A Conservative Semi-Lagrangian Discontinuous Galerkin Scheme on the Cubed Sphere

WEI GUO

Department of Mathematics, University of Houston, Houston, Texas

RAMACHANDRAN D. NAIR

National Center for Atmospheric Research, Boulder, Colorado

JING-MEI QIU

Department of Mathematics, University of Houston, Houston, Texas

(Manuscript received 17 January 2013, in final form 12 July 2013)

ABSTRACT

The discontinuous Galerkin (DG) methods designed for hyperbolic problems arising from a wide range of applications are known to enjoy many computational advantages. DG methods coupled with strong-stability-preserving explicit Runge–Kutta discontinuous Galerkin (RKDG) time discretizations provide a robust numerical approach suitable for geoscience applications including atmospheric modeling. However, a major drawback of the RKDG method is its stringent Courant–Friedrichs–Lewy (CFL) stability restriction associated with explicit time stepping. To address this issue, the authors adopt a dimension-splitting approach where a semi-Lagrangian (SL) time-stepping strategy is combined with the DG method. The resulting SLDG scheme employs a sequence of 1D operations for solving multidimensional transport equations. The SLDG scheme is inherently conservative and has the option to incorporate a local positivity-preserving filter for tracers. A novel feature of the SLDG algorithm is that it can be used for multitracer transport for global models employing spectral-element grids, without using an additional finite-volume grid system. The quality of the proposed method is demonstrated via benchmark tests on Cartesian and cubed-sphere geometry, which employs nonorthogonal, curvilinear coordinates.

1. Introduction

High-order element-based Galerkin methods are becoming increasingly popular in global atmospheric modeling, because these methods have computationally desirable features such as excellent parallel scalability, geometric flexibility, and conservation properties. The spectral-element (SE) method and discontinuous Galerkin (DG) method belong to this class, and they have recently been adopted for new generation atmospheric modeling (Dennis et al. 2011; Nair et al. 2009; Giraldo and Restelli 2008). However, a major drawback of these methods is the stringent Courant–Friedrichs–Lewy (CFL) stability limit resulting from an explicit time stepping. For example, the linear stability analysis for the DG method employing the P^k piecewise-polynomial solution space, combined with the Runge–Kutta discontinuous Galerkin (RKDG) time discretizations show that the CFL limit is $1/(2k + 1)$ (Cockburn and Shu 2001), where k is the degree of the polynomial. A possible remedy for this limitation associated with high-order methods is to reduce the order of polynomial. The SE dynamical core (Dennis et al. 2011) implemented in the Community Atmospheric Model (CAM) employs a fourth-order spatial discretization with an explicit time integration.

In an operational climate model, the total computational expense is dominated by that of the tracer transport scheme, which is accountable for $O(100)$ tracer species including several moisture variables. Recently, Erath and Nair (2014) showed that the Conservative Semi-Lagrangian Multi-Tracer Transport Scheme (CSLAM; Lauritzen et al. 2010) based on the finite-volume semi-Lagrangian philosophy is an efficient alternative to the native SE transport scheme based on the Eulerian

Corresponding author address: J.-M. Qiu, Department of Mathematics, University of Houston, 4800 Calhoun Rd., Houston, TX 77204. E-mail: jingqiu@math.uh.edu

DOI: 10.1175/MWR-D-13-00048.1

approach in the CAM framework. As the number of tracer species increases to more than six or so, the semi-Lagrangian scheme becomes significantly more efficient. This is due to the fact that once the upstream trajectory and other geometric information are computed, they can be reused for each tracer field. Although the semi-Lagrangian scheme can take a larger time step, a *moderate* value CFL ≈ 1 would be desirable to maintain the parallel efficiency. However, the CSLAM scheme employs its own uniform finite-volume cells within each spectral element defined by highly nonuniform Gauss–Legendre–Labotto (GLL) quadrature points (Erath and Nair 2014). This necessitates two grids system, one for the SE dynamics (GLL) and the other one for the CSALM transport, requiring a grid-to-grid remapping.

In this paper, our main goal is to develop a conservative transport scheme by combining the semi-Lagrangian (SL) approach and the DG method. The SLDG scheme considered is a dimension-splitting approach, which simplifies the implementation in Cartesian as well as curvilinear (cubed sphere) geometry. The scheme is particularly designed for a nodal DG discretization employing GLL grid with CFL number approximately 1. The SLDG scheme can be directly implemented for SE grids, and does not require two grids system as in the case of SE and CSLAM combination.

Mass conservative SLDG methods have been introduced in Restelli et al. (2006), Rossmanith and Seal (2011), and Qiu and Shu (2011). More recently, a semi-implicit SLDG scheme was proposed for the shallow-water equations in Tumolo et al. (2013). Our proposed SLDG method shares some similarity with existing methods in the sense that all of the methods consider the DG framework coupled with some characteristics tracing mechanism. It follows the weak formulation of the Lagrangian–Galerkin approach of Childs and Morton (1990) and Russell and Celia (2002) with piecewise discontinuous polynomials as solution spaces. The SLDG method proposed in this paper differs from those in Restelli et al. (2006) and Qiu and Shu (2011). The SLDG methods in Restelli et al. (2006) and Qiu and Shu (2011) follows an Eulerian volume (fixed spatial cell) with numerical fluxes obtained from tracing the characteristics. However, the proposed method follows the Lagrangian volume dynamically moving with the characteristics, in the same spirit as in Rossmanith and Seal (2011). Our method is more general than Rossmanith and Seal (2011), in the sense that it permits a nonuniform velocity field and is extendable to curvilinear coordinates, whereas the SLDG method in Rossmanith and Seal (2011) can only be applied to problems with constant velocity.

The remainder of the paper is organized as follows. In section 2, the SLDG algorithm on a Cartesian mesh is described. An extension of the SLDG scheme to the cubed-sphere geometry is described in section 3. In section 4, numerical experiments are presented, followed by summary and conclusions in section 5.

2. SLDG methods on a Cartesian mesh

In this section, we introduce mass conservative SLDG methods for a 2D linear advection problem with variable coefficients:

$$\phi_t + [u(x,y,t)\phi]_x + [v(x,y,t)\phi]_y = 0. \tag{1}$$

When the velocity field (u, v) is nondivergent, i.e., $\nabla \cdot (u, v) = 0$, the equation is equivalent to the advective form:

$$\phi_t + u(x,y,t)\phi_x + v(x,y,t)\phi_y = 0. \tag{2}$$

Our proposed algorithm is essentially a 1D algorithm for a variable coefficient problem in the conservative (flux) form:

$$\phi_t + [u(x,y,t)\phi]_x = 0. \tag{3}$$

The flux form facilitates element-wise mass conservation. The algorithm can be extended to 2D problems via dimension splitting on a Cartesian mesh (Qiu and Shu 2011).

a. SLDG for 1D problems

We assume a spatial discretization of a 1D domain $[a, b]$ as nonoverlapping cells (elements) such that $\cup I_j = \cup[x_{j-(1/2)}, x_{j+(1/2)}]$ with $h = \max\_j[x_{j+(1/2)} - x_{j-(1/2)}]$. A DG solution space is a vector space $V_h^k = \{v_h : v_h|_{I_i} \in P^k(I_j)\}$, where $P^k(I_j)$ denotes the set of polynomials of degree at most k. To update the solution at the time-level t^{n+1} over a cell I_j from the solution at t^n, we use the weak formulation of characteristic Galerkin method of Childs and Morton (1990) and Russell and Celia (2002). Specifically, we let the test function $\psi(x, t)$ satisfy the adjoint problem with $\forall \Psi \in P^k(I_j)$:

$$\begin{cases} \psi_t + u(x,y,t)\psi_x = 0, \\ \psi(t = t^{n+1}) = \Psi(x). \end{cases} \tag{4}$$

We remark that for the above advective form of equation, the solution stays constant along a trajectory; while for the conservative form of equation, the solution varies along the trajectory. It can be shown that

$$\frac{d}{dt}\int_{I_j(t)} \phi(x,t)\psi(x,t)\,dx = 0, \tag{5}$$

where $I_j(t)$ is a dynamic interval bounded by characteristics emanating from cell boundaries of I_j at $t = t^{n+1}$. Equation (5) can be proved by the following:

$$\frac{d}{dt}\int_{I_j(t)}\phi(x,t)\psi(x,t)\,dx = \left.\frac{dx}{dt}\right|_{x_{j+(1/2)}}\phi[x_{j+(1/2)}(t),t]\psi[x_{j+(1/2)}(t),t] - \left.\frac{dx}{dt}\right|_{x_{j-(1/2)}}\phi[x_{j-(1/2)}(t),t]\psi[x_{j-(1/2)}(t),t]$$
$$-\int_{I_j(t)}[u(x,y,t)\phi(x,t)]_x\psi(x,t)\,dx + \int_{I_j(t)}\phi(x,t)\psi_t(x,t)\,dx = u\phi\psi|_{[x_{j+(1/2)}(t),y,t]} - u\phi\psi|_{[x_{j-(1/2)}(t),y,t]}$$
$$-\left\{u\phi\psi|_{[x_{j+(1/2)}(t),y,t]} - u\phi\psi|_{[x_{j-(1/2)}(t),y,t]}\right\} + \int_{I_j(t)}u(x,y,t)\phi(x,t)\psi_x(x,t)\,dx$$
$$-\int_{I_j(t)}\phi(x,t)u(x,y,t)\psi_x(x,t)\,dx = 0.$$

A semi-Lagrangian time discretization of Eq. (5) leads to

$$\int_{I_j}\phi^{n+1}\Psi\,dx = \int_{I_j^\star}\phi(x,t^n)\psi(x,t^n)\,dx, \tag{6}$$

where $I_j^\star \doteq [x^\star_{j-(1/2)}, x^\star_{j+(1/2)}]$ with $x^\star_{j\pm(1/2)}$ being the foot of trajectory emanating from $(x_{j\pm(1/2)}, t^{n+1})$ at time t^n. To update the numerical solution ϕ^{n+1}, the following procedures are performed.

1) Locate the foot of trajectory $x^\star_{j\pm(1/2)}$ [see Fig. 1 (left)]. We numerically solve the following final-value problem (trajectory equation):

$$\frac{\mathbf{d}}{\mathbf{d}t}x(t) = u[x(t),y,t] \tag{7}$$

with the final-value $x(t^{n+1}) = x_{j\pm(1/2)}$ by a high-order numerical integrator such as a classical fourth-order Runge–Kutta method. Here y is fixed with dimension splitting for multidimensional problems.

2) Detect subintervals within $I_j^\star = \cup_l I_{j,l}^\star$, which are all the intersections between $I_j^\star$ and the grid elements (l is the index for subinterval). For example, in Fig. 1 (left), there are two subintervals: $I_{j,1}^\star = [x^\star_{j-(1/2)}, x_{j-(1/2)}]$ and $I_{j,2}^\star = [x_{j-(1/2)}, x^\star_{j+(1/2)}]$.

3) Locate the $(k+1)$ local GLL points over each $I_{j,l}^\star$, which are mapped from the standard GLL points defined on the reference interval $[-1,1]$ by an affine transformation. We denote them as $x^\star_{j,l,\mathrm{ig}}$ (ig is the index for GLL points). See the red circles as 4 GLL points per subinterval in Fig. 1 (right).

4) Trace trajectories forward in time from $(x^\star_{j,l,\mathrm{ig}}, t^n)$ to $(x_{j,l,\mathrm{ig}},\ t^{n+1})$. Specifically, similar to the final-value problem above, we use a high-order time integrator to numerically solve an initial value problem in Eq. (7) with the initial-value $x(t^n) = x^\star_{j,l,\mathrm{ig}}$ [see the green curve and circles in Fig. 1 (right)]. From the advective form of the adjoint problem in Eq. (4) one has

$$\psi(x^\star_{j,l,\mathrm{ig}}, t^n) = \Psi(x_{j,l,\mathrm{ig}}).$$

5) Use the GLL quadrature rule to evaluate

$$\int_{I_j^\star}\phi(x,t^n)\psi(x,t^n)\,dx \approx \sum_l\left[\sum_{\mathrm{ig}} w_{\mathrm{ig}}\phi(x^\star_{j,l,\mathrm{ig}},t^n)\Psi(x_{j,l,\mathrm{ig}})\Gamma(I_{j,l}^\star)\right], \tag{8}$$

with w_{ig} being the quadrature weights for a unit length interval and $\Gamma(I_{j,l}^\star)$ being the length of subinterval $I_{j,l}^\star$. Note that the accuracy of the GLL quadrature rule is order of $2k$ when $k+1$ GLL points are used. As in the classical DG formulation, the evaluation of volume integral will not destroy the $(k+1)$th order of accuracy for the SLDG scheme. Moreover, the mass conservation properties are not affected since the numerical integration is exact for a polynomial of degree k with the test function $\psi = 1$.

