressure right at the side edge, the backward solution of (2) predicts a moderate variation, whereas the previous ADI results display very little change in pressure across the side edge. Assessing the accuracy yielded by these computations is very difficult without data from other sources. Observations of earlier published results indicate that the shear layer characteristics might have been partially ignored because the mesh system is relatively coarse, even though the computational domain is less than half of that used in the present computations. On the other hand, the appreciable differences between the centered and the backward solutions are affected to some extent by the numerical dissipations of the schemes. The centered scheme is known to provide more accurate results if the solution is smooth, but may exaggerate the results for a rapidly varying solution. Further, due to its lack of damping, small-wavelength disturbances tend to grow and eventually become outbound. Indeed, in the present calculation the centered solution has encountered some difficulties with the iterative procedure after reaching $\chi = 6.6$ at $I = 25$ when negative densities appear in some cells neighboring the side edge. In constrast, the results obtained from the dissipative backward scheme are stable throughout the integration and appear to yield reasonable accuracy. For this reason, the backward scheme has been employed in the entire study.

In addition to the differences existing between theoretical pressure results, there is also slight disagreement with the available experimental data in the flow flux at the midspan of the plate, as shown in Fig. 3b. The reason for the deviation within the boundary layer has not yet been determined. The theoretical predictions of flux agree with each other very well except near the region where the maximum value occurs. These curves are not asymptotic to zero at $y = 0$ since the slip velocity is used there.

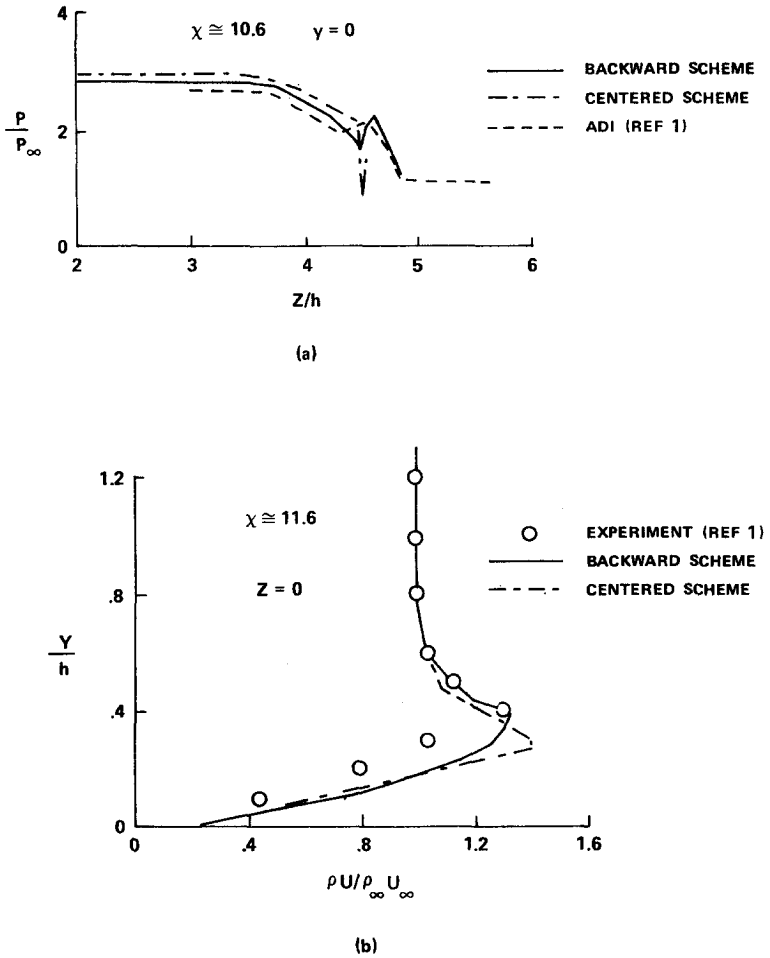


FIG. 3. Comparison of present results with available numerical and experimental data. (a) Lateral pressure distributions on $y = 0$. (b) Transverse mass fluxes at $z = 0$.

