WELL-POSED BOUNDARY CONDITIONS FOR SEMI-LAGRANGIAN SCHEMES; THE ONE-DIMENSIONAL CASE, PART II.

A McDonald
Met Éireann
Dublin, Ireland.

1. INTRODUCTION

The HIRLAM group has set itself the goal of building a numerical weather prediction model with well-posed lateral boundary conditions. As a preliminary step toward achieving that goal McDonald (1999), called M1 from here on, examined some of the technical problems associated with the 'multiply-upstream well-posedness' problem. That is, how do we invent well-posed boundary conditions when using semi-Lagrangian semi-implicit discretizations? This question was addressed for two one-dimensional systems: the advection equation, and the advection/adjustment equation. Three possible answers were considered, 'trajectory truncation', 'time interpolation', and a 'well-posed buffer zone'.

This companion paper has four purposes. (a) The first is to explore the complications for well-posedness introduced by adding the Coriolis terms to the advection/adjustment equation. This causes non-trivial complications which are examined in sections 2 and 3. (b) In M1 we saw that the straightforward implementation of the 'time interpolation scheme' resulted in a cumbersome system of equations. In section 4 we present a more attractive realization of the scheme and show that it gives stable and accurate integrations. (c) In M1 and in sections 3 and 4 of this paper we demonstrate stability experimentally. To complement this, in section 5 we make a first attempt at establishing stability theoretically. We do so by applying the energy method to the one-dimensional advection equation for our three postulated solutions for the multiply-upstream well-posedness problem. (d) lastly, section 6 contains a discussion of the strengths and weaknesses of these different approaches to the boundary value problem.

Naive extrapolations from the interior to the boundary are risky, see section 6 of Gustafsson, Kreiss and Sundström, (1972). Therefore, as a design strategy, they are introduced only when unavoidable in what follows.

2. THE ADVECTION-ADJUSTMENT EQUATION WITH ROTATION.

In this section we consider the complications caused by adding 'Coriolis terms' to the one dimensional advection-adjustment equations. Also we describe the 'slow' and 'fast' experiments we will be using to test our proposed discretization schemes.

2.1 The equations and their discretization.

Including the Coriolis terms leads to the equations

$$\frac{\partial \phi(x,t)}{\partial t} + \bar{u}\frac{\partial \phi(x,t)}{\partial x} + \bar{\phi}\frac{\partial u(x,t)}{\partial x} = 0, \tag{2.1.1}$$

$$\frac{\partial u(x,t)}{\partial t} + \bar{u}\frac{\partial u(x,t)}{\partial x} + \frac{\partial \phi(x,t)}{\partial x} - \bar{f}v(x,t) = 0, \tag{2.1.2}$$

$$\frac{\partial v(x,t)}{\partial t} + \bar{u}\frac{\partial v(x,t)}{\partial x} + \bar{f}u(x,t) = 0, \tag{2.1.3}$$

where the advection velocity $\bar{u}$, the Coriolis parameter $\bar{f}$, and the mean geopotential height $\bar{\phi}$, are constants.

The characteristic variables of these equations (and their associated characteristics) are

$$v,\ (\bar{u}); \qquad p = \phi + \bar{c}u,\ (\bar{u}+\bar{c}); \qquad q = \phi - \bar{c}u,\ (\bar{u}-\bar{c}), \tag{2.1.4}$$

where $\bar{c} = \sqrt{\bar{\phi}}$; see, for example, Sundström and Elvius, 1979.

Substituting $\phi = \hat{\phi}\exp i(kx - \omega t)$ yields three solutions; a 'slow' solution with $\omega = k\bar{u}$ and two 'fast' solutions with $\omega = k(\bar{u} \pm c_k)$, where

$$c_k = \sqrt{\bar{\phi} + \bar{f}^2/k^2}. \tag{2.1.5}$$

For the latter the following ϕ, u and v satisfy Eqs. (2.1.1 - 2.1.3):

$$\phi(x,t) = A_+ cos\left(\frac{2\pi}{L_x}\{x - (\bar{u}+c_k)t\} + \tau_+\right) + A_- cos\left(\frac{2\pi}{L_x}\{x - (\bar{u}-c_k)t\} + \tau_-\right) \tag{2.1.6}$$

$$u(x,t) = \frac{c_k}{\bar{\phi}}\left[A_+ cos\left(\frac{2\pi}{L_x}\{x - (\bar{u}+c_k)t\} + \tau_+\right) - A_- cos\left(\frac{2\pi}{L_x}\{x - (\bar{u}-c_k)t\} + \tau_-\right)\right] \tag{2.1.7}$$

$$v(x,t) = \frac{\bar{f}}{k\bar{\phi}}\left[A_+ sin\left(\frac{2\pi}{L_x}\{x - (\bar{u}+c_k)t\} + \tau_+\right) + A_- sin\left(\frac{2\pi}{L_x}\{x - (\bar{u}-c_k)t\} + \tau_-\right)\right] \tag{2.1.8}$$

A_+, A_-, τ_+ and τ_- are constants; $L_x = 2\pi/k$ is the wavelength.

		0									L	
ϕ_{-1}	$u_{-\frac{1}{2}}$	ϕ_0	$u_{\frac{1}{2}}$	ϕ_1	$u_{\frac{3}{2}}$	.....	ϕ_{i_L-2}	$u_{i_L-\frac{3}{2}}$	ϕ_{i_L-1}	$u_{i_L-\frac{1}{2}}$	ϕ_{i_L}	$u_{i_L+\frac{1}{2}}$
v_{-1}		v_0		v_1			v_{i_L-2}		v_{i_L-1}		v_{i_L}	

FIGURE 2.1 Distribution of variables on the x-axis using the 'C-grid'.

Let us now perform a two time level SLSI discretization on Eqs. (2.1.1 - 2.1.3) using 'C-grid' staggering of the fields (see figure 2.1). The range of validity of these equations is given in table 1.

$$\phi_i^{n+1} + \frac{\bar{\phi}\Delta t}{2\Delta x}\left(u_{i+\frac{1}{2}}^{n+1} - u_{i-\frac{1}{2}}^{n+1}\right) = R_i^{\phi} = (X^{\phi})_{i,*}^n, \tag{2.1.9}$$

$$u_{i+\frac{1}{2}}^{n+1} + \frac{\Delta t}{2\Delta x}\left(\phi_{i+1}^{n+1} - \phi_i^{n+1}\right) - \frac{\bar{f}\Delta t}{4}\left(v_{i+1}^{n+1} + v_i^{n+1}\right) = R_{i+\frac{1}{2}}^{U} = (X^U)_{i+\frac{1}{2},*}^n, \tag{2.1.10}$$

$$v_i^{n+1} + \frac{\bar{f}\Delta t}{4}\left(u_{i+\frac{1}{2}}^{n+1} + u_{i-\frac{1}{2}}^{n+1}\right) = R_i^{v} = (X^v)_{i,*}^n, \tag{2.1.11}$$

where

$$(X^{\phi})_i^n = \phi_i^n - \frac{\bar{\phi}\Delta t}{2\Delta x}\left(u_{i+\frac{1}{2}}^n - u_{i-\frac{1}{2}}^n\right) \tag{2.1.12}$$

$$(X^U)_{i+\frac{1}{2}}^n = u_{i+\frac{1}{2}}^n - \frac{\Delta t}{2\Delta x}\left(\phi_{i+1}^n - \phi_i^n\right) + \frac{\bar{f}\Delta t}{4}\left(v_{i+1}^n + v_i^n\right) \tag{2.1.13}$$

$$(X^v)_i^n = v_i^n - \frac{\bar{f}\Delta t}{4}\left(u_{i+\frac{1}{2}}^n + u_{i-\frac{1}{2}}^n\right), \tag{2.1.14}$$

where $X_{i,*}^n$ means interpolate X to the departure point associated with the grid point i. (I am using the superscript capital U in Eq. (2.1.10) only to distinguish X^U and R^U from their counterparts in M1).

