

# Discontinuous Galerkin Transport on the Spherical Yin–Yang Overset Mesh

DAVID M. HALL AND RAMACHANDRAN D. NAIR

*Computational and Information Systems Laboratory, National Center for Atmospheric Research,\* Boulder, Colorado*

(Manuscript received 28 March 2012, in final form 15 June 2012)

## ABSTRACT

A discontinuous Galerkin (DG) transport scheme is presented that employs the Yin–Yang grid on the sphere. The Yin–Yang grid is a quasi-uniform overset mesh comprising two notched latitude–longitude meshes placed at right angles to each other. Surface fluxes of conserved scalars are obtained at the overset boundaries by interpolation from the interior of the elements on the complimentary grid, using high-order polynomial interpolation intrinsic to the DG technique. A series of standard tests are applied to evaluate its performance, revealing it to be robust and its accuracy to be competitive with other global advection schemes at equivalent resolutions. Under p-type grid refinement, the DG Yin–Yang method exhibits spectral error convergence for smooth initial conditions and third-order geometric convergence for  $C^1$  continuous functions. In comparison with finite-volume implementations of the Yin–Yang mesh, the DG implementation is less complex, as it does not require a wide halo region of elements for accurate boundary value interpolation. With respect to DG cubed-sphere implementations, the Yin–Yang grid exhibits similar accuracy and appears to be a viable alternative suitable for global advective transport. A variant called the Yin–Yang polar (YY-P) mesh is also examined and is shown to have properties similar to the original Yin–Yang mesh while performing better on tests with strictly zonal flow.

## 1. Introduction

For decades, spectral-transform methods have been the preferred choice for global atmospheric modeling. At moderate resolutions, they are spectrally accurate, computationally efficient, and simple to implement. However, the needs of the modeling community are shifting toward higher resolutions, long time integrations, and massive parallelization on distributed-memory machines. None of these trends appear to favor global spectral transforms. Mesh-based methods on the other hand, such as finite-volume, finite-element, and spectral-element methods, appear to be excellent candidates for this new set of requirements.

Mesh-based methods are local, requiring data exchange only between neighboring elements, minimizing communications, and enabling them to scale well on

distributed-memory supercomputers. Furthermore, when cast in conservative form, some versions are able to maintain exact mass conservation over long integration times, which is critical for multidecadal climate simulations. Spectral-element methods, in particular, also retain the high accuracy and exponential error convergence of the spectral transform. The efficiency and accuracy of mesh-based methods depend, in general, on the details of the mesh that is employed. Constructing an optimal global mesh for atmospheric models is nontrivial, and there are many ways to do so, as discussed in the recent review by Staniforth and Thuburn (2012).

The most popular mesh, by far, is the regular longitude–latitude (RLL) grid because it is logically rectangular, orthogonal, and simple to implement. However, it has long been recognized that the RLL grid is plagued by a set of issues collectively known as “the pole problem.” Its meridians converge at the North and South Poles, creating numerical singular points that must be dealt with specially. More troublingly, meridian convergence also produces a longitudinal grid length near the poles that is a small fraction of that at the equator. This severely restricts explicit time stepping schemes due to Courant–Friedrichs–Lewy (CFL) stability limitations, and the ratio of smallest to largest

---

\* The National Center for Atmospheric Research is sponsored by the National Science Foundation.

---

*Corresponding author address:* David M. Hall, Computational and Information Systems Laboratory, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80305.  
E-mail: halldm2000@gmail.com

element becomes increasingly unfavorable as global resolution is increased.

Historically, many grids have been constructed that cover the globe uniformly while avoiding the pole problem. Some of the more popular variants include the cubed-sphere, icosahedral meshes, composite (overset) meshes, and unstructured meshes. Phillips (1959) constructed the first overset mesh, using two polar stereographic projections at high latitudes overlapping a low-latitude Mercator projection. Sadourny (1972) subsequently constructed nonoverlapping polyhedral meshes, including the cubed sphere, hoping to restore conservation by avoiding overset mesh interpolation. The cubed sphere was revived and improved upon two decades later by both Ronchi et al. (1996) and Rančić et al. (1996) and has been employed since in spectral-element solvers by Taylor et al. (1997), Nair et al. (2005), and Thomas and Loft (2000). Geodesic icosahedral grids were employed by Sadourny et al. (1968) and Williamson (1968) and later applied to finite-element techniques by Giraldo (1997). A more comprehensive history of mesh methods in atmospheric modeling may be found in the review articles by Williamson (2007) and Staniforth and Thuburn (2012).

More recently, a very promising overset mesh referred to as the “Yin–Yang” mesh was proposed by Kageyama and Sato (2004) and a similar mesh was proposed by Purser (2004). The Yin–Yang mesh comprises two segments of the RLL mesh placed at right angles to each other, with a small amount of overlap. Some of the advantages of the Yin–Yang mesh include

- it avoids the pole problem of the RLL grid, and there are no singular points
- each grid component is orthogonal, producing a simple analytical form for partial differential equations
- the grid spacing is quasi-uniform with a largest–smallest grid-length ratio of only  $\sqrt{2}$ , allowing a larger explicit time step than the RLL grid
- fewer grid points are needed to mesh the sphere relative to the RLL grid at the same equatorial resolution
- the block rectangular grids facilitate domain decomposition methods for elliptic solvers
- as each component is a section of the RLL grid, existing codes can be adapted for analysis and visualization of results

Although overset methods are not inherently conservative, Peng et al. (2006) showed that exact conservation can be restored through the application of a local, cell wise constraint.

Since its inception, a semi-Lagrangian scheme was developed for the Yin–Yang mesh by Li et al. (2006),

followed by a multimoment finite-volume shallow-water model by Li et al. (2008). Subsequently, the Yin–Yang grid was employed in a terrain-following shallow-atmosphere dynamical core by Baba et al. (2010). The grid has also been applied recently in weather forecasting simulations by Qaddouri and Lee (2011).

Much larger time steps can be achieved with implicit or semi-implicit time stepping methods. Doing so produces an elliptical boundary value problem that must be solved at each time step, which may be accomplished on overset grids using a domain decomposition technique introduced by Schwarz (1870). An optimized version of the Schwarz method was applied to the Yin–Yang grid by Qaddouri et al. (2008) and Qaddouri (2011) to solve the shallow-water equations. Quite recently, Zerroukat and Allen (2012) showed that Krylov solvers may also be employed to solve elliptic problems on the Yin–Yang grid with as much efficiency as a single grid.

Although finite-volume implementations of the Yin–Yang grid have proven to be successful, they require the construction of a large halo region of grid cells at the overset boundary to achieve high-order accuracy. In addition to adding complexity, the halo region limits the method’s parallel scalability by requiring more inter-processor communication. Locality can be improved by reducing the order of accuracy of discretizations at the overset boundary, but this also reduces the accuracy of the scheme as a whole as seen by Peng et al. (2006).

As an alternative, we propose the application of a high-order discontinuous Galerkin (DG) finite-element solver on the Yin–Yang mesh. As the DG method is strictly local, and comes equipped with a high-order interpolation scheme, we anticipate that it will be very well suited to the Yin–Yang grid. While DG overset methods are reasonably well known in the field of aerospace engineering as demonstrated by Nastase et al. (2011), they are much less common in atmospheric science and we are not aware of any prior application of the DG method to the Yin–Yang mesh.

In the remainder of this paper, a DG implementation of the Yin–Yang grid on the sphere is presented and a series of standard benchmarks are applied to analyze its performance. Section 2 introduces details of the Yin–Yang grid construction, and section 3 presents the conservative transport equations on the sphere. Section 4 discusses the nodal discontinuous Galerkin method used for spatial discretization as well as an explicit Runge–Kutta discretization in time. Section 5 presents some details of the overset boundary implementation. In section 6, a series of numerical benchmarks and error convergence studies are applied to examine the performance of the scheme. Finally, conclusions are drawn in section 7.

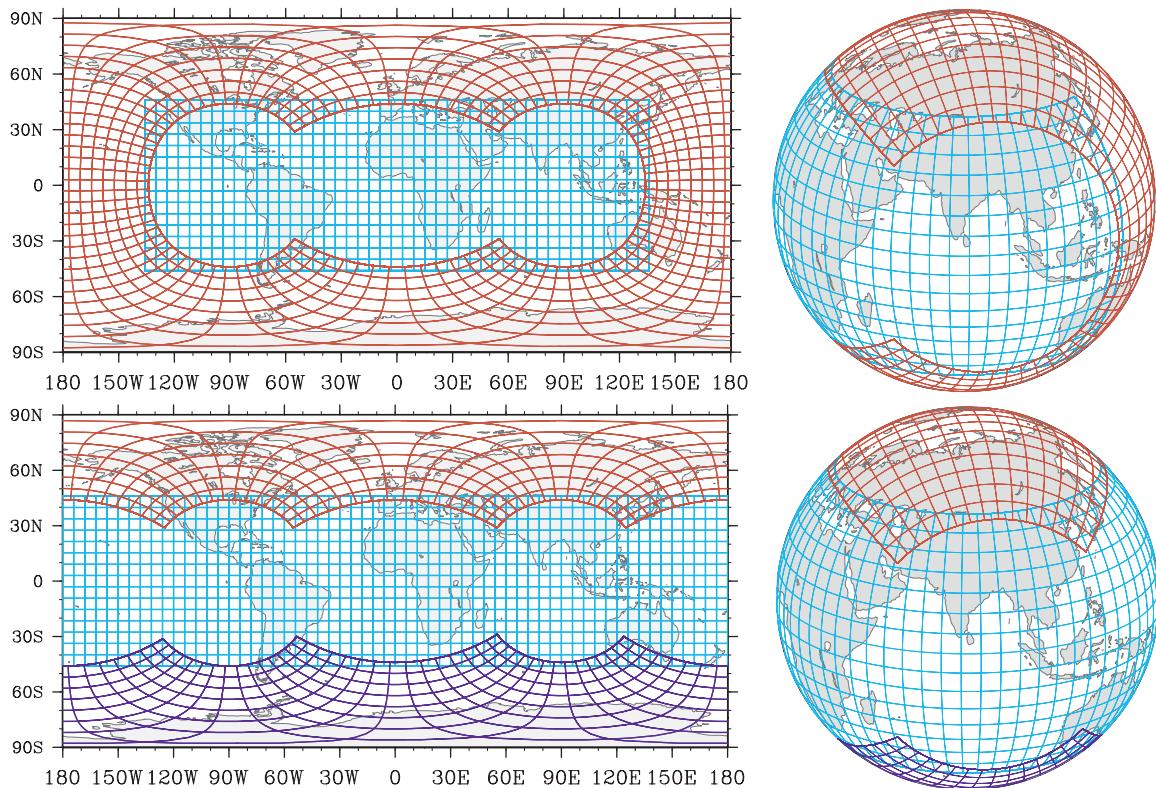


FIG. 1. The YY and YY-P meshes. (top) The YY mesh comprises the Yin region  $\mathcal{Y}$  in blue and the Yang region  $\mathcal{Y}'$  in red. (bottom) In the YY-P mesh, the Yin region  $\mathcal{Y}_E$  is extended to the entire equatorial region, and the Yang region is split into a North Polar cap  $\mathcal{Y}'_N$  and a South Polar cap  $\mathcal{Y}'_S$ .

## 2. The YY meshes

### a. The YY mesh

The Yin–Yang (YY) mesh of Kageyama and Sato (2004) consists of two identical RLL segments that lie at right angles to each other and overlap at their edges as illustrated in the top half of Fig. 1. Denoting the Yin region by  $\mathcal{Y}$  and the Yang region by  $\mathcal{Y}'$ , their union seamlessly covers the sphere such that  $\mathcal{S} = \mathcal{Y} \cup \mathcal{Y}'$ . The longitudinal orientation of the grids is arbitrary, and we have chosen to center the Yin grid at the prime meridian for symmetry.

The Yin grid consists of a rectangular region  $\mathcal{Y}$  with longitude–latitude coordinates  $\boldsymbol{\lambda} = (\lambda, \theta)$ . The longitudinal section spans three-quarters of the globe with  $\lambda \in [-3\pi/4 - \delta, 3\pi/4 + \delta]$ , where  $\delta$  is the minimum grid-overlap parameter, illustrated in Fig. 2, which may be small but must be nonzero. The latitudinal section consists of the low-latitude region  $\theta \in [-\pi/4 - \delta, \pi/4 + \delta]$ , where the grid is nearly uniform. Using the absolute 3D Cartesian coordinates  $\mathbf{x} = (x, y, z)$  for reference, the polar axis of the Yin grid is aligned with the  $z$  axis,  $\lambda = 0$  is aligned with the  $x$  axis, and  $\lambda = \pi/2$  is aligned with the  $y$  axis.