6) Finally, find $\phi^{n+1} \in P^k(I_j)$, s.t. Eq. (6) is satisfied $\forall\Psi \in P^k(I_j)$ with the rhs term evaluated as described above.

Note that the proposed 1D algorithm is fourth-order accurate in time in the sense of local truncation error:

$$\frac{1}{\Delta t}\left|x^\star_{j\pm(1/2)} - x(t^n; x_{j\pm(1/2)}, t^{n+1})\right| = O(\Delta t^4),$$
$$\frac{1}{\Delta t}\left|x_{j,l,\mathrm{ig}} - x(t^{n+1}; x^\star_{j,l,\mathrm{ig}}, t^n)\right| = O(\Delta t^4)$$

with a fourth-order Runge–Kutta method for tracing trajectories. Here $\Delta t = t^{n+1} - t^n$; $x(t^n; x_{j\pm(1/2)}, t^{n+1})$ and $x(t^{n+1}; x^\star_{j,l,\mathrm{ig}}, t^n)$ denote the exact solutions of the trajectory equation in Eq. (7) with final-value $x(t^{n+1}) = x_{j\pm(1/2)}$ and initial-value $x(t^n) = x^\star_{j,l,\mathrm{ig}}$, respectively. The numerical error is likely to be dominated by the spatial error of the DG methods and the second-order splitting error for multidimensional problems. If this is not the case, one can always reduce error from trajectory tracing by either increasing the order of numerical integrator, or by taking smaller time steps.

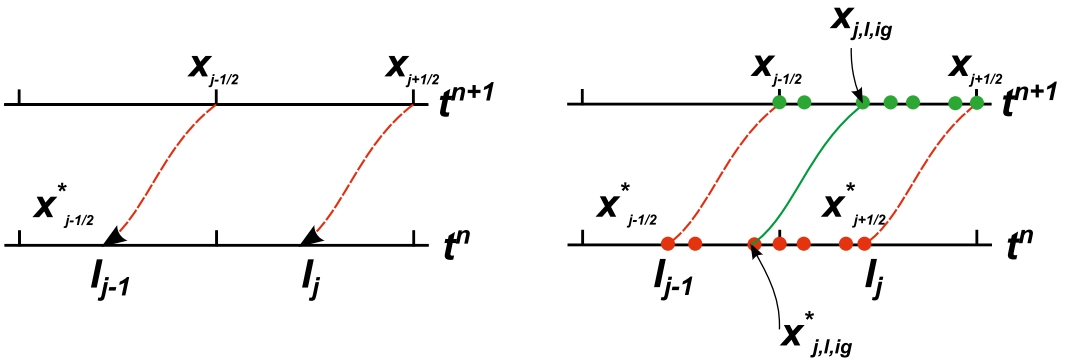

FIG. 1. Schematic showing the 1D SLDG scheme, as described in the text. (left) Steps 1 and 2 and (right) steps 3 and 4. Four GLL points per cell are used as an example.

b. The bound-preserving (BP) filter

It is known that Eq. (3) is mass conservative. However, the solution does not fulfill the maximum principle (i.e., the solution in the future time will not be bounded by the lower and upper bound of the initial condition). On the other hand, if the initial condition for Eq. (3) is positive [$\phi(x, t = 0) \geq 0, \forall x$], then the future solution stays positive [$\phi(x, t) \geq 0, \forall x, t$]. We call such property as positivity preserving (PP). In our SLDG scheme, it can be easily checked that the updated cell average at t^{n+1} [taking $\Psi = 1$ in Eq. (8)] stays positive, if the numerical solution (piecewise polynomial function) at t^n is positive. However, the numerical solution at t^{n+1} does not necessarily stay positive. To ensure PP of the numerical solution, we apply a BP filter (Zhang and Shu 2010a,b; Qiu and Shu 2011; Zhang and Nair 2012), if the initial condition stays positive. The procedure of the BP filter can be viewed as ensuring the positivity of the numerical solution by a linear rescaling around the cell averages, with the assumption that the cell averages are positivity preserving. In particular, the numerical solution is modified from $\phi(x)$ to $\tilde{\phi}(x)$ such that it maintains the high-order accuracy of the original approximation, conserves the cell average (mass) and preserves positivity:

$$\tilde{\phi}(x) = \theta[\phi(x) - \overline{\phi}] + \overline{\phi}, \quad \theta = \min\left[\left|\frac{\overline{\phi}}{m' - \overline{\phi}}\right|, 1\right], \tag{9}$$

where $\overline{\phi}$ is the cell average of the numerical solution and m' is the minimum of $\phi(x)$ over a given cell. A formal proof can be found in Zhang and Shu (2010a) (Lemma 2.4). To implement the BP filter, the minimum of the numerical solution m' is needed. In our numerical tests, we use up to P^3 polynomials, the minimum of which can be easily found by locating the zeros of their derivatives. The proposed SLDG methods with the BP filter enjoy the L^1 (mass) conservation, the proof of which can be found in Qiu and Shu (2011).

c. SLDG schemes for multidimensional problems via operator splitting

The proposed 1D algorithm can be extended to multidimensional algorithms via the second-order Strang dimension splitting based on a Cartesian mesh in Strang (1968). Below, we sketch the idea of the algorithm. For more implementation details, see Qiu and Shu (2011).

1) Split Eq. (1) into two 1D advection problems:

$$\phi_t + [u(x, y, t)\phi]_x = 0, \tag{10}$$

$$\phi_t + [v(x, y, t)\phi]_y = 0. \tag{11}$$

2) Locate $(k + 1)$ GLL points in both x and y directions in each rectangular cell as (x_{ig}, y_{jg}). See Fig. 2 (left).
3) Perform the Strang dimension-splitting strategy, for which the numerical update over a time step Δt is as follows:
   (i) Evolve 1D Eq. (10) at different y_{jg} locations with corresponding velocities $u(x, y_{jg}, t)$ for a half time step $\Delta t/2$, see Fig. 2 (middle).
   (ii) Evolve 1D Eq. (11) at different x_{ig} locations with corresponding velocities $v(x_{ig}, y, t)$ for a whole time step Δt, see Fig. 2 (right).
   (iii) Evolve 1D Eq. (10) for another half time step $\Delta t/2$ as in (i).

Note that the BP filter is applied separately in each direction and the resulting scheme can preserve positivity (see Rossmanith and Seal 2011; Qiu and Shu 2011). It is difficult to design a numerical scheme that preserves a constant field in the dimensional splitting framework. It is our ongoing work to design a nonsplitting SLDG scheme that preserves the constant field when the velocity field is nondivergent.

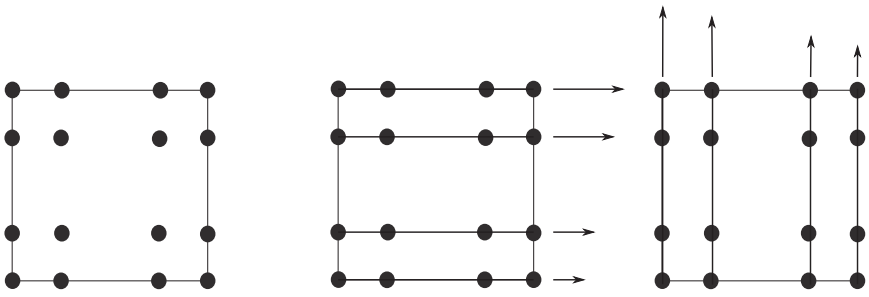

FIG. 2. Schematic showing the 2D SLDG scheme via Strang splitting, as described in the text. (left) Locating $k+1$ GLL points in both x and y directions. (middle) Evolution in x direction at different y_{jg}. (right) Evolution in y direction at different x_{ig}. As an example, 4×4 GLL points per cell are used.

3. SLDG schemes for the cubed-sphere geometry

a. Cubed-sphere geometry

In this section, we extend the SLDG scheme to the cubed-sphere geometry. The cubed-sphere geometry (Sadourny 1972; Ronchi et al. 1996) is free from polar singularities and well-suited for element-based Galerkin methods. In this paper, we consider the cubed-sphere mesh generated by an equiangular central projection as described in Nair et al. (2005). In the cubed-sphere grid system, the spherical domain is partitioned into six identical regions (faces). On each face, grid lines follow nonorthogonal curvilinear coordinate system (x^1, x^2) such that $x^1, x^2 \in [-\pi/4, \pi/4]$ (see Fig. 3). Each face of the cubed sphere is tiled with $N_e \times N_e$ elements (cells) so that $6 \times N_e^2$ elements span the entire spherical domain. See Fig. 3 for the relative position of six faces and the equiangular central projection from the Cartesian mesh on the cube to the curvilinear mesh on the sphere with $N_e = 5$. The metric tensor associated with the equiangular central (gnomonic) mapping is given by

$$g_{ij} = \frac{R^2}{\rho^4 \cos^2 x^1 \cos^2 x^2} \times \begin{bmatrix} 1+\tan^2 x^1 & -\tan x^1 \tan x^2 \\ -\tan x^1 \tan x^2 & 1+\tan^2 x^2 \end{bmatrix}, \tag{12}$$

where $i, j \in \{1, 2\}$, $\rho = 1 + \tan^2 x^1 + \tan^2 x^2$, and R is radius of the sphere. Denote $g = \det(g_{ij})$, then the Jacobian of the transformation is given by $\sqrt{g}$, which is identical on each face. We refer to Nair et al. (2005) for all the conversion formulas between the usual latitude–longitude velocity components u, v and the contravariant components u^1, u^2 on the cubed sphere.

b. SLDG transport on the cubed sphere

The transport equation for a scalar ϕ on the cubed sphere can be rewritten in (x^1, x^2) coordinate on each face as follows (Zhang and Nair 2012):

$$\frac{\partial U}{\partial t} + \frac{\partial F_1(U)}{\partial x^1} + \frac{\partial F_2(U)}{\partial x^2} = 0, \tag{13}$$

where $U = \phi\sqrt{g}$, and the fluxes $F_1 = u^1 U$, and $F_2 = u^2 U$; the Jacobian $\sqrt{g}$ is a given continuous function of the curvilinear coordinate transform. For the DG spatial discretization, each element is further mapped onto $N_p \times N_p$ GLL grids. Figure 3 (bottom) shows a cubed sphere discretized with uniform size elements with $N_e = 5$ and $N_p = 4$. Equation (13) is in a conservative form, similar to Eq. (1) for a Cartesian mesh. Therefore, the SLDG scheme can be implemented in a similar fashion as that for a Cartesian mesh previously described. As we directly work with U, the mass conservation property of numerical solution is preserved. Below, we focus our description on the differences of implementation.

The cubed-sphere grid lines may be interpreted as three families of piecewise closed great-circle arcs (ξ, η, ζ) on the sphere (see Figs. 3 and 4). We exploit this idea for solving transport equations on the cubed sphere. As a result of this special geometric feature of the cubed-sphere grid system, there exists three logical transport directions denoted as ξ, η, and ζ directions as shown in Fig. 4. Specifically,

- ξ is the direction along the x^1 axis of face1, face2, face3, and face4 [see Figs. 3 and 4 (upper left)].
- η is the direction along the x^2 axis of face1, face3, face5, and face6 [see Fig. 4 (upper right)]
- ζ is the direction along the x^2 axis of face2 and face4, and along the x^1 axis of face5 and face6 [see Fig. 4 (bottom)].

