Using Exact Solutions to Develop an Implicit Scheme for the Baroclinic Primitive Equations

D. Marchesin

Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro 22453, Brazil

(Manuscript received 8 June 1982, in final form 31 August 1983)

ABSTRACT

A method on the creation of exact solutions for nonlinear initial value problems is described. These solutions are employed in the development of numerical schemes for computer solution of these problems. The method is applied to a new fully implicit scheme on a vertical slice for the isentropic baroclinic equations.

1. Introduction

Meaningful exact solutions of evolution equations, when they can be found, are very useful in developing and testing numerical methods for solving these equations. They can be used to ascertain accuracy and stability properties of the numerical schemes and, therefore, to compare the performance of competing schemes. In addition, they can be used to verify program correctness.

For nonlinear equations exact solutions are seldom known. Even when available, they are usually too trivial to give more than general indications about the behavior of the numerical algorithm. Comparison between the exact and computed solutions frequently fails to reveal programming errors.

We describe a simple technique to create nontrivial exact solutions for systems in which evolution is described by nonlinear partial differential equations. These solutions can have an arbitrary dependence in time and space. The key idea is to modify the equations by the addition of a new term. This modification does not alter the scheme’s stability and accuracy properties.

The first step in our procedure is to specify the desired solution. This solution determines a new term to be added to the equations. The prescribed solution can be chosen to isolate the influence of any term in the equations. This idea has existed for some time, but our implementation makes it a particularly flexible and useful tool, with minimal programming overhead.

For the dynamics part of primitive global models, Chao and Geller (1982) successfully utilized solutions of the linearized primitive equations as a tool for detecting coding errors. The linear approximation intrinsic in this technique, however, is such that phenomena originated by nonlinear terms are not adequately tested. Our technique, besides being more general, does not suffer from this shortcoming.

Our procedure proved to be a valuable tool in the development of an implicit difference scheme for the baroclinic primitive equations. The analytic solutions we were able to find for these equations on a vertical slice were too trivial and devoid of physical significance to be of any help in testing our program. In fact, it was precisely this difficulty which led us to the technique.

The efficiency of all current primitive equation models is limited by a stability criterion on the size of the time step. We are currently developing a fully implicit method for which a time step is chosen solely on the basis of resolving the physical flow of interest, and is therefore particularly advantageous for fine meshes. This scheme for the global baroclinic equations is second-order accurate in time, second-order accurate vertically and fourth-order accurate horizontally.

Based on the work of Beam and Warming (1976), we have developed and tested such a scheme for the shallow water equations on the sphere (unpublished). A similar scheme has been developed by Fairweather and Navon (1980) for limited area forecasts. Gilliland (1981) has also developed a related scheme based on the formulation of Briley and McDonald (1977). These implicit schemes derive their efficiency from the use of an alternate direction implicit (ADI) method, which requires all the dependent variables in the equations to possess a corresponding prognostic equation.

Our experiments on the shallow water equations indicate that for fine meshes (3° or less) the implicit scheme is more efficient than the explicit scheme of Williamson and Browning (1973).

However, the primitive equations contain diagnostic quantities, and we were forced to generalize Beam-Warming’s scheme in order to resolve this difficulty.

---

1 This work was partially performed at Courant Institute of Mathematical Sciences, New York University and at Sigma Data Services Corporation, GLAS-NASA/Goddard Space Flight Center.

© 1984 American Meteorological Society
Our first tests on the new scheme were made on a vertical slice, as reported in this work. Our scheme appears to be stable on a fine mesh (3') for any time step tested (up to 2 h) even when the mesh spacing simulates polar convergence. All the advantages already observed in the barotropic experiments seem to be preserved.

2. The basic modification

Consider the model for meteorological problems

\[
\begin{align*}
W_t + L(W) &= 0, \\
W(t = 0) &= W_0, 
\end{align*}
\]  
(2.1)

where \(L(W)\) is a (nonlinear) partial differential operator. The problem (2.1) is given with appropriate homogeneous boundary conditions for \(W\). For concreteness let us consider (2.1) with two spatial variables \(x, z\). We assume that Eq. (2.1) is well posed, and has unique, stable solutions for representative classes of initial conditions \(W_0\).

Let \(U = U(x, z, t)\) be a given smooth function satisfying the boundary conditions stipulated for \(W\) as well as the initial condition \(U(t = 0) = W_0\). Then the initial value problem

\[
\begin{align*}
W_t + L(W) &= U_t + L(U), \\
W(t = 0) &= W_0 
\end{align*}
\]  
(2.2)

with the same homogeneous boundary conditions has the known nontrivial solution

\[
W = U. 
\]  
(2.3)

We assume that Eq. (2.2) is well-posed and has unique, stable solutions just as well as (2.1). This is the case when the operator \(L\) is linear: the modified problem (2.2) is equivalent to (2.1) in the sense that a numerical scheme involving some discretization for (2.1) has the same properties of accuracy and stability when applied to (2.2). For bounded times in finite difference schemes, this is precisely true in the limit when space and time discretization intervals tend to zero (see Richtmyer and Morton, 1967, Sections 4.7, 4.8, 5.3, 8.4). For nonlinear systems this equivalence is not strictly true. Nevertheless, it is correct for smooth solutions whenever the behavior of a scheme can be inferred from linear analysis about relatively simple states.

Since there is no general global theory for nonlinear initial value problems, the arbitrary function \(U\) should be chosen to resemble physically meaningful solutions of (2.1); an adequate numerical scheme for (2.1) should be well behaved near solutions of (2.1). In many areas of science it is customary to analyze the behavior of nonlinear initial value problems in terms of traveling waves; this is justified for phenomena in which the nonlinearities play a relatively small role. Thus we consider families of such waves in which \(U\) depends on parameters chosen to simulate different kinds of motion.

