

Logically rectangular finite volume methods with adaptive refinement on the sphere

Marsha J. Berger, Donna A. Calhoun, Christiane Helzel and Randall J. LeVeque

Phil. Trans. R. Soc. A 2009 **367**, 4483-4496

doi: 10.1098/rsta.2009.0168

References

[This article cites 15 articles](#)

<http://rsta.royalsocietypublishing.org/content/367/1907/4483.full.html#ref-list-1>

Rapid response

[Respond to this article](#)

<http://rsta.royalsocietypublishing.org/letters/submit/roypta;367/1907/4483>

Subject collections

Articles on similar topics can be found in the following collections

[computational mathematics](#) (21 articles)

Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click [here](#)

To subscribe to *Phil. Trans. R. Soc. A* go to:

<http://rsta.royalsocietypublishing.org/subscriptions>

Logically rectangular finite volume methods with adaptive refinement on the sphere

BY MARSHA J. BERGER¹, DONNA A. CALHOUN², CHRISTIANE HELZEL³
AND RANDALL J. LEVEQUE^{4,*}

¹*Courant Institute of Mathematical Sciences, New York University,
251 Mercer Street, New York, NY 10012, USA*

²*Laboratoire d'Etudes des Transferts et de Mécanique des Fluides,
DEN/DM2S/SFME/LTMF, Commissariat à l'Energie Atomique (CEA),
Centre de Saclay, 91191 Gif-sur-Yvette Cedex, France*

³*Fakultät für Mathematik, Universitätsstrasse 150, Ruhr-Universität Bochum,
44780 Bochum, Germany*

⁴*Department of Applied Mathematics, Box 352420, University of Washington,
Seattle, WA 98195-2420, USA*

The logically rectangular finite volume grids for two-dimensional partial differential equations on a sphere and for three-dimensional problems in a spherical shell introduced recently have nearly uniform cell size, avoiding severe Courant number restrictions. We present recent results with adaptive mesh refinement using the GEOCLAW software and demonstrate well-balanced methods that exactly maintain equilibrium solutions, such as shallow water equations for an ocean at rest over arbitrary bathymetry.

Keywords: shallow water equations; sphere; finite volume; adaptive mesh refinement; well-balanced schemes; bathymetry

1. Introduction

Recently, Calhoun *et al.* (2008*b*) introduced a class of logically rectangular grids (quadrilateral in two dimensions or hexahedral in three dimensions) that can be used for solving partial differential equations on a sphere, illustrated in [figure 1a](#). These grids have the virtue that the cell sizes are nearly uniform (unlike latitude–longitude grids, for example), while maintaining the simple (i, j, k) indexing of logically rectangular grids. On the sphere the ratio of largest to smallest grid cell area is less than 1.7. A potential disadvantage is that the four corners of the computational rectangle are mapped to quadrilaterals that are nearly triangular; the grid lines are far from orthogonal in these regions. We have demonstrated, however, that appropriate finite volume methods can be effectively used on these

*Author for correspondence (rjl@u.washington.edu).

One contribution of 10 to a Theme Issue ‘Mesh generation and mesh adaptation for large-scale Earth-system modelling’.

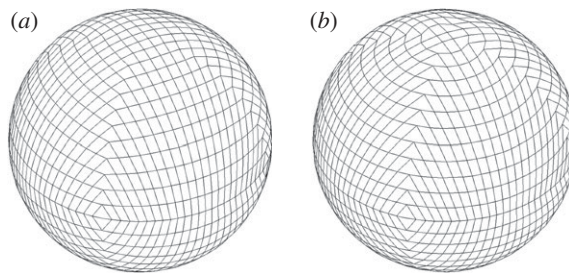


Figure 1. 50×25 quadrilateral grids on the surface of a sphere.

grids that achieve second-order accuracy on smooth solutions and that also capture shock waves and other discontinuities sharply (Calhoun *et al.* 2008*a,b*; Calhoun & Helzel *in press*). In particular, we have presented results for the shallow water equations on the sphere for both shock waves and the Rossby–Haurwitz test problem of Williamson *et al.* (1992) on a single grid as well as some preliminary results with adaptive mesh refinement (AMR).

In this paper we present additional results for shallow water on the sphere and introduce a finite volume method for flow over arbitrary topography or bathymetry on the sphere. We show that the method is ‘well balanced’ for the case of an ocean at rest, meaning that it preserves the stationary steady state in this case and also accurately computes small amplitude wave propagation on top of this equilibrium solution.

The test problems are implemented in GEOCLAW, a subset of the CLAWPACK software that has been developed for geophysical flows in work with David George (George & LeVeque 2006, 2008; LeVeque & George 2007; George 2008), initially in the context of tsunami modelling in the mid-latitudes on a latitude–longitude grid. The complete source code for all the examples presented here will be made available on the Web (www.clawpack.org/links/sphere).

The AMR code in GEOCLAW is based on AMRCLAW, originally developed by Berger & LeVeque (1998). Refinement is done on logically rectangular patches in the computational domain, an approach that has been widely used, including for three-dimensional flow on the sphere in the recent work of Hubbard & Nikiforakis (2003), for example. The grid mappings we use here have the advantage that the computational domain is a single rectangle, greatly simplifying the application of adaptive refinement.

2. The grid mappings

Figure 2 illustrates the mapping from the computational domain (figure 2*a*) to the quadrilateral grid we use on the sphere (figure 2*d*).