In the (ξ, η, ζ) coordinate, the transport equation in Eq. (13) on the cubed sphere can be rewritten as

$$\frac{\partial U}{\partial t} + \frac{\partial \widetilde{F_1}(U)}{\partial \xi} + \frac{\partial \widetilde{F_2}(U)}{\partial \eta} + \frac{\partial \widetilde{F_3}(U)}{\partial \zeta} = 0, \tag{14}$$

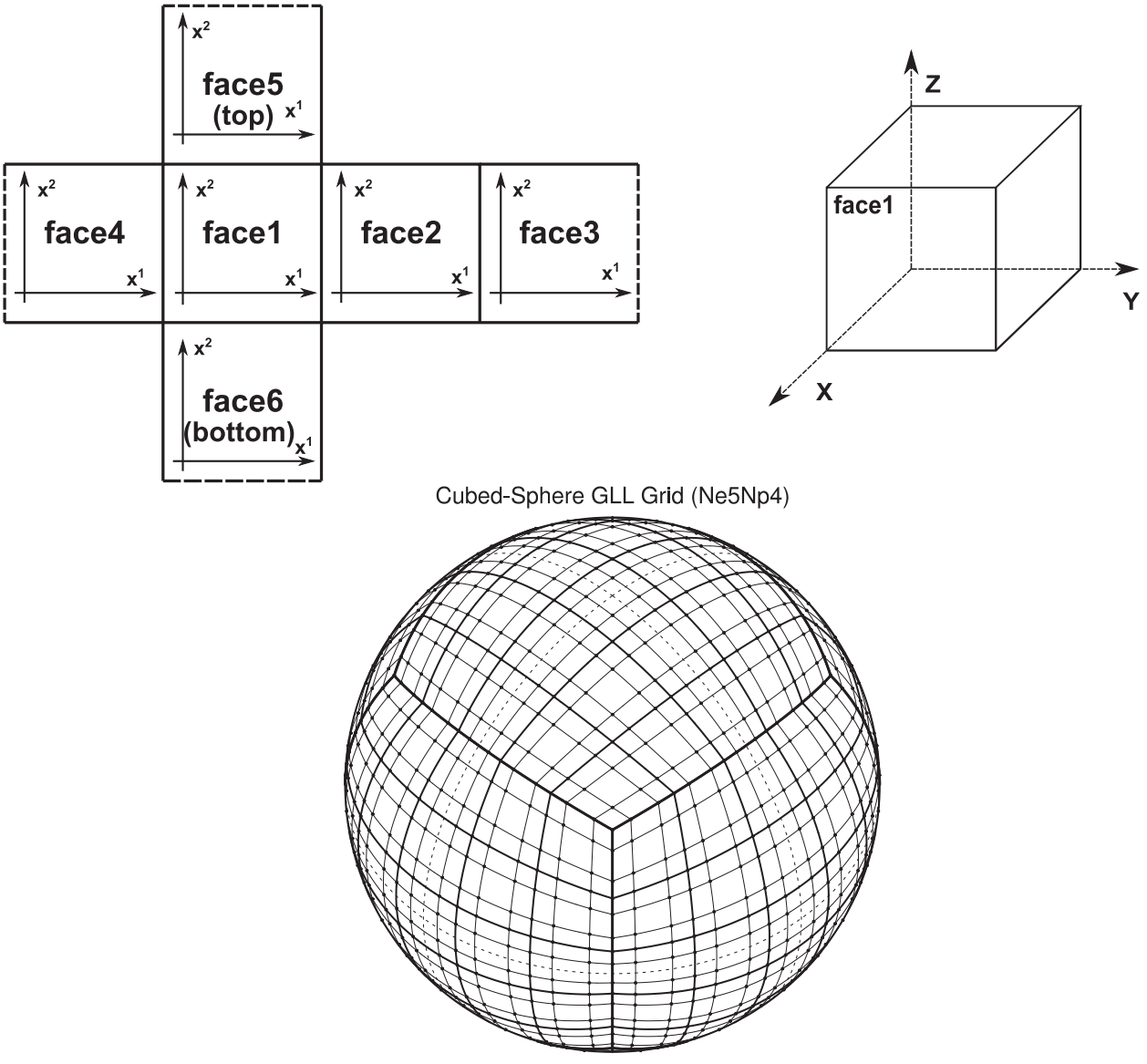

FIG. 3. (top left) Schematic for the relative positions of the six cube faces (from face1 to face6), (top right) their local connectivity, and (bottom) a cubed-sphere with total $6N_e^2 = 150$ elements (cells) ($N_e = 5$). Each element has 4×4 GLL points ($N_p = 4$). The thick lines show element edges. The GLL points within each element are marked as dots. The thin lines denote the GLL grids.

where $\tilde{F}_1, \tilde{F}_2, \tilde{F}_3 \in \{F_1, F_2\}$ such that Eqs. (13) and (14) are equivalent on any given face. We note that the unknown function U *only* depends on two variables (x^1, x^2) on each cube face. For example, U is dependent only on $\xi(x^1)$ and $\eta(x^2)$ on face1 and only on $\xi(x^1)$ and $\zeta(x^2)$ on face2, also see Fig. 4. Thus, Eq. (14) is essentially identical to Eq. (13) on each face. A second-order Strang-type splitting strategy for multiple operators proposed in Gottlieb (1972) can be used in Eq. (14). We summarize the procedure as follows:

1) Equation (14) is split into three 1D advection problems on the cubed sphere:

$$\frac{\partial U}{\partial t} + \frac{\partial \tilde{F}_1(U)}{\partial \xi} = 0, \tag{15}$$

$$\frac{\partial U}{\partial t} + \frac{\partial \tilde{F}_2(U)}{\partial \eta} = 0, \tag{16}$$

$$\frac{\partial U}{\partial t} + \frac{\partial \tilde{F}_3(U)}{\partial \zeta} = 0. \tag{17}$$

2) The numerical solution is updated by a Strang-type splitting strategy for one time step Δt:
   (i) Evolve the 1D equation in Eq. (15) in the direction ξ for $\Delta t/2$ [see Fig. 4 (upper left)].

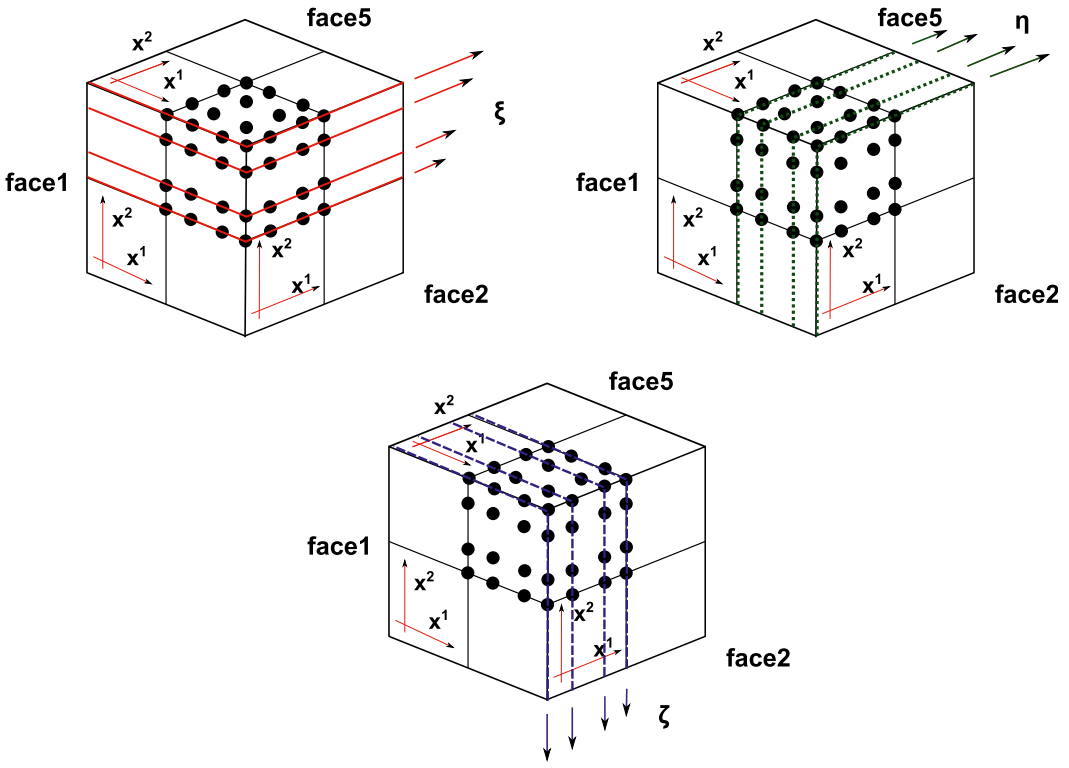

FIG. 4. Schematic showing the SLDG scheme on mesh $2 \times 2 \times 6$ of cubed-sphere geometry. As an example, 4×4 GLL points per element are used. Advection in the (top left) ξ direction, (top right) η direction, and (bottom) ζ direction.

(ii) Evolve the 1D equation in Eq. (16) in the direction η for $\Delta t/2$ [see Fig. 4 (upper right)].
(iii) Evolve the 1D equation in Eq. (17) in the direction ζ for Δt [see Fig. 4 (bottom)].
(iv) Evolve the 1D equation in Eq. (16) in the direction η for $\Delta t/2$ as in (ii).
(v) Evolve the 1D equation in Eq. (15) in the direction ξ for $\Delta t/2$ as in (i).

The evolution of each 1D equation follows a similar procedure as that for the Cartesian mesh, except for the tracing of characteristics across face edges. As before, we employ a fourth-order Runge–Kutta method for solving the trajectory equations:

$$\frac{\mathbf{d}x^1(t)}{\mathbf{d}t} = u^1[x^1(t), x^2_{\mathrm{jg}}, t], \tag{18}$$

where x^2_{jg} is a fixed GLL point. Below we only demonstrate the initial value problem case (forward trajectory), while the backward case is similar. Note that special treatment is needed, since the velocity u^1 takes different expressions on different faces. Below is the procedure implemented in our code in the case of characteristics emanating from $(x^\star, t^n)$ crossing the edge of face1 and face2, see Fig. 5.

1) Find the time point $t^\star \in [t^n, t^{n+1}]$, when the trajectory reaches the face edge, denoted as x_e. The following gives a second-order way of approximating $t^\star$ when the u^1_s ($s = 1$ or 2 being the index for a face) is time independent [see Fig. 5 (left)]:

$$x_e - x^\star = \frac{t^\star - t^n}{2}[u^1_1(x_e) + u^1_1(x^\star)], \quad \text{or}$$
$$t^\star = t^n + \frac{2(x_e - x^\star)}{u^1_1(x_e) + u^1_1(x^\star)}. \tag{19}$$

In some practical applications, such as the multitracer transport, u^1_s is only given at t^n, so it is reasonable to assume that u^1_s is constant in $[t^n, t^{n+1}]$. In the time-dependent case, assume we have a high-order Runge–Kutta method (e.g., a fourth-order Runge–Kutta method) to solve the initial value problem in Eq. (18) with $x^1(t^n) = x^\star$. Denote the numerical solution of Eq. (18) at time t as $\mathrm{RK}(t; x^\star, t^n)$, then we want to find $t^\star$ such that

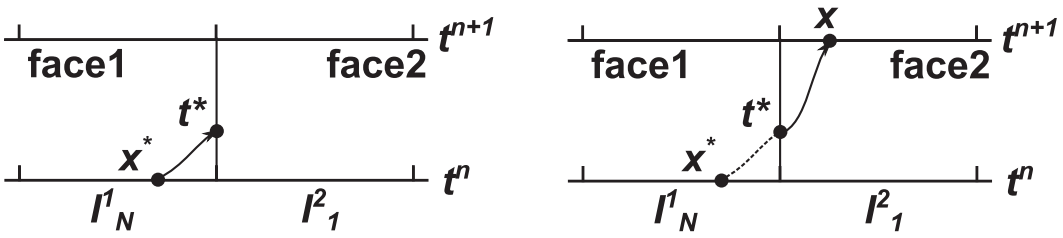

FIG. 5. Obtain the arrival point x of the SLDG scheme at the edge of face1 and face2. (left) Find the time $t^\star$ when the trajectory emanating from $x^\star$ reaches the edge. (right) Get the arrival point x starting at time $t^\star$.

$$\mathrm{RK}(t^\star; x^\star, t^n) = x_e. \tag{20}$$

We adopt the Newton's type method to solve Eq. (20):

(i) Using Eq. (19) to get

$$t^{\star,0} = t^n + \frac{2(x_e - x^\star)}{u_1^1(x_e, t^n) + u_1^1(x^\star, t^n)},$$

which is a good prediction of $t^\star$.

