JOURNAL OF COMPUTATIONAL PHYSICS 37, 355–370 (1980)

A Geodesic Finite-Difference Method for Curved Domains: Simulations of Tidal Motion on a Sphere

W. C. THACKER

Sea–Air Interaction Laboratory, Atlantic Oceanographic and Meteorological Laboratories National Oceanic and Atmospheric Administration, Miami, Florida 33149

Received July 2, 1979; revised November 13, 1979

This finite-difference method for approximating the solutions of partial-differential equations on curved surfaces has two attractive features. First, it offers considerable flexibility in the design of the computational mesh, allowing points to be distributed as desired throughout the curved domain, either uniformly or to achieve variable resolution. Second, the use of three-dimensional Cartesian coordinates rather than two-dimensional curvilinear surface coordinates facilitates computations on geometrically irregular surfaces. Computational tests demonstrate that the method can accurately reproduce time-dependent solutions of Laplace's tidal equations on a sphere, using a large time-step and a quasi-homogeneous grid without any need for filtering or smoothing. This method should be ideally suited for computing oceanic circulation or global-scale weather.

I. INTRODUCTION

The method for solving partial-differential equations on curved surfaces which is presented here is a straightforward generalization of a method for flat surfaces that was developed for numerically forecasting storm surges [10]. The characteristic feature of this method is that it allows the use of an irregular computational mesh. For storm surge calculations, boundary points can coincide with the curving coastlines of bays and barrier islands, and interior points can be spaced more closely in shallow regions and further apart in deep regions in order to avoid unnecessary computational expense while restricting numerical dispersion to a uniform level over the entire basin [12]. For time-dependent problems such as storm surge forecasting, this method is computationally more economical than finite-element methods, which also can be used with irregular computational grids [11].

A problem similar to that of forecasting surges is the prediction of tides for the world oceans. Again, the computational mesh should be tailored to the bathymetry of the oceans, and its boundaries to the shapes of the continents. However, on the global scale, it is necessary to take into account the spherical shape of the earth. This can be done by constraining the grid points to lie on the curved surface of a sphere. As long as the points are closely enough spaced so that the grid appears to be locally flat everywhere, with little modification the finite-difference formulas for irregular planar grids should also provide useful approximations to partial derivatives on curved grids.

0021-9991/80/120355-16$02.00/0

Grid design can also be important for computing the general circulation of the atmosphere. The traditional approach has been to cast the equations in spherical polar coordinates and to approximate their solutions on a computational mesh formed by the intersection of parallels of latitude with longitudinal meridians. The expense of the unneeded resolution provided by the small grid intervals near the poles is compounded by the fact that the smallest interval sets the limit for the maximum size of a stable explicit time-step. This fact has motivated the use of computational grids which are quasi-uniform over the entire globe. Kurihara [3] introduced a grid for which points were situated on circles of constant latitude but with the number of points per circle decreasing linearly with increasing latitude. Williamson [15] and Sadourny *et al.* [9] broke completely with the tradition of restricting points to meridians and parallels by using geodesic grids which are formed by subdividing the faces of icosohedra into smaller triangles and then mapping those triangles onto the surface of a concentric sphere. The approach taken by Williamson [16] to integrate the barotropic primitive equations on such a geodesic grid is quite similar to that presented here, the difference being primarily in the formulas used for approximating partial derivatives. Although he found that an unexpectedly small time-step was needed to avoid computational instability, the method presented here has the attractive feature of allowing for an unusually large explicit time-step. Another approach to the problem of uniform resolution over the sphere is to project the surface of the sphere onto the faces of a concentric cube, or other regular polyhedron, and to use a separate coordinate system and uniform grid for each face. Sadourny [8] found that the internal boundaries corresponding to the edges of the cube were the source of two-grid interval waves which had to be removed by a damping operator. Williamson [16] found that his scheme also required smoothing to remove numerical noise, as did Cullen [1] for his finite-element integrations of the barotropic primitive equations using a similar icosohedral grid. The method presented here yielded good results without resorting to any filtering technique.

The finite-difference method presented here can be used to solve partial-differential equations on irregularly curved surfaces and is not at all limited to the special case of a spherical domain. For example, it could be used to calculate the conduction of heat through a twisted sheet of metal. A curvilinear coordinate system for the curved surface is not required, because three-dimensional Cartesian coordinates can be used to specify the location of points on the curved computational mesh and the surface gradient vector can be resolved into Cartesian components. Although the gradient vector must be parallel to the surface, there is no need to know how to project its components onto directions corresponding to surface coordinates. What is important is to be able to specify the direction of the surface normal at each grid point. An appropriate normal vector, defined in terms of cross products of the position vectors of neighboring grid points, is incorporated into the finite-difference formula in order to guarantee that the gradient vector is indeed parallel to the curved domain.

In order to demonstrate that this method does provide good approximations to the solutions of partial-differential equations, it is necessary to be able to compare the results of computations to known exact solutions. Such solutions are generally

unknown for irregular surfaces. However, Laplace's tidal equations on a sphere, which contain the essential physics of large-scale oceanic and atmospheric circulation, have closed-form solutions corresponding to normal mode oscillations. The numerical simulation of these oscillations provides an excellent test of this method within an important practical context. Because the computations are made entirely without reference to spherical coordinates, it seems fair to conclude that this method should work equally well on non-spherical surfaces.

