77-645

AN IMPLICIT FACTORED SCHEME FOR THE COMPRESSIBLE NAVIER-STOKES EQUATIONS

Richard M. Beam* and R. F. Warming†
Ames Research Center, NASA, Moffett Field, California

Abstract

An implicit finite difference scheme is developed for the numerical solution of the compressible Navier-Stokes equations in conservation-law form. The algorithm is second-order-time accurate, noniterative, and spatially factored. In order to obtain an efficient factored algorithm, the spatial cross-derivatives are evaluated explicitly. However, the algorithm is unconditionally stable and, although a three-time-level scheme, requires only two-time-levels of data storage. The algorithm is constructed in a "delta" form (i.e., increments of the conserved variables and fluxes) that provides a direct derivation of the scheme and leads to an efficient computational algorithm. In addition, the delta form has the advantageous property of a steady-state (if one exists) independent of the size of the time step. Numerical results are presented for a two-dimensional shock boundary-layer interaction problem.

I. Introduction

Numerical computations based on the full compressible Navier-Stokes equations first appeared slightly more than a decade ago. During the relatively brief intervening period, considerable advancement has been made in the calculation of both two- and three-dimensional flow fields. A comprehensive summary of finite-difference methods and calculations for the 1965 to 1975 period has been made by Peyret and Viviand[1] and we will not attempt to duplicate their review. Both explicit and implicit numerical methods have been successfully applied to a variety of flow calculations and neither method has reached its full potential. Traditionally, implicit numerical methods have been praised for their improved stability and condemned for their large arithmetic operation counts. Hence the choice of an implicit algorithm implies that the time-step limit imposed by an explicit stability bound must be significantly less than the time-step limit imposed by the accuracy bound. This situation commonly arises in the numerical solution of a time dependent system of flow equations and results from disparate characteristic speeds and/or length scales. (Such problems are often said to be "stiff.") Recent interest in implicit methods has been spurred by the development of improved nonitiverative algorithms[2-4] and the trend of current computer hardware development to be limited by data transfer speed rather than the speed of arithmetic units.

An efficient implicit finite-difference algorithm for the Eulerian (inviscid) gasdynamic equations in conservation-law form was recently developed.[4] The purpose of this paper is to extend that algorithm to include the compressible Navier-Stokes equations (Section II). The extended algorithm is noniterative and retains the conservation-law form which is essential for the proper treatment of embedded shock waves ("shock capturing"). The temporal difference approximation has been generalized to retain a variety of schemes including a three-level scheme requiring only two levels of data storage. A three-level scheme allows the spatial cross-derivative terms to be efficiently included in a spatially factored second-order-time-accurate algorithm without upsetting the unconditional stability of the algorithm. The development and final algorithm make extensive use of the "delta" form (increments of the conserved variable and flux vectors) to achieve analytical simplicity and numerical efficiency. The delta formulation also retains the advantageous property of a steady-state (if one exists) independent of the time step.

In Section III we develop an implicit time-dependent boundary-condition scheme. We consider two physical problems that provide a variety of boundary conditions. A linear stability analysis, based on model two-dimensional convective and diffusive scalar equations, is summarized in Section IV. The analysis indicates that the factored, second-order-accurate scheme is unconditionally stable. A method for adding numerical dissipation, when required, is presented in Section V.

Numerical examples in Section VI include the transient development of Couette flow and the oscillatory flow generated by a wall moving with sinusoidal velocity in its own plane. The purpose of these simple flow calculations was to test the algorithm and boundary conditions on unsteady problems for which the exact solutions are known. As a more severe test of the algorithm, the numerical solution of a two-dimensional shock-boundary-layer interaction flow was computed. The results of the numerical examples indicate numerical stability and accuracy for Courant numbers much greater than unity.

II. Algorithm Development

The two-dimensional compressible Navier-Stokes equations can be written in the conservation-law form

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = \frac{\partial V_1(U,U_x)}{\partial x} + \frac{\partial V_2(U,U_y)}{\partial x} + \frac{\partial W_1(U,U_x)}{\partial y} + \frac{\partial W_2(U,U_y)}{\partial y} \tag{1}$$

where U is the vector of conserved variables and F, G, V, and W are flux vectors (see Appendix for details). A single-step temporal scheme for advancing the solution of (1) is (from Ref. 5)

$$\Delta U^n = \frac{\theta\Delta t}{1+\xi}\frac{\partial}{\partial t}\Delta U^n + \frac{\Delta t}{1+\xi}\frac{\partial}{\partial t}U^n + \frac{\xi}{1+\xi}\Delta U^{n-1} + O[(\theta - \tfrac{1}{2} - \xi)\Delta t^2 + \Delta t^3] \tag{2}$$

where $U^n = U(n\Delta t)$ and $\Delta U^n = U^{n+1} - U^n$. The time differencing formula (2), with the appropriate choice of the parameters ξ and θ, reproduces many familiar two- and three-level, explicit and implicit schemes as listed in Table 1. In addition it encompasses other variations including virtually all the

*Research Scientist.
†Research Scientist. Member AIAA.

Table 1 Partial list of schemes contained in Eq. (2)

θ	ξ	Scheme	Truncation error
0	0	Euler, explicit	$O(\Delta t^2)$
0	$-\frac{1}{2}$	Leapfrog, explicit	$O(\Delta t^3)$
$\frac{1}{2}$	0	Trapezoidal, implicit	$O(\Delta t^3)$
1	0	Euler, implicit	$O(\Delta t^2)$
1	$\frac{1}{2}$	3-point-backward, implicit	$O(\Delta t^3)$

time difference approximations for the diffusion equation given by Richtmyer and Morton (Ref. 6, p. 189). Note that scheme (2) is second-order-accurate when $\theta = 1/2 + \xi$ and first-order-accurate otherwise. In the applications which follow we are primarily interested in the three-level, second-order-accurate scheme $\theta = 1$, $\xi = 1/2$. In this paper we will not consider the explicit schemes; however, the versatility of the more general class of schemes in the analytical development and the algorithm programming can be achieved with a modicum of effort.

If Eq. (1) is solved for $\partial U/\partial t$ (i.e., F_x and G_y are moved to the right side) and the resulting expression for the temporal derivative is inserted in (2), we obtain

$$\Delta U^n = \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}\left(-\Delta F^n + \Delta V_1{}^n + \Delta V_2{}^n\right) + \frac{\partial}{\partial y}\left(-\Delta G^n + \Delta W_1{}^n + \Delta W_2{}^n\right)\right] + \frac{\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}\left(-F + V_1 + V_2\right)^n + \frac{\partial}{\partial y}\left(-G + W_1 + W_2\right)^n\right] + \frac{\xi}{1+\xi}\Delta U^{n-1} + O\left[\left(\theta - \frac{1}{2} - \xi\right)\Delta t^2 + \Delta t^3\right] \tag{3}$$

where $F^{n+1} = F(U^{n+1})$, $\Delta F^n = F^{n+1} - F^n$, etc. In Eq. (3) and in the equations to follow, the vector denoted by the symbol $U^n = U(n\Delta t)$ is assumed to be a solution of the partial-differential equation (1). When $\partial/\partial x$ and $\partial/\partial y$ are approximated by difference quotients, then the symbol U^n is replaced by $U^n_{i,j}$ where $x = i\Delta x$, $y = j\Delta y$, and the order symbol $O[(\theta - 1/2 - \xi)\Delta t^2 + \Delta t^3]$ will be dropped. The resulting formula will then be the numerical algorithm and $U^n_{i,j}$ will denote the numerical solution.