Figure 4 presents a series of pressure distributions versus the z-coordinate. The pressure starts out with the free stream value, slowly builds up its magnitude as the slip velocity reduces, and finally reaches the peak value, $p = 5.25p_\infty$ at $\chi = 19.5$. At the beginning there is a smooth transition from the surface pressure to the free stream pressure outside of the plate. In Fig. 4a the lateral pressure distribution resembles closely that within a normal shock. Figures 4b–d show the detailed variations of pressure as the flow moves downstream. A discontinuity develops within the shear layer surrounding the edge, then dissipates and diffuses sideways as the shear layer grows thicker and the magnitude of gradients reduces. At the end of the computation, $\chi = 5.35$ or $x = 2$ in., the side edge starts to lose its influence on the local flowfield. The mechanism producing the pressure discontinuities must originate from the strong

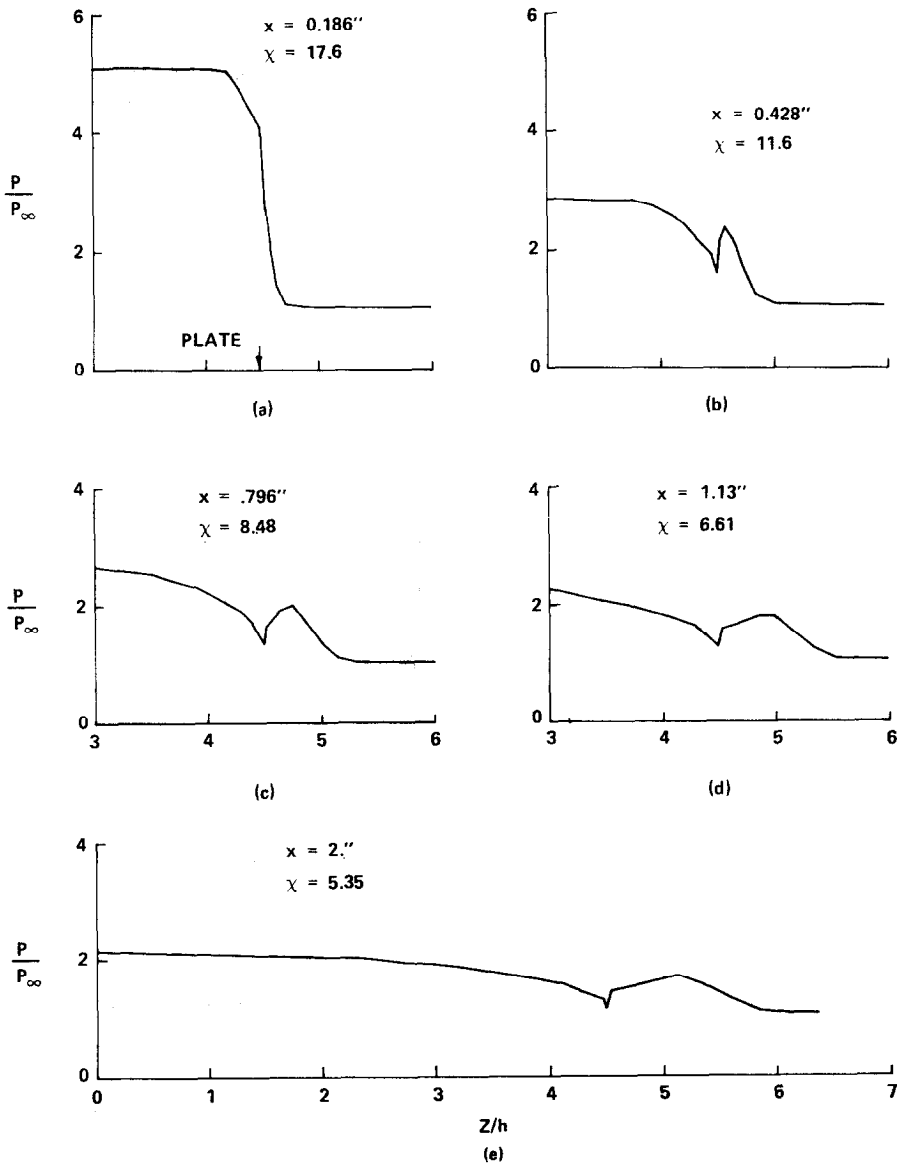


FIG. 4. Pressure distributions on $y = 0$ for zero degree of flow incidence at (a) $x = 0.186$ in., (b) $x = 0.428$ in., (c) $x = 0.796$ in., (d) $x = 1.13$ in., (e) $x = 2$ in.