Boundary	Eq. (2.1.9)	Eq. (2.1.10)	Eq. (2.1.11)	Eq. (2.1.18)
$u_{\frac{1}{2}}, u_{i_L+\frac{1}{2}}$	$1 \to i_L$	$1 \to i_L - 1$	$1 \to i_L$	$2 \to i_L - 1$
ϕ_0, ϕ_{i_L}	$1 \to i_L - 1$	$0 \to i_L - 1$	$1 \to i_L - 1$	$1 \to i_L - 1$
p_0, q_{i_L}	$0 \to i_L$	$0 \to i_L - 1$	$0 \to i_L$	$1 \to i_L - 1$
$p_{\frac{1}{2}}, q_{i_L+\frac{1}{2}}$	$1 \to i_L$	$0 \to i_L$	$1 \to i_L$	$2 \to i_L - 1$

TABLE 1. Range of validity of equations (2.1.9)-(2.1.11) and (2.1.18).

Having computed R^{ϕ}, R^U and R^v, one approach to solving this system of equations is as follows. Eqs. (2.1.9 - 2.1.11) can be re-arranged to yield a tri-diagonal system for u^{n+1}. Having solved this, ϕ^{n+1} and v^{n+1} emerge from Eq. (2.1.9) and (2.1.11), respectively.

If we were to proceed in this way for the two-dimensional shallow water equations we would generate the doubly-averaged terms $\bar{f}\bar{\bar{u}}$ and $\bar{f}\bar{\bar{v}}$, each of which has a 9-point template.

Near the boundary these terms would further complicate the already difficult problem of inventing a well-posed discrete system of equations. With this in mind let us use the following discretization because its two-dimensional generalization will cause minimal complications on the boundary. We use an iterative approach which has the additional advantage that u and ϕ are treated exactly as in M1, allowing us to concentrate on 'v' exclusively. (Compare Eqs. 2.1.15, 2.1.16 and 2.1.18 with 3.7, 3.8, and 3.11 of M1).

$$\left[\phi_i + \frac{\bar{\phi}\Delta t}{2\Delta x}\left(u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}\right)\right]^{(k+1)} = R_i^{\phi} \tag{2.1.15}$$

$$\left[u_{i+\frac{1}{2}} + \frac{\Delta t}{2\Delta x}\left(\phi_{i+1} - \phi_i\right)\right]^{(k+1)} = R_{i+\frac{1}{2}}^{U} + \frac{\bar{f}\Delta t}{4}\left(v_{i+1} + v_i\right)^{(k)} \tag{2.1.16}$$

$$v_i^{(k+1)} = R_i^{v} - \frac{\bar{f}\Delta t}{4}\left(u_{i+\frac{1}{2}} + u_{i-\frac{1}{2}}\right)^{(k)} \tag{2.1.17}$$

Having computed R^{ϕ}, R^{U}, and R^{v}, substituting Eq. (2.1.16) in Eq. (2.1.15) results in

$$\left[\phi_i - \bar{\phi}\left(\frac{\Delta t}{2\Delta x}\right)^2\left(\phi_{i+1} - 2\phi_i + \phi_{i-1}\right)\right]^{(k+1)} =$$

$$R_i^{\phi} - \frac{\bar{\phi}\Delta t}{2\Delta x}\left(R_{i+\frac{1}{2}}^{U} - R_{i-\frac{1}{2}}^{U}\right) - \frac{\bar{\phi}\bar{f}\Delta t^2}{8\Delta x}\left(v_{i+1} + v_{i-1}\right)^{(k)} \tag{2.1.18}$$

Having solved this equation for ϕ with appropriate boundary conditions, $u_{i+\frac{1}{2}}^{(k+1)}$ can be computed from Eq. (2.1.16), and $v_i^{(k+1)}$ from Eq. (2.1.17). The procedure can be iterated until sufficient accuracy is attained.

2.2 Numerical testing.

We will perform numerical tests of the discretized system, Eqs. (2.1.15-2.1.17), using different boundary treatments with two contrasting initial states.

(a) A fast solution. We use a wave of wavelength L as an initial state:

$$\phi(x,0) = \hat{\phi} sin\left(\frac{2\pi}{L}x\right); \qquad u(x,0) = \hat{\phi}\frac{c_K}{\bar{\phi}} sin\left(\frac{2\pi}{L}x\right); \qquad v(x,0) = -\hat{\phi}\frac{L\bar{f}}{2\pi\bar{\phi}} cos\left(\frac{2\pi}{L}x\right) \tag{2.2.1}$$

We choose our boundary conditions such that this wave moves in the positive x direction with a velocity $\bar{u} + c_K$, where c_K is given by Eq. (2.1.5) with $k = 2\pi/L$.

(b) A slow solution. For this solution $u = 0$, and $\bar{f}v = \partial\phi/\partial x$, and the analytical solution transports these fields with a velocity $\bar{u}$ without changing their shape.

With this solution we mirror what we would be attempting with the full three dimensional system of equations in an operational context. There, the analysis and initialization

produces a well-balanced initial state; the slow initial state mirrors this. On the boundary we want the meteorological fields to enter the domain without corruption from the gravity waves; here we model this by imposing the slow solution on the boundary.

For ϕ we use the same 'bell curve' as in section 2 of M1; thus the initial state is

$$\phi(x,0) = \hat{\phi} exp\Big[-\Big\{\frac{(x-x_s)}{(L/10)}\Big\}^2\Big] \; ; \quad v(x,0) = \frac{-2(x-x_s)}{\bar{f}(L/10)^2}\phi(x,0) \; ; \quad u(x,0) = 0. \quad (2.2.2)$$

To mirror the meteorological situation of a 'front' entering the area through the western boundary and exiting through the eastern boundary we will begin the integration placing the 'bell-curve' ϕ-solution with its maximum on the western boundary. We will then impose boundary conditions consistent with the 'bell-curve' moving in the positive x-direction with a velocity $\bar{u}$. See figure 4.1 for the analytical initial and final states.

We use $L = 1000km, \Delta x = 10$km. In addition, $\bar{f} = 10^{-4}$. In the tests which follow the length of the integration (T) will always be chosen such that in the analytical solution the wave moves *exactly* a distance L.

Thus, for the fast solution the initial and final state should overlap identically. This makes any errors 'jump out'. Also, for the fast solution we plot $-L\bar{\phi}v/(2\pi\bar{f})$, making its amplitude the same size as the other fields.

3. WELL POSED BOUNDARIES FOR v ON THE STAGGERED GRID.

For each inward pointing characteristic a field must be imposed on the boundary; see, for example, Sundström and Elvius, 1979. Thus, looking at Eq. (2.1.4) (we only consider the case $\bar{c} > \bar{u}$ in this paper), we must impose two fields at $x = 0$ and one at $x = L$ when $\bar{u} > 0$. We must impose two fields at $x = L$ and one at $x = 0$ when $\bar{u} < 0$. Let us always impose v at inflow, and consider two alternatives for u and ϕ.

3.1 Imposing the boundary conditions on the u fields.

Let us first consider imposing our boundary values on $u_{\frac{1}{2}}$ and $u_{i_L+\frac{1}{2}}$. We called this *Alternative 3* in M1. This seems the simplest option as far as the field v is concerned. At outflow no special treatment is required; v_1 (or v_{i_L}) is computed via Eq. (2.1.11). Also, at inflow, because of the grid staggering, v_{i_L} (or v_1) is computed via Eq. (2.1.11). See sub-section 5.3.1 of Elvius and Sundström (1973), where the Eulerian equivalent is used at odd time steps. Yet, there is a paradox. We know that at inflow we should be *imposing* v, yet v_1 and v_{i_L} are emerging quite naturally from our discretized system of equations. We will examine this puzzle in this sub-section.

