Boundary Conditions for the Method of Lines Applied to Hyperbolic Systems*

by

John Gary
Department of Computer Science
University of Colorado
Boulder, Colorado 80302

TR #CU-CS-073-75 July 1975

* This research was supported by an ARPA grant AFOS-74-2732.

1. Introduction. A fundamental problem in the numerical solution of hyperbolic equations is the proper approximation of the boundary conditions. For example, the leapfrog scheme applied to the following hyperbolic problem is unstable.

$$
\begin{aligned}
&u_t+u_x=0 && 0\leq x\leq \pi \\
&u(0,t)=\sin(-t) && 0\leq t \\
&u(x,0)=\sin(x) &&
\end{aligned}
\qquad (1)
$$

The mesh for this scheme is $x_j=j\Delta x=j\pi/J$ for $0\leq j\leq J$. The exact solution is sin (x-t). If a second order difference approximation for the spatial derivative is combined with a leapfrog scheme for time, then the following scheme is obtained

$$
\begin{aligned}
&u_0^n=\sin(-t_n) \\
&u_J^{n+1}=u_J^{n-1}-2\Delta t(u_J^n-u_{J-1}^n)/\Delta x \\
&u_j^{n+1}=u_j^{n-1}-\Delta t(u_{j+1}^n-u_{j-1}^n)/\Delta x \quad 1\leq j<J.
\end{aligned}
$$

This scheme is unstable. If the outflow boundary is modified as indicated below, then the scheme is stable and has second order accuracy.

$$u_J^{n+1}=u_J^n-\Delta t(u_J^n-u_{J-1}^n)/\Delta x$$

When the method of lines is used for the simple linear hyperbolic equation (1) with periodic boundary equations, then the resulting difference scheme is stable, provided an ODE solver with automatic step-size adjustment is used to solve the system of ordinary differential equations. Even if the ODE solver uses an Euler forward time-step scheme, the integration will converge as the mesh spacing is taken to zero, since the ODE solver will take the step time-step small enough as a function of the mesh size to guarantee convergence. It will not be the case that

$\Delta t = O(\Delta x)$. Note that the semi-discrete approximation produced by the method of lines with periodic boundary conditions can be written in the form

$$\dot{\underline{u}} = A\underline{u} \qquad \underline{u} = (u_{-J}, \ldots, u_{J-1}) \qquad x_j = j\pi/J$$

where the matrix A is

$$A = -\frac{1}{2\Delta x} \begin{vmatrix} 0 & 1 & 0 & . & . & & & -1 \\ -1 & 0 & 1 & 0 & . & . & & 0 \\ 0 & -1 & 0 & 1 & . & . & & 0 \\ \vdots & & & & & & & \\ 1 & & & & & 0 & -1 & 0 \end{vmatrix}$$

The solution of this differential equation is given in terms of an exponential matrix as

$$\underline{u}(t) = \underline{u}(0)e^{At}$$

Since the matrix is skew symmetric and cyclic its eigenvalues are pure imaginary and its eigenvectors are orthogonal. Therefore, the solution is bounded independently of the number of mesh points. This implies that the solution of this semi-discrete approximation will converge to the solution of the original equation (1). Therefore any spatial discretization which yields a skew symmetric, cyclic matrix will define a convergent method of lines approximation. Stability in a finite difference scheme for hyperbolic problems is in a sense associated with the temporal discretization.

Unfortunately, the method of lines does not necessarily produce a stable scheme when the boundary conditions are not periodic. However, the method of lines does seem to be more likely to yield a stable scheme than a leapfrog time discretization.

Our purpose is an experimental study of some boundary difference approximations for use on hyperbolic systems where the method of lines is used for the temporal discretization. Our results will refer mainly to the Runge-Kutta-Fehlberg ODE solver, although we intend to experiment with the Adams method of Shampine [9] in the future. We have found it important to include test cases for hyperbolic systems (more than one independent variable) for which the characteristics lie on both sides of the boundary. This is in agreement with comments by Chu [3] and Sundstrom [10]. We are interested in boundary approximations which can be incorporated into a general PDE solver to treat hyperbolic systems in two dimensions. Such solvers for parabolic equations in one dimension are described by Sincovec and Madsen [11], Carver [2], Loeb [6], Bowen [1], Hastings [12] and others. Because of our interest in general hyperbolic systems, we cannot consider boundary approximations stated in terms of specific variables for specific equations. We can only consider algorithms which can be presented in a general gramework. We will test two such algorithms.