II. Approximation of Partial Derivatives

Formulas for approximating partial derivatives on irregular planar grids were first obtained by Noh [6]. He constructed the formulas so that Green's theorem relating line integrals to surface integrals would be satisfied, guaranteeing that appropriate quantities would be conserved by his finite-difference equations. Later they were also derived by approximating the partial derivatives of a function by the slopes of triangular facets that comprise a piecewise-linear approximation to the function [10]. These formulas can be written quite simply as a single vector equation,

$$(\overline{\nabla f})_j = \frac{\sum_{i=1}^{N} f_i(\mathbf{r}_{i+1} - \mathbf{r}_{i-1}) \times \hat{n}_j}{\sum_{i=1}^{N} (\mathbf{r}_i \times \mathbf{r}_{i+1}) \cdot \hat{n}_j}. \tag{1}$$

The subscripts indicate that the value of the derivative is approximated at the jth grid point in terms of the values of the function at N neighboring grid points with indices i increasing, modulo N, in a counterclockwise direction about j. The three-dimensional vectors $\mathbf{r}_i$ are the position vectors of the grid points and the unit vector $\hat{n}_j$ is normal to the plane of the computational grid. For a planar grid, the subscript on the unit vector is superfluous. The magnitude of the denominator is twice the area of the polygon formed by connecting consecutive neighboring points. The overbar is used to distinguish this finite-difference form from the value of the derivative that it approximates.

Although $(\overline{\nabla f})_j$ is a three-dimensional vector, $(\overline{\nabla f})_j \cdot \hat{n}_j = 0$, so its components are parallel to the plane of the computational grid. Thus, it has only two independent components, corresponding to the two partial derivatives of a function of two variables. Equation (1) simply allows those two variables to define a plane having an arbitrary orientation with respect to a three-dimensional Cartesian coordinate system. If this plane is thought of as being tangent to a curved surface at $\mathbf{r}_j$, then in that vicinity the finite-difference form should provide a good approximation to the derivatives on the curved surface, so long as both the function and the curved domain vary slowly on the scale set by the separation of the grid points. This suggests that Eq. (1) can also be used to approximate derivatives on curved surfaces, if $\hat{n}_j$ is understood to be the normal to the surface at $\mathbf{r}_j$.

For a spherical surface, the direction of the normal at any point can be determined from the values of the spherical polar coordinates of that point. However, a

corresponding system of surface coordinates is not readily available for a general, irregularly curved surface, so it is convenient to be able to define the normal direction without reference to curvilinear surface coordinates. If the grid points are thought of as being the vertices of triangular facets of a piecewise-linear geodesic surface that approximates the curved computational domain, then it is clear that $\hat{n}_j$ can be approximated as an average of the unit normal vectors of the facets surrounding $\mathbf{r}_j$. The vector $\mathbf{A}_{ij} = (\mathbf{r}_i - \mathbf{r}_j) \times (\mathbf{r}_{i+1} - \mathbf{r}_j)$ is normal to the facet with vertices $\mathbf{r}_i$, $\mathbf{r}_{i+1}$, and $\mathbf{r}_j$, so the vector $\mathbf{A}_j = \sum_{i=1}^{N} \mathbf{A}_{ij}$ is in the direction of the area-weighted average of the unit normals of the facets surrounding $\mathbf{r}_j$. The vector, $\mathbf{A}_j$, is a simple function of the position vectors of the neighboring points,

$$\mathbf{A}_j = \sum_{i=1}^{N} \mathbf{r}_i \times \mathbf{r}_{i+1}, \tag{2}$$

and the unit normal vector is simply

$$\hat{n}_j = \frac{\mathbf{A}_j}{|\mathbf{A}_j|}. \tag{3}$$

It is interesting to note that the magnitude of $\mathbf{A}_j$ equals twice the sum of the areas of the facets surrounding $\mathbf{r}_j$ only if the grid is planar; for curved grids, it is less.

Equations (1), (2), and (3) can be combined to give a finite-difference formula for use with non-planar grids,

$$\begin{aligned} \overline{(\nabla f)}_j &= \sum_{i=1}^{N} \boldsymbol{\alpha}_{ij} f_i, \\ \boldsymbol{\alpha}_{ij} &= (\mathbf{r}_{i+1} - \mathbf{r}_{i-1}) \times \left(\frac{\mathbf{A}_j}{\mathbf{A}_j \cdot \mathbf{A}_j} \right). \end{aligned} \tag{4}$$

The coefficients $\boldsymbol{\alpha}_{ij}$ (relating the value of the derivative at j to the value of the function at i) are generalizations of the familiar factors, $\pm(2\Delta x)^{-1}$, for a one-dimensional linear grid of points Δx apart.

The differentiation coefficients are not independent; the sum of the coefficients of the neighboring points must vanish, and if the number of neighbors is even, the sums of coefficients of alternate points must separately vanish. This allows computationally more efficient formulas to be written,

$$\begin{aligned} \overline{(\nabla f)}_j &= \sum_{i=1}^{N-1} \boldsymbol{\alpha}_{ij}(f_i - f_N), \qquad \text{for } N \text{ odd}, \\ \overline{(\nabla f)}_j &= \sum_{\substack{i=1 \\ \text{odd}}}^{N-3} \boldsymbol{\alpha}_{ij}(f_i - f_{N-1}) + \sum_{\substack{i=2 \\ \text{even}}}^{N-2} \boldsymbol{\alpha}_{ij}(f_i - f_N), \qquad \text{for } N \text{ even}. \end{aligned} \tag{5}$$

For an odd number of neighbors, $N-1$ coefficients must be calculated, and for an even number, $N-2$. If the computations are referenced to a system of coordinates within the curved surface, only two components of the coefficients are needed. All three components were used for the examples presented here in order to demonstrate that a system of surface coordinates is unnecessary. The coefficients were computed once and tabulated. Since the computational grid was constructed so that all points had either five or six neighbors, four multiplications, four subtractions, and three additions were required to compute the partial derivative in each direction for each grid point.