interactions between the flow and the plate. Due to the presence of a plate in the flow, the kinetic energy transforms directly into the internal energy on top of the plate. This process does not take place immediately because the rarefaction effect predominates at the leading edge. The flow over the plate adjusts itself rather quickly, but slowly in the shear layer around the edge. Therefore, the pressure retains a higher value for a longer distance outside the plate until the physical dissipation finally smooths out the

pressure distribution. Unfortunately, no experimental data are available in this region, so confirmation of the prediction cannot be made at this time. Note that the side edge effect has almost reached the centerline, as indicated in Fig. 4e. This points out the possibility that the flow cannot be considered two-dimensional at $x = L$ when the aspect ratio, W/L, is less than unity. This observation agrees with the findings reported in Ref. [1].

The distributions of the coefficients of skin friction, $\tau_{xy}/\frac{1}{2}\rho_\infty U_\infty{}^2$, and of heat transfer, $(q + u\tau_{xy})/\rho_\infty U_\infty(H_{t_\infty} - H_{t_w})$, are given in Figs. 5a and b, respectively. As explained earlier, the shear layer increases its extent with decreasing values of χ, and moreover, the shear layer creates relatively higher heating at the edge in comparison to that at the center. Judging from these figures, however, the flowfield remains two-dimensional at the centerline; its characteristics have been discussed in full detail in Ref. [15] and the references cited therein.

Figure 6 shows the velocity vector, constant contours of w, pressure, and temperature plotted on the plane normal to the x-coordinate at $\chi = 5.35$. It indicates an upward motion of flow above the plate. Near the edge, however, a sideways motion predominates. Away from the side region, there are no transverse and lateral velocity components. The lateral velocity is plotted in constant contours to show that the highest value occurs somewhere out in the free stream and that it changes rapidly at the side edge. The pressure contours exhibit a leading-edge oblique shock and a side-edge shock next to the side edge. The temperature contours disclose the location of a discontinuity near the edge, which has a stronger intensity than the one on top of the plate.

More computations have been carried out at angles of attack $\alpha = 10°$ and $-10°$, which represent the wind and the lee sides of the plate, respectively. The primary purpose is to check the capability of the numerical algorithms under more severe conditions. From the practical point of view, a computation of the entire flowfield on both sides of the plate would be of interest, but this would necessitate extensive modification of the program and would add little value to the aforementioned objective. In the present approach, these two cases are treated independently, with boundary conditions imposed at $y = 0$ and $B \leqslant z < W$ to allow for the continuation of the flow vector, and at $y = H$ and $z = B$ to make the free stream velocity equal to $u_\infty = U_\infty \cos\alpha$ and $v_\infty = -U_\infty \sin\alpha$ [see (15)]. The computational domain and the cell system are the same as those for $\alpha = 0°$. Despite the fact that this approach sacrifices physical reality, the predictive potential or limitation of the algorithm can still be evaluated and assessed.

Figure 7 shows the pressure distributions versus the z-coordinate at selected x-stations. The wiggles in the pressure distributions are believed to be caused by insufficient space resolution since flow properties change more drastically around the side edge. Regardless of the local numerical problems, the computation has been completed at $x = 2$ inches for both cases. The x-component of shear stress and the heat fluxes behave in the same manner as those predicted for $\alpha = 0°$; hence, they are not repeated. A new and interesting phenomenon related to $\alpha \neq 0°$ is that the lateral flow on the lee side has reversed its direction from that on the wind side in a limited