3. Implementing the modification

To discuss details of implementation, we assume a particular form of \(L(W)\).

\[
W_t + L(W) = \partial W + \frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial z} + H(W), 
\]  
(3.1)

where

\[
W = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix}, \quad F(W) = \begin{pmatrix} F_1(W) \\ F_2(W) \\ F_3(W) \end{pmatrix},
\]

\[
G(W) = \begin{pmatrix} G_1(W) \\ G_2(W) \\ G_3(W) \end{pmatrix}, \quad H(W) = \begin{pmatrix} H_1(W) \\ H_2(W) \\ H_3(W) \end{pmatrix}. 
\]  
(3.2)

with appropriate boundary conditions.

In our program we create three functions \(U_1, U_2, U_3\), say

\[
\begin{align*}
U_1(x, z, t) &= A_1 + B_1 \sin(\alpha x + \beta z + \gamma t) \\
U_2(x, z, t) &= A_2 + B_2 \sin(\alpha x + \beta z + \gamma t) \\
U_3(x, z, t) &= A_3 + B_3 \sin(\alpha x + \beta z + \gamma t) 
\end{align*}
\]  
(3.3)

with arbitrary parameters \(A_1, A_2, A_3, B_1, B_2, B_3, \alpha, \beta, \gamma\). The functions \(U_1, U_2, U_3\) must be constructed so that they satisfy the boundary conditions for \(W\).

For flexibility and generality, the expressions \(U_t + L(U)\) should be evaluated numerically rather than analytically, as we will describe. In this way changes of the functions \(U_t\) become very simple, minimizing the possibility of introducing programming errors. For this purpose, we create three functions which compute approximately \(\partial F/\partial x\), \(\partial G/\partial z\). These routines are programmed so that they can compute the derivative of any function at \((x, z, t)\). The routines should compute the derivatives much more accurately than the numerical scheme being developed for Eq. (2.1). (In the example discussed in Section 8, we employ a 5-point, centered fourth-order difference approximation computed with a very small increment.)

Next we construct the functions \(F_i(U_1, U_2, U_3), G_i(U_1, U_2, U_3), H_i(U_1, U_2, U_3), i = 1, 2, 3\). Then it is easy to construct the functions \(I_i = [U_t + L(U)]_t\), i.e.,

\[
I_i(x, z, t) = \frac{\partial U_i}{\partial t} + \frac{\partial F_i(U_1, U_2, U_3)}{\partial x} + \frac{\partial G_i(U_1, U_2, U_3)}{\partial z} + H_i(U_1, U_2, U_3). 
\]  
(3.4)

The functions \(I_i\) are the counterterms appearing in the rhs of (2.2).
This procedure is simple to implement and creates a flexible program. Only the routines for \( U_i \) have to be reprogrammed when these functions are changed; all the other routines used in Eq. (3.4), for \( \partial/\partial x, \partial/\partial z, \partial/\partial t, F_i, G_i, H_i \), are not touched. These advantages are achieved at the expense of computer time, because of the large number of (essentially) redundant functions employed. Since this procedure is meant to be used for development rather than production, this is a minor disadvantage.

4. Using the exact solutions

Working with (2.2) instead of (2.1) may be helpful in many situations: similar ideas have been used by many workers. Eq. (2.2) may be used to find the rate of convergence to a steady state when such a state exists (Stoker et al., 1975). It may be used to find the order of accuracy of the scheme employed to solve (2.1), as well as the magnitude of the truncation errors. In the rest of this section we describe how to use it in the detection of programming errors.

Suppose that the scheme used to solve Eq. (2.1) is at least second-order accurate in time. Most programming errors associated with discretization in time reduce the accuracy of the scheme to first order. In practice, errors can be isolated using the following procedure. First, choose the parameters \( \alpha = \beta = 0 \) in (3.3). Thus \( U_1, U_2 \) and \( U_3 \) have no spatial dependence. Then make a series of experiments using time steps \( \Delta t, \Delta t/2, \Delta t/4, \ldots \), etc., and compute the errors at fixed time \( t \)

\[
e(T) = \| W_i (t = T) - U_i (t = T) \|,
\]

where \( \| \cdot \| \) denotes the rms error norm. [Here \( W_i (i = 1, 2, 3) \) is the computed solution and \( U_i (i = 1, 2, 3) \) the prescribed solution.] If the program is correct, the ratio of successive errors in the experiments must behave asymptotically as \( (\frac{1}{2})^n \), where \( n \) is the order of the accuracy of the scheme (in time).

Similarly, most programming errors associated with space discretization reduce the spatial order of accuracy. In practice, these errors can be detected by means of a series of experiments with the parameters in \( U_1, U_2, U_3 \), adjusted in such a way that these functions are independent of one spatial variable. These experiments use mesh sizes \( h, h/2, h/4, \ldots \) and corresponding time steps \( \Delta t, \Delta t/2, \Delta t/4, \ldots \), where \( f \) is the order of accuracy in space \( m \) divided by the order of accuracy in time. If the ratio between successive errors \( e_i \) does not behave asymptotically as \( (\frac{1}{2})^n \), then there is probably a programming error associated with the space variable on which \( U_i (i = 1, 2, 3) \) depend.

Most of the computer experiments described here can be made with the parameters set so that the functions \( U_i \) depend on only one spatial variable. The cost of these experiments can be reduced substantially using a very coarse mesh in the other variables. A serious limitation of the method is the presence of roundoff error, which tends to dominate as space and time mesh sizes tend to zero. When using computers with small word length, one notes that the program should be compiled in double precision.

Errors associated with one of the equations in (3.1) may be isolated by adjusting the parameters \( A, B \) for the corresponding \( U \) function so that the equation disappears from the system. Finally, in conjunction with the techniques described, it is possible to further isolate sources of error due to the availability of the solution \( W = U \). This is accomplished by replacing sections of the program by "fake" sections that use the solution \( U(x, z, t) \) directly instead of advancing the scheme in time.

