Intermediate Boundary Conditions for Time-Split Methods Applied to Hyperbolic Partial Differential Equations*

By Randall J. LeVeque**

Abstract. When time-split or fractional step methods are used to solve partial differential equations numerically, nonphysical intermediate solutions are introduced for which boundary data must often be specified. Here the appropriate boundary conditions are derived for splittings of hyperbolic problems into subproblems with disparate wave speeds. Numerical experiments are performed for the one-dimensional shallow water equations, a quasilinear system with inflow-outflow boundaries. Stability of the initial-boundary value problem is demonstrated for boundary conditions of the type derived here.

1. Introduction. The use of time-split methods for numerically solving hyperbolic partial differential equations which can be split into subproblems with disparate wave speeds has been studied by LeVeque and Oliger [5]. Here we consider in greater depth the problem of properly specifying boundary conditions for the nonphysical intermediate solutions which arise in such schemes. More details and some applications of similar ideas to other partial differential equations may be found in LeVeque [4].

Consider a one-dimensional quasilinear system of the form

\[ u_t = A(x, t, u)u_x, \]

where \( A \) is an \( r \times r \) matrix with real eigenvalues for each \( x, t, \) and \( u \). A time-split method may be advantageous if \( A \) is of the form

\[ A = A_f + A_s, \]

where the problems

\[ u_t = A_f u_x, \]  

and

\[ u_t = A_s u_x, \]


**Current address: Department of Mathematics, University of Washington, Seattle, Washington 98195.**

*1986 American Mathematical Society
0025-5718/86 $1.00 + $25 per page

License or copyright restrictions may apply to redistribution; see http://www.ams.org/journal-terms-of-use
can each be solved more efficiently than (1.1). Typically, \( A_s \) has small eigenvalues so that solutions to (1.3b) consist of slow waves, while \( A_f \) has large eigenvalues but simple structure. For example, \( A_f \) may be a constant matrix for which (1.3a) can be solved exactly on the computational grid. Alternatively, \( A_f \) may be sparse relative to \( A \), so that small time steps can be taken more efficiently on (1.3a) than on the full problem (1.1). Various examples are given in [4] and [5].

Let \( U_m^n \) denote the grid function approximation to the solution \( u(x_m, t_n) \), where \( x_m = mh \) and \( t_n = nk \). For the time-split method, we apply the second-order accurate Strang splitting [7] to the subproblems (1.3). The numerical method for (1.1) is then

\[
U^* = Q_f(t_n + \frac{1}{2}k, t_n)U^n,
\]

\[
U^{**} = Q_s(t_{n+1}, t_n)U^*,
\]

\[
U^{n+1} = Q_f(t_{n+1}, t_n + \frac{1}{2}k)U^{**},
\]

where, for example, \( Q_f(t_n + \frac{1}{2}k, t_n) \) is some approximate solution operator for the problem (1.3a) from time \( t_n \) to time \( t_n + \frac{1}{2}k \).

When the time-split method is used to solve an initial-boundary value problem (IBVP), it is necessary to specify boundary conditions for the nonphysical intermediate solutions \( U^* \) and \( U^{**} \). In [5], it was shown how this could be done for constant coefficient systems at an inflow boundary. Here we show how the same techniques can be applied to handle more general IBVPs. In Section 2, boundary conditions for a variable coefficient system at an inflow boundary are computed. In Section 3, inflow-outflow boundary conditions are derived for constant coefficient problems. These ideas may be combined to handle general problems. In Section 4, the one-dimensional shallow water equations are considered as an example. This is a quasilinear system of equations with inflow-outflow boundaries.

Stability theory is discussed in Section 5. Assuming that the time-split method (1.4) is stable for the Cauchy problem (with domain \( -\infty < x < \infty \)), it is shown that any of the boundary conditions derived here give a stable scheme for the IBVP.

For convenience, we will assume that \( Q_f \) is the exact solution operator for the problem (1.3a), while \( Q_s \) consists of a single step of some finite-difference method. Then \( U^* \) and \( U^{**} \) are the only intermediate solutions that arise. Other intermediate solutions which may arise, for example, if \( Q_f \) consists of several steps of a finite-difference method with time step smaller than \( k/2 \), can be handled similarly.

We will also assume that \( \|A_f\| = O(1) \) while \( \|A_s\| = O(\epsilon) \) with \( \epsilon \ll 1 \). Then the second-order accurate scheme \( Q_s \) will in fact generally be \( O(ek^2) \) accurate on \( u_i = A_i u_x \). It can be shown that the time-split method (1.4) is then \( O(ek^2) \) accurate on the full Cauchy problem (1.1) [4]. By contrast, the use of a second-order accurate method directly on the unsplit problem (1.1) would typically be only \( O(k^2) \) accurate.

In order to maintain the \( O(ek^2) \) accuracy of the time-split method on IBVPs, it is necessary to derive boundary conditions which are locally \( O(\epsilon k^2) \) accurate. The approach we take generates a series expansion for the correct boundary conditions which can be truncated appropriately to achieve this accuracy.

We recall the basic idea used to derive intermediate boundary conditions [4], [5]. Each step of the split method (1.4) is an approximate solution to one of the equations in (1.3). These equations differ from the original equation (1.1), and we
attempt to derive appropriate boundary conditions for these equations based on the
given boundary data for (1.1).

To avoid confusion, it is useful to introduce different variables to denote solutions
to (1.3). For example, the intermediate solution \( U^* \) in (1.4) is an approximation to
\( u^*(x, t_n + k/2) \), where \( u^* \) satisfies (1.3a),
\[
(1.5) \quad u^*_t = A_f u^*_x,
\]
for \( t \geq t_n \) with initial conditions
\[
(1.6) \quad u^*(x, t_n) = u(x, t_n).
\]
It is clear that the new variable \( u^*(x, t) \) evolves differently than the true solution
\( u(x, t) \) which satisfies (1.1). Consequently, the boundary conditions for \( u^* \) will differ
from those given for \( u \), and determining these new boundary conditions will allow
us to specify \( U^* \) at the boundary.

A simple example of this procedure for constant coefficient problems was given in [5].