For the u and ϕ equations we proceed as in M1. The system of equations (2.1.18) is now

closed with the two equations

$$\phi_1^{n+1} - \bar{\phi}\Big(\frac{\Delta t}{2\Delta x}\Big)^2\Big(\phi_2 - \phi_1\Big)^{n+1} = R_1^{\phi} - \bar{\phi}\frac{\Delta t}{2\Delta x}\Big(R^U_{\frac{3}{2}} + \frac{\bar{f}\Delta t}{4}\{v_2 + v_1\}^{(k)} - u^{n+1}_{\frac{1}{2}}\Big) \quad (3.1.1)$$

$$\phi_{i_L}^{n+1} + \bar{\phi}\Big(\frac{\Delta t}{2\Delta x}\Big)^2\Big(\phi_{i_L} - \phi_{i_L-1}\Big)^{n+1} = R^{\phi}_{i_L} - \bar{\phi}\frac{\Delta t}{2\Delta x}\Big(u^{n+1}_{i_L+\frac{1}{2}} - R^u_{i_L-\frac{1}{2}} - \frac{\bar{f}\Delta t}{4}\{v_{i_L} + v_{i_L-1}\}^{(k)}\Big) \quad (3.1.2)$$

For our initial testing with $|\alpha| < 1$ let us use trajectory truncation. Thus $R^U = (X^U)^n_{\frac{3}{2}}$ when $x_* < 3\Delta x/2$; $R^U = (X^U)^n_{i_L-\frac{1}{2}}$ when $x_* > L - \Delta x/2$. $R^{\phi} = (X^{\phi})^n_1$ when $x_* < \Delta x$; $R^{\phi} = (X^{\phi})^n_{i_L}$ when $x_* > L$. $R^v = (X^v)^n_1$ when $x_* < \Delta x$; $R^v = (X^v)^n_{i_L}$ when $x_* > L$.

For the 'fast' solutions the following are the appropriate boundary conditions for u when our initial state is given by Eq. (2.2.1):

$$u(\frac{\Delta x}{2}, t) = \frac{c_K}{\bar{\phi}}\hat{\phi} sin\Big(\frac{2\pi}{L}\{\frac{\Delta x}{2} - (\bar{u} + c_K)t\}\Big); \quad (3.1.3)$$

$$u(L + \frac{\Delta x}{2}, t) = \frac{c_K}{\bar{\phi}}\hat{\phi} sin\Big(\frac{2\pi}{L}\{L + \frac{\Delta x}{2} - (\bar{u} + c_K)t\}\Big) : \quad (3.1.4)$$

The question we would like to address is whether we can use just equations (2.1.9 - 2.1.11), deriving v at inflow from Eq. (2.1.11), without adding something extra to impose v at inflow. To test this let us use the following: $\Delta t = 49.94s, c_K = 300.4m/s, \bar{u} = 100m/s, \alpha = 0.4994$. The outcome is shown in figure 3.1. The fields p and q are well-behaved. (Compare with figure 3.2 of M1). The field v, although stable, has a large error.

Let us see if imposing v at inflow cures this error. There are two obvious ways to do this.

(a) Use of a *staggered boundary* allows us to avoid possibly hazardous extrapolations. To do this over-write v_1 when $\bar{u} > 0$ and v_{i_L} when $\bar{u} < 0$ with the following:

$$\bar{u} > 0 \ : \qquad v(\Delta x, t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi} cos\Big(\frac{2\pi}{L}\{\Delta x - (\bar{u} + c_K)t\}\Big), \quad (3.1.5)$$

$$\bar{u} < 0 \ : \qquad v(L, t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi} cos\Big(\frac{2\pi}{L}\{L - (\bar{u} + c_K)t\}\Big). \quad (3.1.6)$$

Repeating the experiment shows this works well. See figure 3.2. The field v is now stably and accurately predicted.

(b) Do not stagger the boundary. Instead, impose v at the u-points, that is,

$$\bar{u} > 0 \ : \qquad v(\frac{\Delta x}{2}, t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi} cos\Big(\frac{2\pi}{L}\{\frac{\Delta x}{2} - (\bar{u} + c_K)t\}\Big), \quad (3.1.7)$$

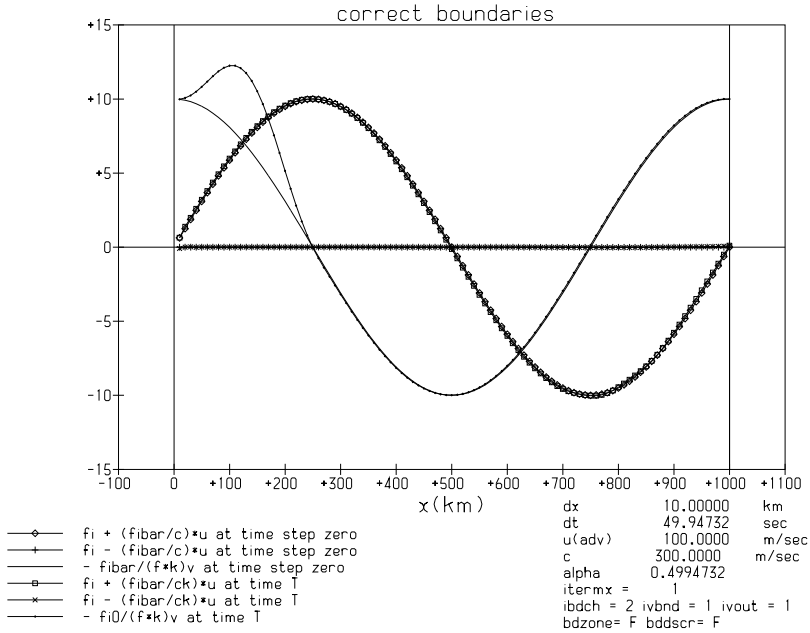

FIGURE 3.1. Graph of the solution of Eqs. (2.1.15-2.1.17) with initial conditions given by Eq. (2.2.1) and u imposed on the boundaries, Eqs. (3.1.3-3.1.4). At time zero p is shown by the diamonds, q by the '+'s, and v by the symbol-free line. These also represent the analytical solution at time T. The result of integrating with $\Delta t = 49.94s, \bar{c} = 300m/s, \bar{u} = 100m/s, \alpha = 0.4994$ is displayed as squares for p, 'x's for q, and dots for v.

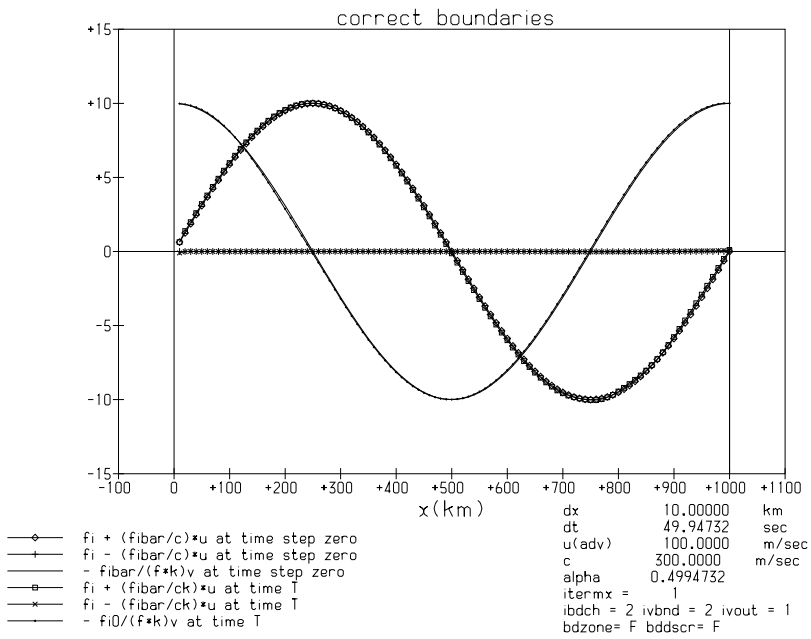

FIGURE 3.2. Same as figure 3.1, but now imposing v using Eq. (3.1.5).

$$\bar{u} < 0 \ : \qquad v(L + \frac{\Delta x}{2}, t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi}cos\Big(\frac{2\pi}{L}\{L + \frac{\Delta x}{2} - (\bar{u} + c_K)t\}\Big), \tag{3.1.8}$$

and re-arrange the equations (2.1.9-2.1.14) so that the system 'knows' about this information. Probably the easiest way to do this is to truncate the trajectory of the v-field at $x = 0$ and $x = L + \Delta x$ and to hope that the extrapolations required to compute X_0^v and $X_{i_L+1}^v$, that is,

$$\bar{u} > 0 \ : \qquad (X^v)_0^n = 2v_{\frac{1}{2}}^n - v_1^n - \frac{\bar{f}\Delta t}{4}\Big(3u_{\frac{1}{2}}^n - u_{\frac{3}{2}}^n\Big), \tag{3.1.9}$$

and

$$\bar{u} < 0 \ : \qquad (X^v)_{i_L+1}^n = 2v_{i_L+\frac{1}{2}}^n - v_{i_L}^n - \frac{\bar{f}\Delta t}{4}\Big(3u_{i_L+\frac{1}{2}}^n - u_{i_L-\frac{1}{2}}^n\Big) \tag{3.1.10}$$

will not introduce an instability. Repeating our test experiment we get a stable and accurate forecast, almost identical to figure 3.2. Thus the extrapolations in Eqs. (3.1.9) and (3.1.10) are stable.

