Journal of Computational Physics **177,** 59–94 (2002)
doi:10.1006/jcph.2002.7003, available online at http://www.idealibrary.com on IDEAL®

A Two-Level Time-Stepping Method for Layered Ocean Circulation Models

Robert L. Higdon

Department of Mathematics, Oregon State University, Corvallis, Oregon 97331-4605
E-mail: higdon@math.orst.edu

Received February 14, 2001; revised September 14, 2001

This paper describes a numerical method with two time levels for solving partial differential equations that represent layered models of ocean circulation. The method is designed to be used with a barotropic–baroclinic splitting that separates the fast and slow motions into subsystems that are solved by different techniques. With this method, some of the dependent variables are predicted and then corrected. After the initial prediction steps, all steps involve centered differencing and averaging about the midpoint of the time interval in question. The Coriolis and pressure terms are evaluated at the same time levels to avoid a first-order truncation error in the geostrophic balance between those terms. Compared to the three-level leapfrog method that is widely used in geophysical fluid dynamics, the present method does not admit a computational mode, and the maximum permissible time step is at least twice as large. In addition, with the two-level method it is possible to use a nearly nonoscillatory advection algorithm to solve the equations for mass and momentum. In a simple test problem, the present method gives less phase error than the leapfrog method and two other methods, and it does not give any amplitude error. The differences are especially large when comparing the present method to the leapfrog method. The two-level method also gives good results in a nonlinear test problem involving vanishing layers at the top and bottom of the fluid domain and an interface moving along sloping bottom topography.

Key Words: stability analysis; time-stepping; barotropic–baroclinic splitting; layered ocean circulation models.

1. INTRODUCTION

The goal of this paper is to develop a numerical method with two time levels for solving partial differential equations that describe layered models of ocean circulation. Such models admit motions varying on a wide range of time scales, and for the sake of computational efficiency the fast and slow motions are often split into subsystems that are solved by different

0021-9991/02 $35.00

techniques. The present time-stepping method is developed for use with such splittings. In addition, the method provides a framework for incorporating nearly nonoscillatory advection schemes for solving the equations for conservation of mass and momentum. More generally, the time-stepping method can be applied to the shallow-water equations for a single-layer fluid, and it is proposed as an alternative to the three-level leapfrog method which is widely used in geophysical fluid dynamics.

The physical context of this work is the following. We consider ocean circulation models based on isopycnic coordinates, for which the vertical coordinate is potential density (density adjusted adiabatically to a reference pressure) or a related quantity. A vertical discretization in this coordinate system amounts to dividing the fluid into layers that are approximately immiscible, which has advantages for modeling the subtle exchanges that occur between the upper and lower regions of the ocean (Bleck *et al.* [3]). An isopycnic model can be regarded as a stack of shallow-water models, with mechanisms for communication between layers.

For a splitting of the fast and slow motions, we use a splitting introduced by Bleck and Smith [4], with modifications proposed by Higdon and de Szoeke [12] to improve the stability of the algorithm. With this splitting, the fast motions are represented by a two-dimensional, vertically integrated system (the “barotropic” system) that is similar to the shallow-water equations. Compared to [4], the main idea in [12] is to use a more explicit vertical averaging when deriving this system. The barotropic system models the motions of fast external gravity waves, which are similar to shallow-water waves, and it is solved either explicitly with short time steps or implicitly with a long time step. The remaining motions are much slower and are modeled by a three-dimensional system (the “baroclinic” system), which is solved explicitly with a long time step.

One option for a time-stepping scheme for such a system is to use the leapfrog method, which is a three-level method based on second-order centered differencing. This method is used, for example, in the Miami Isopycnic Coordinate Ocean Model [3, 4] to solve the baroclinic momentum equations, and it is also used in the Bryan–Cox class of z-coordinate models (e.g., [5]). However, the leapfrog method admits a computational mode consisting of rapid, grid-scale oscillations with respect to time. Essentially, with this method the computed time derivative at time t_n is a centered difference of values from times t_{n-1} and t_{n+1}, and a solution could be balanced in space but admit sawtooth oscillations in t. In a nonlinear model, these oscillations can grow and eventually cause a failure in the computation, so it is necessary to introduce some sort of time smoothing or filtering in order to prevent this from occurring (e.g., Asselin [1]). Such a process is not based on physical modeling but on computational necessity instead. In the case of the Asselin filter, the accuracy of the time discretization is reduced to first order (Durran [6]). In addition, the leapfrog method is unstable when applied to diffusion processes, so special consideration must be given to viscous terms.

If a time-stepping method involves only two time levels, then the above computational mode cannot be present. In addition, a two-level method allows the possibility of using nearly nonoscillatory advection algorithms. In a layered model, the mass conservation equation is essentially a statement about the thickness of the fluid layers, and for this equation it is necessary to use an advection algorithm which guarantees that the computed thicknesses remain nonnegative. Given the development of numerical methods to date, this requirement apparently implies that a two-level method must be used. If a two-level, nearly nonoscillatory scheme were used for the entire set of baroclinic equations, then the mass and momentum equations would be discretized in a consistent manner, and using such a scheme for the momentum equations could prevent the appearance of spurious large velocities that cause computational instability.

In an analysis of stability of barotropic–baroclinic splitting, Higdon and de Szoeke [12] considered two different schemes for advancing the baroclinic equations in time. One is the leapfrog method and the other is a two-level method based on the forward–backward method [10, 14]. As implemented in [12], the mass variable is advanced with a forward step, and the velocity is then advanced with a backward step. With this sequence of operations, the Coriolis and pressure terms are evaluated at different times. Large-scale geophysical flows are typically characterized by a balance (the "geostrophic balance") between the Coriolis effect and the horizontal pressure gradient, and with the scheme in [12] there is a first-order time truncation error in this balance. In addition, in order to provide a stable yet explicit formulation of the Coriolis terms, the Coriolis terms are implemented in [12] with a weighted average of velocities between times t_n and t_{n+1} (Wang and Ikeda [21]), and this biased averaging can have the effect of introducing damping at zero wavenumber. The Coriolis force is not a dissipative mechanism, so this damping is a numerical artifact instead of a physical process. A similar treatment of Coriolis terms is also used in a stable two-level method that Hallberg [9] developed for barotropic–baroclinic splitting. In this method all dependent variables are predicted and then corrected. This scheme uses weighted time averages of various terms, and depending on the weighting coefficients there may be a first-order error in the geostrophic balance.

With the algorithm developed in the present paper, some of the dependent variables are predicted from time t_n to time t_{n+1}; the layer thickness is then advanced to time t_{n+1}; and finally the remaining variables are corrected. After the initial prediction steps, all of the subsequent steps are based on centered differencing and averaging about the middle time $t_{n+1/2} = (t_n + t_{n+1})/2$. In a sense, the method is based on the trapezoidal method for solving initial-value problems for ordinary differential equations. With this method, the Coriolis and pressure terms are always evaluated for the same value of t. During the correction step, the Coriolis terms are given an implicit represention which can be implemented directly or with a simple iteration, depending on the structure of the spatial grid. Under the conditions of a linear stability analysis, the maximum permissible time step with this method is at least twice as large as that of the leapfrog method.

As noted by Smolarkiewicz and Margolin [18], a second-order advection algorithm requires that the advective velocity be evaluated at the intermediate time $t_{n+1/2}$, so that certain error terms are compensated appropriately. With the sequencing of steps used in the present time-stepping method, this velocity is available when the layer thickness is updated and when the momentum variables are corrected.

An outline of the paper is as follows. Section 2 describes the basic properties of the time-stepping method, mainly in the setting of the linearized shallow-water equations, and it includes comparisons to other methods in simple contexts. In Section 3, the method is extended to a barotropic–baroclinic splitting for multilayer models. This section includes a linearized analysis of stability for the case where the barotropic equations are solved exactly and the case where an implicit method of Bates [2] is used to solve the barotropic equations implicitly. Section 4 describes some aspects of implementing the time-stepping method in a nonlinear setting. Here, the multidimensional positive definite advection transport algorithm (MPDATA) of Smolarkiewicz and collaborators (e.g., [18]) is used to solve the mass and momentum equations. This section also addresses some issues related to thin layers and variable bottom topography. Some computational results involving a nonlinear test problem are described in Section 5. This problem involves thin layers at the top and bottom of the fluid domain and a fluid interface moving upward and downward along sloping bottom topography. Some remarks on computational efficiency are included at the end of that section.

2. BASIC PROPERTIES

In this section, we describe some of the basic properties of the time-stepping method developed in this paper. After giving a motivation for the scheme in Section 2.1, we describe the method as applied to the linearized shallow-water equations for a single-layer fluid. This case serves as a prototype for the nonlinear multilayer algorithm discussed in Sections 3 through 5. The discussion in the present section includes comparisons to other time-stepping methods in simple settings.

2.1. Motivation

The time-stepping method used here is motivated by the trapezoidal method for ordinary differential equations. For the equation $u'(t) = f(t, u)$, this method has the form

$$u^{n+1} = u^n + \Delta t \cdot \frac{1}{2}[f(t_{n+1}, u^{n+1}) + f(t_n, u^n)], \tag{1}$$

where u^n denotes an approximation to $u(t_n)$, and $\Delta t = t_{n+1} - t_n$.

Some desirable features of (1) are indicated by its stability region. Suppose that a numerical method is applied to the equation $u'(t) = \lambda u$, where λ is constant. The stability region for that method is the set of all complex values of $\lambda\Delta t$ such that all solutions of the discrete equation are bounded as $t \to \infty$. In the case of the continuous equation $u'(t) = \lambda u$, all solutions are bounded if and only if $\operatorname{Re}\lambda \leq 0$. The trapezoidal method (1) has an analogous property, as a calculation shows that the stability region S for that method is the left half plane. In the case $\lambda = i\omega$ where ω is real, the quantity $\lambda\Delta t$ lies on the boundary of S, and solutions of the difference equation (1) neither grow nor decay as $t \to \infty$. This corresponds to the oscillatory behavior of solutions $u(t) = Ce^{iwt}$ of the continuous problem. Similarly, if $\operatorname{Re}\lambda < 0$, then all solutions of the difference equation decay as $t \to \infty$, as do the solutions of the continuous problem.

On the other hand, the stability region for the leapfrog method,

$$u^{n+1} = u^{n-1} + 2\Delta t f(t_n, u^n), \tag{2}$$

consists entirely of the segment on the imaginary axis between the points $-i$ and i. This method is therefore unstable when applied to problems in which damping is present. However, geophysical fluid models typically include a diffusion term. One approach that has been used to resolve this problem while using the leapfrog method is to evaluate the diffusion term at time t_{n-1} instead of time t_n (e.g., Durran [6]), thus compromising the centered nature of the time differencing. In addition, the leapfrog scheme admits the troublesome computational modes discussed earlier.

In the case of the backward Euler method $u^{n+1} = u^n + \Delta t f(t_{n+1}, u^{n+1})$, the stability region is the exterior of the unit circle centered at $\lambda\Delta t = 1$. This method thus damps all motions that are damped physically, but it also damps motions that should be purely oscillatory. In the case of Runge–Kutta methods or other linear multistep methods, the stability regions are bounded and have boundaries that do not coincide with the imaginary axis (Gear [7]). For none of these methods do the cases of numerical damping and numerical oscillation coincide with the cases of physical damping and physical oscillation, as they do with the trapezoidal method.

2.2. Numerical Algorithm

The preceding properties of the trapezoidal method make it a desirable basis for constructing a two-level time-stepping method. However, the trapezoidal method is implicit, which causes difficulties with implementation in multidimensional models. In this section, we show that a prediction and correction of some of the dependent variables make it possible to avoid a fully implicit algorithm and yet maintain the general flavor of a trapezoidal time discretization. It is not necessary to predict and correct all of the dependent variables.

The algorithm is stated here in terms of the linearized shallow-water equations,

$$\begin{aligned} \frac{\partial u}{\partial t} - fv &= -g\frac{\partial h}{\partial x} \\ \frac{\partial v}{\partial t} + fu &= -g\frac{\partial h}{\partial y} \\ \frac{\partial h}{\partial t} + H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) &= 0 \end{aligned} \tag{3}$$

(Gill [8], Pedlosky [15]). Here, $u(x, y, t)$ and $v(x, y, t)$ are the x- and y-components of velocity, respectively; $h(x, y, t)$ is the perturbation in the elevation of the free surface relative to its equilibrium state; H is the mean depth of the fluid; f is the Coriolis parameter, which represents the effect of a rotating reference frame; and g is the acceleration due to gravity. Here, H and f are assumed constant, and the system has been linearized about a state of zero flow. In anticipation of the later extension of the algorithm to multilayered models, the quantities H and h will be regarded as the equilibrium thickness and the perturbation in layer thickness, respectively.

For the sake of notational simplicity, the spatial derivatives are not discretized in the following description of the time-stepping scheme.

Step 1. Predict the velocity using a forward step.

$$\begin{aligned} u^{pred} &= u^n + (f\Delta t)v^n - (g\Delta t)\frac{\partial h^n}{\partial x} \\ v^{pred} &= v^n - (f\Delta t)u^n - (g\Delta t)\frac{\partial h^n}{\partial y}. \end{aligned} \tag{4}$$

Step 2. Update the layer thickness, using an average of old and predicted velocities.

$$h^{n+1} = h^n - (H\Delta t)\frac{\partial}{\partial x}\left(\frac{u^{pred} + u^n}{2}\right) - (H\Delta t)\frac{\partial}{\partial y}\left(\frac{v^{pred} + v^n}{2}\right). \tag{5}$$

Step 3. Correct the velocity, using an implicit representation of the Coriolis terms and a time average of layer thicknesses.

$$\begin{aligned} u^{n+1} &= u^n + \frac{f\Delta t}{2}(v^{n+1} + v^n) - (g\Delta t)\frac{\partial}{\partial x}\left(\frac{h^{n+1} + h^n}{2}\right) \\ v^{n+1} &= v^n - \frac{f\Delta t}{2}(u^{n+1} + u^n) - (g\Delta t)\frac{\partial}{\partial y}\left(\frac{h^{n+1} + h^n}{2}\right). \end{aligned} \tag{6}$$

If the spatial grid is designed so that u and v are defined at the same grid points, then the Coriolis terms in Step 3 can be implemented by solving a 2×2 linear system at each grid point. On the other hand, if a staggered grid is used in which u and v are defined at different points, then the Coriolis terms can be implemented with a fixed-point iteration. This issue is discussed in Section 4.2.

The preceding algorithm consists of a first-order prediction step, followed by steps that employ centered differencing and averaging about the middle time $t_{n+1/2} = (t_n + t_{n+1})/2$. A calculation of truncation errors shows that the overall method is second-order accurate with respect to t.

The stability of the method can be described as follows. Suppose, for example, that the system (4)–(6) is discretized in space with centered second-order finite differences on the "C-grid" of Arakawa (e.g., [10, 14]). With this grid, mass variables are defined at the centers of grid cells, and values of normal velocities are defined at the centers of boundary segments of mass cells. In this situation, the preceding algorithm is stable if and only if the Courant–Friedrichs–Lewy condition

$$\left(\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta y}\right)^2 < 1 \tag{7}$$

is satisfied. Under such circumstances, the method is also nondissipative. That is, for solutions having a space–time dependence of the form $\lambda^n e^{ikx+ily}$, where λ is a constant and the superscript on λ is an exponent, we have $|\lambda| = 1$ for all (k, ℓ) and all roots λ whenever (7) is satisfied. The stability analysis will be omitted here, for the sake of brevity.

