JOURNAL OF COMPUTATIONAL PHYSICS 48, 302–311 (1982)

# Note

## Boundary Treatments for Implicit Solutions to Euler and Navier–Stokes Equations

### Introduction

Implicit time marching schemes like those of Beam and Warning [1], Briley and MacDonald [2], and MacCormack (1980) [3] generally have not been as robust as would be expected from a stability analysis for the pure initial value problem. Recently, Yee *et al.* [4] illustrated that a more general stability analysis, which includes the effect of boundary conditions, may explain some of the seemingly anomalous behavior of these schemes. The major theoretical basis for this type of modal stability was established in a series of papers by Kreiss [5, 6], Osher [7, 8], and Gustafsson *et al.* [9].

Yee as well as Gustafsson and Oliger [10] considered the effect of inflow–outflow boundary condition formulations on the stability of a class of numerical schemes to solve the Euler equations in one space dimension. The characteristic feature of a subsonic inflow–outflow boundary is that a priori boundary values may be specified for only some problem variables, while remaining boundary values must be determined as part of the solution process. Yee demonstrated a rather large disparity in stability bounds between the use of explicit or implicit extrapolation procedures and in general demonstrated that implicit extrapolation procedures had the least restrictive stability bounds. The intent here is to explore computationally the implication of this work for several two-dimensional Euler and Navier–Stokes simulations.

### Numerical Procedures

The two-dimensional Navier–Stokes equations may be written in vector form as [1]

$$\frac{\partial U}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}. \tag{1}$$

The strong conservation law form may be retained under a general coordinate mapping as illustrated in Viviand [11]. All computations to be described were conducted in a mapped computational domain but for simplicity numerical and



0021-9991/82/110302-10$02.00/0

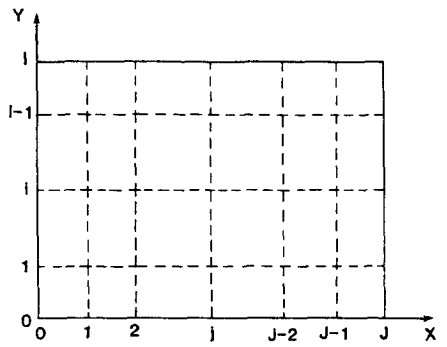


FIG. 1. Grid numbering schene for boundary condition formulation.

boundary condition procedures will be described in the simple two-dimensional geometry shown in Fig. 1.

A 1979 paper by Beam and Warning [12] outlined a solution scheme for systems of equations of form (1) which included most numerical schemes for which the modal boundary condition analysis has been conducted. This scheme uses the well-developed methods for ordinary differential equations as a guide to developing numerical methods for partial differential equations. The scheme presented combines linear multistep methods, local linearization, approximate factorization, and one leg methods. The shceme, a generalization of the scheme presented in [1], solves for a variable $\rho(E)\, u$ which is equivalent to $\Delta u^n$ in the class of schemes represented by the earlier paper. The earlier scheme is somewhat easier to understand as $\Delta u^n$ is just the change in the solution from time level $n$ to level $n+1$, while $\rho(E)\, u$ is a more general time differencing formula.

The solution schemes chosen are implemented as

$$(I + L_y^n)\, \Delta u^* = RHS^n, \tag{2}$$

$$(I + L_x^n)\, \Delta U^n = \Delta U^*, \tag{3}$$

$$U^{n+1} = U^n + \Delta U^n, \tag{4}$$

where $RHS^n$ is very nearly the finite difference approximation to the steady state equations, and $L_x$ and $L_y$ are linearized difference operators representing a particular time and spatial differencing scheme.