If the spatial derivatives of (3) were approximated by finite differences, the first obvious difficulty in solving the algebraic equations for ΔU^n would be the nonlinearity of the set of equations. The nonlinearity is a consequence of the fact that the flux vector increments (ΔF^n, ΔG^n, ΔV^n, ΔW^n) are nonlinear functions of the conserved variables U. A linear equation with the same temporal accuracy as Eq. (3) can be obtained if we use the Taylor series expansion

$$F^{n+1} = F^n + \left(\frac{\partial F}{\partial U}\right)^n (U^{n+1} - U^n) + O(\Delta t^2)$$

or

$$\Delta F^n = A^n \Delta U^n + O(\Delta t^2) \tag{4a}$$

where A is the Jacobian matrix $\partial F/\partial U$ (Appendix, Eq. (A13)). Likewise

$$\Delta G^n = \left(\frac{\partial G}{\partial U}\right)^n \Delta U^n + O(\Delta t^2) = B^n \Delta U^n + O(\Delta t^2) \tag{4b}$$

$$\Delta V_1{}^n = \left(\frac{\partial V_1}{\partial U}\right)^n \Delta U^n + \left(\frac{\partial V_1}{\partial U_x}\right)^n \Delta U_x{}^n + O(\Delta t^2) = P^n \Delta U^n + R^n \Delta U_x{}^n + O(\Delta t^2) = (P - R_x)^n \Delta U^n + \frac{\partial}{\partial x}(R\Delta U)^n + O(\Delta t^2) \tag{4c}$$

where R is the Jacobian $\partial V_1/\partial U_x$ and $R_x = \partial R/\partial x$. Similarly,

$$\Delta W_2{}^n = (Q - S_y)^n \Delta U^n + \frac{\partial}{\partial y}(S\Delta U)^n + O(\Delta t^2) \tag{4d}$$

A second but perhaps less obvious difficulty arises from the spatial cross-derivative terms $\partial V_2/\partial x$ and $\partial W_1/\partial y$. If these terms were treated in the same manner as (4) we would encounter considerable difficulty in constructing an efficient spatially factored algorithm. Another method of treating these cross-derivative terms is to evaluate them explicitly. This can be done without loss of accuracy and with minimal computational effort if we note that

$$\Delta V_2{}^n = \Delta V_2{}^{n-1} + O(\Delta t^2), \quad \Delta W_1{}^n = \Delta W_1{}^{n-1} + O(\Delta t^2) \tag{5}$$

for a uniform time step Δt. One might anticipate that the explicit treatment of the cross-derivative terms would have an adverse effect on the numerical stability; however, the final factored implicit algorithm will be unconditionally stable (Section IV).

If the approximations (4) and (5) are introduced in (3) we obtain

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(A - P + R_x)^n - \frac{\partial^2}{\partial x^2}(R)^n + \frac{\partial}{\partial y}(B - Q + S_y)^n - \frac{\partial^2}{\partial y^2}(S)^n\right]\right\}\Delta U^n = \frac{\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(-F + V_1 + V_2)^n + \frac{\partial}{\partial y}(-G + W_1 + W_2)^n\right] + \frac{\bar\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(\Delta V_2)^{n-1} + \frac{\partial}{\partial y}(\Delta W_1)^{n-1}\right] + \frac{\xi}{1+\xi}\Delta U^{n-1} + O\left[\left(\theta - \frac{1}{2} - \xi\right)\Delta t^2, (\bar\theta - \theta)\Delta t^2, \Delta t^3\right] \tag{6}$$

where a $\bar\theta$ has been introduced in the coefficient of the cross-derivative terms for notational convenience.‡ For second-order-accurate schemes $\bar\theta$ should be set equal to θ. However, for first-order-accurate schemes ($\theta \neq 1/2 + \xi$) it is consistent and, for some calculations, advantageous to set $\bar\theta$ equal to zero. The spatially-factored form of (6) which retains the temporal accuracy can be easily obtained if we note that

‡In Eq. (6) and throughout the remainder of this paper, notation of the form

$$\left[\frac{\partial}{\partial x}(A - P + R_x)^n\right]\Delta U^n$$

denotes

$$\frac{\partial}{\partial x}\left[(A - P + R_x)^n \Delta U^n\right], \quad \text{etc.}$$

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(A - P + R_x)^n - \frac{\partial^2}{\partial x^2}(R)^n\right]\right\} \times$$

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial y}(B - Q + S_y)^n - \frac{\partial^2}{\partial y^2}(S)^n\right]\right\}\Delta U^n$$

$$= \mathrm{LHS}(6) + O(\Delta t^3) \qquad (7)$$

where LHS (6) is used to indicate the left-hand side of Eq. (6). Thus a spatially factored algorithm with the same temporal accuracy as (3) but linear in ΔU^n is

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(A - P + R_x)^n - \frac{\partial^2}{\partial x^2}(R)^n\right]\right\} \times$$

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial y}(B - Q + S_y)^n - \frac{\partial^2}{\partial y^2}(S)^n\right]\right\}\Delta U^n$$

$$= \frac{\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(-F + V_1 + V_2)^n + \frac{\partial}{\partial y}(-G + W_1 + W_2)^n\right]$$

$$+ \frac{\bar{\theta}\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(\Delta V_2)^{n-1} + \frac{\partial}{\partial y}(\Delta W_1)^{n-1}\right]$$

$$+ \frac{\xi}{1+\xi}\Delta U^{n-1} + O\left[\left(\theta - \frac{1}{2} - \xi\right)\Delta t^2, (\theta - \bar{\theta})\Delta t^2, \Delta t^3\right] \qquad (8)$$

In practice (8) is implemented by the sequence

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(A - P + R_x)^n - \frac{\partial^2}{\partial x^2}(R)^n\right]\right\}\Delta U^* = \mathrm{RHS}(8) \qquad (9a)$$

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial y}(B - Q + S_y)^n - \frac{\partial^2}{\partial y^2}(S)^n\right]\right\}\Delta U^n = \Delta U^* \qquad (9b)$$

$$U^{n+1} = U^n + \Delta U^n \qquad (9c)$$

The spatial derivatives appearing in (9) are to be approximated by appropriate finite-difference quotients. For example, the following three-point second-order-accurate central difference approximations were used for the numerical computations described below:

$$\left.\frac{\partial f}{\partial x}\right|_{i,j} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} + O(\Delta x^2) \qquad (10a)$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + O(\Delta x^2) \qquad (10b)$$

with analogous formulae for the y derivatives. With three-point central difference approximations, the x- and y-operators on the left side of (9a,b) each require the solution of a block-tridiagonal system of equations with each block having dimensions q by q, where q is the number of components of U ($q = 4$ for the two-dimensional Navier-Stokes equations). However, the block-tridiagonal solution algorithm is the same as that required in the original algorithm[4] for the Eulerian (inviscid) equations. The additional computational effort generated by the viscous terms is reflected in the evaluation of the coefficient matrices P, R, Q, and S and the flux vectors V and W.