The condition (7) is less restrictive than the corresponding condition for the leapfrog method. A calculation shows that if $f = 0$, the leapfrog method is stable if and only if $(c\Delta t/\Delta x)^2 + (c\Delta t/\Delta y)^2 < 1/4$. For given Δx and Δy, the maximum allowable time step is half that of the method described above. More generally, if $f \neq 0$, then the leapfrog method is stable if and only if

$$(\nu_x \sin(k\Delta x/2))^2 + (\nu_y \sin(\ell\Delta y/2))^2 + \left[\frac{1}{2}(f\Delta t)\cos(k\Delta x/2)\cos(\ell\Delta y/2)\right]^2 < 1/4$$

for all (k, ℓ). Here, $\nu_x = c\Delta t/\Delta x$ and $\nu_y = c\Delta t/\Delta y$. In this case, the time step must be restricted further.

2.3. Comparisons to Other Methods

We next give two comparisons of the two-level method (4)–(6) with other time-stepping schemes. In the first case, we compare the dispersion relations for the leapfrog method and the two-level method. We then compare phase and amplitude errors in a simple oscillatory test problem. The second case includes comparisons to the two-level methods developed in [9], [12] for barotropic–baroclinic splittings.

2.3.1. *Dispersion Relation*

One reason for using a two-level time-stepping method is to avoid the computational modes that are allowed by the leapfrog scheme. Here we derive numerical dispersion relations and verify that the computational modes are in fact avoided with the present two-level method.

For simplicity, consider the one-dimensional shallow-water equations

$$\begin{aligned} \frac{\partial u}{\partial t} &= -g\frac{\partial h}{\partial x} \\ \frac{\partial h}{\partial t} + H\frac{\partial u}{\partial x} &= 0. \end{aligned} \tag{8}$$

Discretize in space using second-order centered differences on a staggered grid, in analogy to the two-dimensional C-grid described above. Then discretize in time using the two-level method of Section 2.2 and the leapfrog method (2). For each method, insert oscillatory solutions of the form $u(x,t) = \hat{u}(k,\omega)e^{ikx-i\omega t}$, $h(x,t) = \hat{h}(k,\omega)e^{ikx-i\omega t}$, with k and ω real. Each method is stable and nondissipative if the Courant number $\nu = c\Delta t/\Delta x$ is limited suitably, so it is no restriction to assume that the frequency ω is real.

For the two-level scheme (4)–(6), a calculation yields the dispersion relation

$$\left(\sin\frac{\omega\Delta t}{2}\right)^2 = \nu^2\left(\sin\frac{k\Delta x}{2}\right)^2, \tag{9}$$

provided $\nu < 1$, which is the stability condition for this method in the case of one space dimension. In the case of the leapfrog method, the dispersion relation is

$$\frac{1}{4}(\sin\omega\Delta t)^2 = \nu^2\left(\sin\frac{k\Delta x}{2}\right)^2, \tag{10}$$

and the stability condition is $\nu < 1/2$. If the system (8) is discretized with respect to x but not with respect to t, then an analogous calculation yields the relation

$$\left(\frac{\omega\Delta t}{2}\right)^2 = \nu^2\left(\sin\frac{k\Delta x}{2}\right)^2, \tag{11}$$

which can be compared with the preceding relations to illustrate the effects of time discretizations. For the continuous problem (8), the frequency and wavenumber are related by $\omega^2 = c^2k^2$. In each of these cases, the dispersion relation indicates the possible modes of wave propagation in the system, and the group velocity $c_g = d\omega/dk$ gives the rate at which energy is propagated (Trefethen [20]).

Figure 1 shows plots of dispersion relations for the case $\nu = 0.4$, which is a value for which the two-level and leapfrog methods are both stable. Due to aliasing, it suffices to plot the relations for $|k\Delta x| \le \pi$ and $|\omega\Delta t| \le \pi$. In the left frame, the solid curves are the graph of the dispersion relation (9) for the two-level scheme, and in the right frame, the solid curves show the dispersion relation (10) for the leapfrog scheme. The dashed curves

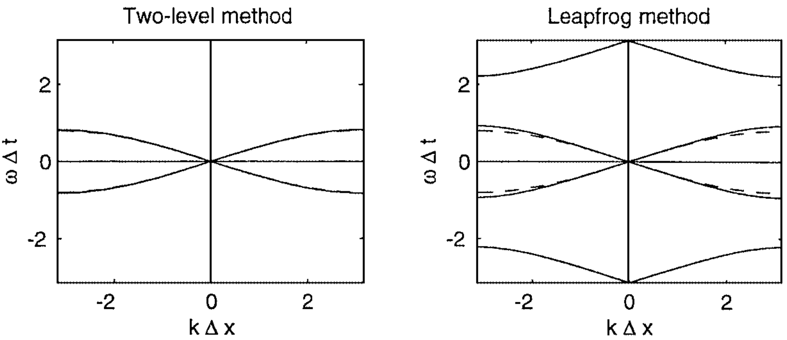

FIG. 1. Plots of dispersion relations for the two-level method and the leapfrog method, with $\nu = 0.4$. In each graph frame, the dashed curves represent the dispersion relation for the semidiscrete scheme obtained by discretizing in space but not in time.

in each frame show the dispersion relation (11) for the semidiscrete case. In the case of the leapfrog method, the lobes in the dispersion relation near $\omega\Delta t = \pm\pi$ represent computational modes which bear no relation to the modes of propagation allowed by the continuous system (8). These computational modes are not present in the dispersion relation for the two-level scheme.

In the case shown in Fig. 1, the dispersion relation for the two-level method gives a better approximation to the dispersion relation for the semidiscrete case than does the dispersion relation for the leapfrog method. This suggests that the two-level method generates less time discretization error. This observation is confirmed by the following oscillatory test problem.

2.3.2. *Phase and Amplitude Errors*

To construct a test problem for comparing phase and amplitude errors due to time-stepping, first transform the system (3) as follows. Let $c = \sqrt{gH}$ denote the speed of gravity waves, let $\eta = h/H$ denote the relative perturbation in the free-surface elevation, and replace the nondimensional ratios u/c and v/c with the symbols u and v, respectively. Then apply a Fourier transform with respect to x and y to obtain

$$\begin{aligned}\frac{\partial \hat{u}}{\partial t} - f\hat{v} &= -c(ik)\hat{\eta} \\ \frac{\partial \hat{v}}{\partial t} + f\hat{u} &= -c(i\ell)\hat{\eta} \\ \frac{\partial \hat{\eta}}{\partial t} + c(ik\hat{u} + i\ell\hat{v}) &= 0.\end{aligned} \tag{12}$$

Equivalently, consider solutions of the system (3) having the form

$$\begin{aligned}u(x,y,t) &= \hat{u}(k,\ell,t)e^{ikx+i\ell y} \\ v(x,y,t) &= \hat{v}(k,\ell,t)e^{ikx+i\ell y} \\ \eta(x,y,t) &= \hat{\eta}(k,\ell,t)e^{ikx+i\ell y}.\end{aligned} \tag{13}$$

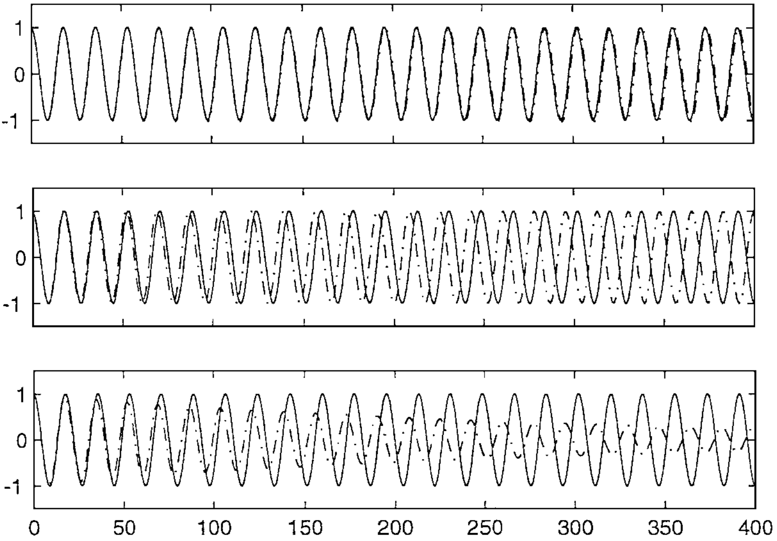

FIG. 2. Plots of the free-surface elevation for an oscillatory test problem. The horizontal scale represents the number of time steps, and the vertical scale represents the elevation relative to the maximum elevation at a fixed (x, y). In each plot, the solid curve represents the exact solution. In the top frame, the dot–dash curve shows the solution obtained with the two-level method (4)–(6). The dot–dash curve in the middle frame shows the solution obtained with the leapfrog method. The dot–dash curve in the bottom frame was obtained from the leapfrog method, combined with an Asselin filter having weight coefficients (0.05, 0.90, 0.05).

The evolution of the amplitude coefficients $\hat{u}$, $\hat{v}$, and $\hat{\eta}$ through time yields the behavior of the solutions (13) at a fixed point in space.

In the following tests, assume $f\Delta t = 0.25$, $kc\Delta t = 0.25$, and $\ell c\Delta t = 0$. (Each of these quantities is dimensionless.) If $L = 2\pi/k$ denotes a wavelength with respect to x, then $c\Delta t = L/(8\pi)$. The distance traveled by a wave in one time step is thus much less than the wavelength, and in this sense the wave is well-resolved.

Figure 2 shows plots of the free-surface elevation $\hat{\eta}(k, \ell, t)$ over 400 time steps. In each frame, the solid curve shows the exact solution to (13), assuming $\hat{\eta}(k, \ell, 0) = 1$. In the top frame, the dot–dash curve shows the solution computed with the two-level method (4)–(6). A small phase error has become noticeable by the end of the period of integration. In the middle frame, the dot–dash curve shows the solution computed with the leapfrog method. In this case, the phase error is much larger than in the top frame, and by the end of the period of integration the numerical solution is completely out of phase.

In the bottom frame in Fig. 2, the dot–dash curve shows the solution computed with the leapfrog method, with the Asselin [1] filter included. This filter, when applied to a function ϕ, can be represented in the form

$$\bar{\phi}^n := \gamma\phi^{n+1} + (1 - 2\gamma)\phi^n + \gamma\bar{\phi}^{n-1}. \tag{14}$$

Here, ϕ^n denotes the solution before filtering, $\bar{\phi}^n$ denotes the solution after filtering, and γ is a constant between 0 and 1. The smoothing operation (14) is applied to the solution at time t_n immediately after the solution at time t_{n+1} is computed, and it is often used in geophysical calculations in order to prevent the computational mode from overwhelming the computed solution. In the case shown in Fig. 2, the value $\gamma = 0.05$ is used, so that the weighting coefficients are (0.05, 0.90, 0.05). The computed solution shows a large amplitude error, in addition to the phase error. Larger values of γ have been used in geophysical calculations (see Durran [6]), and these larger values cause larger amplitude errors.

We next consider the two-level methods used by Higdon and de Szoeke [12] and Hallberg [9] in the context of a barotropic–baroclinic splitting for multilayer models. To apply these methods to the present case, we delete the terms in the baroclinic (slow) equations which involve barotropic (fast) quantities, and the result is a system having the same structure as the shallow-water equations.

In the present situation, the method of Higdon and de Szoeke [12] can be written in the form

$$\begin{aligned}
\hat{\eta}^{n+1} &= \hat{\eta}^n - i(kc\Delta t)\hat{u}^n - i(\ell c\Delta t)\hat{v}^n \\
\hat{u}^{pred} &= \hat{u}^n + (f\Delta t)\hat{v}^n - i(kc\Delta t)\hat{\eta}^{n+1} \\
\hat{v}^{pred} &= \hat{v}^n - (f\Delta t)\hat{u}^n - i(\ell c\Delta t)\hat{\eta}^{n+1} \\
\hat{u}^{n+1} &= \hat{u}^n + (f\Delta t)[a\hat{v}^{pred} + (1-a)\hat{v}^n] - i(kc\Delta t)\hat{\eta}^{n+1} \\
\hat{v}^{n+1} &= \hat{v}^n - (f\Delta t)[a\hat{u}^{pred} + (1-a)\hat{u}^n] - i(\ell c\Delta t)\hat{\eta}^{n+1}.
\end{aligned} \tag{15}$$

Here, a is a constant with $a > 0.5$. The parameter a represents a weighted time averaging of the Coriolis terms, and the method (15) is unstable if $a = 0.5$. In [12] it is shown that the method is stable for the shallow-water equations, with centered differencing on a C-grid, if and only if $a \geq 1/(1 + \sqrt{1-(f\Delta t)^2})$. In addition, the method is nondissipative for given spatial wavenumbers (k, ℓ) if and only if $a = 1/(1+\sqrt{1-\phi^2})$, where $\phi = f\Delta t\cos(k\Delta x/2)\cos(\ell\Delta y/2)$. If $f\Delta t = 0.25$ and $k = \ell = 0$, then the minimum value of a is approximately 0.508. The dot–dash curve in the upper frame in Fig. 3 shows the solution obtained with the method (15) with $a = 0.53$. Compared to the solution shown in the top frame in Fig. 2, this solution shows a greater phase error, and there is an amplitude error due to damping. Larger values of a give stronger damping.

In large-scale geophysical flows, the dominant balance is typically the geostrophic balance between the Coriolis terms and pressure gradient. In the system (12), the pressure

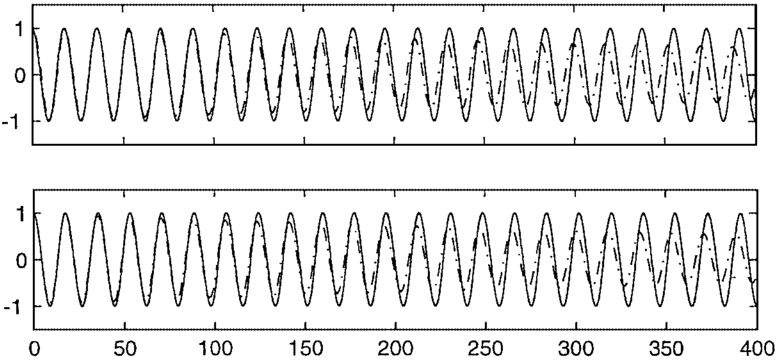

FIG. 3. Plots of free-surface elevation. The top frame shows the exact solution and the solution obtained with the method (15) of Higdon and de Szoeke [12], with $a = 0.53$. The bottom frame shows the exact solution and the solution obtained with the method (16) of Hallberg [9], with $\beta = \mu = 0.5$ and $\psi = \zeta = 0.52$. For each method, larger values of the stated coefficients give stronger damping.

gradient is represented by the spatial derivatives of η. In the method (15), the Coriolis terms and pressure gradient are evaluated at different times, so there is a first-order time truncation error in the geostrophic balance.

