JOURNAL OF COMPUTATIONAL PHYSICS 13, 522–535 (1973)

Numerical Boundary Conditions and Computational Modes

J. H. CHEN*

Geophysical Fluid Dynamics Laboratory/NOAA, Princeton University, Princeton, New Jersey 08540

Received June 5, 1973

Methods of specifying the boundary conditions for the finite difference approximation of the $1 - D$ linearized shallow water wave equation are tested. First, extrapolation schemes for boundary values are studied. In particular, "characteristic extrapolations" based on the characteristic equations are proposed for the inflow boundaries. Second, discussions are given on the overall setting of the boundary conditions. An improper treatment of "computational" boundary conditions excites the computational modes in the numerical solutions. Two types of boundary treatments are found to avoid or suppress the unfavorable oscillation due to the computational modes. One type uses order-consistent equation for extrapolations in a well-posed case, and the other uses a local smoothing of the solutions at the boundaries in an over-specified case.

1. INTRODUCTION

In solving the hydrodynamic equations inside a bounded region, when the flow at the boundaries is quite variable in time and space, the setting of boundary conditions presents an important and formidable problem. This problem is often further complicated by the requirement of additional boundary conditions in a numerical calculation using a finite difference method. These additional boundary conditions are called "computational" boundary conditions, in contrast to "physical" boundary conditions which are originally required. In [1], it has been shown that the order-inconsistency between the difference equations and the corresponding differential equations leads to computational modes (or extraneous solutions) and also to the requirement of computational boundary conditions (or extraneous boundary conditions).

In order to specify the computational boundary conditions, arbitrary extrapolation methods are usually used. Mathematically speaking, the values of the variables at the boundaries are to be determined as part of the solution. One possibility is to specify the computational boundary conditions by extrapolating the solution at the interior points to the boundary using the governing equations.

* National Research Council/NOAA Resident Research Associate.

However, this approach has not been widely used except in a few cases (e.g., [3] and [8]). This is probably because improper extrapolations tend to cause numerical instabilities or unreasonable oscillations in the solution. If the proper number of computational boundary conditions (in addition to the physical boundary conditions) are specified, the numerical solution will be determinate. But computational boundary conditions chosen in an ad hoc manner may excite the computational modes excessively. In contrast, by determining the computational boundary conditions by means of an appropriate extrapolation method, i.e., one which is based on the governing equations and which does not admit any computational mode, one can avoid or suppress the computational modes at interior points, at least in the linear cases. It is one of the present objectives to discuss methods of suppressing the computational modes by proper boundary treatments. We hope that the discussion in this paper sheds light on the handling of the boundary setting of the "nested grid" models (e.g., [4], [6], etc.)

2. Governing Equations

The one-dimensional linearized shallow water wave equation is chosen as a test case, i.e.,

$$U_t{}^* + CU_x{}^* + \Phi_x{}^* = 0, \tag{1A}$$

$$\Phi_t{}^* + C\Phi_x{}^* + \Phi_0 U_x{}^* = 0, \tag{1B}$$

where t^* and x^* are the dimensional time and space, U is the velocity, Φ is the geopotential, C is the constant mean velocity assumed positive, and Φ_0 is the constant mean geopotential.

The equations are nondimensionalized as

$$u_t + u_x + \phi_x = 0, \tag{2A}$$

$$\phi_t + \phi_x + (1/\mathrm{Fr})\, u_x = 0, \tag{2B}$$

where the new independent and dependent variables are:

$$t \equiv t^*/(L/C), \qquad x \equiv x^*/L,$$

$$u \equiv U/C, \qquad \phi \equiv \Phi/C^2.$$

$\mathrm{Fr} \equiv C^2/\Phi_0$, and Fr is the Froude number, L is the length of the limited domain in consideration. It is assumed that the Froude number is less than unity, because we are mainly interested in large scale atmospheric flows.

