A Finite Volume Grid for Solving Hyperbolic Problems on the Sphere

Donna Calhoun[1], Christiane Helzel[2], and Randall J. LeVeque[3]

[1] Commissariat á l'Energie Atomique (donna.calhoun@cea.fr)
[2] Institut für Angewandte Mathematik, University of Bonn (helzel@iam.uni-bonn.de)
[3] Department of Applied Mathematics, University of Washington (rjl@amath.washington.edu)

1 Introduction

Uniform Cartesian grids are well suited for solving problems in rectangular domains. Mapped grids and domain embedding techniques are often used to apply rectangular grids to more general domains. In particular, the use of the logically rectangular polar grid is widely used for problems in circular or spherical domains. Other grids of this type include the standard latitude-longitude grid used for the sphere, and the spherical grid used for the ball.

However, these three standard grids – the polar grid, the spherical grid, and the latitude-longitude grid – all suffer from the problem that the ratio of the largest cells to the smallest grows as the grid is refined. As a result, when using explicit time stepping schemes with these grids, one is forced to take a time step that is much smaller than desirable, in order to respect the CFL stability limits imposed by the smallest cells. A second but related problem with these grids is that the grid resolution is poorly distributed. The polar grid, for example, has cells with very large aspect ratios near the outer edge and tiny cells near the center. For many problems of practical interest, this distribution of large and small cells does not match the physical requirements of the problem.

Because of the limitations imposed by these standard grids, several researchers have proposed grid mappings which fix the problem of cell size distribution. One popular approach is to map multiple grids in patchwork fashion to the disk, the sphere or the ball. One example of such a grid is the *gnomic projection* grid (sometimes called the "cubed-sphere" grid), first described by [6]. Using this approach, one patches together six logically Cartesian grids to cover the sphere. In a related approach, one maps five grids to the disk, or seven cubes to the ball. This approach can lead to good results when used with appropriate solvers (e.g. see [5]). However, the chief disadvantage of these mappings is that they require that one manage a multi-block structure with non-trivial communication between bordering cells of the patches. Furthermore, modifying a standard adaptive mesh refinement code for use in a multi-block setting presents many technical challenges.

In the work presented here, we focus our attention on describing a grid mapping for the sphere that attempts to fix the cell size distribution problem. Unlike the multi-block grid for the sphere described above, our mapping covers the entire sphere with a single logically Cartesian grid. Furthermore, the resulting mesh has the desirable property that the area of the largest to smallest cell is about 2, and it is easily adapted for use with an existing adaptive mesh refinement code. We show that this works well in practice when used with explicit finite-volume schemes for hyperbolic problems.

2 The mappings

In this section, we describe a grid mapping for the unit disk and the unit sphere. Our mapping for the sphere (e.g. the surface of the solid ball) is based on the mapping for the unit disk, and so we start with a description of that mapping. In Section 2.2, we extend this mapping to the sphere.

2.1 Mapping the square to the unit disk

In this section, we describe a mapping which maps the computational domain $[-1,1] \times [-1,1]$ to the unit disk. To describe the mapping, we focus our attention on the region of the square in which computational coordinates (ξ,η) satisfy $|\xi| \leq \eta$. This region corresponds to the upper triangular region between the two diagonals of the square. We refer to this region as the *north sector* of the computational domain. The mapping in this sector is based on the idea that we can map a horizontal line segment between points $(-d,d)$ and (d,d) to a circular arc of radius $R(d)$, passing through points $(-D(d),D(d))$ and $(D(d),D(d))$. The center of the circle on which this arc lies is given by $(x_0,y_0) = (0, D(d) - \sqrt{R(d)^2 - D(d)^2})$. Using this idea, a general point (ξ,η) in the north sector is mapped to the unit disk by the mappings $X_n(\xi,\eta)$ and $Y_n(\xi,\eta)$ given by

$$\left.\begin{aligned} X_n(\xi,\eta) &= D(\eta)\frac{\xi}{\eta} \\ Y_n(\xi,\eta) &= D(\eta) - \sqrt{R(\eta)^2 - D(\eta)^2} + \sqrt{R(\eta)^2 - X_n(\xi,\eta)^2}, \end{aligned}\right. \qquad |\xi| \leq \eta. \tag{1}$$