For the present test problem, the method of Hallberg [9] can be written in the form

$$\begin{aligned}
\hat{u}^{pred} &= \hat{u}^n + (f\Delta t)\hat{v}^n - i(kc\Delta t)\hat{\eta}^n \\
\hat{v}^{pred} &= \hat{v}^n - (f\Delta t)\hat{u}^n - i(\ell c\Delta t)\hat{\eta}^n \\
\hat{\eta}^{pred} &= \hat{\eta}^n - i(kc\Delta t)[\beta\hat{u}^{pred} + (1-\beta)\hat{u}^n] - i(\ell c\Delta t)[\beta\hat{v}^{pred} + (1-\beta)\hat{v}^n] \\
\hat{u}^{n+1} &= \hat{u}^n + (f\Delta t)[\psi\hat{v}^{pred} + (1-\psi)\hat{v}^n] - i(kc\Delta t)[\zeta\hat{\eta}^{pred} + (1-\zeta)\hat{\eta}^n] \\
\hat{v}^{n+1} &= \hat{v}^n - (f\Delta t)[\psi\hat{u}^{pred} + (1-\psi)\hat{u}^n] - i(\ell c\Delta t)[\zeta\hat{\eta}^{pred} + (1-\zeta)\hat{\eta}^n] \\
\hat{\eta}^{n+1} &= \hat{\eta}^n - i(kc\Delta t)[\mu\hat{u}^{n+1} + (1-\mu)\hat{u}^n] - i(\ell c\Delta t)[\mu\hat{v}^{n+1} + (1-\mu)\hat{v}^n].
\end{aligned} \tag{16}$$

Here, β, ψ, ζ, and μ are constants which represent weighted time averages. If all of these coefficients are equal to 0.5, then the method is unstable. The lower frame in Fig. 3 shows the solution obtained with the method (16) with $\beta = \mu = 0.5$ and $\psi = \zeta = 0.52$. This solution shows a greater phase error than the solution shown in the top frame in Fig. 2, and there is an amplitude error due to damping. Larger values of the parameters β, ψ, ζ, and μ are discussed in [9], and these give stronger damping. If $\psi \neq \zeta$, then the method (16) admits a first-order time truncation error in the geostrophic balance.

If $kc\Delta t = \ell c\Delta t = 0$, then the two methods produce solutions that are similar to those shown in Fig. 3. In this case, the analytical solution consists entirely of inertial oscillations with angular frequency f. However, the numerical methods can introduce damping, apparently due to the treatment of the Coriolis terms. Numerical damping is often used in geophysical calculations in order to suppress grid-scale noise, but the present methods can induce numerical damping of motions that are resolved well by the grid.

3. THREE-DIMENSIONAL LAYERED MODEL

Here we extend the algorithm of the preceding section to the case of a three-dimensional layered ocean model.

3.1. Governing Equations

In this discussion, the governing equations are expressed in terms of isopycnic coordinates. In addition, it is assumed that the fluid is hydrostatic, which is equivalent to assuming that the depth is small relative to the horizontal length scale.

Here we consider a vertically discrete system consisting of R layers, each of constant density. Number the layers 1 through R, with indices increasing downward, and let α_r denote the specific volume (reciprocal of density) in layer r. Let $\mathbf{u}_r(x, y, t) = (u_r(x, y, t), v_r(x, y, t))$ denote the horizontal velocity in layer r, $\Delta p_r(x, y, t)$ the vertical pressure difference between the bottom and top of layer r, and $M_r(x, y, t) = \alpha p + gz$ the Montgomery potential in layer r. Here, g is the acceleration due to gravity. The hydrostatic relation $\partial p/\partial z = -\alpha^{-1}g$ implies that the Montgomery potential is independent of depth in a layer of constant density and that $\Delta p_r/g$ is the mass per unit horizontal area in layer r. We then consider the system of equations

$$\frac{\partial \mathbf{u}_r}{\partial t} + (\mathbf{u}_r \cdot \nabla)\mathbf{u}_r + f\mathbf{k} \times \mathbf{u}_r = -\nabla M_r + \mathbf{D}(\mathbf{u}_r, \Delta p_r) \tag{17}$$

$$\frac{\partial \Delta p_r}{\partial t} + \nabla \cdot (\mathbf{u}_r \Delta p_r) = 0 \tag{18}$$

$$M_{r+1} - M_r = p_{r+1/2}(\alpha_{r+1} - \alpha_r) \tag{19}$$

(Bleck and Smith [4]). Here, $\nabla = (\partial/\partial x, \partial/\partial y)$, $p_{r+1/2}$ denotes the pressure at the interface between layers r and $r + 1$, $\mathbf{k}$ is the unit vector in the upward direction, and

$$\mathbf{D}(\mathbf{u}_r, \Delta p_r) = \frac{g\Delta\tau_r}{\Delta p_r} + \frac{1}{\Delta p_r}\nabla \cdot (A_H \Delta p_r \nabla \mathbf{u}_r) \tag{20}$$

contains the terms involving stresses and horizontal viscosity. The stresses contained in the vertical differences $\Delta\tau_r$ in Eq. (20) are due to wind stress at the top of the fluid domain, frictional stress at the bottom, and shear stress due to vertical variations in the horizontal velocity. The present formulation of the governing equations includes an assumption that mass is not exchanged between layers. Such exchanges are possible in the real ocean, since the vertical transport of heat and salt can cause surfaces of constant density to move, but these effects will not be considered here.

3.2. Barotropic–Baroclinic Splitting

The system (17)–(19) admits motions varying on a wide range of timescales, and this creates major problems related to computational efficiency. One approach to this situation is to split the fast and slow motions into separate subproblems and then solve the individual subproblems using different techniques that are appropriate for those cases. Here we use a splitting introduced by Bleck and Smith [4], with modifications developed by Higdon and de Szoeke [12] to improve its stability. In the following, the "barotropic" system is a two-dimensional system that represents the dynamics of fast external gravity waves, which are manifested in part by perturbations in the elevation of the free surface. The solutions of this system are similar to those of the shallow-water equations for a fluid of constant density. The "baroclinic" system is a three-dimensional system that represents the remaining slower motions, which include internal waves and advective motions.

Let $p'_b(x, y)$ be the pressure at the bottom of the fluid domain at a reference state, such as an equilibrium state or initial state, and let $p_b(x, y, t)$ be the bottom pressure at an arbitrary state. Define a dimensionless quantity $\eta(x, y, t)$ by $p_b = p'_b + p'_b\eta$, so that η is the relative perturbation in bottom pressure, with $|\eta| \ll 1$. The quantity $p'_b\eta/g$ is the perturbation in the total mass per unit horizontal area in a water column, and $p'_b\eta$ serves as the dependent mass variable in the barotropic system. A baroclinic mass variable $\Delta p'_r$ is then defined by $\Delta p_r(x, y, t) = (1 + \eta(x, y, t))\Delta p'_r(x, y, t)$ for $1 \le r \le R$. This relation is based on the idea that an external wave causes all fluid layers to thicken or thin by approximately the same proportion.

The barotropic velocity is the mass-weighted vertical average of $\mathbf{u}$ given by

$$\bar{\mathbf{u}}(x, y, t) = \sum_{r=1}^{R} \frac{\Delta p'_r}{p'_b}\mathbf{u}_r. \tag{21}$$

The quantity $p_b'\bar{\mathbf{u}}/g$ is the net horizontal mass flux over all layers. The baroclinic velocity is the residual $\mathbf{u}_r'(x, y, t) = \mathbf{u}_r(x, y, t) - \bar{\mathbf{u}}(x, y, t)$, which has a mass-weighted vertical average equal to zero.

The barotropic system is

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + f\mathbf{k} \times \bar{\mathbf{u}} = -\overline{\nabla M} + \mathbf{G} \tag{22}$$

$$\frac{\partial}{\partial t}(p_b'\eta) + \nabla \cdot (p_b'\bar{\mathbf{u}}) = 0 \tag{23}$$

and the baroclinic system is

$$\frac{\partial \mathbf{u}_r'}{\partial t} + (\mathbf{u}_r \cdot \nabla)\mathbf{u}_r + f\mathbf{k} \times \mathbf{u}_r' = -(\nabla M_r - \overline{\nabla M}) + \mathbf{D}(\mathbf{u}_r, \Delta p_r) - \mathbf{G} \tag{24}$$

$$\frac{\partial \Delta p_r'}{\partial t} + \nabla \cdot (\mathbf{u}_r \Delta p_r') = \frac{\Delta p_r'}{p_b'} \nabla \cdot (p_b'\bar{\mathbf{u}}). \tag{25}$$

(See Bleck and Smith [4] and Higdon and de Szoeke [12].) The barotropic momentum equation (22) is a vertical average of the momentum equation (17). The quantity $\overline{\nabla M}$ is the mass-weighted vertical average of ∇M; and the residual term $\mathbf{G}(x, y, t)$ includes the vertical averages of the nonlinear, stress, and viscosity terms in (17). An explicit representation of $\overline{\nabla M}$ is given by Higdon [11]. The baroclinic momentum equation (24) is obtained by subtracting (22) from (17). When (24) is solved, the zero-mean condition on $\mathbf{u}'$ is used to compute $\mathbf{G}$, which is then transferred to the barotropic momentum equation (22) to serve as a forcing term.

The three-dimensional baroclinic system can be solved explicitly with a long time step that is appropriate for resolving the slow motions. The two-dimensional barotropic system can be solved explicitly with shorter time steps or implicitly with the same time step as the baroclinic system. The implicit approach is discussed and used below.

3.3. Numerical Algorithm

Here we describe a two-level time-stepping scheme for the coupled barotropic–baroclinic system (22)–(25). Essentially, the algorithm developed in Section 2.2 for the shallow-water equations is applied to the baroclinic equations, and some additional steps are introduced in order to account for the barotropic equations. The following algorithm describes evolution of the system from time t_n to time t_{n+1}, where the time step $\Delta t = t_{n+1} - t_n$ is determined by the slow motions described by the baroclinic equations.

Step 1. Predict the baroclinic velocity $\mathbf{u}_r' = (u_r', v_r')$ at time t_{n+1} by applying a forward time difference to the baroclinic momentum equation (24). After all of the other terms in that equation are computed, the quantity $\mathbf{G}$ is computed by enforcing the zero-mean condition on $\mathbf{u}_r'$ at time t_{n+1}. This process is similar to a procedure described in Section 2.3 of Higdon and de Szoeke [12].

Step 2. Predict the barotropic variables $\bar{u}$, $\bar{v}$, and $p_b'\eta$ by solving the barotropic system (22), (23). The quantity $\overline{\nabla M}$ in (22) involves both the barotropic quantity $p_b'\eta$ and the baroclinic mass variables $\Delta p_r'$ (see Higdon [11]). During the present step, use the values of $\Delta p_r'$ from time t_n.

Step 3. Use the thickness equation (25) to compute $\Delta p_r'$ at time t_{n+1}. For an advective velocity $\mathbf{u}_r$, use the average of the old and predicted total velocities $\mathbf{u}_r' + \bar{\mathbf{u}}$ in order to obtain a velocity at time $t_{n+1/2} = t_n + \Delta t/2$.

Step 4. Correct the baroclinic velocity $\mathbf{u}_r' = (u_r', v_r')$. In the pressure, viscous, and stress terms, use averages of values from times t_n and t_{n+1}. The values from time t_{n+1} are based on the updated values of $\Delta p_r'$ and the predicted values of other quantities. Implement the Coriolis terms implicitly, using an iteration if necessary.

Step 5. Correct the barotropic variables $\bar{u}$, $\bar{v}$, and $p_b'\eta$. For the baroclinic quantities appearing in $\overline{\nabla M}$, use averages of values from times t_n and t_{n+1}.

In Step 3, the thickness equation must be solved with an advection algorithm which ensures that the layer thicknesses remain nonnegative. In addition, the baroclinic momentum equation (24) can be converted to a flux form in which the dependent variable is a baroclinic momentum density, and the resulting equation can also be solved with an advection algorithm. This process is discussed in Section 4, and it was used in the numerical computations reported in Section 5.

As pointed out by Smolarkiewicz and Margolin [18], a second-order advection algorithm requires that the advective velocity be evaluated at the intermediate time $t_{n+1/2}$, so that certain terms in the truncation error are compensated appropriately. These authors note that the advective velocity at time $t_{n+1/2}$ can be computed with a prediction step or with an extrapolation, and this strategy proved successful in the examples reported in [18]. In the setting of a barotropic–baroclinic splitting, an analogous procedure would be to predict the baroclinic and barotropic velocities directly at time $t_{n+1/2}$. However, the resulting barotropic–baroclinic algorithm turns out to be linearly unstable, according to a stability analysis similar to the one given in the next section. On the other hand, a linearly stable method is obtained if the predicted variables are obtained at time t_{n+1} and then time averages are used when needed. This result does not prove stability in the nonlinear case, but it does suggest that time averages be tried in the algorithm described above. The algorithm gave good results in the nonlinear calculations described in Section 5.

3.4. Stability Analysis

We now analyze the stability of the preceding algorithm as applied to a linearization of the barotropic–baroclinic system (22)–(25). For reasons of space, we do not include an analysis of the analogous (unstable) algorithm in which the predicted quantities are computed directly at time $t_{n+1/2}$.

For the sake of the analysis, assume that the bottom of the fluid domain is level and that the fluid consists of two layers having equilibrium thicknesses $\Delta\tilde{p}_1$ and $\Delta\tilde{p}_2$ and specific volumes α_1 and α_2, with index 1 referring to the upper layer. Also assume that the stress and viscosity terms are zero and that the flow is a small perturbation of a state of zero flow and level fluid interfaces.

As noted in Section 3.2, the baroclinic quantity $\mathbf{u}'$ has mass-weighted vertical average equal to zero. In the present linearized setting, this relation has the form $\mathbf{u}_1'(\Delta\tilde{p}_1/\tilde{p}_b) + \mathbf{u}_2'(\Delta\tilde{p}_2/\tilde{p}_b) = 0$, where $\tilde{p}_b = \Delta\tilde{p}_1 + \Delta\tilde{p}_2$, so if $\mathbf{u}_1'$ is known then $\mathbf{u}_2'$ is determined uniquely. In addition, the baroclinic layer thicknesses $\Delta p_1'$ and $\Delta p_2'$ satisfy $\Delta p_1' + \Delta p_2' = p_b' = \tilde{p}_b$, so if $\Delta p_1'$ is known then $\Delta p_2'$ is determined. In the following analysis, it therefore suffices to consider the baroclinic equations in the upper layer only.

The linearization of the system (22)–(25) was derived by Higdon and de Szoeke [12]. In order to state this system, let

$$c_0 = \left(\alpha_2 \tilde{p}_b + \frac{(\Delta\tilde{p}_1)^2 \Delta\alpha)}{\tilde{p}_b}\right)^{1/2} \tag{26}$$

$$c_1 = \left(\frac{\Delta\tilde{p}_1 \Delta\tilde{p}_2 \Delta\alpha}{\tilde{p}_b}\right)^{1/2}, \tag{27}$$

where $\Delta\alpha = \alpha_1 - \alpha_2$. The quantities c_0 and c_1 are the speeds of external and internal waves, respectively, in the present model. Replace the barotropic velocities $\bar{u}(x, y, t)$ and $\bar{v}(x, y, t)$ with the dimensionless variables $\bar{u}/c_0$ and $\bar{v}/c_0$, and let $u'(x, y, t) = u'_1/c_1$ and $v'(x, y, t) = v'_1/c_1$ denote dimensionless components of baroclinic velocity. Also let $\delta(x, y, t)$ denote the perturbation in $\Delta p'_1$, divided by the equilibrium thickness $\Delta\tilde{p}_1$. The linearized barotropic equations can then be written in the form

$$\frac{\partial\bar{u}}{\partial t} - f\bar{v} = -c_0\frac{\partial\eta}{\partial x} - \gamma c_0\frac{\partial\delta}{\partial x} \tag{28}$$

$$\frac{\partial\bar{v}}{\partial t} + f\bar{u} = -c_0\frac{\partial\eta}{\partial y} - \gamma c_0\frac{\partial\delta}{\partial y} \tag{29}$$

$$\frac{\partial\eta}{\partial t} + c_0\left(\frac{\partial\bar{u}}{\partial x} + \frac{\partial\bar{v}}{\partial y}\right) = 0, \tag{30}$$

and the baroclinic equations are

$$\frac{\partial u'}{\partial t} - fv' = -c_1\frac{\partial\delta}{\partial x} - c_1\frac{\partial\eta}{\partial x} \tag{31}$$

$$\frac{\partial v'}{\partial t} + fu' = -c_1\frac{\partial\delta}{\partial y} - c_1\frac{\partial\eta}{\partial y} \tag{32}$$

$$\frac{\partial\delta}{\partial t} + c_1\left(\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y}\right) = 0. \tag{33}$$

Here $\gamma = (\frac{c_1}{c_0})^2(\frac{\Delta\tilde{p}_1}{\Delta\tilde{p}_2}) = \mathcal{O}(\frac{\Delta\alpha}{\alpha_2}) \ll 1$.