Details of this mapping are presented in Calhoun *et al.* (2008*b*) and a Python implementation is displayed in figure 3. Here $(\mathbf{x}\mathbf{c}, \mathbf{y}\mathbf{c})$ is a point in the computational domain $[-3, 1] \times [-1, 1]$ and $(\mathbf{x}\mathbf{p}, \mathbf{y}\mathbf{p}, \mathbf{z}\mathbf{p})$ is the image on the sphere. The square $[-3, -1] \times [-1, 1]$ is mapped to the lower hemisphere, while $[-1, 1] \times [-1, 1]$ is mapped to the upper hemisphere.

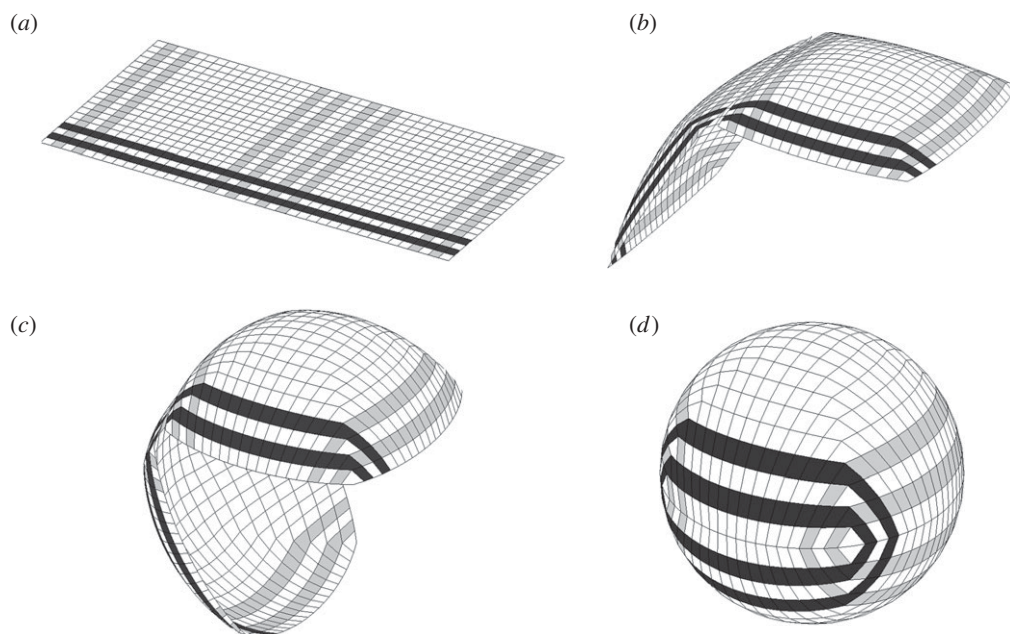


Figure 2. (a) The 40×20 rectangular grid in the computational domain. (b,c) Some grid lines are shaded to help visualize the mapping. (d) The final quadrilateral grid on the sphere is shown.

Note one modification from the mapping presented in Calhoun *et al.* (2008b): the expression for D used here involves $\sin(\pi d/2)$ rather than $d(2-d)$ as in the original mapping. This leads to slightly more uniform cells and is more natural since equally spaced points along a diagonal are then mapped to points that are equally spaced in latitude in the image. In practice it makes little difference in computed results. As written here, this mapping produces the grid shown in figure 1a.

We also mention a modified version of this mapping, which can be obtained by un-commenting the line `R = Rsphere * sin(pi*d/2)`. With this choice the grid shown in figure 1b is obtained. This grid has the possible advantage that one set of grid lines are lines of constant latitude. (The latitude lines are concentric squares in the computational domain.) However, this grid is not as smooth near the poles as the grid shown in figure 1a, and in practice has not been found to be advantageous, even for the test problems of §7 where the true solution is axisymmetric (results not shown here). The grid may prove useful for specialized applications however, such as purely zonal advection.

The quadrilateral grid we use on the sphere is easily extended to a three-dimensional hexahedral grid in the normal direction. The computational domain is again purely Cartesian $[-3, 1] \times [-1, 1] \times [R_1, R_2]$ for a shell with inner radius R_1 and outer radius R_2 . We are currently working on the development of a well-balanced AMR method for solving atmospheric flow problems on this grid, for situations in which the flow is a small perturbation of an equilibrium with a hydrostatic pressure gradient in the normal direction. This work will be presented elsewhere.

```

def mapc2m(xc,yc):
    # Maps computational point (xc,yc) in [-3,1] x [-1,1]
    # to the physical point (xp,yp,zp) on the manifold (sphere).
    from numpy import sqrt, abs, where, sign, ones, pi, sin, maximum

    Rsphere = 6.371e6                # radius of Earth
    sgnz = where(xc<-1, -1, 1)       # lower hemisphere for xc <-1
    xc = where(xc<-1., -2.*xc, xc)   # reflect these points to right side
    absxc = abs(xc)
    absyc = abs(yc)
    d = maximum(absxc, absyc)        # distance to concentric square

    D = Rsphere * sin(pi*d/2)/sqrt(2) # different than in [8] (see text)
    R = Rsphere * ones(d.shape)       # produces Figure 1(a)
    # R = Rsphere * sin(pi*d/2)       # produces Figure 1(b)

    center = D - sqrt(R**2 - D**2)
    xp = where(d>0, D/d, 0.) * absxc  # d = 0 only if absxc=absyc=0
    yp = where(d>0, D/d, 0.) * absyc

    yp = where(absyc>=absxc, center + sqrt(R**2 - xp**2), yp)
    xp = where(absxc>=absyc, center + sqrt(R**2 - yp**2), xp)

    xp = sign(xc) * xp
    yp = sign(yc) * yp
    zp = sgnz * sqrt(Rsphere**2 - (xp**2 + yp**2))
    return xp,yp,zp

```

Figure 3. Python code for the sphere mapping, as described more fully in Calhoun *et al.* (2008*b*; which includes a Matlab version).