The Yang region  $\mathcal{Y}'$  is identical to the Yin region, but in a rotated coordinate system  $\boldsymbol{\lambda}' = (\lambda', \theta')$ . The polar axis lies along the  $-x$  Cartesian axis,  $\lambda' = 0$  is aligned with the  $y$  axis, and  $\lambda' = \pi/2$  is aligned with the  $z$  axis. If a rotated Cartesian coordinate system  $\mathbf{x}' = (x', y', z')$  is associated with the Yang grid, the Yin and Yang axes are related by  $(x', y', z') = (-x, z, y)$ .

Because of the overlapping elements at the corners, the surface area of the Yin–Yang mesh exceeds that of the sphere by 6.2%, when the minimum overlap is  $\delta = 0$ .

### b. The YY-P mesh

A modified version of the Yin–Yang grid was recently discussed in the review paper by Staniforth and Thuburn (2012), where the Yin grid is extended to cover the entire equatorial region and the Yang grid is split to form two polar caps as shown in the bottom half of Fig. 1. This variant, which we refer to as the Yin–Yang polar (YY-P) mesh, would appear to be more advantageous for purely zonal flows, as little to no interpolation is necessary in such a case.

The YY-P mesh is defined such that  $\mathcal{Y}_E$  is the equatorial region  $\boldsymbol{\lambda} \in [-\pi, \pi] \otimes [-\pi/4 - \delta, \pi/4 + \delta]$  of the RLL coordinates system. The  $\mathcal{Y}'_N$  region is the North

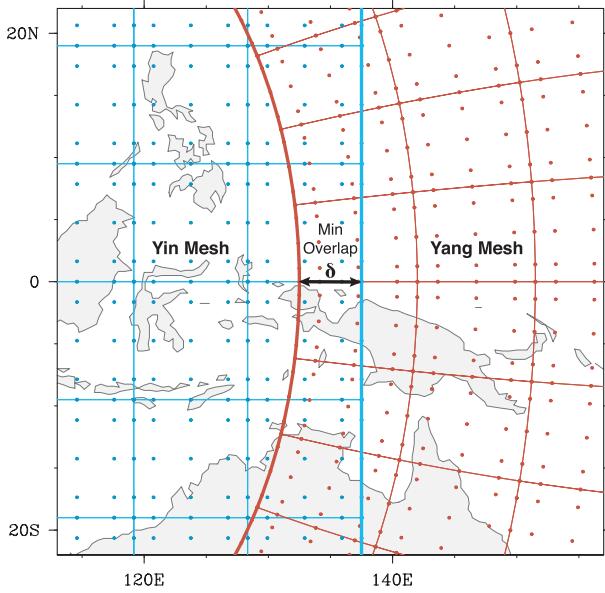


FIG. 2. Close-up view of the YY overset boundary where thin lines denote elements, thick lines denote the overset boundaries, and dots indicate the nodal GLL quadrature points. The factor  $\delta$  controls the minimum overlap (shown here is the value  $\delta = 2.5^\circ$ ). A default overlap of  $\delta = 0.1^\circ$  is used unless otherwise specified.

Polar section  $\lambda' \in [\pi/4 - \delta, 3\pi/4 + \delta] \otimes [-\pi/4 - \delta, \pi/4 + \delta]$  of the Yang coordinate system, and the  $\mathcal{Y}'_S$  region is the South Polar section  $\lambda' \in [-\pi/4 + \delta, -3\pi/4 - \delta] \otimes [-\pi/4 - \delta, \pi/4 + \delta]$  of the Yang coordinate system. As before, the union of all three regions seamlessly covers the sphere such that  $\mathcal{S} = \mathcal{Y}_E \cup \mathcal{Y}'_N \cup \mathcal{Y}'_S$ .

### 3. Conservative transport on the sphere

Consider the transport equation in flux form for a conserved scalar field  $\psi$  on the sphere  $\mathcal{S}$ ,

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{F} = 0 \quad \text{in } \mathcal{S} \times (0, T], \quad (1)$$

where  $\mathbf{F} = \psi \mathbf{v}$  is the mass flux and  $\mathbf{v}$  is the advecting velocity field. The divergence of  $\mathbf{F}$ , written in RLL (Yin) coordinates  $(\lambda, \theta)$ , is

$$\nabla \cdot \mathbf{F} = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial \lambda} (\sqrt{g} F^\lambda) + \frac{\partial}{\partial \theta} (\sqrt{g} F^\theta) \right], \quad (2)$$

where  $g = r \cos^2 \theta$  is the determinant of the covariant metric from (A9),  $F^\lambda = \psi \dot{\lambda} = \psi u / (r \cos \theta)$  is the longitudinal flux, and  $F^\theta = \psi \dot{\theta} = \psi v / r$  is the latitudinal flux. Upon substitution, the transport equation is

$$\frac{\partial \psi}{\partial t} + \frac{1}{r \cos \theta} \left[ \frac{\partial}{\partial \lambda} (u \psi) + \frac{\partial}{\partial \theta} (v \cos \theta \psi) \right] = 0, \quad (3)$$

where  $u = (r \cos \theta) \dot{\lambda}$  is the zonal wind speed and  $v = r \dot{\theta}$  is the meridional wind speed.

The transport equation in Yang coordinates takes the same form in the rotated coordinate system  $\lambda'$ , but the wind speeds  $u' = r \cos \theta' \dot{\lambda}'$  and  $v' = r \dot{\theta}'$  are neither strictly zonal nor strictly meridional. However, the zonal and meridional components of  $\mathbf{u}' = (u', v')$  may be obtained by applying the transformation matrix  $\mathbf{M}$ :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{M} \begin{bmatrix} u' \\ v' \end{bmatrix}, \quad (4)$$

where

$$\mathbf{M} = \begin{bmatrix} -\sin \lambda \sin \lambda' & -\cos \lambda' / \cos \theta \\ \cos \lambda' / \cos \theta & -\sin \lambda \sin \lambda' \end{bmatrix}. \quad (5)$$

### 4. DG method

The discontinuous Galerkin method may be viewed as a hybrid technique combining many of the best characteristics of spectral, finite-element, and finite-volume methods. The DG spatial discretization combined with Runge–Kutta time integration provides a class of robust algorithms known as the RKDG method for conservation laws as described by Cockburn and Shu (1989). For an excellent introduction to both discontinuous and continuous Galerkin spectral-element methods, please see Kopriva (2009) as well as Karniadakis and Sherwin (2005).

Both nodal and modal variants of the DG method are popular in the atmospheric sciences, as demonstrated by Giraldo et al. (2002), Nair et al. (2005), and Blaise and St-Cyr (2012). A comprehensive review of DG methods in atmospheric modeling may be found in Nair et al. (2011). In this work, a twice integrated nodal method is chosen, as discussed by Kopriva and Gassner (2010), with nodes placed at the Gauss–Lobatto–Legendre (GLL) quadrature points to simplify the implementation of the overset boundary.

#### a. Nodal DG formulation

Beginning with the transport equation,

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{F} = 0 \quad \text{in } \mathcal{Y} \text{ or } \mathcal{Y}', \quad (6)$$

the nodal DG formulation proceeds by partitioning the domain into a set of nonoverlapping elements  $\Omega_e$  such that  $\mathcal{Y} = \cup_{e=0}^{N_y} \Omega_e$ , where  $N_y = N_e \times 3N_e$  is the number of elements in the region and  $N_e$  is the number of elements spanning the short edge of the Yin or Yang mesh. Each element subtends a square angular region

$\Delta\lambda = \Delta\theta = \pi/2N_e$  containing  $N_g \times N_g$  nodal grid points for a total of  $N = 6N_e^2N_g^2$  degrees of freedom for the entire mesh.

On a given element  $\Omega_e$ , the approximate solution  $\psi_e$  is assumed to exist in the vector space  $V_e$  of polynomials of up to degree  $P = N_g - 1$  defined by

$$V_e = \{\varphi \in L^2(\mathcal{Y}): \varphi|_{\Omega_e} \in \mathbf{P}_P(\Omega_e), \quad \forall \quad \Omega_e \in \mathcal{Y}\}, \quad (7)$$

where

$$P_P = \text{span}\{\lambda^m \phi^n: 0 \leq m, n \leq P\}. \quad (8)$$

The transport equation is then multiplied by a smooth test function  $\varphi_e$  chosen from the same polynomial space  $V_e$  and integrated over the element  $\Omega_e$  to obtain the weak (integral) Galerkin form as described in Cockburn and Shu (2001). The divergence theorem is then applied to generate a surface term

$$\int_{\Omega_e} \left( \varphi_e \frac{\partial \psi_e}{\partial t} - \mathbf{F} \cdot \nabla \varphi_e \right) d\Omega + \int_{\Gamma_e} \hat{\mathbf{F}} \cdot \hat{\mathbf{n}} \varphi_e d\Gamma = 0, \quad (9)$$

where  $\hat{\mathbf{F}}$  is the flux boundary value called the ‘‘numerical flux,’’  $\hat{\mathbf{n}}$  is the outward-facing normal vector on the element boundary  $\Gamma_e$ , and the terms  $d\Omega$  and  $d\Gamma$  are differential units of area and length. The divergence theorem is applied a second time to recover the divergence form of the volume term

$$\int_{\Omega_e} \varphi_e \left( \frac{\partial \psi_e}{\partial t} + \nabla \cdot \mathbf{F} \right) d\Omega + \int_{\Gamma_e} (\hat{\mathbf{F}} - \mathbf{F}) \cdot \hat{\mathbf{n}} \varphi_e d\Gamma = 0. \quad (10)$$

All integrals are approximated using Gaussian numerical quadrature over the GLL quadrature points. To apply Gaussian quadrature, the element must first be mapped onto a square reference element with coordinates  $\xi = (\xi, \eta)$ , where  $\xi \in (-1, 1)$  and  $\eta \in (-1, 1)$ . In reference coordinates, the divergence of  $\mathbf{F}$  is

$$\nabla \cdot \mathbf{F} = \frac{1}{\sqrt{g_r}} \left[ \frac{\partial}{\partial \xi} (\sqrt{g_r} F^\xi) + \frac{\partial}{\partial \eta} (\sqrt{g_r} F^\eta) \right], \quad (11)$$

where  $g_r$  is the determinant of the composite transformation from reference to Cartesian coordinates as derived in (A38). Transforming the velocity into reference coordinates gives the reference fluxes and reference Jacobian

$$F^\xi = \psi_e \dot{\xi} = (2/\Delta\lambda)(\psi_e u/r \cos\theta), \quad (12)$$

$$F^\eta = \psi_e \dot{\eta} = (2/\Delta\theta)(\psi_e v/r), \quad \text{and} \quad (13)$$

$$\sqrt{g_r} = \frac{\Delta\lambda\Delta\theta}{4} (r^2 \cos\theta). \quad (14)$$

Spatial discretization is performed by approximating each function as a sum of polynomial basis functions  $\ell(\xi)$ , which are high-order Lagrange interpolating polynomials:

$$\ell_i(\xi) = \prod_{\substack{j=0 \\ j \neq i}}^P \frac{\xi - \xi_j}{\xi_i - \xi_j}. \quad (15)$$

A Lagrange polynomial  $\ell_i(\xi)$  takes the value  $\ell_i(\xi_i) = 1$  at node  $i$ ,  $\ell_i(\xi_j) = 0$  at node  $j \neq i$ , and interpolates smoothly between nodes. The interpolation root nodes  $\{\xi_i\}_0^P$  and  $\{\eta_j\}_0^P$  are chosen to coincide with the  $N_g$  GLL quadrature points to facilitate numerical integration. Two-dimensional basis functions are constructed from a tensor product of the one-dimensional bases, such that  $\psi$  is approximated by

$$\psi_e(\xi, \eta) = \sum_{i=0}^P \sum_{j=0}^P \psi_e(\xi_i, \eta_j) \ell_i(\xi) \ell_j(\eta). \quad (16)$$

Substituting the discretized scalar field and test functions into (10) and replacing integrals by Gaussian quadrature converts the partial differential equation into a set of ordinary differential equations (ODE) in time.

#### b. Time discretization

After spatial discretization, a set of ordinary differential equations remain that may be written abstractly as

$$\frac{d}{dt} U = \mathcal{L}(U), \quad (17)$$

where  $\mathcal{L}$  is a linear operator acting on the set of discretized unknown coefficients  $U$ . For the transport equation, the set of unknowns  $U$  corresponds to the set of discrete nodal values of  $\psi$ .