Finally, we want to stress that the tests must be made at a fixed time, since the long time behavior of (2.1) and (2.2) may be different, because of the inhomogeneous term present in the latter. As a reviewer pointed out, a discretized version of (2.2) may have resonant solutions which deviate unboundedly in time from the corresponding solution of (2.1). On the other hand, this deviation tends to zero at the correct rate for any fixed time as space and time discretization intervals tend to zero (Richtmyer and Morton, 1967).

5. The primitive equations

We describe a simplified set of baroclinic primitive equations. We consider these equations to illustrate how the scheme of Beam and Warming (1976), devised for hyperbolic systems, can be generalized to the baroclinic equations. These equations consist of a hyperbolic system with more variables than equations, supplemented by constraints (prognostic and diagnostic equations).

We use the equations of conservation of mass, longitudinal and latitudinal momentum. The hydrostatic approximation, as well as the ideal gas law, are also used. The humidity equation is suppressed, and for simplicity the potential temperature is assumed constant in space and time; thus there are no diabatic processes. Some terms are neglected from the equations so that the total energy is not conserved: this makes no difference for our purposes. The equations are written in divergence form, which greatly simplifies development of the implicit scheme in Section 6.

The standard sigma coordinates are employed. Zero vertical velocity boundary conditions are used at the top and bottom of the atmosphere; i.e., \( \sigma = 0 \) for \( \sigma = 0 \) and \( \sigma = 1 \), which correspond to \( P_{top} = 10 \) mb and \( P_{bottom} = P(x, t) \) in our model. We use the notation

\[
\begin{align*}
\Pi & = \Pi(x, t) = p_b - p_r \\
\sigma & = (p - p_b)/\Pi \\
\dot{\sigma} & = d\sigma/dt \\
u & = \frac{dx}{dt} \\
v & = \frac{dy}{dt}
\end{align*}
\]

(5.1)
We use the following equations on a vertical slice [see Kalnay and Hoitsma (1979)].

\[ \frac{\partial \Pi}{\partial t} + \frac{\partial \Pi u}{\partial x} + \frac{\partial \Pi \sigma}{\partial \sigma} = 0, \]  
\[ \frac{\partial \Pi u}{\partial t} + \frac{\partial \Pi u^2}{\partial x} + \frac{\partial \Pi \sigma u}{\partial \sigma} + \Pi \left( \frac{\partial \phi_b}{\partial x} + c_p \frac{\partial}{\partial x} p_b \right) - f \Pi v = 0, \]  
\[ \frac{\partial \Pi v}{\partial t} + \frac{\partial \Pi v w}{\partial x} + \frac{\partial \Pi \sigma v}{\partial \sigma} = 0, \]  
\[ \frac{\partial \Pi}{\partial \sigma} = 0. \]  

Equation (5.2a) represents the conservation of mass. The form of the pressure gradient term

\[ \Pi \left( \frac{\partial \phi_b}{\partial x} + c_p \frac{\partial}{\partial x} p_b \right) \]

in the equation of conservation of longitudinal momentum may be easily derived from the hydrostatic equation

\[ \frac{\partial \phi}{\partial p^*} = -c_p \theta, \]  

where \( \theta \) is the (constant) potential temperature. Here \( \phi \) is the geopotential \( \phi = gz \), and \( \phi_b \), the geopotential at the “bottom” of the atmosphere, is specified by the orography. The Coriolis parameter is denoted by \( f \). Note that the Coriolis force was suppressed from (5.2c), the equation of conservation of latitudinal momentum. Eq. (5.2d) is stated explicitly because we will regard \( \Pi \) as a function of \( t, x, \) and \( \sigma \).

We define the momentum variables

\[ U = \Pi u \]
\[ V = \Pi v \]
\[ S = \Pi \sigma \]

and the column vectors

\[ W = (\Pi, U, V)^T \]
\[ \tilde{W} = (\Pi, U, V, S)^T \]

Note that \( \tilde{W} \) differs from \( W \) in that it contains the additional diagnostic variable \( S \).

Equations (5.2a,b,c) may be written compactly as

\[ W_t + L(W) = 0. \]  

We introduce notations for the parts of \( L(W) \) associated with horizontal and vertical motion, so that

\[ L(W) = L_h(W) + L_v(\tilde{W}), \]

where

\[ L_h(W) = \frac{\partial}{\partial x} \left( \begin{array}{c} U \left( \frac{U^2}{\Pi} \right) + \left( \Pi \left( \frac{\partial \phi_b}{\partial x} + c_p \frac{\partial}{\partial x} p_b \right) \right) \\ + \left( \frac{\partial}{\partial x} \right) \end{array} \right), \]  
\[ L_v(\tilde{W}) = \frac{\partial}{\partial \sigma} \left( \begin{array}{c} U \left( \frac{US}{\Pi} \right), \frac{V S}{\Pi} \end{array} \right). \]  

The equations to be solved are given by (5.6) and (5.2d).