2. Variable Coefficient Systems—Inflow Boundaries. Consider the quarter-plane
problem
\[
(2.1a) \quad u_t = A(x, t) u_x, \quad x \geq 0, \ t \geq 0,
\]
with initial conditions
\[
(2.1b) \quad u(x, 0) = f(x),
\]
and inflow boundary conditions
\[
(2.1c) \quad u(0, t) = g(t).
\]
Here, \( A(x, t) \) is an \( r \times r \) matrix with negative eigenvalues. Assume the matrix \( A \) is
split as in (1.2) and, for simplicity, suppose that \( A_f \) is constant while \( A_s = A_s(x, t) \).
It should be clear from this discussion how to handle problems in which \( A_f \) is also
variable as well as quasilinear problems in which \( A \) also depends on \( u \).

We wish to determine appropriate boundary conditions at \( x = 0 \) for the inter-
mediate solutions \( U^* \) and \( U^{**} \) in (1.4). First, consider \( U^* \). Suppose that at time \( t_n \),
\( U^*_m = u(x_m, t_n) \) for \( m = 0, 1, \ldots \). Since we have assumed that \( Q_f \) is the exact
solution operator for (1.3a), it follows that, away from the boundary at least,
\( U^*_m = u^*(x_m, t_n + \frac{1}{2}k) \) where \( u^*(x, t) \) is the solution to the problem
\[
(2.2a) \quad u^*_t(x, t) = A_f u^*_x(x, t), \quad x \geq 0, \ t \geq t_n,
\]
with initial conditions
\[
(2.2b) \quad u^*(x, t_n) = u(x, t_n).
\]
More generally, if \( Q_f \) is a second-order accurate approximation to the exact solution
operator, then
\[
U^*_m = u^*(x_m, t_n + \frac{1}{2}k) + O(k^3).
\]
The problem of determining the correct boundary conditions for \( U^* \) can be replaced
by that of determining the correct boundary conditions for the continuum function
\( u^* \) solving (2.2). These can be calculated in terms of the boundary data \( g(t) \) in
(2.1c).
For \( \tau \geq 0 \), we have
\[
(2.3) \quad u^*(0, t_n + \tau) = u^*(0, t_n) + \tau u_t^*(0, t_n) + \frac{1}{2} \tau^2 u_{tt}^*(0, t_n) + \mathcal{O}(\tau^3).
\]
Using (2.2a), this becomes
\[
(2.4) \quad u^*(0, t_n + \tau) = u^*(0, t_n) + \tau A_f u_x^*(0, t_n) + \frac{1}{2} \tau^2 A_f^2 u_{xx}^*(0, t_n) + \mathcal{O}(\tau^3).
\]
Since the initial conditions (2.2b) hold for all \( x \), that relation can be differentiated with respect to \( x \), giving \( u_t^*(x, t_n) = u_x(x, t_n) \) and similarly for higher derivatives. So (2.4) becomes
\[
(2.5) \quad u^*(0, t_n + \tau) = u^*(0, t_n) + \tau A_f u_x^*(0, t_n) + \frac{1}{2} \tau^2 A_f^2 u_{xx}^*(0, t_n) + \mathcal{O}(\tau^3).
\]
Using (2.1a), we can reexpress this in terms of time-derivatives of \( u \) along the boundary. We have
\[
u_x(0, t_n) = A_f^{-1}(0, t_n) u_t(0, t_n),
\]
and by differentiating (2.1a) with respect to both \( x \) and \( t \) and solving for \( u_{xx} \), we find that
\[
u_{xx} = A_f^{-1} \left[ A_f^{-1}(u_{tt} - A_f A_f^{-1} u_t) - A_f A_f^{-1} u_t \right].
\]
Higher-order derivatives can also be computed. Using these expressions in (2.5), we obtain
\[
(2.6) \quad u^*(0, t_n + \tau) = u^*(0, t_n) + \tau A_f A_f^{-1}(0, t_n) u_t(0, t_n) + \frac{1}{2} \tau^2 A_f^2 A_f^{-1}(0, t_n)
\]
\[
\times \left[ A_f^{-1}(0, t_n) u_{tt}(0, t_n) - (A_f^{-1}(0, t_n) A_f(0, t_n) + A_f x(0, t_n)) \right]
\]
\[
= n^* \left[ A_f^{-1}(0, t_n) u_t(0, t_n) \right] + \mathcal{O}(\tau^3).
\]
We can replace \( u(0, t_n) \) by \( g(t_n) \), giving an expression for the boundary conditions \( u^*(0, t_n + \tau) \) in terms of the boundary data \( g(t) \) (and its derivatives). Evaluating this at \( \tau = \frac{1}{2} k \) gives a series expansion for the boundary data \( U^*_0 = u^*(0, t_n + \frac{1}{2} k) \). This must, in general, be truncated at some point. In many cases, it can also be simplified.

For simplicity, we will first assume that \( \kappa(A) = O(1) \), where \( \kappa(A) \) is the condition number of the matrix \( A \) defined by \( \kappa(A) = \| A \| \| A^{-1} \| \). This means, in particular, that all eigenvalues of \( A \) are \( O(1) \) and hence, all waves in the original problem travel with speeds which are \( O(1) \). This is the case, for example, in the one-dimensional shallow water equations discussed in Section 4.

In practice, splitting is often used in situations where \( A \) also has \( O(\epsilon) \) eigenvalues, so that waves travel at disparate speeds in the original problem. We will discuss this case later in this section.

Recall that we desire an \( O(\epsilon k^2) \) accurate approximation to the boundary data \( u^*(0, t_n + k/2) \). We wish to simplify and truncate (2.6) to obtain this accuracy. In the present case, with \( A_f \) constant, \( A_x \) and \( A_t \) are \( O(\epsilon) \), so we can drop several terms in (2.6), leaving
\[
(2.7) \quad u^*(0, t_n + k/2) = g(t_n) + \frac{1}{2} k A_f A_f^{-1}(0, t_n) g'(t_n)
\]
\[
+ \frac{1}{2} k^2 A_f^2 A_f^{-2}(0, t_n) g''(t_n) + O(\epsilon k^2 + k^3).
\]
Moreover, since all eigenvalues of $A$ are $O(1)$, the matrix $A_j$ must be of full rank, so that $A_j^{-1}$ exists and $\kappa(A_j^{-1}) = O(1)$. Then,

$$A_j A_j^{-1}(0, t_n) = \left[ (A_j + A_s) A_j^{-1} \right]^{-1} = \left[ I + A_s A_j^{-1} \right]^{-1} = I + O(\epsilon)$$

and, similarly,

$$A_j A_j^{-1}(0, t_n) = I + O(\epsilon)$$

for $j = 1, 2, \ldots$. Factoring an identity matrix out of $A_j A_j^{-1}(0, t_n)$ in each term of (2.7) we obtain the $O(\epsilon k^2)$ accurate boundary conditions

$$U_0^* = g(t_n + \frac{1}{2}k) + \frac{1}{2}k \left( A_j A_j^{-1}(0, t_n) - I \right) g'(t_n).$$