The examples considered here involve equations with only first derivatives, but this is not a restriction. Higher derivatives can be computed simply by repeated application of the finite-difference formulas (5).

III. The Computational Mesh

The formulas for approximating partial derivatives offer the freedom to design the grid to suit the computation for which it is intended. Although in general there may be a problem in recognizing a priori an advantageous distribution of mesh points, this is not the case for computations based upon Laplace's tidal equations on a sphere. If the sphere does not rotate and the depth of the undisturbed fluid is the same everywhere, then the spherical symmetry of the equations indicates that the variability of the solutions should be uniform over the sphere and therefore, that a uniform computational mesh should be appropriate. However, if the sphere does rotate, that symmetry is destroyed and resolution which varies in the direction of the axis of rotation might be more appropriate. Indeed, Longuet-Higgins [5] points out that, as the rate of rotation gets very large, the variability of free oscillations is concentrated near the equator, implying that the grid spacing should increase from the poles toward the equator, in the opposite sense to what is available using a grid formed by the intersections of meridians with parallels; on the other hand, the "negative-depth" solutions, which are important for forced motion, are most variable near the poles. The fact that the length of a wave is diminished as it propagates into shallow water indicates that, when the depth is not uniform, the mesh should be denser for the shallow region than for the deep regions. The spacing should be proportional to the phase speed (or to the square root of the depth), if the number of grid points per wavelength is to be the same everywhere. The closed-form solutions which provide tests of the finite-difference method correspond to the case of uniform depth and moderate rotation, so a homogeneous mesh is appropriate for the test computations.

Experience with planar grids [10, 11] has shown that computational accuracy can be diminished if the grid is not constructed so that each point is at the center of the polygon formed by its closest neighbors. If neighboring points are thought of as being connected by springs, then the equilibrium configuration of the spring system corresponds to a centered grid [13]. For points constrained to lie on the surface of a sphere

of radius R, the position of each grid point is related to the positions of its neighbors through a system of simultaneous non-linear equations,

$$\mathbf{x}_j = \frac{R \sum_{i=1}^{N} \mathbf{x}_i}{\left| \sum_{i=1}^{N} \mathbf{x}_i \right|}. \tag{6}$$

Before Eqs. (6) can be solved to determine the location of the grid points, it is necessary to know which points should be neighbors. In order to achieve a quasi-uniform distribution of points over the sphere, the neighbors were specified so that the mesh would be topologically equivalent to one obtained by subdividing each face of a regular icosohedron into n^2 equilateral triangles. Equations (6) were solved using an iterative technique of the Gauss–Seidel type, for the location of all the points except those corresponding to the vertices of the icosohedron. These were held fixed in order to avoid unwanted solutions such as the one for which all points are located at the same spot. The iterations were initialized by placing the points along $3n - 1$ latitudinal circles in order to achieve a fairly homogeneous approximation to the final grid.

Although this grid is topologically equivalent to the grids used by Sadourny *et al.* [9], Cullen [1], Hinsman [2], and Platzman [7], the details of the grid geometry are different. For example, the maximum, average, and minimum angular separations of neighboring grid points as given by Platzman [7] for the configuration specified by $n = 10$ are, respectively, 7.72, 6.92, and 6.34 degrees. The corresponding values for the topologically equivalent centered grid are 7.26, 6.87, and 4.75. The requirement

TABLE 1

Features of Centered Quasi-Uniform Spherical Grids

	$n = 5$	$n = 10$	$n = 15$
Number of points	252	1002	2252
Maximum distance between neighboring points $/2\pi R$	0.0402	0.0202	0.0135
Average distance between neighboring points $/2\pi R$	0.0381	0.0191	0.0127
Minimum distance between neighboring points $/2\pi R$	0.0302	0.0132	0.0081
Sum of areas of triangular facets $/4\pi R^2$	0.9878	0.9969	0.9986
Maximum angle between computational normal and true normal in radians	0.0191	0.0092	0.0031

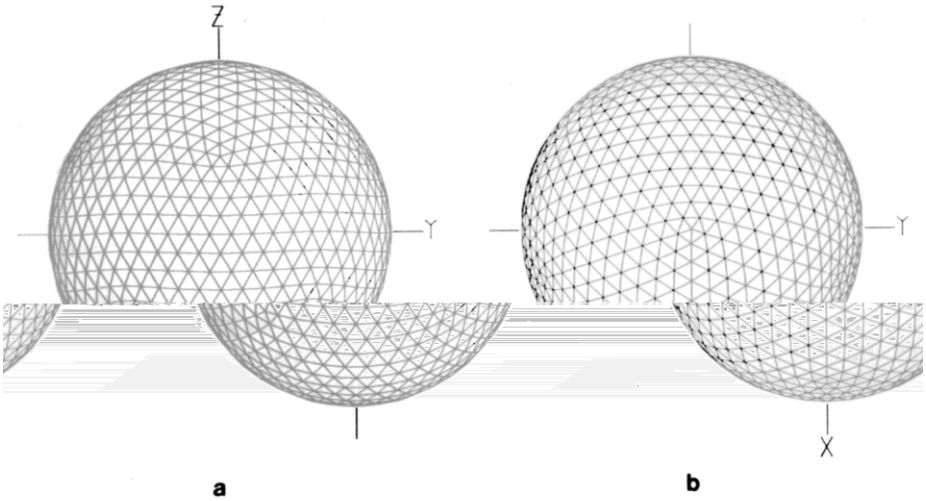

FIG. 1. The mesh used for test computations is shown as viewed from a distant point (a) along the x-axis and (b) along the z-axis.