(ii) Set a threshold ε and do the following iteration:

$$x_e^j = \mathrm{RK}(t^{\star,j}; x^\star, t^n);$$

$$\text{If} \quad |x_e^j - x_e| < \varepsilon, \text{stop, and let } t^\star = t^{\star,j},$$

else

$$t^{\star,j+1} = t^{\star,j} + \frac{x_e^j - x_e}{u_1^1(x_e^j, t^{\star,j})}.$$

In the simulation, it takes about three–five iterations to reach an error tolerance of $\varepsilon = 10^{-10}$.

2) Continue evolving the characteristic on face2 to locate the arrival point x at the next time step t^{n+1} [see Fig. 5 (right)].

4. Numerical experiments

a. 2D Cartesian test

In this section, we apply the proposed SLDG scheme with the BP filter to several benchmark transport problems on the 2D Cartesian mesh, including solid-body rotation of a smooth Gaussian hill and deformation flow. In the simulations, P^3 with 4 GLL points is used. The numerical results will demonstrate the accuracy, efficiency and reliability of the proposed scheme when solving the transport equations.

1) SOLID-BODY ROTATION OF A GAUSSIAN HILL

Consider the following 2D solid-body rotation problem:

$$\phi_t + (u\phi)_x + (v\phi)_y = 0, \tag{21}$$

where the nondivergent velocity is defined to be $(u, v) = (-y, x)$, on a periodic domain $\mathcal{D} = [-1, 1]^2$. For the solid-body rotation test, the initial distribution will translate on a circular trajectory centered at origin without incurring any deformation. Moreover, the exact solution is known at any time and will return to the initial state after a 2π evolution. We choose the initial condition as a smooth Gaussian hill:

$$\phi(x, y, t = 0) = a_c \exp\{-b_c[(x - x_0)^2 + (y - y_0)^2]\},$$

where $a_c = 1$, $b_c = 100/3$, and $x_0 = y_0 = 0.2$. We let CFL= 0.92, which is approximately 7 times larger than CFL= 0.13 for the RKDG P^3 scheme. We compute the numerical solution at $T = 2\pi$ and report the l_2 and l_∞ errors and orders of accuracy in Table 1. The $N_e \times N_e$ in Table 1 denotes the number of the elements we used in the simulation. It is clear that the second order of convergence is observed, which comes from the splitting error.

2) DEFORMATIONAL FLOW TEST

We consider the 2D deformation flow on Cartesian mesh proposed by Blossey and Durran (2008), which is a challenging benchmark test as opposed to the solid-body rotation test. Unlike the solid-body rotation test, the velocity field in Eq. (21) is space and time dependent, whereas the exact solution is available only at

TABLE 1. Normalized standard errors and order of accuracy for ϕ for the solid-body rotation of cosine bell test on Cartesian mesh. The SLDG P^3 scheme with the BP filter is used and CFL= 0.913. The numerical solution is computed at $T = 2\pi$. Do the same in the next tables.

$N_e \times N_e$	l_2	l_2 order	l_∞	l_∞ order
16×16	3.40×10^{-4}	—	1.57×10^{-2}	—
32×32	5.44×10^{-5}	2.65	2.98×10^{-3}	2.40
64×64	1.23×10^{-5}	2.14	5.28×10^{-4}	2.50
128×128	3.03×10^{-6}	2.03	1.02×10^{-4}	2.37
256×256	7.54×10^{-7}	2.01	2.20×10^{-5}	2.22

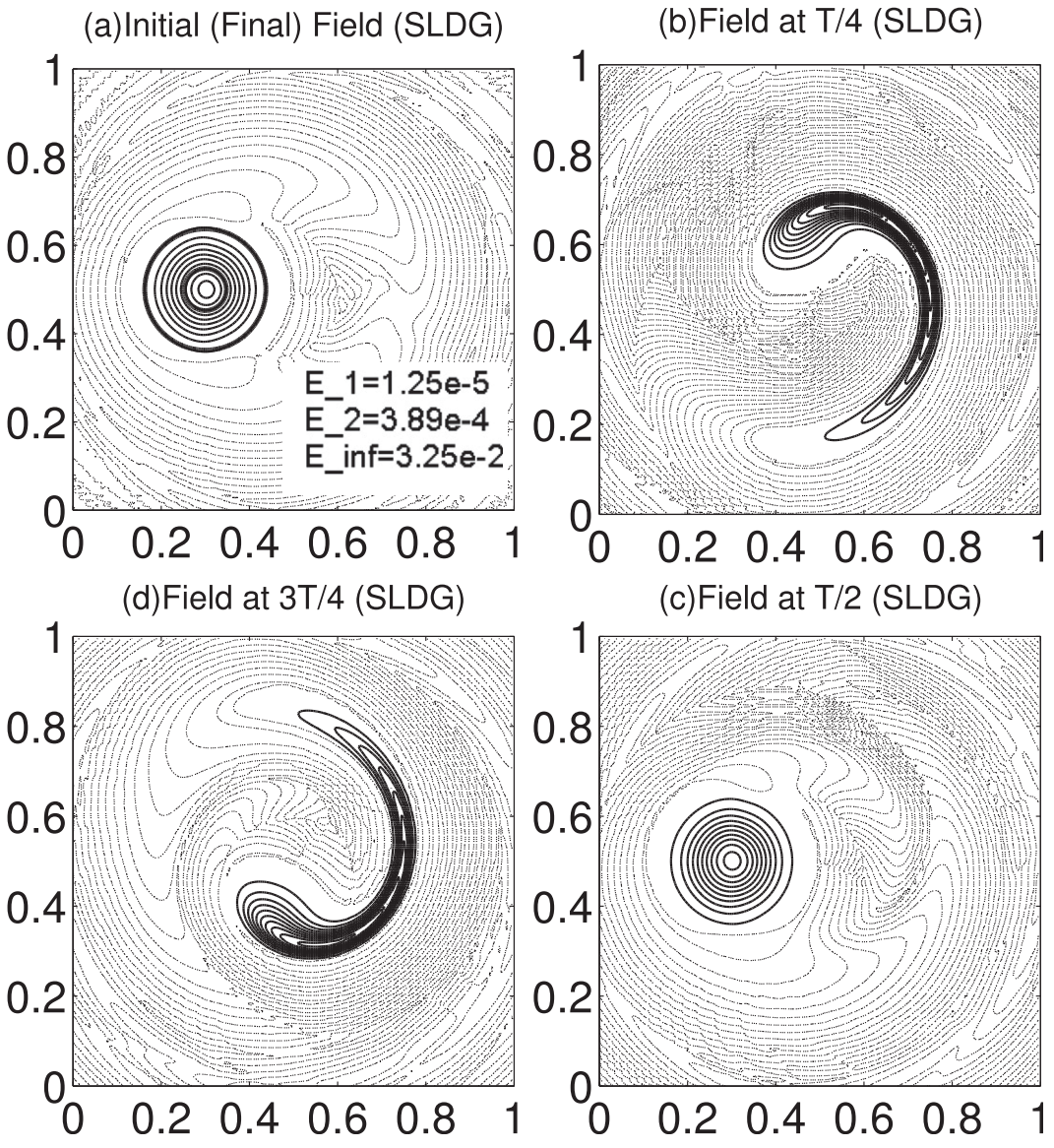

FIG. 6. The 2D deformation flow on Cartesian mesh. The SLDG P^3 scheme without the BP filter is used. $N_e = 60$ in both x and y direction and CFL = 0.95. The deformation of the initial distribution during the simulation at time (a) T, (b) $T/4$, (c) $3T/4$, and (d) $T/2$. The contours are plotted in the range from -0.05 to 0.95 with increment of 0.1, and an additional contour at 0 is added (see the dashed line). Numerical oscillations appear and a negative numerical solution is observed. Thick contours are the highlighted exact (initial) solution for the contour values 0.05 and 0.75.

the final time $t = T$. The initial circular distribution deforms into a crescent shape as it moves on the domain, and returns to the initial position when the flow reverses. The computational domain $\mathcal{D} = [0, 1]^2$, with periodic boundaries, and the velocity field is defined as

$$u(x, y, t) = u_\theta(r, t)\sin(\theta), \quad v(x, y, t) = -u_\theta(r, t)\cos(\theta),$$

where $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$, $\theta = \tan^{-1}[(y-0.5)/(x-0.5)]$, and

$$u_\theta(r, t) = \frac{4\pi r}{T}\left\{\left[1 - \cos\left(\frac{2\pi t}{T}\right)\right]\frac{1-(4r)^6}{1+(4r)^6}\right\}.$$

The initial condition is given by

$$\phi(x, y, t=0) = \begin{cases} \phi_0 + \left[\dfrac{1+\cos(\pi\tilde{r})}{2}\right]^2 & \text{if} \quad \tilde{r} \le 1 \\ \phi_0 & \text{if} \quad \tilde{r} > 1, \end{cases}$$

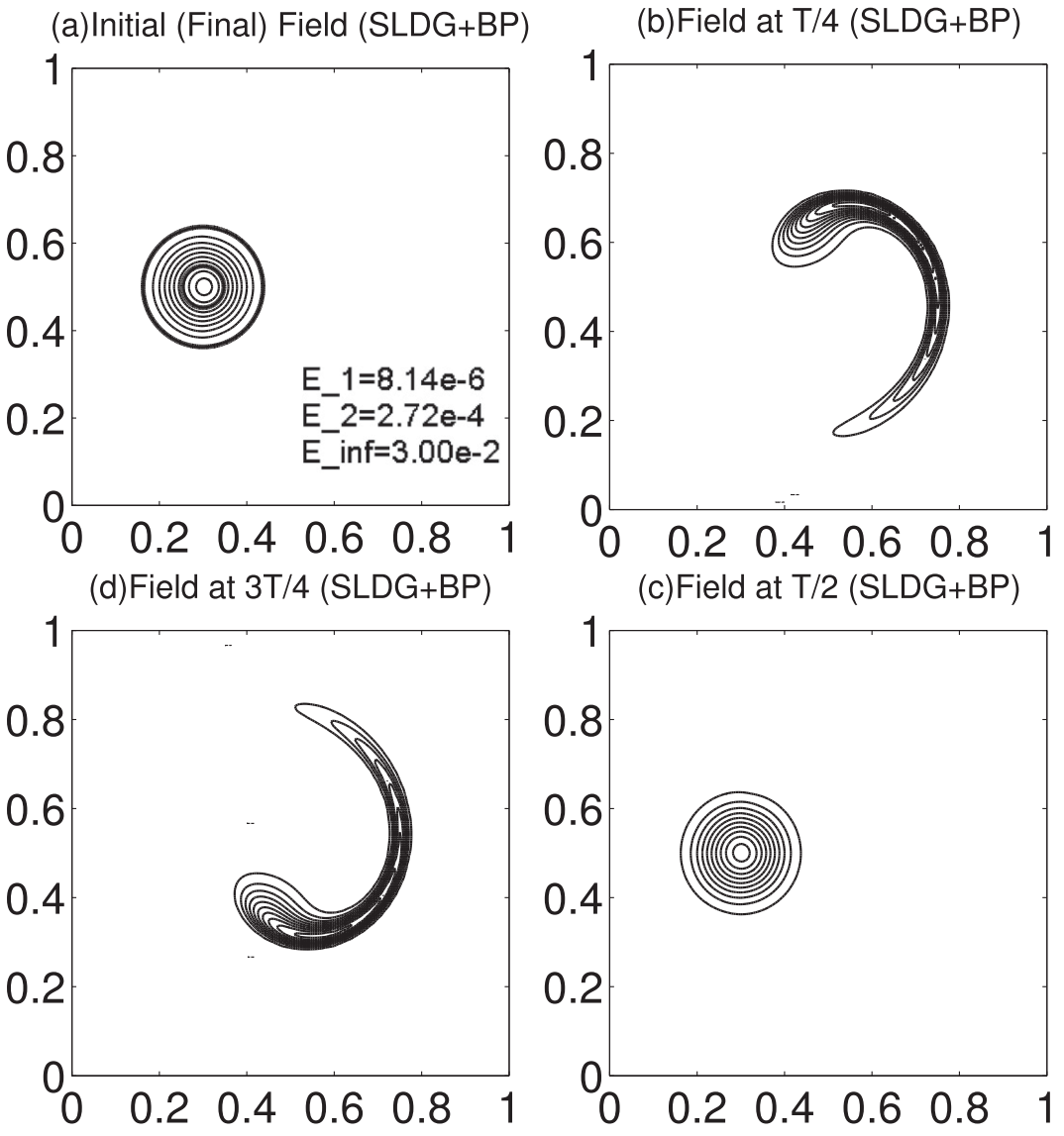

FIG. 7. As in Fig. 6, but with the BP filter. The numerical solution is exactly positivity preserving.

where $\tilde{r} = 5\sqrt{(x-0.3)^2 + (y-0.5)^2}$ and the background field is set as $\phi_0 = 0$. Though the velocity field is very complicated, the analytical solution is known at the final time $t = T$ and identical to the initial state; therefore, the error measures can be computed at time T.