The mapping to other sectors is easily obtained by negating and/or swapping the arguments (ξ,η) or functions (X_n, Y_n). For example, the mapping in the west sector is given by $X_w(\xi,\eta) = Y_n(\eta,-\xi)$ and $Y_w(\xi,\eta) = -X_n(\eta,-\xi)$. MATLAB scripts for this and other mappings are given in [2].

There are many choices for $R(d)$ and $D(d)$ that lead to reasonable grids. For example, one choice that is well suited for the unit disk is

$$\begin{aligned} D(d) &= d/\sqrt{2} \\ R(d) &= 1 \end{aligned} \tag{2}$$

This choice leads to the grid on the left in Figure 1. One key advantage of this grid for explicit computations is that the ratio of the largest to the smallest grid cells is about 2, and so there is no need to take artificially small times steps because of the presence of a few small cells. In [2], we use this grid to simulate a blast wave in a circular domain and observed no artifacts attributable to the non-smoothness of the grid.

If one wants increased refinement near the boundary, either to model a boundary layer or, as we will see, to construct the sphere mapping, it may be useful to define $D(d)$ so that azimuthal grid lines are compressed near the outer edge. An example of a function $D(d)$ which does this is

$$D(d) = d(2-d)/\sqrt{2} \tag{3}$$

The plot at the right in Figure 1 shows a grid generated using this $D(d)$, along with $R(d) = 1$.

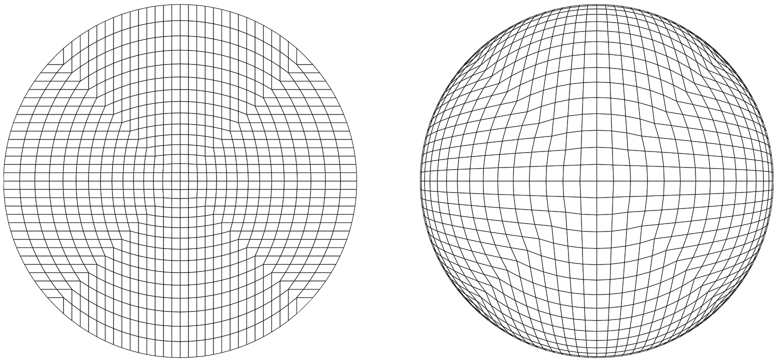

Fig. 1. Grids for circular domains. The grid on the left can be used for calculations in the disk. The grid on the right is useful for constructing the sphere grid described in Section 2.2.

2.2 A mapping for the unit sphere

The above mapping for the unit disk can be used directly to create a grid for the unit sphere. In general, if we are given a mapping $X(\xi,\eta)$ and $Y(\xi,\eta)$ from the square $[-1,1]\times[-1,1]$ to the unit circle, we can define mapping functions $X_s(\xi,\eta)$, $Y_s(\xi,\eta)$ and $Z_s(\xi,\eta)$ from the rectangular region $[-3,1],\times[-1,1]$ to the sphere as

$$
\begin{aligned}
X_s(\xi,\eta) &= \begin{cases} X(-(\xi+2),\eta) & \text{if} \quad |\xi+2| \le 1 \\ X(\xi,\eta) & \text{if} \quad |\xi| \le 1 \end{cases} \\
Y_s(\xi,\eta) &= Y(\xi,\eta) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4) \\
Z_s(\xi,\eta) &= \begin{cases} -\sqrt{1-(X(-(\xi+2),\eta)^2+Y(\xi,\eta)^2)} & \text{if} \quad |\xi+2| \le 1 \\ \sqrt{1-(X(\xi,\eta)^2+Y(\xi,\eta)^2)} & \text{if} \quad |\xi| \le 1. \end{cases}
\end{aligned}
$$

Using functions $X(\xi,\eta)$ and $Y(\xi,\eta)$ based on $D(d)$ as defined in (2) leads to a mapping with extremely elongated cells near the equator. Using (3) leads to a sphere mesh with cells which are more equi-distributed in size. The results are shown in Figure 2.