3.2 Imposing the boundary conditions on the ϕ fields.

If we choose to impose our boundary values on ϕ_0 and ϕ_{i_L}, (*Alternative 1.* in M1), they fit naturally into Eq. (2.1.18), as does the v-boundary condition at inflow. There is a complication: we need to know v on the outflow boundary line. In this sub-section we test methods for doing this. We will also see that v at inflow is problematical.

We will test two schemes for extrapolating v to the outflow boundary point.

(i) *space and time extrapolation* ; see, for example, Elvius and Sundström (1973).

$$\bar{u} < 0 : \qquad v_0^{n+1} = 2v_1^n - v_2^{n-1}, \tag{3.2.1}$$

$$\bar{u} > 0 : \qquad v_{i_L}^{n+1} = 2v_{i_L-1}^n - v_{i_L-2}^{n-1}. \tag{3.2.2}$$

(ii) *Semi-Lagrangian extrapolation.* Use the upstream nature of the semi-Lagrangian scheme to compute v. That is, assume an approximation to Eq. (2.1.3) is obeyed at the outflow boundary point, and extrapolate any unknown fields from the interior:

$$\bar{u} < 0 : \qquad v_0^{n+1} + \frac{\bar{f}\Delta t}{4}\Big(3u_{\frac{1}{2}}^{n+1} - u_{\frac{3}{2}}^{n+1}\Big) = R_0^v, \tag{3.2.3}$$

$$\bar{u} > 0 : \qquad v_{i_L}^{n+1} + \frac{\bar{f}\Delta t}{4}\Big(3u_{i_L-\frac{1}{2}}^{n+1} - u_{i_L-\frac{3}{2}}^{n+1}\Big) = R_{i_L}^v. \tag{3.2.4}$$

For $|\alpha| < 2$ we will need $(X^v)_0^n$ and $(X^v)_{i_L}^n$ for the computation of R^v. For consistency we define them as

$$(X^v)_0^n = v_0^n - \frac{\bar{f}\Delta t}{4}\Big(3u_{\frac{1}{2}}^n - u_{\frac{3}{2}}^n\Big) \tag{3.2.5}$$

$$(X^v)^n_{i_L} = v^n_{i_L} - \frac{\bar{f}\Delta t}{4}\Big(3u^n_{i_L-\frac{1}{2}} - u^n_{i_L-\frac{3}{2}}\Big) \tag{3.2.6}$$

As in section 3.1, let us use trajectory truncation. Thus, $R^U = (X^U)^n_{\frac{1}{2}}$ when $x_* < \Delta x/2$; $R^U = (X^U)^n_{i_L-\frac{1}{2}}$ when $x_* > L - \Delta x/2$. $R^\phi = (X^\phi)^n_1$ when $x_* < \Delta x$; $R^\phi = (X^\phi)^n_{i_L-1}$ when $x_* > L - \Delta x$. When we are computing v at outflow using (i) above then $R^v = (X^v)^n_1$ when $x_* < \Delta x$; $R^v = (X^v)^n_{i_L-1}$ when $x_* > L - \Delta x$. When we are computing v at outflow using (ii) above then $R^v = (X^v)^n_0$ when $x_* < 0$; $R^v = (X^v)^n_{i_L}$ when $x_* > L$.

For the *fast solution* the boundary conditions for ϕ and v are

$$\phi(0,t) = \hat{\phi} sin\Big(\frac{2\pi}{L}\{0 - (\bar{u} + c_K)t\}\Big); \qquad \phi(L,t) = \hat{\phi} sin\Big(\frac{2\pi}{L}\{L - (\bar{u} + c_K)t\}\Big) : \tag{3.2.7}$$

$$\bar{u} > 0 \ : \qquad v(0,t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi} cos\Big(\frac{2\pi}{L}\{0 - (\bar{u} + c_K)t\}\Big), \tag{3.2.8}$$

$$\bar{u} < 0 \ : \qquad v(L,t) = -\frac{L\bar{f}}{2\pi\bar{\phi}}\hat{\phi} cos\Big(\frac{2\pi}{L}\{L - (\bar{u} + c_K)t\}\Big). \tag{3.2.9}$$

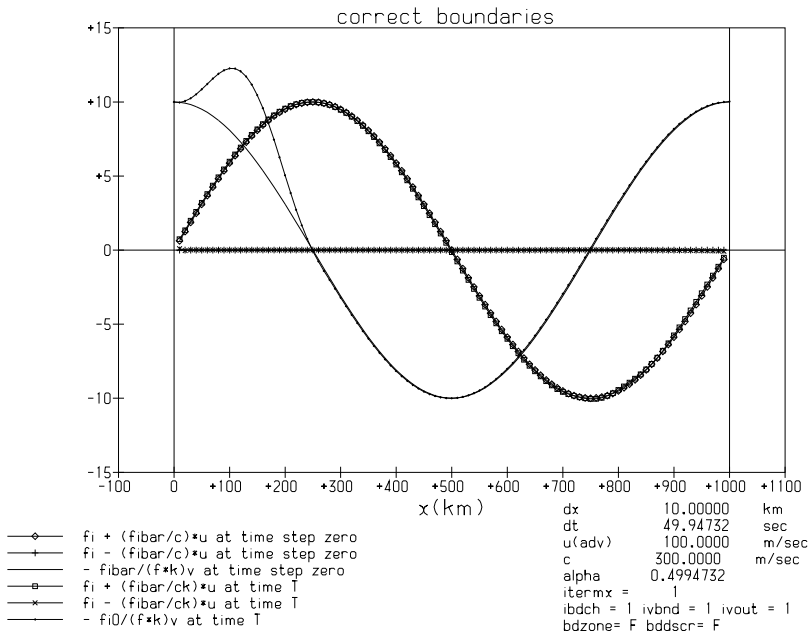

FIGURE 3.3. Graph of the solution of Eqs. (2.1.15-2.1.17) with initial conditions given by Eq. (2.2.1) and ϕ imposed on the boundaries, Eqs. (3.2.7-3.2.8). At time zero p is shown by the diamonds, q by the '+'s, and v by the symbol-free line. These also represent the analytical solution at time T. The result of integrating with $\Delta t = 49.94s, \bar{c} = 300m/s, \bar{u} = 100m/s, \alpha = 0.4994$ is displayed as squares for p, 'x's for q, and dots for v.

Let us first test extrapolation (i), using the same parameters as in section 3.1: $\Delta t = 49.94s, c_K = 300.4m/s, \bar{u} = 100m/s, \alpha = 0.4994$. As can be seen from figure 3.3, the extrapolation at outflow is stable and accurate, but there is a problem at inflow.

The v solution is so similar to that in figure 3.1 that we must suspect that the system still 'does not know about' v at inflow. Looking at equations (2.1.9-2.1.14) we see that the boundary value of v is always multiplied by $(\bar{f}\Delta t/4)$, a small number. This may explain its lack of influence on the solution. To overcome this let us attempt to make sure that boundary value of v has an $O(1)$ influence on the system. To do this we must again resort to extrapolation. Let us change the truncation of the trajectory associated with v so that $R^v = (X^v)_0^n$ when $x_* < 0$ and $R^v = (X^v)_{i_L}^n$ when $x_* > L$. To accommodate this we must again define X_0^v and $X_{i_L}^v$ using extrapolations; see Eqs. (3.2.5) and (3.2.6). Repeating our standard test shows this to be a stable and accurate solution to this problem. The figure, not shown, is almost identical to figure 3.2.