Boundary conditions for $U^*$ can be determined similarly. The easiest way to proceed is to work backwards from time $t_{n+1}$. We define $u^*$ as the solution to

$$u^*(0, t_{n+1}; x) = A_j u^*(x, t_{n+1}); x > 0, t \leq t_{n+1},$$

with "initial" conditions

$$u^*(x, t_{n+1}) = u(x, t_{n+1}); x > 0.$$

Then, $U^* \approx u^*(t_{n+1} - \frac{1}{2}k)$. Proceeding as before, we find an expression analogous to (2.6):

$$u^* (0, t_{n+1} - \tau) = u(0, t_{n+1}) - \tau A_j A_j^{-1}(0, t_{n+1}) u(0, t_{n+1})$$

$$+ \frac{1}{2} \tau^2 A_j^2 A_j^{-2}(0, t_{n+1}) u_{tt}(0, t_{n+1}) + O(\epsilon^2 \tau^2 + \tau^3).$$

Corresponding to (2.8), we have the boundary condition

$$U_0^* = g(t_{n+1} + \frac{1}{2}k) - \frac{1}{2}k \left( A_j A_j^{-1}(0, t_{n+1}) - I \right) g'(t_{n+1}).$$

Finally, we note that data for points near the boundary may be determined similarly. For example, if boundary conditions $U_j^*$ for $0 \leq j \leq p$ are needed, they can be obtained as approximations to

$$u^*(j h, t_n + \tau) = u^*(0, t_n + \tau) + j h u_x^*(0, t_n + \tau) + \frac{1}{2} (j h)^2 u_{xx}^*(0, t_n + \tau) + \cdots$$

$$= u(0, t_n) + \left( \tau A_j + j h I \right) u_x(0, t_n) + \frac{1}{2} \left( \tau A_j + j h I \right)^2 u_{xx}(0, t_n) + \cdots.$$

From here, we can proceed as from (2.5), to again obtain an expression in terms of $g$ and its derivatives.

Now suppose that the original problem has waves traveling at disparate speeds, so that the matrix $A$ has some eigenvalues which are $O(1)$ and some which are $O(\epsilon)$. In this case, $\|A^{-1}\| = O(1/\epsilon)$. However, if we restrict our attention to smooth solutions and assume that $\partial_j^\tau u = O(1)$ for all $j$, then the expansions (2.5) and (2.6) are still valid. Note that (2.6) involves the matrix $A^{-1}$, but only in expressing $x$-derivatives in terms of $t$-derivatives and so, by our smoothness assumptions, the factors multiplying powers of $\tau$ are all $O(1)$. Simplifying (2.6) may be more complicated, however. For example, $A_j$ no longer has full rank and so $A_j A_j^{-1} \neq I + O(\epsilon)$.

For simplicity, we will assume that the system is partitioned into fast and slow components at the boundary $x = 0$, i.e.,

$$(2.12) \quad A(0, t) = \begin{pmatrix} A_1(0, t) & 0 \\ 0 & \varepsilon A_2(0, t) \end{pmatrix}.$$
where \( \kappa(A_1) \) and \( \kappa(A_2) \) are both \( O(1) \). The additional complications which arise if coupling terms are present can also be handled. Consider a splitting in which \( A_f \) is again constant, and of the form

\[
(2.13a) \quad A_f = \begin{bmatrix} A_{1f} & 0 \\ 0 & 0 \end{bmatrix},
\]

so that

\[
(2.13b) \quad A_x(0,t) = \begin{bmatrix} \epsilon A_{1x}(0,t) & 0 \\ 0 & \epsilon A_{2x}(0,t) \end{bmatrix}.
\]

Then, if \( A_x \) and \( A_f \) are again \( O(\epsilon) \), we easily compute that

\[
A_f u_x = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A_{1f} A_{1}^{-1}(0,t) - I & 0 \\ 0 & 0 \end{bmatrix} u_t
\]

and

\[
A_f^2 u_{xx} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + O(\epsilon) u_{tt}.
\]

Partitioning \( g(t) = [g_1(t), g_2(t)]^T \), we obtain an expression analogous to (2.8) which again gives \( O(\epsilon k^2) \) accurate boundary conditions:

\[
U_0^* = \begin{bmatrix} g_1(t_n + k/2) \\ \frac{g_2(t_n)}{\epsilon} \end{bmatrix} + \frac{k}{2} \left( A_1^{-1}(0,t_n) - I \right) g_1'(t_n).
\]

Notice that the second component of \( U_0^* \) is simply \( g_2(t_n) \). This is natural, since with the splitting (2.13), the second component of \( u \) remains unchanged in the first step of the split method.

If coupling terms between fast and slow components are present at the boundary, then further complications arise, but can also be handled. To get an indication of what happens without unnecessary complication, we consider only the constant coefficient system of two equations

\[
(2.14) \quad u_x = \begin{bmatrix} -1 & -\epsilon \\ 0 & -\epsilon \end{bmatrix} u_x, \quad u(0,t) = g(t).
\]

We write \( u = (v,w)^T \). If the solution is to have bounded \( x \)-derivatives of all orders, then the boundary conditions must be of the form

\[
(2.15) \quad g(t) = \begin{bmatrix} g_1(t) \\ g_2(\epsilon t) \end{bmatrix},
\]

where the \( g_j(t) \) have bounded derivatives. Taking the natural splitting with \( A_f = [-1 0; 0 0] \), we find that

\[
A_f \partial_x^j u = A_f A^{-j} \partial_x^j u = \begin{bmatrix} 1 & -(1 + \epsilon^{-1} + \cdots + \epsilon^{-j+1}) \\ 0 & 0 \end{bmatrix} \partial_x^j u
\]

\[
= \begin{bmatrix} g_1^{(j)}(t) - (\epsilon + \epsilon^2 + \cdots + \epsilon^j) g_2^{(j)}(\epsilon t) \\ 0 \end{bmatrix}.
\]