For the simulation, the SLDG P^3 scheme with the number of elements $N_e = 60$, in both x and y directions were used, and the CFL number is set to 0.95. In Figs. 6 and 7, we show the contour plots of the numerical solutions at (Figs. 6a and 7a) $t = T$, (Figs. 6b and 7b) $t = T/4$, (Figs. 6c and 7c) $t = T/2$ and (Figs. 6d and 7d) $t = 3T/4$, without and with the BP filter, respectively. The numerical solution of the SLDG scheme captures the fine features of exact solution (Fig. 6) even without the BP filter. Note that the BP filter enforces the positivity on numerical solution and helps to get rid of unphysical oscillations. The SLDG scheme with the BP filter provides high-quality numerical result, at the same time, keeps the solution positivity preserving and conservative.

b. 2D spherical tests

In this section, we consider two types of 2D spherical advection tests for the SLDG scheme on the sphere. The tests include a solid-body rotation and two deformation flow tests.

1) SOLID-BODY ROTATION OF A COSINE BELL

Solid-body rotation of a cosine bell is a widely used standard test for 2D spherical advection problem

TABLE 2. Normalized standard errors for ϕ for the 2D solid-body rotation test with $\alpha = 0$ and $\alpha = \pi/4$ on the cubed-sphere geometry. The SLDG P^3 scheme is used on a $20 \times 20 \times 6$ (1.5° resolution) and time step is set as $\Delta t = 3600$ s. The numerical solution is computed after a full rotation.

	Scheme	l_1	l_2	l_∞	Mass error
$\alpha = 0$	SLDG	1.04×10^{-2}	7.03×10^{-3}	6.53×10^{-3}	-5.20×10^{-13}
	SLDG + BP	8.50×10^{-3}	5.82×10^{-3}	6.72×10^{-3}	-5.20×10^{-13}
$\alpha = \pi/4$	SLDG	1.17×10^{-2}	7.70×10^{-3}	7.20×10^{-3}	-4.13×10^{-13}
	SLDG + BP	8.93×10^{-3}	6.06×10^{-3}	7.46×10^{-3}	-4.12×10^{-13}

(Williamson et al. 1992). The initial scalar distribution (cosine bell) is defined by

$$\phi(\lambda, \theta, t = 0) = \begin{cases} (h_0/2)[1 + \cos(\pi r_d/r_0)] & \text{if} \quad r_d < r_0 \\ 0 & \text{if} \quad r_d \geq r_0, \end{cases}$$

where r_d is the great-circle distance between (λ, θ) and the bell center, which is $(3\pi/2, 0)$ at $t = 0$, $h_0 = 1000$ m is the maximum height of the cosine bell, and $r_0 = R/3$ represents the radius of the bell, here $R = 6.371\,22 \times 10^6$ is the earth's radius. The wind components in the longitudinal (λ) and latitudinal (θ) directions are defined as follows:

$$u = u_0(\cos\alpha_0 \cos\theta + \sin\alpha_0 \cos\lambda \sin\theta),$$
$$v = -u_0 \sin\alpha_0 \sin\lambda,$$

where $u_0 = 2\pi R/(12 \text{ days})$ and α is the rotation angle, which is between the axis of the solid-body rotation and the polar axis of the spherical coordinate. The flow is oriented along the equatorial direction when $\alpha = 0$ and the northeast direction when $\alpha = \pi/4$. Note that the configuration with $\alpha = \pi/4$ is more challenging for the cubed-sphere geometry. In this case, the cosine bell goes through four vertices, two edges, and all six faces. The wind field is nondivergent, which means the maximum principle holds. The exact solution is available at all times and the cosine bell reaches its initial state after a complete (12 days) rotation, thus error measures can be computed.

We apply the SLDG P^3 scheme to the solid-body rotation problem with mesh $20 \times 20 \times 6$ corresponding to 1.5° resolution at the equator for the cubed-sphere geometry. The time step is set as $\Delta t = 3600$ s, which is 6 times larger than that used by the RKDG P^2 scheme in Zhang and Nair (2012). In Table 2, we report the standard normalized error norms based on Williamson et al. (1992) with $\alpha = 0$ and $\alpha = \pi/4$. The error measured are comparable to those by the RKDG scheme in Nair et al. (2005) and Zhang and Nair (2012). Note that the proposed scheme is exactly mass conservative. In Fig. 8, the contour plots of the numerical solution are reported for $\alpha = \pi/4$. The results are observed to be comparable to those produced by a nonoscillatory RKDG scheme presented in Zhang and Nair (2012). The evolution of error norms are given in Fig. 9 for $\alpha = 0$ (Figs. 9a,b) and $\alpha = \pi/4$ (Figs. 9c–f). Note that the l_∞ error grows

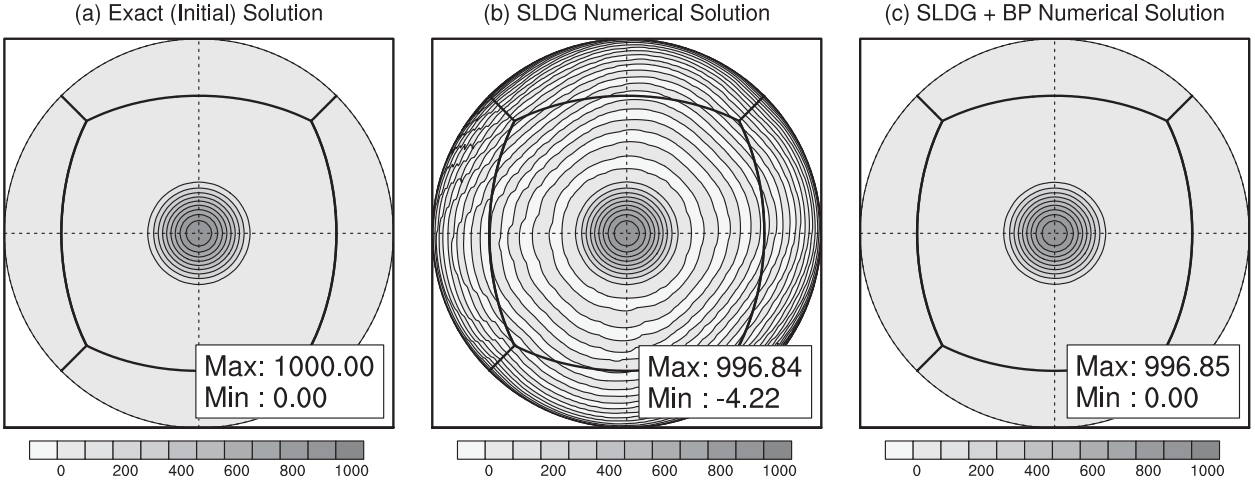

FIG. 8. An orthographic projection of the solution of the solid-body rotation of a cosine bell with $\alpha = \pi/4$. The SLDG P^3 scheme is applied on a cubed-sphere mesh $20 \times 20 \times 6$ (1.5° resolution) and time step is set as $\Delta t = 3600$ s. (a) Exact (initial) solution. (b) The numerical solution by the SLDG P^3 scheme without the BP filter. Negative height is observed. (c) The numerical solution by the SLDG P^3 scheme with the BP filter.

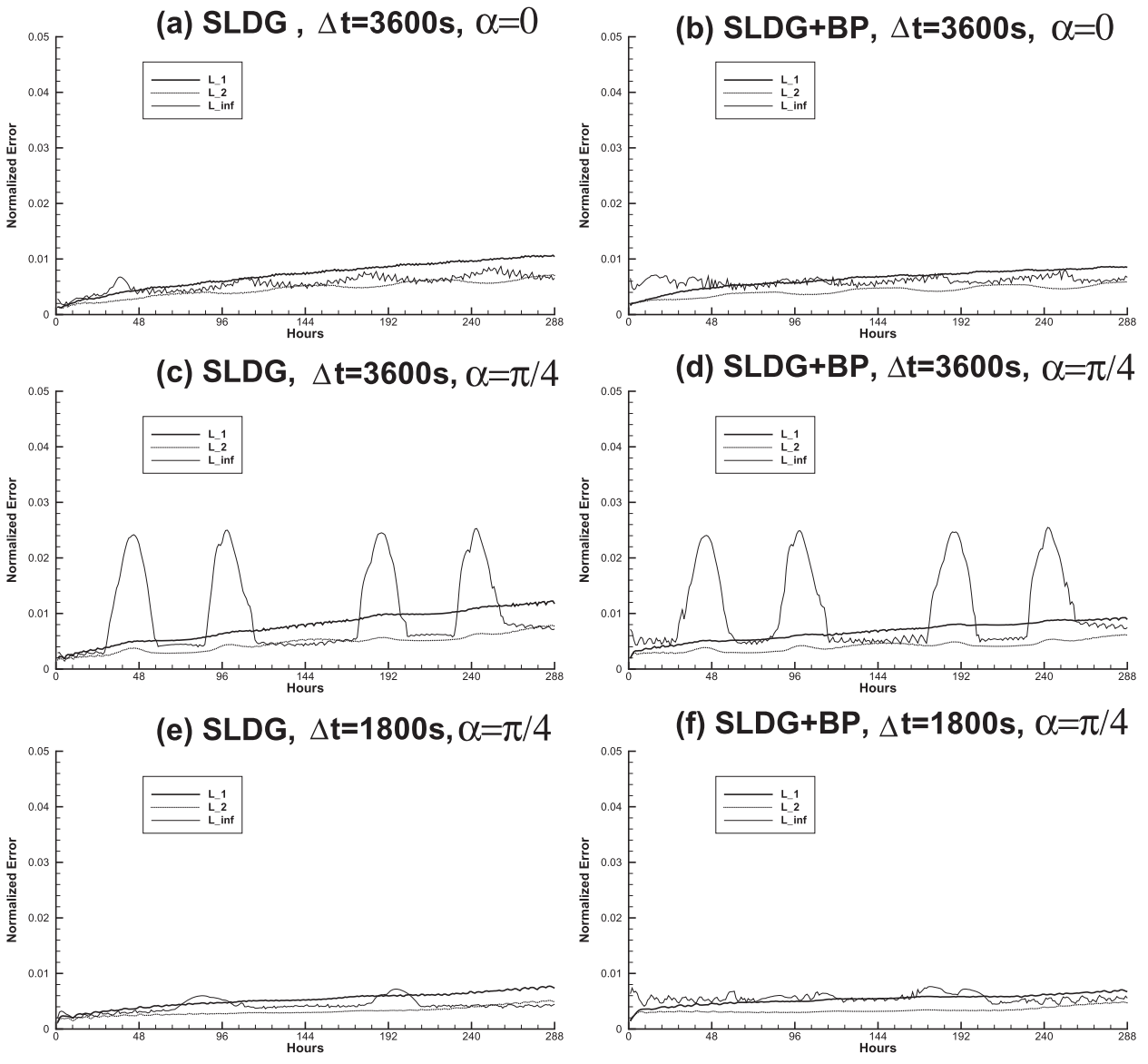

FIG. 9. The histories of error norms evolution for the solid-body rotation. The SLDG P^3 scheme is applied on a cubed-sphere mesh $20 \times 20 \times 6$ (1.5° resolution). The time step is set as (a)–(d) $\Delta t = 3600$ s and (e),(f) $\Delta t = 1800$ s. (a) Evolution of error norms for the SLDG scheme without the BP filter when $\alpha = 0$. (b) Evolution of error norms for the SLDG scheme with the BP filter when $\alpha = 0$. Comparable result is observed in (a). (c) Evolution of error norms for the SLDG scheme without the BP filter when $\alpha = \pi/4$. The l_∞ grows when the cosine bell goes through a corner, where the splitting error is larger than elsewhere. (d) Evolution of error norms for the SLDG scheme with the BP filter when $\alpha = \pi/4$. (e) Evolution of error norms for the SLDG scheme without the BP filter when $\alpha = \pi/4$ and $\Delta t = 1800$ s. The l_∞ does not excessively grow when the cosine bell goes through the corner. (f) Evolution of error norms for the SLDG scheme with the BP filter when $\alpha = \pi/4$ and $\Delta t = 1800$.