3. Shallow water equations on the sphere

Since there is no smooth coordinate system for our grids on the sphere, we keep track of the fluid velocity in terms of a full three-dimensional vector $\vec{u} = (u, v, w)$ in each finite volume grid cell. These define a velocity vector that should be tangent to the sphere. Similar approaches are used, for example, in Swarztrauber *et al.* (1997) and Giraldo *et al.* (2002).

We use the geopotential height $\phi(\vec{x}, t)$ and solve the shallow water equations on the sphere in the form

$$\left. \begin{aligned} \phi_t + \nabla \cdot (\phi \vec{u}) &= 0, \\ (\phi \vec{u})_t + \mathbb{P}(\nabla \cdot \tilde{F}(q)) &= -\frac{2\Omega z\phi}{R^2} (\vec{x} \times \vec{u}) - \phi \mathbb{P}(\nabla_s \bar{\phi}). \end{aligned} \right\} \quad (3.1)$$

Here $q = [\phi, \phi u, \phi v, \phi w]^T$ and $\bar{\phi}$ is the surface geopotential.

$$\tilde{F}(q) = \begin{bmatrix} \phi u^2 + \frac{1}{2}\phi^2 \\ \phi uv \\ \phi uw \end{bmatrix} \vec{i} + \begin{bmatrix} \phi uv \\ \phi v^2 + \frac{1}{2}\phi^2 \\ \phi vw \end{bmatrix} \vec{j} + \begin{bmatrix} \phi uw \\ \phi vw \\ \phi w^2 + \frac{1}{2}\phi^2 \end{bmatrix} \vec{k} \quad (3.2)$$

is the momentum flux. The symbol \mathbb{P} denotes the projection matrix that projects a vector at \vec{x} to the tangent plane,

$$\mathbb{P} = I - \frac{\vec{x}\vec{x}^T}{R^2}. \quad (3.3)$$

The divergence of the flux and the last term on the right in equation (3.1), which appears for flow over variable topography specified by a surface geopotential $\phi(\vec{x})$, are projected to the tangent plane. The Coriolis term involving the rotation rate Ω has already been projected to the tangent plane. Giraldo *et al.* (2002) provide justification that these equations are equivalent to standard formulations of shallow water on the sphere.

Note that for shallow incompressible water of depth h on the sphere the geopotential is $\phi = gh$, where g varies with latitude to incorporate the centrifugal force on a rotating sphere.

4. Topography and well-balanced methods

In applications such as tsunami modelling, it is very important that the numerical method maintains the equilibrium solution consisting of an ocean at rest, for which the depth h and the bathymetry B both vary greatly, but the surface elevation $\eta \equiv h + B$ is constant and the velocities are identically 0. Numerical methods often do not maintain this steady state unless care is taken, since the divergence of the flux and the source term are both large but cancel out. In particular, fractional step methods in which one alternates between solving the homogeneous conservation laws and the source term equations do not work well at all (e.g. LeVeque 1998). In terms of the geopotential ϕ , the ocean at rest equilibrium solution has $\phi + gB$ constant (where again g varies with latitude if the sphere is rotating).

The high-resolution wave-propagation method that we present in §6 exactly preserves such an equilibrium solution and also accurately captures small amplitude waves on top of such a steady state. Much research has recently been devoted to developing such ‘well-balanced’ methods, particularly for shallow water equations (e.g. Gosse 2000, 2001; Audusse *et al.* 2004; Bouchut 2004; Castro *et al.* 2008).

5. Boundary conditions

For finite volume methods of the type we use, boundary conditions at the edges of the computational rectangle are generally implemented using the standard procedure of introducing two rows of ‘ghost cells’ at each edge of the computational domain. Values in these cells are set at the start of each time step by copying data from elsewhere in the domain, since each edge of the domain is adjacent to another edge after performing the mapping illustrated in figure 2. After setting the ghost cells in this manner, the wave-propagation algorithm is applied at all points in the computational domain. We fill two rows of ghost cells because this method has a stencil that extends two grid cells out in each direction (the adjacent cell value is used in solving the

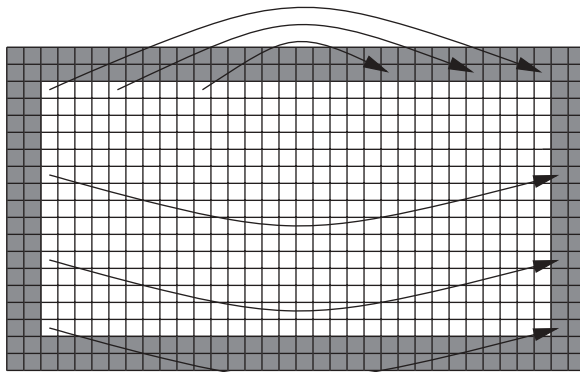


Figure 4. The grey ghost cells receive data from interior cells as indicated.

Riemann problem at the cell interface and the value in the next proximate grid cell is required in applying limiters to the resulting waves for the high-resolution method).