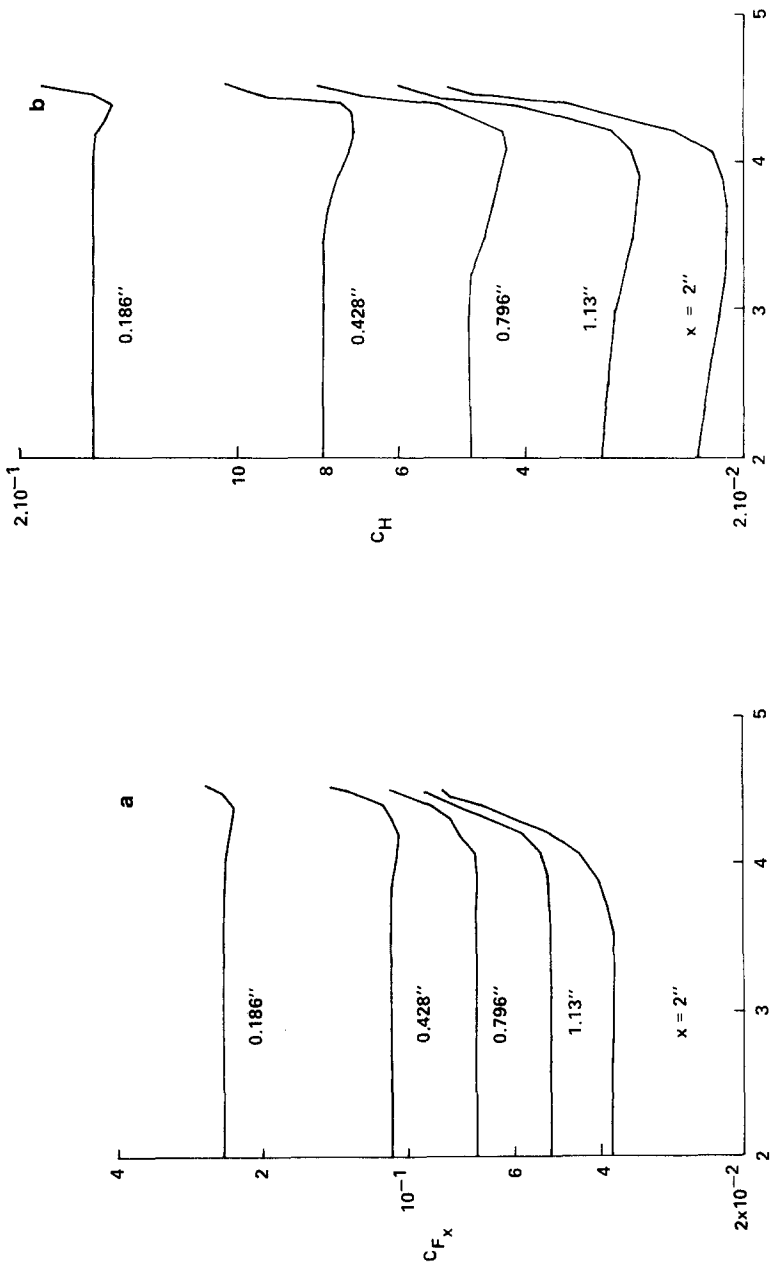


FIG. 5. Variations of wall friction and heat-transfer coefficients at several x-stations. (a) Streamwise friction coefficient. (b) Heat-transfer coefficient.