Although (8) contains three time levels of data ($n+1$, n, $n-1$), only two "levels" of data, U and ΔU, need be stored for each spatial grid point. The computation of the spatial differences of the incremental V_2 and W_1 (cross-derivative terms in RHS(8)) requires the "reconstruction" of U^{n-1}; however, the cost of computing these terms is only a few percent of the total computation cost.

The Jacobian matrices A, B, R, and S (Eqs. (4)) have relatively simple elements (see Appendix; Eqs. (A.13) through (A.16)). In general the viscous coefficients, λ and μ, are functions of the temperature which is a function of the elements of U (A.7). Consequently, the Jacobian matrices P and Q have quite complex elements in the most general case. For some cases certain physical approximations can be made that significantly simplify the calculations. For example, if the viscous coefficients are changing slowly with time, the Jacobians P and Q (4c,d) can be adequately represented by neglecting the dependence of λ and μ on U (i.e., Eqs. (A.17) and (A.18)). Further simplifications occur if the viscous coefficients are assumed to be locally constant in which case

$$-P + R_x = 0 \qquad (11a)$$

$$-Q + S_y = 0 \qquad (11b)$$

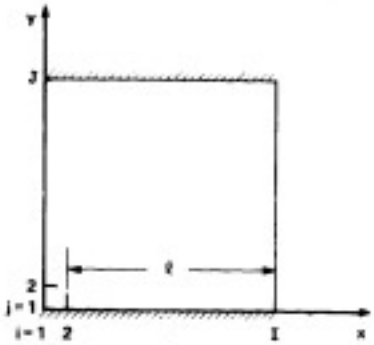
and the LHS (9a,b) contain only the Jacobians A, B, R, and S. Note that the steady-state solution (if one exists), which forms part of the RHS (8), is not affected by the assumptions made in computing the Jacobians.

III. Boundary Conditions

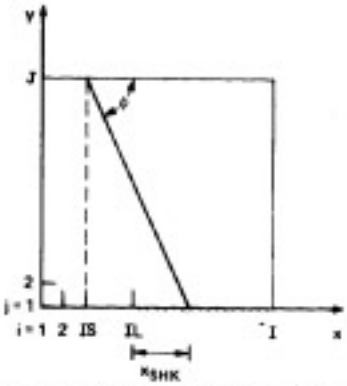
The application of the algorithm (9) at the boundaries of the computation region can be conveniently explained by stepping through the sequence of operations required to advance the solution from level n to $n+1$. Two physical problems, which provide a variety of boundary conditions, are considered in the following discussion. For both problems the spatial computational domain is divided into a rectangular Cartesian grid $x = i\Delta x$, $y = j\Delta y$.

The first problem (Fig. 1a) is a Couette flow where the upper and lower boundaries ($j = 1, J$) are rigid walls which may have nonzero velocity in their own plane. The other boundaries ($i = 1, I$) are prescribed by spatially periodic (period = ℓ) boundary conditions.

The second problem (Fig. 1b), a shock-boundary-layer interaction, has a supersonic flow into the region $i = 1$, $j = 1, J$. The flat plate ($j = 1$, $i = IL$ to I) is a rigid wall and ahead of the leading edge ($j = 1$, $i = 2$ to $IL-1$) a y-symmetry condition is applied. The upper boundary conditions ($j = J$) are chosen to generate the shock-wave that impinges on the flat plate. Ahead of the shock ($j = J$, $i = 1$ to $IS - 1$) the supersonic inflow conditions are chosen and behind the shock ($j = J$, $i = IS$ to I) the post-shock conditions are set. At the final boundary ($i = I$, $j = 1$ to J) the outflow conditions are parabolic in the boundary layer and hyperbolic in the freestream which permit extrapolation of data from the interior to the boundary.



(a) Couette flow mesh.



(b) Shock boundary layer mesh.

Fig. 1 Indexing of computational mesh for flow calculations.

Next we consider the details of the numerical application of the boundary conditions. First, we consider the explicit portion of the boundary conditions and then the implicit portion.

Explicit Portion

At the start of the calculation ($t = n\Delta t = 0$) we assume that a complete description of the initial flow field is provided at each mesh point. In general, the algorithm (9) is a three-level scheme and two levels of starting data are required. However, if two levels of initial data are not available, the second level can be obtained by applying the algorithm (9) as a two-level scheme for one step; for example, $\bar{\theta} = \xi = 0$.

The first encounter with the boundary conditions occurs in the evaluation of the "steady-state" portion of the right side of (8); that is,

$$\frac{\partial}{\partial x}\left(-F^n + V_1{}^n + V_2{}^n\right) + \frac{\partial}{\partial y}\left(-G^n + W_1{}^n + W_2{}^n\right) \qquad (12)$$

Expression (12) is called the steady-state portion because this quantity will be equal to zero if the solution converges (i.e., $\Delta U^N = 0$). The spatial accuracy of this portion of the calculation thus determines the spatial accuracy of the steady-state solution.

Let us begin with the treatment of a rigid boundary (Fig. 1a, $j = 1$). If three-point central differences are used to approximate $\partial/\partial x$ and $\partial/\partial y$ in (12), the boundary $j = 1$ is encountered in the $\partial/\partial y$ approximation and we require

$$\left(-G^n + W_1{}^n + W_2{}^n\right)_{j=1} \qquad (13)$$

where, for simplicity, the i index is suppressed. Since u and v are prescribed on the boundary ($u_{j=1} = 0$, $v_{j=1} = 0$) and T can be obtained from the condition at the wall (for example by extrapolation from the flow field at an adiabatic wall, $\partial T/\partial y = 0$), the W part of (13) (see Appendix, Eq. (A.5)) is easily obtained by using one-sided difference approximations for $(\partial u/\partial y)_{j=1}$ and $(\partial v/\partial y)_{j=1}$; for example,

$$\left.\frac{\partial u}{\partial y}\right|_{j=1} = \frac{-3u_1 + 4u_2 - u_3}{2\Delta y} + O(\Delta y^2) \qquad (14)$$

The no-slip boundary condition ($u_{j=1} = 0$, $v_{j=1} = 0$) simplifies the computation of $G_{j=1}$, Eq. (A.3), to the evaluation of $p_{j=1}$; that is,

$$G_{j=1} = \begin{bmatrix} 0 \\ 0 \\ p \\ 0 \end{bmatrix}_{j=1} \qquad (15)$$

We use the normal momentum equation

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x}\left(\frac{mn}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{n^2}{\rho} + p\right) - \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right]$$
$$- \frac{\partial}{\partial y}\left[(\lambda + 2\mu)\frac{\partial v}{\partial y} + \lambda\frac{\partial u}{\partial x}\right] = 0 \qquad (16)$$

with the no-slip boundary condition ($u_{j=1} = v_{j=1} = 0$) to obtain

$$\left.\frac{\partial p}{\partial y}\right|_{j=1} = \left\{\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left[(\lambda + 2\mu)\frac{\partial v}{\partial y}\right]\right\}_{j=1} \qquad (17)$$

If $\partial p/\partial y$ is approximated by a one-sided difference quotient, for example

$$\left.\frac{\partial p}{\partial y}\right|_{j=1} = \frac{p_2 - p_1}{\Delta y} + O(\Delta y) \qquad (18)$$

and the right side of (17) is evaluated by an appropriate difference approximation (u and v are known at all grid points), then we obtain an explicit expression for p on the boundary (p_1). If the rigid wall were moving in its own plane ($u_{j=1} = u_0(t)$, $v_{j=1} = 0$) the analysis would proceed as before except with the proper prescription of $u_{j=1}$. The upper boundary ($j = J$) of the Couette flow problem is identical in treatment to the lower boundary. At the other boundaries, $i = 1$, $i = I$, spatially periodic boundary conditions are applied, that is, $U_I = U_2$, $U_1 = U_{I-1}$.