Full details of these operators are contained in [1]. If the spatial differencing is taken to be centered, the computational form of either Eq. (2) or (3) appears at each interior point as

$$A_i^n \Delta U_{i-1}^n + B_i^n \Delta U_i^n + C_i^n \Delta U_{i+1}^n = D_i^n, \tag{5}$$

where $A_i$, $B_i$, and $C_i$ are $4 \times 4$ matrices known at time level $n$, $D_i$ is the right-hand side vector at node point $i$ known at time level $n$, and $\Delta U_i^n$ is the unknown vector at

node point $i$. The boundary points will be assumed to involve only the nearest two points in the $x$ direction.

$$A_0^n \Delta U_0 + B_0^n \Delta U_1 + C_0^n \Delta U_2 = D_0^n. \tag{6}$$

The restriction to extrapolation along grid lines (actually transformed grid lines), is necessary to maintain the block tridiagonal form and avoids possible instabilities due to skewed extrapolation, see [13].

The full matrix equation will reduce to tridiagonal form if the first and $n$th equations are substituted into the second and the $(n-i)$th equations, for example,

$$B_1' = B_1 - A_1 A_0^{-1} B_0. \tag{7}$$

## Boundary Treatments

### *Inflow–outflow Boundary*

The finite difference algorithms studied usually require more boundary values than are required for the partial differential equations which they simulate. These extra numerical boundary conditions cannot be set arbitrarily and are usually determined through an extrapolation procedure. These extrapolation procedures may either be explicit, that is boundary values needed at a new time are determined uniquely from the old time level solution, or implicit, that is, the boundary values are determined as part of the new time level solution. The analytical boundary conditions or the extrapolation quantities are usually not conservation variables but primitive variables, and a local linearization is usually required as part of defining the extrapolation procedure.

Consider, for example, an implicit subsonic outflow boundary at which the local static pressure is specified as a boundary condition and all other variables are to be determined by extrapolation. Figure 1 shows a typical computational grid and defines the subscripts used.

$$P_{i,j}^{n+1} = P_{i,j}^n \qquad \text{given,} \tag{8}$$

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{i,j}^{n+1} = 2 \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{i,j-1}^{n+1} - \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{i,j-2}^{n+1} \qquad \text{implicit space extrapolation.} \tag{9}$$

In order to complete the boundary formulation, all equations must be expressed in delta form and in terms of conservation variables. For the total internal energy this may be done through its definition

$$E_t = P/(\gamma - 1) + \tfrac{1}{2}(\rho u)^2/\rho + (v)^2/\rho. \tag{10}$$

Since the relations between conservation variables are nonlinear, some linearization

step will be necessary before the boundary condition formulation may be used. We choose to introduce our linearization step here as

$$\Delta E_t = (E_t^{n+1} - E_t^n) = (1/(\gamma-1))\,\Delta P - \tfrac{1}{2}(u^2+v^2)^n\,\Delta\rho + u^n\Delta(\rho u) + v^n\Delta(\rho v) + (\Delta u\Delta v, \Delta u^2, \Delta v^2, \Delta\rho\Delta u, \Delta\rho\Delta v). \tag{11}$$

If terms of order $\Delta u\Delta v$ are neglected, the error is equivalent to the linearization error of the interior point scheme. We may express the transformation from boundary variables to conservation variables as

$$U_{i,j} = \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta E_t \end{pmatrix}_{i,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2}(u^2+v^2)^n & u^n & v^n & 1/(\gamma-1) \end{pmatrix} \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta P \end{pmatrix} = N_{i,j}\Delta W_{i,j} \tag{12}$$

we shall in general denote transformation from conservative to primitive variables as

$$\Delta W_{i,j} = T_{i,j}\Delta U_{i,j}. \tag{13}$$

The extrapolation conditions for $W_{i,j}$ are

$$\Delta W_{i,J} = \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta P \end{pmatrix}_{i,J} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta P \end{pmatrix}_{i,J-1} + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta P \end{pmatrix}_{i,J-2} \tag{14}$$

or

$$\Delta W_{i,J} = P_{J-1}W_{i,j-1} + P_{J-2}W_{i,J-2}. \tag{15}$$

The final equations relating the boundary conservation variables and the interior conservation variables are

$$\Delta U_{i,J} = N_{i,J}^n(P_{J-1}T_{i,J-1}^n\Delta U_{i,J-1} + P_{J-2}T_{i,J-1}^n\Delta U_{i,J-2}) \tag{16}$$

or

$$\Delta U_{i,J}G_{i,J-1}^n\Delta U_{i,J-1} + H_{i,J-2}^n\Delta U_{i,J-2}. \tag{17}$$

With the definition of $P_{J-1}$ and $P_{J-2}$ given in Eq. (15), $T_{i,J-1}$ and $T_{i,J-2}$ are identity matrices.