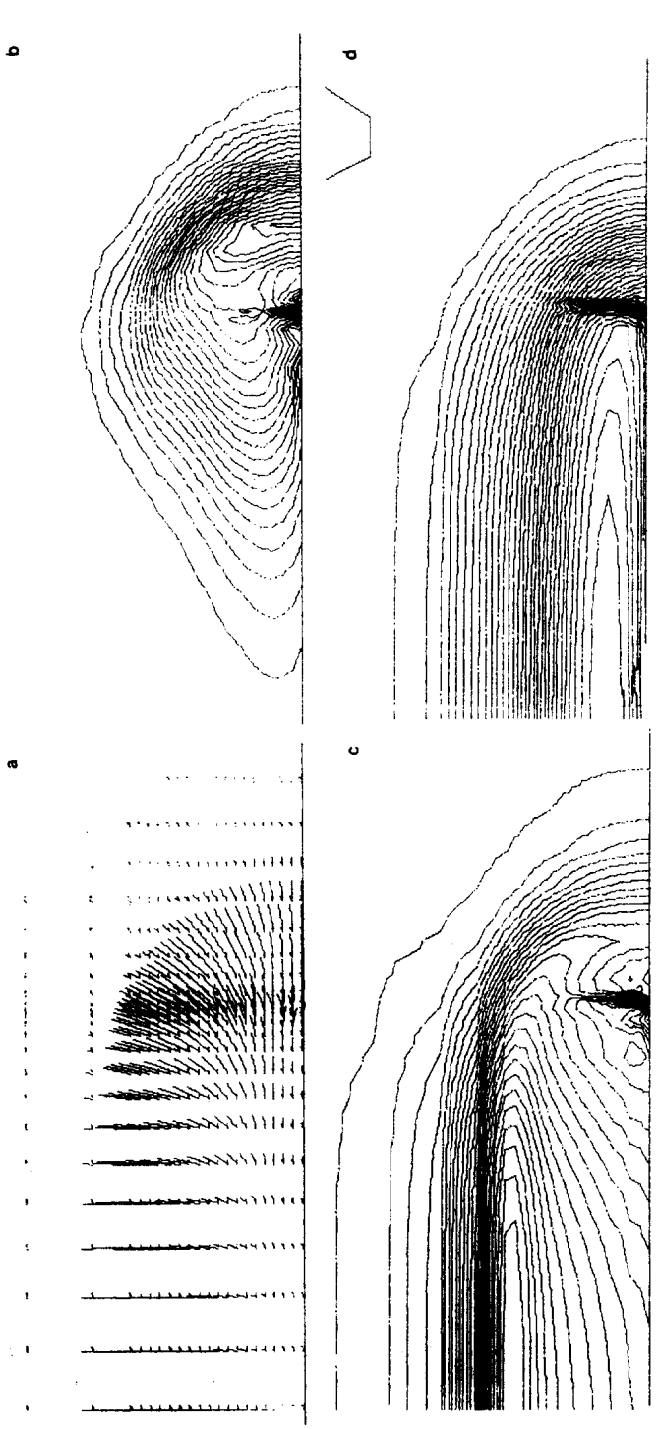


FIG. 6. Constant contours of flow variables on the crossflow plane at $x = 2$ in. (a) Velocity vector. (b) Lateral velocity component. (c) Pressure. (d) Temperature.