Assume that the system (28)–(33) is discretized in (x, y) with centered second-order finite differences on a C-grid. Also assume that the system is defined for all (x, y) and that f is constant, and apply a Fourier transform with respect to x and y. The actions of spatial differences will be represented with the quantities

$$\frac{iK}{\Delta x} = \frac{e^{ik\Delta x/2} - e^{-ik\Delta x/2}}{\Delta x}$$

$$\frac{iL}{\Delta y} = \frac{e^{i\ell\Delta y/2} - e^{-i\ell\Delta y/2}}{\Delta y};$$

thus $K = 2\sin(k\Delta x/2)$ and $L = 2\sin(\ell\Delta y/2)$.

In the following analysis, we first assume that the barotropic equations are solved exactly with respect to t for $t_n < t < t_{n+1}$ during both the prediction and correction steps. This is done in order to isolate the effects of the barotropic–baroclinic splitting and the overall time-stepping scheme. In Section 3.5, we include the effects of one particular implicit method for solving the barotropic equations.

After a Fourier transform, the barotropic system (28)–(30) becomes

$$\frac{\partial}{\partial t}\begin{pmatrix}\hat{u}\\ \hat{v}\\ \hat{\eta}\end{pmatrix} = \frac{1}{\Delta t}\begin{pmatrix} 0 & \phi & -\frac{1}{r}(iK\nu_x)\\ -\phi & 0 & -\frac{1}{r}(iL\nu_y)\\ -\frac{1}{r}(iK\nu_x) & -\frac{1}{r}(iL\nu_y) & 0\end{pmatrix}\begin{pmatrix}\hat{u}\\ \hat{v}\\ \hat{\eta}\end{pmatrix} + \frac{1}{\Delta t}\begin{pmatrix}-\frac{\gamma}{r}(iK\nu_x)\\ -\frac{\gamma}{r}(iL\nu_y)\\ 0\end{pmatrix}\hat{\delta}. \tag{34}$$

Here, $\Delta t = t_{n+1} - t_n$ is the time step for solving the baroclinic equations; $r = c_1/c_0 \ll 1$; $\nu_x = c_1\Delta t/\Delta x$ and $\nu_y = c_1\Delta t/\Delta y$ denote the Courant numbers with respect to x and y, respectively; and

$$\phi = (f\Delta t)\cos\frac{k\Delta x}{2}\cos\frac{\ell\Delta y}{2}.$$

The definition of ϕ contains trigonometric factors because four-point averages are used to obtain values of v at u-points and to obtain values of u at v-points when evaluating the Coriolis terms. In accordance with Steps 2 and 5 of the algorithm of Section 3.3, $\hat{\delta} = \hat{\delta}^n$ during the prediction step and $\hat{\delta} = (\hat{\delta}^{n+1} + \hat{\delta}^n)/2$ during the correction step.

The system (34) has the form

$$\frac{\partial s}{\partial t} = \frac{1}{\Delta t}As + \frac{1}{\Delta t}b, \tag{35}$$

where $s = (\hat{u}, \hat{v}, \hat{\eta})^T$, A is a 3×3 matrix, and b is a column vector. The matrix iA is self-adjoint and thus has real eigenvalues and orthogonal eigenvectors. A calculation shows that exactly one of these eigenvalues is zero, so the matrix A has two nonzero imaginary eigenvalues and one eigenvalue equal to zero. Let Q be a unitary matrix such that $Q^{-1}AQ$ is diagonal, and use this matrix to diagonalize the system (35). A calculation yields

$$s(t_{n+1}) = e^A s(t_n) + Z\hat{\delta}, \tag{36}$$

where A is obtained by comparing (34) and (35),

$$Z = Q\begin{pmatrix}\frac{e^{\lambda_1}-1}{\lambda_1} & 0 & 0\\ 0 & \frac{e^{\lambda_2}-1}{\lambda_2} & 0\\ 0 & 0 & 1\end{pmatrix} Q^{-1}\begin{pmatrix}-\frac{\gamma}{r}(iK\nu_x)\\ -\frac{\gamma}{r}(iL\nu_y)\\ 0\end{pmatrix}, \tag{37}$$

and λ_1 and λ_2 are the nonzero eigenvalues of A.

We now formulate the prediction step in matrix–vector form. The predicted values of baroclinic velocity are given by

$$\begin{aligned}\hat{u}'^{pred} &= \hat{u}'^n + \phi\hat{v}'^n - iK\nu_x\hat{\delta}^n - iK\nu_x\hat{\eta}^n\\ \hat{v}'^{pred} &= \hat{v}'^n - \phi\hat{u}'^n - iL\nu_y\hat{\delta}^n - iL\nu_y\hat{\eta}^n.\end{aligned}$$

The relative perturbation δ in the layer thickness is not predicted and corrected but instead is updated in one step. However, for the sake of the matrix formulation, it is convenient to define $\hat{\delta}^{pred} = \hat{\delta}^n$. The predicted values of the barotropic variables are given by Eqs. (36), (37), with $\hat{\delta} = \hat{\delta}^n$. The prediction step is then

$$\begin{pmatrix} \hat{u}'^{pred} \\ \hat{v}'^{pred} \\ \hat{\delta}^{pred} \\ \hat{\bar{u}}^{pred} \\ \hat{\bar{v}}^{pred} \\ \hat{\eta}^{pred} \end{pmatrix} = \begin{pmatrix} 1 & \phi & -iK\nu_x & 0 & 0 & -iK\nu_x \\ -\phi & 1 & -iL\nu_y & 0 & 0 & -iL\nu_y \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & Z_1 & B_{11} & B_{12} & B_{13} \\ 0 & 0 & Z_2 & B_{21} & B_{22} & B_{23} \\ 0 & 0 & Z_3 & B_{31} & B_{32} & B_{33} \end{pmatrix} \begin{pmatrix} \hat{u}'^n \\ \hat{v}'^n \\ \hat{\delta}^n \\ \hat{\bar{u}}^n \\ \hat{\bar{v}}^n \\ \hat{\eta}^n \end{pmatrix}, \tag{38}$$

where $(Z_1, Z_2, Z_3)^T = Z$, and B_{ij} is entry (i, j) of the matrix $B = e^A$ in (36). For later usage, Eq. (38) is written in the form

$$w^{pred} = Ew^n. \tag{39}$$

Next consider the final calculation of the solution at time t_{n+1}. The updated value of the baroclinic thickness is

$$\hat{\delta}^{n+1} = \hat{\delta}^n - iK\nu_x \cdot \frac{1}{2}(\hat{u}'^{pred} + \hat{u}'^n) - iL\nu_y \cdot \frac{1}{2}(\hat{v}'^{pred} + \hat{v}'^n),$$

and the corrected components of baroclinic velocity are

$$\begin{aligned} \hat{u}'^{n+1} &= \hat{u}'^n + \frac{\phi}{2}(\hat{v}'^{n+1} + \hat{v}'^n) - iK\nu_x \cdot \frac{1}{2}(\hat{\delta}^{n+1} + \hat{\delta}^n) - iK\nu_x \cdot \frac{1}{2}(\hat{\eta}^{pred} + \hat{\eta}^n) \\ \hat{v}'^{n+1} &= \hat{v}'^n - \frac{\phi}{2}(\hat{u}'^{n+1} + \hat{u}'^n) - iL\nu_y \cdot \frac{1}{2}(\hat{\delta}^{n+1} + \hat{\delta}^n) - iL\nu_y \cdot \frac{1}{2}(\hat{\eta}^{pred} + \hat{\eta}^n). \end{aligned}$$

The corrected values of the barotropic variables are given by (36), (37), with $\hat{\delta} = (\hat{\delta}^{n+1} + \hat{\delta}^n)/2$. The solution at time t_{n+1} is then

$$F_1 w^{n+1} = F_0 w^n + F_{pred} w^{pred}, \tag{40}$$

where

$$F_1 = \begin{pmatrix} 1 & -\phi/2 & iK\nu_x/2 & 0 & 0 & 0 \\ \phi/2 & 1 & iL\nu_y/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -Z_1/2 & 1 & 0 & 0 \\ 0 & 0 & -Z_2/2 & 0 & 1 & 0 \\ 0 & 0 & -Z_3/2 & 0 & 0 & 1 \end{pmatrix},$$

$$F_0 = \begin{pmatrix} 1 & \phi/2 & -iK\nu_x/2 & 0 & 0 & -iK\nu_x/2 \\ -\phi/2 & 1 & -iL\nu_y/2 & 0 & 0 & -iL\nu_y/2 \\ -iK\nu_x/2 & -iL\nu_y/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & Z_1/2 & B_{11} & B_{12} & B_{13} \\ 0 & 0 & Z_2/2 & B_{21} & B_{22} & B_{23} \\ 0 & 0 & Z_3/2 & B_{31} & B_{32} & B_{33} \end{pmatrix},$$

and

$$F_{pred} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -iK\nu_x/2 \\ 0 & 0 & 0 & 0 & 0 & -iL\nu_y/2 \\ -iK\nu_x/2 & -iL\nu_y/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $w^j = (\hat{u}'^j, \hat{v}'^j, \hat{\delta}^j, \hat{\bar{u}}^j, \hat{\bar{v}}^j, \hat{\eta}^j)^T$.

The steps (39) and (40) can be combined to yield

$$w^{n+1} = F_1^{-1}(F_0 + F_{pred}E)w^n \equiv Gw^n. \tag{41}$$

The amplification matrix G and the vectors w^{n+1} and w^n are functions of the wavenumbers k and ℓ. If the powers G^n are bounded for $0 < n < \infty$, uniformly in (k, ℓ), then the algorithm is stable.

Because of the complexity of the algorithm, we do not try to develop an analytical description of the eigenvalues and eigenvectors of G, but instead we compute some numerical examples that suggest stability. Suppose, for example, that the equilibrium layer thicknesses satisfy $\Delta\tilde{p}_1/\tilde{p}_b = 0.25$ and $\Delta\tilde{p}_2/\tilde{p}_b = 0.75$, the Courant numbers are $\nu_x = \nu_y = 0.5$, the dimensionless Coriolis parameter is $f\Delta t = 0.5$, and the relative difference in α is $\Delta\alpha/\alpha_2 = 0.004$. The top frame of Fig. 4 shows a plot of the absolute values of the eigenvalues λ of G in this case, for $0 \le k\Delta x \le \pi$ and $\ell = 0$. The middle frame shows a plot for $0 \le k\Delta x \le \pi$ along the line $\ell = k \tan 30°$ in the (k, ℓ) plane, and the bottom frame shows a plot along the line $\ell = k$. For each plot, the interval $0 \le k\Delta x \le \pi$ was divided into 1000 subintervals, and the eigenvalues of G were computed at each sample point. In each case, the maximum of the absolute values is 1, and the minimum is very close to 1. Similar behavior was found along numerous other lines of the form $\ell = k \tan\theta$ for $0 \le \theta \le 45°$. (Because of symmetries, it suffices to search in this range of angles.) The empirical condition $|\lambda| \le 1$ suggests stability, and the additional condition $|\lambda| \approx 1$ suggests little dissipation.

Eigenvalues were also calculated with numerous other values of $\Delta\alpha/\alpha_2$, $f\Delta t$, ν_x, and ν_y, with the constraint $\nu_x = \nu_y$. Overall, these experiments suggest that the method is stable and nearly nondissipative for arbitrary values of $f\Delta t$, provided that the Courant number $\nu_x = \nu_y$ is less than a bound which is approximately equal to $1/\sqrt{2}$.

3.5. Implicit Solution of the Barotropic Equations

In the barotropic–baroclinic system (22)–(25), the fast barotropic equations could be solved implicitly over the time interval $t_n < t < t_{n+1}$, or they could be solved explicitly

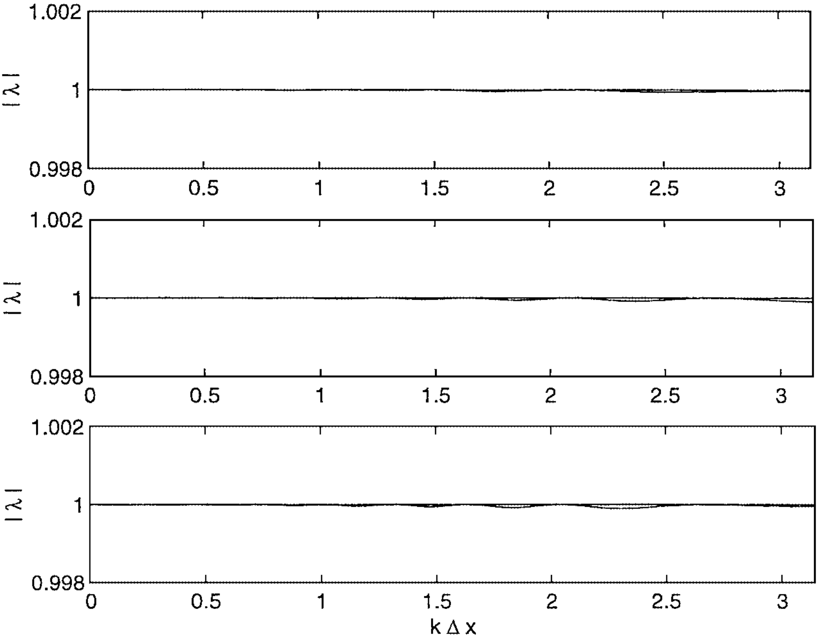

FIG. 4. Plots of absolute values of the eigenvalues of the amplification matrix G along lines of the form $\ell = k \tan\theta$ in the (k, ℓ) plane. The top, middle, and bottom frames correspond to $\theta = 0$, $\theta = 30°$, and $\theta = 45°$, respectively. In each case, the horizontal coordinate is $k\Delta x$ for $0 \le k\Delta x \le \pi$. The model parameters are $\nu_x = \nu_y = f\Delta t = 0.5$, $\Delta\alpha/\alpha_2 = 0.004$, $\Delta\tilde{p}_1/\tilde{p}_b = 0.25$, and $\Delta\tilde{p}_2/\tilde{p}_b = 0.75$. In this algorithm, the barotropic equations are solved exactly with respect to t for $t_n < t < t_{n+1}$.

using many short subintervals of that interval. The efficiency of this process is a major consideration with the time-stepping scheme discussed in the present paper, as the barotropic equations are solved twice during each baroclinic step. In the present section, we outline an implicit method which is much more efficient than explicit methods, and we discuss the effect of this method on the stability of the coupled barotropic–baroclinic system. The main conclusion is that the implicit scheme maintains stability and introduces a small amount of dissipation at high wavenumbers.