Notice that even though \( A_f A^{-j} = O(\epsilon^{-j+1}) \), the required smoothness gives an \( O(1) \) expression for the terms in (2.5).
For this problem we can obtain \(O(\varepsilon k^2)\) accurate boundary conditions by taking
\[
U_0^* = \begin{bmatrix} g_1(t_n) \\ g_2(t_n) \end{bmatrix} + \frac{1}{2} k \begin{bmatrix} g_1(t_n) - \varepsilon g_2'(\varepsilon t_n) \\ 0 \end{bmatrix} + \frac{1}{8} k^2 \begin{bmatrix} g_1''(t_n) \\ 0 \end{bmatrix} + O(\varepsilon k^2).
\]
(2.16)

In fact, for this simple problem we can determine the exact boundary condition
\[
U_0^*.
\]
Since \(1 + \varepsilon^{-1} + \cdots + \varepsilon^{-j+1} = \varepsilon(1 - \varepsilon^{-j})/(\varepsilon - 1)\), we find that
\[
A_j \partial_x v = \partial_t^j v - \frac{\varepsilon}{\varepsilon - 1} (1 - \varepsilon^{-j}) \partial_x^j w,
\]
so that
\[
v^*(0, t_n + k/2) = v(0, t_n) + \frac{1}{2} k A_j v_x + \frac{1}{8} k^2 A_j^2 v_{xx} + \cdots
\]
(2.17a)
\[
= v(0, t_n + k/2) - \varepsilon \left[ w(0, t_n + k/2) - w(0, t_n + k/2\varepsilon) \right]/(\varepsilon - 1)
\]
\[
= g_1(t_n + k/2) - \varepsilon \left[ g_2(\varepsilon(t_n + k/2)) - g_2(\varepsilon t_n + k/2) \right]/(\varepsilon - 1),
\]
while
\[
w^*(0, t_n + k/2) = w^*(0, t_n) = g_2(\varepsilon t_n).
\]

3. Inflow-Outflow Boundaries. We next consider the case in which \(A\) has both positive and negative eigenvalues. For simplicity we only treat the constant coefficient problem and assume that \(\kappa(A) = O(1)\) in order to isolate the essential new features which arise. More general problems can be handled by combining the techniques used here with those of Section 2.

Consider \(u_i = A u_x\) for \(x \geq 0, \ t \geq 0\) and assume that \(A\) is in block-diagonal form
\[
A = \begin{bmatrix} A^I & 0 \\ 0 & A^{II} \end{bmatrix},
\]
with the eigenvalues of \(A^I\) negative and those of \(A^{II}\) positive. \(A^I\) and \(A^{II}\) are assumed to have the same form and are partitioned similarly into blocks, e.g., \(A^I_j\) and \(A^{II}_j\). Partition \(u = (v, w)^T\) conformally with \(A\). Then at \(x = 0\), the elements of \(v\) are inflow variables while those of \(w\) are outflow variables. The boundary conditions are assumed to be of the form
\[
v(0, t) = Sw(0, t) + g(t),
\]
where \(S\) is a constant matrix and \(g\) is a given function. We now split \(A\) as
\(A = A^I + A^s\), with \(A^I\) and \(A^s\) again block-diagonal. Moreover, we suppose that the eigenvalues of \(A^I_j\) are negative and those of \(A^{II}_j\) positive.

We consider only the problem of computing \(U_0^*\) and suppose, as usual, that \(Q_j\) is the exact solution operator \(\exp(kA_j \partial_x)\). Then \(W_0^*\) is determined from the interior and we need only specify \(V_0^*\).
If \( Q_j \) is not the exact solution operator, then it is necessary to specify \( W_0^* \) as well. Since \( W_0^* \) approximates the outflow variables satisfying \( w^*_t = A^3 w^*_x \), it is typically easy to derive a one-sided scheme for this equation which can be applied to obtain \( W_0^* \) from \( W_j^n, j \geq 0 \).

To determine \( V_0^* \), we introduce \( u^* = (v^*, w^*) \) which solves the subproblem \( u^*_t = A^3 u^*_x \) and find as usual that

\[
u^*(0, t_n + k/2) = u(0, t_n) + \frac{1}{2} k A^3 A^{-1} u_t(0, t_n) + \frac{1}{2} k^2 A^3 A^{-2} u_{tt}(0, t_n) + \ldots .
\]

Again using \( A^3 A^{-2} = I + O(\varepsilon) \), we obtain the \( O(\varepsilon k^2) \) accurate boundary conditions

\[
(3.3) \quad U_0^* = u(0, t_{n+1/2}) + \frac{1}{2} k \left(A^3 A^{-1} - I\right) u_t(0, t_n).
\]

Introducing the matrix

\[
B = A^3 A^{-1} - I = \begin{bmatrix} A^3(A^{-1})^T & 0 \\ 0 & A^{3T}(A^{-1})^T \end{bmatrix} = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix},
\]

we can rewrite (3.3) as

\[
(3.4a) \quad V_0^* = v(0, t_{n+1/2}) + \frac{1}{2} kB_{11} v_t(0, t_n),
\]

\[
(3.4b) \quad W_0^* = w(0, t_{n+1/2}) + \frac{1}{2} kB_{22} w_t(0, t_n).
\]

By differentiating the boundary conditions (3.2) we obtain

\[
v_t(0, t_n) = Sw_t(0, t_n) + g'(t_n).
\]

Using this and (3.2), (3.4a) becomes

\[
V_0^* = S [w(0, t_{n+1/2}) + g(t_{n+1/2})] + \frac{1}{2} kB_{11} [Sw_t(0, t_n) + g'(t_n)].
\]

Recall that \( W_0^* \) is already known. We can thus solve (3.4b) for \( w(0, t_{n+1/2}) \).

Using this in (3.5), yields

\[
V_0^* = S \left[ W_0^* - \frac{1}{2} kB_{22} w_t(0, t_n) \right] + g(t_{n+1/2}) + \frac{1}{2} kB_{11} \left[ Sw_t(0, t_n) + g'(t_n) \right]
\]

\[
(3.6) \quad V_0^* = S W_0^* + g(t_{n+1/2}) + \frac{1}{2} k \left[B_{11} g'(t_n) + (B_{11} S - SB_{22}) w_t(0, t_n) \right].
\]

The \( w_t \) term must in general be approximated by a finite difference,

\[
(3.7) \quad V_0^* = S W_0^* + g(t_{n+1/2}) + \frac{1}{2} k B_{11} g'(t_n) + (B_{11} S - SB_{22}) (W_0^n - W_0^{n-1})
\]

Alternatively, we can replace \( w_t \) by \( A^3 w_t \) and approximate this by a finite difference of \( W \) at time \( t_n \). This approach is particularly useful when more terms of the series are kept and higher-order derivatives must be approximated.

