Table 1: Wave propagation scheme

scheme acronym	full scheme name	primary reference	implementation grid
CLAW	Wave propagation algorithm on mapped grids	LeVeque (2002)	two-patch sphere grid

A standard test case suite for 2D linear transport on the sphere: results from 17 state-of-the-art schemes

Contribution : D. Calhoun, Boise State University

3 Transport schemes

3.4 Other grids

3.4.1 Wave-Propagation algorithm on the two-patch sphere grid

The CLAW scheme is based on the wave-propagation algorithms described by LeVeque in [4] and implemented in the CLAWPACK package [3]. The sphere grid used for the computations is described in [2] and is based on a novel mapping which transforms a single logically rectangular uniform Cartesian grid to the sphere. Our grid is similar to the cubed-sphere grid in that it is made up of $N \times N$ grid patches stretched to fit the sphere. Whereas the cubed-sphere uses six square patches, our grid consists of two square patches, one for each hemisphere, as shown in Figure 1. The computational grid is a single logically Cartesian $2N \times N$ grid mapped to the sphere via a simple mapping $\mathbf{T}(\xi, \eta)$ [2]. The resulting finite-volume mesh cells are nearly uniform in size. The effective angle for this mesh is $\lambda_{eff} = N/90$.

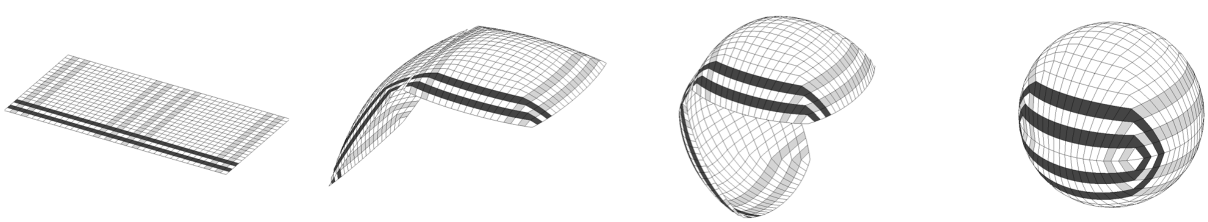

Figure 1: The two-patch sphere grid used by the CLAW scheme.

An advantage of this mapping from a single rectangle to the sphere is that standard block-structured adaptive mesh refinement (AMR) algorithms can be applied. This requires handling the boundary conditions properly, which are similar but not identical to periodic boundary conditions. The AMR algorithms in CLAWPACK have been extended to work on these grids; see [1] for a discussion of the boundary conditions and convergence tests for shallow water equations on the sphere.

The wave-propagation algorithm, as implemented in CLAWPACK can solve equations in both conservation (flux) form and non-conservative (advective) form. Results presented here are based on solving

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0 \tag{1}$$

In the nondivergent case, this form of the equation is equivalent to the conservative form. In the divergent case, this form, along with mass conservation, is equivalent to the conservative form. In both cases, we specify a normal velocity at cell edges. In the nondivergent case, we obtain the average normal velocity at mesh cell edges by differencing a streamfunction evaluated at mesh cell corners. By using the streamfunction in this manner, we exactly preserve the non-divergence of the wind field at the discrete level. As a result, we do not need to evolve the constant density field ρ, since the the wave propagation algorithm, when used in advective form, preserves constant tracer fields at the discrete level. For the divergent wind field, we compute the normal velocity at a cell edge by projecting the velocity vector onto the edge normal vector.

The wave-propagation algorithm can be run in first order or second order mode, with or without limiters, and as either a split scheme or an unsplit scheme. Here, we use the second order, unsplit scheme with the centralized monotonized difference limiter described in [4]. The maximum CFL for the wave propagation algorithm is one. All of our results for the benchmark exercises were run at a CFL of 0.95.

Computationally, the wave-propagation algorithm can be made very efficient when several tracers are present. For both the conservative and non-conservative form, the wave propagation algorithm is based on solving Riemann problems at mesh cell edges. Since all of the tracers are traveling at the same wave speed, the results of the Riemann problem can be combined into a single wave. This avoids the need to iterate over seperate waves when updating fluctuations and computing second order correction terms. Also, by combining tracer fields into a single wave, we can save on storage that would otherwise be required for storing seperate waves. The stencil required to update a given mesh cell involves eight neighboring cells in the unlimited case and an additional 12 neighboring cells in the limited case. Therefore, the maximum stencil size required to update a given mesh cell involves all but the corners cells of a 5×5 block surrounding the mesh cell.

Details of our scheme can be found in [4] and [2].

References

[1] M. J. Berger, D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular finite volume methods with adaptive refinement on the sphere. *Phil. Trans. R. Soc. A*, 367:4483–4496, 2009.

[2] D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains. *SIAM Review*, 50(4):723–752, 2008.

[3] R. J. LeVeque. ClawPack : Conservation Law Package, Version. 4.3 (April. 2006). `http://www.clawpack.org`.

[4] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.