The method used here is an alternating direction implicit (ADI) method that was developed by Bates [2] for the (single-layer) shallow-water equations. This method, with a note regarding adaptation to a barotropic–baroclinic splitting, is as follows.

Step 1. Integrate from time t_n to time $t_{n+1/2} = t_n + \Delta t/2$.

(a) Compute $\bar{v}^{n+1/2}$ explicitly.

(b) Discretize the momentum equation (22) to express $\bar{u}^{n+1/2}$ in terms of $p_b'\eta^{n+1/2}$ and $\bar{v}^{n+1/2}$, and use the mass equation (23) to express $p_b'\eta^{n+1/2}$ in terms of $\bar{u}^{n+1/2}$ and $\bar{v}^n$. Implement by solving a tridiagonal system for $p_b'\eta^{n+1/2}$. This step is implicit in the x-direction.

Step 2. Integrate from time $t_{n+1/2}$ to time t_{n+1}.

(a) Compute $\bar{u}^{n+1}$ explicitly.

(b) Compute $\bar{v}^{n+1}$ and $p_b'\eta^{n+1}$. This step is implicit in the y-direction.

Note. The barotropic momentum equation includes the term $\overline{\nabla M}$, which involves $p_b'\eta$ and quantities involving the baroclinic layer thicknesses $\Delta p_r'$ (Higdon [11]). For reasons of stability, the baroclinic quantities are not stepped along with $p_b'\eta$ but instead are held constant during the integration from time t_n to $t_{n+1/2}$ to t_{n+1}. During the prediction step, the baroclinic quantities are taken from time t_n, and during the correction step, the baroclinic quantities are averages of values from times t_n and t_{n+1}.

When Steps 1 and 2 are applied to the linearized shallow-water equations, the algorithm is stable, nondissipative, and second-order accurate (Bates [2]).

In the tridiagonal system that is used to compute $p_b'\eta$, the off-diagonal entries include a factor which is the value of p_b' at velocity points. Under a method used in Section 4.5 for calculating layer thicknesses at velocity points, this factor is zero at locations where a grid row is interrupted by a land mass. In such situations, the linear system for that grid row is decoupled into separate linear systems for the disjoint water regions, so the presence of land masses does not disrupt the process of solving the equations implicitly. However, these issues do not arise in the computations described in Section 5, so further details will not be included here.

Since the linear system is tridiagonal, the computational work in implementing each of Steps 1 and 2 is comparable to the work involved in solving the barotropic equations explicitly for one time step. However, the small time step for an explicit solution, relative to a baroclinic time step, is determined by the ratio of the speeds of external waves to internal waves. These speeds can differ by up to two orders of magnitude, so the total operation count for an implicit solution is much smaller than the operation count for an explicit solution over many substeps.

We now modify the stability analysis of Section 3.4 in order to account for the effects of solving the barotropic equations with the implicit scheme outlined above.

When Step 1(a) of the implicit scheme is applied to the barotropic equations (34) in the linearized system, the result is

$$\hat{v}^{n+1/2} = \hat{v}^n - \frac{1}{2}\phi\hat{u}^n - \frac{1}{2r}iL\nu_y\hat{\eta}^n - \frac{\gamma}{2r}iL\nu_y\hat{\delta},$$

and Step 1(b) yields

$$\begin{aligned}\hat{u}^{n+1/2} &= \hat{u}^n + \frac{1}{2}\phi\hat{v}^{n+1/2} - \frac{1}{2r}iK\nu_x\hat{\eta}^{n+1/2} - \frac{\gamma}{2r}iK\nu_x\hat{\delta}\\ \hat{\eta}^{n+1/2} &= \hat{\eta}^n - \frac{1}{2r}iK\nu_x\hat{u}^{n+1/2} - \frac{1}{2r}iL\nu_y\hat{v}^n.\end{aligned}$$

Step 1 can then be written in the matrix–vector form

$$E_h s^{n+1/2} = E_0 s^n + b\hat{\delta}, \tag{42}$$

where $s^j = (\hat{u}^j,\ \hat{v}^j,\ \hat{\eta}^j)T$ and

$$E_h = \begin{pmatrix} 1 & -\frac{\phi}{2} & \frac{1}{2r}iK\nu_x \\ 0 & 1 & 0 \\ \frac{1}{2r}iK\nu_x & 0 & 1 \end{pmatrix}$$

$$E_0 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\phi}{2} & 1 & -\frac{1}{2r} i L \nu_y \\ 0 & -\frac{1}{2r} i L \nu_y & 1 \end{pmatrix}$$

$$b = \begin{pmatrix} -\frac{\gamma}{2r} i K \nu_x \\ -\frac{\gamma}{2r} i L \nu_y \\ 0 \end{pmatrix}.$$

Similarly, Step 2 can be written in the form

$$F_1 s^{n+1} = F_h s^{n+1/2} + b\hat{\delta}, \tag{43}$$

where

$$F_1 = \begin{pmatrix} 1 & 0 & 0 \\ \frac{\phi}{2} & 1 & \frac{1}{2r} i L \nu_y \\ 0 & \frac{1}{2r} i L \nu_y & 1 \end{pmatrix}$$

$$F_h = \begin{pmatrix} 1 & \frac{\phi}{2} & -\frac{1}{2r} i K \nu_x \\ 0 & 1 & 0 \\ -\frac{1}{2r} i K \nu_x & 0 & 1 \end{pmatrix}.$$

Equations (42) and (43) imply

$$s^{n+1} = \left(F_1^{-1} F_h E_h^{-1} E_0\right) s^n + \left(F_1^{-1} F_h E_h^{-1} + F_1^{-1}\right) b\hat{\delta} \equiv B s^n + Z\hat{\delta}. \tag{44}$$

The quantity $\hat{\delta}$ is given by $\hat{\delta} = \hat{\delta}^n$ during the prediction step and $\hat{\delta} = (\hat{\delta}^{n+1} + \hat{\delta}^n)/2$ during the correction step.

Equation (44) has the same form as the solution (36) obtained by integrating the barotropic equations exactly with respect to t for $t_n < t < t_{n+1}$, except that the matrix $B = F_1^{-1} F_h E_h^{-1} E_0$ replaces the matrix e^A, and the vector $Z = (F_1^{-1} F_h E_h^{-1} + F_1^{-1})b$ replaces the vector Z defined in Eq. (37). With these substitutions, the matrix–vector form of the barotropic–baroclinic system, with the barotropic equations solved with the ADI method described above, has the same form as the relation (41) that is obtained when the barotropic equations are solved exactly.

As in Section 3.4, we compute numerical examples of eigenvalues of the amplification matrix instead of attempting to characterize the eigenvalues analytically. Figure 5 shows plots of eigenvalues over the same lines in the (k, ℓ) plane as in Fig. 4. The same model parameters and vertical scaling are used in the two figures. In both cases, the maximum of the absolute values of the eigenvalues is 1. A comparison of the two figures indicates that the ADI method introduces a small amount of dissipation relative to the earlier case. The amount of dissipation is at most a fraction of a percent over each time step. Experiments with other values of θ and model parameters yield results similar to those shown in Fig. 5.

An alternative to using $\hat{\delta} = (\hat{\delta}^{n+1} + \hat{\delta}^n)/2$ throughout the correction step is to use the values $\hat{\delta} = \hat{\delta}^n$, $\hat{\delta} = (\hat{\delta}^{n+1} + \hat{\delta}^n)/2$, and $\hat{\delta} = \hat{\delta}^{n+1}$ when $\hat{\eta}^n$, $\hat{\eta}^{n+1/2}$, and $\hat{\eta}^{n+1}$, respectively, are used. However, an analysis similar to the above indicates that the barotropic–baroclinic system is unstable in that case.

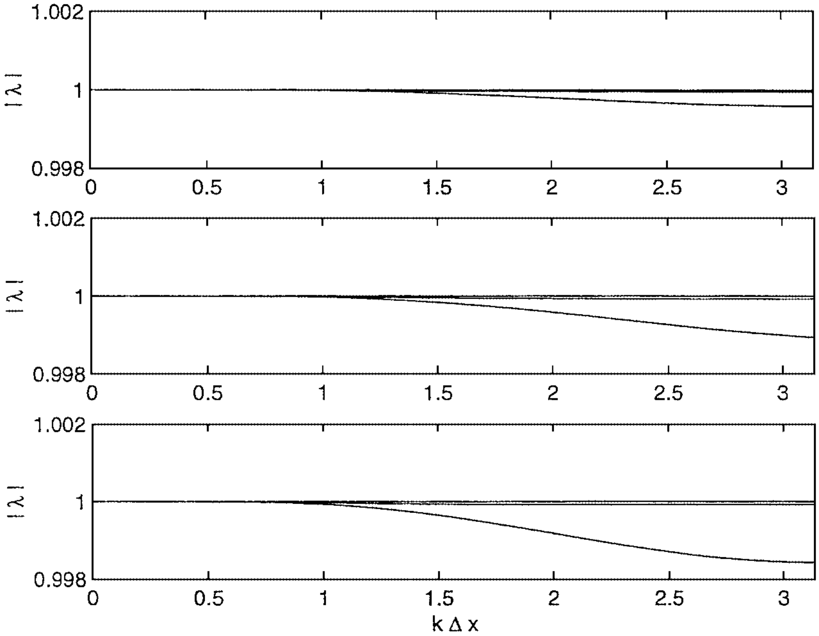

FIG. 5. Plots of absolute values of the eigenvalues of the amplification matrix, for the case where the barotropic equations are solved using the alternating direction implicit method. Plots are shown along lines of the form $\ell = k \tan\theta$ in the (k, ℓ) plane. The top, middle, and bottom frames correspond to $\theta = 0$, $\theta = 30°$, and $\theta = 45°$, respectively. The model parameters and vertical scaling are the same as in Fig. 4.

4. IMPLEMENTATION ISSUES

Section 5 contains the results of some numerical computations involving the time-stepping method developed in this paper. The present section gives some details of the implementation of this scheme for those computations. This discussion includes advection schemes for solving the thickness and momentum equations, the representation of stresses at interfaces between fluid layers, and some issues related to thin layers and bottom topography.

4.1. Thickness Equation

The baroclinic layer thickness equation (25) governs the time evolution of the baroclinic mass variable $\Delta p_r'$ and is given by

$$\frac{\partial \Delta p_r'}{\partial t} + \nabla \cdot (\mathbf{u}_r \Delta p_r') = \frac{\Delta p_r'}{p_b'} \nabla \cdot (p_b' \bar{\mathbf{u}}). \tag{45}$$

This equation is solved with the multidimensional positive definite advection transport algorithm (MPDATA) of Smolarkiewicz and collaborators (e.g., [18]), which consists of an initial upwind step followed by some antidiffusive corrections to improve accuracy. The corrections consist of upwind steps using pseudovelocities calculated from the results of the preceding iteration.

This scheme is designed to ensure that nonnegative fields remain nonnegative, which is an essential feature for the present application. In a layered ocean model, fluid interfaces can intersect the bottom or top of the fluid domain. One way to handle this situation is to assume that all fluid layers are defined in all mass cells and allow transitions between positive and negligible layer thickness. For the sake of computational stability, negative thicknesses must then be disallowed by the algorithm that is used for the thickness equation.

The forcing term on the right side of (45) consists of a depth-independent quantity multiplied by $\Delta p'_r$. This term is taken to be an average of values from times t_n and t_{n+1}, and at each time level the barotropic mass divergence $\nabla \cdot (p'_b \bar{\mathbf{u}})$ is calculated with centered spatial differences. The forcing term is then implemented as follows. First add half of the forcing term at time t_n (times Δt) to $(\Delta p')^n$, and apply MPDATA to the result. To the result of the MPDATA operation, add half the forcing term at time t_{n+1} (times Δt). The latter is calculated from the predicted values of barotropic velocity $\bar{\mathbf{u}}$. The factor $\Delta p'$ in the forcing term at time t_{n+1} means that $(\Delta p')^{n+1}$ appears linearly in two different terms in the discrete equation. The quantity $(\Delta p')^{n+1}$ is then computed readily. The preceding steps follow a procedure suggested by Smolarkiewicz and Margolin [18] for implementing forcing terms with MPDATA, and it is essentially an application of the splitting of Strang [19] to the present situation.

4.2. Momentum Equation

MPDATA is nearly nonoscillatory, so there are advantages to using this method to solve the baroclinic momentum equation as well. In this case, the dependent variables can be of varying sign. This requires a slight modification of the process for calculating pseudo-velocities for the antidiffusive correction steps, as described in [18].

In order to apply MPDATA in the present situation, the momentum equation must be converted to a flux form. In the case of the x-component, multiply the x-component of the momentum equation (24) by $\Delta p'_r$ and multiply the thickness equation (25) by u'_r. Some manipulation then yields

$$\begin{aligned}
&\frac{\partial}{\partial t}(u'_r \Delta p'_r) + \frac{\partial}{\partial x}[u_r(u'_r \Delta p'_r)] + \frac{\partial}{\partial y}[v_r(u'_r \Delta p'_r)] \\
&\quad = f v'_r \Delta p'_r - \Delta p'_r \left(\frac{\partial M_r}{\partial x} - (\overline{\nabla M})_x \right) + g(\Delta \tau_r)_x + \nabla \cdot (A_H \Delta p'_r \nabla u_r) - G_x \Delta p'_r \\
&\qquad - u_r \Delta p'_r \frac{\partial \bar{u}}{\partial x} - v_r \Delta p'_r \frac{\partial \bar{u}}{\partial y} + \frac{u'_r \Delta p'_r}{p'_b} \nabla \cdot (p'_b \bar{\mathbf{u}}). \qquad (46)
\end{aligned}$$

The left side of (46) represents advection of the quantity $u'_r \Delta p'_r$, with an advective velocity given by the total velocity $\mathbf{u}_r = (u_r, v_r)$. The remaining terms in that equation will be regarded as forcing terms. As noted above, the quantity $\Delta p_r / g$ is the mass per unit horizontal area in layer r. The quantity $(u'_r \Delta p'_r / g, v'_r \Delta p'_r / g)$ can then be regarded as a baroclinic, two-dimensional momentum density in layer r and also as a baroclinic rate of horizontal mass transport. The advection of the y-component $v'_r \Delta p'_r$ is described by an equation that is analogous to (46).

The computations reported in Section 5 employ a C-grid in space. This grid is used, for example, in the Miami Isopycnic Coordinate Ocean Model [3, 4]. Suppose that the mass points are indicated by indices having integer values, and consider the mass cell centered at (x_i, y_j). Values of u_r and $u'_r \Delta p'_r$ are then defined at the points $(x_{i\pm1/2}, y_j)$, and values of

v_r and $v_r' \Delta p_r'$ are defined at the points $(x_i, y_{j\pm1/2})$. The points $(x_{i\pm1/2}, y_j)$ and $(x_i, y_{j\pm1/2})$ are taken to be centers of x-momentum cells and y-momentum cells, respectively. Spatial averages of velocities are used to obtain advective velocities at the edges of the momentum cells.