In order to specify the boundary conditions and to choose an exact solution in closed form, we construct the characteristic equations. Multiplying (2A) by a and adding the resultant equation to (2B), we have:

$$[(\partial/\partial t) + (1 + a)(\partial/\partial x)](\phi + au) = 0, \tag{3A}$$

and, subtracting the resultant equation from (2B), we have:

$$[(\partial/\partial t) + (1 - a)(\partial/\partial x)](\phi - au) = 0 \tag{3B}$$

where $a \equiv \mathrm{Fr}^{-1/2}$. Equation (3A) is an equation for the right-running characteristic with speed of $(1 + a)$, and Eq. (3B), for the left-running characteristic with speed of $(1 - a)$, as can be seen in Fig. 1. The trajectory with advection speed

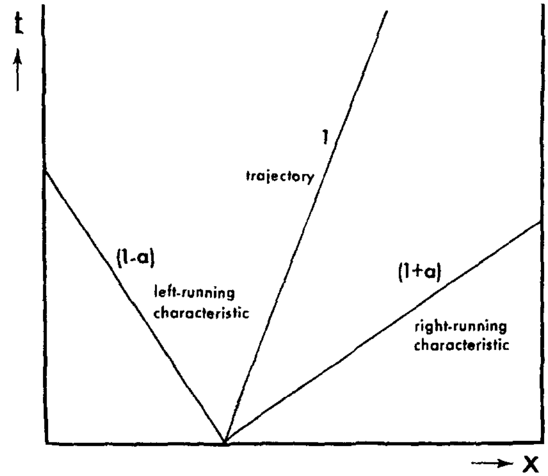

FIG. 1. Characteristics and trajectory.

of 1 is also indicated. One can readily see that the solutions below satisfy Eqs. (3A) and (3B),

$$\phi + au = f(x - \beta t), \tag{4A}$$

$$\phi - au = g(x + \gamma t), \tag{4B}$$

where $\beta \equiv a + 1$, $\gamma \equiv a - 1$, and f and g are arbitrary functions. Since the Froude number is less than unity and hence a is greater than 1, one of these two characteristics goes downstream and the other goes upstream. Consequently, we need one boundary condition downstream and another upstream.

Let us next assume the following parameters and the following functional forms for f and g:

$$\mathrm{Fr} = 1/9, \text{ so that } a = 3 \text{ and } \beta = 4,\ \gamma = 2,$$

and

$$f(\xi) = 2\sin(\alpha\xi), \tag{5A}$$

$$g(\xi) = -2\sin(2\alpha\xi), \tag{5B}$$

where $\alpha = 2\pi$, wavenumber 1 for f, 2 for g, and ξ is a dummy variable. Hence we have the exact solutions as follows:

$$\phi = \sin\alpha(x - \beta t) - \sin 2\alpha(x + \gamma t), \tag{6A}$$

$$u = (1/a)[\sin\alpha(x - \beta t) + \sin 2\alpha(x + \gamma t)]. \tag{6B}$$

The initial conditions are given by setting $t = 0$, i.e.,

$$\phi = \sin(\alpha x) - \sin(2\alpha x), \tag{7A}$$

$$u = (1/a)[\sin(\alpha x) + \sin(2\alpha x)]. \tag{7B}$$

The boundary conditions are taken to be:

$$u = 0 \text{ at } x = 0 \text{ and } x = 1,$$

according to (6B). These boundary conditions do not change in time due to the choice of β, γ, and $\alpha(= 2\pi)$.

3. Finite Difference Equations

Equations (2A) and (2B) are approximated by finite difference. In particular, the Euler-backward scheme [5] is used, which consists of two steps.

Step 1.

$$up_j^{n+1} = u_j^n - (\Delta t/2\Delta x)(u_{j+1}^n - u_{j-1}^n) - (\Delta t/2\Delta x)(\phi_{j+1}^n - \phi_{j-1}^n), \tag{8A}$$

$$\phi p_j^{n+1} = \phi_j^n - (\Delta t/2\Delta x)(\phi_{j+1}^n - \phi_{j-1}^n) - (\Delta t/2\Delta x)(1/\mathrm{Fr})(u_{j+1}^n - u_{j-1}^n), \tag{8B}$$

where up, ϕp are u, ϕ at the first, or the predictor step, Δx is the grid size, Δt is the time interval, superscript n denotes $t = n\Delta t$, and subscript j denotes $x = j\Delta x$, etc.