that the points be centered causes the grid spacing to be smaller in the vicinity of the 12 points which correspond to the corners of the icosohedron than they would be using the other methods of grid construction. Properties of centered grids are listed in Table I for $n = 5$, 10, and 15. Test computations were made with the grid illustrated in Fig. 1, corresponding to $n = 10$ and having 1002 points.

IV. FINITE-DIFFERENCE SIMULATIONS OF TIDAL OSCILLATIONS

The free motion of a thin layer of fluid held by gravity to a rotating sphere is governed by Laplace's tidal equations,

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{f} \cdot \hat{n})\hat{n} \times \mathbf{U} + gD\nabla H = 0, \qquad \frac{\partial H}{\partial t} + \nabla \cdot \mathbf{U} = 0, \tag{7}$$

which express the conservation of momentum and mass of the fluid. The variables, $\mathbf{U}$ and H, are the vertical integral of the horizontal velocity and the surface elevation of the fluid, respectively; and the parameters $\mathbf{f}$, g, and D, are respectively the Coriolis vector (equal to twice the angular velocity of the sphere), the acceleration of gravity, and the depth of the fluid. Note that only the component of the Coriolis vector perpendicular to the surface contributes to the Coriolis force term because $\mathbf{U}$ is parallel

to the surface of the sphere. Likewise, the gradient operator, ∇, is parallel to the surface.

The finite-difference analogues of these equations are

$$\frac{1}{\tau}(\mathbf{U}_j^{n+1/2} - \mathbf{U}_j^{n-1/2}) + \frac{1}{2}(\mathbf{f}\cdot\hat{n}_j)\hat{n}_j \times (\mathbf{U}_j^{n+1/2} + \mathbf{U}_j^{n-1/2}) + gD(\overline{\nabla H})_j^n = 0, \tag{8}$$

$$\frac{1}{\tau}(H_j^{n+1} - H_j^n) + (\overline{\nabla\cdot\mathbf{U}})_j^{n+1/2} = 0.$$

The variables, $\mathbf{U}_j^{n-1/2}$ and H_j^n, provide approximations to $\mathbf{U}$ and H at the grid points, $\mathbf{x}_j$, and time levels, $t = (n - \frac{1}{2})\tau$ and $t = n\tau$, respectively. The leap-frog time structure allows the variables at each time level to be evaluated explicitly from known values from previous time levels. Spatial derivatives are approximated according to Eqs. (5), and unit normals according to Eq. (3).

Computations with the grid shown in Fig. 1 were found to be stable for $\tau \leqslant 2.1659\ \Delta(gD)^{-1/2}$, where τ is the size of the time-step and Δ is the smallest distance between neighboring grid points. This is greater than the limit for a planar grid of equilateral triangles, $\tau \leqslant 1.7044\ \Delta(gD)^{-1/2}$ [11]. Since a corresponding leap-frog scheme with centered differences on a planar rectangular grid requires $\tau \leqslant 1.4142\ \Delta(gD)^{-1/2}$, the size of the time-step used here can be considered to be quite large. It is interesting to note that, for a planar grid, the leap-frog finite-difference scheme with averaged Coriolis term is computationally neutral, so long as the size of the time-step is less than the stability limit; there is no numerical damping.

The smallest grid intervals are those associated with the points which correspond

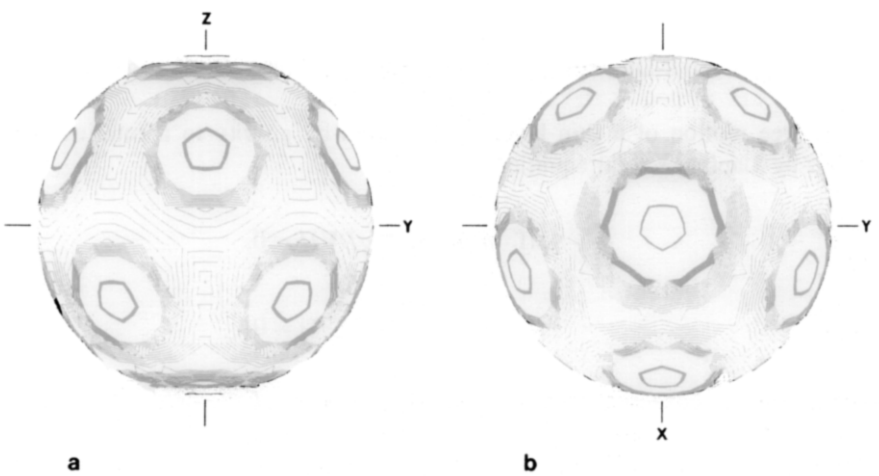

FIG. 2. Contours of constant surface elevation indicate that, for a time-step exceeding the stability limit by only 3 %, energy is radiated from the regions corresponding to the smallest grid spacing. The views (a) and (b) in this and the following figures are the same as defined for Fig. 1.

to the corners of the icosohedron and which have only five neighbors. Fig. 2 illustrates that a time-step about 3 % larger than the stability limit causes these points to radiate energy, which confirms the notion that the stability limit is set by the smallest grid interval.