Let (x_c, y_c) represent a point in the computational domain. After the mapping, the boundaries $x_c = -3$ and $x_c = +1$ meet up along the equator and these boundaries simply have standard periodic boundary conditions:

$$\left. \begin{aligned} q(-3 - \epsilon, y_c) &= q(1 - \epsilon, y_c), \\ q(1 + \epsilon, y_c) &= q(-3 + \epsilon, y_c), \end{aligned} \right\} \quad (5.1)$$

for $-1 \leq y_c \leq 1$. At the boundary $\epsilon = 0$, but to fill the ghost cells these expressions are used for $\epsilon > 0$. These are easily implemented on a $2m \times m$ grid by setting

$$\left. \begin{aligned} Q_{i,j} &= Q_{2m+i,j}, & \text{for } i = 0, -1, \\ Q_{2m+i,j} &= Q_{i,j}, & \text{for } i = 1, 2, \end{aligned} \right\} \quad (5.2)$$

for $j = 1, 2, \dots, m$ at the start of each time step.

Along $y_c = \pm 1$ the boundary conditions are only slightly more complicated. At each $y_c = \pm 1$ boundary, the segment $-3 \leq x_c \leq -1$ matches up with the segment $-1 \leq x_c \leq 1$ with the orientation reversed, as seen in figures 2 and 4, and hence

$$\left. \begin{aligned} q(x_c, 1 + \epsilon) &= q(-2 - x_c, 1 - \epsilon), \\ q(x_c, -1 - \epsilon) &= q(-2 - x_c, -1 + \epsilon), \end{aligned} \right\} \quad (5.3)$$

for $-3 \leq x_c \leq 1$. This is implemented by copying data according to

$$\left. \begin{aligned} Q_{i,j} &= Q_{2m+1-i,1-j}, & \text{for } j = 0, -1, \\ Q_{i,m+j} &= Q_{2m+1-i,m-1-j}, & \text{for } j = 1, 2, \end{aligned} \right\} \quad (5.4)$$

for $i = 1, 2, \dots, 2m$.

These boundary conditions were implemented for single-grid calculations in the example presented in Calhoun *et al.* (2008b) and Fortran code is available at the webpage for that paper. These boundary conditions were recently implemented in the AMR version of GEOCLAW as one possible choice of boundary conditions, facilitating the use of AMR for flow problems on the sphere. With AMR, the

non-standard boundary conditions (5.3) require some work to implement, since a grid that is adjacent to the top or bottom boundary will generally have to fill ghost cell values from a different grid at the same level or by interpolating from coarser grids, and logic must be incorporated to search for the relevant data at the appropriate interior point as defined by equation (5.3). Similar issues arise for standard periodic boundary conditions, which were implemented in Berger's original AMR code that formed the basis for the AMR routines in CLAWPACK.

Some preliminary results using AMR on the sphere were presented in Calhoun *et al.* (2008a), but with a different approach to specifying boundary conditions. We mention this alternative since it may be useful to those wishing to apply other AMR software to the sphere using our grid mapping. The computational domain can be doubled in size to a full square $[-3, 1] \times [-3, 1]$ by defining

$$q(x_c, y_c, t) = q(-2 - x_c, -2 - y_c, t), \quad (5.5)$$

for $-3 \leq x_c \leq 1$. This simply reflects all the data through the point $(-1, -1)$. The mapping can be extended in an obvious way so that the new computational rectangle $[-3, 1] \times [-3, -1]$ is mapped to a second copy of the sphere and each copy will contain the same value of q at each point on the sphere. The advantage of introducing this second copy is that the boundary conditions at $y_c = -3$ and $y_c = 1$ are now the standard periodic boundary conditions in this direction,

$$\left. \begin{aligned} q(x_c, -3 - \epsilon) &= q(x_c, 1 - \epsilon), \\ q(x_c, 1 + \epsilon) &= q(x_c, -3 + \epsilon), \end{aligned} \right\} \quad (5.6)$$

for $-3 \leq x_c \leq 1$. Hence, any AMR code that can handle periodic boundary conditions on a rectangular domain can be applied on the sphere, at double the computer time but without any need to reprogram the AMR boundary conditions.

6. The finite volume method

The cell average Q_{ij}^n in cell (i, j) at time t_n is updated by waves that propagate into the cell from the neighbouring cell interfaces, which are determined by solving one-dimensional Riemann problems at each interface. The approach we use is the 'f-wave formulation' of the wave-propagation algorithm introduced in Bale *et al.* (2002) and also discussed in ch. 17 of LeVeque (2002). This means that the waves are determined by computing the jump in the flux plus a contribution from the source terms, and then decomposing this vector into eigenvectors of a linearized Jacobian matrix defined at each cell interface, as described in more detail below. These waves are then also used to define correction terms that yield a method that is second-order accurate on smooth solutions. Limiters are applied to these waves (by comparing with waves from adjacent Riemann problems) in order to avoid non-physical oscillations and produce high-resolution results. 'Transverse Riemann problems' must also be solved to determine the portion of waves that move transversely into corner-adjacent cells and this appears to be an important aspect of these algorithms in successfully solving multi-dimensional problems on the highly skewed grids of figure 1. Complete details of this approach can be found in LeVeque (2002).

Here we concentrate on describing the unique aspects of the algorithm for flow over topography on the sphere, which is primarily concerned with the algorithm for solving the one-dimensional Riemann problem at a cell edge to obtain a suitable set of f -waves.

Consider, for example, the interface $(i - 1/2, j)$ between cells $(i - 1, j)$ and (i, j) . Let \tilde{Q}_{ij}^n be the momentum components of the cell averages Q_{ij}^n on the (i, j) grid cell at time t_n , which we suppose to lie in the tangent plane at the cell centre. We will solve a one-dimensional Riemann problem between states

$$Q_l = \begin{bmatrix} \phi_l \\ \phi_l u_l \\ \phi_l v_l \end{bmatrix} \quad \text{and} \quad Q_r = \begin{bmatrix} \phi_r \\ \phi_r u_r \\ \phi_r v_r \end{bmatrix}, \quad (6.1)$$

where $\phi_l = \phi_{i-1,j}^n$ and $\phi_r = \phi_{i,j}^n$. To obtain the velocities we first project the momentum components of the data in the two neighbouring cells, $\tilde{Q}_{i-1,j}^n$ and $\tilde{Q}_{i,j}^n$, onto a common direction. We choose the direction that is normal to the cell edge and that lies in the tangent plane at the midpoint of the cell edge. This defines the normal momentum components of the Riemann problem data, $\phi_l u_l$ and $\phi_r u_r$. We also compute tangential momentum components $\phi_l v_l$ and $\phi_r v_r$ by projecting the cell momenta onto the direction determined by the cell edge.