For the experiments described above in this sub-section *space and time extrapolation* was used to compute v at the outflow boundary point. *Semi-Lagrangian extrapolation* was also tested. For the latter the forecast is indistinguishable from figure 3.2 and hence is not displayed.

4. TIME INTERPOLATION RE-VISITED.

Recall that in section 3.2.2 of M1 our attempt to implement time interpolation on the boundary resulted in a system of equations which looked so expensive computationally that we did not pursue that option further. In this section we re-examine 'time interpolation', using the iterative philosophy to avoid the massive complications we saw there.

In order to put a severe strain on the system let us use the following parameters in the experimental tests described in sections 4.2 and 4.3 below: $\Delta t = 416.7s, \bar{c} = 300m/s, \bar{u} = 100m/s, \alpha = 4.167$.

4.1 The equations to be solved adjacent to the boundary.

Following the procedure of section 3.2.2 of M1 we arrive at the following equations which are to be used when the departure point is outside the integration area.

$$\left\{\phi + \frac{\mu}{\alpha}\frac{\bar{\phi}\Delta t}{2}\frac{\partial u}{\partial x}\right\}_{b+\mu}^{n+1} = \left\{\phi - \frac{\mu}{\alpha}\frac{\bar{\phi}\Delta t}{2}\frac{\partial u}{\partial x}\right\}_{b}^{n+1-\frac{\mu}{\alpha}} \equiv \left\{\hat{X}^{\phi}(\frac{\mu}{\alpha})\right\}_{b}^{n+1-\frac{\mu}{\alpha}}, \tag{4.1.1}$$

$$\left\{u + \frac{\nu}{\alpha}\frac{\Delta t}{2}\left(\frac{\partial \phi}{\partial x} - \bar{f}v\right)\right\}_{b+\nu}^{n+1} = \left\{u - \frac{\nu}{\alpha}\frac{\Delta t}{2}\left(\frac{\partial \phi}{\partial x} - \bar{f}v\right)\right\}_{b}^{n+1-\frac{\nu}{\alpha}} \equiv \left\{\hat{X}^{U}(\frac{\nu}{\alpha})\right\}_{b}^{n+1-\frac{\nu}{\alpha}} \tag{4.1.2}$$

$$\left\{v + \frac{\mu}{\alpha}\frac{\bar{f}\Delta t}{2}u\right\}_{b+\mu}^{n+1} = \left\{v - \frac{\mu}{\alpha}\frac{\bar{f}\Delta t}{2}u\right\}_{b}^{n+1-\frac{\mu}{\alpha}} \equiv \left\{\hat{X}^{v}(\frac{\mu}{\alpha})\right\}_{b}^{n+1-\frac{\mu}{\alpha}}, \tag{4.1.3}$$

Here b represents a boundary line, and μ is the integer number of points the ϕ-point of interest is from the boundary. Also, ν is a half odd-integer which measures the distance of the u-point of interest from the boundary.

Proceeding as in M1, but now moving the troublesome terms to the right hand side of the iterative system we arrive at

$$\left\{\phi + \bar{\phi}\frac{\mu\Delta t}{2\alpha}\frac{\partial u}{\partial x}\right\}_{b+\mu}^{(k+1)} = \hat{R}_i^{\phi} \equiv$$
$$C_+\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{\phi}(\frac{\mu}{\alpha})\right\}_b^{(k)} + C_0\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{\phi}(\frac{\mu}{\alpha})\right\}_b^{n} + C_-\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{\phi}(\frac{\mu}{\alpha})\right\}_b^{n-1} \tag{4.1.4}$$

$$\left\{u + \frac{\nu\Delta t}{2\alpha}\frac{\partial \phi}{\partial x}\right\}_{b+\nu}^{(k+1)} = \hat{R}_i^{U} \equiv \frac{\nu}{\alpha}\frac{\bar{f}\Delta t}{2}v_{b+\nu}^{(k)} +$$
$$C_+\left(\frac{\nu}{\alpha}\right)\left\{\hat{X}^{U}(\frac{\nu}{\alpha})\right\}_b^{(k)} + C_0\left(\frac{\nu}{\alpha}\right)\left\{\hat{X}^{U}(\frac{\nu}{\alpha})\right\}_b^{n} + C_-\left(\frac{\nu}{\alpha}\right)\left\{\hat{X}^{U}(\frac{\nu}{\alpha})\right\}_b^{n-1} \tag{4.1.5}$$

$$v_{b+\mu}^{(k+1)} = -\frac{\mu}{\alpha}\frac{\bar{f}\Delta t}{2}u_{b+\mu}^{(k)} +$$
$$C_+\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{v}(\frac{\mu}{\alpha})\right\}_b^{(k)} + C_0\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{v}(\frac{\mu}{\alpha})\right\}_b^{n} + C_-\left(\frac{\mu}{\alpha}\right)\left\{\hat{X}^{v}(\frac{\mu}{\alpha})\right\}_b^{n-1} \tag{4.1.6}$$

When discretized these equations give rise to a set which is formally the same as Eqs. (2.1.15-2.1.17) and is thus very easy to program. (The tridiagonal system now has variable coefficients and there are a few extra terms on the right side to be evaluated during each iteration).

The terms $u, \partial u/\partial x$, and $\partial\phi/\partial x$ are not known on the boundary, and thus must be extrapolated. The simplest possible extrapolations give, at $x = 0$,

$$u_0 = \frac{3}{2}u_{\frac{1}{2}} - \frac{1}{2}u_{\frac{3}{2}}; \quad \left(\frac{\partial u}{\partial x}\right)_0 = \frac{1}{\Delta x}\left(u_{\frac{3}{2}} - u_{\frac{1}{2}}\right); \quad \left(\frac{\partial \phi}{\partial x}\right)_0 = \frac{1}{\Delta x}\left(\phi_1 - \phi_0\right), \tag{4.1.7}$$

and at $x = L$,

$$u_{i_L} = \frac{3}{2}u_{i_L-\frac{1}{2}} - \frac{1}{2}u_{i_L-\frac{3}{2}}; \quad \left(\frac{\partial u}{\partial x}\right)_{i_L} = \frac{1}{\Delta x}\left(u_{i_L-\frac{1}{2}} - u_{i_L-\frac{3}{2}}\right); \quad \left(\frac{\partial \phi}{\partial x}\right)_{i_L} = \frac{1}{\Delta x}\left(\phi_{i_L} - \phi_{i_L-1}\right). \tag{4.1.8}$$

4.2 Imposing the boundary conditions on the ϕ fields.

Let us use the *slow solution* for our tests in this section. Then if we impose the boundary conditions on ϕ_0 and ϕ_{i_L} and v_0 ($\bar{u} > 0$), the following describe the 'bell-curve' ϕ-solution

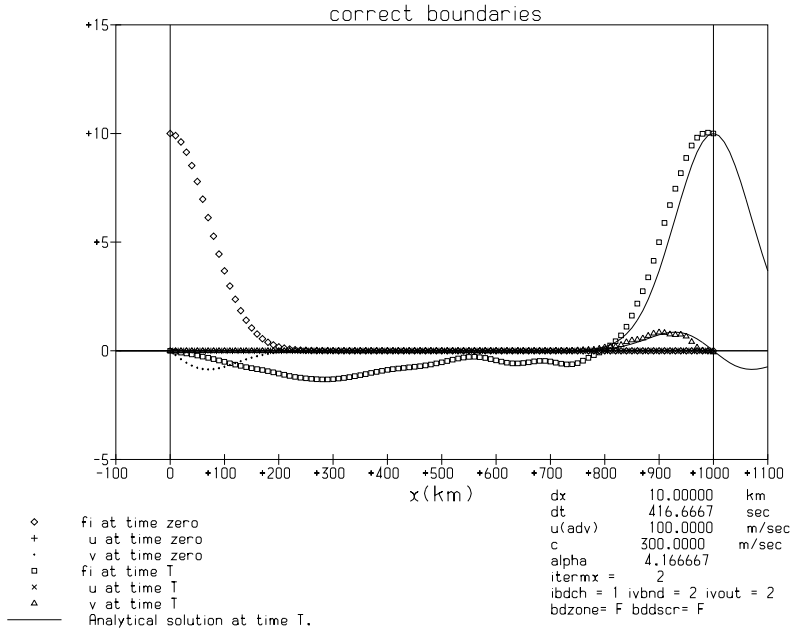

FIGURE 4.1. Graph of the solution of Eqs. (2.1.15-2.1.17) with initial conditions given by Eq. (2.2.2) and ϕ and v imposed on the boundaries, Eqs. (4.2.1-4.2.2), using the 'trajectory truncation' scheme. At time zero ϕ is shown by the diamonds, u by the '+'s, and v by the dots. The analytical solutions at time T for all three fields are shown by symbol-free lines. The result of integrating with $\Delta t = 416.667s, \bar{c} = 300m/s, \bar{u} = 100m/s, \alpha =$ 4.16667 is displayed as squares for ϕ, 'x's for u, and triangles for v.