The time-stepping method described in Section 3.3 requires a prediction and then a correction of the baroclinic velocity. When the prediction step is applied to the momentum formulation (46), the advective terms are discretized with the first-order upwind method, since the prediction step is necessarily first-order because of the forward difference in time. The layer thickness is also predicted with the upwind method, so that the velocities u_r' and v_r' can be extracted from the predicted momentum densities $u_r' \Delta p_r'$ and $v_r' \Delta p_r'$. The extraction of velocity from momentum is needed in order to obtain advective velocities and calculate forcing terms. The extraction process is discussed in Section 4.3.

During the correction step, the effects of shear stresses are incorporated in a manner described in Section 4.4. The remainder of the forcing from times t_n and t_{n+1} is implemented with a Strang splitting, in a manner similar to what is done for the layer thickness equation. Spatial discretization of the forcing terms is achieved through centered differencing and averaging, and antidiffusive corrections are included in the application of MPDATA to the advective terms. The antidiffusive corrections in the thickness and momentum equations constitute a significant portion of the overall computational time in the numerical experiments described in Section 5.

The Coriolis terms are represented with time averages of momentum densities from times t_n and t_{n+1}, and thus the representation is implicit. In particular, $(u_r' \Delta p_r')^{n+1}$ is represented in terms of a four-point average of $\frac{f\Delta t}{2}(v_r' \Delta p_r')^{n+1}$, due to the structure of the C-grid. Similarly, $(v_r' \Delta p_r')^{n+1}$ is represented in terms of a four-point average of $\frac{f\Delta t}{2}(u_r' \Delta p_r')^{n+1}$. The implicit equations can be solved with a fixed-point iteration. After each step of the iteration, the zero-sum condition on the momentum is enforced via the residual terms $G_x \Delta p_r'$ and $G_y \Delta p_r'$. The convergence of this iteration can be described as follows.

Let $(u_r' \Delta p_r')^{[m]}$ and $(v_r' \Delta p_r')^{[m]}$ denote the results of the mth iterations of the momentum densities. We can define $(u_r' \Delta p_r')^{[m+1]}$ in terms of $(v_r' \Delta p_r')^{[m]}$ and then $(v_r' \Delta p_r')^{[m+1]}$ in terms of $(u_r' \Delta p_r')^{[m+1]}$. Thus,

$$\max_{(x,y)} \left| (u_r' \Delta p_r')^{[m+1]} - (u_r' \Delta p_r')^{[m]} \right| \le \left| \frac{f\Delta t}{2} \right| \max_{(x,y)} \left| (v_r' \Delta p_r')^{[m]} - (v_r' \Delta p_r')^{[m-1]} \right|,$$

and similarly,

$$\max_{(x,y)} \left| (v_r' \Delta p_r')^{[m]} - (v_r' \Delta p_r')^{[m-1]} \right| \le \left| \frac{f\Delta t}{2} \right| \max_{(x,y)} \left| (u_r' \Delta p_r')^{[m]} - (u_r' \Delta p_r')^{[m-1]} \right|.$$

A difference of consecutive iterates is thus contracted by the factor $(f\Delta t/2)^2$ at each iteration. In the computations described in Section 5, $(f\Delta t/2)^2 \doteq 0.0152$. Four iterations are used, and the contraction after four iterations is $(f\Delta t/2)^8 \doteq 5.4 \times 10^{-8}$. Relative execution times for various aspects of the code are discussed at the end of Section 5.

4.3. Extracting Velocity from Momentum

Once a momentum density $u_r' \Delta p_r'$ has been computed, the velocity u_r' can be obtained by dividing by the values of $\Delta p_r'$ at u-points. The layer thickness equation yields values of $\Delta p_r'$ at mass points, so an interpolation is required in order to obtain values at velocity points. The interpolation method described in Section 4.5 consists of a linear interpolation between the adjoining mass cells, with modifications near the bottom of the fluid domain to account for the effects of variablebottom topography.

However, the division of momentum by thickness is problematic in situations where the thickness approaches zero. In principle, the ratio should be bounded, but in practice the ratio could become unbounded because of numerical errors in computing the quantities $u'_r \Delta p'_r$ and $\Delta p'_r$. The resulting large velocities can then cause a violation of the Courant–Friedrichs–Lewy condition.

In order to avoid this situation, we limit the velocities as follows. For a given mass cell in a given layer, compute the total outward mass flow that would be obtained over one time step if the upwind method were used. With the upwind method, the mass flux at a cell edge consists of the normal component of total velocity ($u'_r + \bar{u}$ or $v'_r + \bar{v}$) times the layer thickness in the cell that is being drained. For a given mass cell, the calculation of the outward mass flow involves all cell edges having outward normal velocity and does not involve any cell edges having inward normal velocity. If the outward mass flow exceeds the existing mass in the cell, and if the thickness in the cell is below some small threshold, then all of the outgoing velocities are limited proportionately so that the outward mass flow does not exceed the mass currently available in the cell. When the velocities are limited, the baroclinic velocities (u'_r or v'_r) are modified, not the depth-independent barotropic velocities ($\bar{u}$ or $\bar{v}$).

This procedure is applied only when the thickness in a mass cell is below some small threshold, which was arbitrarily chosen to be 1 cm for the computations described in Section 5. If the procedure were applied everywhere, then it could conceal algorithmic or coding problems that are unrelated to the problem of dividing by small thicknesses. However, in the present formulation, the limiting procedure is applied only in situations where a layer has little physical significance.

The above procedure does not cause a direct modification of the prognostic variables $u'_r \Delta p'_r$ and $v'_r \Delta p'_r$. Instead, the diagnostic quantities u'_r and v'_r are modified, and these are used to determine advective velocities. The above process thus amounts to a kind of flux-limiting procedure.

A different method for obtaining velocity from momentum and thickness was developed by Schär and Smolarkiewicz [17]. In their formulation, the upwind method is used to compute momentum and thickness, and the antidiffusive corrections to these quantities are then limited so that the ratio of momentum and thickness is bounded. In [17], the momenta and thickness are defined at the same grid points, and this methodology did not seem to extend readily to the present case of a staggered C-grid. The above method was then developed as an alternative.

4.4. Stresses

The momentum equation is forced by wind stress at the upper surface and by frictional stress along the bottom. In the present series of computations, the magnitude of the wind stress decays linearly to zero over a prescribed vertical distance which was taken to be 1 m. If the top layer has a thickness greater than 1 m, then the wind stress acts as a body force on that layer. On the other hand, if the thickness of the top layer is less than 1 m, then some of the wind stress is transmitted directly to lower layers. This procedure prevents large forcing from being concentrated in arbitrarily thin layers.

The bottom stress is parameterized by $\tau_{bot} = c_D \rho_0 |\mathbf{u}|\mathbf{u}$, where $c_D = 0.003$ is a dimensionless drag coefficient, and ρ_0 and $\mathbf{u}$ are the density and total velocity, respectively, at the bottom of the fluid (Bleck and Smith [4]). (Here, we adopt a convention that a stress is the force per unit area exerted by an upper region on a lower region, so the drag exerted by the bottom on the fluid is $-\tau_{bot} = -c_D \rho_0 |\mathbf{u}|\mathbf{u}$). The value of the bottom velocity $\mathbf{u}$ is a mass-weighted vertical average over a prescribed vertical range, which was taken to be 5 m. The magnitude of the applied bottom stress decays linearly to zero over a distance of 1 m, in a manner analogous to what is done with the wind stress.

Vertical variations of horizontal velocity generate shear stresses between layers. Shear stress is often modeled by the relation $\tau = \rho A_V \frac{\partial \mathbf{u}}{\partial z}$, where ρ is the density and A_V is the vertical viscosity (Pedlosky [15]). Typically, A_V is orders of magnitude smaller than the horizontal viscosity A_H in (46). If the momentum equation is written in a form where velocity is the dependent variable, then the effect of shear stress is represented as $\frac{1}{\rho}\frac{\partial \tau}{\partial z} = \frac{1}{\rho}\frac{\partial}{\partial z}(\rho A_V \frac{\partial \mathbf{u}}{\partial z})$.

Now suppose that finite differences in the vertical direction are used to approximate the effect of shear stresses. In the case of a layered ocean model, it is natural to associate horizontal velocities with the centers of layers. However, the increments Δz in z can be highly irregular, due to vertical variations in layer thickness, and Taylor expansions in z show that finite differences on an irregular z-grid need not be consistent with the analytical limit. Representing the stress formula $\tau = \rho A_V \frac{\partial \mathbf{u}}{\partial z}$ is thus problematic in the present situation.

For the sake of the test computations described in Section 5, it is useful to have some kind of friction between layers, so that under steady forcing the time-dependent system will converge to a steady state that can be checked against a simple analytical solution. For present purposes, we represent the shear stress with a finite difference involving a uniform value of Δz. This value was chosen to be $\Delta z = \sqrt{2A_V/f}$, the thickness of the Ekman frictional layer (Pedlosky [15]). The stress at the interface between layers r and $r+1$ is then represented by

$$\tau_{r+1/2} = \frac{\rho A_V}{\Delta z}(\mathbf{u}_r - \mathbf{u}_{r+1}), \tag{47}$$

where ρ is a representative constant value of density. In this representation, the friction between layers is proportional to the velocity contrast between the layers, without regard to the thicknesses of the layers. In the velocity difference in (47), it suffices to use the baroclinic velocity instead of the total velocity, since $\mathbf{u}_r - \mathbf{u}_{r+1} = (\mathbf{u}'_r + \bar{\mathbf{u}}) - (\mathbf{u}'_{r+1} + \bar{\mathbf{u}})$. However, using baroclinic velocities at time t_{n+1} presents a difficulty, since the extraction procedure of Section 4.3 cannot be employed until the computation of the momentum densities $u'_r \Delta p'_r$ and $v'_r \Delta p'_r$ is complete.

We therefore use the following implicit procedure for implementing the stresses. Let $(u'_r \Delta p'_r)^*$ denote the value of $u'_r \Delta p'_r$ that would be obtained at time t_{n+1} if shear stresses, the residual term $G_x \Delta p'_r$ in (46), and the iteration of Coriolis terms were excluded. This quantity includes the effects of wind stress and bottom stress. In all layers except the top and bottom layers, define u'_r at time t_{n+1} by

$$\begin{aligned}(\Delta p'_r)u'_r &= g\big(\tau_{r-1/2} - \tau_{r+1/2}\big) + (u'_r \Delta p'_r)^* \\ &= \frac{\rho g A_V}{\Delta z}(u'_{r+1} - 2u'_r + u'_{r-1}) + (u'_r \Delta p'_r)^*. \end{aligned} \tag{48}$$

The quantity $(\Delta p'_r)$ on the left side of (48) is a layer thickness at u-points; the entire left side of (48) is *not* the variable that represents momentum density. For the top layer, we have

$$(\Delta p'_1)u'_1 = g\left(-\tau_{1+1/2}\right) + (u'_1\Delta p'_1)^* = \frac{\rho g A_V}{\Delta z}(-u'_1 + u'_2) + (u'_1\Delta p'_1)^*, \tag{49}$$

and a similar equation is obtained for the bottom layer. For each horizontal spatial position, the preceding equations constitute a simultaneous system of linear equations for the baroclinic velocities $u'_1, \ldots, u'_R$, where R is the total number of layers. If $\Delta p'_r \to 0$ with $1 < r < R$, then Eq. (48) implies that u'_r approaches the average of the values in the layers immediately above and below. Similarly, (49) implies $u'_1 \to u'_2$ as $\Delta p'_1 \to 0$.

Once the values of u'_r are obtained, the baroclinic momentum density $u'_r(\Delta p'_r)$ is obtained via a multiplication. The residual term and the iteration of Coriolis terms are then implemented, followed by the extraction process described in Section 4.3.

An implicit implementation of shear stresses was also used by Holland and Jenkins [13]. In their formulation, the stress at an interface is proportional to the square of the velocity contrast between the adjacent layers. This representation is then linearized for the sake of numerical implementation.

4.5. Bottom Topography

In the x-momentum equation (46), the horizontal pressure forcing is represented by the term

$$-\Delta p'_r\left(\frac{\partial M_r}{\partial x} - (\overline{\nabla M})_x\right). \tag{50}$$

This term is evaluated at u-points, and here we describe a method for calculating $\Delta p'_r$ at such points. Some care must be exercised in computing these values, in order to avoid false pressure gradients related to variable topography at the bottom of the fluid domain. Higdon [11] developed a representation of $\overline{\nabla M}$ that is suitable for use with variable bottom topography, and in that case the possibility of false pressure gradients is also a concern. However, the present discussion addresses an issue that would be encountered even if a barotropic–baroclinic splitting were not used.

For definiteness, consider the u-point $(x_{i-1/2}, y_j)$ which lies between the mass cells centered at (x_{i-1}, y_j) and (x_i, y_j). Two possibilities for a value of $\Delta p'_r$ at the u-point are the arithmetic mean,

$$\Delta p_r^{arith} = \frac{\Delta p'_r(x_{i-1}, y_j) + \Delta p'_r(x_i, y_j)}{2},$$

and the harmonic mean,

$$\Delta p_r^{harm} = \frac{2\Delta p'_r(x_{i-1}, y_j)\Delta p'_r(x_i, y_j)}{\Delta p'_r(x_{i-1}, y_j) + \Delta p'_r(x_i, y_j)}.$$

However, each of these possibilities has some deficiencies. For example, suppose that $\Delta p'_r(x_{i-1}, y_j) > 0$ but $\Delta p'_r(x_i, y_j) = 0$. In this case, $\Delta p_r^{arith} > 0$ and $\Delta p_r^{harm} = 0$. Following are two scenarios in which one could have such values of $\Delta p'_r(x_{i-1}, y_j)$ and $\Delta p'_r(x_i, y_j)$.

(1) The top of layer r is horizontal and intersects a sloping bottom between the mass points (x_{i-1}, y_j) and (x_i, y_j). In this case, a finite difference of M_r between these points could be nonzero, due to the greater elevation of the top of layer r at the point (x_i, y_j). However, this computed gradient has no significance in terms of the dynamics of the fluid, but instead is due entirely to the bottom topography. If this gradient is then multiplied by a positive value of $\Delta p_r'$, then a false forcing is applied to the momentum equation. In particular, a system that should remain at rest could start moving. In this situation, Δp_r^{arith} is unsuitable, whereas Δp_r^{harm} would have the desirable effect of shutting off the pressure gradient.

(2) The top of layer r is sloping and intersects a horizontal bottom between the mass points (x_{i-1}, y_j) and (x_i, y_j). In this case, the pressure gradient should remain active. However, Δp_r^{harm} would shut off the pressure gradient, whereas Δp_r^{arith} would leave it active.

These examples imply that quantities Δp_r^{arith} and Δp_r^{harm} are both unsuitable for general usage in (50). Instead, we use a method which is a variation of one used by Bleck and Smith [4]. Under the method in [4], the value of the baroclinic interface pressure $p'_{r-1/2}$ at the top of layer r at the u-point $(x_{i-1/2}, y_j)$ is defined by

$$p'_{r-1/2}\big(x_{i-1/2}, y_j\big) = \min\left(\frac{p'_{r-1/2}(x_{i-1}, y_j) + p'_{r-1/2}(x_i, y_j)}{2}, p'_b(x_{i-1}, y_j), p'_b(x_i, y_j)\right). \tag{51}$$

Here, p'_b denotes the baroclinic pressure at the bottom of the fluid. The formula (51) can be visualized as follows. Represent the bottom of the fluid domain with “stairstep” topography, in which the bottom is regarded as piecewise constant, with elevations given by cell averages. The interface pressure at a velocity point is an arithmetic mean of the pressures in the adjacent mass cells, except that an interface is not allowed to extend below the stairstep bottom. The layer thicknesses at u-points are then defined by $\Delta p'_r = p'_{r+1/2} - p'_{r-1/2}$. With this method $\Delta p'_r = \Delta p_r^{arith}$, except perhaps near the bottom of the fluid domain.