Equations (8) were used to simulate five different normal-mode oscillations, three without rotation ($\mathbf{f}=0$) and two with rotation about the z-axis ($\mathbf{f}=f\hat{z}$). Exact solu-

[redacted]

to the orientation of the grid as shown in Fig. 1. Additional computations indicated that the orientation of the grid is unimportant.

Case 1. For $\mathbf{f}=0$, normal-mode solutions for surface elevations are the familiar spherical harmonic functions [4]. The first example is the mode with lowest frequency, $\omega=(2gD)^{1/2}/R$, that is azimuthly symmetric about the z-axis with latitudinal variation corresponding to the associated Legendre polynomial, $P_1^0(z/R)$,

$$\begin{aligned} H &= H_0\left(\frac{z}{R}\right)\cos\omega t,\\ U &= H_0\left(\frac{gD}{R\omega}\right)\left(\frac{xz}{R^2}\right)\sin\omega t,\\ V &= H_0\left(\frac{gD}{R\omega}\right)\left(\frac{yz}{R^2}\right)\sin\omega t,\\ W &= -H_0\left(\frac{gD}{R\omega}\right)\left(\frac{R^2-z^2}{R^2}\right)\sin\omega t, \end{aligned} \tag{9}$$

where U, V, and W are the x-, y-, and z-components of $\mathbf{U}$, respectively, and where the constant, H_0, determines the amplitude of the oscillation. The contours of constant surface elevation after 100 cycles of simulation, which are shown in Fig. 3, indicate excellent agreement with the exact solution.

Case 2. The second example, also with $\mathbf{f}=0$ corresponds to a zonally propagating wave with frequency, $\omega=(6gD)^{1/2}/R$, with latitudinal variation corresponding to the associated Legendre polynomial, $P_2^1(z/R)$,

$$\begin{aligned} H &= H_0\left(\frac{z}{R}\right)\left(\frac{x}{R}\cos\omega t+\frac{y}{R}\sin\omega t\right),\\ U &= -H_0\left(\frac{gD}{R\omega}\right)\left(\frac{z}{R}\right)\left(\frac{2xy}{R^2}\cos\omega t+\frac{R^2-2x^2}{R^2}\sin\omega t\right),\\ V &= H_0\left(\frac{gD}{R\omega}\right)\left(\frac{z}{R}\right)\left(\frac{R^2-2y^2}{R^2}\cos\omega t-\frac{2xy}{R^2}\sin\omega t\right),\\ W &= H_0\left(\frac{gD}{R\omega}\right)\left(\frac{R^2-2z^2}{R^2}\right)\left(\frac{y}{R}\cos\omega t-\frac{x}{R}\sin\omega t\right). \end{aligned} \tag{10}$$

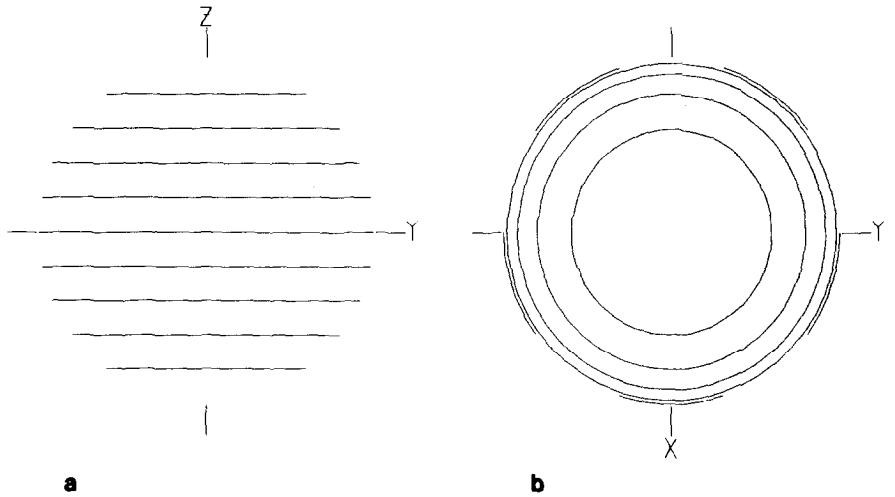

FIG. 3. Case 1, $f=0$, $\omega=(2gD)^{1/2}/R$. The equally spaced contours of constant surface elevation show that the shape of this standing wave is preserved after 100 cycles of simulation.