Of course, such a general algorithm requires the user to apply it in such a way as to produce a properly posed hyperbolic problem. We must allow the user the flexibility to set the boundary conditions. Eventually, we might be able to supply an optional check to see that the boundary conditions are consistent with the hyperbolic system.

2. Computational results indicating stability and accuracy of the method of lines. In this section we consider difference approximations for the system (1). These are semi-discrete approximations of the form

$$\underline{u}'=A\underline{u}+\underline{f} \tag{2}$$

where $\underline{u} = \underline{u}(t)=(\ldots u_j(t)\ldots)^T$ is a vector of mesh point values. In this section we will look at the eigenvalues of A and the norm of the exponential matrix

$$||e^A|| \tag{3}$$

for four finite difference approximations. If this norm is bounded independent of the spatial mesh, then the semi-discrete approximation is stable. This follows from the integral form of the solution of (2)

$$u(t) = u(o)e^{At} + \int_0^t f(\tau)e^{A(t-\tau)}d\tau \tag{4}$$

If the eigenvectors of A are orthogonal, then a bound for the norm of the exponential matrix can be obtained from the eigenvalues of A. Therefore, we compute these eigenvalues and also the norm (3) in order to gain insight into the stability of the following four schemes.

A. <u>An inconsistent scheme</u>. Here a one-sided difference is used at both boundaries in spite of the fact that the solution should be specified at the inflow or left boundary. This must yield an unstable approximation. The approximation is consistent, and if it were also stable, then it would be convergent. That is, if the norm of the exponential matrix

$$e^{At}$$

were bounded independently of the mesh spacing the approximation would be convergent, which is impossible since no boundary condition has been specified on the inflow boundary. The scheme is

$$u_0'(t)=-\frac{1}{\Delta x}(u_1(t)-u_0(t)) \tag{5}$$
$$u_j'(t)=-\frac{1}{2\Delta x}(u_{j+1}(t)-u_{j-1}(t)) \qquad 1\leq j<J$$
$$u_J'(t)=-\frac{1}{\Delta x}(u_J(t)-u_{J-1}(t))$$

B. A second order scheme. This scheme is the same as the previous one except

$$u_0(t)=\sin(-t)$$

It is only first order at the boundary, but the overall accuracy should be second order.

C. Fourth order with a third order boundary. This scheme is given below. Oliger [7] has shown that subtle changes are required in this spatial approximation when it is used with a leapfrog time discretization, in order that the resultant difference scheme be stable. However, it seems to be stable without modification when it is used with a variable step ODE solver.

$$u_0(t)=\sin(-t)$$

$$u_1'(t)=-(2u_0-3u_1+6u_3-u_4)/6\Delta x$$

$$u_j'(t)=-(2u_{j-2}-16u_{j-1}+16u_{j+1}-2u_{j+2})/(24\Delta x) \tag{6}$$

$$u_{J-1}'(t)=-(u_{J-3}-6u_{J-2}+3u_{J-1}+2u_J)/(6\Delta x)$$

$$u_J'(t)=-(-2u_{J-3}+9u_{J-2}-18u_{J-1}+11u_J)/(6\Delta x)$$

D. <u>A fourth order scheme with a fourth order boundary approximation</u>. This is the same scheme as the one above except that one-sided fourth order differences are used at the boundary.

$$u_1'(t)=-(-6u_0-20u_1+36u_2-12u_3+2u_4)/(24\Delta x)$$

$$u_{J-1}'(t)=-(-2u_{J-4}+12u_{J-3}-36u_{J-2}+20u_{J-1}+6u_J)/(24\Delta x) \qquad (7)$$

$$u_J'(t)=-(6u_{J-4}-32u_{J-3}+72u_{J-2}-96u_{J-1}+50u_J)/(24\Delta x)$$

The above four schemes can all be written in the matrix form of equation (2). The maximum of the real parts of the eigenvalues of the matrix A for these four schemes are given in Table I. The eigenvalues of A were determined by using the IMSL QR routine on the CDC 6400 at the University of Colorado. The exponential matrix was determined by summing its series expansion. The norm is that induced by the vector maximum norm. For the inconsistent scheme (A) the eigenvalues are all pure imaginary with a double or triple root at zero depending on whether J is even or odd. The instability of this scheme is evident from the norm of the exponential matrix but not from the eigenvalues.