An explicit outflow boundary treatment was constructed using

$$P^{n+1} = P^n, \qquad \text{given},$$

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{i,J}^{n+1} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{i,J-1}^{n}, \tag{18}$$

and setting $G_{i,J-1} = H_{i,J-2} = 0$.

In forming Eq. (9), we choose to extrapolate the local momentum flux rather than a specific primitive or characteristic variable; choice of other extrapolation variables would alter only the transformation matrix $T_{i,j}$. Extrapolation of the momentum flux is somewhat arbitrary, but its choice did not affect the accuracy of the computational results to be presented.

*Solid Wall Boundary Procedures*

The boundary treatment procedure illustrated for inflow–outflow boundary are easily extended to cover solid walls in either inviscid or viscous flow situations. Here

$$\Delta U_{0,J} = \begin{pmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta E_t \end{pmatrix} = \left(\begin{array}{ccccc} \gamma/T & \vdots & \rho/T & \vdots & 0 \\ \gamma u/T & \vdots & \rho u/T & \vdots & 0 \\ \gamma v/T & \vdots & \rho v/T & \vdots & 0 \\ \dfrac{1}{\gamma-1}+\dfrac{1}{2}\dfrac{\gamma q^2}{T} & \vdots & -\dfrac{1}{2}\dfrac{\rho q}{T} & \vdots & \rho q \end{array}\right.$$

$$\left.\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \end{array}\begin{array}{c} 0 \\ 0 \\ Sq \\ 0 \end{array}\right) \begin{pmatrix} \Delta P \\ \Delta T \\ \Delta q \\ \Delta u \end{pmatrix} \tag{19}$$

or

$$\Delta U_{0,j} = N_{0,j}^n \Delta W_{0,j}, \tag{20}$$

where $q$ is the velocity parallel to the wall and $S$ is the wall slope. For the inviscid flow examples $\partial P/\partial y$, $\partial T/\partial y$, and $\partial q/\partial y$ are set equal to zero, while, for the viscous flow examples $v$, $u$, and $\partial T/\partial Y$ are set equal to zero and $\partial P/\partial y$ is equal to $4/3\mu(\partial^2/\partial y^2)(v)$. All derivatives are evaluated by one-sided finite difference formulas.

As indicated by Buggeln *et al.* [14], an ADI type procedure requires boundary conditions for the intermediate step. Usually the intermediate step was in the $y$ direction and the boundary conditions were applied as if the intermediate results were physical quantities, that is, the boundary conditions of Eq. (19) were applied to the quantities $\Delta U^*$ of Eq. (2).

Explicit wall boundary treatments are generated by applying the primitive variable form of Eq. (19) and forcing the correction matrices to be zero.

## Numerical Results

Three geometries were selected for detailed study: an inviscid supersonic diffuser with weak oblique shock, supersonic in/supersonic out; an inviscid supersonic

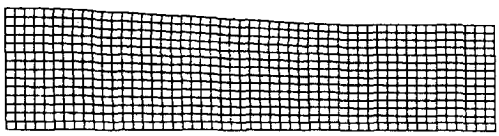

FIG. 2. Computational grid for weak shock diffuser calculations.