Step 2.

$$u_j^{n+1} = u_j^n - (\Delta t/2\Delta x)(up_{j+1}^{n+1} - up_{j-1}^{n+1}) - (\Delta t/2\Delta x)(\phi p_{j+1}^{n+1} - \phi p_{j-1}^{n+1}), \tag{9A}$$

$$\phi_j^{n+1} = \phi_j^n - (\Delta t/2\Delta x)(\phi p_{j+1}^{n+1} - \phi p_{j-1}^{n+1}) - (\Delta t/2\Delta x)(1/\mathrm{Fr})(up_{j+1}^{n+1} - up_{j-1}^{n+1}). \tag{9B}$$

The domain extends from $j = 1$ to $J(=39)$, and $\Delta x = 1/38$. The condition of computational stability is $\beta(\Delta t/\Delta x) \leqslant 1$. With $\beta = 4$, it is clearly satisfied for $\Delta t = 0.1\Delta x$.

4. Boundary Conditions

In connection with extrapolation of boundary values, we will use two terminologies, i.e., "scheme" and "method." A scheme refers to a specific procedure for obtaining the computational boundary values by extrapolating the solution of a difference equation. It may be applied to the simple advection, or more general wave equations. In the appendix, the "upwind scheme" and the "backward-time scheme" are presented as examples of practical extrapolation schemes. Both of them use upwind differences in space and can therefore be used at an outflow boundary for extrapolating the boundary values. When a scheme is applied to a given wave equation, the resulting difference equation is called a method of extrapolation. Thus, when the upwind scheme is applied to the advection equation (2A) or (2B), the resulting difference equation is called the "advection upwind method," or simply, the "upwind method," and when it is applied to the characteristic equation (3A) or (3B), the resulting difference equation is called the "characteristic-upwind method." Similarly, when the backward-time scheme is used, we have the "advection backward-time method," or simply the "backward-time method," and the "characteristic backward-time method."

Let us next turn to Eqs. (2A) and (2B). The derivatives of u and ϕ in these equations are first order in space, whereas the differences for u and ϕ in (8A) and (8B) are second order.[1] In other words, the difference equations are one order higher in space than the original differential equations for both u and ϕ. Therefore, two computational boundary conditions are required.

In order to discuss boundary settings, two situations need special attention: one is the case in which the boundary conditions are "well-posed," and the other is the case in which they are "over-specified."

For the first case, i.e., a well-posed boundary value problem, we shall discuss only the computational boundary conditions because there is no special difficulty for the physical boundary conditions. The boundary on the right is an outflow

[1] A difference equation is m-th order difference when it spans over $(m + 1)$ grid points.

boundary; thus either the upwind or the backward-time scheme can be applied to either of the advection equations (2A) or (2B) in order to obtain the computational boundary values at this right boundary. On the other hand, the boundary on the left is an inflow boundary, so that neither scheme can be applied directly to these equations. Although the downwind scheme can be used at an inflow boundary, it is an amplifying scheme (see the appendix) and therefore will be avoided here. In order to find an alternate method for treating this boundary, let us consider the left-running characteristic, represented by Eq. (3B). The speed of the characteristic $(1 - a)$ will be negative since the Froude number is less than unity as has been assumed. This implies that the left-running characteristic goes out at the left boundary, so that this boundary becomes effectively an "outflow boundary" as far as this characteristic is concerned, even though it is an inflow boundary for the advection equations. Therefore, either scheme in the appendix can be applied to the characteristic equation (3B) to obtain the computational boundary values at the inflow boundary on the left. Similarly, the right-running characteristic equation (3A) can be used at the right boundary in order to extrapolate the boundary values there. This is because the right boundary is an outflow boundary for this characteristic (as well as for the advection equations) since its speed $(1 + a)$ is positive for any value of a. We shall refer to such methods as "characteristic extrapolation methods," and shall use the terminology "characteristic-upwind method" or "characteristic backward-time method" to refer to the method using the upwind or the backward-time scheme as applied to Eqs. (3A) and (3B). Let us summarize the discussion in this first case. At the outflow boundary (on the right) there are two types of extrapolation methods: one type bases on the advection equations (2A) and (2B), the other bases on the right-running characteristic equation (3A). On the other hand, at the inflow boundary (on the left) only one type of extrapolation method is available, i.e., one that bases on the left-running characteristic equation (3B).