Schemes (B) and (C) would appear to be stable from this analysis. However, we might expect the solution of scheme (D) to show exponential growth in time since its matrix has an eigenvalue with positive real part.

In order to provide a more complete test of these four schemes we wrote a code for these schemes applied to equation (1). This provides a direct test of the stability and accuracy of these schemes.

Table II shows the error obtained with the various schemes after integration to the indicated value of t=T using the mesh resolution determined by J. Note that the number of intervals per wave is 2(J-1) since the mesh runs from $x=0$ to $x=\pi$ and J+1 is the number of mesh points. Scheme (A) is clearly unstable. Schemes (B) and (C) seem to be stable which is consistent with the results in Table I giving the characteristics of the matrices corresponding to these schemes. Scheme (D) seems to be weakly unstable when the system is solved with the RKF ODE solver. However this scheme seems to be stable when the Runge-Kutta scheme with a fixed ratio $\Delta t/\Delta x$ is used.

3. <u>A variable coefficient problem</u>. A hyperbolic problem which is more typical of many applications than equation (1) is the following defined on the interval $0 \leq x \leq \pi$.

$$\begin{aligned} &u_t+\cos(t)u_x=\cos(x-t)(\cos(t)-1)=r(x,t) \\ &\text{If } \cos(t)\geq 0 \text{ then } u(0,t)=\sin(-t) \\ &\text{If } \cos(t)\leq 0 \text{ then } u(\pi,t)=\sin(\pi-t) \\ &u(x,0)=\sin x \end{aligned} \tag{8}$$

The solution of this problem is $u(x,t)=\sin(x-t)$. The mesh is $x_j=j\pi/J$, for $0\leq j\leq J$. In this problem the inflow and outflow boundary alternate between the two endpoints of the interval. When $\cos(t)\geq 0$ the left boundary is the inflow point. This makes the use of an ODE solver awkward if the method of equation (6) is used to define the system of differential equations. When $\cos(t)\geq 0$ the unknowns are $(u_1(t),\ldots,u_J(t))$ and when $\cos(t)\leq 0$ the unknown vector has shifted to $(u_0(t),\ldots,u_{J-1}(t))$.

Therefore we differentiate the boundary condition so that the system of differential equations always contains the same unknowns.

E. A second order scheme for equation (8).

If $\cos(t)\geq 0$ then

$$u_0'(t)=\frac{d}{dt}(\sin(-t))=-\cos(t)$$

otherwise

$$u_0'(t)=-(u_1-u_0)/\Delta x+r(0,t) \qquad (9)$$

If $\cos(t)\leq 0$ then

$$u_J'(t)=\frac{d}{dt}(\sin(\pi-t))=\cos(t)$$

otherwise

$$u_J'(t)=-(u_J-u_{J-1})/\Delta x+r(\pi,t)$$

This scheme uses a differentiated form of the boundary condition at an inflow boundary and a one-sided first order difference approximation to the differential equation at an outflow boundary. The definition of the differential equation used to define the solution along the boundary line varies depending on the inflow-outflow nature of the boundary. However, the solution along these boundary lines is always determined by a differential equation.

F. A fourth order scheme for equation (8).

If $\cos(t)\geq 0$ then

$$u_0'(t)=-\cos(t)$$

otherwise

$$u_0'(t)=-\delta_3'(\underline{u})_0+r(0,t)$$

Here δ_3' is the third order difference approximation of $u_x(0)$ using (x_0,x_1,x_2,x_3). The equation for $u_J'(t)$ is similar. The remainder of the system is identical with that of equation (6).

The results of using these schemes to approximate the solution of equation (8) is given in Table III. These results indicate that these schemes are stable. The norm of the second order approximation shows a slow linear growth with time. The error shows the expected asymptotic behavior with J (approximately). The behavior of this method on a more complex multidimensional problem awaits testing which we hope to carry out in the near future.

4. General boundary approximation algorithms. In this section we consider a program for the following, more general class of nonlinear hyperbolic equations.

$$\frac{\partial \underline{u}}{\partial t} = \frac{\partial}{\partial x}(\underline{g}(\underline{u},x,t))+\underline{h}(\underline{u},x,t) \tag{11}$$

or the nonconservation form

$$\frac{\partial \underline{u}}{\partial t} = f(\frac{\partial \underline{u}}{\partial x},\underline{u},x,t)+h(\underline{u},x,t) \tag{11}$$

Here $\underline{f}$, $\underline{g}$, and $\underline{h}$ are general vector valued functions and $\underline{u}(x,t)$ is the vector solution. We assume that boundary conditions are given at two end points x=a and x=b. We consider two methods to specify these boundary conditions.