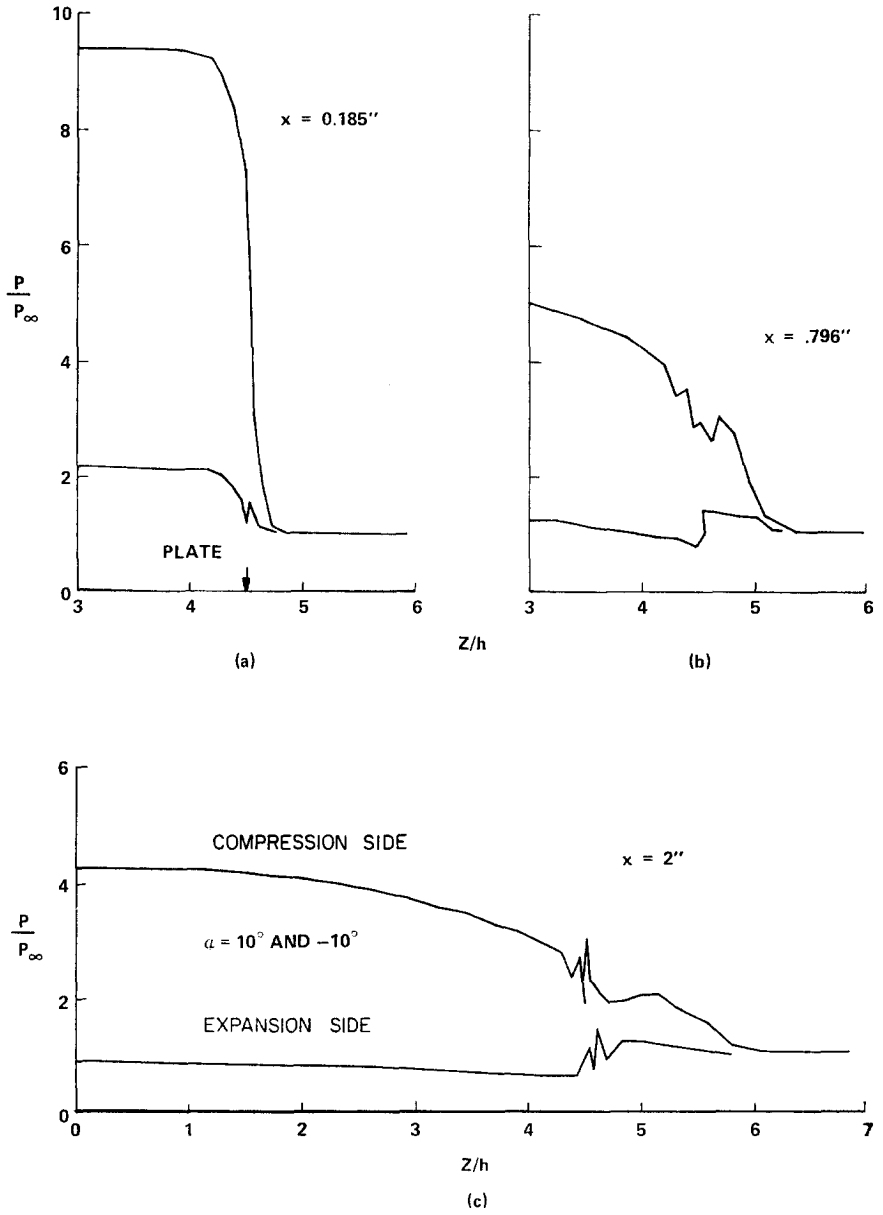
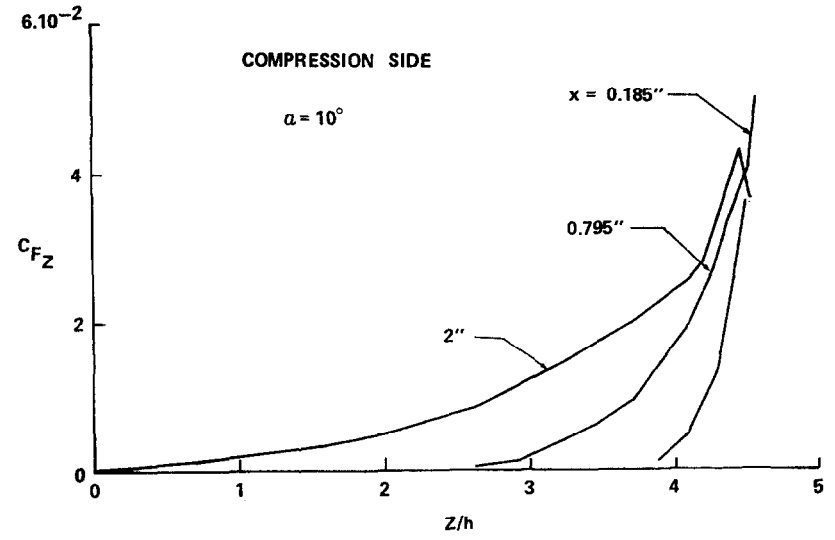
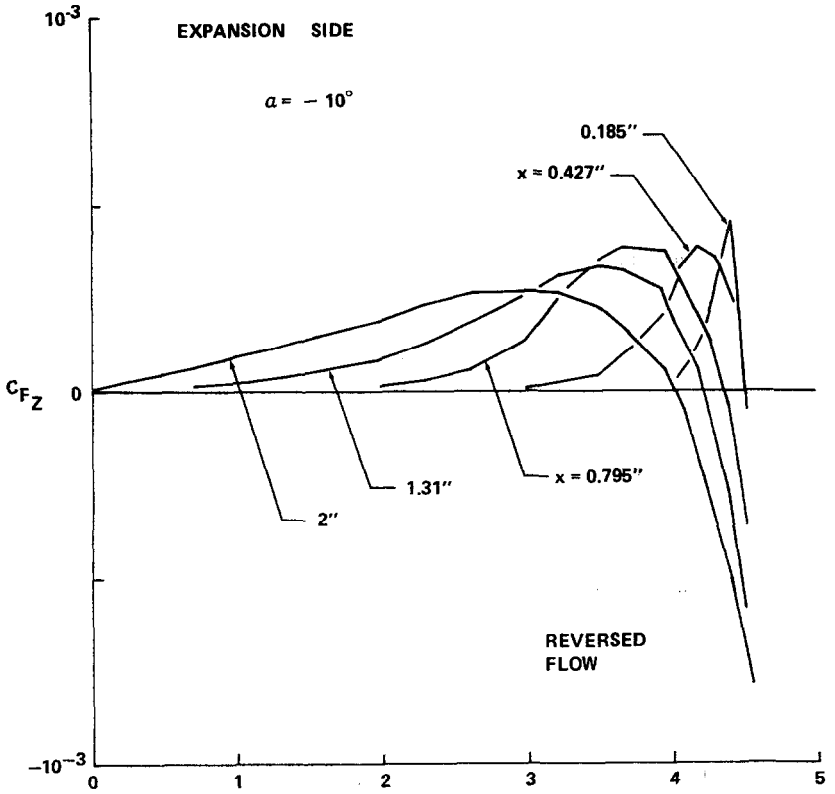