From these Riemann data we can define an approximate Jacobian matrix, using for example the Roe-averaged Jacobian \hat{A} . The eigenvalues of this matrix define the wave speeds and the eigenvectors are used to determine the f -waves. This is done by decomposing

$$\Delta = \begin{bmatrix} \phi_r u_r \\ \phi_r u_r^2 + \frac{1}{2} \phi_r^2 \\ \phi_r u_r v_r \end{bmatrix} - \begin{bmatrix} \phi_l u_l \\ \phi_l u_l^2 + \frac{1}{2} \phi_l^2 \\ \phi_l u_l v_l \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ (\phi_l + \phi_r)(\bar{\phi}_r - \bar{\phi}_l) \\ 0 \end{bmatrix} \quad (6.2)$$

as a linear combination of the eigenvectors \hat{r}^p of \hat{A} ,

$$\Delta = \sum_{p=1}^3 \beta^p \hat{r}^p = \sum_{p=1}^3 \mathcal{Z}^p. \quad (6.3)$$

The vectors \mathcal{Z}^p are the f -waves used in the wave-propagation algorithm.

The quantity Δ in equation (6.2) is the flux difference modified by the source term from the varying bathymetry. For the special case of an ocean at rest, all terms in equation (6.2) involving velocities drop out and all that remains are the following terms in the second element of Δ : the pressure gradient $\frac{1}{2}(\phi_r^2 - \phi_l^2)$ and the source term $\frac{1}{2}(\phi_l + \phi_r)(\bar{\phi}_r - \bar{\phi}_l)$. For the ocean at rest, the sea surface $\phi + \bar{\phi}$ is constant and so it can be easily verified that these two terms cancel. Hence, Δ is the zero vector in this case, with the result that the waves \mathcal{Z}^p all have zero strength. For an ocean near rest, it is only the deviation from steady state that is split into waves and propagated with our wave-propagation algorithm.

When AMR is used, some care must be exercised in interpolating values between coarse grids and finer level grids, as required for example when initializing a new fine grid in some region or interpolating ghost cell values from coarser levels.

One crucial aspect is to do interpolations based on $\phi + \bar{\phi}$ and then reconstruct ϕ , rather than interpolating ϕ itself. This is discussed in more detail in the context of tsunami modelling in George (2008).

The area of a quadrilateral grid cell on the sphere can exactly be computed as the sum of the area of two spherical triangles. Note that with our grid mapping, refined grid cells do not exactly match up with the underlying coarse grid cell in physical space. Therefore, we calculate the area of every grid cell as the sum of the areas of grid cells on the finest possible level that might be obtained by refining the grid cell. In an analogous way we also calculate the surface geopotential in each grid cell.

7. Numerical results

As a test case for the well-balanced method with adaptive refinement as described in this paper, we have chosen a problem for which the solution varies only with latitude so that a one-dimensional axisymmetric code can be used to generate a fine grid reference solution. A test of this nature was also presented in Calhoun *et al.* (2008*b*), but only on a single grid with no topography.

The one-dimensional reference solution (plotted as a solid line in figure 6) is obtained by approximating the system

$$\left. \begin{aligned} \phi_t + \frac{1}{a}(\phi U)_\theta &= \frac{1}{a}\phi U \tan \theta, \\ (\phi U)_t + \frac{1}{a}\left(\phi U^2 + \frac{1}{2}\phi^2\right)_\theta &= -\frac{1}{a}\phi\bar{\phi}_\theta + \frac{1}{a}\phi(U^2 - V^2)\tan \theta - f\phi V, \\ (\phi V)_t + \frac{1}{a}(\phi UV)_\theta &= \frac{2}{a}\phi UV \tan \theta + f\phi U. \end{aligned} \right\} \quad (7.1)$$

Here $\phi = \phi(\theta, t)$ is the geopotential height, $\bar{\phi} = \bar{\phi}(\theta)$ is the surface geopotential. $U = U(\theta, t)$ and $V = V(\theta, t)$ are velocities in the latitudinal and the longitudinal direction, respectively. Initial value problems of system (7.1) are approximated for latitude $\theta \in [-\pi/2, \pi/2]$. Reflecting boundary conditions are used on both sides of this interval.

For our test problem we specify axisymmetric topography by choosing the surface potential to be the following function of latitude θ :

$$\bar{\phi}(\theta) = -40\,000 + 20\,000 \exp\left(-1000\left(\theta + \frac{\pi}{6}\right)^2\right), \quad (7.2)$$

where θ varies from $-\pi/2$ at the south pole to $\pi/2$ at the north pole. Sea level is taken to be the surface of geopotential 0, so the surface potential (7.2) corresponds to an ocean of depth $40\,000/g \approx 4000$ m with an axisymmetric ridge about the sphere at a latitude of $-\pi/6$ or 30° south. At the peak of the ridge the ocean depth is reduced by half. The ocean at rest corresponds to $\phi(\theta, t) \equiv -\bar{\phi}(\theta)$ so that $\phi(\theta, t) + \bar{\phi}(\theta) \equiv 0$.