significantly when a cosine bell goes through a corner for $\alpha = \pi/4$ with $\Delta t = 3600$ s. However, the l_∞ error drops back quickly when it is away from the corner (see Figs. 9c,d). If the time step Δt is reduced to 1800 s, the l_∞ error is also reduced (see Figs. 9e,f). This indicates that the l_∞ error around corners comes from time discretization. In Fig. 10 (top), numerical error is plotted when the cosine bell approaches (left), reaches (middle), and passes (right) a corner on the cubed sphere. It is observed that error magnitude at cube edges is much larger than elsewhere. The error grows as the peak of the cosine bell approaches a corner on the cubed sphere; the error starts to drop as the peak passes the corner. We then consider a different ordering for dimensional splitting: we first evolve Eq. (17), then Eq. (15), and finally Eq. (16). Similar error patterns but with opposite signs are observed

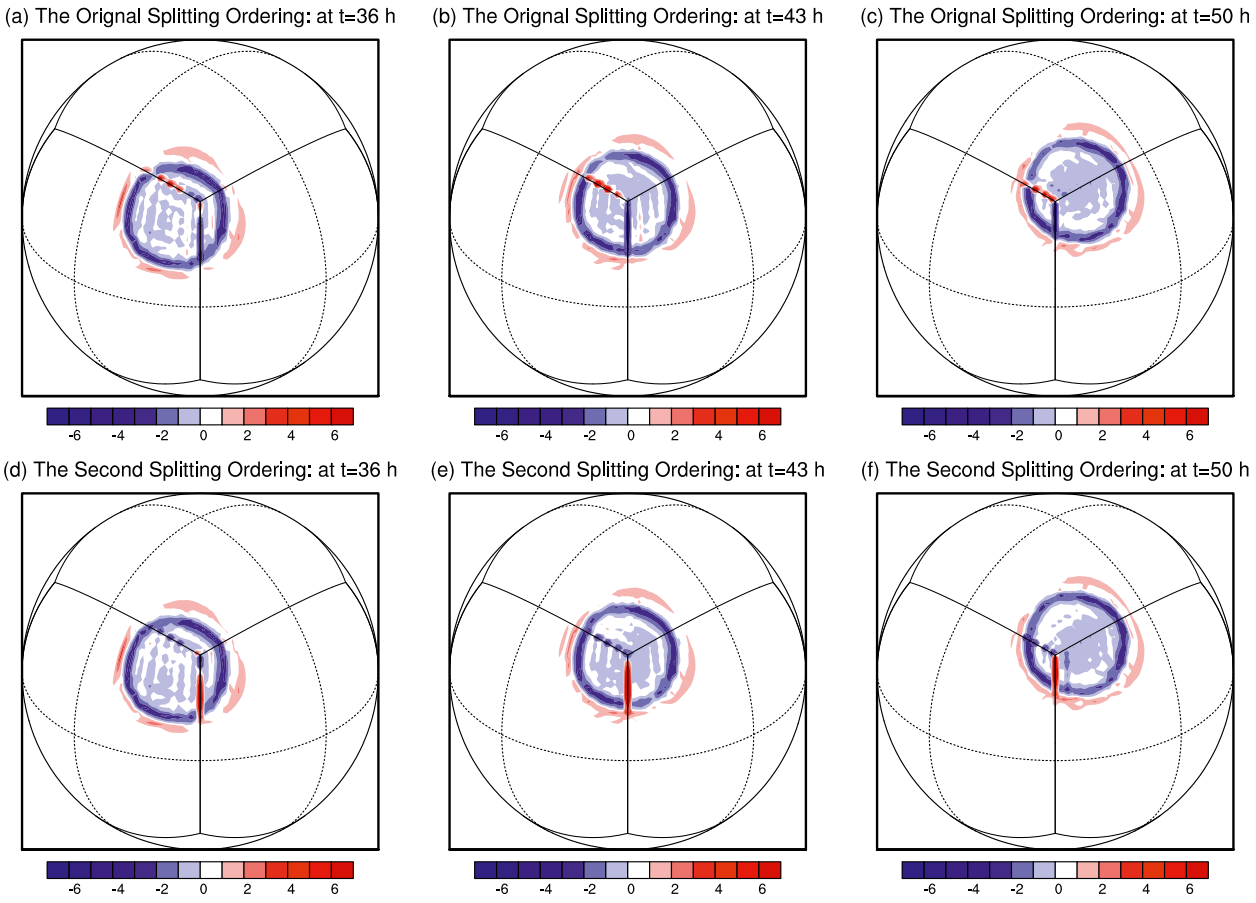

FIG. 10. The error pattern for the solid-body rotation problem when the cosine bell passes one corner. The SLDG P^3 scheme is applied on a cubed-sphere mesh $20 \times 20 \times 6$ (1.5° resolution). The time step is set as $\Delta t = 3600$ s. (a)–(c) The method using the dimensional splitting as described in section 3b. (d)–(f) The method with another ordering for dimensional splitting as described in section 4b. (a),(d) $t = 36$ h; (b),(e) $t = 43$ h; (c),(f) $t = 50$ h.

in Fig. 10 (bottom). This is an indication that such error comes from the dimensional splitting, and there exists a certain symmetry property for different dimensional-splitting orderings. Such symmetry property, together with the symmetry of the cosine bell profile, may contribute to the dropping of the l_∞ error after the cosine bell passed the corner.

We compare the proposed SLDG scheme with the RKDG scheme and the "CSLAM" scheme by Lauritzen et al. (2010), in terms of error norms. In Table 3, we show the comparison between the SLDG scheme and the RKDG scheme when P^3 is used. The mesh is set as $30 \times 30 \times 6$ ($N_e = 30$), which corresponds to approximately 1° resolution at the equator. Note that the SLDG scheme can take a very large time step but the RKDG scheme suffers from the time step restriction (1400 s is nearly the limit for time step in this case). The CPU time for the SLDG scheme is 3 times smaller than that of the RKDG scheme. We remark that the computational cost per time step of the SLDG scheme is larger than that of the RKDG scheme. It is due to the need to evaluate volume integrals in several subintervals for one element, see Eq. (8). We also remark that the SL scheme is significantly efficient for the multitracer transport because the geometric information such as upstream trajectories can be reused for each field. In spite of the larger time steps used with the SLDG scheme, the error norms are still

TABLE 3. Comparison between the RKDG scheme and the SLDG scheme in terms of error norms and CPU time when solving the solid-body rotation of a cosine bell. The mesh is set as $30 \times 30 \times 6$ corresponding to 1° resolution. DG P^3 is adopted. The numerical solutions are computed after a full rotation.

	Scheme	Time step (s)	CPU time (s)	l_2	l_∞
$\alpha = 0$	RKDG P^3	1440	9.76	1.14×10^{-2}	7.55×10^{-3}
	SLDG P^3	7200	3.32	3.70×10^{-3}	4.33×10^{-3}
	SLDG P^3	3600	6.31	2.80×10^{-3}	3.12×10^{-3}
$\alpha = \pi/4$	RKDG P^3	1440	9.76	1.21×10^{-2}	7.97×10^{-3}
	SLDG P^3	7200	3.32	1.72×10^{-2}	3.56×10^{-2}
	SLDG P^3	3600	6.31	5.15×10^{-3}	9.25×10^{-3}

TABLE 4. Comparison between the SLDG P^3 scheme and CSLAM (Lauritzen et al. 2010) in terms of error norms when solving the solid-body rotation of a cosine bell. The mesh is set as $10 \times 10 \times 6$ for SLDG scheme corresponding to 3° resolution and $32 \times 32 \times 6$ for CSLAM corresponding to 2.8125° resolution. The numerical solutions are computed after a full rotation with time step $\Delta t = 4050$ s.

	Scheme	l_1	l_2	l_∞
$\alpha = 0$	SLDG P^3	7.52×10^{-2}	4.20×10^{-2}	3.31×10^{-2}
	CSLAM	7.9×10^{-2}	4.6×10^{-2}	3.4×10^{-2}
$\alpha = \pi/4$	SLDG P^3	7.15×10^{-2}	3.66×10^{-2}	2.25×10^{-2}
	CSLAM	7.6×10^{-2}	4.1×10^{-2}	2.5×10^{-2}

comparable to those of the RKDG scheme. This shows the SLDG scheme to be more efficient when solving the solid-body rotation problem. Finally, we compare the SLDG scheme with the CSLAM (Lauritzen et al. 2010), when the horizontal resolution and the time step are comparable. For this experiment, the mesh for the SLDG P^3 scheme is set to $10 \times 10 \times 6$ ($N_e = 10$), which corresponds to 3° resolution, and a $32 \times 32 \times 6$ mesh is set for the CSLAM scheme, which corresponds to 2.8125° resolution. Note that we use a little lower-resolution mesh for the SLDG scheme. Table 4 shows the error norms of the two schemes performing the solid-body rotation of a cosine bell with $\alpha = 0$ and $\alpha = \pi/4$. The time step is set as 4050 s. It is observed that the error norms of SLDG P^3 scheme are a little smaller than those of the CSLAM. The comparison with the two popular transport schemes as discussed above shows the SLDG scheme is very competitive.

2) DEFORMATIONAL FLOW ON THE SPHERE (MOVING-VORTEX PROBLEM)

The second test is a challenging deformational flow test, the moving-vortex problem, proposed by Nair and Jablonowski (2008). The test represents the roll-up of an idealized moving atmospheric vortex such as hurricane or tropical cyclone (Hall and Nair 2013). In this test case, two vortices are generated located on the diametrically opposite sides of the sphere. The wind field is a combination of wind vectors of the solid-body rotation, which is considered in the previous case, and that of the deformational flow. The two vortices move along a great circle and the exact solution is available at any time. In a rotated coordinate system (λ', θ'), the scaled tangential velocity V_t of the vortex field is defined by

$$V_t = u_0 \frac{3\sqrt{3}}{2} \mathrm{sech}^2(\rho) \tanh(\rho),$$

where $\rho = \rho_0 \cos(\theta')$ is the radial distance of the vortex with the parameters $\rho_0 = 3$, and rotational speed $u_0 = 2\pi R/12$ (days) such that 12 days are required for a full vortex evolution. The associated angular velocity is defined to be

$$\omega(\theta') = \begin{cases} V_t/(R\rho) & \text{if} \quad \rho \neq 0, \\ 0 & \text{if} \quad \rho = 0. \end{cases}$$

The exact solution in rotated coordinates is

$$\phi(\lambda', \theta', t) = 1 - \tanh\left\{ \frac{\rho}{\gamma_0} \sin[\lambda' - \omega(\theta')t] \right\}, \quad (22)$$

where parameter $\gamma_0 = 5$. The time-dependent wind field for the moving vortex is given by

$$\begin{aligned} u(t) &= u_0(\cos\theta \cos\alpha + \sin\theta \cos\lambda \sin\alpha) \\ &\quad + R\omega(\theta')\{\sin\theta_c(t) \cos\theta \\ &\quad - \cos\theta_c(t) \cos[\lambda - \lambda_c(t)] \sin\theta\}, \\ v(t) &= -u_0 \sin\lambda \sin\alpha \\ &\quad + R\omega(\theta')\{\cos\theta_c(t) \sin[\lambda - \lambda_c(t)]\}, \end{aligned}$$

where α is the flow orientation parameter described earlier and $[\lambda_c(t), \theta_c(t)]$ is the center of one of the moving vortices, which is directed along a great circle trajectory. For the current test, the initial vortex center is located at $[\lambda_c(0), \theta_c(0)] = (3\pi/2, 0)$, which is also the location of the north pole of the rotated sphere.