As desired, this scheme gives $\Delta p'_r = 0$ in scenario (1) discussed above, and it gives $\Delta p'_r > 0$ in scenario (2). However, it is possible to have $\Delta p'_r = 0$ when $\Delta p'_r(x_{i-1}, y_j)$ and $\Delta p'_r(x_i, y_j)$ are both positive, as in the following third scenario.

(3) Layer r has positive thickness at each of the mass points (x_{i-1}, y_j) and (x_i, y_j), but the top and bottom of layer r both lie on the bottom of the fluid domain at the intermediate u-point, as determined by (51). This situation could occur, for example, with thin layers and steep topography.

In this situation one wants $\Delta p'_r > 0$ at the u-point, since otherwise the pressure gradient would not be allowed to let the fluid in layer r flow downhill. We therefore use the following modification to (51). Starting at the bottom of the fluid and moving upward, lift each interface at the u-point (as needed) so that the thickness of each layer at the u-point is at least some fixed fraction of the minimum of the thicknesses at each of the adjacent mass points.

This procedure can be formulated as follows. Suppose that the pressure $p'_{r+1/2}(x_{i-1/2}, y_j)$ at the bottom of layer r has been determined. Let

$$\tilde{p}'_{r-1/2}\big(x_{i-1/2}, y_j\big) = p'_{r+1/2}\big(x_{i-1/2}, y_j\big) - \beta \min(\Delta p'_r(x_{i-1}, y_j), \Delta p'_r(x_i, y_j)), \tag{52}$$

where β is a constant satisfying $0 < \beta \le 1$. Then define $p'_{r-1/2}(x_{i-1/2}, y_j)$ to be the minimum of $\tilde{p}'_{r-1/2}(x_{i-1/2}, y_j)$ and the value that would be given by the formula in (51). The layer

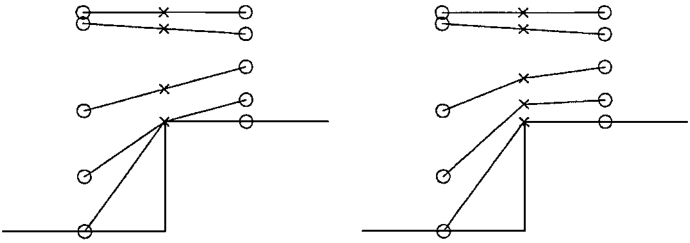

FIG. 6. Illustration of methods for calculating interface pressures at velocity points. The left frame shows the result of using the method of [4], and the right frame shows the result of using the modification developed in the present paper.

thicknesses at u-points are then defined by $\Delta p'_r = p'_{r+1/2} - p'_{r-1/2}$. This representation of layer thicknesses has the desired properties in scenarios (1)–(3) above.

Figure 6 illustrates the preceding processes. In each frame, the piecewise constant graph near the bottom illustrates the stairstep topography, the circles represent heights of layer interfaces at mass points, the x's denote heights of layer interfaces at the velocity point, and the solid lines represent the interpolation process which yields interface pressures at u-points. The left frame shows the result of using the interpolation method of Bleck and Smith [4], and the right frame shows the result of the modification described above. For this example, the value $\beta = 0.8$ was chosen (rather arbitrarily), and this is the same value used in the computations described in Section 5. The fourth layer in the left frame illustrates scenario (3) which was discussed above.

A related but different procedure was used by Holland and Jenkins [13] in order to yield $\Delta p'_r > 0$ at any velocity point for which layer r has positive thickness in either of the adjacent mass cells. The procedure in [13] was motivated by reasons related to the calculation of mass fluxes at the edges of mass cells.

In the case where the top of a fluid layer lies on the bottom of the fluid domain at a velocity point, with the modified interpolation method developed in the present paper, the physical significance of the momentum equation in that layer at that point is questionable. Instead of employing a solution of this equation at such a point, we instead set $u'_r \Delta p'_r = 0$ (or $v'_r \Delta p'_r = 0$), as the layer is massless anyway. We also set advective velocities at the edges of any such momentum cell to zero, so as to avoid a spurious loss or gain of momentum in adjacent cells. In the computations described in Section 5, this procedure is applied whenever the top of a layer is within 1 cm of the bottom of the fluid domain.

5. NUMERICAL COMPUTATIONS

Here we describe the results of some numerical computations which test the algorithms developed in this paper. In addition to testing the time-stepping method itself, these computations also test the ideas involving thin layers and bottom topography that were developed in Section 4. We consider a system that is subjected to steady forcing, and the numerical solution converges to a steady state that approximates an analytical solution having piecewise linear layer interfaces and a discontinuous velocity. In these computations, the horizontal viscosity is zero. On the other hand, when the code was modified to use the leapfrog method, the resulting algorithm was unable to complete the computation, even with substantial time and space smoothing and a greatly reduced time step.

For the test problem considered here, the fluid domain is an infinite channel having a trapezoidal cross section, and a layer interface intersects the sloping bottom. The horizontal coordinates satisfy $0 \le x \le 1000$ km, $-\infty < y < \infty$. The horizontal grid spacing is $\Delta x = \Delta y = 10$ km, so there is a total of 100 mass cells in the x-direction. Solid boundaries are imposed at $x = 0$ and $x = 1000$ by placing mass cells of zero depth on each side of the channel. From the zero-depth cell at $x = 0$, the depth of the fluid domain varies linearly to a depth of 500 m at cell 40. A similar linear variation is imposed at the other side of the channel, and the domain has a constant depth of 500 m for $400 < x < 600$.

The system is forced by a constant wind stress at the upper boundary which is directed in the positive y-direction and has a magnitude of 0.1 N/m^2. The Coriolis parameter has a constant value corresponding to latitude 45°N, i.e., $f = 2\Omega \sin 45^\circ \doteq 1.028 \times 10^{-4}$, where Ω is the angular rate of rotation of the Earth. All aspects of the problem are independent of y, so the solution is also independent of y. The infinite extent in y is simulated by using periodic boundary conditions, with a total of two mass cells in the y-direction.

The fluid consists of two layers having specific volumes $\alpha_1 = 0.975 \times 10^{-3}$ m^3/kg and $\alpha_2 = 0.970 \times 10^{-3}$ m^3/kg. At time $t = 0$, the free surface is level and located at $z = 0$, and the interface between the layers is located at $z = -50$. The interface intersects the sloping bottom on each side of the channel. The lower fluid layer is assumed to be defined at all grid points, so in the regions where only the upper layer is active, the lower layer is initially given a negligible thickness (10^{-8} m).

In these computations, the horizontal viscosity A_H is set to zero. A nonzero value of A_H is generally used in geophysical fluid computations, partly as a parameterization of sub-grid-scale turbulence and partly as a means of suppressing grid-scale numerical noise. However, the assumption $A_H = 0$ facilitates the construction of analytical solutions, so the value $A_H = 0$ was chosen for the sake of testing the algorithm. Some initial experiments were performed with $A_H > 0$, but it was found experimentally that nonzero viscosity is apparently not needed to suppress grid-scale noise with the present algorithm.

Bottom stress and shear stress are used in the model in order to counteract the wind forcing at the upper boundary and enable the system to converge to a steady state. These stresses are represented in the manner described in Section 4.4. In particular, the bottom stress is assumed to be $\tau_{bot} = c_D \rho_0 |\mathbf{u}|\mathbf{u}$, where $c_D = .003$ and $\rho_0 = 1/\alpha_2$. The shear stress at the interface between the layers is represented as $\tau_{shear} = \frac{\rho A_V}{\Delta z}(\mathbf{u}_1 - \mathbf{u}_2)$, where $\Delta z = \sqrt{2A_V/f}$ is the Ekman thickness, the vertical viscosity is $A_V = 10^{-4}$ m^2/sec, and $\rho = 1/\alpha_1$.

Given the nature of the wind forcing and friction mechanisms used here, one expects on physical grounds that the system should converge to a steady state that is constant in y and involves no motion in the x-direction. Accordingly, we characterize solutions of the governing equations (17)–(19) for which $u = 0$, $A_H = 0$, all quantities are independent of y and t, and all stresses are directed in the y-direction. Under these conditions, the momentum equations in layer r are

$$-f v_r = -\frac{\partial M_r}{\partial x} \tag{53}$$

$$0 = \Delta\tau = \tau_{r-1/2} - \tau_{r+1/2} \tag{54}$$

for $r = 1, 2$. Equation (53) represents a geostrophic balance between the Coriolis term and the pressure gradient, and as seen below, Eq. (54) represents a balance between wind stress and bottom stress. The layer thickness equation is not included here, as all of the terms in that equation are zero in the present situation. The jump condition (19) is

$$M_1 - M_2 = p_{int}(\alpha_1 - \alpha_2), \tag{55}$$

where p_{int} is the pressure at the interface.

First consider the case where layer r is the only active layer on a given interval in x, i.e., the other layer has zero thickness in that interval. In that case, $\tau_{r-1/2}$ is the wind stress, and $\tau_{r+1/2}$ is the bottom stress. Equation (54) then implies $\tau_{wind} = \tau_{bot} = (c_D/\alpha_2)v_r^2$, and thus

$$v_r = \left(\frac{\alpha_2 \tau_{wind}}{c_D}\right)^{1/2}. \tag{56}$$

Equations (53) and (56) then determine $\partial M_r/\partial x$. But $M_1 = \alpha_1 p_{top} + g z_{top} = g z_{top}$, where z_{top} is the elevation of the free surface. If layer 2 is the active layer, then layer 1 is massless, and the jump condition (55) implies $M_2 = M_1$. The Montgomery potential in the active layer therefore satisfies $M_r = g z_{top}$, whether $r = 1$ or $r = 2$. We then have $\partial z_{top}/\partial x = f v_r/g$, where v_r is given by (56). With the parameters used in the present situation, $v_r \doteq 0.180$ m/s, and $\partial z_{top}/\partial x \doteq 1.89 \times 10^{-6}$.

Next consider the case where both layers have positive thickness on an interval in x. Equation (54) implies $\Delta\tau = 0$ in each layer, so $\tau_{wind} = \tau_{shear} = \tau_{bot} = (c_D/\alpha_2)v_2^2$ and thus $v_2 = (\alpha_2 \tau_{wind}/c_D)^{1/2}$. The representation (47) of shear stress, with $\rho = 1/\alpha_1$, implies $v_1 = v_2 + (\tau_{shear}\alpha_1 \Delta z)/A_V$. The slope of the free surface is then $\partial z_{top}/\partial x = f v_1/g$. The slope of the interface between the layers can be characterized as follows. If z_{int} denotes the elevation of this interface, then the pressure at the interface is $p_{int} = (g/\alpha_1)(z_{top} - z_{int})$, and the jump condition (55) implies $M_1 - M_2 = (g\Delta\alpha/\alpha_1)(z_{top} - z_{int})$. Here, $\Delta\alpha = \alpha_1 - \alpha_2$. Thus,

$$\frac{\partial z_{int}}{\partial x} = \frac{\partial z_{top}}{\partial x} + \frac{\alpha_1}{g\Delta\alpha}\frac{\partial}{\partial x}(M_2 - M_1) = \frac{f v_1}{g} + \frac{f\alpha_1}{g\Delta\alpha}(v_2 - v_1). \tag{57}$$

With the parameters used in the present computations, we have $v_1 \doteq 1.54$ m/s, $v_2 \doteq 0.180$ m/s, $\partial z_{top}/\partial x \doteq 1.61 \times 10^{-5}$, and $\partial z_{int}/\partial x \doteq -0.00276$.

The time step was chosen to be $\Delta t = 2400$ s, for the following reason. If the fluid domain had a constant depth of 500 m, then the given choices of densities and initial layer thicknesses would imply that the external wave speed is $c_0 \doteq 70.0$ m/s and the internal wave speed is $c_1 \doteq 1.51$ m/s (see (26), (27)). If v denotes the largest component of velocity encountered in the steady state, then $(c_1 + v)\Delta t/\Delta x \doteq 0.73 < 1$. Choosing $\Delta t = 2400$ leaves some room for larger values of v and nonzero values of u during the transient phase of the computation.

When MPDATA is used to solve the momentum and layer thickness equations, two anti-diffusive corrections are used. Some experiments were performed with different numbers of iterations. For the particular problem configuration used here, two corrections gave slightly better results in some places than did one correction, whereas three corrections did not seem to make a noticeable difference.

In these computations the system is initially at rest, and beginning at $t = 0$ the uniform wind stress is applied to the upper surface of the fluid. Because of the Coriolis effect, the fluid in the upper layer shifts in the positive x-direction, so the fluid in the lower layer wells upward at the left end of the domain. In particular, the fluid interface must move upward

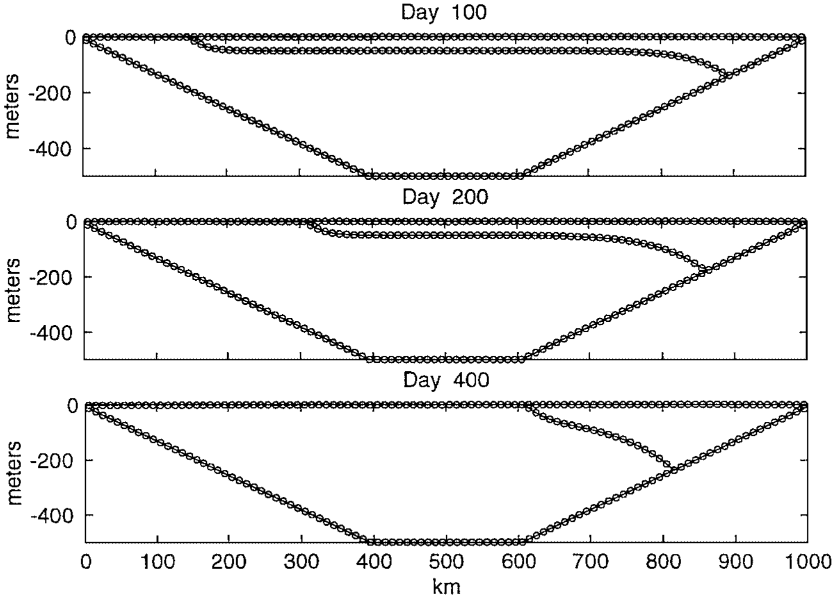

FIG. 7. Cross sections of the fluid domain for fixed y. The horizontal axis represents the horizontal position x in kilometers, and the vertical coordinate z is given in meters. The free surface is located at $z \approx 0$. The bottom of the domain is represented by the trapezoidal, piecewise linear function that varies between $z = 0$ and $z = -500$. In each frame, the interface between the two fluid layers is indicated by the curve between the free surface and the bottom. The dots indicate vertical coordinates associated with individual mass cells. At time $t = 0$, the interface is horizontal and located at $z = -50$. For $t > 0$, a uniform wind stress is applied at the free surface in the positive y-direction, which is into the page. Due to the Coriolis effect, the fluid in the upper layer is shifted to the right, so the fluid in the lower layer wells upward along the left end of the domain.

along the sloping bottom, and eventually the lower layer reaches the upper surface. At the right end, the upper layer becomes thicker, and the interface must move downward along the sloping bottom. This aspect of the problem tests the ability of the model to move mass into empty cells and drain cells that initially contain some fluid.