The use of such boundary conditions is illustrated in the next section, where the one-dimensional shallow water equations are considered.

Boundary data at points near the boundary can be found in a similar manner. For example, if data \( V_j^{n+1} \) is needed for some \( 0 < j \leq p \), we can expand \( v \) in \( x \)-derivatives, switch to \( t \)-derivatives along the boundary, convert these to \( t \)-derivatives of \( w \) using \( v_t = Sw_t + g' \), and finally switch back to \( x \)-derivatives of \( w \), obtaining

\[
v(j, t_{n+1}) = v(0, t_{n+1}) + jh (A^3)^{-1} [SA^3 w_x(0, t_{n+1}) + g'(t_{n+1})]
\]

\[
+ \frac{1}{2} (jh)^2 (A^3)^{-2} [S(A^3)^2 w_{xx}(0, t_{n+1}) + g''(t_{n+1})] + \ldots .
\]

These boundary conditions are suggested by Goldberg and Tadmor [1], [2] for general inflow-outflow problems.
4. The Shallow Water Equations. In order to illustrate the derivation of intermediate boundary conditions for a specific example, we will consider the one-dimensional shallow water equations on a strip, which we write in symmetric form as

\[
\begin{bmatrix}
u \\
\phi
\end{bmatrix}_t = - \begin{bmatrix}
u & \phi/2 \\
\phi/2 & u
\end{bmatrix} \begin{bmatrix}
u \\
\phi
\end{bmatrix}, \quad 0 \leq x \leq 1, \ t \geq 0.
\]

Here \(u(x, t)\) is the velocity and \(\phi(x, t) = 2\sqrt{gH(x, t)}\), where \(H\) is the height of the fluid and \(g\) is the gravitational constant. We will make the realistic assumption that \(u\) is small compared to \(\phi\) and that variations in \(\phi\) are small compared to some mean value \(\phi_0\):

\[
|\phi - \phi_0| \leq \epsilon \phi_0, \quad |u| \leq \epsilon \phi_0
\]

with \(\epsilon \ll 1\). Moreover, we consider only smooth solutions for which \(u_x, \phi_x\) and higher derivatives are also \(O(\epsilon \phi_0)\).

For computational convenience we change variables and compute in the characteristic variables \(p\) and \(\sigma\) defined by

\[
p(x, t) = u(x, t) + \phi(x, t), \quad \sigma(x, t) = u(x, t) - \phi(x, t).
\]

We can always transform back to find \(u = (p + \sigma)/2\) and \(\phi = (p - \sigma)/2\). Rewriting the differential equation (4.1) in terms of \(p\) and \(\sigma\), gives

\[
\begin{bmatrix}
p \\
\sigma
\end{bmatrix}_t = \frac{1}{4} \begin{bmatrix}3p + \sigma & 0 \\
0 & p + 3\sigma
\end{bmatrix} \begin{bmatrix}
p \\
\sigma
\end{bmatrix}_x.
\]

Under the assumption (4.2), the variable \(p\) always flows to the right while \(\sigma\) always flows to the left. Appropriate boundary conditions are thus

\[
\begin{align}
\rho(0, t) &= \alpha_0 \sigma(0, t) + g_0(t), \\
\sigma(1, t) &= \alpha_1 \rho(1, t) + g_1(t).
\end{align}
\]

For concreteness, we will specify \(\phi(0, t) = g(t)\) at \(x = 0\) and use nonreflecting boundary conditions at \(x = 1\). These boundary conditions can be written in the form (4.4) as

\[
\begin{align}
\rho(0, t) &= \sigma(0, t) + 2g(t), \\
\sigma(1, t) &= -\phi_0.
\end{align}
\]

We will split the coefficient matrix \(A\) appearing in (4.3) as \(A = A_f + A_s\) with

\[
A_f = \frac{1}{2} \begin{bmatrix} -\phi_0 & 0 \\
0 & \phi_0
\end{bmatrix}, \quad A_s = \frac{1}{4} \begin{bmatrix} 3p + \sigma - 2\phi_0 & 0 \\
0 & p + 3\sigma + 2\phi_0
\end{bmatrix}.
\]

Then \(\|A_s\| = O(\epsilon)\), while \(v_t = A_f v_x\) is a constant coefficient problem for which the exact solution operator is easily computed. Taking \(k = 4h/\phi_0\) and denoting the grid-function approximations to \(p\) and \(\sigma\) by \(R\) and \(S\), respectively, the time-split method (1.4) on \(0 \leq x \leq 1\) with \(h = 1/N\) is simply

\[
\begin{align}
R^m_e &= R^{m-1}_e, & m = 1, 2, \ldots, N, \\
S^m_e &= S^{m+1}_e, & m = -1, 0, \ldots, N - 1, \\
R^0 = Q(k) R^0, & m = 0, 1, \ldots, N - 1, \\
S^m = Q(k) S^m, & m = 0, 1, \ldots, N - 1.
\end{align}
\]

Here \(Q(k)\) is a second-order accurate scheme (say Lax-Wendroff) for the problem \(v_t = A_s v_x\). Such a scheme will in fact generally be \(O(\epsilon_2 k^2)\) accurate on this problem, the two factors of \(\epsilon\) arising because the coefficients \(A_s\) are \(O(\epsilon)\) and, in addition,
derivatives of the solution are $O(\varepsilon)$. The time-split method remains $O(\varepsilon^2 k^2)$ accurate for the full problem (4.3) [5]. By contrast, applying Lax-Wendroff directly to the unsplit problem is only $O(\varepsilon k^2)$ accurate (derivatives of the solution are still $O(\varepsilon)$, but the coefficients are $O(1)$).

At the left boundary we need to specify $R_0^*$, $R_{-1}^*$, and $R_{0}^{n+1}$. Note that by specifying $R_{-1}^*$ and computing $S_{-1}^*$ we avoid having to specify any boundary values for $R_{-1}^*$.