We applied the SLDG P^3 scheme to the moving-vortex problem with a mesh $30 \times 30 \times 6$ (1° resolution) on the cubed-sphere geometry. The time step is set as $\Delta t = 3600$ s which is 6 times larger than that used in Zhang and Nair (2012). In Fig. 11, the evolution of numerical solution is shown for $\alpha = \pi/4$ as a series of orthographic projections centered on one of the vortices. The initial condition is shown in Fig. 11a and the numerical solution at day 3, day 6, and day 12 are shown in Figs. 11b–d, respectively. The numerical solutions are visually indistinguishable from the exact solution, which is not shown for saving space. At this resolution, the SLDG scheme resolves the fine filaments of the vortex field and is comparable to the results shown in Fig. 2 of Nair and Jablonowski (2008) and Fig. 8 of Pudykiewicz (2011). When approximated to two decimal places, the normalized standard l_1, l_2, and l_∞ errors are 4.40×10^{-4}, 1.06×10^{-3}, and 9.55×10^{-3}, respectively. The histories of error norms evolution are plotted in Fig. 12 for $\alpha = 0$ and $\alpha = \pi/4$. A similar phenomenon is observed that the l_∞ error norm grows when the vortices go through the corners, then it drops back when they are away from the corners.

We employ the SLDG scheme with high-order elements (P^6) to study its performance. For this test a mesh $15 \times 15 \times 6$ is adopted, which corresponds to 1° resolution on the cubed-sphere geometry. The time step is

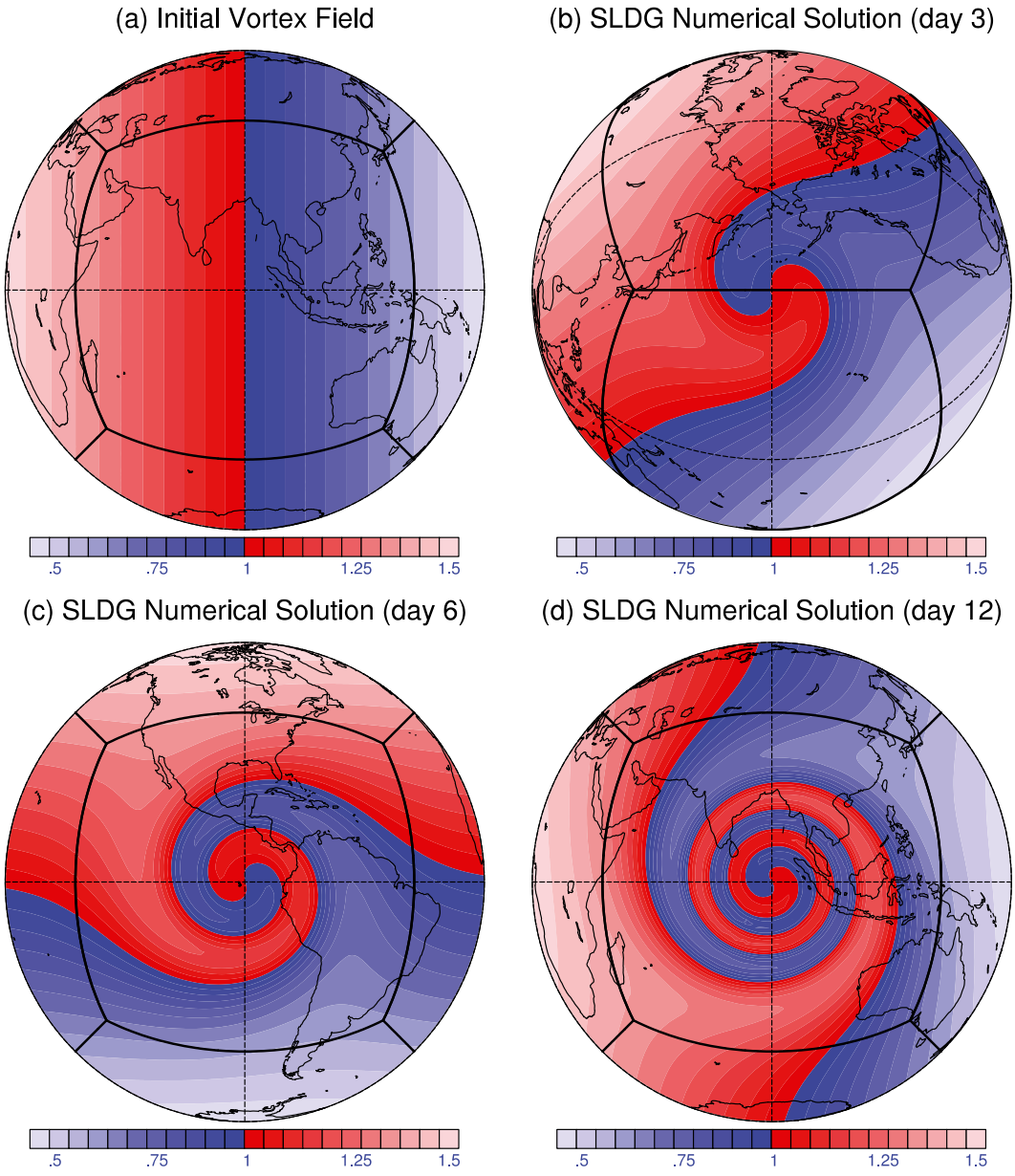

FIG. 11. An orthographic projection of the solution of moving-vortex test with $\alpha = \pi/4$. The SLDG P^3 scheme is applied on a cubed-sphere mesh $30 \times 30 \times 6$ (1° resolution) and time step is set as $\Delta t = 3600$ s. (a) Initial vortex field. (b) The numerical solution by the SLDG P^3 scheme at day 3. (c) The numerical solution by the SLDG P^3 scheme at day 6. (d) The numerical solution by the SLDG P^3 scheme at day 12. The numerical solutions are visually identical to the exact solution; therefore, it is not shown.

set to be 3600 s as before. The numerical solution is visually indistinguishable from the exact solution, so it is not shown to save space. Here we give the normalized standard l_1, l_2, and l_∞ errors after 12 model days. They are 3.53×10^{-4}, 7.28×10^{-4}, and 8.28×10^{-3}, respectively. Note that the error norms are somewhat smaller than those computed by the SLDG P^3 scheme with the same resolution; this is an expected behavior of high-order methods with smooth problems.

3) DEFORMATIONAL FLOW ON THE SPHERE (SLOTTED CYLINDER)

The last benchmark test we consider is a challenging test from a class of deformational flow tests proposed by Nair and Lauritzen (2010). This test is quite challenging for any advection scheme on the sphere because the flow filed is extremely deformational (nondivergent) with a nonsmooth initial condition. Since the analytic solution

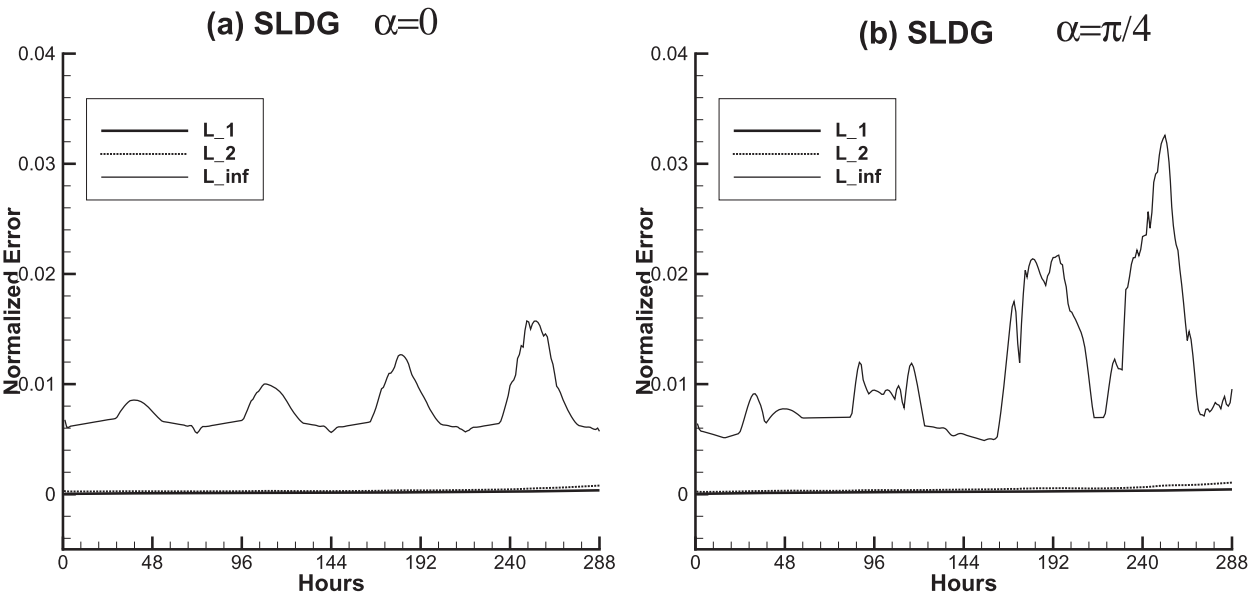

FIG. 12. The histories of error norms evolution for the moving-vortex test. The SLDG P^3 scheme is applied on a cubed-sphere mesh $30 \times 30 \times 6$ (1° resolution) and time step is set as $\Delta t = 3600$ s. Evolution of error norms for the SLDG scheme when (a) $\alpha = 0$ and (b) $\alpha = \pi/4$. The l_∞ grows when the vortices go through the edges and corners, where the the splitting error is larger than elsewhere. However, it drops back when the vortex is away from the edges or corners.

is available at the final time, errors measures can be made available for comparison.

The nondivergent wind field is defined to be

$$u(\lambda,\theta,t) = \kappa \sin^2(\lambda')\sin(2\theta)\cos(\pi t/T) + 2\pi\cos(\theta)/T,$$
$$v(\lambda,\theta,t) = \kappa \sin(\lambda')\cos(\theta)\cos(\pi t/T),$$

where $\lambda' = \lambda - 2\pi t/T$, $\kappa = 2$, and $T = 5$ units. Note that the wind field is a combination of a deformational field and a zonal background flow, avoiding the possible cancellations of errors due to the reversal of the flow along the same flow path after the half time $T/2$ (see Nair and Lauritzen 2010). The initial condition is the twin slotted-cylinder defined by

$$\psi(\lambda,\theta) = \begin{cases} c & \text{if} \quad r_i \le r \quad \text{and} \quad |\lambda - \lambda_i| \ge r/6 \quad \text{for} \quad i = 1,2, \\ c & \text{if} \quad r_1 \le r \quad \text{and} \quad |\lambda - \lambda_1| < r/6 \quad \text{and} \quad \theta - \theta_1 < -\dfrac{5}{12}r, \\ c & \text{if} \quad r_2 \le r \quad \text{and} \quad |\lambda - \lambda_2| < r/6 \quad \text{and} \quad \theta - \theta_2 > \dfrac{5}{12}r, \\ b & \text{otherwise} \end{cases} \tag{23}$$

where $c = 1$, $b = 0$, $r = 1/2$ is the radius of the cylinder, and $r_i = r_i(\lambda,\theta)$ is the great-circle distance between (λ,θ) and a specified center (λ_i, θ_i) of the cylinder, which is given by

$$r_i(\lambda,\theta) = \arccos[\sin\theta_i \sin\theta + \cos\theta_i \cos\theta \cos(\lambda - \lambda_i)].$$

The centers of the initial distribution are located at $(\lambda_1,\theta_1) = (5\pi/6, 0)$ and $(\lambda_2,\theta_2) = (7\pi/6, 0)$, respectively. The slots are oriented in opposite directions for the two cylinders so that they are symmetric with respect to the flow. The wind field and initial distributions are defined in nondimensional units on the unit sphere ($R = 1$). Note that the distribution is stretched into thin filaments and at half time $T/2$ while they are being transported along the zonal direction by the background flow. The exact solution is only available at the final time $t = T$, which is identical to the initial condition.