The first method requires the specification of a subset of the unknowns at each boundary point. Consider the left boundary x=a. The unknowns are $(u_1(x,t),\ldots,u_M(x,t))$. The p unknowns $(u_{m_1},\ldots,u_{m_p})$ from the set $I = \{p_1,\ldots,p_m\}$ are specified as follows:

$$
\begin{aligned}
u_{m_1}(a,t) &= S_1(\underline{u}_{II}(a,t),t) \\
&\vdots \\
u_{m_p}(a,t) &= S_p(\underline{u}_{II}(a,t),t)
\end{aligned}
\tag{13}
$$

Here $\underline{u}_{II} = (u_{m_1},\ldots,u_{m_{M-p}})$ is the compliment of $u_I = (u_{m_1},\ldots,u_{m_p})$. The problem specification must include the integer p and the functions $S_1,\ldots,S_p$ at both boundary points. Note that p may depend on the time t. The functions S_i are used to set the values of u_I at the boundary. The values of u_{II} are computed from the hyperbolic equation, using one sided approximations for spatial derivatives.

For example, consider the variable coefficient problem given by equation (8). At the left boundary (x=0), if $\cos(t)\geq 0$, then for the number of boundary constraints we have p=1. The function $S_1(u_{II},t)= S_1(t)=-\sin(t)$. Note that u_{II} is empty in this case. If $\cos(t)<0$, then p=0 at the left boundary and u_I is empty. In this case the value of $U_0(t)$ (here $U_j(t)$ denotes the approximation to $u(x_j,t)$ on the "time line") is obtained from the differential equation

$$
\frac{dU_0}{dt} = -\cos(t)\delta_3(\underline{U})_0+r(x_0,t) \tag{14}
$$

where δ_3 represents the onesided difference approximation.

When the ODE solver, such as the Runge-Kutta-Fehlberg is used, there is a slight problem in implementing this algorithm. When the characteristic slope cos(t) changes sign, the nature of the system of ordinary differential equations changes. When $\cos(t)>0$, the unknowns in the system (8) are $(U_1,\ldots,U_j)$, but when $\cos(t)<0$ the unknowns are $(U_0,\ldots,U_{J-1})$.

The ODE solver always works with the full set of unknowns including the boundary values, that is $(U_0,\ldots,U_J)$. However, in computing the "right side" functions in the Runge-Kutta steps the boundary constraints are applied to set the boundary values for variables in the U_I sets. If the system of ODE's is written

$$\frac{dU_j}{dt} = \underline{F}_j(U_0,\ldots,U_J,t) \tag{15}$$

and $\underline{U}_0$ is in the constrained set U_I for $t = t_n+1/2\Delta t$, then the function $\underline{F}_j(U_0,\ldots U_j,t_n+1/2\Delta t)$ used in the Runge-Kutta step is replaced by $F_j(S(U_{II},t+1/2\Delta t),U_1,\ldots,U_J,t+1/2\Delta t)$. Also, at the end of the step the value of U_0 computed by the ODE solver is replaced by $S(U_0,t)$ provided U_0 is still in the constrained set U_I. Obviously this requires modification of the ODE solver. There is no guarantee that this method will converge. In fact, as we will see shortly, it does not always converge. The algorithm can be implemented as part of a PDE package once the user has supplied the subroutines to evaluate p, the sets U_I, and the functions $S_i(U_{II},t)$.

The second method is a generalization of the differentiated boundary conditions described in section 3 in equations (9) and (10). In this case the user is allowed to reset the time derivatives used by the ODE solver to compute the boundary values, that is

$$\frac{dU_{m,0}}{dt} = \hat{F}_{m,0}(\underline{U}_0,\underline{F}_0,t) \quad \text{and} \quad \frac{dU_{m,J}}{dt} = F_{m,J} = \hat{F}_{m,J}(\underline{U}_J,\underline{F}_J,t) \tag{16}$$

Here we assume a system of equations for the unknowns $u_{m,j}$ where there are M unknowns $(1\leq m\leq M)$ and the mesh points are $X_j(a=x_0<x_1< \ldots<x_J=b)$.