diffuser with a strong normal shock, supersonic in/subsonic out, and a viscous supersonic diffuser with weak oblique shock illustrating a shock-boundary layer interaction. Sketches of the geometries are shown in Figs. 2–4. Solutions for each geometry were run to steady state for a range of time step sizes. For convenience, time step sizes are reported in terms of $x$ and $y$ CFL numbers

$$(\mathrm{CFL}_x) = \max(\Delta t(u + c)_{i,j}/\Delta x_{i,j}), \tag{21}$$

$$(\mathrm{CFL})_y = \max(\Delta t(v + c)_{i,j}/\Delta y_{i,j}). \tag{22}$$

The time step size was uniform over each calculation which results in nonuniform $\mathrm{CFL}_x$ and $\mathrm{CFL}_y$ numbers. The maximum value of each is reported. Sample convergence history plots are shown in Fig. 5 which shows the log of the value of the point maximum steady state residual

$$\mathrm{SSR} = \partial E/\partial x - \partial R/\partial x + \partial F/\partial y - \partial S/\partial y \tag{23}$$

plotted against the iteration number. A solution was not termed stable unless the residual converged to the machine accuracy, about $1 \times 10^{-6}$. All calculations used a 32 bit floating point word size.

Each geometry calculation was run with fully explicit extrapolations, $\Delta u = 0$, and with fully implicit extrapolations; the results are summarized in Table I. The most interesting of these results are shown in Fig. 5. At a tipe step size corresponding to a $\mathrm{CFL}_x$ number of 15, convergence was rapid and very nearly monotonic in time. At smaller time step sizes, the convergence was slower but nearly monotonic. At a $\mathrm{CFL}_x$ of 45, convergence rates initially appeared to be faster than for a $\mathrm{CFL}_x$ of 15, but the final residual values oscillated sugnificantly about its minimum value. At a $\mathrm{CFL}_x$ of 90, the convergence rate was substantially slower than at a $\mathrm{CFL}_x$ of 15, and at larger $\mathrm{CFL}_x$ values the solution diverged.

The results for the strong shock diffuser can reasonably be compared to those of Yee *et al.* [4]. They reported a CFL number stability limit between 10 and 20, while

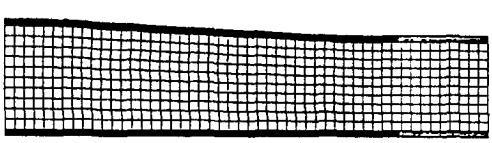

FIG. 3. Computational grid for shock-boundary layer calculations.

TABLE I

| | Inviscid weak oblique shock diffuser | | | Inviscid strong shock diffuser | | | Shock boundary layer | | |
|---|---|---|---|---|---|---|---|---|---|
| | $CFL_x$ | $CFL_y$ | Iterations to convergence | $CFL_x$ | $CFL_y$ | Iterations to convergence | $CFL_x$ | $CFL_y$ | Iterations to convergence |
| Explicit boundary treatments | 1.5 | 0.6 | 225 | 1.5 | 0.6 | 3000 | 1.5 | 125 | 1300 |
| | 6.0 | 2.4 | 130 | 6.0 | 2.4 | 1200 | 6.0 | 500 | unstable |
| | 15.0 | 6.0 | 225 | 15.0 | 6.0 | 800 | 15.0 | 1250 | unstable |
| | 45.0 | 18.0 | 500 | 45.0 | 18.0 | 800 | 45.0 | 3750 | unstable |
| | 90.0 | 36.0 | 1000 | 90.0 | 36.0 | 1000 | 90.0 | 7500 | unstable |
| | 150.0 | 60.0 | 3000 | 150.0 | 60.0 | 3000 | 150.0 | 12500 | unstable |
| Implicit boundary treatments | 1.5 | 0.6 | 225 | 1.5 | 0.6 | 3000 | 1.5 | 125 | 1300 |
| | 6.0 | 2.4 | 130 | 6.0 | 2.4 | 950 | 6.0 | 500 | 400 |
| | 15.0 | 6.0 | 225 | 15.0 | 6.0 | 400 | 15.0 | 1250 | unstable |
| | 45.0 | 18.0 | 500 | 45.0 | 18.0 | 1000 | 45.0 | 3750 | unstable |
| | 90.0 | 36.0 | 1000 | 90.0 | 36.0 | 1500 | 90.0 | 7500 | unstable |
| | 150.0 | 60.0 | unstable | 150.0 | 60.0 | unstable | 150.0 | 12500 | unstable |
| | 15.0 | 1250 | 250 | | | | 15.0 | 600 | unstable |

*Note.* Backward Euler time differencing was used for all reported calculations.