Third- and fourth-order Runge–Kutta schemes are popular in the DG literature for the solution of this type of ODE. However, in cases where the fields are smooth and well resolved and the polynomial order is high ( $N_g > 4$ ), the numerical error is entirely dominated by the low-order time stepping scheme, requiring an exceedingly small time step to achieve full accuracy.

Therefore, we prefer the strong stability-preserving Runge–Kutta (SSP-RK) method of Gottlieb et al. (2001), which produces time stepping schemes of arbitrarily high-order  $m$ , while keeping the CFL number at or near 1. The scheme is described as follows:

$$\begin{aligned}
 U^{(0)} &= U^n \\
 U^{(i)} &= U^{(i-1)} + \Delta t L U^{(i-1)} \quad \text{for } i = 1, \dots, m-1 \\
 U^{n+1} &= \sum_{k=0}^{m-1} \alpha_{m,k} U^{(k)} + \Delta t \alpha_{m,m-1} L U^{(m-1)}, \quad (18)
 \end{aligned}$$

where the coefficients  $\alpha$  are given by  $\alpha_{1,0} = 1$  and the recursive relationship

$$\begin{aligned}
 \alpha_{m,k} &= \frac{1}{k} \alpha_{m-1,k-1} \quad \text{for } k = 1, \dots, m-1 \\
 \alpha_{m,m-1} &= \frac{1}{m!} \quad \alpha_{m,0} = 1 - \sum_{k=1}^{m-1} \alpha_{m,k}. \quad (19)
 \end{aligned}$$

By matching the order  $m$  to the order of the approximating polynomials such that  $m = N_g - 1$ , full accuracy is achieved with the largest possible time step. The time step is computed using the estimate

$$\Delta t = C \frac{\Delta x}{v(2m + 1)}, \quad (20)$$

where  $C$  is the CFL number,  $\Delta x$  is the element width, and  $v$  is the maximum velocity. For this set of Runge–Kutta schemes, the maximum stable time step corresponds to a CFL number  $C = 1$ . While a more accurate estimate of the CFL number is available from Gassner and Kopriva (2011), the use of the above estimate is retained for simplicity, as it ensures that the simulation is stable up to a CFL number of 1.

*c. Riemann solver*

After a single time step, the advected field depends upon field values  $\psi(\xi_i, \eta_j)$  interior to the element but not on values from the exterior. Thus, the solution is local to each element and its value may change discontinuously at element boundaries. The piecewise continuous solutions are then coupled together by the exchange of mass via the mass flux  $\mathbf{F}$  across element interfaces.

The mass flux  $\mathbf{F}$  at the interface is not uniquely determined, and the correct numerical flux  $\hat{\mathbf{F}}$  must be determined by solving a Riemann problem (familiar from finite-volume schemes) as discussed in Toro (2009). For advective transport, if mass is flowing out of the element, the interior value  $\mathbf{F}^- = \psi^- \mathbf{v}$  should be used, and if mass is flowing into the element, the exterior value  $\mathbf{F}^+ = \psi^+ \mathbf{v}$  should be used, leading to the upwind flux formula

$$\hat{\mathbf{F}} \cdot \hat{\mathbf{n}} = \psi^- \frac{v_n + |\mathbf{v}_n|}{2} + \psi^+ \frac{v_n - |\mathbf{v}_n|}{2}, \quad (21)$$

where  $v_n = \mathbf{v} \cdot \hat{\mathbf{n}}$  is the velocity component in the outward normal direction. Regrouping terms, this may be written as

$$\hat{F}(F^-, F^+) = \frac{1}{2} (\mathbf{F}^- + \mathbf{F}^+) \cdot \hat{\mathbf{n}} + \frac{|\mathbf{v} \cdot \hat{\mathbf{n}}|}{2} (\psi^- - \psi^+), \quad (22)$$

which is called the Rusanov flux or sometimes the local Lax–Freidrichs flux.

**5. Overset boundary implementation**

The edges of the Yin and Yang grids are overset boundaries and as such their values must be interpolated from the interior of the complimentary grid. A given node point  $\lambda_k = (\lambda_k, \theta_k)$  on the Yin overset boundary corresponds to a nonnodal location within the Yang grid. Its position is found in Yang coordinates by applying the coordinate transformation (as derived in the appendix)

$$\theta'_k = \arcsin(\sin \lambda_k \cos \theta_k), \quad (23)$$

$$\lambda'_k = \arctan(-\tan \theta_k / \cos \lambda_k). \quad (24)$$

The element  $\Omega'_e$  containing the point is determined by a simple boundary intersection test. The Yang coordinates are then mapped into the reference element by the Yang to reference coordinate transformation

$$\xi'_k = 2(\lambda'_k - \bar{\lambda}'_e) / \Delta \lambda'_e, \quad (25)$$

$$\eta'_k = 2(\theta'_k - \bar{\theta}'_e) / \Delta \theta'_e, \quad (26)$$

where  $(\bar{\lambda}'_e, \bar{\theta}'_e)$  is the element center and  $(\Delta \lambda'_e, \Delta \theta'_e)$  is its extent. The element  $\Omega'_e$  and reference point  $\xi'_k$  associated with overset node  $\lambda_k$  need be computed only once at the beginning of the simulation and are cached for later use. The same process is repeated to locate the Yang overset nodes within the Yin grid.

At each time step of the simulation, values of the scalar field  $\psi$  are interpolated from the interior of the Yang grid at the overset points  $\{\lambda_k\}$  by employing the high-order Lagrange interpolation scheme inherent to the nodal DG method:

$$\psi_e(\xi'_k, \eta'_k) = \sum_{i=0}^P \sum_{j=0}^P \psi_e(\xi'_i, \eta'_j) \ell_i(\xi'_k) \ell_j(\eta'_k). \quad (27)$$

The value of a scalar field is independent of the coordinate system used, so  $\psi_e(\xi_k, \eta_k) = \psi_e(\xi'_k, \eta'_k)$ .

Each component  $(u, v)$  of the velocity field is interpolated individually from the Yang grid. Then the velocity components are transformed into the destination coordinate system using the Yang–Yin transformation matrix  $\mathbf{M}$  from (5).

The surface flux  $\mathbf{F}$  at the overset interface is computed from the interpolated scalar field values and the

transformed velocity field components and stored in a “ghost surface” representing the missing neighbor element. The usual Riemann solver may then be applied to compute the numerical flux.

## 6. Numerical experiments

A series of numerical experiments is performed to evaluate the effectiveness of the DG Yin–Yang scheme, including both rotational and deformational tests. Revolution of a Gaussian scalar field about the globe by a “solid-body” wind field is examined to assess the scheme’s performance and error convergence properties on fully resolved, smooth initial conditions, as suggested by Levy et al. (2007). Solid-body revolution of a cosine bell is also tested, as suggested by Williamson et al. (1992), to facilitate comparison with other global advection schemes. Last, the challenging moving-vortex test of Nair and Jablonowski (2008) is applied to measure the scheme’s performance on a more realistic case. For each numerical experiment, initial conditions are applied to the Yin and Yang grids independently using analytic values computed at their respective node points.

### a. Solid-body rotation of a Gaussian

The first numerical experiment examines the scheme’s ability to advect a smooth nonzero scalar field uniformly around the globe. The initial condition is the Gaussian scalar field

$$\psi(\mathbf{r}, 0) = \exp[-(\mathbf{r} - \mathbf{r}_0)^2/2a^2], \quad (28)$$

where  $\mathbf{r}$  is a point in Cartesian space, the center of the Gaussian is located at  $\mathbf{r}_0 = (r, 0, 0)$ , and its half-width is  $a = r/6$ . The wind velocity field is given by

$$\mathbf{v} = \mathbf{r} \times \boldsymbol{\omega}, \quad (29)$$

which is purely rotational with constant angular velocity  $\boldsymbol{\omega} = A\hat{\boldsymbol{\omega}}$  of amplitude  $A = (2.0\pi/12 \text{ days})$ . The tilt of the rotational axis is set by the parameter  $\alpha$  such that  $\hat{\boldsymbol{\omega}} = (0, -\sin\alpha, \cos\alpha)$ . The angle  $\alpha = 0$  corresponds to an equatorial flow from west to east while  $\alpha = \pi/2$  corresponds to a flow from north to south along the prime meridian, as described in Williamson et al. (1992).

#### 1) NUMERICAL RESULTS

Transport of the Gaussian about the globe is considered for three different orientations,  $\alpha = 0^\circ, 45^\circ$ , and  $90^\circ$ , on the YY mesh. The number of elements along the latitudinal section is fixed at  $N_e = 5$ , and the number of grid points per element edge is fixed at  $N_g = 10$ , producing a moderately sized simulation with 15 000

degrees of freedom. In each case, the numerical solution is compared to the analytical solution, the normalized error measures  $L_1$ ,  $L_2$ , and  $L_\infty$  are computed, and the results are plotted in Fig. 3. The normalized error measures are defined to be

$$L_1 = \int |\psi - \psi_a| d\Omega / \int |\psi_a| d\Omega, \quad (30)$$

$$L_2 = \left[ \int |\psi - \psi_a|^2 d\Omega \right]^{1/2} / \left[ \int |\psi_a|^2 d\Omega \right]^{1/2}, \quad \text{and} \quad (31)$$

$$L_\infty = \max(|\psi - \psi_a|) / \max(|\psi_a|), \quad (32)$$

where the integrals and maxima are taken over the entire mesh. In practice, the integrals are assessed using Gaussian numerical quadrature evaluated at the node points. As the polynomial order is  $P = N_g - 1 = 9$ , a ninth-order Runge–Kutta time stepping scheme is employed with CFL = 0.5. This CFL number was chosen to ensure the time-discretization error is small in comparison with the spatial-discretization error.

It can be seen in Fig. 3 that the errors are quite small (on the order of  $10^{-6}$ ) and relatively level, indicating that time-discretization error is not a significant factor. For equatorial  $\alpha = 0^\circ$  and polar  $\alpha = 90^\circ$  revolutions, the Gaussian curve crosses the Yin and Yang overset grids nearly simultaneously, while in the  $\alpha = 45^\circ$  case, the Yin and Yang boundaries are spaced farther apart.

In the  $\alpha = 0$  case, the center of the Gaussian crosses the Yin and Yang boundaries at  $t = 4.5$  days and again at  $t = 7.5$  days. Looking closely at the error norms in each case, we see that no significant error increase is observable at these times, indicating that the Gaussian is being passed smoothly between the two grids.

To compare the performance of the two Yin–Yang meshes, the above experiments were repeated on the YY-P mesh and  $L_\infty$  error norms for both sets are plotted in Fig. 4. Small differences do exist, with the YY-P mesh performing slightly better on the equatorial case  $\alpha = 0^\circ$  and the YY mesh performing slightly better on the polar case  $\alpha = 90^\circ$ . But the error norms differ by no more than a factor of 2, indicating that the choice of YY mesh versus YY-P mesh is fairly insignificant for smooth, well-resolved functions.