The second case, i.e., the case of over-specification, is the following. Four boundary conditions are required for the system of difference equations (8A) and (8B), whereas only two boundary conditions are needed for the system of differential equations (2A) and (2B). If one specifies u at both ends, the problem of the differential equation system is well-posed. Likewise, if one specifies ϕ at both ends, the problem is also well-posed. In this sense, there are two sets of well-posed conditions. If, on the other hand, one specifies both sets of conditions above, the problem is over-specified. Although the problem of the differential equation system is over-specified, the numerical solutions to the difference equation system are determinate. The only thing is that they include the computational modes, which lead to the grid-to-grid oscillations. These oscillations may not be attributed to numerical instabilities as Shapiro and O'Brien [4] did. Smoothing through use of artifical diffusion is often applied in order to suppress these oscillations. But if the coefficient of diffusion is large enough to be effective, the smoothing will distort the physical modes of the solutions as well. In order to avoid this trouble, a local smoothing could be applied at the boundary. It reduces practically to zero the amplitudes of the computational modes at the boundary, which are the primary sources for the computational modes inside the domain. Since the local smoothing is applied only at the boundaries, the physical modes are left intact. Also, because the computational modes are suppressed, there is no longer any constraint of the additional boundary conditions. Therefore, the local smoothing relaxes the constraint of the over-specification, such that it seems as if some set of well-posed boundary conditions were specified. A question one should ask here is what "equivalent" set of well-posed boundary conditions do the resultant solutions correspond to? If this equivalent set of boundary conditions is incompatible with the original set of well-posed boundary conditions, then the numerical solution will not correspond to the solution for (either set of) well-posed boundary conditions. However, if the sets of boundary conditions are "approximately compatible," the solution should be close to the solution of the well-posed problem.

Six versions of boundary settings are tested (see also Fig. 2).

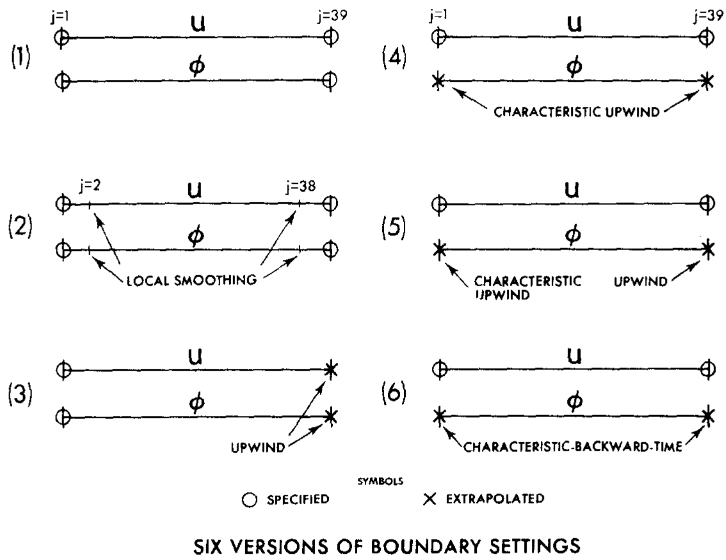

FIG. 2. Six versions of boundary settings.