6. The time discretization of the scheme

We describe briefly the Beam–Warming method for the model system (6.1) in conservation form, which contains only prognostic variables \( W \)

\[ W_t + \frac{\partial}{\partial x} F(W) + \frac{\partial}{\partial z} G(W) = 0. \]  

We apply the Crank–Nicolson time discretization to (6.1) obtaining for \( W^n = W(t = t_n) \)

\[ \frac{W^{n+1} - W^n}{\Delta t} + \frac{\partial}{\partial x} \frac{F(W^{n+1}) + F(W^n)}{2} + \frac{\partial}{\partial z} \frac{G(W^{n+1}) + G(W^n)}{2} = O(\Delta t^2). \]  

We introduce the notation \( \Delta^n W = W^{n+1} - W^n \), and perform the linearizations

\[ F(W^{n+1}) = F(W^n) + \mathbf{A}^n \Delta^n W + O(\Delta t^2), \]  
\[ G(W^{n+1}) = G(W^n) + \mathbf{B}^n \Delta^n W + O(\Delta t^2), \]  

where \( \mathbf{A}^n, \mathbf{B}^n \) are the Jacobian matrices

\[ \mathbf{A}^n = \frac{\partial F(W^n)}{\partial W}, \quad \mathbf{B}^n = \frac{\partial G(W^n)}{\partial W}. \]

Substituting (6.3) in (6.2), we obtain the linear system in \( \Delta^n W \)

\[ \Delta^n W + \frac{\Delta t}{2} \frac{\partial}{\partial x} (\mathbf{A}^n \Delta^n W) + \frac{\Delta t}{2} \frac{\partial}{\partial z} (\mathbf{B}^n \Delta^n W) = -\Delta t \frac{\partial F^n}{\partial x} - \Delta t \frac{\partial G^n}{\partial z} + O(\Delta t^3). \]  

Neglecting the error term on the rhs, Eq. (6.4) defines a scheme that is second-order in time. The linearization employed to derive this scheme is particularly simple because Eq. (6.1) is in conservation form.

Replacing the derivatives in (6.4) by differences, one obtains a linear system that can be solved numerically. Because it involves both spatial dimensions, the size of this system is too large for practical computations. A crucial idea in the method is to factorize the operator.
on the lhs of (6.4) into operators which involve only one spatial dimension. Because we are using $\Delta^n W$ (instead of $W^{n+1}$) as unknown, this factorization is accomplished by adding a $O[(\Delta t)^2]$ term and the second-order accuracy of the scheme in time is preserved:

\[
\begin{align*}
\Delta^n W + \frac{\Delta t}{2} \frac{\partial}{\partial x} (A^n \Delta^n W) + \frac{\Delta t}{2} \frac{\partial}{\partial z} (B^n \Delta^n W) \\
= \left( I + \frac{\Delta t}{2} \frac{\partial}{\partial x} A^n \right) \left( I + \frac{\Delta t}{2} \frac{\partial}{\partial z} B^n \right) \Delta^n W \\
- \left( \frac{\Delta t}{2} \right)^2 \frac{\partial}{\partial x} A^n \frac{\partial}{\partial z} B^n \Delta^n W. 
\end{align*}
\] (6.5)

In Eq. (6.5) and thereafter we use the understanding that the derivatives, \( \partial/\partial x \) and \( \partial/\partial z \), act on all quantities which follow these symbols. The last term in Eq. (6.5) is \( O[(\Delta t)^2] \). Thus, from Eq. (6.4) we obtain the two-level, spatially factored scheme:

\[
\left( I + \frac{\Delta t}{2} \frac{\partial}{\partial x} A^n \right) \left( I + \frac{\Delta t}{2} \frac{\partial}{\partial z} B^n \right) \Delta^n W
= -\Delta t \frac{\partial F^n}{\partial x} - \Delta t \frac{\partial G^n}{\partial z}. 
\] (6.6)

Replacing the derivatives in (6.6) by differences, we obtain sets of one-dimensional linear problems along lines of constant \( x \) and \( z \), which can be solved efficiently. The procedure is analogous to the alternate direction implicit method (ADI), but it is second-order accurate in time. The systems in (6.6) can be solved because there are as many equations as unknowns since Eq. (6.1) contains only prognostic variables.

We indicate how to adapt the method of Beam and Warming to Eqs. (5.6) and (5.2d). The hydrostatic approximation embedded in the baroclinic primitive equations couples horizontal derivatives at all points of vertical columns. Thus our main difficulty is to extend the factorization technique which is essential for the efficiency of the method.

Applying the Crank–Nicolson time discretization to (5.6) we obtain

\[
\begin{align*}
\frac{W^{n+1} - W^n}{\Delta t} + \frac{L_1(W^{n+1}) + L_1(W^n)}{2} \\
+ \frac{L_2(W^{n+1}) + L_2(W^n)}{2} = O[(\Delta t)^2]. 
\end{align*}
\] (6.7)

The linearization yields the scheme

\[
\begin{align*}
\Delta^n W + \frac{\Delta t}{2} \left( A^n + \frac{\partial}{\partial x} B^n \right) \Delta^n W + \frac{\Delta t}{2} \frac{\partial}{\partial \sigma} \hat{C}^n \Delta^n W \\
= -\Delta t \left[ L_1(W^n) + L_2(W^n) \right] 
\end{align*}
\] (6.8)

where, from (5.7)

\[
\left( A^n + \frac{\partial}{\partial x} B^n \right) = \frac{\partial L_1(W^n)}{\partial W}, \quad \hat{C}^n = \frac{\partial L_2(W^n)}{\partial W} \tag{6.8a,b}
\]

with

\[
A^n = \begin{pmatrix} 0 & 0 & 0 \\ \frac{\partial \phi}{\partial x} & 0 & -f \\ 0 & 0 & 0 \end{pmatrix}, \tag{6.8c}
\]

\[
B^n = \begin{pmatrix} 0 & 0 & 1 \\ R \beta^{n+1}(p \beta^{-1} - (u \cdot u)^2) & 2u & 0 \\ -u v & u & v \end{pmatrix}, \tag{6.8d}
\]

\[
\hat{C}^n = \begin{pmatrix} 0 & 0 & 0 \\ -\hat{\sigma} u & 0 & u \\ -\hat{\sigma} v & 0 & v \end{pmatrix}. \tag{6.8e}
\]