#### 2) CONVERGENCE STUDIES

Both h-type and p-type grid-refinement studies were performed on the Gaussian advection test on the YY mesh, at resolutions ranging from roughly  $N = 2000$  to  $N = 100\,000$  degrees of freedom, where  $N = 6N_e^2N_g^2$ . The h-refinement studies examine convergence of the numerical solution to the analytical solution by increasing

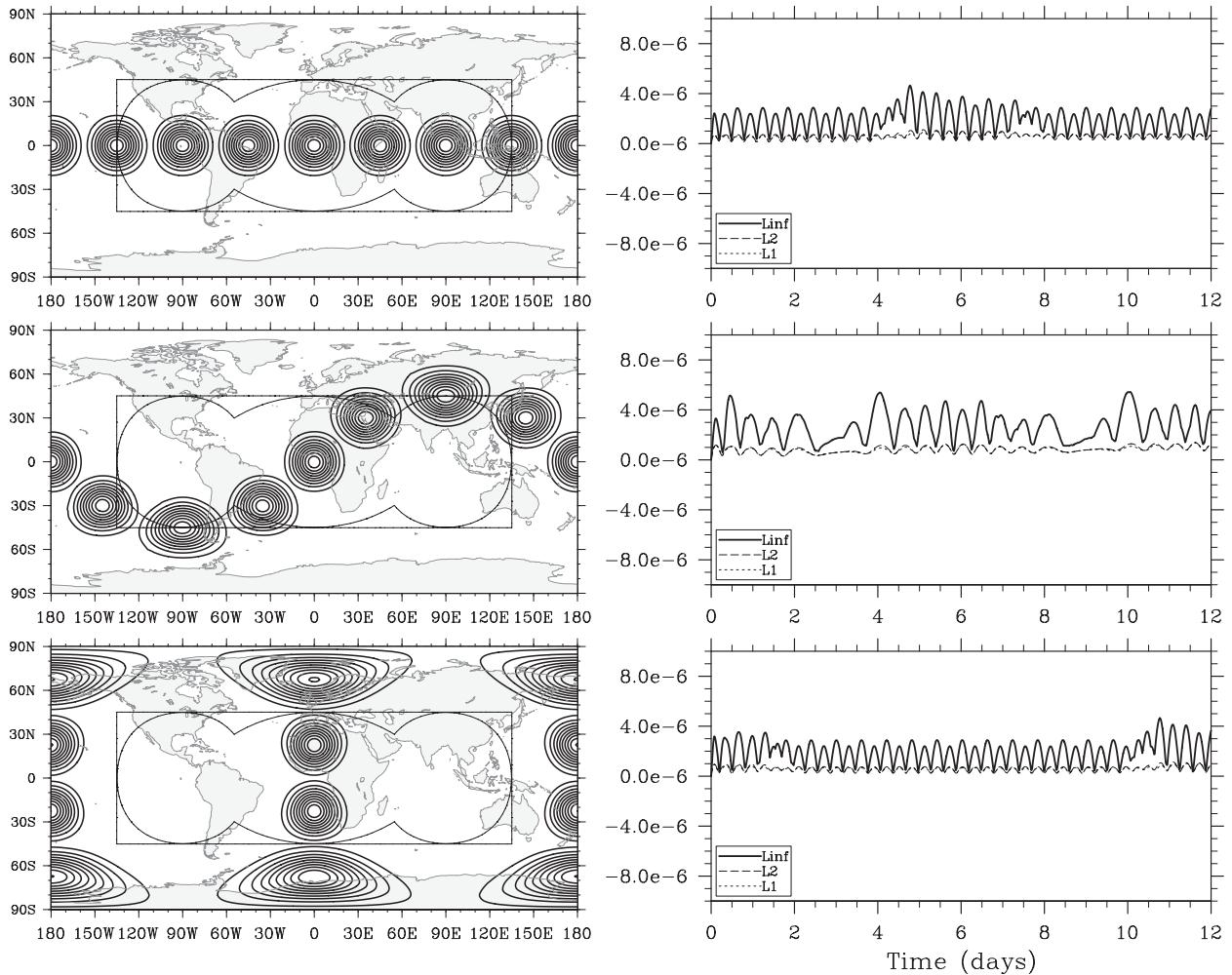


FIG. 3. Normalized error measures for Gaussian advection on the YY grid with  $6N_e^2N_g^2 = 6 \times 5^2 \times 10^2 = 15\,000$  degrees of freedom. The Gaussian crosses the overset boundaries smoothly in each case. (top) West-east revolution,  $\alpha = 0^\circ$ , (middle) southwest-northeast revolution,  $\alpha = 45^\circ$ , and (bottom) south-north revolution,  $\alpha = 90^\circ$ .

the number of elements per edge  $N_e$  while keeping the number of nodes per element  $N_g$  fixed, and the p-refinement studies examine the convergence by keeping the number of elements  $N_e$  fixed while increasing the polynomial order  $N_g$  of each element. Errors in each case were sampled at  $t = 12$  days, corresponding to a single revolution about the globe.

The second-order h-refinement study ( $N_g = 3$ ) exhibited the expected third-order convergence ( $-3.42$ ), achieving a minimum error of  $L_2 = 1.1 \times 10^{-3}$  at 43 elements per edge ( $N_e = 43$ ) corresponding to 99 846 degrees of freedom. The fourth-order h-refinement study ( $N_g = 5$ ) exhibited fifth-order convergence ( $-5.21$ ) with a minimum error of  $L_2 = 3.9 \times 10^{-6}$  at 26 elements per edge ( $N_e = 26$ ) corresponding to 101 400 degrees of freedom.

A p-refinement study with 10 elements per edge ( $N_e = 10$ ) exhibited spectral (exponential) convergence, achieving

a minimum error of  $L_2 = 1.1 \times 10^{-12}$  using twelfth-order polynomials ( $N_g = 13$ ) corresponding to 101 400 degrees of freedom. A p-refinement study with 3 elements per edge ( $N_e = 3$ ) also exhibited spectral convergence, achieving a minimum error of  $L_2 = 3.0 \times 10^{-13}$  using twenty-second-order polynomials ( $N_g = 23$ ) corresponding to only 28 566 degrees of freedom. Using polynomials above the twenty-second order produced no additional error reduction for this mesh.

Figure 5 displays the  $L_2$  errors for all four convergence studies with solid lines denoting h convergence and dashed lines denoting p convergence. For a smooth initial condition, like the Gaussian, it is clear that increasing the polynomial order  $N_g$  is strongly preferred over increasing the number of elements  $N_e$ , as the error is nearly 12 orders of magnitude smaller for the highest-order polynomials compared to the

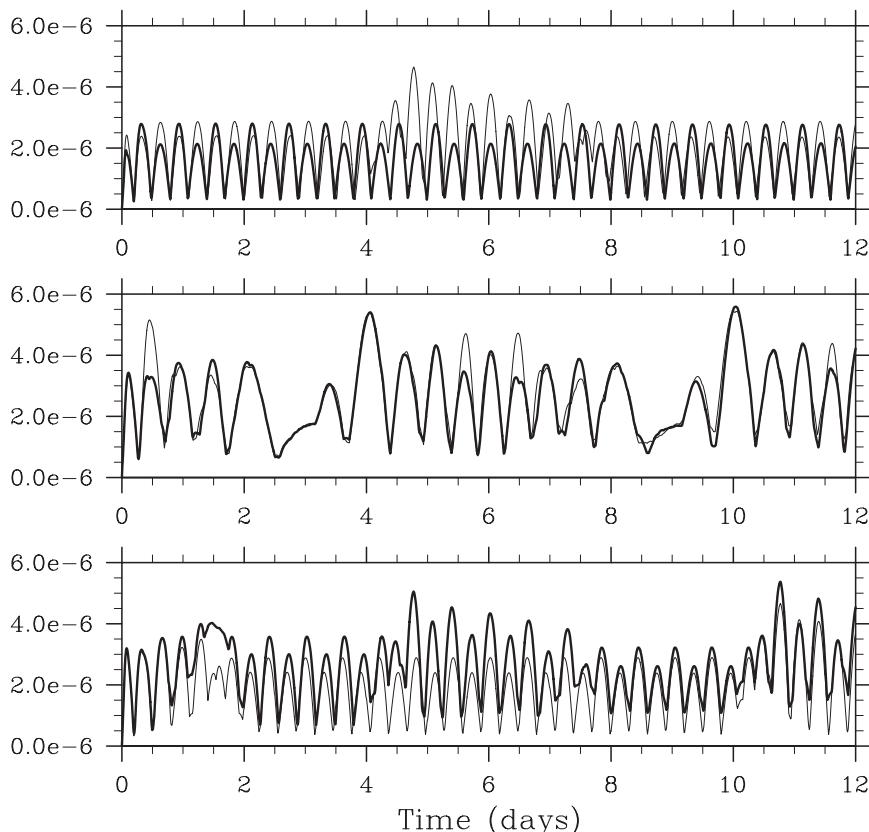


FIG. 4. Comparison of  $L_\infty$  norms on the YY mesh (thin line) vs the YY-P mesh (thick line) at axis angles of (from top to bottom)  $0^\circ$ ,  $45^\circ$ , and  $90^\circ$ . Differences between the two are not large for smooth, well-resolved functions.

lowest-order polynomials for identical degrees of freedom.

### 3) MASS CONSERVATION

Although the transport equation is written in conservative form, the overset technique is not strictly mass conserving without additional constraints. On a conforming mesh, such as an unstructured grid, or the cubed sphere, the numerical flux leaving one element is identical to the numerical flux entering its neighbor element, resulting in mass conservation within the limits of numerical round-off error.

For an overset mesh boundary, this is not the case. The mass flux entering the surface of a Yang element and the flux exiting from a Yin element, for example, are spatially separated by the overlap distance  $\delta$  at a minimum and by  $\delta + 10^\circ$  at the mesh corners. As the mass passes between the two surfaces, it incurs errors due to advection and, to a lesser extent, errors due to numerical interpolation. Therefore, the mass leaving Yin does not strictly match the mass entering Yang.

However, the error incurred is quite small, typically a couple of orders of magnitude smaller than the  $L_2$  norm for the scheme as a whole. As an example, Fig. 6 shows the numerical mass error for the west-east Gaussian advection case on the YY mesh examined earlier with  $N_e = 5$ ,  $N_g = 10$ , giving  $N = 15\,000$  degrees of freedom. While the maximum  $L_2$  error for the simulation is on the order of  $10^{-6}$ , the mass error is on the order of  $10^{-8}$ .

Peng et al. (2006) demonstrated that mass conservation can be restored for an overset finite-volume mesh by applying a constraint locally within each cut-cell (volume intersecting the overset boundary). In principle, it should be possible to construct a similar elementwise constraint for the DG overset boundary as well. While such a constraint may not be necessary for weather prediction applications, it is essential for long-duration climate simulations.

#### b. Solid-body rotation of a cosine bell

The cosine bell test is widely employed to compare the performances of global advection schemes. The initial mass field is

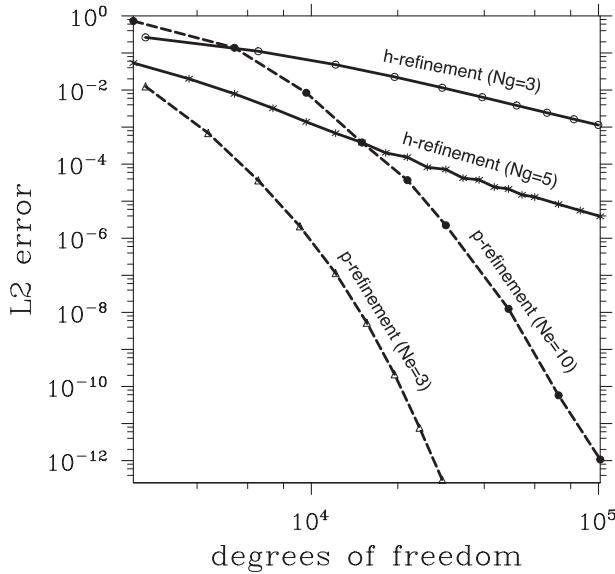


FIG. 5. Numerical error convergence of Gaussian solid-body revolution on the YY grid, under h-type and p-type refinement. The two h-type refinement studies exhibit geometric convergence rates of  $-3.42$  for  $N_g = 3$  and  $-5.21$  for  $N_g = 5$ . The two p-type refinement studies both exhibit spectral convergence, with the  $N_e = 10$  study achieving its minimum error at 101 400 degrees of freedom and the  $N_e = 3$  study achieving its minimum error at only  $N_e = 28\,566$  degrees of freedom.

$$\psi_0 = \begin{cases} \frac{1}{2} \left[ 1 + \cos\left(\pi \frac{d}{d_0}\right) \right] & \text{for } d < d_0, \\ 0 & \text{otherwise} \end{cases}, \quad (33)$$

where  $d = r[\arccos(\mathbf{r} \cdot \mathbf{r}_0/r^2)]$  is the great circle distance between a point on the sphere  $\mathbf{r}$  and the cosine bell center at  $\mathbf{r}_0$  and  $d_0 = R/3$  is the radius of the cosine bell.

As before, the mass field is advected by the velocity field  $\mathbf{v} = \mathbf{r} \times \boldsymbol{\omega}$ , which is rotational with angular velocity  $\boldsymbol{\omega}(\alpha) = (2.0\pi/12 \text{ days})(-\sin\alpha\hat{\mathbf{e}}_x + \cos\alpha\hat{\mathbf{e}}_z)$ , where  $\alpha$  is the axis tilt angle. In contrast with the infinitely differentiable ( $C^\infty$ ) Gaussian considered in the previous test, the cosine bell is discontinuous in its second derivative and all higher derivatives ( $C^1$  continuous), which is known to limit the convergence rate under uniform grid refinement.

To facilitate comparison with other schemes, simulations were performed at a resolution of  $N_e = 4$  and  $N_g = 8$  representing 6144 degrees of freedom. This is equivalent to the  $(32 \times 96 \times 2)$  finite-volume grid used in finite-volume Yin–Yang studies such as Li et al. (2008). Rotational angles of  $\alpha = 0^\circ$  and  $45^\circ$  were analyzed on both the YY and YY-P grids, and the CFL number was fixed at 0.5.