FIG. 7. Pressure distributions along $y = 0$ on compression and expansion sides of the plate for 10 degrees of flow incidence. (a) $x = 0.185$ in., (b) $x = 0.796$ in., (c) $x = 2$ in.



(a)



(b)

FIG. 8. Variation of lateral friction coefficients on the plate at several x-stations. (a) Compression side. (b) Expansion side.

region close to the side edge. Shown in Fig. 8 are the skin friction coefficients, $\tau_{yz}/\frac{1}{2}\rho_\infty U_\infty^2$, versus the z-coordinate on either side of the plate. The influence of the side edge is felt at an earlier station on the center plane. The lateral extent of reversal is seen to be proportional to the streamwise coordinate, and its magnitude becomes stronger as x increases. Figure 9 shows that the transverse extent of the cross-flow reversal also increases with x. For a plate of aspect ratio equal to unity, the flow on the plane of symmetry is definitely not two-dimensional when the plate is placed at an angle to the free stream.

All the computations were performed on an IBM 370-168 computer using approximately 4 min of central processing unit time. The computation time required can vary from one case to another because, even with the same number of cells and streamwise steps, some may need more iterations in solving the nonlinear difference equations than others. An understanding of flow characteristics is helpful since the careful selection of computational region and mesh could reduce the requirement of computation time and increase the accuracy of the results.

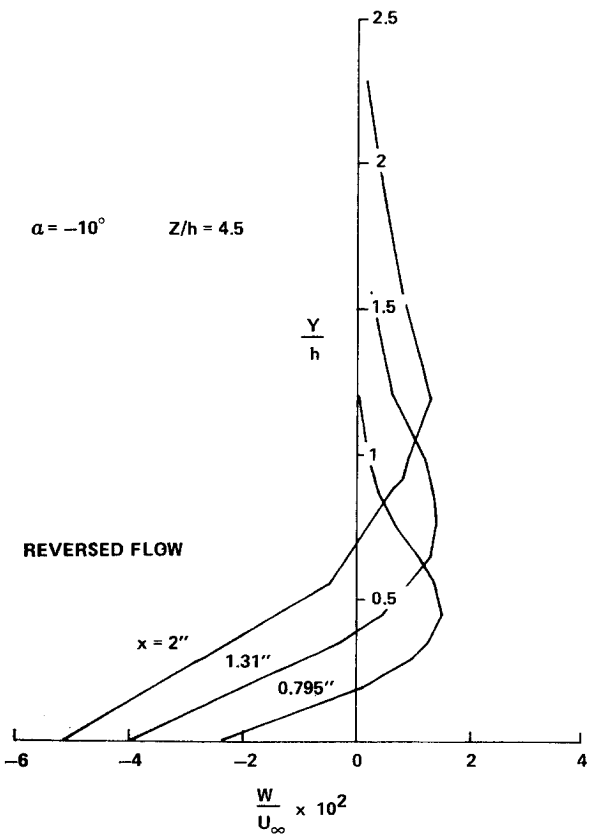


FIG. 9. Profiles of reversed lateral velocities at the side edge on the expansion side of the plate.