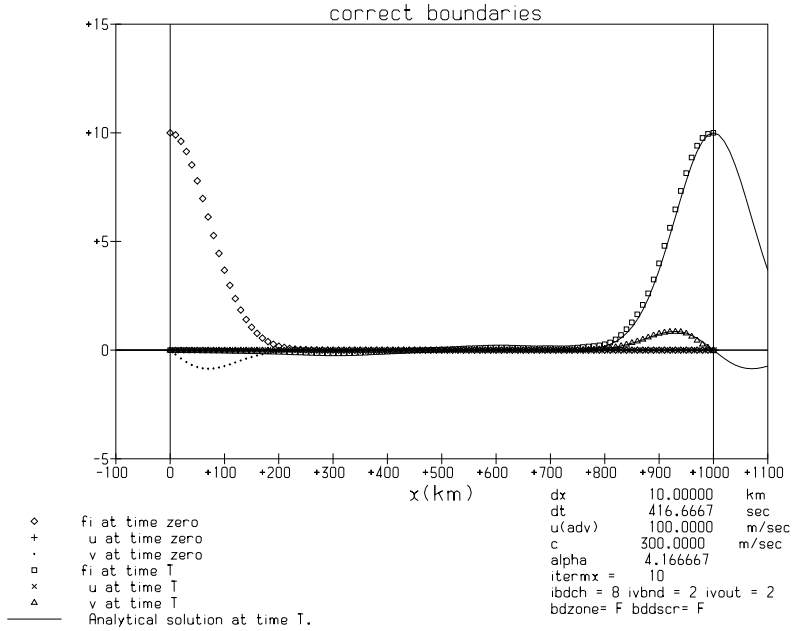

FIGURE 4.2. Same as figure 4.1, but now using 'time interpolation' on the boundary instead of 'trajectory truncation'.

with its maximum on the boundary at time zero and moving in the positive x-direction with a velocity $\bar{u}$:

$$\phi(0,t) = \hat{\phi} exp\Big[-\Big\{\frac{0-\bar{u}t}{(L/10)}\Big\}^2\Big] \; ; \quad \phi(L,t) = \hat{\phi} exp\Big[-\Big\{\frac{L-\bar{u}t}{(L/10)}\Big\}^2\Big] \; ; \tag{4.2.1}$$

$$\bar{u} > 0 \; : \qquad v(0,t) = \frac{-2(0-\bar{u}t)}{\bar{f}(L/10)^2}\phi(0,t) \; ; \tag{4.2.2}$$

For our initial state, which can be seen in figure 4.1, we use Eq. (2.2.2) with $x_s = 0$.

Now we proceed as in section 3.2. For the tests which follow I have used *semi-Lagrangian extrapolation* to compute v_{i_L}. First, to check what errors it causes for the slow solution, the forecast was run with the trajectory truncated as follows: $R^u = (X^u)^n_{\frac{1}{2}}$ when $x_* < \Delta x/2$; $R^u = (X^u)^n_{i_L-\frac{1}{2}}$ when $x_* > L - \Delta x/2$. $R^\phi = (X^\phi)^n_0$ when $x_* < 0$; $R^\phi = (X^\phi)^n_{i_L}$ when $x_* > L$. $R^v = (X^v)^n_0$ when $x_* < 0$; $R^v = (X^v)^n_{i_L}$ when $x_* > L$. The result is shown in figure 4.1. The positive aspects of this forecast are (a) that the 'front' has entered and exited the area reasonably accurately; (b) that there are no false reflections at the boundary, and (c) that there is no short-wave noise generated. The negative aspects are, of course, (a) the 'tail' which the ϕ-solution has attained, and (b) the slowing down of the ϕ and v solutions, just as we saw in section 2 of M1.

Compare this with the the forecast obtained using time interpolation on the boundary, shown in figure 4.2. The positive aspects of the solution are retained and the negative ones eliminated. The solution is almost as good as the analytical one.

4.3 Imposing the boundary conditions on the p/q fields.

As we saw in section 3.3 of M1, and as we will see below, it may be advantageous to impose boundary values on the fields corresponding to the in-going characteristics rather than ϕ. In this section we examine the implications of this for time interpolation. Not for the first time we find that the C-grid causes headaches. As previously, we consider $\bar{u} > 0$ without loss of generality.

Consider imposing p_0 and q_{i_L}. Then, if we follow approach we used in *alternative 2* on page 11 of M1 we are faced with a problem. We need to compute ϕ_0^{n+1}. However, we no longer have the freedom to do this. When $\mu = 0$, Eq. (4.1.4) redundantly says $\phi_0^{(k+1)} = \phi_0^{(k)}$. If, on the other hand, we define $p_0 = \phi_0 + \bar{c}(3u_{\frac{1}{2}} - u_{\frac{3}{2}})/2$ we will start to re-introduce the complications demonstrated in the appendix of M1.

Thus we are driven to consider the following 'staggered' solution: impose $p_{\frac{1}{2}}$ and compute ϕ_0 from it: $\phi_0 = 2(p_{\frac{1}{2}} - \bar{c}u_{\frac{1}{2}}) - \phi_1$. We can then proceed with our iterative approach, being careful to close the tri-diagonal system of equations correctly:

$$\{1 + c_1^\phi(c^u_{\frac{3}{2}} + 2\tau_{\frac{1}{2}}c^u_{\frac{1}{2}})\}\phi_1^{(k+1)} - c_1^\phi c^u_{\frac{3}{2}}\phi_2^{(k+1)} = \hat{R}_1^\phi - c_1^\phi\{\hat{R}^U_{\frac{3}{2}} - \tau_{\frac{1}{2}}(\hat{R}^U_{\frac{1}{2}} + 2c^u_{\frac{1}{2}}p^{n+1}_{\frac{1}{2}})\} \tag{4.3.1}$$

$$\{1 + c^\phi c^u(1+2\tau)\}\phi_{i_L}^{(k+1)} - c^\phi c^u \phi_{i_L-1}^{(k+1)} = R_{i_L}^\phi - c^\phi\{\tau(R_{i_L+\frac{1}{2}}^U - 2c^u q_{L+\frac{1}{2}}^{n+1}) - R_{i_L-\frac{1}{2}}^U\} \quad (4.3.2)$$

where

$$c_\mu^\phi = \frac{\mu}{\alpha}\frac{\bar{\phi}\Delta t}{2\Delta x}; \quad c_\nu^u = \frac{\nu}{\alpha}\frac{\Delta t}{2\Delta x}; \quad \tau_{\frac{1}{2}} = (1 + 2\bar{c}c_{\frac{1}{2}}^u)^{-1}; \quad (4.3.3)$$

$$c^\phi = \frac{\bar{\phi}\Delta t}{2\Delta x}; \quad c^u = \frac{\Delta t}{2\Delta x}; \quad \tau = (1 + 2\bar{c}c^u)^{-1}. \quad (4.3.4)$$

Equation (4.3.2) is the same as Eq. (3.19) of M1. It is included for completeness.