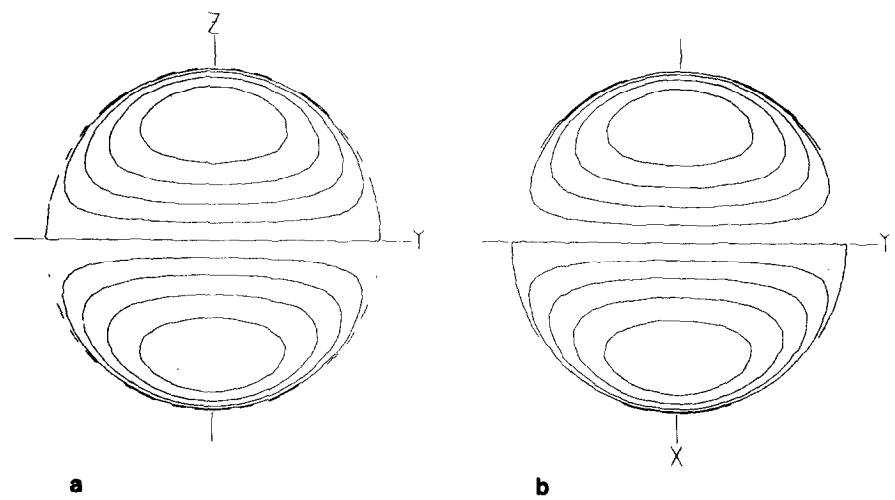

FIG. 4. Case 2, $f=0$, $\omega=(6gD)^{1/2}/R$. After 150 cycles this traveling wave has maintained its shape.

Again, surface elevation contours after 150 cycles (Fig. 4) indicate excellent agreement with the exact solution.

Case 3. A zonally symmetric solution can also be found for the special case, $\omega = f = \pi(gD)^{1/2}/2R$,

$$
\begin{aligned}
H &= H_0 \sin\frac{\pi z}{2R}\cos\omega t,\\
U &= -H_0(gD)^{1/2}\cos\frac{\pi z}{2R}\left(\frac{yz}{R^2-z^2}\cos\omega t - \frac{xz}{R^2-z^2}\sin\omega t\right),\\
V &= H_0(gD)^{1/2}\cos\frac{\pi z}{2R}\left(\frac{xz}{R^2-z^2}\cos\omega t + \frac{yz}{R^2-z^2}\sin\omega t\right),\\
W &= -H_0(gD)^{1/2}\cos\frac{\pi z}{2R}\sin\omega t.
\end{aligned}
\tag{11}
$$

This solution corresponds to Solberg's K_2 zonal oscillations [14]. Although the contours of surface elevation after 100 cycles of simulation (Fig. 5) deviate from parallels of latitude due to truncation error, the solutions are quite good.

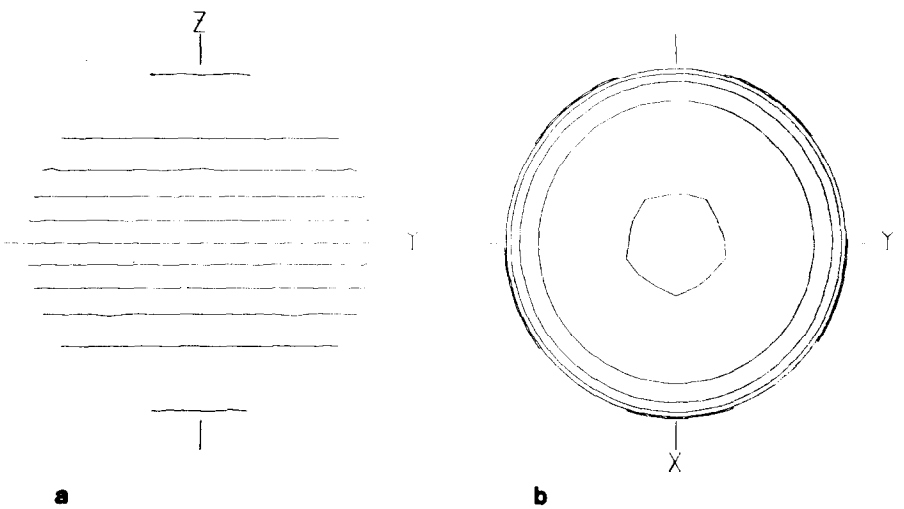

FIG. 5. Case 3, $f = \omega = \pi(gD)^{1/2}/2R$. The deviations of surface elevation from azimuthal symmetry about the z-axis indicate the presence of truncation errors which have accumulated after 100 cycles of simulation (50 revolutions of the sphere about its axis).

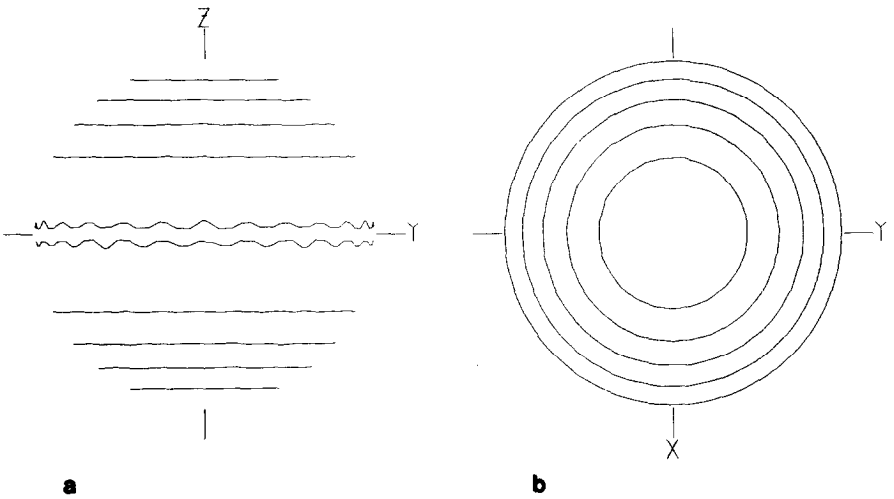

FIG. 6. Case 4, $f=(gD)^{1/2}/R$, $\omega=0$. Again, departure from symmetry is an indication of truncation errors due to small differences of the discrete normal modes from their exact counterparts. These contours correspond to the shape of the surface after approximately 36 revolutions of the sphere.