The supersonic inflow boundary conditions for the shock-boundary-layer problem (Fig. 1b) were fixed at free-stream values $U_{i=1} = U_{\text{free stream}}$. These same free-stream conditions were applied at the upper boundary ahead of the shock ($j = J$, $i = 1$ to $IS - 1$). The post-shock conditions were fixed at the remaining upper boundary points ($j = J$, $i = IS$ to I). Since the character of the flow in the boundary layer far downstream is parabolic and the flow in the free stream hyperbolic (supersonic), the outflow conditions were obtained by simple

extrapolation, $U_1 = U_{I-1}$. The low-order extrapolation is used since no influence should be felt upstream in the region of the shock-boundary-layer interaction. The boundary conditions upstream of the plate leading edge ($j = 1$, $i = 1$ to $IL - 1$) were obtained from the flow symmetry conditions about the x-axis,

$$\left.\frac{\partial u}{\partial y}\right|_{j=1} = \left.\frac{\partial T}{\partial y}\right|_{j=1} = \left.\frac{\partial p}{\partial y}\right|_{j=1} = v\big|_{j=1} = 0$$

and appropriate one-sided difference approximations; for example, (18).

Implicit Portion

The previous discussion dealt with the steady-state portion of the calculation. The treatment at the boundaries was identical to that one might use for an explicit numerical scheme. We consider now the remainder of the algorithm and the effect of the implicit algorithm on the treatment of the boundary conditions.

After the computation of the RHS (8) we proceed to the implicit x-sweep (Eq. (9a)). Three-point central difference approximations for the derivatives $\partial/\partial x$ and $\partial^2/\partial x^2$ produce a system of block-tridiagonal equations

$$L_{i-1}\Delta U^*_{i-1} + M_i \Delta U_i^* + N_{i+1}\Delta U^*_{i+1} = H_i\ , \quad i = 2,\ I - 1 \tag{19}$$

where L, M, and N are 4×4 matrices. The periodic boundary condition for the Couette flow ($U_I = U_2$, $U_1 = U_{I-1}$) applied to (19) produces the periodic block-tridiagonal system

$$\begin{bmatrix} M_2 & N_3 & & & L_1 \\ L_2 & M_3 & N_4 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{I-3} & M_{I-2} & N_{I-1} \\ N_I & & & L_{I-2} & M_{I-1} \end{bmatrix} \begin{bmatrix} \Delta U_2^* \\ \Delta U_3^* \\ \vdots \\ \Delta U^*_{I-2} \\ \Delta U^*_{I-1} \end{bmatrix} = \begin{bmatrix} H_2 \\ H_3 \\ \vdots \\ H_{I-2} \\ H_{I-1} \end{bmatrix} \tag{20}$$

Although about twice as costly in computer time as the nonperiodic block-tridiagonal solvers, the solution algorithms for (20) are available.[7] For the shock boundary-layer problem Eq. (19) is still applicable; however, the boundary conditions are $U_1 = U_{\text{free stream}}$ and $U_I = U_{I-1}$ which produce the system

$$\begin{bmatrix} M_2 & N_3 & & & \\ L_2 & M_3 & N_4 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{I-3} & M_{I-2} & N_{I-1} \\ & & & L_{I-2} & (M_{I-1} + N_I) \end{bmatrix} \begin{bmatrix} \Delta U_2^* \\ \Delta U_3^* \\ \vdots \\ \Delta U^*_{I-2} \\ \Delta U^*_{I-1} \end{bmatrix} = \begin{bmatrix} H_2 \\ H_3 \\ \vdots \\ H_{I-2} \\ H_{I-1} \end{bmatrix} \tag{21}$$

After the computation of ΔU^* at the interior mesh points we are ready for the implicit y-sweep (Eq. (9b)). Again we use three-point central difference approximations to the spatial derivatives and, for simplicity, we assume locally constant viscous coefficients (11b) to obtain

$$\left[\frac{\theta\Delta t}{1+\xi}\left(-\frac{1}{2\Delta y}B^n_{j-1} - \frac{1}{\Delta y^2}S^n_{j-1}\right)\right]\Delta U^n_{j-1} + \left(I + \frac{\theta\Delta t}{1+\xi}\frac{2}{\Delta y^2}S_j^n\right)\Delta U_j^n + \left[\frac{\theta\Delta t}{1+\xi}\left(\frac{1}{2\Delta y}B^n_{j+1} - \frac{1}{\Delta y^2}S^n_{j+1}\right)\right]\Delta U^n_{j+1} = \Delta U_j^*$$

$$j = 2,\ J - 1 \tag{22}$$

For the two sample problems (Fig. 1) we encounter a rigid wall in the y sweep. For example, in the Couette flow problem the application of (22) at $j = 2$ introduces the quantity

$$H_B = \left[\frac{\theta\Delta t}{1+\xi}\left(-\frac{1}{2\Delta y}B_1^n - \frac{1}{\Delta y^2}S_1^n\right)\right]\Delta U_1^n \tag{23}$$

which requires data at the rigid boundary $j = 1$. If we introduce the no-slip boundary conditions ($u_1 = 0$, $v_1 = 0$) into B, S, and U (Eqs. (A.14), (A.16), and (A.1))

$$B_1^n\Delta U_1^n = -\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-\gamma) \\ 0 & 0 & \dfrac{\gamma e_1^n}{\rho_1^n} & 0 \end{bmatrix}\begin{bmatrix} \Delta\rho_1^n \\ 0 \\ 0 \\ \Delta e_1^n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (\gamma-1)\Delta e_1^n \\ 0 \end{bmatrix} \tag{24}$$

$$S_1^n\Delta U_1^n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \dfrac{\mu}{\rho_1^n} & 0 & 0 \\ 0 & 0 & \dfrac{\lambda+2\mu}{\rho^n} & 0 \\ -\dfrac{k}{c_v}\left(\dfrac{e_1}{\rho_1^2}\right)^n & 0 & 0 & \dfrac{k}{c_v}\dfrac{1}{\rho_1^n} \end{bmatrix}\begin{bmatrix} \Delta\rho_1^n \\ 0 \\ 0 \\ \Delta e_1^n \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{k}{c_v}\left(-\dfrac{e_1\Delta\rho_1}{\rho_1^2} + \dfrac{\Delta e_1}{\rho_1}\right)^n \end{bmatrix} \tag{25}$$