Having solved the tri-diagonal system we can compute

$$\phi_0^{(k+1)} = 2(p_{\frac{1}{2}}^{n+1} - \bar{c}u_{\frac{1}{2}}^{(k+1)}) - \phi_1^{(k+1)}, \quad (4.3.5)$$

enabling us to re-compute $(\hat{X}^\phi)_0^{(k)}$, and $(\hat{X}^U)_0^{(k)}$ during the next iteration.

The boundary conditions for p and q for the *slow solution* test are

$$p(\frac{\Delta x}{2}, t) = \hat{\phi} exp\Big[-\Big\{\frac{\Delta x/2 - \bar{u}t}{(L/10)}\Big\}^2\Big] \; ; \quad q(L + \frac{\Delta x}{2}, t) = \hat{\phi} exp\Big[-\Big\{\frac{L + \Delta x/2 - \bar{u}t}{(L/10)}\Big\}^2\Big]. \quad (4.3.6)$$

The boundary condition for v is given by Eq. (4.2.2).

First, trajectory truncation was tested, that is, $R^u = (X^u)_{\frac{1}{2}}^n$ when $x_* < \Delta x/2$; $R^u = (X^u)_{i_L+\frac{1}{2}}^n$ when $x_* > L + \Delta x/2$. $R^\phi = (X^\phi)_0^n$ when $x_* < 0$; $R^\phi = (X^\phi)_{i_L}^n$ when $x_* > L$. $R^v = (X^v)_0^n$ when $x_* < 0$; $R^v = (X^v)_{i_L}^n$ when $x_* > L$. Repeating the large α test results in a forecast for the ϕ-field with a more accurate tail. See figure 4.3 and compare it with figure 4.1. The ϕ and v solutions are still slowed down. Using time interpolation eliminates this error as can be seen from the forecast shown in figure 4.4. Thus the slightly discomforting imposition of a boundary condition $p_{\frac{1}{2}}$ half a grid-length in from the v-boundary point has worked well.

4.4 Time interpolation and iterating.

Implementing the time interpolation approach iteratively has turned out to be an elegant and attractive way of imposing well-posed boundary conditions when the departure point is outside the integration area. Unfortunately, there is a potential drawback. If the number of iterations of the solver is large then the computer cost will be unacceptable. For example, with a straightforward implementation of the two schemes described in sub-sections 4.2 and 4.3 convergence can be very slow for some integrations.

To speed up convergence I used the fact that 'trajectory truncation' gives a more accurate solution than the 'zeroth' iteration of of the discretized versions of Eqs. (4.1.4-4.1.6) to to a produce good first guess of the solution. Thus $\phi^{(1)}, u^{(1)}$, and $v^{(1)}$ are generated by solving the system with 'trajectory truncated' boundaries. All subsequent sweeps used

time interpolation. This accelerated convergence significantly. Nevertheless, 10 iterations of the solver were necessary to generate figure 4.2.

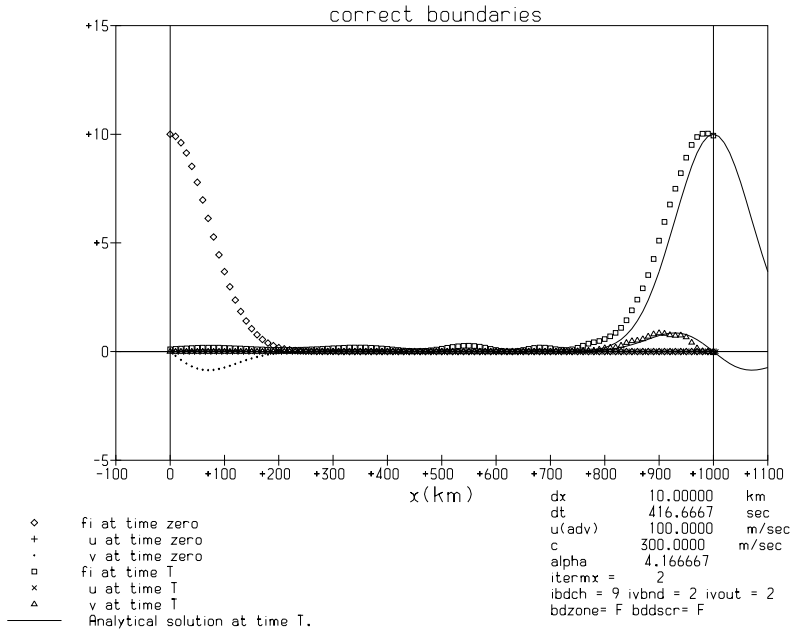

FIGURE 4.3. Same as figure 4.1, but now imposing boundary conditions on p/q via Eq. (4.3.6).

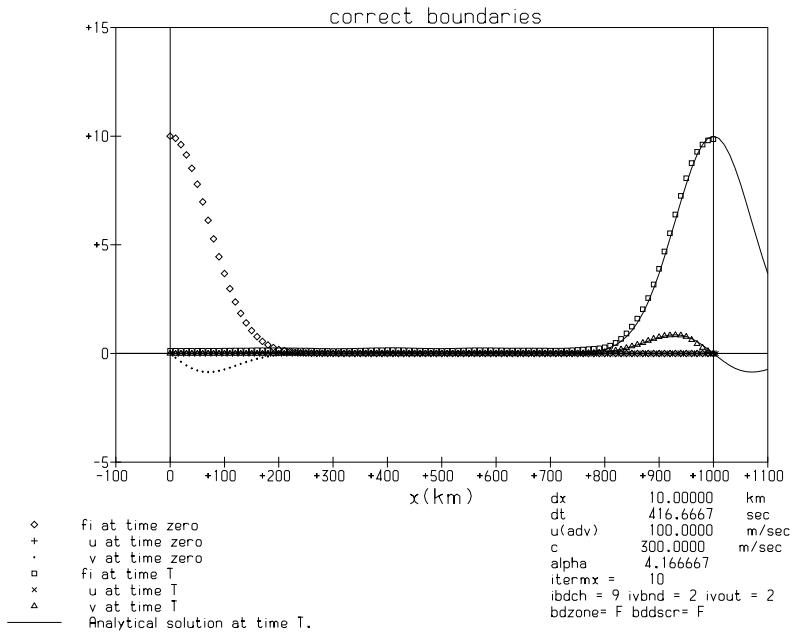

FIGURE 4.4. Same as figure 4.3, but now using 'time interpolation' on the boundary instead of 'trajectory truncation'.

This is not to condemn this approach, but rather to caution that more work may be needed to ensure sufficiently rapid convergence, for example, by introducing pre-conditioning to accelerate convergence or by using a more sophisticated iteration scheme, or both.

5. STABILITY.

When the departure point is outside the integration area an added complication is introduced into the discretized initial-boundary value problem. We have seen that our three solutions to this complication, 'trajectory truncation', a 'well posed buffer zone', and 'time interpolation' yielded stable integrations in practical tests. In this section we make a first attempt at confirming these results analytically by using the energy method to demonstrate the stability of the one-dimensional advection equation with constant velocity $\bar{u} > 0$:

$$\frac{\partial \phi(x,t)}{\partial t} + \bar{u}\frac{\partial \phi(x,t)}{\partial x} = 0; \quad \phi(x,0) = f(x); \quad \phi(0,t) = g(t). \tag{5.1}$$

where $0 \leq x \leq L$ and $t \geq 0$.