We apply the SLDG P^3 scheme with the BP filter to the deformational flow problem with a mesh $30 \times 30 \times 6$ corresponding to 1° resolution at the equator for the cubed-sphere geometry. The time step is set as $\Delta t = T/800$ such that it takes 800 steps to complete a full evolution. Note that Δt is chosen as 5 times larger than that used for the RKDG P^2 scheme in Zhang and Nair (2012). Figure 13 shows the initial condition (Fig. 13a),

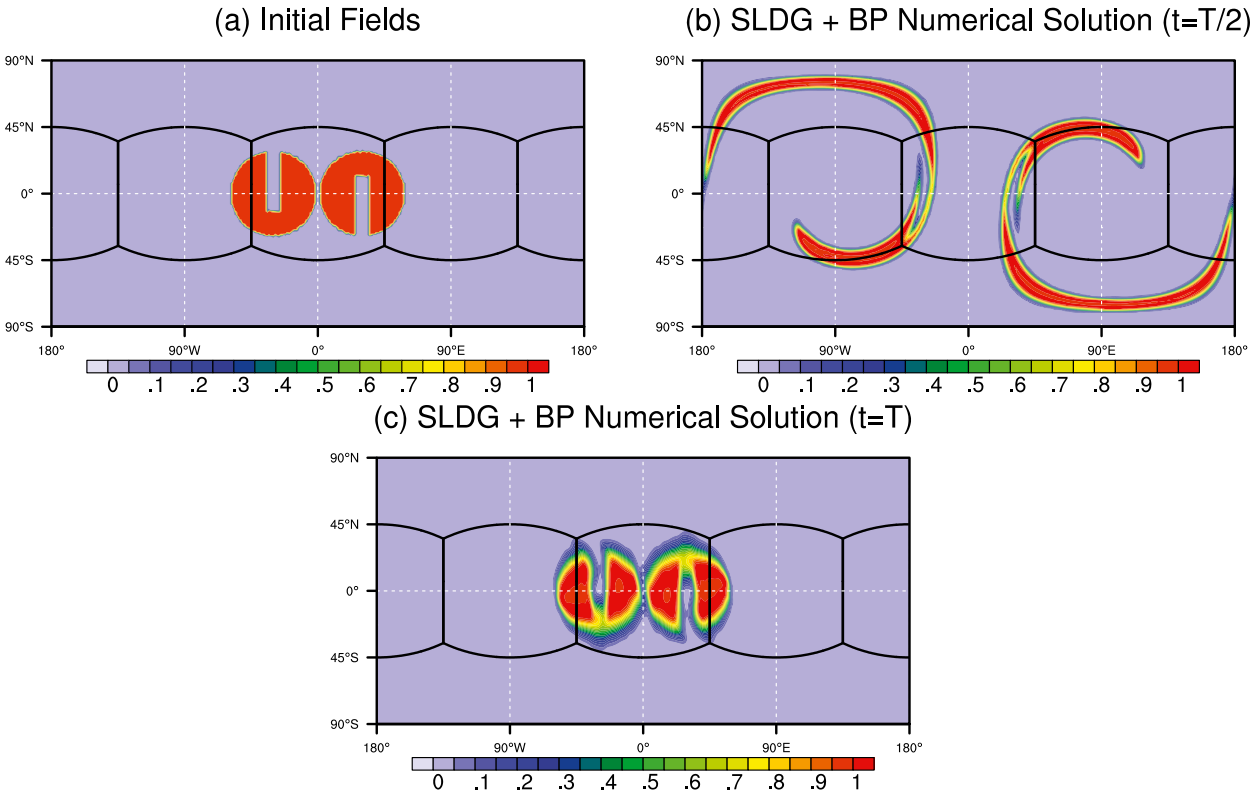

FIG. 13. Numerical solution for the deformational flow test with twin slotted cylinder as the initial condition. The SLDG P^3 scheme with BP filter is applied on a cubed-sphere mesh $30 \times 30 \times 6$ (1° resolution) and time step is set as $t = T/800$. (a) Initial condition: twin slotted-cylinder. (b) The numerical solution by the SLDG P^3 at half time $t = T/2$. The thin filaments are resolved. (c) The numerical solution by the SLDG P^3 at the final time $T = 5$. The slotted cylinders are captured by the SLDG scheme. The value of T is set as 5. Also note that, the numerical solution is exactly positivity preserving.

the numerical solution at half time $t = T/2$ (Fig. 13b) and final time $t = T$. It is clearly observed that the SLDG scheme resolves the very thin filament solution structures at half time and the original shape of the twin slotted cylinder at final time is also captured. Moreover, the numerical solution preserves the positivity exactly and compares to that reported in the Zhang and Nair (2012).

We consider a variant of this test by changing the initial condition where the non-smooth twin slotted cylinder is replaced by two symmetrically located quasi-smooth cosine bells defined as follows (Nair and Lauritzen 2010):

$$\psi(\lambda,\theta) = \begin{cases} b + ch_1(\lambda,\theta) & \text{if} \quad r_1 < r, \\ b + ch_2(\lambda,\theta) & \text{if} \quad r_2 < r, \\ b & \text{otherwise}, \end{cases} \tag{24}$$

where $c = 0.9$, $b = 0.1$, and

$$h_i(\lambda,\theta) = \frac{1}{2}[1 + \cos(\pi r_i/r)] \quad \text{if} \quad r_i < r, \quad \text{for} \quad i = 1, 2.$$

Other parameters are the same as those used for the slotted-cylinder case. Note that the initial condition is quasi-smooth (C^1 smoothness). We want to use this test case to compare the proposed SLDG scheme with the CSLAM scheme in terms of error norms. The mesh is set as $20 \times 20 \times 6$ corresponding to 1.5° resolution at the equator and the time step is set as $\Delta t = T/600$. In Table 5, we show the l_1, l_2, and l_∞ error norms of the SLDG P^3 scheme and those of CSLAM reported in Nair and Lauritzen (2010) with the same resolution and time step. It is observed that the l_1, l_2, and l_∞ error norms by SLDG are approximately ⅔–¾ of those by CLSAM. We conclude that when the CFL number approximately equals 1 and the same resolutions are considered, the errors by the SLDG scheme are smaller than those by CSLAM when solving the deformational flow problem.

5. Summary and conclusions

In the paper, a SLDG transport scheme has been developed on the Cartesian domain and extended to the cubed-sphere geometry. The scheme is inherently

TABLE 5. Comparison between the SLDG P^3 scheme and the CSLAM scheme Lauritzen et al. (2010) in terms of error norms when solving the deformational flow of two symmetrically located cosine bells. The mesh is set as $20 \times 20 \times 6$ for the SLDG scheme corresponding to 1.5° resolution and $60 \times 60 \times 6$ for CSLAM corresponding to the same 1.5° resolution. The result of the CSLAM scheme is from Nair and Lauritzen (2010). The numerical solutions are computed after a full rotation with time step $\Delta t = T/600$.

Scheme	l_1	l_2	l_∞
SLDG P^3	0.0393	0.0673	0.1109
CSLAM	0.0533	0.1088	0.1421

conservative with positivity preserving (or bound preserving) property. The main advantage of the SLDG scheme is that it can take an arbitrary time step without a stability issue. Such a property makes the proposed SLDG scheme more efficient than the well-known Runge–Kutta discontinuous Galerkin (RKDG) scheme when solving the transport equations. A set of benchmark tests were performed to demonstrate the robustness of the proposed SLDG scheme on the Cartesian plane and sphere. The comparison between the SLDG scheme and another two popular global transport schemes including the RKDG scheme and Conservative Semi-Lagrangian Multi-Tracer Transport Scheme (CSLAM) was performed.

Multitracer transport schemes are essential for practical climate models where hundreds of tracers species need to be advected. In a computational point of view, the semi-Lagrangian transport scheme has several advantages in this context, because the upstream trajectory information can be efficiently reused for each tracer, and monotonicity (or positivity) conditions can be enforced. For new generation models based on highly scalable spectral-element and discontinuous Galerkin methods, application of multitracer scheme is not obvious because of the highly nonuniform Gaussian quadrature grids (elements) they use. A possible option is to employ finite-volume semi-Lagrangian methods such as the CSLAM scheme on spectral-element grids. However, this necessitates a dual finite-volume overlaid grid combined with grid-to-grid remapping. The proposed SLDG scheme has the potential to address this issue because the scheme is developed on the native spectral-element grid and no dual grid is required. In the context of multitracer transport on the sphere, the proposed P^3 SLDG scheme is an efficient alternative for the Eulerian (RKDG) transport scheme and qualitatively comparable or better than the semi-Lagrangian-based CSLAM scheme.

Our approach for the SLDG scheme relies on the Strang-type splitting, which is subject to splitting errors. For the cubed-sphere geometry splitting error is further exacerbated by the grid discontinuity at the cube-sphere edges resulting from the *patched* mapping. We are currently investigating a way to avoid splitting approach, and reduce the error norms at the cubed-sphere edges. In the future, we plan to extend the SLDG scheme to more general cases, such as the Euler system and shallow-water equations.

Acknowledgments. The authors are thankful for Dr. Richard Loft (NCAR) for the SIParCS internship support for Wei Guo. R. Nair is partially supported by the DOE BER Program DE-SC0001658. W. Guo and J.-M. Qiu are partially supported by Air Force Office of Scientific Computing YIP Grant FA9550-12-0318, NSF Grant DMS-0914852, and DMS-1217008, University of Houston.

REFERENCES

Blossey, P., and D. Durran, 2008: Selective monotonicity preservation in scalar advection. *J. Comput. Phys.,* **227** (10), 5160–5183.

Childs, P., and K. Morton, 1990: Characteristic Galerkin methods for scalar conservation laws in one dimension. *SIAM J. Numer. Anal.,* **27** (3), 553–594.

Cockburn, B., and C.-W. Shu, 2001: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.,* **16** (3), 173–261.

Dennis, J., and Coauthors, 2011: CAM-SE: A scalable spectral-element dynamical core for the Community Atmosphere Model. *Int. J. High Perform. Comput. Appl.,* **26,** 74–89.

Erath, C., and R. D. Nair, 2014: A conservative multi-tracer transport scheme for spectral-element spherical grids. *J. Comput. Phys.,* **256**, 118–134, doi:10.1016/j.jcp.2013.08.050.

Giraldo, F., and M. Restelli, 2008: A study of spectral element and discontinuous Galerkin method for the Navier–Stokes equations in nonhydrostatic mesoscale modeling: Equation sets and test cases. *J. Comput. Phys.,* **227,** 3847–3877.

Gottlieb, D., 1972: Strang-type difference schemes for multidimensional problems. *SIAM J. Numer. Anal.,* **9** (4), 650–661.

Hall, D. M., and R. D. Nair, 2013: Discontinuous Galerkin transport on the spherical Yin–Yang overset mesh. *Mon. Wea. Rev.,* **141,** 264–282.

Lauritzen, P., R. Nair, and P. Ullrich, 2010: A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *J. Comput. Phys.,* **229** (5), 1401–1424.

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

——, and P. Lauritzen, 2010: A class of deformational flow test cases for linear transport problems on the sphere. *J. Comput. Phys.,* **229** (23), 8868–8887.

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

——, H.-W. Choi, and H. M. Tufo, 2009: Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core. *Comput. Fluids,* **38,** 309–319.

Pudykiewicz, J., 2011: On numerical solution of the shallow water equations with chemical reactions on icosahedral geodesic grid. *J. Comput. Phys.,* **230** (5), 1956–1991.

Qiu, J., and C. Shu, 2011: Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov–Poisson system. *J. Comput. Phys.,* **230** (23), 8386–8409.

Restelli, M., L. Bonaventura, and R. Sacco, 2006: A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *J. Comput. Phys.,* **216,** 195–215.

Ronchi, C., R. Iacono, and P. Paolucci, 1996: The cubed sphere: A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.,* **124,** 93–114.

Rossmanith, J., and D. Seal, 2011: A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov–Poisson equations. *J. Comput. Phys.,* **230,** 6203–6232.

Russell, T., and M. Celia, 2002: An overview of research on Eulerian–Lagrangian localized adjoint methods (ELLAM). *Adv. Water Resour.,* **25** (8), 1215–1231.

Sadourny, R., 1972: Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.,* **100,** 136–144.

Strang, G., 1968: On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.,* **5** (3), 506–517.

Tumolo, G., L. Bonaventura, and M. Restelli, 2013: A semi-implicit, semi-Lagrangian, *p*-adaptive discontinuous Galerkin method for the shallow water equations. *J. Comput. Phys.,* **232,** 46–67.

Williamson, D., J. Drake, J. Hack, R. Jakob, and P. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.,* **102,** 211–224.

Zhang, X., and C.-W. Shu, 2010a: On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.,* **229,** 3091–3120.

——, and ——, 2010b: On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *J. Comput. Phys.,* **229,** 8918–8934.

Zhang, Y., and R. D. Nair, 2012: A nonoscillatory discontinuous Galerkin transport scheme on the cubed sphere. *Mon. Wea. Rev.,* **140,** 3106–3126.