The given boundary conditions (4.5a) provide $R_{0}^{n+1}$,

\begin{equation}
R_{0}^{n+1} = S_{0}^{n+1} + 2g(t_{n+1}).
\end{equation}

We next apply the procedure of Section 3 to compute $R_0^*$. It can be verified that the expression (3.3) yields $O(\varepsilon^2 k^2)$ accurate boundary data for this problem, provided $A^{-1}$ is evaluated at $(\rho(0, t_n), \sigma(0, t_n))$. The matrix $B = A^{-1} - I$ is given by

\begin{equation}
B = \begin{bmatrix}
2\phi_0/(3\rho + \sigma) - 1 & 0 \\
0 & -2\phi_0/(\rho + 3\sigma) - 1
\end{bmatrix} = O(\varepsilon),
\end{equation}

and the expression (3.6) becomes

\begin{equation}
R_0^* = S_0^* + 2g(t_{n+1/2})
+ \frac{1}{2}k \left[ \left( \frac{8\phi_0(\rho + \sigma)}{(3\rho + \sigma)(\rho + 3\sigma)} \right) \sigma(0, t_n) + 2 \left( \frac{2\phi_0}{3\rho + \sigma} - 1 \right) g'(t_n) \right]
= S_0^* + 2g(t_n + \frac{2\phi_0}{3\rho + \sigma}) + \frac{1}{2}k \left( \frac{8\phi_0(\rho + \sigma)}{(3\rho + \sigma)(\rho + 3\sigma)} \right) \sigma(0, t_n),
\end{equation}

where $\rho$ and $\sigma$ are evaluated at $(0, t_n)$. This can be approximated by

\begin{equation}
R_0^* = S_0^* + 2g(t_n + \alpha \phi_0 k) + \left( \frac{4\alpha \phi_0(R_0^n + S_0^n)}{R_0^n + 3S_0^n} \right) (S_0^n - S_0^{n-1}),
\end{equation}

where $\alpha = 1/(3R_0^n + S_0^n)$.

In order to find $R_{-1}^*$ we approximate $\rho^*(-h, t_n + k/2)$. This is equal to $\rho^*(0, t_n + k)$ and proceeding as in Section 3 we find the approximation

\begin{equation}
R_{-1}^* = S_{-1}^* + 2g(t_n + 2\alpha \phi_0 k) + \left( \frac{8\alpha \phi_0(R_0^n + S_0^n)}{R_0^n + 3S_0^n} \right) (S_0^n + S_0^{n-1}),
\end{equation}

with $\alpha$ as above.

At the right boundary, we still need to specify $S_0^*$, $S_{0}^{**}$ and $S_{0}^{n+1}$. Since the boundary condition (4.5b) is time-independent, applying the general procedure at this boundary yields simply

\begin{equation}
S_0^* = S_{0}^{**} = S_{0}^{n+1} = -\phi_0.
\end{equation}

Figure 1 shows the results of some computations using the boundary conditions (4.6) through (4.9). The following initial and boundary conditions were used:

\begin{align*}
u(x, 0) &= \varepsilon e^{-2\pi^2(x-1)^2}, \\
\phi(x, 0) &= \phi_0 + \varepsilon \cos(2\pi x), \\
\phi(0, t) &= \phi_0 + \varepsilon \cos(\phi_0 \pi t), \\
\sigma(1, t) &= -\phi_0.
\end{align*}
Errors in solutions to the shallow water equations at $t = 0.05$ with the $O(\varepsilon^2 k^2)$ accurate boundary conditions (4.6) through (4.9) with various values of $k$ and $\varepsilon$. Errors due to the boundary conditions are present only for $x < 0.2$.

Computed solutions were compared with results obtained on a much finer mesh. In Figure 1 the 2-norm of the error at each mesh point is plotted at time $t = 0.05$. In each calculation, $\phi_0 = 8$ and $k = h/2$, but different values of $\varepsilon$ and $h$ have been used to investigate convergence. Signals propagate at speed $c = \phi_0/2 = 4$, so that at the time shown the effects of improper boundary conditions have been felt only for $x < 0.2$ and $x > 0.8$. For $0.2 < x < 0.8$ errors are due solely to the time-split method used in the interior. It is this accuracy which we are trying to match at the boundary. From Figure 1 it is clear that the boundary conditions have the same order of accuracy ($O(\varepsilon^2 k^2)$) as the interior scheme. In fact, the boundary conditions (4.9) used at $x = 1$ are the correct outflow boundary conditions and introduce no additional error. At $x = 0$, the boundary conditions are $O(\varepsilon^2 k^2)$ accurate, but apparently have a larger error constant than the interior scheme. In all cases, the error near $x = 0$ is roughly 10 times larger than the error in the interior.
The striking oscillations near this boundary are due to the fact that the boundary conditions (4.6) for \( R_0^{n+1} \) have much smaller error than the boundary conditions derived for \( R_0^* \) and \( R_1^* \). This leads to a larger error in odd-numbered mesh points than in even-numbered ones. These oscillations die out as the wave propagates into the interior, owing to the dissipative nature of Lax-Wendroff, but die out slowly, because Lax-Wendroff is applied only with the small coefficients \( A_j \).

The oscillations do not indicate any stability problems. Calculations to much larger times show that the method is stable and maintains \( O(\varepsilon^2 k^2) \) accuracy. For example, Figure 2 shows the errors at time \( t = 1 \).

**Higher-Order Accuracy at the Boundary.** As the results in Figure 1 demonstrate, the boundary conditions (4.6) through (4.9) have the same order of accuracy as the interior scheme, but may have a larger error constant. To avoid the loss of accuracy which this implies, it may be desirable to use boundary conditions with a higher order of accuracy. This can be accomplished by retaining more terms in the asymptotic...
expansion of $\rho^*(0, t_n + \tau)$. After some manipulations involving $\sigma^*(0, t_n + \tau)$, one obtains

$$
\rho^*(0, t_n + \tau) = \sigma^*(0, t_n + \tau) + 2g \left( t_n + \frac{2\tau \phi_0}{3\rho + \sigma} \right) + \tau \left( \frac{8\phi_0(\rho + \sigma)}{(3\rho + \sigma)(\rho + 3\sigma)} \right) \sigma_{tt} 
+ 2\tau^2 \phi_0 \left( \frac{1}{(3\rho + \sigma)^2} - \frac{1}{(\rho + 3\sigma)^2} \right) \sigma_{ttt} 
- 6 \left( \frac{\rho_i^2}{(3\rho + \sigma)^3} - \frac{\sigma_i^2}{(\rho + 3\sigma)^3} \right) \right) + \mathcal{O}(\epsilon^3 k^2 + \epsilon^2 k^3).
$$