Because the pressure gradient term in Eq. (5.6) is not in conservation form, we derive the formula for the elements in the second row of (6.8c) and (6.8d):

\[
L_1(W^{n+1})_2 - L_1(W^n)_2
\]

\[
\approx \frac{\partial}{\partial x} \left[ \frac{U^n}{\Pi^n} \Delta \Pi + 2 \frac{U^n}{\Pi^n} \Delta \Pi U \right] + \frac{\partial \phi}{\partial x} \Delta \Pi\Pi
\]

\[
+ c \beta \left[ \Delta \Pi \frac{\partial}{\partial x} (p \beta^{-1} - (\kappa(p \beta^{-1}))^{-1} \Delta \Pi) \right]
\]

\[
- f \Delta \Pi V = \frac{\partial \phi}{\partial x} \Delta \Pi \Pi - f \Delta \Pi V + \frac{\partial}{\partial x} [-((u \cdot u)^2 \Delta \Pi)
\]

\[
+ 2u \Delta \Pi U] + R \beta \frac{\partial}{\partial x} [\Pi^n(p \beta^{-1})^{-1} \Delta \Pi].
\]

The factorization of Eq. (6.8) is not straightforward, because of the presence of the diagnostic variable \( \Delta \Pi \). To circumvent this difficulty, we introduce the \( 3 \times 3 \) and \( 3 \times 4 \) matrices

\[
\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{i} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{6.9}
\]

From (6.8) we obtain a factorized scheme for (5.6), which is second-order accurate in time.

\[
\left[ I + \frac{\Delta t}{2} \left( A^n + \frac{\partial}{\partial x} B^n \right) \right] \left[ \hat{I} + \frac{\Delta t}{2} \frac{\partial}{\partial \sigma} \hat{C}^n \right] \Delta^n W
\]

\[
= -\Delta t \left[ L_1(W^n) + L_2(W^n) \right]. \tag{6.10a}
\]

\[
\frac{\partial}{\partial \sigma} \Delta \Pi \Pi = 0. \tag{6.10b}
\]

Here the constraint (6.10b) comes from Eq. (5.2d).

In order to proceed, suppose for now that \( \partial/\partial x \) and \( \partial/\partial \sigma \) in Eq. (6.10) are approximated by three-point difference formulas, on a mesh with \( M \) points in the \( x \) direction and \( K \) points in the \( \sigma \) direction. Details of our spatial discretization are described in Section 7.
Equation (6.10) is solved as follows:

I. **Horizontal sweep.** Solve $K$ equations of the type

$$
\left[ I + \frac{\Delta t}{2} \left( A^n + \frac{\partial}{\partial x} B^n \right) \right] \left( \begin{array}{c} \Delta U \\ \Delta V \\
\end{array} \right) = -\Delta t \left[ L_0(W^n) + L_2(W^n) \right] \tag{6.11}
$$

for the auxiliary vectors $\Delta U$, $\Delta V$. Note that $\Delta U$ is a quantity which depends not only on $x$ but also on $\sigma$. Each equation requires solving a linear system involving only the horizontal dimension. Taking into account the periodicity in $x$ and the three-point formula we note that the matrices of these systems have the form

$$
\begin{pmatrix}
\alpha_1 & \gamma_1 & \cdots & 0 & \beta_1 \\
\beta_2 & \alpha_2 & \gamma_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \beta_3 & \alpha_3 & \cdots & 0 \\
\gamma_M & \beta_M & \alpha_M & \cdots & 0 \\
\end{pmatrix}
\tag{6.12}
$$

where $\alpha_i$, $\beta_i$, $\gamma_i$ are $3 \times 3$ blocks.

II. **Vertical sweep.** Solve $M$ equations of the type

$$
\left[ I + \frac{\Delta t}{2} \frac{\partial}{\partial \sigma} \tilde{\sigma} \right] \Delta^\sigma W = \Delta W \\
\frac{\partial}{\partial \sigma} \Delta^\sigma \Pi = 0
$$

(6.13)

In this step Eq. (6.10b) is used explicitly. As we will see, it has the effect of defining the prognostic quantity $\Delta^\sigma S$ in such a way as to extract a $\sigma$-independent $\Delta^\sigma \Pi$ from the auxiliary $\sigma$-dependent $\Delta \Pi$. Eq. (6.13) may be rewritten as

$$
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
\Delta^\sigma \Pi \\
\Delta^\sigma U \\
\Delta^\sigma V \\
\Delta^\sigma S \\
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 0 & 1 \\
\Delta U \\
\Delta V \\
0 \\
\end{pmatrix}
\tag{6.14}
$$

The first and last rows of (6.14) are, respectively,

$$
\Delta^\sigma \Pi + \frac{\Delta t}{2} \frac{\partial}{\partial \sigma} \Delta^\sigma S = \Delta \Pi, \tag{6.15a}
\frac{\partial}{\partial \sigma} \Delta^\sigma \Pi = 0. \tag{6.15b}
$$

Integrating (6.15a) from top to bottom and taking into account (6.15b), as well as the boundary conditions $S = 0$ for $\sigma = 0$, $\sigma = 1$, we obtain the prognostic equation for $\Pi$

$$
\Delta^\sigma \Pi = \int_0^1 \Delta \Pi d\sigma. \tag{6.16}
$$

Substituting (6.16) in (6.15a) and integrating from 0 to $\sigma$, we obtain the diagnostic equation for $\Delta^\sigma S$

$$
\Delta^\sigma S(\sigma) = \frac{2}{\Delta t} \int_0^\sigma (\Delta \Pi - \Delta^\sigma \Pi) d\sigma. \tag{6.17}
$$

Now the second and third row of (6.14) can be rewritten as

$$
\begin{pmatrix}
1 + \frac{\Delta t}{2} \frac{\partial}{\partial \sigma} \tilde{\sigma} & \cdots & 0 \\
m. \vdots & \ddots & \vdots \\
0 & \cdots & 1 \\
\end{pmatrix}
\begin{pmatrix}
\Delta^\sigma U \\
\Delta^\sigma V \\
\Delta^\sigma S \\
\end{pmatrix}
= \begin{pmatrix}
\frac{\Delta t}{2} \left( \frac{\partial}{\partial \sigma} (\tilde{\sigma} \Delta^\sigma \Pi - \Delta^\sigma \Pi) \right) + \Delta U \\
\frac{\Delta t}{2} (\tilde{\sigma} \Delta^\sigma \Pi - \Delta^\sigma \Pi) + \Delta V \\
\end{pmatrix}. \tag{6.18a}
$$

Thus we have to solve $M$ equations in each of (6.18a,b). Each equation requires solving a linear system involving only the vertical dimension. The matrices of these systems are tridiagonal.