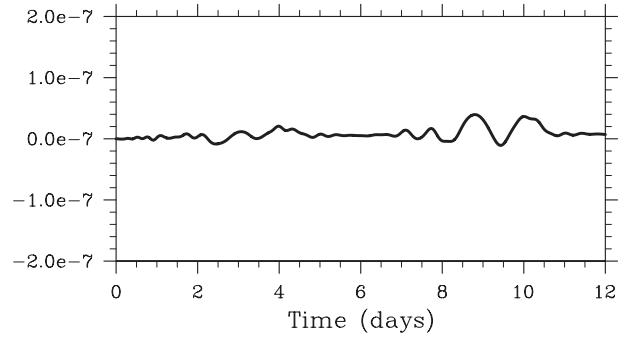


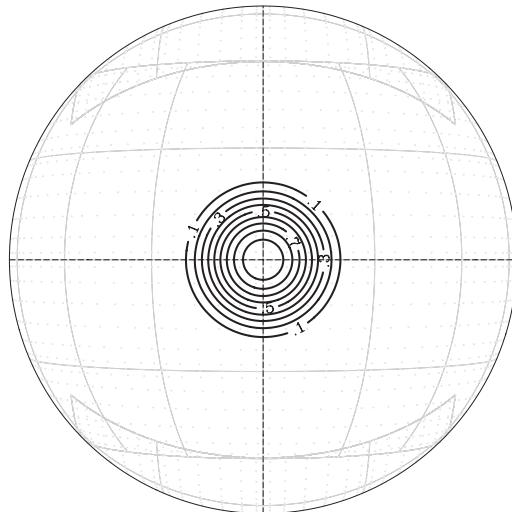
FIG. 6. Normalized mass error for Gaussian revolution for  $N_e = 5$ ,  $N_g = 10$ ,  $\alpha = 45$ . Mass is not strictly conserved without an additional constraint, but the errors are small, with a maximum error roughly two orders of magnitude smaller than the  $L_2$  norm.

### 1) NUMERICAL RESULTS

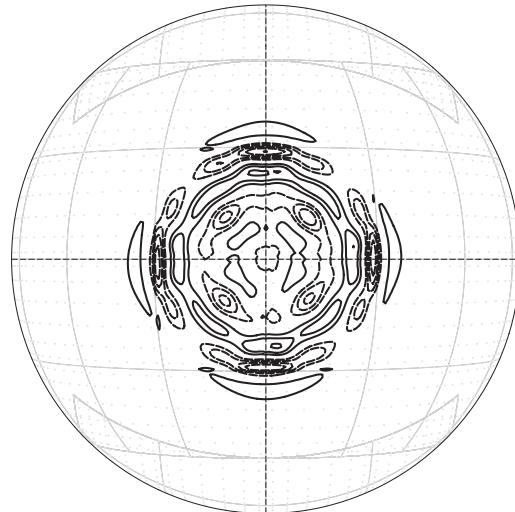
Normalized errors for both the west–east ( $\alpha = 0^\circ$ ) and southwest–northeast ( $\alpha = 45^\circ$ ) trajectories on the YY mesh are plotted in Figs. 7c,d, demonstrating that the scheme succeeds in transporting the cosine bell scalar field about the globe with little distortion. After one revolution, the numerical solution closely matches the exact solution, and has maximum error norms of  $L_1 = 0.030$ ,  $L_2 = 0.014$ , and  $L_\infty = 0.012$ . Comparison with other global advection schemes shows this method to be very competitive, producing smaller error norms for a fixed number of degrees of freedom. For a detailed comparison, please refer to Table 4 in Li et al. (2008).

The same tests were applied to the YY-P grid as illustrated by Figs. 7e,f. The  $\alpha = 0$  case shows improved performance on this mesh, as anticipated. The cosine bell completes the circuit without crossing any overset boundaries and as a result the signal is somewhat smoother, although the overall order of magnitude is unchanged. For this case, the maximum error norms are  $L_1 = 0.021$ ,  $L_2 = 0.013$ , and  $L_\infty = 0.012$ , which represents a small improvement over the YY grid measurements. The  $\alpha = 45^\circ$  test on the YY-P mesh, on the other hand, produces error norms that are nearly indistinguishable from those of the YY mesh, indicating that neither mesh has a distinct advantage over the other if the transport is not predominantly zonal or polar.

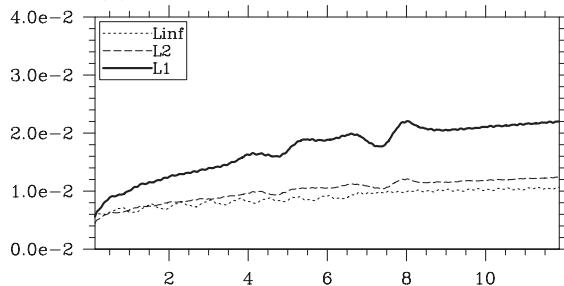
The numerical solution and numerical error field (difference from the analytical solution) are plotted in Figs. 7a,b for  $\alpha = 45^\circ$  on the YY grid. Error contours range from  $-0.025$  to  $0.025$  in  $0.018\,75$  increments where solid lines represent positive values and dashed contours represent negative values. We observe that although small, the contours are somewhat oscillatory, which is characteristic of high-order spectral methods. No filters or slope limiters were applied.

(a) Numerical results,  $t = 1$  day

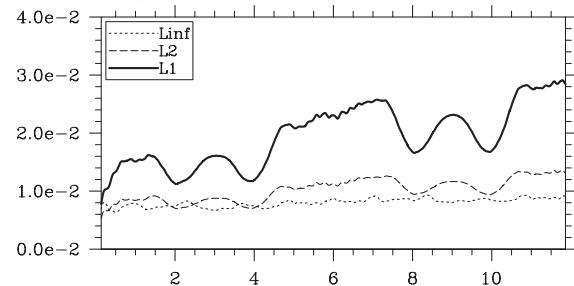
(b) Difference from analytic solution



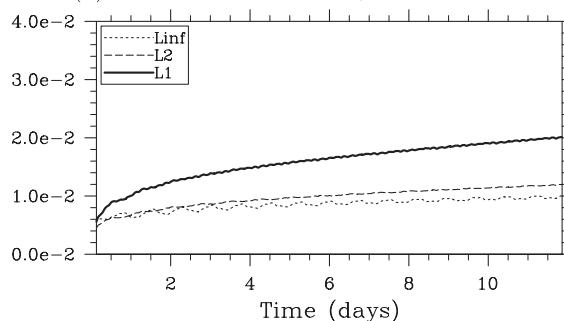
(c) Error norms YY, W-E



(d) Error norms YY, SW-NE



(e) Error norms YY-P, W-E



(f) Error norms YY-P, SW-NE

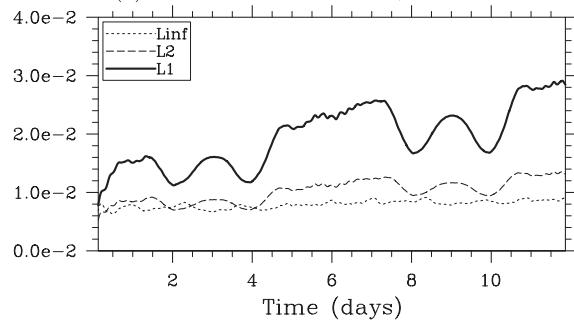


FIG. 7. Cosine bell transport results with  $N_e = 4$  elements and  $N_g = 8$  nodes per element (6144 degrees of freedom). (a) Numerical results after one revolution for the southwest–northeast,  $\alpha = 45^\circ$  case on the YY mesh show no visible distortion. Contours range from 0.1 to 0.9 at intervals of 0.1. (b) Difference from analytic solution with error contours from  $-0.025$  to  $0.025$  by  $0.01875$ . Dashed contours are negative. Error norms for (c) west–east revolution on the YY mesh, (d) southwest–northeast revolution on the YY mesh, (e) west–east revolution on the YY-P mesh, and (f) southwest–northeast revolution on the YY-P mesh.

## 2) CONVERGENCE STUDIES

For the case in question, with 32 points per mesh edge, there is considerable flexibility in how the degrees of freedom are divided between  $N_e$  and  $N_g$ . The choices range from 32 elements with 1 grid point to a single element with 32 grid points. The Gaussian test suggests

that high-order elements are preferable over a high element count. That conclusion remains valid for the cosine bell case as well, but to a lesser extent. For polynomial orders of  $N_g = \{2, 4, 8, 16, 32\}$ , simulations produce errors of  $L_2 = \{0.87, 0.062, 0.014, 0.010, 0.0087\}$ . However, the extra computational time required for the highest-order  $N_g = 32$  case does not

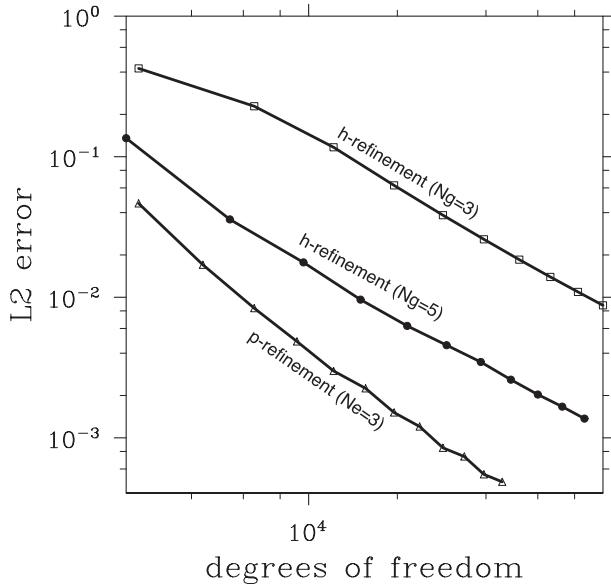


FIG. 8. Error convergence of the cosine bell test on the YY grid under h-type and p-type mesh refinement. The h-type refinement studies shows geometric convergence with orders of  $-2.3$  for  $N_g = 3$  elements per edge and  $-2.5$  for  $N_g = 5$ . The p-type refinement study also produces geometric convergence due to  $C^1$  continuity of the cosine bell with a convergence order of  $-3.2$ .

justify the modest error reduction, and the  $N_g = 16$  or  $N_g = 8$  options are preferable in practice (for a  $C^1$  continuous function). In general, the best choice of polynomial order will depend upon the problem being solved and the amount of time available.

To quantify the error convergence in greater detail, h-refinement studies were performed on the YY mesh at fixed polynomial orders of  $N_g = 3$  and  $N_g = 5$ . A p-refinement study was also performed with a fixed element count of  $N_e = 3$ . The  $L_2$  errors are plotted on a log-log scale in Fig. 8, and linear regression reveals that the  $N_g = 3$  h-refinement exhibits a slope of  $-2.3$  while the  $N_g = 5$  h-refinement produces a slope of  $-2.5$ . In contrast with the Gaussian advection test, the p-refinement study does not exhibit spectral convergence. Instead it exhibits geometric convergence with a measured slope of  $-3.2$ .

From this, one might conclude that low-order polynomials are just as good as high-order polynomials, as convergence is geometric for both. However, the error plots in Fig. 8 reveal that higher-order elements are still the better choice (within the limits of computational time), as they produce results with an absolute error that is several orders of magnitude smaller than that produced by the lower-order elements.

*c. Moving-vortices test*

The third test is the challenging moving-vortex test proposed by Nair and Jablonowski (2008), intended to

address a more realistic atmospheric flow. In this test, a hyperbolic initial condition, dividing the globe into two regions, is subjected to a vortical velocity field representing an idealized cyclogenesis flow. The cyclonic velocity field is then superimposed upon the solid-body velocity field described earlier, resulting in a pair of vortices that revolve about the globe as they rotate.

The test represents the roll-up of an idealized moving atmospheric vortex such as a hurricane or tropical cyclone. As with the previous tests, its exact solution is known at all times, making it popular for error assessment and convergence studies on arbitrary structured and unstructured grids as demonstrated by Flyer and Lehto (2010) and Li and Xiao (2010).