The vectors $\underline{F}_0 = (F_{1,0},\ldots,F_{M,0})^T$ and $\underline{F}_J$ are the time derivatives obtained using one sided difference approximations in the hyperbolic system at the boundary. The vectors $\underline{U}_0,\underline{F}_0$ and the time t are supplied to a user written subroutine which must then determine the set I and return values for $\frac{dU_{p,0}}{dt} = \hat{F}_{p,0}$, for $p\varepsilon I$. The remaining time derivatives for $p\notin I$ are the values $F_{p,0}$ obtained from onesided differences in the hyperbolic system. This method is going to be difficult to explain to a user. However, it is the only method that has, so far, worked reliably.

We will illustrate this method by the following example. This is a system with characteristics of different sign. Chu [3] and Sundstrom [10] have noted the difficulties of setting boundary conditions for such systems.

$$\frac{\partial u_1}{\partial t} = \frac{3\partial u_1}{\partial x} - \frac{4\partial u_2}{\partial x} \qquad 0\le t$$
$$\frac{\partial u_2}{\partial t} = \frac{2\partial u_1}{\partial x} - \frac{3\partial u_2}{\partial x} \qquad 0\le x\le b \tag{17}$$

This system is derived from

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \qquad u = u_1-u_2 \qquad u_1=2u+v$$
$$\frac{\partial v}{\partial t} = -\frac{\partial v}{\partial x} \qquad v = 2u_2-u_1 \qquad u_2=u+v$$

Therefore the following boundary conditions are proper, since they amount to a specification of the characteristic variable on the inflow boundary.

$$\text{at } x=0 \qquad v=2u_2(0,t)-u_1(0,t)=-\sin 2\pi t$$
$$\text{at } x=b \qquad u=u_1(b,t)-u_2(b,t)=\sin 2\pi(b+t) \tag{18}$$

We have chosen the boundary conditions to correspond to the following solution

$$
\begin{aligned}
u_1(x,t) &= 2\sin 2\pi(x,t)+\sin 2\pi(x-t) \\
u_2(x,t) &= \sin 2\pi(x+t)+\sin 2\pi(x-t)
\end{aligned}
\tag{19}
$$

To use the first method of setting the boundary conditions we must specify the set U_I at each boundary point. There is no unique choice here, since neither u_1 nor u_2 are characteristic variables. We will try to specify u_1 at each boundary from the given boundary conditions, namely

$$
\begin{aligned}
&\text{at } x=0 \qquad u_1(0,t)=2u_2(0,t)+\sin 2\pi t \\
&\text{at } x=b \qquad u_1(b,t)=u_2(b,t)+\sin 2\pi(b+t)
\end{aligned}
\tag{20}
$$

In this case $I = \{1\}$, $U_I = \{u_1\}$, $II = \{2\}$, $U_{II} = \{u_2\}$, and $p = 1$ at both boundary points.

A derivative rather than a constrained boundary condition can be obtained by differentiation of the above equation, namely

$$
\begin{aligned}
\frac{du_{1,0}}{dt} &= 2\frac{du_{2,0}}{dt} + 2\pi\cos 2\pi t \qquad \text{at } x=0 \\
\frac{du_{1,0}}{dt} &= \frac{du_{2,0}}{dt} + 2\pi\cos 2\pi(b+t) \qquad \text{at } x=b
\end{aligned}
\tag{21}
$$

The derivative du_2/dt on the right can be computed from the hyperbolic system using one sided differences and then used in the user supplied routine to compute du_1/dt by equation (21) above.

As our results show neither of these methods given by equations (20) and (21) work satisfactorily. They both specify the inflow characteristic variable. The outflow characteristic should be computed using one sided differences. In equation (20) the inflow characteristic is specified by the boundary constraint. However, there is certainly error in the computed value of u_2 used on the right side of the boundary constraint. This error can be transmitted to the other boundary and reflected back. The boundary condition probably should not allow much error in the incoming characteristic.