The solution of Eq. (6.10) requires for the horizontal sweep the solution of $K$ periodic block-tridiagonal systems with size $3M$, and for the vertical sweep the solution of $M$ pairs of tridiagonal systems with size $K$. The overall cost is entirely dominated by the horizontal sweep. Experiments with barotropic models (our own and in Fairweather and Navon, 1980; Gilliland 1981) indicate that the block-tridiagonal systems can be solved efficiently.

Finally, we emphasize that the choice of the higher order term added on the l.h.s. of (6.4) to obtain (6.6) is crucial in the algorithm. In Beam–Warming's original scheme, a different choice yields an equation similar to (6.6) but with horizontal and vertical factors reversed. We do not know how to combine this reversed equation with (5.2d); thus our choice is dictated by the existence of the diagnostic variable $\tilde{\sigma}$.

III. Update $\Pi$, $u$, $v$, $\tilde{\sigma}$. We apply a fourth-order horizontal Shapiro (1970) filter to the quantities $\Delta^\sigma \Pi$, $\Delta^\sigma U$, $\Delta^\sigma V$, $\Delta^\sigma S$. Then it is easy to compute $\Pi^{n+1}$, $U^{n+1}$, $V^{n+1}$, $S^{n+1}$ and $u^{n+1}$, $v^{n+1}$, $\tilde{\sigma}^{n+1}$.

Because the filter is applied to the increments $\Delta^\sigma \tilde{W}$ as opposed to the basic quantities $\tilde{W}^n$, this filtering is very mild. It preserves the accuracy of the scheme (fourth–order horizontally, second–order in time). It does not have any effect on steady states even for small scales. The need for this filtering arose in our earlier experiments with a barotropic model on the sphere. In that model this filtering was applied both latitudinally and longitudinally to stabilize the scheme near
the poles. In their simulation of shocks, Beam and Warming (1976) also employed smoothing to stabilize their scheme in the presence of such nonlinear phenomena.

7. The spatial discretization of the scheme

The vertical $\sigma$ interval $[0, 1]$ is subdivided in $K$ equal intervals with $\Delta \sigma = 1/K$. We use an unstaggered grid: all the variables $u$, $v$, $\sigma$, as well as the $\Delta$-variables are located at the center of the $K$ layers. The use of staggered grids does not seem practical, because our scheme is implicit and fourth-order accurate horizontally.

a. The horizontal discretization

Following Beam and Warming (1976), we show how to discretize horizontal derivatives using the fourth-order accurate Padé formula, written symbolically as

$$\frac{df}{dx} \approx \frac{1}{\Delta x} \frac{\mu \delta}{1 + \delta^2/6} f.$$  \hspace{1cm} (7.1a)

Here the symbols $\mu$ and $\delta$ indicate the operations

$$\mu f_i = \frac{1}{6} (f_{i+1/2} + f_{i-1/2}), \quad \delta f_i = f_{i+1/2} - f_{i-1/2}.$$  

By eliminating the denominator in Eq. (7.1a) we obtain the expression in operator form

$$(1 + \delta^2/6) f' = (1/\Delta x) \mu \delta f$$

which yields the following interpretation of (7.1a)

$$\frac{\Delta f_{i+1}}{\delta} + \frac{3 \Delta f_i}{\delta} + \frac{\Delta f_{i-1}}{\delta} = \frac{1}{\Delta x} (f_{i+1} - f_{i-1}).$$ \hspace{1cm} (7.1b)

We see that to compute the derivative of $f$ through (7.1), first the difference on the rhs of (7.1b) is formed, then the linear system on the lhs of (7.1b) is solved. This system involves a cyclic tridiagonal matrix, and it can be solved very efficiently [see Temperton (1975)]. The three-point formula (7.1b) is employed to compute the horizontal derivatives on the rhs of Eq. (6.10a).

Formula (7.1) is also used to discretize the horizontal derivatives in Eq. (6.11):

$$\left[ I + \frac{\Delta t}{2} \left( A^n + \frac{1}{\Delta x} \frac{\mu \delta}{1 + \delta^2/6} B^n \right) \right] \Delta W = -\Delta t L(\check{W}^n).$$  \hspace{1cm} (7.2a)

Eliminating the denominator, we obtain

$$\left\{ \left( 1 + \frac{\delta^2}{6} \right) I + \frac{\Delta t}{2} \left[ \left( 1 + \frac{\delta^2}{6} \right) A^n + \frac{1}{\Delta x} \mu \delta B^n \right] \right\} \Delta W = -\Delta t \left( 1 + \frac{\delta^2}{6} \right) L(\check{W}^n)$$

or

$$\frac{\Delta W_{i+1}}{\delta} + \frac{3 \Delta W_i}{\delta} + \frac{\Delta W_{i-1}}{\delta}$$

$$+ \frac{\Delta t}{2} \left( \frac{1}{6} A_{i-1}^n \Delta W_{i-1} + \frac{2}{3} \Delta W_i + \frac{1}{6} A_{i+1}^n \Delta W_{i+1} \right)$$

$$+ \frac{\Delta t}{2 \Delta x} \left( -B^n_{i-1} \Delta W_{i-1} + B^n_{i+1} \Delta W_{i+1} \right)$$

$$= -\Delta t \left[ \frac{1}{6} L(\check{W}^n)_{i-1} + \frac{2}{3} L(\check{W}^n)_i + \frac{1}{6} L(\check{W}^n)_{i+1} \right].$$  \hspace{1cm} (7.2b)