Hence, we need approximations for $[(\gamma-1)\Delta e^n]_{j=1}$ and $[(k/c_v)(-e\Delta\rho/\rho^2 + \Delta e/\rho)^n]_{j=1}$ as functions of the increments of the conservative variable at the interior points $j = 2,3$. Note that

$$\Delta\left(\frac{e}{\rho}\right)^n = \left(\frac{1}{\rho}\Delta e - \frac{e}{\rho^2}\Delta\rho\right)^n + O(\Delta t^2) \tag{26}$$

and at an adiabatic wall ($\partial T/\partial y|_{j=1} = 0$) with no-slip conditions ($u_{j=1} = v_{j=1} = 0$) we obtain from (A.7)

$$\left.\frac{\partial T}{\partial y}\right|_{j=1} = \frac{1}{c_v}\frac{\partial}{\partial y}\left(\frac{e}{\rho}\right)\Big|_{j=1} = 0 \tag{27}$$

If we use a one-sided 3-point difference approximation for the normal derivative

$$\frac{\partial}{\partial y}\left(\frac{e}{\rho}\right)\Big|_{j=1} = \frac{1}{2\Delta y}\left[-3\left(\frac{e}{\rho}\right)_1 + 4\left(\frac{e}{\rho}\right)_2 - \left(\frac{e}{\rho}\right)_3\right] + O(\Delta y^2) \tag{28}$$

and use (26) and (27) we obtain

$$\left(\frac{1}{\rho}\Delta e - \frac{e}{\rho^2}\Delta\rho\right)^n_{j=1} = \frac{1}{3}\left[4\left(\frac{1}{\rho}\Delta e - \frac{e}{\rho^2}\Delta\rho\right)^n_{j=2} - \left(\frac{1}{\rho}\Delta e - \frac{e}{\rho^2}\Delta\rho\right)^n_{j=3}\right] \tag{29}$$

which provides the desired expression for the right side of (25). Next we seek an approximation of the right side of (24). A relation between pressure and internal energy at the rigid wall can be obtained from (A.8)

$$\left.\frac{\partial p}{\partial y}\right|_{j=1} = (\gamma - 1)\left.\frac{\partial e}{\partial y}\right|_{j=1} \tag{30}$$

and between pressure and the velocities from the normal momentum equation at the wall, Eq. (16). In the present calculations we have neglected the cross-derivative terms in (16) to avoid the coupling with adjacent i mesh points at the boundary. A more accurate approach, which also avoids the implicit coupling of adjacent boundary points, would be to treat the cross-derivative terms explicitly (i.e., as functions of ΔU^{n-1}). This later approximation was not tested in the present calculations and we proceed by neglecting the cross-derivative terms; that is,

$$\left.\frac{\partial p}{\partial y}\right|_{j=1} \approx (\lambda + 2\mu)\left.\frac{\partial^2}{\partial y^2}\left(\frac{n}{\rho}\right)\right|_{j=1} \tag{31}$$

Again we use three-point one-sided difference approximations for the spatial derivatives and combine (30) and (31) to obtain

$$\frac{(\gamma-1)}{2\Delta y}\left[-3\Delta e_1{}^n + 4\Delta e_2{}^n - \Delta e_3{}^n\right] = \frac{(\lambda+2\mu)}{\Delta y^2}\left[\Delta\left(\frac{n}{\rho}\right)^n_1 - 2\Delta\left(\frac{n}{\rho}\right)^n_2 + \Delta\left(\frac{n}{\rho}\right)^n_3\right] + O(\Delta y^2) \tag{32}$$

Note that

$$\Delta\left(\frac{n}{\rho}\right)^n = \left(\frac{\Delta n}{\rho}\right)^n - \left(\frac{n}{\rho^2}\Delta\rho\right)^n + O(\Delta t^2) \tag{33}$$

and recall that $n = 0$ at the no-slip boundary (i.e., $n_1 = 0$); thus, (32) and (33) provide the expression for $\Delta e_1{}^n$ required in (24). If we now combine (23), (24), (25), (29), (32), and (33) we can write

$$H_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c_{31} & 0 & c_{33} & c_{34} \\ c_{41} & 0 & 0 & c_{44} \end{bmatrix}\Delta U_2{}^n + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ d_{31} & 0 & d_{33} & d_{34} \\ d_{41} & 0 & 0 & d_{44} \end{bmatrix}\Delta U_3{}^n \tag{34}$$

or

$$H_B = C_2\Delta U_2{}^n + D_3\Delta U_3{}^n \tag{35}$$

where c_{kl} and d_{kl} represent the nonzero elements of the "correction" matrices C_2 and D_3. Similar considerations can be included at the upper boundary ($j = J$). Finally, for the Couette problem the implicit y-sweep requires the solution of the block-tridiagonal system

$$\begin{bmatrix} (M+C)_2{}^n & (N+D)_3{}^n & & & \\ L_2{}^n & M_3{}^n & N_4{}^n & & \\ & \ddots & \ddots & \ddots & \\ & & L^n_{J-3} & M^n_{J-2} & N^n_{J-1} \\ & & & (L+D)^n_{J-2} & (M+C)^n_{J-1} \end{bmatrix}\begin{bmatrix} \Delta U_2{}^n \\ \Delta U_3{}^n \\ \vdots \\ \Delta U^n_{J-2} \\ \Delta U^n_{J-1} \end{bmatrix} = \begin{bmatrix} \Delta U_2{}^* \\ \Delta U_3{}^* \\ \vdots \\ \Delta U^*_{J-2} \\ \Delta U^*_{J-1} \end{bmatrix} \tag{36}$$

The remaining boundary conditions for the shock boundary-layer problem are combinations of those discussed above and, therefore, are not presented in detail.

IV. Stability

The numerical stability of the algorithm (8) was investigated by applying it to the model linear scalar convective (hyperbolic) equation

$$\frac{\partial u}{\partial t} + c_1\frac{\partial u}{\partial x} + c_2\frac{\partial u}{\partial y} = 0 \tag{37}$$

and the diffusive (parabolic) equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x\partial y} + c\frac{\partial^2 u}{\partial y^2} \tag{38}$$

subject to

$$a, c > 0\ , \quad b^2 < 4ac \tag{39}$$

The inequalities (39) are the conditions under which (38) is parabolic.

The details of the analysis are presented in Ref. 5. For example, it is shown that the factored second-order-temporal-accurate algorithm, that is,

$$\theta = \frac{1}{2} + \xi \tag{40}$$

applied to (37)

$$\left(1 + \frac{\theta\Delta t}{1+\xi} c_1 \frac{\partial}{\partial x}\right)\left(1 + \frac{\theta\Delta t}{1+\xi} c_2 \frac{\partial}{\partial y}\right)\Delta u^n = -\frac{\Delta t}{1+\xi}\left(c_1 \frac{\partial u}{\partial x} + c_2 \frac{\partial u}{\partial y}\right)^n + \frac{\xi}{1+\xi}\Delta u^{n-1} \tag{41}$$

is unconditionally stable for $\xi > 0$. Similarly it is shown that when the algorithm is applied to (38)

$$\left(1 - \frac{\theta\Delta t}{1+\xi} a \frac{\partial^2}{\partial x^2}\right)\left(1 - \frac{\theta\Delta t}{1+\xi} c \frac{\partial^2}{\partial y^2}\right)\Delta u^n = \frac{\Delta t}{1+\xi}\left(a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2}\right)^n + \frac{\theta\Delta t}{1+\xi} b\left(\frac{\partial^2 \Delta u}{\partial x \partial y}\right)^{n-1} + \frac{\xi}{1+\xi}\Delta u^{n-1} \tag{42}$$

the algorithm is unconditionally stable for the more stringent condition

$$\frac{(1+2\xi)^3}{1+\xi} \geq 4 \tag{43}$$

or

$$\xi \geq 0.385 \tag{44}$$

In the numerical examples considered below we chose the 3-point-backward scheme (Table 1), $\xi = 1/2$, when second-order-temporal accuracy was desired.