The dual vortices are located at the two opposing poles of the axis  $\hat{\mathbf{e}}_z^*(t)$  in a rotated coordinate system  $\lambda^* = (\lambda^*, \theta^*)$ , and arise from the velocity field

$$\mathbf{v}^* = v_0 \frac{3\sqrt{3}}{2} \tanh(\rho) \operatorname{sech}^2(\rho), \tag{34}$$

which is a function of the distance parameter  $\rho^* = (r/a_0) \cos\theta^*$ . The parameter  $a_0$  sets the characteristic size of the cyclone and  $v_0$  sets its rotational speed. The velocity field is directed longitudinally in the rotated coordinate system, along the vector  $\hat{\mathbf{e}}_\lambda^* = \hat{\mathbf{e}}_z^* \times \hat{\mathbf{r}}$ . The associated angular velocity is defined to be

$$\omega^*(\theta^*) = \begin{cases} v^*/\rho^* & \text{if } \rho^* \neq 0 \\ 0 & \text{if } \rho^* = 0 \end{cases}, \tag{35}$$

where  $\rho^* = r \cos\theta^*$  is the distance to the  $\mathbf{e}_z^*$  axis. The analytical solution in rotated coordinates is

$$\psi(\lambda^*, \theta^*, t) = 1 - \tanh\left[\frac{\rho}{\gamma} \sin(\lambda^* - \omega^* t)\right], \tag{36}$$

where  $\gamma$  sets the transition width, defining the sharpness of the interface between the two regions at  $t = 0$ . The total velocity field is found by summing the cyclonic field with the solid-body rotation field,

$$\mathbf{v} = \mathbf{v}^* + (\boldsymbol{\omega} \times \mathbf{r}), \tag{37}$$

where the orbital angular speed is  $\omega = 2\pi/(12 \text{ days})$ . The following numerical experiments employ a cyclone of diameter  $a_0 = R/3$ , a velocity scale factor  $v_0 = 1/6$ , and a transition width  $\gamma = 5R$ , where  $R$  is the radius of the earth.

A resolution of  $N_e = 5$  elements per mesh edge and  $N_g = 8$  grid points per element is chosen to match the resolution of the experiments in Nair and Jablonowski (2008). This setup has 9600 degrees of freedom, giving

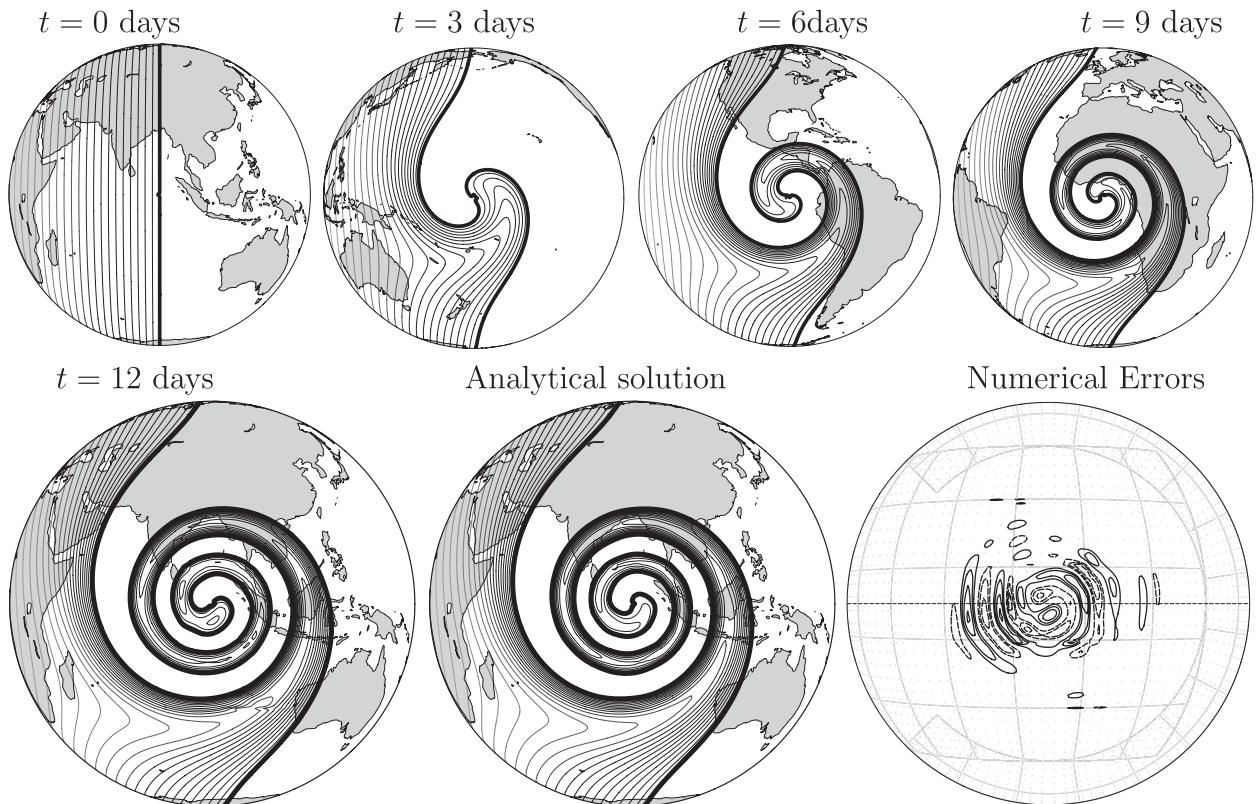


FIG. 9. (top) Numerical solution for the west–east  $\alpha = 0$  moving-vortices test at a resolution of  $N_e = 5$ ,  $N_g = 8$  (average of  $2.25^\circ$  per degree of freedom) on the YY mesh. Contours are sampled at (top) (from left to right) 0, 3, 6, and 9 days. (bottom) (from left to right) The final numerical solution at 12 days closely matches the analytical solution with the difference field shown to the right. Contours range from  $-0.05$  to  $0.05$  with an interval of  $0.0111$ .

an average resolution of  $2.6^\circ \times 2.6^\circ$  per node. With the CFL number fixed at 0.5, each simulation took just over 1300 steps to complete (using the eighth-order Runge–Kutta scheme), which corresponds to roughly 800 s per step.

Figure 9 shows the evolution of the numerical solution for the  $\alpha = 0$  west–east case on the YY grid as a series of orthographic projections centered on one of the vortices. The numerical results, plotted at  $t = 0, 3, 6, 9,$  and  $12$  days, are visually indistinguishable from the analytic solution. The last frame in the figure plots the numerical error with contours from 0.5 to 1.5 by 0.11, exhibiting errors that are aligned primarily in the orbital direction.

The numerical solution is compared to the analytical solution and the time-dependent error measures  $L_1$ ,  $L_2$ , and  $L_\infty$  are plotted in Figs. 10a,b for  $\alpha = 0^\circ$  and  $\alpha = 45^\circ$  orbits. The error norms for the  $\alpha = 0^\circ$  case look remarkably similar to those produced by the cubed-sphere DG scheme in Nair and Jablonowski (2008), with maximum error values of  $L_1 = 1.9 \times 10^{-3}$ ,  $L_2 = 5.8 \times 10^{-3}$ , and  $L_\infty = 5.2 \times 10^{-2}$ . However, in contrast with the cubed-sphere DG scheme, the Yin–Yang DG

scheme performs equally well at  $45^\circ$  with maximum normalized errors of  $L_1 = 1.8 \times 10^{-3}$ ,  $L_2 = 5.6 \times 10^{-3}$ , and  $L_\infty = 5.7 \times 10^{-2}$ .

Examining the  $\alpha = 45^\circ$  case in greater detail, Fig. 11 displays contour plots of the numerical results at 3, 6, 9, and 12 days. It is clear from the figure that the vortices are well resolved at each time (analytic solutions are not plotted, as they are visually indistinguishable from the numerical results). There is no noticeable signal due to the presence of the Yin and Yang oversight boundaries.

Both tests were repeated on the YY-P mesh, the results of which are plotted in Figs. 10c,d. In each case, the scalar fields were visually indistinguishable from the YY results and the error norms were very similar as well. The  $\alpha = 0^\circ$  case produced maximum error norms of  $L_1 = 1.9 \times 10^{-3}$ ,  $L_2 = 5.8 \times 10^{-3}$ , and  $L_\infty = 4.7 \times 10^{-2}$ , which is a bit better than the corresponding YY case, while the  $\alpha = 45^\circ$  case produced error norms of  $L_1 = 1.8 \times 10^{-3}$ ,  $L_2 = 5.8 \times 10^{-3}$ , and  $L_\infty = 6.4 \times 10^{-2}$ , which is marginally noisier than the corresponding YY case. In contrast with the  $\alpha = 0$  cosine bell test, the moving-vortex test did not benefit significantly from the improved  $\mathcal{Y}_E$  equatorial

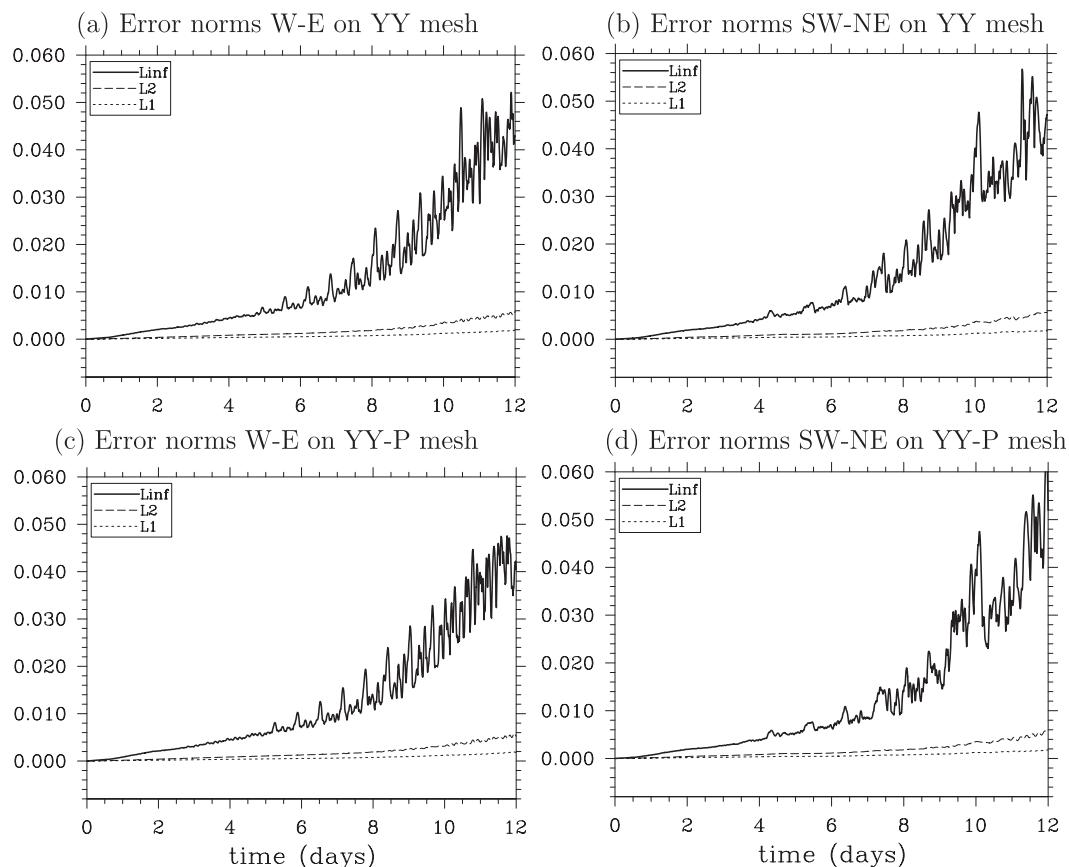


FIG. 10. Error norms for the moving-vortices test. The error norms for (a) west–east flow  $\alpha = 0$  on the YY mesh are very similar to those for (b) southwest–northeast flow  $\alpha = 45^\circ$  in contrast with DG on the cubed sphere (which shows increased error at  $45^\circ$ ). (c) West–east flow on the YY-P mesh. (d) Southwest–northeast flow on the YY-P mesh.

Yin grid, as the cyclonic field is global in extent and the wind fields are not strictly zonal.

### 7. Conclusions

We have presented an implementation of the Yin–Yang overset grid in a nodal discontinuous Galerkin (DG) setting. In comparison with finite-volume and finite-difference implementations of the Yin–Yang grid, the discontinuous Galerkin approach is considerably simpler, as the overset interpolation is local, requiring information from the interior of a single element and avoiding the complex halo-cell construction needed for high-order interpolation in those schemes. The DG implementation benefits from the interpolation scheme provided by its polynomial representation, which is of high order and accurate.

The method was subjected to a series of standard benchmarks used to compare global transport schemes including solid-body rotation of a smooth Gaussian field, solid-body rotation of a ( $C^1$  continuous) cosine bell

curve, and the challenging moving-vortex test representing an idealized cyclone. In each case, the scheme performed very well, producing results of equivalent or greater accuracy when compared with other global advection schemes. For smooth initial conditions, the method exhibited spectral convergence under p-type mesh refinement, reaching an error norm of  $L_2 = 3 \times 10^{-13}$  using fewer than 29 000 degrees of freedom. When coupled with an appropriately high-order explicit Runge–Kutta scheme, CFL numbers as large as 1 may be employed and the scheme is quite fast even for high-accuracy simulations.