**Figure 3**

Errors in solutions to the shallow water equations at $t = 0.05$ with the $\mathcal{O}(\epsilon^3 k^2 + \epsilon^2 k^3)$ accurate boundary conditions (4.6), (4.9), (4.10) and (4.11).
Boundary conditions which are $\mathcal{O}(\varepsilon^3 k^2 + \varepsilon^2 k^3)$ accurate are obtained by approximating this at $\tau = \frac{1}{2} k$:

$$R^*_0 = S_0^* + 2g(t_n + \alpha \phi_0 k) + 4\phi_0 \alpha \beta (R^n_0 + S_0^n)\left(\frac{3}{2} S_0^n - 2 S_0^{n-1} + \frac{1}{2} S_0^{n-2}\right)$$

$$+ \frac{1}{2} \phi_0^2 \left(\alpha^2 - \beta^2\right)\left(S_0^n - 2 S_0^{n-1} + S_0^{n-2}\right)$$

$$- 6\left(\alpha^3 \left(R^n_0 - R_0^{n-1}\right)^2 - \beta^3 \left(S_0^n - S_0^{n-1}\right)^2\right),$$

(4.10)

where $\alpha = 1/(3R_0^n + S_0^n)$ and $\beta = 1/(R_0^n - 3S_0^n)$.

The corresponding expression for $R^*_1$ is

$$R^*_1 = S_1^* + 2g(t_n + 2\alpha \phi_0 k) + 8\phi_0 \alpha \beta (R_0^n + S_0^n)\left(\frac{3}{2} S_0^n - 2 S_0^{n-1} + \frac{1}{2} S_0^{n-2}\right)$$

$$+ 2\phi_0^2 \left(\alpha^2 - \beta^2\right)\left(S_0^n - 2 S_0^{n-1} + S_0^{n-2}\right)$$

$$- 6\left(\alpha^3 \left(R^n_0 - R_0^{n-1}\right)^2 - \beta^3 \left(S_0^n - S_0^{n-1}\right)^2\right),$$

(4.11)
When the calculations shown in Figure 1 are repeated using these boundary conditions, the errors shown in Figure 3 result. Now the errors due to the boundary conditions are no larger than the errors inherent in the time-split method. Figure 4 shows the errors at time \( t = 1 \).

5. Stability. In this section we prove stability of the time-split method for the initial-boundary value problem when boundary conditions of the type derived in Sections 2 and 3 are used. Stability is proved for the general variable coefficient problem, provided the following conditions hold:

(1) The time-split method is Cauchy stable (see [5] for some general conditions under which this holds).

(2) Boundary conditions specified for inflow variables are independent of values of those variables in the interior, i.e., they depend only on outflow variables and the given boundary data \( g(t) \) (which we will assume is bounded in some appropriate Sobolev norm).

Condition 2 is satisfied by the boundary conditions derived in Section 3, e.g., (3.7). Note that for a pure inflow problem, this condition means that the boundary conditions must be completely independent of the interior solution, as are, for example, the conditions (2.8) and (2.11). For such problems, stability of the initial-boundary value problem is easily proved directly from Cauchy stability. This result will be shown first and then used together with the theory of Gustafsson, Kreiss and Sundström [3] to prove stability at an inflow-outflow boundary.

Stability of the time-split method at an inflow boundary can be proved using the following general theorem (having nothing to do with splittings), which states that any Cauchy stable scheme is also stable for the initial-boundary value problem, provided that the specified boundary data \( \{U^n_m\}_{m=0}^p \) is independent of the interior solution.

**Theorem 5.1.** Suppose \( Q(k) \) is Cauchy stable. For the initial-boundary value problem, define \( U^{n+1} \) by

\[
U^{n+1}_m = \begin{cases} 
Q(k)U^n_m, & m > p, \\
G^{n+1}_m, & m = 0, 1, \ldots, p.
\end{cases}
\]

Then the approximation is stable in the sense that

\[
\|U^n\|^2 \leq K_T \|U^0\|^2 + \tilde{K}_T \|G\|^2
\]

for \( nk \leq T, k < k_0 \).

where \( K_T \) and \( \tilde{K}_T \) are constants depending only on \( T \).

Here the following norms are used:

\[
\|U^n\|^2 = h \sum_{m=0}^\infty |U^n_m|^2, \quad \|G\|^2 = k \sum_{q=1}^T \sum_{j=0}^p |G^j_q|^2,
\]

where \( |\cdot| \) is the vector norm given below in (5.3).

**Proof.** By the Cauchy stability of \( Q \), there exists a constant \( \alpha \) and a norm \( \|\cdot\| \), equivalent to the \( l^2 \)-norm, such that

\[
\|Q(k)\|^2 \leq 1 + \alpha k \quad \text{for} \quad k < k_0.
\]
The norm $\| \cdot \|$ is given by
\[
\| U^n \|^2 = h \sum_{m=-\infty}^{\infty} |U_m^n|^2
\]
with $| \cdot |$ a vector norm equivalent to the 2-norm, i.e.,
\[
(5.3) \quad |U_m^n| = |SU_m^n|_2
\]
for some nonsingular matrix $S$. (See, e.g., Chapter 4 of [6].)

Extend the given initial data $\{U_m^0\}_{m=0}^{\infty}$ to all $m$ by setting $U_m^0 = 0$, $m = -1$, $-2, \ldots$. Then, solving the quarter-plane problem is equivalent to solving the Cauchy problem and then redefining $\{U_j^n\}_{j=0}^{\infty}$ at each step. Specifically, we set
\[
\tilde{U}_m^{n+1} = Q(k)U_m^n, \quad m = 0, \pm 1, \pm 2, \ldots,
\]
and then take
\[
U_m^{n+1} = \begin{cases} 
G_m^{n+1}, & m = 0, 1, \ldots, p, \\
\tilde{U}_m^{n+1}, & \text{otherwise.}
\end{cases}
\]
(5.4)
The resulting $\{U_m^n\}_{m=0}^{\infty}$ constitute the solution of the quarter-plane problem.

By (5.2), we have
\[
(5.5) \quad \| \tilde{U}^{n+1} \|^2 \leq (1 + ak)\| U^n \|^2.
\]
By (5.4), we obtain the following bound for $U^{n+1}$:
\[
(5.6) \quad \| U^{n+1} \|^2 \leq \| \tilde{U}^{n+1} \|^2 + \| G^{n+1} \|^2,
\]
where $\| G^{n+1} \|^2 = h \sum_{j=0}^{p} |G_j^{p+1}|^2$.