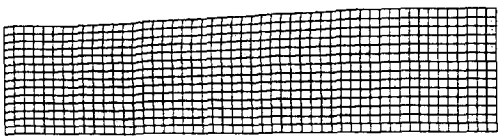

FIG. 4. Computational grid for strong shock calculations.

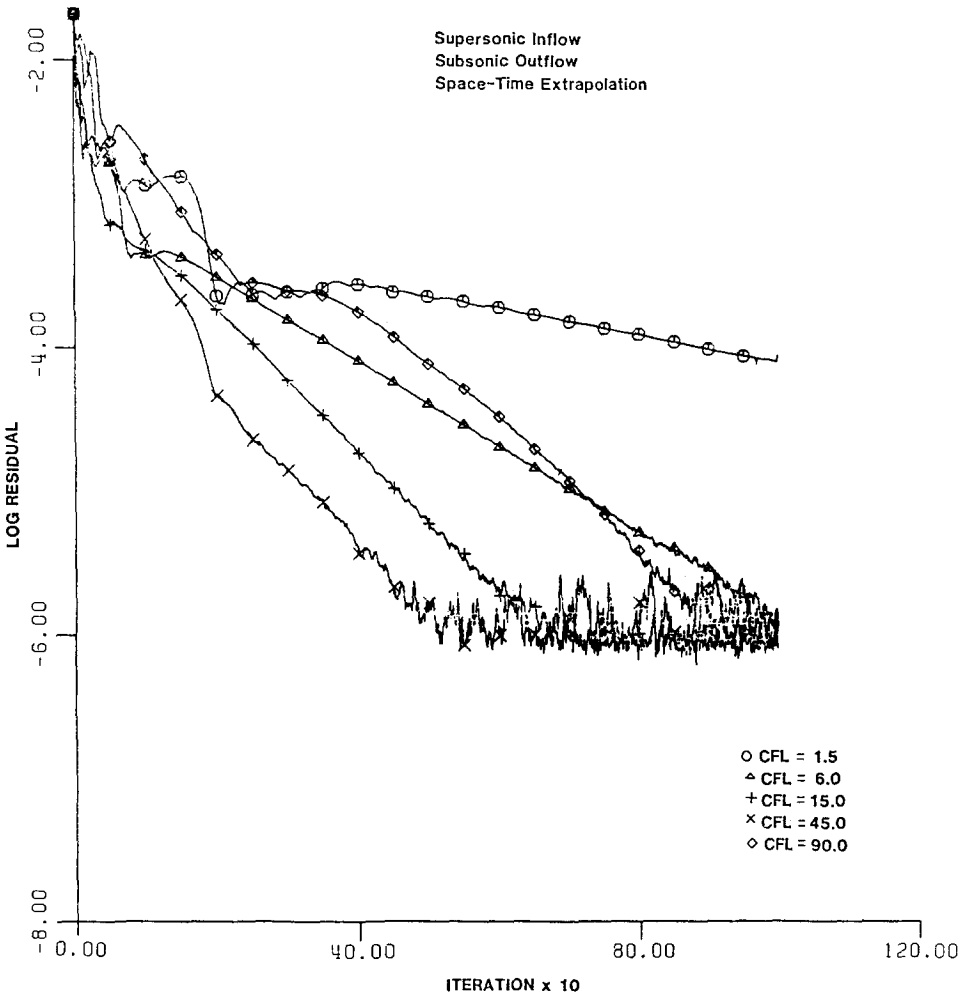


FIG. 5. Convergence history for strong shock diffuser calculation.

we found stability limits between 90 and 150. Thus the analysis in one space dimension does appear to provide a sufficient condition for stability, but it may not provide a close approximation to the stability limit. It is essential, however, to emphasize that the largest convergence rates were observed at time steps corresponding to CFL numbers of order 10 and that only a marginal computational time advantage for the implicit boundary formulations was observed.