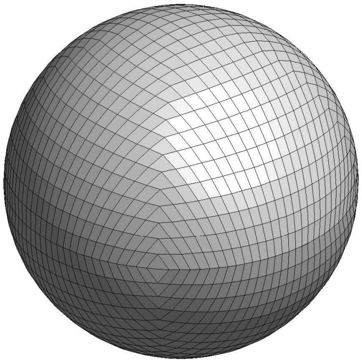

Fig. 2. Finite-volume sphere grid based on a single logically Cartesian grid.

3 Numerical results

In this section, we present the results obtained using our mappings to solve hyperbolic problems. The algorithms we use are the wave propagation algorithms described in [4]. These algorithms are finite volume Godunov-type methods based on solving Riemann problems at the mapped cell interfaces. One reason for choosing these algorithms is that it is easy to take advantage of the rotational invariance that is often present in the equations of practical and physical interest. In particular, we can re-use existing one-dimensional Riemann solvers to solve Riemann problems at mapped cell interfaces aligned along any direction. Moreover, our approach does not require that we provide metric terms analytically, but only that we approximate these terms using corners of mapped mesh cells. We do not present the details of these solvers here, but rather refer the interested reader to [2] and [4].

3.1 Advection on the sphere

Here we illustrate the use of the unit sphere mesh described in Section 2.2 for the approximation of advective transport problems. We discretize the equation

$$q_t + \mathbf{u} \cdot \nabla q = 0, \tag{5}$$

where $\mathbf{u}$ is a divergence free velocity field on the sphere and $q(\mathbf{x}, t)$ is a concentration that depends on space and time. A particularly simple flow situation is solid body rotation. In this case the exact solution after N rotations is equal to the initial conditions and it is therefore easy to perform convergence studies.

We use the wave propagation algorithm for quadrilateral grids on a sphere as described in [2]. This requires appropriately scaled flow speeds normal to each grid cell interface. In general, we can obtain a divergence-free velocity field at the centers of cell interfaces by differencing a streamfunction ψ at cell vertices adjacent to the cell interface. For solid body rotation, the streamfunction we use is

$$\psi(\mathbf{x}) = 2\pi(\mathbf{x} \cdot \mathbf{a}_{rot}), \tag{6}$$

where $\mathbf{x} \in R^3$ is a position vector (e.g. point) on the surface of the sphere and $\mathbf{a}_{rot} \in R^3$ is a normalized vector in the direction of the specified axis of rotation.

In order to perform a convergence study we initialize the concentration field q with a smooth function given by

$$q(\mathbf{x}, 0) = 2\exp(-10d^2),$$

where $d = \arccos(\mathbf{x} \cdot \mathbf{x}_0)$ is the geodesic distance (i.e. distance along the surface of the sphere) between the points $\mathbf{x}$ and $\mathbf{x}_0$. For this example, we choose $x_0 = (1, 0, 0)$.

We compare the final results after one rotation with our initial conditions. Table 1 shows the error in the L_1-norm and the experimental order of convergence (EOC) for rotation about the y-axis ($\mathbf{a}_{rot} = (0, 1, 0)$) and rotation about the z-axis ($\mathbf{a}_{rot} = (0, 0, 1)$). For this smooth solution we obtain second order convergence rates for rotation about both these axes.