We tried a third type of boundary condition obtained by differentiating the boundary constraint and combining it with the equation for the outgoing characteristic variable obtained from the hyperbolic system. That is, at $x = 0$

$$-\frac{du_{1,0}}{dt} + 2\frac{du_{2,0}}{dt} = -2\pi\cos 2\pi t$$

$$\frac{du_{1,0}}{dt} - \frac{du_{2,0}}{dt} = F_{1,0} - F_{2,0} .$$

Here $F_{1,0}$ is an approximation to

$$\frac{3\partial u_1}{\partial x} - \frac{4\partial u_2}{\partial x}$$

and $F_{2,0}$

$$\frac{2\partial u_1}{\partial x} - \frac{3\partial u_2}{\partial x}$$

obtained using one sided differences. These equations yield

$$\frac{du_{1,0}}{dt} = 2F_{1,0} - 2F_{2,0} - 2\pi\cos 2\pi t$$
$$\frac{du_{2,0}}{dt} = F_{1,0} - F_{2,0} - 2\pi\cos 2\pi t \qquad (22)$$

There are errors in computing $F_{1,0}$ and $F_{2,0}$, but these will cancel out in the computation of the time derivative of the inflow characteristic ($v=2u_2-u_1$) when this boundary condition is used. Perhaps this is the reason for the superior performance of condition (22) over (20) and (21). However, we do not have a solid theoretical understanding of these results.

5. Some computational results. These results all refer to the solution of equation (17) using a fourth order centered finite difference approximation in the interior and third order one sided differences near the boundary to approximate the spatial derivation $\partial/\partial x$. The Runge-Kutta-Fehlberg [5] method was modified to allow use of the "constrained" boundary condition (20). The "derivative-constrained" condition (21) and the "derivative-characteristic" condition (22) were also used. The parameter ε refers to the error tolerance used in the Runge-Kutta-Fehlberg. The variable J is the number of mesh points, and x=b is the right boundary. The results depend on b, probably because of the way the error is reflected between the two boundaries. The error is the relative error in the computed solution at the indicated time t=T. The parameter N_E is the number of evaluations of the time derivative required in the integration. Each time step requires 6 evaluations (5 if it follows an unsuccessful step).

There seems to be little difference between the results for the constrained-boundary (20) and the derivative-constrained method (21), except for a slight difference in the number of functional evaluations. This difference can be largely eliminated by omitting the error estimate for the constrained boundary variables - at least this was our experience for the single equation (1). Only the characteristic derivative method (22) is free from the error growth which is probably due to multiple reflections from the boundaries. Note that the severity of the error growth depends on the length of the interval (the parameter b). The error reinforcement upon reflection is probably dependent on the phase angle which in turn depends on b. Of course, these results are based on a single, simple test case and may not apply to a given problem.

These computations were performed on the CDC 6400 at the University of Colorado.

References

[1] S. Bowen, "AMPLCT, A Numerical Integration Routine for Systems of Stiff Differential Equations", Department of Meteorology, University of Michigan, report #033390-2-T (1971).

[2] M. Carver, "The FORSIM System for Automated Solution of Sets of Implicitly Coupled Partial Differential Equations", in Advances in Computer Methods for Partial Differential Equations, Vichnevetsky (ed.), AICA, Rutgers University (1975).

[3] C. Chu and A. Sereny, "Boundary Conditions in Finite Difference Fluid Dynamic Codes", Jour. Comp. Phys., V. 15, pp 476-491 (1974).

[4] T. Elvius and A. Sundstrom, "Computationally Efficient Schemes and Boundary Conditions for a Fine Mesh Baratropic Model Based on the Shallow Water Equations", Tellus, v. 25, pp 132-156 (1973).

[5] T. Hull and W. Enright, "A Structure for Programs that Solve Ordinary Differential Equations", Technical Report #66, Department of Computer Science, University of Toronto (1974).

[6] A. Loeb, "Users Guide to a New User-oriented Subroutine for the Automatic Solution of one dimensional Partial Differential Equations", in Advances in Computer Methods for Partial Differential Equations, Ivchnevetsky (ed.), AICA, Rutgers University (1975).

[7] J. Oliger, "Fourth Order Difference Methods for the Initial Boundary-Value Problem for Hyperbolic Equiations", Math. Comp., V. 28, pp 15-25 (1974).

[8] A. Sunderstrom, "Efficient Numerical Methods for Solving Wave Propagation Equations for non-homogeneous Media", C4576-A2, Uppeala University (1974).

[9] L. Shampine and M. Gordon, " Computer Solution of Ordinary Differential Equations", Freeman and Company, San Francisco (1975).

[10] A. Sundstrom, "Note on the paper, 'Boundary Conditions in Finite Difference Fluid Dynamic Codes'", Jour. Comp. Phys., v. 17, pp 450-454 (1975).