As initial conditions we perturb this to

$$\phi(\theta, t=0) = -\bar{\phi}(\theta) + A \exp\left(-1000\left(\theta - \frac{\pi}{6}\right)^2\right), \quad (7.3)$$

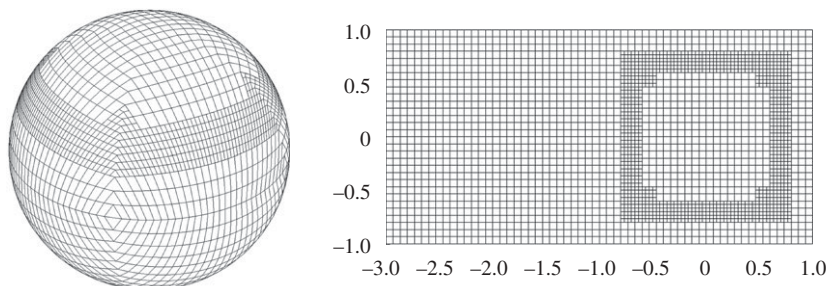


Figure 5. A coarse grid with one level of refined patches with a refinement factor 2 in each direction. Shown on the grid and in the computational rectangle, where refinement is on rectangular patches.

corresponding to an axisymmetric elevation above sea level in a ring of latitude $\pi/6$. Note the decay factor -1000 in the Gaussian, giving a width of roughly 200 km on the Earth. This is a reasonable wave length for a tsunami wave.

We initialize the zonal velocity V to $V(\theta, t=0) = 0$ and the meridional momentum ϕU to

$$(\phi U)(\theta, t=0) = -200A \exp\left(-1000\left(\theta - \frac{\pi}{6}\right)^2\right). \quad (7.4)$$

Here $200 = \sqrt{40\,000}$ is the approximate wave speed and so the vector $(\phi, \phi U)$ is approximately a 1-wave eigenvector, which results in most of the initial perturbation moving southwards, with a much weaker wave moving north. In the two-dimensional code on the sphere we must initialize the three-dimensional momentum components of the solution vector $(\phi u, \phi v, \phi w)$ to agree with this in each grid cell by

$$\left. \begin{aligned} (\phi u)(x, y, z, t=0) &= \frac{(\phi U)(\theta)xz}{R\sqrt{x^2 + y^2}}, \\ (\phi v)(x, y, z, t=0) &= \frac{(\phi U)(\theta)yz}{R\sqrt{x^2 + y^2}}, \\ (\phi w)(x, y, z, t=0) &= -\frac{(\phi U)(\theta)(x^2 + y^2)}{R\sqrt{x^2 + y^2}}. \end{aligned} \right\} \quad (7.5)$$

We take $A = 10$ as the amplitude of the perturbation in the initial data. This gives $A/g \approx 1$ m, which is realistic for tsunami modelling and tests the ability of our approach to handle a wave that has very small amplitude relative to the variations in topography (about 0.05% in this case).

We include Coriolis effects and take $\Omega = 7.292 \times 10^{-5}$, the rotation rate of the Earth. For the radius of the Earth, we use $R = 6.371 \times 10^6$. In both the one- and two-dimensional code we compute latitude relative to a perfect sphere, though this could be replaced by an appropriate geoid.

Figure 5 shows how the AMR refinement might look at time $t = 0$ in the case of very coarse grids.