Thus, to compute $\Delta W$ in (7.2), we first compute the rhs of Eq. (7.2b), then the linear system on the lhs of (7.2b) is solved. If we write the unknown vector $\Delta W$ as

$$\Delta W = (\Delta U_1, \Delta U_1, \Delta V_1, \Delta U_2, \Delta U_2, \Delta V_2, \cdots)^T,$$

the system in (7.2b) involves a cyclic block-tridiagonal matrix of the form (6.12) with

$$\alpha_i = \frac{1}{6} \left( I + \frac{\Delta t}{2} A_{i-1}^n \right),$$  \hspace{1cm} (7.3a)

$$\gamma_i = \frac{1}{6} \left( I + \frac{\Delta t}{2} A_{i+1}^n \right) + \frac{\Delta t}{2 \Delta x} B_{i+1}^n, \quad (7.3b)$$

$$\beta_i = \frac{1}{6} \left( I + \frac{\Delta t}{2} A_{i-1}^n \right) - \frac{\Delta t}{2 \Delta x} B_{i-1}^n, \quad \beta_k$$

where $A^n$ and $B^n$ are given by Eq. (6.8c,d). This system is solved using the $L - U$ decomposition for cyclic block-tridiagonal matrices. (More details may be found in Isaacson and Keller (1966, Chap. 2, Section 3.3).)

Both the explicit three-point difference formula and the Padé derivative formula yield systems of the type (6.12). Thus fourth-order accuracy is achieved by the latter at essentially no extra cost. This gain is possible because the form of the lhs of (7.2a) allows a simple elimination of the Padé denominator. This elimination would not be so simple for equations in advection form, for which this lhs has the form

$$\left[ I + \frac{\Delta t}{2} \left( A^n + D^n \frac{\partial}{\partial x} E^n \right) \right] \Delta W.$$  \hspace{1cm} (7.2a)

b. The vertical discretization and boundary conditions

To solve Eq. (6.18) we use the second-order accurate formula

$$\frac{df}{d\sigma} \approx \frac{1}{2 \Delta \sigma} (f_{k+1} - f_{k-1}).$$

Substituting this formula in (6.18a), we get the system

$$\begin{pmatrix} \alpha_1 & \gamma_1 & 0 & \cdots & \cdots & \cdots \\ \beta_2 & \alpha_2 & \gamma_2 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ \beta_K & \cdots & \cdots & \cdots & \cdots & \alpha_K \end{pmatrix} \begin{pmatrix} \Delta U_1 \\ \Delta U_2 \\ \vdots \\ \Delta U_K \end{pmatrix} = \begin{pmatrix} \Delta \sigma \end{pmatrix}.$$  \hspace{1cm} (7.4)
where
\[
\begin{align*}
\alpha_1 &= 1 + \frac{\Delta t}{4\Delta \sigma} \sigma_1 \\
\alpha_2 &= 1 \\
\alpha_3 &= 1 \\
\vdots \\
\alpha_{K-1} &= 1 \\
\alpha_K &= 1 - \frac{\Delta t}{4\Delta \sigma} \sigma_K \\
\gamma_1 &= \frac{\Delta t}{4\Delta \sigma} \sigma_2 \\
\gamma_2 &= \frac{\Delta t}{4\Delta \sigma} \sigma_3 \\
\vdots \\
\gamma_{K-1} &= \frac{\Delta t}{4\Delta \sigma} \sigma_K \\
\beta_2 &= -\frac{\Delta t}{4\Delta \sigma} \sigma_1 \\
\beta_3 &= -\frac{\Delta t}{4\Delta \sigma} \sigma_2 \\
\vdots \\
\beta_K &= -\frac{\Delta t}{4\Delta \sigma} \sigma_{K-1}
\end{align*}
\] (7.5a)

The vertical index \( k \) corresponds to \( \sigma = (k - 0.5)/K \), \( k = 1, 2, \ldots, K \).

The formulas for \( \alpha_i \) and \( \alpha_K \) are obtained by imposing the boundary conditions \( \dot{\sigma} = 0 \) for \( \sigma = 0 \) and for \( \sigma = 1 \). This amounts to reflecting \( \dot{\sigma} \Delta U \) and \( \dot{\sigma} \Delta V \) with opposite sign to the center of a fictitious layer adjoining the boundary. The scheme (7.4) obtained in this way is second-order accurate vertically.

To evaluate the integral in Eq. (6.16), we use the integration formula
\[
\Delta^* \Pi = \Delta \sigma \sum_{k=1}^{K} \Delta \Pi_k.
\] (7.6)

Similarly the integral in Eq. (6.17) is evaluated by the formulas
\[
\begin{align*}
\Delta^\sigma S_1 &= \frac{2}{\Delta t} \left\{ \frac{\Delta \sigma}{2} \left[ \Delta \Pi_1 - \Delta^\sigma \Pi \right] \right\} \\
\Delta^\sigma S_K &= \Delta^\sigma S_{K-1} \\
&+ \frac{2}{\Delta t} \left\{ \frac{\Delta \sigma}{2} \left[ (\Delta \Pi_k - \Delta^\sigma \Pi) + (\Delta \Pi_{k-1} - \Delta^\sigma \Pi) \right] \right\},
\end{align*}
\] (7.7)

The formulas (7.6) and (7.7) are second-order accurate. They are compatible with the implementation of the derivative and of the boundary conditions used in Eq. (7.5), in the sense that mass is conserved exactly, i.e., \( \Delta x \sum_{i=1}^{M} \Pi_i \) does not change in time.