[11] Sincovec and N. Madsen, "Software for Nonlinear Partial Differential Equations", Math. Software II Conf., Purdue University (1974).

[12] J. Hastings and R. Roble, "A Technique for Solving the Coupled Time-Dependent Electron and Ion Energy Equations in the Ionosphere", NCAR manuscript, Boulder, Colorado 80302 (1972).

	J	Max λ_r	$\|A\|_\infty$
Inconsistent 2nd order scheme (A).	6	0.0	50.
	11	0.0	199.
	21	0.0	798.
Consistent 2nd order scheme (B).	6	0.0	2.6
	11	0.0	3.3
	21	0.0	4.4
4th order with 3rd order boundary (C).	6	-0.04	2.5
	11	-0.03	3.6
	21	0.0001	4.4
4th order with 4th order boundary (D).	6	-0.54	5.8
	11	0.26	10.1
	21	0.26	15.9

Table I. Behavior of the matrix A of the semidiscrete scheme $\underline{u}'=A\underline{u}+\underline{g}$. Here λ_r denotes the real part of an eigenvalue of A.

	J	T=6.28	T=62.8	T=1256.
(A) Inconsistent, RKF ODE solver.	11	9.44	665.	unstable
(B) Second order spatial, RKF ODE solver.	11	0.056	0.057	0.058
(C) Fourth order spatial. Fourth order Runge-Kutta with fixed $\lambda=\Delta t/\Delta x=1.8$. Third order at boundary.	6 11 21	0.061 0.0059 0.00039	0.069 0.0067 0.00042	0.070 0.0068 0.00042
(D) Fourth order spatial. Fourth order Runge-Kutta with fixed Δt. Fourth order at boundary.	6 11	0.052 0.0031	0.066 0.0038	0.069 0.0039
(C) Fourth order spatial. RKF ODE solver. Third order at boundary.	6 11 21	0.012 0.0043 0.00024	0.016 0.0043 0.00024	0.015 0.0043 0.00025
(D) Fourth order spatial. RKF ODE solver. Fourth order at boundary.	6 11	0.033 0.00069	0.025 0.0047	unstable

Table II. Error for various schemes applied to equation (16)

	J	T=6.28	T=101	T=201	T=402	T=804
(E) Second order, solved by RKF	6	0.21	0.99	1.13	1.57	2.18
	11	0.065	0.33	0.56	0.88	1.39
			$\|u\|=1.14$	$\|u\|=1.31$	$\|u\|=1.65$	$\|u\|=2.27$
	21	0.015	0.10	0.18	0.35	0.72
(F) Fourth order, solved by RKF, 3^{rd} order at body	6	0.043	0.20	0.23	0.26	0.30
	11	0.0022	0.012	0.020	0.034	0.052
			$\|u\|=1.00$	$\|u\|=1.00$	$\|u\|=1.01$	$\|u\|=1.01$
	21	8.4E-5	6.1E-4	1.1E-3	2.2E-3	4.7E-3

Table III. Error for the solution of equation (21). Here T=time, and $\|u\|$ is the maximum norm of the solution.

	J	b	ε	T	N_E	Error
Constrained boundary (20)	11	1.0	0.01	1.0	348	0.052
	11	1.0	0.01	2.0	684	0.14
	11	1.0	0.01	4.0	1344	0.33
	11	1.0	0.01	10.0	3312	0.27
	11	1.0	0.01	20.0	6600	0.63
	6	0.5	0.01	1.0	354	0.011
	6	0.5	0.01	4.0	1350	0.031
	6	0.5	0.01	10.0	3312	1.16
	6	0.5	0.01	20.0	10122	213.00
Derivative-constrained (21)	11	1.0	0.01	1.0	318	0.052
	11	1.0	0.01	2.0	6.2	0.14
	6	0.5	0.01	1.0	318	0.011
	6	0.5	0.01	4.0	1194	0.031
Characteristic-derivative (22)	11	1.0	0.01	1.0	174	0.047
	11	1.0	0.01	2.0	342	0.061
	11	1.0	0.01	4.0	666	0.060
	11	1.0	0.01	10.0	1638	0.060
	11	1.0	0.01	50.0	8094	0.060
	11	0.5	0.001	1.0	606	3.1E-3
	11	0.5	0.001	20.0	11592	3.1E-3

Table IV. Error for solution of equation (17) with various boundary conditions.