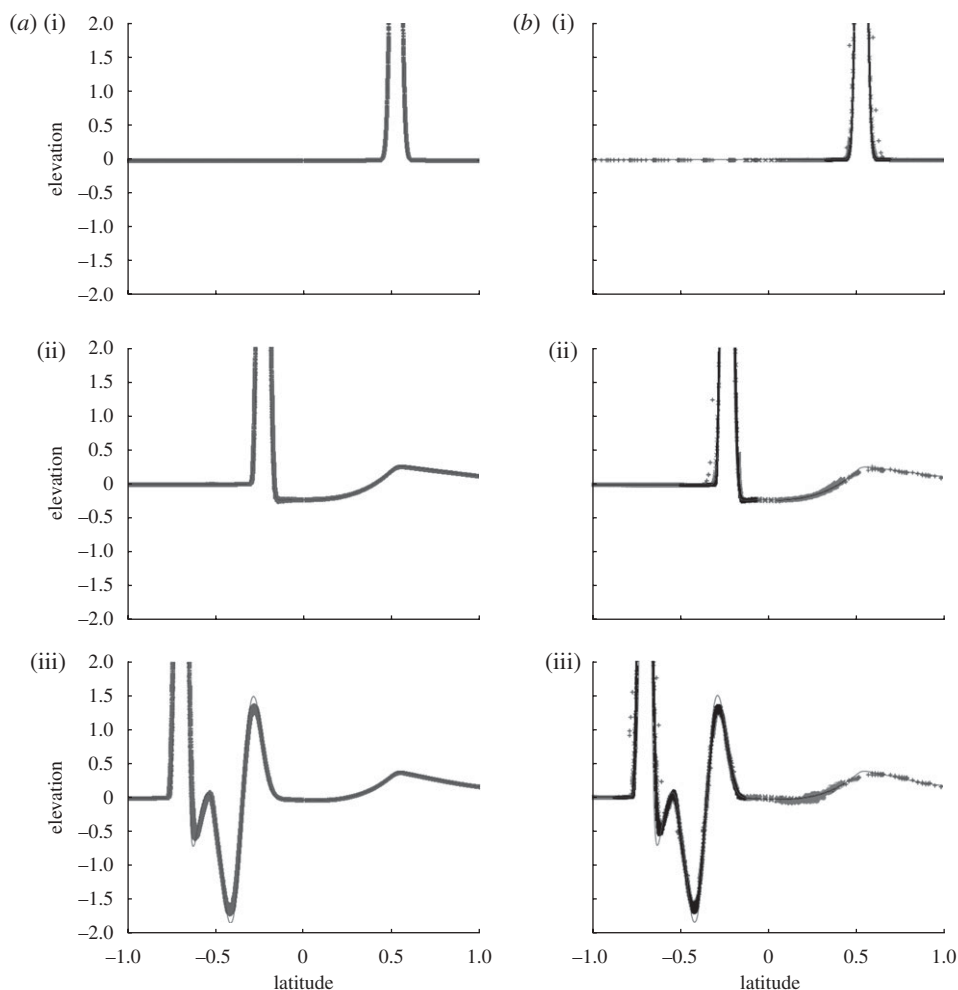


Figure 6. Zoomed-in view of the computed solutions (geopotential elevation above sea level) to the axisymmetric test cases described in §7. In each figure (scatter plots of surface elevation at time (i) 0, (ii) 24 000 and (iii) 40 000) the solid line was computed with a one-dimensional code using 5000 grid cells. The symbols show the solution on the sphere with the value of $\phi_{ij} - \bar{\phi}(\theta)$ in each finite volume cell plotted against the latitude θ_{ij} of the centre of the cell. (a) The solution on a single 800×400 computational grid. (b) The solution when the AMR algorithm is used. In this figure the plus symbols show values from the coarsest level 1 grids, crosses show values from level 2 grids, and circles are values from level 3 grids.

Results of this test are shown in figure 6. Figure 6*a* shows scatter plots of the solution when a single 800×400 grid is used on the sphere, at three different times. Figure 6*b* shows results where refinement is done near the main wave with the following refinement factors:

- level 1: 50×25 grid,
- level 2: refined by 4 (effective resolution 200×100), and
- level 3: refined by 4 (effective resolution 800×400).

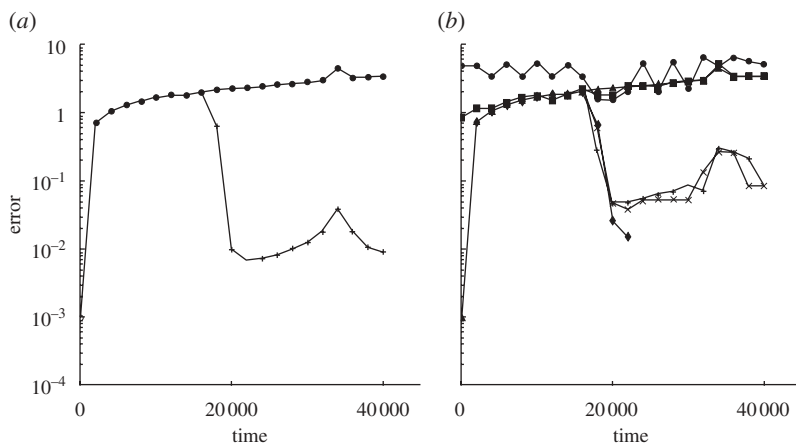


Figure 7. Max-norm errors for the two calculations shown in figure 6, as a function of time. The largest error occurs near the peak of the Gaussian. Also shown is the error only over grid points that lie in the upper hemisphere. After time $t \approx 20\,000$ the peak has moved into the lower hemisphere and this error decreases. In (b) the errors are calculated separately on each of the three levels of grid refinement. There are no level 3 grids in the upper hemisphere after time 21 000. (a) Filled circles: $mx = 800$, $mxnest = 1$, level = 1; plus symbols: only over upper hemisphere. (b) Filled circles: $mx = 50$, $mxnest = 3$, level = 1; plus symbols: only over upper hemisphere; filled boxes: $mx = 50$, $mxnest = 3$, level = 2; cross symbols: only over upper hemisphere; filled triangles: $mx = 50$, $mxnest = 3$, level = 3; filled diamonds: only over upper hemisphere.

For this particular example, the uniform grid test took roughly 1300 s on a desktop machine, while the AMR calculation required 360 s. For this axisymmetric calculation much of the grid was refined since the hump extends around the sphere. For modelling a more localized region the benefits of AMR would be more pronounced.

The AMR results exhibit comparable accuracy to the uniform fine grid within the region of maximum refinement. This can be seen in the plots of max-norm error versus time in figure 7, which track the error in the Gaussian peak and are essentially the same with or without refinement. The peak is cut off in figure 6 in order to show the accuracy better in regions away from the peak.

In regions where the grid is less refined, such as the upper hemisphere at later times, the AMR calculation gives somewhat reduced accuracy but still quite acceptable given the coarseness of the grid. Note in particular that there is no spurious reflection of wave energy near the edges of the refinement patches.

Also note that in all cases the scatter plot looks as smooth near $\theta = 0$ as elsewhere, indicating that the highly skewed cells on the equator are not causing accuracy problems. Finally, note that no spurious waves are generated from the region near $\theta = -\pi/6 \approx -0.5$ where the large topography feature is located. This test problem is analogous to the one introduced by LeVeque (1998) for one-dimensional shallow water equations, where it is demonstrated that all accuracy is lost if a fractional step procedure is used for the topography source term. The ability to capture this small amplitude wave depends on our use of the f -wave technique.

We have also tested the code with a wave of amplitude $A = 5000$, in which case the Gaussian rapidly steepens into a shock wave. The high-resolution shock capturing algorithms employed in GEOCLAW can handle this case as well. Results analogous to figure 6 have been obtained (not shown here).