For rotation about the z-axis, we also show the L_1-error and the EOC for calculations on a latitude-longitude grid with the same number of grid cells. On this latitude-longitude grid, the grid cells near the equator are larger than the grid cells on our new grid, and in fact, our new grid requires about twice as many times steps as the lat-long grid. The reduced resolution near the equator for the smooth lat-long grid may explain why this grid does not give better results than our new grid for flow around the equator, a flow field for which the lat-long grid is ideally suited. For rotation about the y-axis (e.g. over the poles) the situation is reversed, and the CFL restriction places severe time step restrictions on the lat-long grid.

Table 1. Convergence study for smooth advection test, $CFL = 0.9$, no limiters.

	axis of rotation : y-axis		axis of rotation : z-axis			
	new grid		new grid		lat-long grid	
grid size	L_1-error	EOC	L_1-error	EOC	L_1-error	EOC
100 × 50	0.231495		0.098360		0.100258	
200 × 100	0.063987	**1.85**	0.025643	**1.94**	0.027075	**1.89**
400 × 200	0.015991	**2.00**	0.006438	**1.99**	0.006833	**1.99**
800 × 400	0.004004	**2.00**	0.001607	**2.00**	0.001693	**2.01**

Next we illustrate the performance of the method for advective transport of a discontinuous profile defined as

$$q(\mathbf{x},0) = \begin{cases} 1 & \text{if} \quad \sin(3\theta) > 0 \\ 0 & \quad\;\;\; \text{otherwise.} \end{cases}$$

Figure 3 shows plots of the solution after one rotation along the equator.

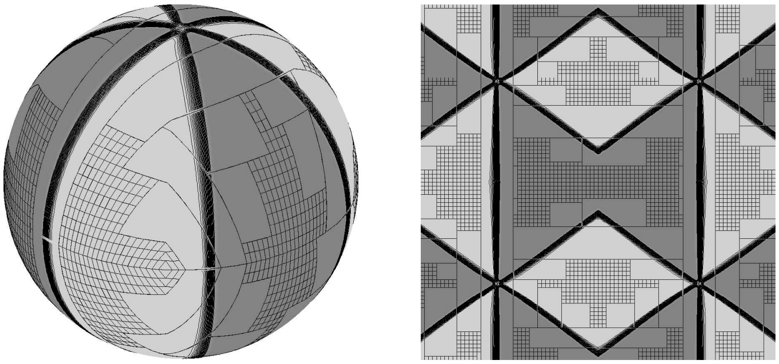

Fig. 3. Adaptive mesh refinement calculation for rotation of a discontinuous profile. The left plot shows the solution on the sphere after one rotation, and the right plot shows the solution in the computational domain used for the AMRCLAW calculation.

Adaptive mesh refinement on the sphere grid

One advantage of having the sphere grid represented by a single logically rectangular grid (as opposed to the multi-block structures required, for example, by [6]) is that we can easily use adaptive mesh refinement on our grid. To simplify the communication between boundaries, we double the size of the computational grid in the η-direction, creating a second copy of the sphere reflected across $\eta = -1$. Then the boundary conditions are simply periodic in both ξ

and η, a case that is already handled in AMRCLAW [1]. This could be avoided with some rewriting of the refinement routines to implement the appropriate boundary conditions on the original rectangular domain, which are periodic in ξ but along $\eta = -1$ and $\eta = +1$ are given by $q(\xi, \pm 1) = q(-2 - \xi, \pm 1)$ for $-3 \leq \xi \leq 1$. For this example, we use the AMRCLAW algorithm with two levels of refinement. On the coarse grid the sphere is discretized with 100×50 grid cells. On the fine mesh a refinement factor of 4 is used in each direction.