(1) All four values, i.e., u, ϕ at both ends ($j = 1$ and 39), are specified according to Eqs. (6A) and (6B). This is an over-specified case.

(2) All four values are specified as in (1) and, in addition, local smoothing is

applied only at $j = 2$ and 38 for all time. For example the smoothing for u at $j = 2$ is processed as u_2 (smoothed) $= (1/2)\, u_2 + (1/4)(u_1 + u_3)$. This is also an over-specified case.

(3) u and ϕ are specified at $j = 1$ and those at $j = 39$ are extrapolated by the upwind method. This is over-specification at the left boundary and under-specification at the right boundary. Therefore, this specification is not well-posed.

(4) u is specified at both ends and ϕ is extrapolated at both ends by the characteristic-upwind method. It is a well-posed case.

(5) u is specified at both ends and ϕ at $j = 1$ is extrapolated by the characteristic-upwind method and ϕ at $j = 39$ by the upwind method. This specification is also well-posed.

(6) u is specified at both ends and ϕ is extrapolated at both ends by the characteristic-backward-time method. This specification is well-posed.

Apart from these six calculations, another calculation was carried out using cyclic boundary conditions. This provides a special situation in which a numerical solution free of boundary treatments can be obtained.

5. Results

In Fig. 3A and 3B are shown the solutions of the geopotential height and the velocity respectively at the 150-th time step for various boundary settings, in comparison with the exact solution and the solution for the cyclic boundary conditions case.

Methods (4), (5), and (6) are all well-posed and they use order-consistent difference equations for extrapolating the computational boundary values. They yield almost the same and equally good results, and, therefore, only the solution by Method (4) is displayed in the figure. The amplitude is slightly lower than that of the exact solution not due to the numerical boundary conditions, but due to the (interior) difference equations.

Method (1) is over-specified, and in fact the computational modes were excited as is seen in the grid-to-grid oscillations of the solution. When smoothing is applied at points next to the boundary (Method (2)), the computational modes are suppressed, and, hence, the solution appears to be as good as those of Methods (4), (5), and (6). It should be noted that the boundary conditions specified in Methods (1) and (2) are exact solutions to the differential equations (2A) and (2B), to which the difference equations (8) and (9) are approximations, therefore these boundary conditions are approximately compatible to these difference equations.

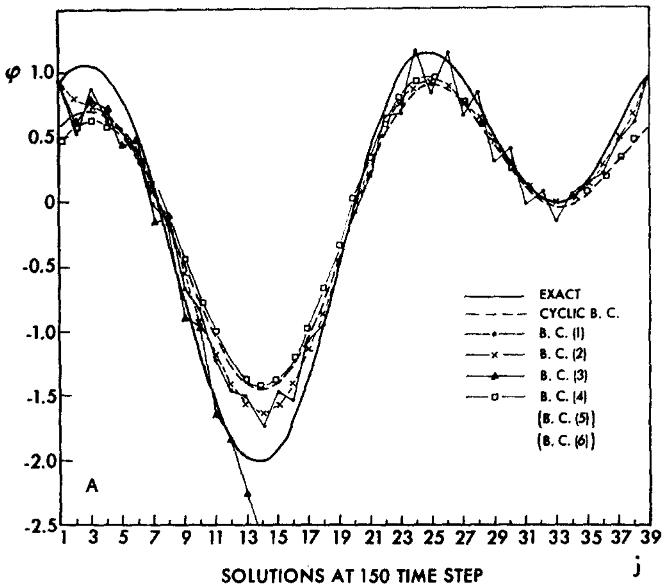

FIG. 3A. Solutions ϕ at the 150-th time step.