Combining (5.5) and (5.6) gives
\[
\| U^{n+1} \|^2 \leq (1 + ak)\| U^n \|^2 + \| G^{n+1} \|^2,
\]
so that by induction we obtain
\[
\| U^n \|^2 \leq (1 + ak)^n\| U^0 \|^2 + \sum_{q=0}^{n-1} (1 + ak)^q \| G^{n-q} \|^2
\]
\[
\leq e^{\alpha T}\left(\| U^0 \|^2 + \sum_{q=0}^{n-1} \| G^{n-q} \|^2\right)
\]
for $nk \leq T$. Since $\| U^n \|^2 \leq \| U^n \|^2$, $\| U^0 \|^2 = \| U^0 \|^2$,
and
\[
\sum_{q=0}^{n-1} \| G^{n-q} \|^2 = h \sum_{q=0}^{n-1} \sum_{j=0}^{p} |G_j^q|^2 \leq \frac{h}{k} \| G \|_i^2
\]
for $nk \leq T$, we obtain the desired bound (5.1) with $\bar{K}_T = e^{\alpha T}$ and $\bar{k}_T = e^{\alpha T}h/k$. □

To see how this theorem applies to a time-split method, consider the method
\[
(5.7) \quad U_{m+n+1}^* = Q_f(k)U_m^n, \quad U_{m+n+1} = Q_s(k)U_{m+n+1}^*.
\]
This splitting is in general only first-order accurate but has the advantage that only a single intermediate solution $U^*$ is introduced. The technique used to prove stability for this splitting readily extends to methods in which additional intermediate solutions are present.
We assume that the split method is stable on the Cauchy problem, i.e., that 
\( Q_s(k)Q_f(k) \) is a stable operator (see [5]).

Suppose that the boundary data are of the form
\[
\begin{align*}
U_{j}^{*n+1} &= G_{j}^{*n+1}, \quad j = 0,1,\ldots, p, \\
U_{j}^{n+1} &= G_{j}^{n+1}, \quad j = 0,1,\ldots, p.
\end{align*}
\]

For convenience we have assumed that the same number of boundary conditions are
needed for both \( U^{*n+1} \) and \( U^{n+1} \), but this is not essential. The quantities \( G_{j}^{*n+1} \) and 
\( G_{j}^{n+1} \) are determined as in Section 2 in terms of the given boundary function \( g(t) \) 
and some of its derivatives (say, \( d \) derivatives). Suppose that the corresponding 
Sobolev norm of \( G(t) \) is bounded by some constant \( \gamma \), uniformly in \( k \) and \( h \),
\[
\|g\|_{d}^{2} = \sum_{j=0}^{d} \|g^{(j)}\|_{r}^{2} < \gamma.
\]

Then, we have
\[
\begin{align*}
\|G^{*}\|_{r}^{2} &\leq K_{1}, \\
\|G\|_{r}^{2} &\leq K_{2},
\end{align*}
\]
for some constants \( K_{1} \) and \( K_{2} \).

In order to apply Theorem 5.1, we rewrite \( (5.7) \) as
\[
\begin{bmatrix}
I \\
-Q_{s}(k) \\
I
\end{bmatrix}
\begin{bmatrix}
U^{*} \\
U \\
U
\end{bmatrix}_{m}^{n+1} = 
\begin{bmatrix}
0 & Q_{f}(k) \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
U^{*} \\
U
\end{bmatrix}_{m}^{n}
\]
to obtain a Cauchy stable scheme for the “super-vector” \((U^{*},U)^{T}\). Note that the 
method is formally implicit even if the original method was explicit, as it must be, 
since the boundary conditions specified for \( U^{*n+1} \) affect the computation of \( U^{n+1} \).
The Cauchy stability of \( (5.10) \) follows from the Cauchy stability of \( Q_{s}(k)Q_{f}(k) \),
which gives \( \|U^{n}\| \leq C\|U^{0}\| \), together with
\[
\|U^{*n}\| = \|Q_{f}(k)U^{n}\| \leq C_{1}\|U^{0}\|,
\]
where \( C_{1} = C\|Q_{f}(k)\| \). Using Theorem 5.1 and the bounds \( (5.9) \) we find that \( (5.10) \)
is stable for the initial-boundary value problem and that, in particular,
\[
\|U^{n}\|^{2} \leq K_{1}\|U^{0}\|^{2} + \tilde{K}_{1}(K_{1} + K_{2})\gamma.
\]

We now turn to inflow-outflow problems with boundary conditions of the form 
discussed in Section 3. As above, the time-split nature of the scheme can be handled 
by introducing super-vectors. Hence we will only discuss the stability of a general 
one-step scheme in which the inflow variables \( V \) and the outflow variables \( W \) are 
coupled only through the boundary conditions. As usual, we assume Cauchy stabil-
ity. Our discussion will be rather brief but similar arguments can be found in 
Goldberg and Tadmor [1], [2].

The scheme for \( W \) is independent of \( V \) and we will assume, as we did in Section 3, 
that the time-split method yields a one-sided scheme for \( W \), so that no boundary 
conditions need be specified. Then, from Cauchy stability, we clearly have \( \|W^{n}\|^{2} \leq 
\|W^{0}\|^{2} \), since the introduction of the boundary does not affect the computation of
Moreover, such a scheme for $W$ is also stable in the sense of Definition 3.3 of Gustafsson, Kreiss and Sundström [3] (we refer to this as GKS-stability). This stability condition also requires bounds on a norm of $W$ along the boundary. The GKS-stability follows easily from the theory of [3] for a one-sided scheme.

GKS-stability of the outflow problem is just what we need to prove stability of the inflow problem. By assumption, the boundary conditions for $V$ depend only on $g(t)$ and on values of $W$ along the boundary, and can be bounded in terms of $\|g\|_\infty$ and $\|W\|$. The former of these is assumed to be uniformly bounded, while the latter is bounded by the GKS-stability of $W$. Theorem 5.1 thus applies to the inflow problem, and hence, the entire approximation is stable on the initial-boundary value problem.

These stability results are supported by large-time numerical calculations for a wide variety of examples, including the boundary conditions of Section 4 for the shallow water equations.

Acknowledgment. Much of this work is contained in the author’s Ph.D. dissertation, completed at Stanford University under the direction of Joseph Oliger, to whom many thanks are due.

Department of Mathematics
University of California
Los Angeles, California 90024