V. Added Higher-Order Dissipation

In our applications with central spatial difference approximations, we have found it necessary to add dissipative terms to damp the short wave lengths. We chose fourth-order terms which were appended to the algorithm (9) as follows:

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial x}(A - P + R_x)^n - \frac{\partial^2}{\partial x^2}(R)^n\right]\right\}\Delta U^* = \mathrm{RHS}(8) - \frac{\Delta x^4}{1+\xi}\frac{\omega_x}{8}\frac{\partial^4}{\partial x^4}U^n \tag{45a}$$

$$\left\{I + \frac{\theta\Delta t}{1+\xi}\left[\frac{\partial}{\partial y}(B - Q + S_y)^n - \frac{\partial^2}{\partial y^2}(S)^n\right]\right\}\Delta U^n = \Delta U^* - \frac{\Delta y^4}{1+\xi}\frac{\omega_y}{8}\frac{\partial^4}{\partial y^4}(U^n + \Delta U^*) \tag{45b}$$

$$U^{n+1} = U^n + \Delta U \tag{45c}$$

The dissipative terms are of higher order and consequently do not disrupt the formal accuracy of the method. For the calculations, the fourth derivatives in (45) were replaced by the finite-difference approximations

$$\Delta x^4 \frac{\partial^4}{\partial x^4} U\Big|_{i,j} \approx U_{i+2,j} - 4U_{i+1,j} + 6U_{i,j} - 4U_{i-1,j} + U_{i-2,j} \tag{46a}$$

$$\Delta y^4 \frac{\partial^4}{\partial y^4} U\Big|_{i,j} \approx U_{i,j+2} - 4U_{i,j+1} + 6U_{i,j} - 4U_{i,j-1} + U_{i,j-2} \tag{46b}$$

According to a linear von Neumann stability analysis, the stable range of dissipative coefficients, ω_x and ω_y, is $0 \leq \omega \leq 1 + 2\xi$. At mesh points adjacent to rigid boundaries (e.g., $j = 2$, Fig. 1a) the data required for mesh points "outside" the boundary (i.e., $U_{i,o}$ Eq. (46b)) were obtained by setting $\rho_{i,o} = \rho_{i,2}$, $u_{i,o} = -u_{i,2}$, $v_{i,o} = -v_{i,2}$, and $e_{i,o} = e_{i,2}$. At mesh points adjacent to nonrigid boundaries, the dissipative coefficients were set equal to zero.

VI. Numerical Results

Couette Flow

The calculation of unsteady flow between two infinite parallel walls was chosen as an initial test of the temporal as well as spatial accuracy and stability of the numerical algorithm and boundary conditions. A 6×11 uniform grid ($I = 6$, $J = 11$) with periodic spatial boundary conditions in the x-direction (Fig. 1a) was used in the calculations. In the first calculation (Fig. 2) the flow field and upper boundary were initially at rest and the lower boundary had initial velocity u_o in its own plane. The transient development of the velocity profile between the two walls is shown in Fig. 2. The exact (incompressible) solution (see, e.g., Schlichting[8]) is shown for comparison. The Courant number for this calculation was approximately one. If larger Courant numbers are used (i.e., greater Δt), the transient solutions deviate from the exact solution; however, the correct steady-state solution was reached in a smaller number of time steps — for a Courant number of 100 the steady-state solution was obtained after 10 time steps.

The next example was chosen to show that accurate temporal solutions can be obtained at Courant numbers much greater than unity. In this calculation the lower wall was moving with sinusoidal velocity in its own plane. The time step Δt was chosen as 1/40 of the period of the oscillation.

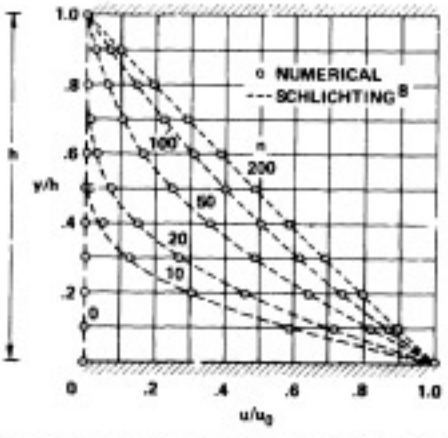


Fig. 2 Flow formation in Couette motion: $u_o = 100$ ft/sec, $h = 0.1\times10^{-4}$ ft, $\Delta t = 0.116\times10^{-8}$ sec, $t = n\Delta t$; initial data: $\rho = 0.00234$ lb sec^2/ft^4, $T = 527.7$ °R, $\mu = 0.378\times10^{-6}$ lb sec/ft^2, $Re_h = 6.2$, Mach number = 0.09.

Comparison is made with the exact (incompressible) analytical solution[8] in Fig. 3. Although the Courant number for the calculation was 10, the agreement with the analytical solution is good for the entire period of oscillation.

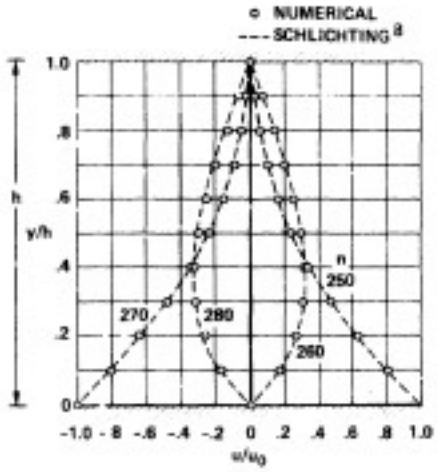


Fig. 3 Velocity distribution (after transient decay) between a moving wall, $u(0,t) = u_0 \sin(\omega t)$, and a stationary wall, $u(h,t) = 0$: $u_0 = 100$ ft/sec, $h = 0.1\times10^{-4}$ ft, $\Delta t = 0.116\times10^{-7}$ sec, $t = n\Delta t$; initial data: $\rho = 0.00234$ lb sec^2/ft^4, $T = 527.7°$ R, $\mu = 0.378\times10^{-6}$ lb sec/ft^2, $\omega = 2\pi/40\Delta t$, $Re_h = 6.2$, Mach number = 0.09.

Shock-Boundary-Layer Interaction

The second problem, Figs. 1b and 4, presents a more severe test for the algorithm. A shock wave interacts with the boundary layer that develops on the flat plate. If the shock wave has sufficient strength it will cause boundary-layer separation (as depicted in Fig. 4).

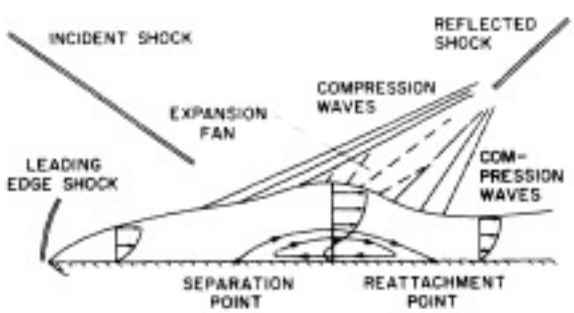


Fig. 4 Sketch of shock boundary-layer interaction.