8. Conclusions

We have demonstrated that the quadrilateral grids introduced in Calhoun *et al.* (2008*b*) can be successfully used to solve the shallow water equations on the surface of the sphere with adaptive mesh refinement. This has been accomplished using GEOCLAW by enhancing this software package with a new choice of numerical boundary conditions needed for our quadrilateral geometry, as described in §5. This software uses high-resolution shock capturing finite volume methods that accurately capture shock wave propagation. We have also implemented the numerical method using the f -wave formulation described in §4, which ensures that the code is well balanced for an ocean at rest. Waves of small amplitude relative to the variation in the bottom topography are well captured, as illustrated in figure 6. Our quadrilateral grid has nearly constant cell areas, leading to efficient solution with explicit methods having Courant number between 0.5 and 1 in all cells. A potential difficulty is that some cells are highly skewed, but with the multi-dimensional wave propagation algorithms we employ, this does not appear to cause any loss of accuracy.

The computer code used to generate the numerical results and figures presented in this paper, along with links to the GEOCLAW software, can be found at www.clawpack.org/links/sphere.

This work was supported in part by DOE grant DE-FG02-88ER25053, AFOSR grant FA9550-06-1-0203, NSF grant DMS-0106511 and DFG grant HE 4858/1-1.

References

- Audusse, E., Bouchut, F., Bristeau, M. O., Klein, R. & Perthame, B. 2004 A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comput.* **25**, 2050–2065. (doi:10.1137/S1064827503431090)
- Bale, D., LeVeque, R. J., Mitran, S. & Rossmannith, J. A. 2002 A wave-propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM J. Sci. Comput.* **24**, 955–978. (doi:10.1137/S106482750139738X)
- Berger, M. J. & LeVeque, R. J. 1998 Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.* **35**, 2298–2316. (doi:10.1137/S0036142997315974)
- Bouchut, F. 2004 *Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources*. Boston, MA: Birkhäuser.
- Calhoun, D. A. & Helzel, C. In press. A finite-volume method for solving parabolic equations on logically Cartesian curved surface meshes. *SIAM J. Sci. Comput.* See <http://www.amath.washington.edu/~calhoun/Surfaces/>.
- Calhoun, D. A., Helzel, C. & LeVeque, R. J. 2008*a* A finite volume grid for solving hyperbolic problems on the sphere. In *Hyperbolic problems: theory, numerics, applications. Proc. 11th Int. Conf. on Hyperbolic Problems, Lyon, France, 17–21 July 2006*, pp. 355–362. Berlin, Germany: Springer.
- Calhoun, D. A., Helzel, C. & LeVeque, R. J. 2008*b* Logically rectangular finite volume grids and methods for ‘circular’ and ‘spherical’ domains. *SIAM Rev.* **50**, 723–752. (doi:10.1137/060664094)

- Castro, M., Gallardo, J. M., Lpez, J. A. & Parés, C. 2008 Well-balanced high order extensions of Godunov's method for semilinear balance laws. *SIAM J. Numer. Anal.* **46**, 1012–1039. (doi:10.1137/060674879)
- George, D. L. 2008 Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *J. Comput. Phys.* **227**, 3089–3113. (doi:10.1016/j.jcp.2007.10.027)
- George, D. L. & LeVeque, R. J. 2006 Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. *Sci. Tsunami Hazards* **24**, 319–328.
- George, D. L. & LeVeque, R. J. 2008 High resolution methods and adaptive refinement for tsunami propagation and inundation. In *Hyperbolic problems: theory, numerics, applications. Proc. 11th Int. Conf. on Hyperbolic Problems, Lyon, France, 17–21 July 2006*, pp. 541–550. Berlin, Germany: Springer.
- Giraldo, F. X., Hesthaven, J. S. & Warburton, T. 2002 Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *J. Comput. Phys.* **181**, 499–525. (doi:10.1006/jcph.2002.7139)
- Gosse, L. 2000 A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Comput. Math. Appl.* **39**, 135–159. (doi:10.1142/S021820250100088X)
- Gosse, L. 2001 A well-balanced scheme using non-conservative products designed for hyperbolic systems of conservation laws with source terms. *Math. Models Meth. Appl. Sci.* **11**, 339–365. (doi:10.1142/S021820250100088X)
- Hubbard, M. E. & Nikiforakis, N. 2003 A three-dimensional, adaptive, Godunov-type model for global atmospheric flows. *Mon. Wea. Rev.* **131**, 1848–1864. (doi:10.1175//2568.1)
- LeVeque, R. J. 1998 Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *J. Comput. Phys.* **146**, 346–365. (doi:10.1006/jcph.1998.6058)
- LeVeque, R. J. 2002 *Finite volume methods for hyperbolic problems*. Cambridge, UK: Cambridge University Press.
- LeVeque, R. J. & George, D. L. 2007 High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In *Advanced numerical models for simulating tsunami waves and runup* (eds P. L.-F. Liu, H. Yeh & C. Synolakis). Advances in Coastal and Ocean Engineering, vol. 10, pp. 43–73. Singapore: World Scientific.
- Swarztrauber, P. N., Williamson, D. L. & Drake, J. B. 1997 The Cartesian method for solving partial differential equations in spherical geometry. *Dyn. Atmos. Oceans* **27**, 679–706. (doi:10.1016/S0377-0265(97)00038-9)
- Williamson, D. L., Drake, J. B., Hack, J. J., Jakob, R. & Swarztrauber, P. N. 1992 A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.* **102**, 211–224. (doi:10.1016/S0021-9991(05)80016-6)