Case 4. If the motion is steady, $\omega=0$, azimuthly symmetric solutions can be found. This one corresponds to the case of a rigid rotation,

$$H=H_0\left(\frac{x^2+y^2}{R^2}\right),$$
$$U=-2H_0\left(\frac{gD}{fR}\right)\frac{y}{R}, \tag{12}$$
$$V=2H_0\left(\frac{gD}{fR}\right)\frac{x}{R},$$
$$W=0.$$

This solution corresponds to geostrophic flow. (The constant, $8\pi H_0/3$, can be subtracted from H so that the global mean of H is zero.) Contours of surface elevation after 2500 time-steps for the case $f=(gD)^{1/2}/R$ (Fig. 6) also show the presence of truncation errors through their deviations from latitudinal parallels. When the computations were repeated for other values of f, the pattern of truncation errors was different, but in every case the dominant part of the solution was quite good.

Case 5. It is interesting to ask how well this computational method works for modes with spatial structure that is barely resolvable by the computational grid. For example consider the $\mathbf{f}=0$ mode with latitudinal variation corresponding to $P_5^5(z/R)$,

$$
\begin{aligned}
H &= H_0\left[\left(\frac{x^4-10x^2y^2+5y^4}{R^4}\right)\frac{x}{R}\cos\omega t+\left(\frac{5x^4-10x^2y^2+y^4}{R^4}\right)\frac{y}{R}\sin\omega t\right],\\
U &= H_0\left(\frac{5gD}{6}\right)^{1/2}\left\{\left[4\left(\frac{x^2-y^2}{R^2}\right)-\left(\frac{5x^4-10x^2y^2+y^4}{R^4}\right)\right]\frac{xy}{R^2}\cos\omega t\right.\\
&\quad\left.-\left[\left(\frac{x^4-6x^2y^2+y^4}{R^4}\right)-\left(\frac{x^4-10x^2y^2+5y^4}{R^4}\right)\frac{x^2}{R^2}\right]\sin\omega t\right\},\\
V &= H_0\left(\frac{5gD}{6}\right)^{1/2}\left\{\left[\left(\frac{x^4-6x^2y^2+y^4}{R^4}\right)-\left(\frac{5x^4-10x^2y^2+y^4}{R^4}\right)\frac{y^2}{R^2}\right]\cos\omega t\right.\\
&\quad\left.+\left[4\left(\frac{x^2-y^2}{R^2}\right)+\left(\frac{x^4-10x^2y^2+5y^4}{R^4}\right)\right]\frac{xy}{R^2}\sin\omega t\right\},\\
W &= H_0\left(\frac{5gD}{6}\right)^{1/2}\left\{\left(\frac{5x^4-10x^2y^2+y^4}{R^4}\right)\frac{y}{R}\cos\omega t\right.\\
&\quad\left.-\left(\frac{x^4-10x^2y^2+5y^4}{R^4}\right)\frac{x}{R}\sin\omega t\right\}.
\end{aligned}
\tag{13}
$$

This is a zonally propagating wave with frequency, $\omega=(30gD)^{1/2}/R$, which has 5 meridional maxima and minima and 10 meridional nodes. Fig. 7 gives surface elevation contours at $t=0$ and shows that the nodal lines are poorly resolved in the vicinity of the poles, which is due to the fact that the polar grid points have only five nearest neighbors. However, the amplitude of this normal mode diminishes rapidly

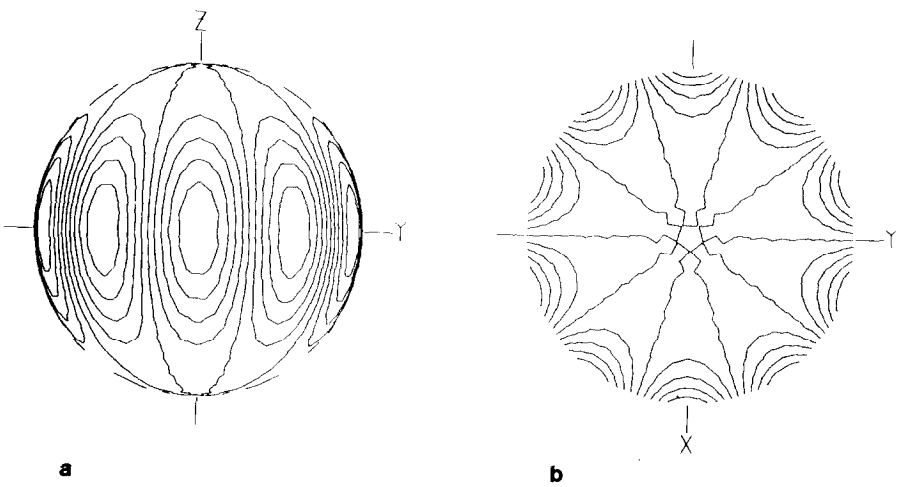

FIG. 7. Case 5, $f=0$, $\omega=(30gD)^{1/2}/R$. Contours of the initial surface elevation field indicate that the grid cannot represent the 10 meridional nodal lines in the vicinity of the poles.