The shock angle ϕ (Fig. 1b) was set equal to 32.6° by proper selection of the post-shock boundary conditions. The parameters IL = 5 and IS = 2 were selected so that $x_{SHK} = 0.16$ ft. The free-stream Mach number was 2.0 and the Reynolds number $Re_{x_{SHK}} = 0.296\times10^6$.

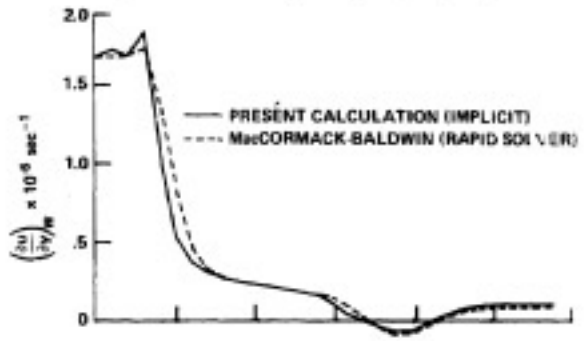
The computational mesh contained 32×45 mesh points (I = 32, J = 45) with uniform mesh increments in the x-coordinate ($\Delta x \approx 0.01$ ft) and an exponentially stretched mesh in the y-coordinate for j = 1 to 33 and uniform mesh spacing for j = 34 to 45. The mesh increments in the y-coordinate varied from Δy_{min} 0.00010 ft at the plate to $\Delta y_{max} = 0.00639$ ft at j = 33 as determined by the formulae

$$\Delta y_j = \Delta y_{max}\left(\frac{\Delta y_{min}}{\Delta y_{max}}\right)^{(33-j)/32} \qquad 1 \le j \le 33$$

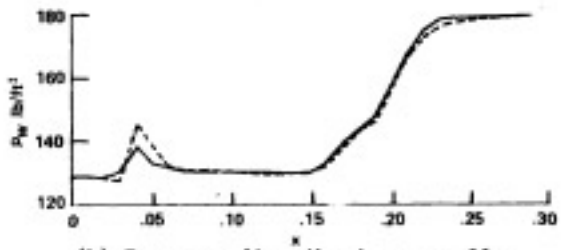
$$\Delta y_j = \Delta y_{max} \qquad 33 < j \le J-1$$

In general, the selection of the grid spacing depends on the Reynolds number and the Mach number.[9]

No temporal results were available for comparison; therefore, the steady-state wall-shear velocity and pressure distributions are compared (Fig. 5) with those of MacCormack and Baldwin[10] who used the rapid-solver method of MacCormack.[9] Similar agreement was obtained in comparisons of velocity profiles through the boundary layer (Fig. 6).



(a) Velocity gradient at wall.



(b) Pressure distribution at wall.

Fig. 5 Comparison of computational results for shock-boundary-layer interaction problem by implicit (present calculation) and rapid solver (MacCormack[9] and MacCormack and Baldwin[10]) methods. M = 2.0, $Re_{x_{SHK}} = 0.296\times10^6$, $x_{SHK} = 0.16$ ft, 32×45 mesh points.

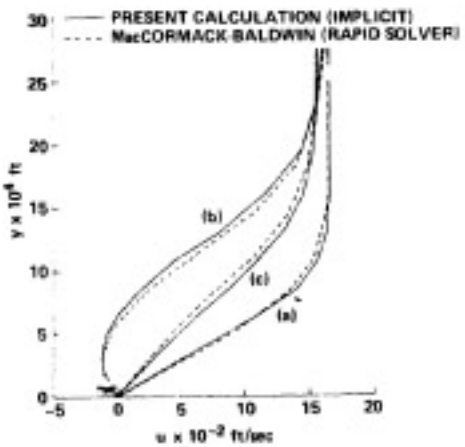


Fig. 6 Velocity profiles through the boundary layer for shock-boundary-layer interaction: (a) upstream of separation; (b) at maximum separation; and (c) downstream of separation.

Computational Effort

All calculations were done on a CDC 7600 with an FTN 4.5 level 414 compiler and required only small core storage (for grids up to 64×50). Based on the shock-boundary-layer calculations, the computational time (CP seconds) per mesh point per time step, that is,

$$\tau = \frac{\text{CP seconds}}{I \times J \times \text{number of time steps}}$$

was $\tau_{NS} = 4.6\times10^{-4}$ sec which compares favorably with the calculation time for the inviscid Eulerian equations[4] $\tau_E = 3.2\times10^{-4}$ sec. The calculations for the shock-boundary-layer equations required less than 100 time steps to reach a steady state and the maximum Courant number was approximately 170.

VII. Concluding Remarks

The distinguishing features of the second-order method described herein include the retention of the conservation-law form, a direct derivation of the basic scheme, the simplicity of the computational algorithm, the use of generalized time differencing, the "delta" formulation, and the second-order treatment of the mixed spatial derivatives.

The implicit algorithm and numerical examples considered in this paper are limited to a Cartesian coordinate system with a uniform mesh or an exponentially expanded mesh in one direction. However, under an arbitrary time-dependent coordinate transformation, Eq. (1) retains the same conservation form.[11] Extensions of the (Cartesian) implicit scheme described above have recently been made for both inviscid[12,13] and viscous[12] compressible flows with arbitrary (two-dimensional) body geometries.

APPENDIX

The vector of conserved variables, U, and flux vectors of Eq. (1) are

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} = \begin{bmatrix} \rho \\ m \\ n \\ e \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix} = \begin{bmatrix} m \\ (m^2/\rho) + p \\ mn/\rho \\ (e+p)m/\rho \end{bmatrix}, \quad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix} = \begin{bmatrix} n \\ mn/\rho \\ (n^2/\rho) + p \\ (e+p)n/\rho \end{bmatrix} \qquad \begin{matrix} (A.1) \\ (A.2) \\ (A.3) \end{matrix}$$

$$V_1 + V_2 = \begin{bmatrix} 0 \\ \lambda(u_x + v_y) + 2\mu u_x \\ \mu(v_x + u_y) \\ \mu v(u_y + v_x) + \lambda u(u_x + v_y) + 2\mu u u_x + kT_x \end{bmatrix} \qquad (A.4)$$

$$W_1 + W_2 = \begin{bmatrix} 0 \\ \mu(u_y + v_x) \\ \lambda(u_x + v_y) + 2\mu v_y \\ \mu u(v_x + u_y) + \lambda v(u_x + v_y) + 2\mu v v_y + kT_y \end{bmatrix} \qquad (A.5)$$

where $u_x = \partial u/\partial x$, etc. The primitive variables are density, ρ; velocity components, u and v; pressure, p; and total energy per unit volume e. In addition

$$m = \rho u, \qquad n = \rho v \qquad (A.6)$$

k is the coefficient of heat conductivity, λ and μ are the viscous coefficients, which are in general functions of the temperature, T

$$T = \frac{1}{\rho c_v}\left[e - \frac{1}{2}(\rho u^2 + \rho v^2)\right] \qquad (A.7)$$

where c_v is the specific heat at constant volume. The pressure, p, is given by the equation of state

$$p = (\gamma - 1)\left[e - \frac{1}{2}(\rho u^2 + \rho v^2)\right] \qquad (A.8)$$

where γ is the ratio of specific heats.