In comparison with the DG cubed sphere, the Yin–Yang overset method has the advantage that it comprises fewer regions, minimizing the number of boundary crossings, and in some cases (particularly at  $45^\circ$ ) the Yin–Yang method showed smaller error norms. The coordinate systems are orthogonal in both component meshes, resulting in a simpler representation of some equations, whereas the cubed-sphere coordinate systems are nonorthogonal. Also, postprocessing is simplified by the familiar RLL structure of the mesh components.

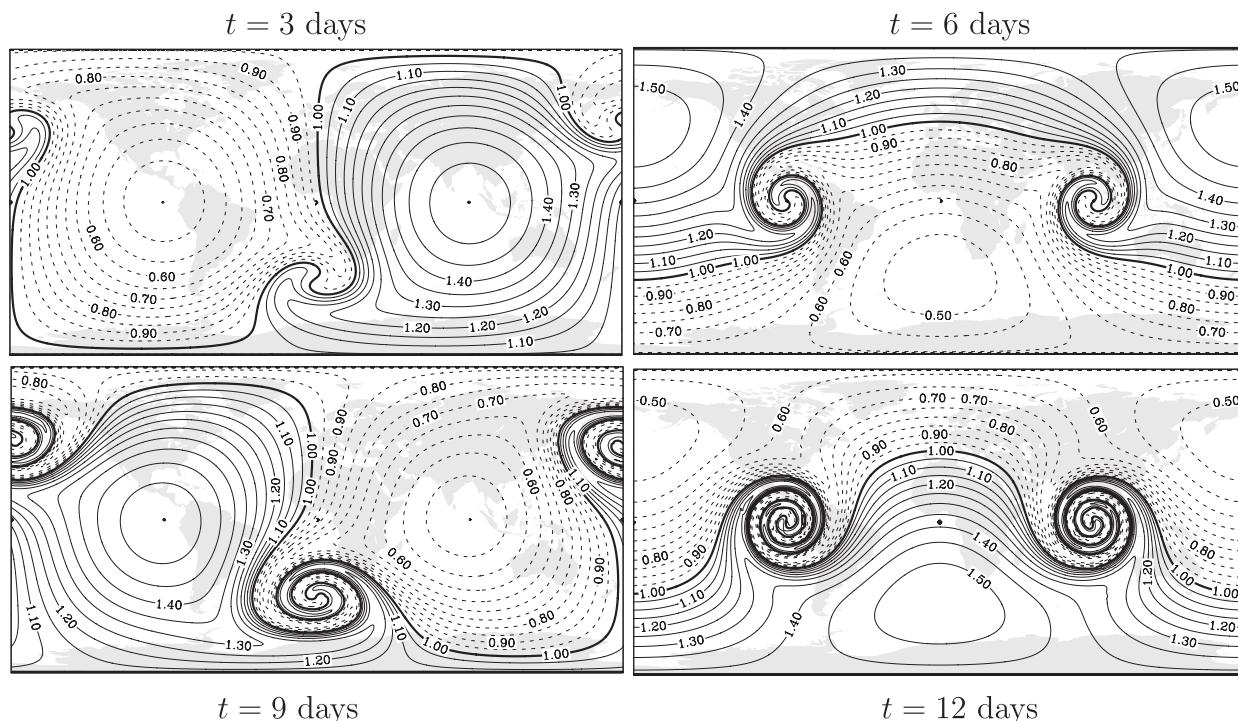


FIG. 11. Contour plots for the numerical solution of the southwest–northeast ( $\alpha = 45$ ) moving-vortices test at  $t =$  (from top left clockwise to bottom left) 3, 6, 9, and 12 days on the YY mesh. Notice the vortex is well resolved at the  $2.25^\circ$  resolution ( $N_e = 5, N_g = 8$ ).

Overall, the implementation complexity of the DG Yin–Yang and DG cubed-sphere methods is quite similar.

In addition to the familiar Yin–Yang (YY) mesh of Kageyama and Sato (2004), we also examined a modified Yin–Yang polar (YY-P) variant discussed in Staniforth and Thuburn (2012). In general we found that the modified YY-P and YY meshes exhibited similar performances on most tests, while the YY-P mesh performed slightly better on cases with strictly zonal flow, such as west–east solid-body rotation of a cosine bell.

For both meshes, the most straightforward implementation of the overset boundary comes at the price of losing the strict mass conservation inherent to the conservative DG formulation. The flux leaving the Yin grid is spatially separated from the flux entering the Yang grid and the two are not guaranteed to exactly cancel each other without additional constraints. However, this problem is common to overset methods, and the recent work of Peng et al. (2006) demonstrated that a local constraint was able to restore mass conservation in finite-volume simulations. We believe that it should be reasonably straightforward to modify Peng’s conservative constraint to restore strict mass conservation to the DG Yin–Yang method as well, which is something we plan to examine in the near future.

*Acknowledgments.* The present study was carried out within the scope of the project Collaborative Research:

A Multiscale Unified Simulation Environment for Geoscientific Applications, which is funded by the National Science Foundation under Peta-Apps Grant 0904599.

## APPENDIX

### Yin, Yang, and Cartesian Transforms

At times, it is necessary or convenient to transform quantities between the Yin, Yang, and Cartesian coordinate systems. In this section, we derive transformations from Yin to Cartesian and from Yang to Cartesian, and then compose them to obtain the transformations from Yin to Yang and vice versa.

#### a. Yin–Cartesian transform

The Yin coordinates are the RLL coordinates  $\boldsymbol{\lambda} = (\lambda, \theta)$  with longitude  $\lambda$ , latitude  $\theta$ , and radial distance  $r$  (constant on the sphere). The transformation from a point in Yin coordinates  $\mathbf{r}(\lambda, \theta)$  to a point in Cartesian coordinates  $\mathbf{r}(x, y, z)$  is given by

$$\begin{aligned} x(\lambda, \theta) &= r \cos \lambda \cos \theta \\ y(\lambda, \theta) &= r \sin \lambda \cos \theta \\ z(\lambda, \theta) &= r \sin \theta, \end{aligned} \quad (\text{A1})$$

where the  $z$  axis points toward the North Pole and the  $x$  axis point toward the prime meridian. The inverse transformation is

$$\theta = \sin^{-1}(z/r), \tag{A2}$$

$$\lambda = \tan^{-1}(y/x). \tag{A3}$$

The velocity  $\mathbf{v}$  at point  $\mathbf{r}$  may be expressed in curvilinear coordinates by employing the chain rule

$$\mathbf{v} = \frac{d\mathbf{r}(\lambda, \theta)}{dt} = \dot{\lambda} \frac{\partial \mathbf{r}}{\partial \lambda} + \dot{\theta} \frac{\partial \mathbf{r}}{\partial \theta}. \tag{A4}$$

From this expression, the covariant basis vectors are identified as  $\mathbf{e}_\lambda = \partial \mathbf{r} / \partial \lambda$  and  $\mathbf{e}_\theta = \partial \mathbf{r} / \partial \theta$  and the contravariant velocity components are  $v^\lambda = \dot{\lambda}$  and  $v^\theta = \dot{\theta}$ , such that

$$\mathbf{v} = \dot{\lambda} \mathbf{e}_\lambda + \dot{\theta} \mathbf{e}_\theta. \tag{A5}$$

Written in matrix form this equation is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \phi} \\ \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \phi} \end{bmatrix} \begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \end{bmatrix}, \tag{A6}$$

where the  $3 \times 2$  matrix is the Jacobian  $\partial \mathbf{x} / \partial \boldsymbol{\lambda}$  of the transformation to Cartesian coordinates from Yin coordinates whose columns are the covariant bases  $\mathbf{e}_\lambda = \partial \mathbf{r} / \partial \lambda$  and  $\mathbf{e}_\theta = \partial \mathbf{r} / \partial \theta$ .

Repeating the process for the curvilinear velocity components gives

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial \lambda}{\partial x} & \frac{\partial \lambda}{\partial y} & \frac{\partial \lambda}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{A7}$$

where the matrix is the Jacobian  $\partial \boldsymbol{\lambda} / \partial \mathbf{x}$  of the transformation to Yin coordinates from Cartesian coordinates whose row vectors are the contravariant bases  $\mathbf{e}^\lambda = \nabla \lambda$  and  $\mathbf{e}^\theta = \nabla \theta$ .

The Yin–Cartesian Jacobian may be easily evaluated using equation set (A1):

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} -r \sin \lambda \cos \theta & -r \cos \lambda \sin \theta \\ r \cos \lambda \cos \theta & -r \sin \lambda \sin \theta \\ 0 & r \cos \theta \end{bmatrix}, \tag{A8}$$

while the inverse Jacobian is most easily obtained by first calculating the contravariant bases.

The covariant metric components are  $g_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j$ , which is equivalent to the matrix product

$$g_{ij} = \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}} \right)^T \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} r^2 \cos^2 \theta & 0 \\ 0 & r^2 \end{bmatrix}. \tag{A9}$$

The Ying coordinates are orthogonal, so the cross terms in the metric are zero. The determinant of the metric is  $g = \det(g_{ij}) = r^4 \cos^2 \theta$ , such that  $\sqrt{g} = r^2 \cos \theta$  (a term that appears in the advective transport equation).

The contravariant metric  $g^{ij} = \mathbf{e}^i \cdot \mathbf{e}^j$  is the inverse of the covariant metric  $g^{ij} = (g_{ij})^{-1}$ :

$$g^{ij} = \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \left[ \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \right]^T = \begin{bmatrix} \frac{1}{r^2 \cos^2 \theta} & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix}, \tag{A10}$$

defined for  $\cos \theta \neq 0$  and  $r \neq 0$ . The contravariant basis vectors may now be extracted by application of the contravariant metric  $\mathbf{e}^i = \sum_j g^{ij} \mathbf{e}_j$ , such that  $\mathbf{e}^\lambda = \mathbf{e}_\lambda / (r^2 \cos^2 \theta)$  and  $\mathbf{e}^\theta = \mathbf{e}_\theta / r^2$ . Assembling the contravariant bases as row vectors gives the inverse Jacobian matrix

$$\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} = \begin{bmatrix} -\frac{\sin \lambda}{r \cos \theta} & \frac{\cos \lambda}{r \cos \theta} & 0 \\ -\frac{\cos \lambda \sin \theta}{r} & -\frac{\sin \lambda \sin \theta}{r} & \frac{\cos \theta}{r} \end{bmatrix}. \tag{A11}$$

Application of the Jacobian and inverse Jacobian matrices allows one to readily convert the velocity from Yin to Cartesian coordinates and back again. Note, as a check, it is easily verified that the product of the two transformation matrices gives the identity matrix

$$\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}} = \mathbf{I}. \tag{A12}$$

The wind speeds ( $u, v$ ) are defined in Yin coordinates such that  $\mathbf{v} = u \hat{\mathbf{e}}_\lambda + v \hat{\mathbf{e}}_\theta$ , and as such are related to the contravariant components by  $u = \dot{\lambda} |\mathbf{e}_\lambda| = \dot{\lambda} (r \cos \theta)$  and  $v = \dot{\theta} |\mathbf{e}_\theta| = \dot{\theta} r$ .