VII. Conclusions

The finite-difference algorithms constructed using the method of fractional steps have shown promising potential in solving the parabolic Navier–Stokes equations. They are featured with the conservative-law formulation in the difference equations and with a linearization procedure to maintain consistency with conservative law. If the method is used after linearizing the governing equations, other variants of differing capabilities can be devised. This effort is being pursued in the attempt to improve computational efficiency.

This study has concentrated on the computation of viscous flow passing a finite-width plate and has utilized both centered and backward differencing schemes to integrate the equations along the streamwise coordinate. The applicability of the centered scheme has been found to be susceptible to shocks and shear layers in the flowfield and, hence, this scheme is not recommended in such calculations. The solutions obtained from the backward scheme appear to be stable and reasonably accurate.

Conclusions pertaining to the flat-plate problem at flow incidence are summarized as follows:

(1) This work has suggested the existence of a shear layer around the side edge of the plate. Its intensity reduces from the peak value near the leading edge as the flow moves downstream, but it remains discernible well into the weak-interaction regime.

(2) The pressure distribution along the centerline is more sensitive than other measurable quantities to the flow expansion about the edge. At a small flow incidence angle, the flow should be treated three-dimensionally when the aspect ratio, W/L, is close to unity.

(3) The parabolic formulation is restricted to a class of viscous flow problems which do not involve a local reversed flow region in the direction of the main stream. A related problem concerning the flowfield in a rectangular corner has also been used to justify the present method of solution; the results will be reported in the near future.

References

1. C. T. Nardo and R. J. Cresci, *J. Comput. Phys.* **8** (1971), 268.
2. S. G. Rubin and T. C. Lin, *J. Comput. Phys.* **9** (1972), 339.
3. T. C. Lin and S. G. Rubin, Computers and Fluids **1** (1973), 37.
4. W. S. Helliwell and S. C. Lubard, *Comput. Fluids* **3** (1975), 83.
5. S. V. Patankar and D. B. Spalding, *Int. J. Heat Mass Transfer* **15** (1972), 1787.
6. W. R. Briley, *J. Comput. Phys.* **14** (1974), 8.
7. N. N. Yanenko, "The Method of Fractional Steps," Springer-Verlag, Berlin/Heidelberg, 1975.
8. S. I. Cheng, "A Critical Review of Numerical Solution of Navier–Stokes Equations in Progress in Numerical Fluid Dynamics" (H. J. Wirz, Ed.), Springer-Verlag, Berlin/Heidelberg, 1975.

9. W. M. BEAM AND R. F. WARMING, AIAA Paper No. 77-645, presented at AIAA 3rd Computational Fluid Dynamics Conference, 1977.
10. W. R. BRILEY AND H. MCDONALD, *J. Comput. Phys.* **24** (1977), 372.
11. J. DOUGLAS, "Advances in Computers" (F. L. Alt. Ed.), Vol. 2, Academic Press, New York, 1961.
12. C. P. LI, AIAA Paper No. 77-168, AIAA 15th Aerospace Sciences Meeting, 1977.
13. E. ISAACSON AND H. B. KELLER, "Analysis of Numerical Methods," Wiley, New York, 1966.
14. R. D. RICHTMEYER AND K. W. MORTON, "Difference Methods for Initial-Value Problems," 2nd ed., Interscience, New York, 1967.
15. C. P. LI, "Viscous Flowfield Solutions on a Sharp Plate," Technical Report No. TR-7005 (LEC-11421), Lockheed Electronics Company, Inc., Houston, Texas, January 1978.