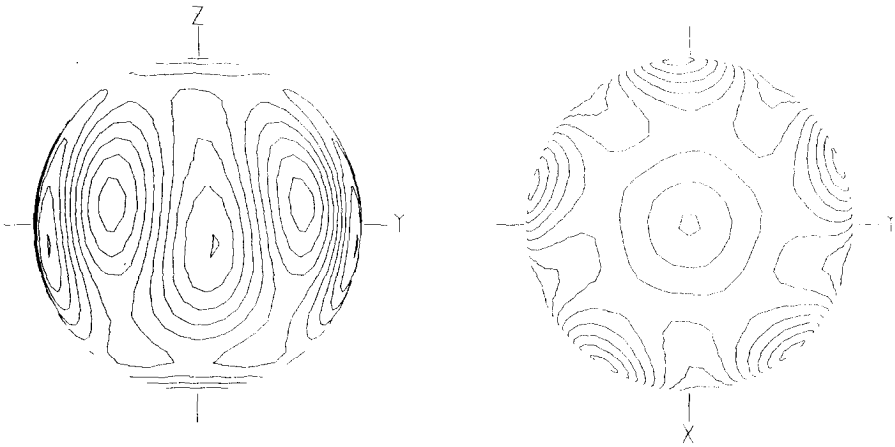

FIG. 8. Case 5. The shape of the normal mode is fairly well preserved after 250 cycles of simulation.

toward the poles, so the poor resolution of the nodal lines in the polar region should not be expected to cause numerical difficulties. Indeed, the result of simulating 250 cycles of oscillation (Fig. 8) is that the mode shape is preserved remarkably well.

Another way to assess the accuracy of the computations is to compare the frequency of the computed oscillations with the frequency of the corresponding exact solutions. The computational frequency can be defined as $\omega_c = 2\pi P/n\tau$, where n is the number of steps of length, τ, required to simulate P periods of the oscillations. The computational and exact frequencies, which are compared in Table II, show excellent agreement.

TABLE II

Comparison of Computed and Exact Frequencies for $\tau = 0.18\, R/(gD)^{1/2}$

	Case			
	1	2	3	5
Number of periods simulated	100	150	100	250
Number of time-steps required	2471	2143	2226	1619
Computed frequency, $\omega_c R/(gD)^{1/2}$	1.4127	2.4433	1.5681	5.3901
Exact frequency, $\omega R/(gD)^{1/2}$	1.4142	2.4494	1.5708	5.4772

V. Conclusions

The computational tests clearly demonstrate that this geodesic finite-difference method is a practical way to approximate the solutions of partial-differential equations on curved domains. Even though the tests involved a spherical domain, the method is applicable to any irregularly curved domain. This is due to the fact that the finite-difference formulas involve three-dimensional Cartesian coordinates rather than two-dimensional curvilinear surface coordinates.

An important feature of this method is that it allows the grid to be tailored to the problem for which it is to be used. The only constraints on the grid are that the points should be distributed densely enough to resolve both the curvature of the computational domain and the curvature of the unknown function and that each point should be centered with respect to its neighbors. This feature can be useful for problems of oceanic circulation, since the grid points can be distributed on a spherical surface according to the bathymetry of the ocean basin with boundary points providing a smooth representation of continental coastlines. The construction of such a grid should not be difficult, because the scheme for automatically generating storm surge grids [13] can be adapted to provide grids for curved domains.

Laplace's tidal equations, which were used here to test the computational method, contain the essential linear dynamics of global atmospheric circulation. This suggests that this method might be particularly useful for predicting weather on a global scale. The advantages of a large explicit time-step and a quasi-uniform computational mesh should allow it to be economically competitive with other finite-difference methods, with finite-element methods, and with spectral methods based upon expansions in spherical harmonic functions. Comparison of the computational efficiency of this method with that of other methods, in particular for cases in which the non-linear advective terms are important, are to be the subject of a future paper.

References

1. M. J. P. Cullen, *Quart. J. Roy. Met. Soc.* **100** (1974), 555.
2. D. E. Hinsman, M.S. thesis, Naval Postgraduate School, 1975.
3. Y. Kurihara, *Mon. Wea. Rev.* **93** (1965), 399.
4. H. Lamb, "Hydrodynamics," Section 199, Dover, New York, 1945.
5. M. S. Longuet-Higgins, *Phil. Trans. Roy. Soc. Lond. A* **262** (1968), 511.
6. W. F. Noh, *Methods Comput. Phys.* **3** (1964), 117.
7. G. W. Platzman, *J. Phys. Oceanogr.* **8** (1978), 323.
8. R. Sadourny, *Mon. Wea. Rev.* **100** (1972), 136.
9. R. Sadourny, A. Arakawa, and Y. Mintz, *Mon. Wea. Rev.* **96** (1968), 351.
10. W. C. Thacker, *J. Phys. Oceanogr.* **7** (1977), 284.
11. W. C. Thacker, *J. Phys. Oceanogr.* **8** (1978), 680.
12. W. C. Thacker, "Marine Forecasting" (J. C. J. Nihoul, Ed.), p. 261, Elsevier, Amsterdam, 1979.
13. W. C. Thacker, A. Gonzalez, and G. E. Putland, *J. Comput. Phys.* **37** (1980), 371.

14. M. V. WILKES, "Oscillations of the Earth's Atmosphere," p. 41, Cambridge Univ. Press, Cambridge Mass., 1949.
15. D. L. WILLIAMSON, *Tellus* **20** (1968), 642.
16. D. L. WILLIAMSON, *Mon. Wea. Rev.* **98** (1970), 512.