Using a linear interpolation to find the departure point value of the field the semi-Lagrangian discretization of Eq. (5.1) gives

$$\phi_i^{n+1} = (1-\hat{\alpha})\phi_{i-p}^n + \hat{\alpha}\phi_{i-p-1}^n \quad \phi_i^0 = f_i; \quad \phi_0^n = g^n. \tag{5.2}$$

The notation is exactly as defined on page 3 of M1. Squaring both sides and re-arranging the terms on the right hand side results in

$$(\phi_i^{n+1})^2 = (\phi_{i-p}^n)^2 - (\hat{\alpha} - \hat{\alpha}^2)(\phi_{i-p}^n - \phi_{i-p-1}^n)^2 - \hat{\alpha}(\phi_{i-p}^n)^2 + \hat{\alpha}(\phi_{i-p-1}^n)^2 \tag{5.3}$$

(a) *trajectory truncation.* This assumption implies that for the first p points Eq. (5.2) is replaced with

$$\phi_1^{n+1} = \phi_2^{n+1} = ... = \phi_p^{n+1} = g^n. \tag{5.4}$$

Therefore we get the following when we sum over the grid points

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 = (p+\hat{\alpha})(g^n)^2 + \sum_{i=1}^{i_L-p}(\phi_i^n)^2 - (\hat{\alpha} - \hat{\alpha}^2)\sum_{i=1}^{i_L-p}(\phi_i^n - \phi_{i-1}^n)^2 - \hat{\alpha}(\phi_{i_L-p}^n)^2 \tag{5.5}$$

Recall that the stability of the semi-Lagrangian scheme is dictated by the condition $0 \leq \hat{\alpha} \leq 1$. Therefore $0 \leq (\hat{\alpha} - \hat{\alpha}^2) \leq 1$, enabling us to write

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq (p+\hat{\alpha})(g^n)^2 + \sum_{i=1}^{i_L}(\phi_i^n)^2 \tag{5.6}$$

Using this equation again to substitute for ϕ_i^n on the right hand side we get

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq (p+\hat{\alpha})\Big\{(g^n)^2 + (g^{n-1})^2\Big\} + \sum_{i=1}^{i_L}(\phi_i^{n-1})^2 \tag{5.7}$$

Repeating this process we finally see that the solution is bounded by the initial condition and the boundary condition, thus establishing stability:

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq (p+\hat{\alpha})\sum_{m=0}^{n}(g^m)^2 + \sum_{i=1}^{i_L}(f_i)^2 \tag{5.8}$$

(b) *Time interpolation.* For the first p points

$$\phi_\mu^{n+1} = (1-\frac{\mu}{\alpha})\phi_0^{n+1} + \frac{\mu}{\alpha}\phi_0^n; \qquad \mu = 1, p. \tag{5.9}$$

Squaring this, summing from 1 to p, and re-arranging the terms we arrive at

$$\sum_{\mu=1}^{p}(\phi_\mu^{n+1})^2 = p(\phi_0^{n+1})^2 - \frac{p(p+1)}{2\alpha}\left\{1-\frac{(2p+1)}{3\alpha}\right\}(\phi_0^n-\phi_0^{n+1})^2 - \frac{p(p+1)}{2\alpha}\left\{(\phi_0^{n+1})^2-(\phi_0^n)^2\right\} \tag{5.10}$$

Since we only 'time interpolate' when $\alpha > p \geq 1$ we can write

$$\sum_{\mu=1}^{p}(\phi_\mu^{n+1})^2 \leq p(\phi_0^{n+1})^2 + \frac{p(p+1)}{2\alpha}(\phi_0^n)^2 \tag{5.11}$$

Once the departure point is inside the integration area we switch to the standard space interpolation scheme, which means the remainder of the derivation is the same as above, resulting in

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq \sum_{i=1}^{i_L}(f_i)^2 + \sum_{m=0}^{n}\left\{p(g^{m+1})^2 + \left(\hat{\alpha}+\frac{p(p+1)}{2\alpha}\right)(g^m)^2\right\} \tag{5.12}$$

Thus the solution is bounded by the initial condition and the boundary condition, establishing stability.

(c) *Well-posed buffer zone.* For the points outside the area we define the fields in terms of the imposed boundary condition (see Eqs. (2.14-2.15) of M1):

$$\phi_{-i}^n = \left(g + \frac{i\Delta x}{\bar{u}}\frac{dg}{dt} + \frac{(i\Delta x)^2}{2\bar{u}^2}\frac{d^2g}{dt^2} + ...\right)^n \tag{5.13}$$

Thus, in Eq. (5.3) the fields outside the area are defined in terms of the boundary condition and when we sum over all the grid points we get

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 = \sum_{i=-p+1}^{i_L-p}(\phi_i^n)^2 - (\hat{\alpha}-\hat{\alpha}^2)\sum_{i=-p+1}^{i_L-p}(\phi_i^n-\phi_{i-1}^n)^2 + \hat{\alpha}(\phi_{-p}^n)^2 - \hat{\alpha}(\phi_{i_L-p}^n)^2 \tag{5.14}$$

Since $(\hat{\alpha}-\hat{\alpha}^2) \leq 1$ we can write

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq \sum_{i=1}^{i_L}(\phi_i^n)^2 + \sum_{i=-p+1}^{0}(\phi_i^n)^2 + \hat{\alpha}(\phi_{-p}^n)^2 \tag{5.15}$$

As before we can continue this procedure until we arrive at

$$\sum_{i=1}^{i_L}(\phi_i^{n+1})^2 \leq \sum_{i=1}^{i_L}(f_i)^2 + \sum_{m=0}^{n}\left\{\hat{\alpha}(\phi_{-p}^m)^2 + \sum_{i=-p+1}^{0}(\phi_i^m)^2\right\} \tag{5.16}$$

Using Eq. (5.13), we finally see that the solution is bounded by the initial condition and the boundary condition, thus establishing stability, provided g is sufficiently smooth.

6. **DISCUSSION.**

We have seen that adding the Coriolis term to the advection/adjustment equations causes additional problems in a C-grid environment. Lowest order extrapolations from the interior overcame these problems and produced stable and accurate integrations in our tests. This is consistent with the findings of Gustafsson, Kreiss and Sundström, (1972), who showed that first order extrapolations are stable for the Lax Wendroff scheme (which is the same as a semi-Lagrangian scheme with quadratic interpolation and $|\alpha| < 1$).

We have seen that both the 'time interpolation' and 'well-posed buffer zone' schemes overcome the accuracy limitation of 'trajectory truncation' for large $|\alpha|$. Looking ahead, it is interesting to ask what are the limitations of the former schemes in the context of an operational numerical weather prediction model? There is a limitation which applies to both, that is, their accuracy depends on how often the boundary is updated. For instance, if this is done every three hours, neither scheme will be much more accurate than trajectory truncation. Each scheme has a particular limitation, at least as we have implemented them here and in M1. As was pointed out in section 4.4 convergence may be a problem for an iterative implementation of the time interpolation scheme. Also, for our particular implementation of the well-posed buffer zone, looking at Eqs. (3.38)-(3.41) of M1 we see that it cannot be used when $|\bar{u}| = |\bar{c}|$, which can happen for internal gravity waves. This does not condemn either method but warns us the that these preliminary implementations are not without possible weaknesses.

Although these are early days it is still nice to have a 'fall-back' position. Here are two. One is to use trajectory truncation and accept the associated inaccuracy when $|\alpha|$ is large on the boundary. The second is based on an idea of Jan Erik Haugen (private communication). If $u\Delta t/\Delta x < 0.5$ on the western boundary, say, we need never invoke the time interpolation nor well posed buffer zone schemes. To accomplish this we surround the uniformly discretized area with a buffer zone in which we use stretched coordinates, each Δx being, say, 15% bigger than its neighbour until this criterion is fulfilled for the maximum advecting velocity we expect perpendicular to the boundary. We choose this as our boundary and apply trajectory truncation there as an insurance against instability for unanticipated large inflow velocities.

Acknowledgement. Thanks to Ivar Lie and Peter Lynch for their helpful constructive criticism of the original manuscript.

7. **REFERENCES.**

Elvius, T. and A. Sundström, 1973: Computationally efficient schemes and boundary conditions for a fine mesh barotropic model based on the shallow water equations. *Tellus,* **25**, 132-156.

Gustafsson, B., H.-O. Kreiss and A. Sundström, 1972: Stability theory of difference approximations for mixed initial boundary value problems. *Math. Comput.,* **26**, 649-686.

McDonald, A., 1999: Well-posed boundary conditions for semi-Lagrangian schemes: the one-dimensional case. *HIRLAM technical report* **43** .

Sundström, A. and T. Elvius 1979: Computational problems related to limited area modelling. *Numerical methods used in atmospheric models, vol. 2. (GARP publication series no. 17)*, 379-416.