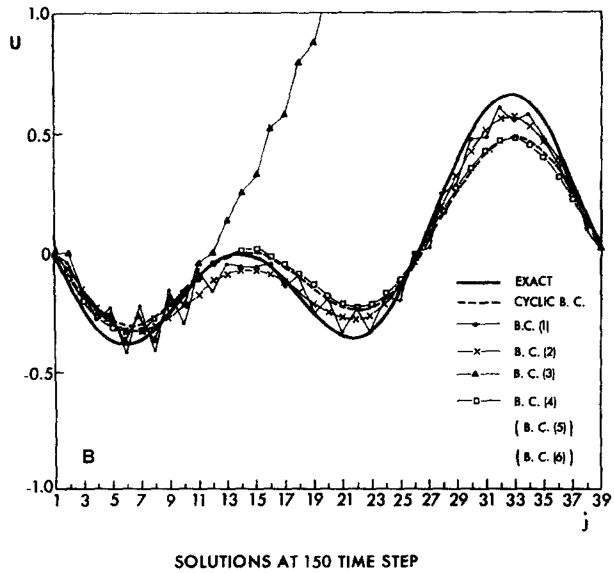

FIG. 3B. Solutions u at the 150-th time step.

Otherwise, the solution may be free of numerical oscillations, but it can be quite different from the solution of the well-posed problem such as Methods (4), (5), and (6). Also, it should be noted that this kind of smoothing be discriminated from smoothing applied everywhere, which would seriously distort the physical modes.

Method (3) is not well-posed; the boundary on the right is free (i.e., underspecified). The resulting solution deviates sharply from the exact solution. The situation for the geopotential height is also shown in Fig. 4, which is the same as

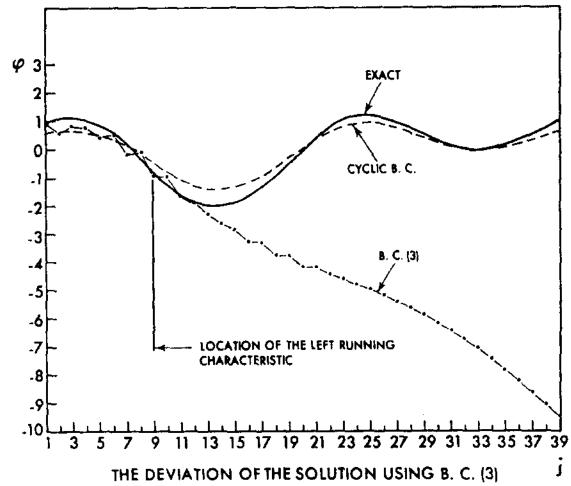

FIG. 4. The deviation of the solution using B.C. (3).

Fig. 3A but with a different scale to show the whole solution. The deviation appears in the range of influence of the boundary on the right, as would be expected. The range of influence extends toward the left with the speed γ of the left-running characteristic. Computational instability could be another possible cause of the deviation. However, in Fig. 4, it can be seen that the solution on the right-hand side is smooth, indicating that this is not the case.

6. CONCLUSIONS

The increase in order in the difference equations over the differential equations creates computational modes, and at the same time requires computational boundary conditions.

The computational boundary conditions can be supplied by using extrapolations based on the governing equations. The upwind scheme and the backward-time scheme are satisfactory for obtaining the outflow computational boundary values by extrapolation. They cannot, however, be used to obtain the inflow boundary values. For this purpose, the left-running characteristic equation, instead of an advection equation, can be used.

If the computational boundary conditions are properly specified, the computational modes in the numerical solution can be suppressed, at least in the linear cases. In fact, using order-consistent difference equations to obtain the computational boundary values we were able to avoid large amplitudes of the computational modes. Therefore, the solution will depend only on the physical boundary conditions, which are required to be properly posed. In other words, a set of well-posed boundary conditions, coupled with proper computational boundary conditions, yields a good numerical solution.

Over-specifications excite computational modes. However, smoothing at points just next to the boundaries suppresses these modes. Also, the resulting solution is reasonable if the boundary conditions are approximately compatible to the difference equation. This method serves as an alternative to the extrapolation of computational boundary values in a well-posed problem, and what is more important, it may be the only method available when the derivation of the well-posed boundary conditions is too difficult to analyze. On the other hand, under-specifications lead to a solution which deviates sharply from both the exact solution and the numerical solution with cyclic boundary conditions.