*b. Yang–Cartesian transform*

The Yang coordinate system  $\boldsymbol{\lambda}' = (\lambda', \theta')$  is an RLL system that has been rotated in Cartesian space such that  $(x', y', z') = (-x, z, y)$ . The transform to Cartesian components is

$$x(\lambda', \theta') = -r \cos \lambda' \cos \theta', \tag{A13}$$

$$y(\lambda', \theta') = r \sin \theta', \tag{A14}$$

$$z(\lambda', \theta') = r \sin \lambda' \cos \theta', \quad (\text{A15})$$

and the inverse transformation is

$$\theta' = \sin^{-1}(y/r), \quad (\text{A16})$$

$$\lambda' = \tan^{-1}(-z/x). \quad (\text{A17})$$

Following the procedure outlined in the previous section, the Jacobian of the transformation is found to be

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}'} = \begin{bmatrix} r \sin \lambda' \cos \theta' & r \cos \lambda' \sin \theta' \\ 0 & r \cos \theta' \\ r \cos \lambda' \cos \theta' & -r \sin \lambda' \sin \theta' \end{bmatrix}. \quad (\text{A18})$$

The covariant metric is

$$g'_{ij} = \left[ \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}'} \right]^T \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}'} = \begin{bmatrix} r^2 \cos^2 \theta' & 0 \\ 0 & r^2 \end{bmatrix} \quad (\text{A19})$$

with determinant  $g' = \det(g'_{ij}) = R^2 \cos^2 \theta'$ . The contravariant metric is

$$(g')^{ij} = (g'_{ij})^{-1} = \begin{bmatrix} \frac{1}{r^2 \cos^2 \theta'} & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix} \quad (\text{A20})$$

and the Jacobian of the inverse transformation is

$$\frac{\partial \boldsymbol{\lambda}'}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\sin \lambda'}{r \cos \theta'} & 0 & \frac{\cos \lambda'}{r \cos \theta'} \\ \frac{\cos \lambda' \sin \theta'}{r} & \frac{\cos \theta'}{r} & -\frac{\sin \lambda' \sin \theta'}{r} \end{bmatrix}. \quad (\text{A21})$$

### c. Yin–Yang transform

The Yin–Cartesian and Yang–Cartesian transforms may be composed to give direct transformations between the Yin and Yang coordinate systems. The coordinate transformation from Yang to Yin coordinates (found by equating  $x$ ,  $y$ , and  $z$ ) is

$$\theta = \arcsin(\sin \lambda' \cos \theta'), \quad (\text{A22})$$

$$\lambda = \arctan(-\tan \theta' / \cos \lambda'), \quad (\text{A23})$$

and the coordinate transformation from Yin to Yang is

$$\theta' = \arcsin(\sin \lambda \cos \theta), \quad (\text{A24})$$

$$\lambda' = \arctan(-\tan \theta / \cos \lambda). \quad (\text{A25})$$

The Jacobian transformation from Yang to Yin is found by multiplying the forward and inverse Jacobians

$$\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\lambda}'} = \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}'}, \quad (\text{A26})$$

and the Jacobian transformation matrix from Yin to Yang is obtained from the matrix product

$$\frac{\partial \boldsymbol{\lambda}'}{\partial \boldsymbol{\lambda}} = \frac{\partial \boldsymbol{\lambda}'}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}}. \quad (\text{A27})$$

### d. Reference to curvilinear transforms

In the DG formulation, integrals over the elements are approximated by Gaussian numerical quadrature. To perform the quadrature, the area of integration must first be mapped into the reference coordinate system  $\boldsymbol{\xi} = (\xi, \eta)$ , where  $\xi \in [-1, 1]$  and  $\eta \in [-1, 1]$ .

Elements of the Yin–Yang grid are rectangular in  $(\lambda, \theta)$  coordinates, and the transformation from reference to curvilinear coordinates is a simple linear interpolation,

$$\lambda = \lambda_i(1 - \xi)/2 + \lambda_{i+1}(1 + \xi)/2, \quad (\text{A28})$$

$$\theta = \theta_i(1 - \eta)/2 + \theta_{i+1}(1 + \eta)/2, \quad (\text{A29})$$

but in general the relationship may be more complex (e.g., for an unstructured mesh). The inverse coordinate transformation is

$$\xi = 2(\lambda - \bar{\lambda})/\Delta\lambda, \quad (\text{A30})$$

$$\eta = 2(\theta - \bar{\theta})/\Delta\theta, \quad (\text{A31})$$

where  $\bar{\lambda} = (1/2)(\lambda_{i+1} + \lambda_i)$  and  $\Delta\lambda = \lambda_{i+1} - \lambda_i$ . The Jacobian of the transformation is

$$\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \frac{\partial \lambda}{\partial \xi} & \frac{\partial \lambda}{\partial \eta} \\ \frac{\partial \theta}{\partial \xi} & \frac{\partial \theta}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \Delta\lambda/2 & 0 \\ 0 & \Delta\theta/2 \end{bmatrix} \quad (\text{A32})$$

and the inverse Jacobian is

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} \frac{\partial \xi}{\partial \lambda} & \frac{\partial \xi}{\partial \theta} \\ \frac{\partial \eta}{\partial \lambda} & \frac{\partial \eta}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 2/\Delta\lambda & 0 \\ 0 & 2/\Delta\theta \end{bmatrix}. \quad (\text{A33})$$

Specifically, this means that the velocity components transform as  $\dot{\boldsymbol{\xi}} = (2/\Delta\lambda)\dot{\boldsymbol{\lambda}}$  and  $\dot{\boldsymbol{\eta}} = (2/\Delta\theta)\dot{\boldsymbol{\theta}}$ .

### e. Transformation composition

In general, the transformation from reference to Cartesian coordinates  $\mathbf{r}(\xi, \eta)$  may be obtained by functional composition  $\mathbf{r}[\lambda(\xi, \eta), \theta(\xi, \eta)]$ . Rather than calculating the

Jacobian and metric terms for each transformation from scratch, it is simpler to combine them:

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\lambda}} \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\xi}}, \quad (\text{A34})$$

The metric of the composite transformation is

$$g(\mathbf{x}, \boldsymbol{\xi})_{ij} = \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)^T \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \quad (\text{A35})$$

$$= \left( \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\xi}} \right)^T g(\mathbf{x}, \boldsymbol{\lambda}) \left( \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\xi}} \right), \quad (\text{A36})$$

and the determinant of the metric is the product of the determinants,

$$g_r = \det g(\mathbf{x}, \boldsymbol{\xi}) = \det g(\boldsymbol{\lambda}, \boldsymbol{\xi}) \det g(\mathbf{x}, \boldsymbol{\lambda}), \quad (\text{A37})$$

such that

$$\sqrt{g_r} = \left| \frac{\Delta \lambda \Delta \theta}{4} \right| r^2 \cos \theta. \quad (\text{A38})$$

REFERENCES

Baba, Y., K. Takahashi, T. Sugimura, and K. Goto, 2010: Dynamical core of an atmospheric general circulation model on a Yin–Yang grid. *Mon. Wea. Rev.*, **138**, 3988–4005.

Blaise, S., and A. St-Cyr, 2012: A dynamic *hp*-adaptive discontinuous Galerkin method for shallow-water flows on the sphere with application to a global tsunami simulation. *Mon. Wea. Rev.*, **140**, 978–996.

Cockburn, B., and C.-W. Shu, 1989: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comput.*, **52**, 411–435.

—, and —, 2001: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, **16**, 173–261, doi:10.1023/A:1012873910884.

Flyer, N., and E. Lehto, 2010: Rotational transport on a sphere: Local node refinement with radial basis functions. *J. Comput. Phys.*, **229**, 1954–1969.

Gassner, G., and D. A. Kopriva, 2011: A comparison of the dispersion and dissipation errors of Gauss and Gauss–Lobatto discontinuous Galerkin spectral element methods. *SIAM J. Sci. Comput.*, **33** (5), 2560–2579, doi:10.1137/100807211.

Giraldo, F. X., 1997: Lagrange–Galerkin methods on spherical geodesic grids. *J. Comput. Phys.*, **136**, 197–213, doi:10.1006/jcph.1997.5771.

—, J. Hesthaven, and T. Warburton, 2002: Nodal high-order discontinuous Galerkin methods for spherical shallow water equations. *J. Comput. Phys.*, **181**, 499–525.

Gottlieb, S., C.-W. Shu, and E. Tadmor, 2001: Strong stability-preserving high-order time discretization methods. *SIAM Rev.*, **43** (1), 89–112.

Ii, S., and F. Xiao, 2010: A global shallow water model using high order multi-moment constrained finite volume method and

icosahedral grid. *J. Comput. Phys.*, **229**, 1774–1796, doi:10.1016/j.jcp.2009.11.008.

Kageyama, A., and T. Sato, 2004: “Yin–Yang grid”: An overset grid in spherical geometry. *Geochem. Geophys. Geosyst.*, **5**, Q09005, doi:10.1029/2004GC000734.

Karniadakis, G. E., and S. J. Sherwin, 2005: *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scientific Computation Series, Oxford University Press, 657 pp.

Kopriva, D., 2009: *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Springer, 411 pp.

—, and G. Gassner, 2010: On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *J. Sci. Comput.*, **44**, 136–155, doi:10.1007/s10915-010-9372-3.

Levy, M. N., R. D. Nair, and H. M. Tufo, 2007: High-order Galerkin method for scalable global atmospheric models. *Comput. Geosci.*, **33**, 1022–1035.

Li, X., D. Chen, X. Peng, F. Xiao, and X. Chen, 2006: Implementation of the semi-Lagrangian advection scheme on a quasi-uniform overset grid on a sphere. *Adv. Atmos. Sci.*, **23**, 792–801, doi:10.1007/s00376-006-0792-9.

—, —, —, K. Takahashi, and F. Xiao, 2008: A multimoment finite-volume shallow-water model on the Yin–Yang overset spherical grid. *Mon. Wea. Rev.*, **136**, 3066–3086.

Nair, R. D., and C. Jablonowski, 2008: Moving vortices on the sphere: A test case for horizontal advection problems. *Mon. Wea. Rev.*, **136**, 699–711.

—, S. J. Thomas, and R. D. Loft, 2005: A discontinuous Galerkin transport scheme on the cubed sphere. *Mon. Wea. Rev.*, **133**, 814–828.

—, M. N. Levy, and P. H. Lauritzen, 2011: Emerging numerical methods for atmospheric modeling. *Numerical Techniques for Global Atmospheric Models*, P. H. Lauritzen et al., Eds., Lecture Notes in Computational Science and Engineering Series, Vol. 80, Springer, 251–311.

Nastase, C., D. Mavriplis, and J. Sitaraman, 2011: An overset unstructured mesh discontinuous Galerkin approach for aerodynamic problems. *Proc. 49th Aerospace Sciences Meeting and Exhibit*, Orlando, FL, AIAA, 195.

Peng, X., F. Xiao, and K. Takahashi, 2006: Conservative constraint for a quasi-uniform overset grid on the sphere. *Quart. J. Roy. Meteor. Soc.*, **132**, 979–996, doi:10.1256/qj.05.18.

Phillips, N. A., 1959: Numerical integration of the primitive equations on the hemisphere. *Mon. Wea. Rev.*, **87**, 333–345.

Purser, R. J., 2004: The bi-mercator grid as a global framework for numerical weather prediction. Preprints, *Workshop on the Solution of Partial Differential Equations on the Sphere*, Yokohama, Japan, Frontier Research Center for Global Change, 46.

Qaddouri, A., 2011: Nonlinear shallow-water equations on the Yin–Yang grid. *Quart. J. Roy. Meteor. Soc.*, **137**, 810–818, doi:10.1002/qj.792.

—, and V. Lee, 2011: The Canadian Global Environmental Multiscale model on the Yin–Yang grid system. *Quart. J. Roy. Meteor. Soc.*, **137**, 1913–1926, doi:10.1002/qj.873.

—, L. Laayouni, S. Loisel, J. Côté, and M. J. Gander, 2008: Optimized Schwarz methods with an overset grid for the shallow-water equations: Preliminary results. *Appl. Numer. Math.*, **58** (4), 459–471, doi:10.1016/j.apnum.2007.01.015.

Rančić, M., R. J. Purser, and F. Mesinger, 1996: A global shallow-water model using an expanded spherical cube: Gnomonic

- versus conformal coordinates. *Quart. J. Roy. Meteor. Soc.*, **122**, 959–982, doi:10.1002/qj.49712253209.
- Ronchi, C., R. Iacono, and P. S. Paolucci, 1996: The “cubed sphere”: A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, **124** (1), 93–114, doi:10.1006/jcph.1996.0047.
- Sadourny, R., 1972: Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.*, **100**, 136–144.
- , A. Arakawa, and Y. Mintz, 1968: Integration of the non-divergent barotropic vorticity equation with an icosahedral-hexagonal grid for the sphere. *Mon. Wea. Rev.*, **96**, 351–356.
- Schwarz, H. A., 1870: Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschr. Naturforsch. Ges. Zuerich*, **15**, 272–286.
- Staniforth, A., and J. Thuburn, 2012: Horizontal grids for global weather and climate prediction models: A review. *Quart. J. Roy. Meteor. Soc.*, **138**, 1–26, doi:10.1002/qj.958.
- Taylor, M., J. Tribbia, and M. Iskandarani, 1997: The spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.*, **130** (1), 92–108, doi:10.1006/jcph.1996.5554.
- Thomas, S. J., and R. D. Loft, 2000: Parallel semi-implicit spectral element methods for atmospheric general circulation models. *J. Sci. Comput.*, **15**, 499–518, doi:10.1023/A:1011188832645.
- Toro, E. F., 2009: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. 3rd ed. Springer, 724 pp.
- Williamson, D. L., 1968: Integration of the barotropic vorticity equation on a spherical geodesic grid. *Tellus*, **20** (4), 642–653, doi:10.1111/j.2153-3490.1968.tb00406.x.
- , 2007: The evolution of dynamical cores for global atmospheric models. *J. Meteor. Soc. Japan*, **85B**, 241–269.
- , J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.*, **102**, 211–224.
- Zerroukat, M., and T. Allen, 2012: On the solution of elliptic problems on overset/Yin–Yang grids. *Mon. Wea. Rev.*, **140**, 2756–2767.