The results for the shock-boundary layer calculation are very interesting but they demonstrate a substantial computational advantage for the implicit solid wall conditions, not for the inflow–outflow extrapolation. Here the stability boundary and the best convergence rates were observed at time step sizes corresponding to $CFL_x$ numbers of 5 to 10. When using the implicit wall conditions, the algorithm stability appeared to be independent of grid spacing in the normal direction as might be hoped. When using the explicit wall condition, the algorithm stability was limited to a $CFL_y$ number of about 500.

## Conclusions and Discussion

While it is difficult to generalize from only a few test examples, it is apparent that a better appreciation of the role that boundary treatments play in implicit algorithms has allowed the development of far more robust Beam and Warming type solvers. For both explicit and implicit boundary treatments, we were able to compute solutions accurately with time steps 50 to 100 times larger than explicit time limits while retaining the ability to choose rather arbitrary initial conditions. In many cases, our limiting time steps for the two-dimensional test problems were in fact larger than the limit which a one-dimensional analysis would suggest.

The most important computational result we observed was that while an improved appreciation of boundary treatments did allow very large time step sizes to be used, the largest convergence rates to steady state were observed at relatively small time step sizes. For the two-dimensional test problems, the best $CFL_x$ numbers were of order 10, not of order 100. One-dimensional test examples showed no such convergence rate behavior. Presently unpublished analysis by Abarbanel *et al.* [15] has linked this behavior to the approximate factorization form of Eqs. (2) and (3). This effect now seems to be setting the time step sizes for our viscous flow computations and new work should focus on methods for overcoming this limitation.

## References

1. R. M. Beam and R. F. Warming, *AIAA J.* **16** (4) (1978), 393.
2. W. R. Briley and H. McDonald, *J. Comput. Phys.* **19** (1975), 150.
3. R. W. MacCormack, "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper 81–0110, 1980.
4. H. C. Yee, R. M. Beam, and R. M. Warming, *in* "Proc. of AIAA Computational Fluid Dynamics Conference," Paper No. 81–1009, June 1981.

5. H. O. KREISS, *Math. Comp.* **22** (1968), 703.
6. H. O. KREISS, *Proc. Roy. Soc. London Ser. A* **323** (1971), 225.
7. S. OSHER, *Trans. Amer. Math. Soc.* **137** (1969), 177.
8. S. OSHER, *Math. Comp.* **23** (1969), 335.
9. B. GUSTAFSSON, H. O. KREIS, AND A. SUNDSTRÖM, *Math. Comp.* **26** (1972), 649.
10. B. GUSTAFSSON AND J. OLIGER, "Stable Boundary Approximations for a Class of Time Discretizations of $U_t = AD_0U$," Report No. 87, Dept. of Computer Science, Upsala Univ. Sweden, September 1981.
11. H. VIVIAND, *Rech. Aerospat.* **1** (1974), 65.
12. R. M. BEAM AND R. F. WARMING, "An Implicit Factor Scheme for the Compressible Navier–Stokes Equations, II: The Numerical ODE Connection," AIAA Paper 79–1446, Williamsburg, Va., 1979.
13. S. S. ABARBANEL AND E. M. MURMAN, "Stability of Two-Dimensional Hyperbolic Initial Boundary Value Problems for Explicit and Implicit Schemes," Symposium on Numerical Boundary Conditions, NASA Ames Research Center, October 1981.
14. R. C. BUGGELN, W. R. BRIELY, AND H. MCDONALD, *in* "Proceedings of AIAA Computational Fluid Dynamics Conference," Paper No. 81–1023, June 1981.
15. S. S. ABARBANEL, D. L. DWOYER, AND D. GOTTLIEB, private communication, July 1981.

RECEIVED: January 6, 1982; REVISED: June 15, 1982

W. T. THOMPKINS, JR. AND R. H. BUSH

*Department of Aeronautics and Astronautics,*
*Massachusetts Institute of Technology,*
*Cambridge, Massachusetts 02139*