Figure 7 shows cross sections of the fluid domain at model days 100, 200, and 400. By day 100, the interface has already migrated to the upper surface at the left end of the domain. Over the time period shown in Fig. 7, the left edge of the upper layer propagates rightward.

The lateral transport of fluid in the upper layer can be estimated as follows. For the transient problem, suppose that the dominant balance in the y-component of the momentum equation (17) is between the Coriolis term and the wind stress, so that the fluid motion consists of the “Ekman transport” (Gill [8]). Then $fu_1 \approx g\tau_{wind}/\Delta p = \alpha_1 \tau_{wind}/\Delta z$, where Δz is the vertical distance across layer 1. With the parameters used in the present problem, $u_1 \approx 1.9$ cm/s. Movement at this rate over 100 days yields a horizontal displacement of approximately 160 km, which agrees with the displacements indicated in Fig. 7. The time evolution of the system is thus consistent with what is to be expected in the present circumstances.

After about 800 to 1000 days, the system reaches an approximate steady state. The integration was continued to 4000 days, but there was little change in the solution between days 1000 and 4000. Some plots of the solution at day 1200 are given in Fig. 8. In the bottom frame, the solid horizontal lines for $x > 715$ illustrate the discontinuous

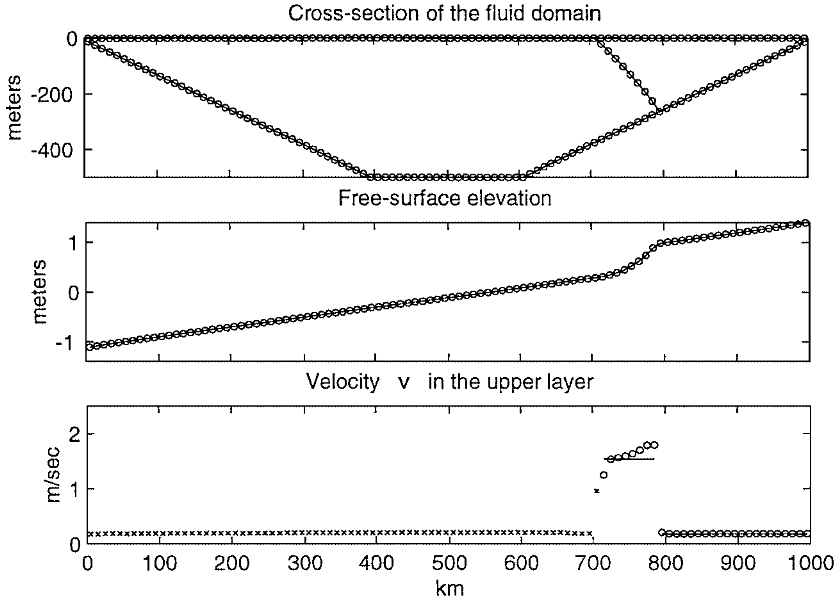

FIG. 8. Plots of the steady-state solution. After about 800 to 1000 model days, the solution reaches an approximate steady state, and the present figure shows plots of the solution at day 1200. The top frame shows a cross section of the fluid domain. In this case, the upper fluid layer is confined to a triangular region in the upper right corner, and elsewhere the upper layer has negligible thickness. Near the right end of the domain, the lower layer has negligible thickness. The middle frame shows the elevation of the free surface relative to the initial elevation $z = 0$. The bottom frame shows the velocity v_1, which is the component of velocity in the upper layer along the direction of the channel; relative to the plots shown in the upper two frames, this direction is into the page. The solid horizontal lines for $715 < x < 785$ and $785 < x < 1000$ are values of velocity found in an analytical steady-state solution characterized by geostrophic balance and a balance between wind stress and bottom stress. The x's in the bottom frame are the computed velocities at locations where the upper layer has negligible thickness, and these have little physical significance. The text contains comparisons of the slopes of the free surface and interface with the slopes found in the analytical steady state.

velocity v_1 in the steady-state solution discussed above. The small circles indicate the computed values of v_1 on that interval. For $x < 715$, the upper layer has negligible thickness, and the velocities computed on that interval have little physical significance. These values of velocity are plotted with x's in Fig. 8. These values are essentially equal to the values of v_2 on that interval, due to the effect of the shear stress as described in Section 4.4.

We next compare the slopes of the free surface and interface with the values obtained earlier for analytical steady-state solutions. First, there is a span of approximately eight mass cells on which both layers are active. A linear least-squares fit to the interface elevations at those points yields a slope of approximately -0.00290, whereas the slope in the analytical steady state discussed above is approximately -0.00276. On this interval, the free-surface elevation is rather smeared, with a maximum slope of approximately 1.48×10^{-5}; in the analytical case, the slope is approximately 1.61×10^{-5}. On the remainder of the domain only one layer is active, and in the analytical case the slope of the free surface is approximately 1.89×10^{-6}. On the left end of the domain, the slope of the computed free surface is approximately 1.98×10^{-6}, and on the right end the slope is approximately 2.02×10^{-6}.

For the sake of comparing numerical methods, the code was modified to use the leapfrog method for the baroclinic momentum equation. This amounted to deleting the correction steps and making some modifications to the remaining routines. However, the two-level implementation of MPDATA was retained for the layer thickness equation, due to the necessity of preserving nonnegative layer thickness. This kind of mixed time differencing is also used in the Miami Isopycnic Coordinate Ocean Model ([3, 4]). This author is not aware of any nonoscillatory advection schemes for the leapfrog method, so the nonlinear and Coriolis terms were combined into the form $(\mathbf{u}_r \cdot \nabla)\mathbf{u}_r + f\mathbf{k} \times \mathbf{u}_r' = \nabla(\frac{1}{2}|\mathbf{u}_r|^2) + (\zeta_r + f)\mathbf{k} \times \mathbf{u}_r' + \zeta_r \mathbf{k} \times \bar{\mathbf{u}}$, where $\zeta_r = \partial v_r/\partial x - \partial u_r/\partial y$, and the enstrophy-conserving scheme of Sadourny [16] was then used to discretize the result. This scheme uses centered differencing and averaging; in particular, the vorticity term $(\zeta_r + f)\mathbf{k} \times \mathbf{u}_r'$ is written in terms of potential vorticity and mass flux as $(\frac{\zeta_r + f}{\Delta p_r'})\mathbf{k} \times (\mathbf{u}_r' \Delta p_r')$, and the terms in this expression are represented by various second-order spatial averages on a C-grid. The division by layer thickness is problematic in the case of thin layers, so in that case an analogous formulation was used which does not involve multiplication and division by layer thickness. The code employs a continuous transition between these two cases. In keeping with the usage of momentum densities as dependent variables, the preceding representations were then multiplied by $\Delta p_r'$ at velocity points. The Sadourny scheme is a classical scheme which is widely regarded as inhibiting the nonlinear cascade of energy to short-wavelength energy on grid scales. The usage of centered differencing in the nonlinear terms may place the leapfrog method at a disadvantage relative to a two-level scheme that employs a nearly nonoscillatory advection scheme, but this illustrates part of the reason for using a two-level method.

The revised code could not run for long with the model parameters used above, even with a reduced time step, so some horizontal viscosity and an Asselin filter were introduced. For most of the subsequent computations, the horizontal viscosity was set at $A_H = 100\ \mathrm{m}^2/\mathrm{s}$. In the initial stages of the computation, the fluid velocity is on the order of a few tenths of a meter per second. Relative to a fluid velocity of $u = 0.1$ m/s and a grid spacing of 10,000 m, the value $A_H = 100\ \mathrm{m}^2/\mathrm{s}$ gives a Reynolds number of $u\Delta x/A_h = 10$. The horizontal diffusion term was evaluated at time t_{n-1} in order to avoid a leapfrog approximation to diffusion, which would be unstable. In the Asselin filter, the weighting coefficients were (0.125, 0.75, 0.125). This is a stronger filter than the case (0.05, 0.90, 0.05) which displayed strong damping in Fig. 2. In addition, the velocity-limiting procedure of Section 4.3 was applied to all layers with thicknesses less than 1 m instead of 1 cm, and the wind stress and bottom stress were modified to decay linearly to zero over a vertical distance of 10 m instead of 1 m. With these modifications, and with a succession of smaller time steps down to 300 s, the computations crashed, usually after about 100 model days. Increasing the horizontal viscosity from $A_H = 100\ \mathrm{m}^2/\mathrm{s}$ to $A_H = 1000\ \mathrm{m}^2/\mathrm{s}$ made little difference. Diagnostic output for various runs indicated that the model was producing reasonable results for most of the period of integration, and apparently the problem was related to localized large velocities allowed by the centered space and time differencing.

Some additional experiments were performed to check execution times for the codes used in these computations. All of the experiments were performed on a serial workstation. If two antidiffusive corrections are used when MPDATA is used to solve the momentum and layer thickness equations, then the execution time for one time step with the two-level code is about 2.3 times as much as the execution time for one time step with the leapfrog version. If one correction is used, then the execution time is 2.0 times as much as for leapfrog. However, under a linearized stability analysis, the allowable time step for the leapfrog method is less than half the time step allowed by the present two-level method, so the two-level method does not suffer a performance penalty relative to the leapfrog method.

Further experiments were performed in order to check execution times of certain individual aspects of the two-level algorithm, as coded for these particular computations. The following remarks assume that two antidiffusive corrections are used with MPDATA. In this situation, the antidiffusive corrections by themselves account for about 28.5% of the total execution time. During the correction step for the baroclinic momentum equations, the Coriolis terms are iterated four times, and this consumes about 6.5% of the time. As noted in Section 4.2, the residual terms G_x and G_y are used to enforce the zero-sum condition on the momentum densities after each iteration, and this process is included in the 6.5%. In this time-stepping algorithm, the barotropic equations are predicted and then corrected, and these two steps together account for a total of about 10.5% of the execution time.

6. SUMMARY

This paper describes a two-level time-stepping method for use with barotropic–baroclinic splittings for layered ocean circulation models. Numerical models of ocean circulation typically admit motions varying on widely differing timescales, and splittings are often used to separate the fast and slow motions into subsystems that are solved by different techniques.

The present method involves a prediction and a correction of some of the dependent variables. After the initial prediction steps, all subsequent steps are based on centered differencing and averaging about the intermediate time $t_{n+1/2} = (t_n + t_{n+1})/2$. In particular, the sequencing of steps means that an advective velocity is available at time $t_{n+1/2}$ when solving the equations for layer thickness and momentum densities in the slow (baroclinic) system. Such a velocity is needed when employing second-order nonoscillatory advection algorithms. In this time-stepping scheme the fast (barotropic) system is predicted and then corrected. However, solving this system twice does not involve a large computational cost, as it is solved implicitly.

When the time-stepping method is applied to a simple test problem involving the linearized shallow-water equations for a single-layer fluid, it displays less phase error than the leapfrog method and two other two-level methods that were developed for barotropic–baroclinic splitting. In addition, it does not generate any amplitude error. The differences are especially pronounced when comparing the present method with the leapfrog method.

In a layered ocean model, interfaces between layers can intersect the top and/or bottom of the fluid domain. A common way to deal with this situation is to define all layers at all grid points and allow for the possibility of layers having negligible thicknesses in some locations. For the present numerical method, the presence of thin layers and variable bottom topography raises issues which are discussed in the paper. The new scheme produces good results in a nonlinear test problem involving vanishing layers at the top and bottom of the fluid domain and a fluid interface moving upward and downward along sloping bottom topography. In these computations, the horizontal viscosity is set to zero, and a nearly nonoscillatory advection algorithm is used to solve the equations for mass and momentum.

ACKNOWLEDGMENTS

I thank Rainer Bleck, Roland de Szoeke, John Dukowicz, and Scott Springer for useful conversations related to the contents of this paper. In particular, Springer suggested that the Bates method be used to solve the barotropic equations implicitly, and he also suggested that the shear stresses be implemented implicitly. The algorithms developed in this paper were implemented by modifying the Parallel Oregon State University Model, of which Springer is the principal creator. This work was supported by National Science Foundation Grant DMS-9803331.

REFERENCES

1. R. Asselin, Frequency filter for time integrations, *Mon. Weather Rev.* **100**, 487 (1972).
2. J. R. Bates, An efficient semi-Lagrangian and alternating direction implicit method for integrating the shallow water equations, *Mon. Weather Rev.* **112**, 2033 (1984).
3. R. Bleck, C. Rooth, D. Hu, and L. T. Smith, Salinity-driven thermocline transients in a wind- and thermohaline-forced isopycnic coordinate model of the North Atlantic, *J. Phys. Oceanogr.* **22**, 1486 (1992).
4. R. Bleck and L. T. Smith, A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean 1. Model development and supporting experiments, *J. Geophys. Res. C* **95**, 3273 (1990).
5. J. K. Dukowicz and R. D. Smith, Implicit free-surface method for the Bryan–Cox–Semtner ocean model, *J. Geophys. Res. C* **99**, 7991 (1994).
6. D. R. Durran, *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics* (Springer-Verlag, New York, 1999).
7. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1971).
8. A. E. Gill, *Atmosphere–Ocean Dynamics* (Academic Press, San Diego, 1982).
9. R. Hallberg, Stable split time stepping schemes for large-scale ocean modeling, *J. Comput. Phys.* **135**, 54 (1997).
10. G. J. Haltiner and R. T. Williams, *Numerical Prediction and Dynamic Meteorology* (Wiley, New York, 1980).
11. R. L. Higdon, Implementation of a barotropic–baroclinic time splitting for isopycnic coordinate ocean modeling, *J. Comput. Phys.* **148**, 579 (1999), doi:10.1006/jcph.1998.6130.
12. R. L. Higdon and R. A. de Szoeke, Barotropic–baroclinic time splitting for ocean circulation modeling, *J. Comput. Phys.* **135**, 30 (1997), doi:10.1006/jcph.1997.5733.
13. D. M. Holland and A. Jenkins, Adaptation of an isopycnic coordinate ocean model for the study of circulation beneath ice shelves, *Mon. Weather Rev.* **129**, 1905 (2001).
14. F. Mesinger and A. Arakawa, *Numerical Methods Used in Atmospheric Models*, GARP Publ. Ser. No. 17, Vol. 1. (WMO–ICSU Joint Organizing Committee, Geneva, 1976).
15. J. Pedlosky, *Geophysical Fluid Dynamics*, 2nd ed. (Springer-Verlag, New York, 1987).
16. R. Sadourny, The dynamics of finite-difference models of the shallow-water equations, *J. Atmos. Sci.* **32**, 680 (1975).
17. C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations, *J. Comput. Phys.* **128**, 101 (1996).
18. P. K. Smolarkiewicz and L. G. Margolin, MPDATA: A finite-difference solver for geophysical flows, *J. Comput. Phys.* **140**, 459 (1998), doi:10.1006/jcph.1998.5901.
19. G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **5**, 506 (1968).
20. L. N. Trefethen, Group velocity in finite difference schemes, *SIAM Rev.* **24**, 113 (1982).
21. J. Wang and M. Ikeda, Inertial stability and phase error of time integration schemes in ocean general circulation models, *Mon. Weather Rev.* **125**, 2316 (1997).