3.2 Shallow water equations on the sphere

The numerical solution to the shallow water equations on the sphere is of great interest to the global atmospheric and oceanic communities. Here, we solve the shallow water equations with no bottom topography on non-rotating sphere. The equations we solve are given by

$$\partial_t \mathbf{q} + \nabla \cdot \mathbf{f}(\mathbf{q}) = \mathbf{s}(\mathbf{x}, \mathbf{q}) \tag{7}$$

where $\mathbf{q} = (h, hu, hv, hw)$ is a vector of conserved quantities, and the flux function $\mathbf{f}(\mathbf{q})$ is given by

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} hu & hv & hw \\ hu^2 + \frac{1}{2}gh^2 & huv & huw \\ huv & hv^2 + \frac{1}{2}gh^2 & hvw \\ huw & hvw & hw^2 + \frac{1}{2}gh^2 \end{pmatrix} \tag{8}$$

The source term $\mathbf{s}(\mathbf{x}, \mathbf{q})$ which acts only on the momentum equations has the form

$$\mathbf{s}(\mathbf{x}, \mathbf{q}) = \begin{pmatrix} 0 \\ (\mathbf{x} \cdot \nabla \cdot \widetilde{\mathbf{f}})\mathbf{x} \end{pmatrix} \tag{9}$$

where $\mathbf{x}$ is the position vector on the sphere. The flux vector $\widetilde{\mathbf{f}} \in R^3$ consists of the second through fourth components of the flux vector $\mathbf{f}(\mathbf{q})$. The source term is included to ensure that the fluid velocity remains on the surface of the sphere. This approach was also taken by [3].

The height field is initialized on the sphere using the smooth function

$$h(\mathbf{x}) = 1 + 2\exp(-40(1 - (\mathbf{x} \cdot \mathbf{x}_0))^2). \tag{10}$$

This initial condition corresponds to a smooth, circular hump of fluid centered at the point $\mathbf{x}_0$ on the sphere. To solve the shallow water equations, we again use the algorithm described in [2]. As described there, we maintain conservation by subtracting out non-conservative terms after each time step.

In Figure 4, we show the results at time $t = 0.9$, on an adaptively refined mesh with three levels of refinement used. From these results, we see that the solution appears to remain symmetric with respect to the axis of symmetry,

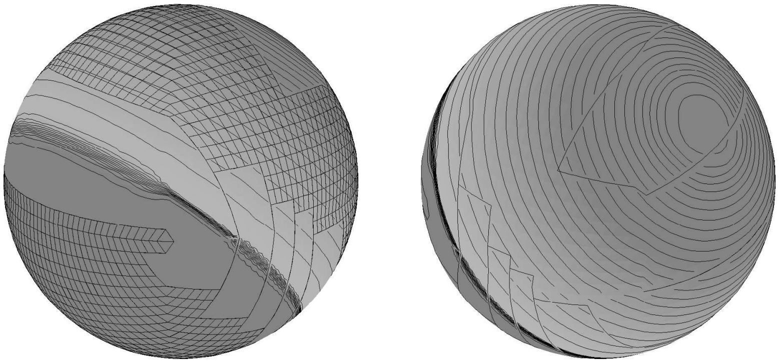

Fig. 4. Two views of the height field solution to the shallow water wave equation on the sphere at time $t = 0.9$. In the left plot, the mesh for the level 2 grid (out of 3 AMR grids) is shown.

4 Conclusions and Acknowledgements

The mappings and the code used for the examples shown in this paper are available at the website

`http://www.amath.washington.edu/~rjl/pubs/hyp2006`

This work was supported in part by DOE grant DE-FC02-01ER25474, NSF grant DMS-0106511 and German Science Foundation (DFG) grant SFB611. The first author also gratefully acknowledges the support of the Laboratoire d'Études des Transferts et de Méchanique des Fluides at the C.E.A.

References

1. M. J. Berger and R. J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35:2298–2316, 1998.
2. D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains. *submitted*, 2006. `http://www.amath.washington.edu/~rjl/pubs/circles`.
3. F. X. Giraldo. A spectral element shallow water model on spherical geodesic grids. *Int. J. Numer. Meth. Fluids*, 35:869–901, 2001.
4. R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.
5. J. A. Rossmanith. A wave propagation method for hyperbolic systems on the sphere. *J. Comput. Phys.*, 213:629–658, 2006.
6. R. Sadourny. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Monthly Weather Review*, 100:211–224, 1972.