The vectors V_1, V_2, W_1, and W_2 can be rewritten in terms of the conservative variables as

$$V_1 = \begin{bmatrix} 0 \\ (\lambda + 2\mu)\rho^{-2}(\rho m_x - m\rho_x) \\ \mu\rho^{-2}(\rho n_x - n\rho_x) \\ (\lambda + 2\mu - k/c_v)m\rho^{-3}(\rho m_x - m\rho_x) + (\mu - k/c_v)n\rho^{-3}(\rho n_x - n\rho_x) + (k/c_v)\rho^{-2}(\rho e_x - e\rho_x) \end{bmatrix} \tag{A.9}$$

$$V_2 = \begin{bmatrix} 0 \\ \lambda\rho^{-2}(\rho n_y - n\rho_y) \\ \mu\rho^{-2}(\rho m_y - m\rho_y) \\ \lambda m\rho^{-3}(\rho n_y - n\rho_y) + \mu n\rho^{-3}(\rho m_y - m\rho_y) \end{bmatrix} \tag{A.10}$$

$$W_1 = \begin{bmatrix} 0 \\ \mu\rho^{-2}(\rho n_x - n\rho_x) \\ \lambda\rho^{-2}(\rho m_x - m\rho_x) \\ \lambda n\rho^{-3}(\rho m_x - m\rho_x) + \mu m\rho^{-3}(\rho n_x - n\rho_x) \end{bmatrix} \tag{A.11}$$

$$W_2 = \begin{bmatrix} 0 \\ \mu\rho^{-2}(\rho m_y - m\rho_y) \\ (\lambda + 2\mu)\rho^{-2}(\rho n_y - n\rho_y) \\ (\lambda + 2\mu - k/c_v)n\rho^{-3}(\rho n_y - n\rho_y) + (\mu - k/c_v)m\rho^{-3}(\rho m_y - m\rho_y) + (k/c_v)\rho^{-2}(\rho e_y - e\rho_y) \end{bmatrix} \tag{A.12}$$

Although expressions (A.9) - (A.12) are more complex than (A.4) and (A.5), they are used only in the analytical evaluation of the Jacobians and are not computed numerically.

The Jacobian matrices A, B, R, and S (Eqs. (4)) are

$$A = -\begin{bmatrix} 0 & -1 & 0 & 0 \\ \frac{3-\gamma}{2}u^2 + \frac{1-\gamma}{2}v^2 & (\gamma - 3)u & (\gamma - 1)v & 1 - \gamma \\ uv & -v & -u & 0 \\ \frac{\gamma e u}{\rho} + (1-\gamma)u(u^2 + v^2) & -\frac{\gamma e}{\rho} + \frac{\gamma - 1}{2}(3u^2 + v^2) & (\gamma - 1)uv & -\gamma u \end{bmatrix} \tag{A.13}$$

$$B = -\begin{bmatrix} 0 & 0 & -1 & 0 \\ uv & -v & -u & 0 \\ \frac{3-\gamma}{2}v^2 + \frac{1-\gamma}{2}u^2 & (\gamma - 1)u & (\gamma - 3)v & 1 - \gamma \\ \frac{\gamma e v}{\rho} + (1-\gamma)v(u^2 + v^2) & (\gamma - 1)uv & -\frac{\gamma e}{\rho} + \frac{\gamma - 1}{2}(3v^2 + u^2) & -\gamma v \end{bmatrix} \tag{A.14}$$

$$R = \rho^{-1}\begin{bmatrix} 0 & 0 & 0 & 0 \\ -(\lambda + 2\mu)u & (\lambda + 2\mu) & 0 & 0 \\ -\mu v & 0 & \mu & 0 \\ -(\lambda + 2\mu - k/c_v)u^2 - (\mu - k/c_v)v^2 - (k/c_v)(e/\rho) & (\lambda + 2\mu - k/c_v)u & (\mu - k/c_v)v & k/c_v \end{bmatrix} \tag{A.15}$$

$$S = \rho^{-1}\left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -\mu u & \mu & 0 & 0 \\ -(\lambda + 2\mu)v & 0 & (\lambda + 2\mu) & 0 \\ -(\lambda + 2\mu - k/c_v)v^2 - (\mu - k/c_v)u^2 - (k/c_v)(e/\rho) & (\mu - k/c_v)u & (\lambda + 2\mu - k/c_v)v & k/c_v \end{array}\right] \quad (A.16)$$

and, if we neglect the dependence of λ and μ on t (see Section II), the Jacobian's P and Q combined with R_x and S_y, Eq. (9), are

$$-P + R_x = \rho^{-1}\left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -u(\lambda + 2\mu)_x & (\lambda + 2\mu)_x & 0 & 0 \\ -v\mu_x & 0 & \mu_x & 0 \\ -u^2(\lambda + 2\mu)_x - v^2\mu_x & u(\lambda + 2\mu)_x & v\mu_x & 0 \end{array}\right] \quad (A.17)$$

and

$$-Q + S_y = \rho^{-1}\left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -u\mu_y & \mu_y & 0 & 0 \\ -v(\lambda + 2\mu)_y & 0 & (\lambda + 2\mu)_y & 0 \\ -v^2(\lambda + 2\mu)_y - u^2\mu_y & u\mu_y & v(\lambda + 2\mu)_y & 0 \end{array}\right] \quad (A.18)$$

References

[1]Peyret, R. and Viviand, H., "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations," AGARD-AG-212, 1975.

[2]Lindemuth, I. and Killeen, J., "Alternating Direction Implicit Techniques for Two-Dimensional Magnetohydrodynamic Calculations," Journal of Computational Physics, Vol. 13, 1973, pp. 181-208.

[3]McDonald, H. and Briley, W. R., "Three-Dimensional Supersonic Flow of a Viscous or Inviscid Flow," Journal of Computational Physics, Vol. 19, 1975, pp. 150-178.

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

[5]Warming, R. F. and Beam, R. M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," SIAM-AMS Proceedings (to appear).

[6]Richtmyer, R. D. and Morton, K. W., Difference Methods for Initial-Value Problems, John Wiley and Sons, New York, 2nd Edition, 1967.

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

[8]Schlichting, H., Boundary Layer Theory, Pergamon Press, New York, 1955.

[9]MacCormack, R., "A Rapid Solver for Hyperbolic Systems of Equations," Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 59, Springer-Verlag, Berlin, 1976, pp. 307-317.

[10]MacCormack, R. and Baldwin, B., private communication.

[11]Viviand, H., "Formes Conservatives des Equations de la Dynamique des Gaz," La Recherche Aérosp., No. 1974-1, 1974, pp. 65-66.

[12]Steger, J. L., "Implicit Finite Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665. Alburquerque, N.M., 1977.

[13]Kutler, P., Pulliam, T., and Steger, J. L., "A Generalized Implicit Unsteady Euler Equation Solver Applied to Supersonic Flows with Imbedded Subsonic Regions," in preparation.