In short, two methods are particularly recommended. One is to specify a set of well-posed boundary conditions and to use order-consistent difference equations based on the governing equations for the computational boundary conditions. The other is to specify all numerical boundary conditions which are approximately compatible, and then to apply local smoothing at grid points next to the boundaries. They may find their application to the nested grid problem in which all the boundary values for the finer resolution domain are available from the solution of the coarser resolution domain.

For extension to the two-dimensional case, all the conclusions above are valid except for the characteristic extrapolations. Instead of characteristic lines in the one-dimensional case, we have characteristic cones, called the Monge cones, in the two-dimensional case, for which one-sided difference equation can not be written for the purpose of obtaining the boundary values by extrapolation. However, this problem can be overcome by using the projection of the Monge cone [7]. For extension to the nonlinear case, it should be noted that the computational modes can also be excited in the interior region, in addition to at the boundaries, of the flow field. Therefore additional effort may be needed when the flow field has a very strong variation.

APPENDIX: SCHEMES OF EXTRAPOLATION

Two kinds of general schemes of extrapolation are discussed in the following using the simple advection equation in the one-dimensional case and the advection equation in the two-dimensional case. The schemes are intended to be applicable to more general wave equations. In order that they do not admit any computational mode, they must be order-consistent to the original differential equations. Therefore, one-sided difference schemes are used.

1. *The Upwind Schemes*

Let us consider the simple advection equation

$$u_t + cu_x = 0,$$

where c is assumed constant and positive. This equation is approximated by the following finite difference equation.

FTBS (forward in time and backward in space)

$$\begin{aligned} u_j^{n+1} &= u_j^n - c(\Delta t/\Delta x)(u_j^n - u_{j-1}^n), \\ &= (1 - r)\, u_j^n + r u_{j-1}^n , \end{aligned}$$

where $r \equiv c(\Delta t/\Delta x)$, being the Courant number, and the Courant condition, $|r| < 1$, is assumed satisfied.

An extrapolation scheme is "stable" [2], or rather, "nonamplifying," if the value extrapolated is bounded by those of the neighboring points. It will be shown below that the above scheme possesses such a property.

From the finite difference equation, it follows that

$$\begin{aligned} |u_j^{n+1}| &\leqslant |1 - r| \cdot |u_j^n| + |r| \cdot |u_{j-1}^n| \\ &= (1 - r)\, |u_j^n| + r\, |u_{j-1}^n| \\ &\leqslant (1 - r) \max_{(j)} |u_j^n| + r \max_{(j)} |u_j^n| \\ &= \max_{(j)} |u_j^n|, \end{aligned}$$

so that

$$|u_j^{n+1}| \leqslant \max_{(j)} |u_j^n|.$$

One sees that $|u_j^{n+1}|$ is bounded by values at the previous time step, and therefore the computation is nonamplifying. In order to apply this relation at the boundary on the right, i.e., $j = J$, one inserts J into j in the above difference equation.

Similarly, for $c \leqslant 0$, the scheme of "forward in time and forward in space" is used to obtain the value at $j = 1$, which was used by Platzman [2]. Nitta [3] also used this kind of extrapolation, but he used a forward time step of $2\Delta t$. The stability condition, $u(\Delta t/\Delta x) \leqslant 1/2$, is more restrictive than the Courant condition for the (interior) difference equation. It is to be noted that an extrapolation based on a nonamplifying scheme may not be strictly necessary or sufficient for the stability of a numerical problem because its interaction with the interior difference equation may also cause or suppress instability. However, a nonamplifying extrapolation seems to have less chance of causing instability.

In the case of 2-dimensional advection equation, i.e.,

$$u_t + cu_x + au_y = 0,$$

where a is the mean velocity component in y, the following finite difference scheme can be used:

FTBXBY (forward in time, backward in X and backward in Y)
for $c > 0$, $a > 0$,
FTBXFY for $c > 0$, $a < 0$,
FTFXBY for $c < 0$, $a > 0$,
FTFXFY for $c < 0$, $a < 0$.