8. Results

Our implicit scheme was developed using the technique described in Sections 2, 3 and 4, which was instrumental in isolating programming errors. We now indicate how the method was used to verify accuracy of the implicit scheme.

In order to construct functions analogous to Eq. (3.4), we prescribe the following solutions:
\[
\begin{align*}
\Pi &= \Pi_0(1 + \Pi_1 \sin(\alpha x + \gamma t + \delta)) \\
u &= u_0 + u_1 \sin(\alpha x + \beta \sigma + \gamma t + \delta) \\
v &= v_0 + v_1 \sin(\alpha x + \beta \sigma + \gamma t + \delta) \\
\dot{\sigma} &= s_1 \sin(\alpha x + \beta \sigma + \gamma t + \delta) \sin(3\pi \sigma)
\end{align*}
\]

Parameter values were chosen as follows:
\[
\begin{align*}
\Pi_0 &= 990 \text{ mb} \\
u_0 &= 10 \text{ m s}^{-1} \\
v_0 &= 4 \text{ m s}^{-1} \\
\Pi_1 &= 0.01 \\
u_1 &= 4 \text{ m s}^{-1} \\
v_1 &= 2 \text{ m s}^{-1} \\
s_1 &= 10^{-6} \text{ s}^{-1} \\
\theta_0 &= 293 \text{ K}/(1000 \text{ mb})' \\
\kappa &= 1 - 1/1.4 \\
\alpha &= a/\rho \text{ earth} \cos(\text{latitude}) \\
\beta &= b\pi \\
\gamma &= 2\pi c/ \left(10^6 \text{ s} \right) (10^6 \text{ s} \sim 10 \text{ days}).
\end{align*}
\]

The numbers \( a, b, c \) are small integers, whose values

<table>
<thead>
<tr>
<th>Time step (h)</th>
<th>1</th>
<th>4</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error in</td>
<td>( \Pi - \Pi_0 )</td>
<td>( u )</td>
<td>( v )</td>
</tr>
<tr>
<td>5</td>
<td>( 3.8 \times 10^{-4} )</td>
<td>( 13.0 \times 10^{-4} )</td>
<td>( 6.8 \times 10^{-3} )</td>
</tr>
<tr>
<td>10</td>
<td>( 9.6 \times 10^{-5} )</td>
<td>( 2.9 \times 10^{-5} )</td>
<td>( 1.6 \times 10^{-3} )</td>
</tr>
<tr>
<td>20</td>
<td>( 2.4 \times 10^{-5} )</td>
<td>( 7.1 \times 10^{-4} )</td>
<td>( 4.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>40</td>
<td>( 1.8 \times 10^{-4} )</td>
<td>( 1.1 \times 10^{-4} )</td>
<td>( 1.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>Number of vertical intervals</td>
<td>5</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>
Table 2. Relative rms errors for a horizontally moving solution after 12 h. \((a = c = 1, b = 0, s_i = 0)\).

<table>
<thead>
<tr>
<th>Time step ((h))</th>
<th>(1)</th>
<th>(\frac{1}{4})</th>
<th>(\frac{1}{16})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error in</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>(\Pi - \Pi_0)</td>
<td>(3.4 \times 10^{-3})</td>
<td>(1.9 \times 10^{-4})</td>
<td>(1.2 \times 10^{-5})</td>
</tr>
<tr>
<td>(u)</td>
<td>(8.3 \times 10^{-4})</td>
<td>(4.9 \times 10^{-5})</td>
<td>(3.0 \times 10^{-6})</td>
</tr>
<tr>
<td>(v)</td>
<td>(3.9 \times 10^{-4})</td>
<td>(2.0 \times 10^{-5})</td>
<td>(1.2 \times 10^{-6})</td>
</tr>
<tr>
<td>(\bar{\sigma})</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

are indicated in the tables. The latitude was 45° and we set \(\phi_0 = 0\).

The phase \(\bar{\sigma}\) was taken to be zero, except when \(\alpha = 0\). In this case, a nonzero value of \(\bar{\sigma}\) was used to prevent \(\bar{\sigma}\) from behaving quadratically in \(\sigma\) at the boundaries. If this is not done, the rate of decay of the error in the vertical direction attains its asymptotic value very slowly.

The second-order vertical accuracy and the fourth-order horizontal accuracy are clearly displayed in Tables 1 and 2.

The overall second-order accuracy of the scheme is indicated in Table 3.

We succeeded in constructing tables that exhibit the correct behavior of truncation errors only after a number of programming errors were removed. Since then, the program has been tested extensively and no new errors found.

To test the stability of the scheme, the program was run for periods of 6 days at latitudes of 0°, 45°, 88.5°, 89.5° and meshes of up to 64 points. In these experiments, we did not impose the prescribed solutions, but it was necessary to introduce a relative vertical slope of 0.1 in the potential temperature in order to simulate a stable stratification. Runs were also made with orography \(\phi_0\) simulating continents. No stability problems were observed.

Unfortunately, we were unable to find a global scale phenomenon that could be simulated by the baroclinic primitive equations on a vertical slice. Thus further tests of our implicit scheme will be made on a three-dimensional program incorporating the thermodynamic equation. We expect that the favorable properties of the scheme observed so far in two dimensions can be preserved, and will be described in a future paper.

Acknowledgments. The author expresses gratitude to Stephen Cohn, Michael Ghil, Milton Halem, Amiram Harten, Eugene Isaacson, Eugenia Kalnay and Gideon Zwas for many enlightening discussions. Appreciation also is expressed to the referees for several suggestions on the manuscript. This research was supported in part by NASA Grant NSG 5034 at CIMS and by FINEP 4.3.82.0719.00 at PUC/RJ.

REFERENCES