Another kind of scheme that immediately comes into one's mind is a reversal of the roles of t and x, i.e., BTFS for $c > 0$ and FTFS for $c < 0$. Unfortunately, this is stable only if the Courant number is greater than 1.

2. *The Backward-Time Schemes*

We shall use again the simple advection equation mentioned earlier, where c is assumed constant and positive. The finite difference scheme for the one-dimensional case is:

$$\text{BTBS} \qquad u_j^{n+1} - u_j^{\,n} + c\,\frac{\Delta t}{\Delta x}\,(u_j^{n+1} - u_{j-1}^{n+1}) = 0$$

or

$$u_j^{n+1} = (u_j^{\,n} + ru_{j-1}^{n+1})/(1 + r),$$

where $r \equiv c(\Delta t/\Delta x)$, and no assumption about the Courant condition is needed here. Even though backward-time difference is used, the scheme is explicit for the determination of the boundary value. In this case, we have

$$\begin{aligned} | u_j^{n+1} | &\leqslant (| u_j^{\,n} | + r\, | u_{j-1}^{n+1} |)/(1 + r) \\ &\leqslant \max(| u_j^{\,n} |, | u_{j-1}^{n+1} |). \end{aligned}$$

So that u_j^{n+1} is bounded by its neighboring values. To extrapolate to $j = J, j = J$ is inserted in the difference scheme. Similarly, for $c < 0$, BTFS can be used to obtain boundary value at $j = 1$.

For the 2-dimensional advection equation, one of the following schemes is used:

$$\begin{aligned} &\text{BTBXBY for } c > 0, \quad && a > 0, \\ &\text{BTBXFY for } c > 0, && a < 0, \\ &\text{BTFXBY for } c < 0, && a > 0, \\ &\text{BTFXFY for } c < 0, && a < 0. \end{aligned}$$

In the upwind scheme the way of difference in y depends on the sign of a; whereas in the backward-time scheme, it depends on the sign of (a/c). The upwind scheme requires the Courant condition. Therefore, it can be used only when explicit schemes are used for interior points. In contrast, the backward-time schme does not require the Courant condition, and thus can be used when either an explicit or an implicit scheme is used for interior points.

Acknowledgments

The author extends grateful acknowledgment to the National Research Council for the support of a NOAA Resident Research Associateship. He is indebted to his colleagues in the Geophysical Fluid Dynamics Laboratory, in particular, his scientific advisor, Dr. K. Miyakoda, for their helpful comments and interesting discussions. Gratitude is also expressed to Drs. K. Miyakoda, Y. Kurihara, and C. T. Gordon for reviewing the paper. In addition, he would like to thank Mr. P. G. Tunison who drafted the figures and Mrs. B. M. Williams who typed the manuscript.

References

1. J. H. Chen, Finite Difference Methods and the Leading Edge Problem, Ph.D. Thesis, AMS Dept. Princeton University, July, 1971.
2. G. W. Platzman, The lattice structure of the finite-difference primitive and vorticity equations, *Mon. Wea. Rev.*, **86** (1958), 8, 285–292.
3. T. Nitta, The outflow boundary condition in numerical time integration of advective equations, *J. Met. Soc. Japan*, **40** (1962), 1, 13–24.
4. M. A. Shapiro and J. J. O'Brien, Boundary conditions for fine-mesh limited-area forecasts, *J. Appl. Mat.*, **9** (1970), 345–349.
5. T. Matsuno, Numerical integrations of the primitive equations by a simulated backward difference method, *J. Met. Soc. Japan, Ser.* **2, 44** (1966), 76–84.
6. G. E. Hill, Grid telescoping in numerical weather prediction, *J. Appl. Meteor.*, **7** (1968), 29–38.
7. J. H. Chen and K. Miyakoda, to be published.
8. T. Matsuno, False reflection of waves at the boundary due to the use of finite differences, *J. Met. Soc. Japan*, **44** (1966), 145–157.