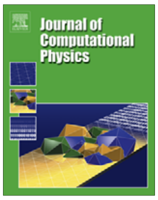
Journal of Computational Physics 229 (2010) 1401–1424

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid ☆

Peter H. Lauritzen [a,*], Ramachandran D. Nair [b], Paul A. Ullrich [c]

[a] Climate and Global Dynamics Division, National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA
[b] Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA
[c] Department of Atmospheric, Oceanic and Space Sciences, University of Michigan, Space Research Building, University of Michigan, 2455 Hayward Street, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 16 March 2009
Received in revised form 21 October 2009
Accepted 23 October 2009
Available online 1 November 2009

Keywords:
Advection
Conservation
Cubed-sphere
Finite-volume
Semi-Lagrangian
Monotonicity
Remapping
PPM
Multi-tracer transport

ABSTRACT

A conservative multi-tracer transport algorithm on the cubed-sphere based on the semi-Lagrangian approach (CSLAM) has been developed. The scheme relies on backward trajectories and the resulting upstream cells (polygons) are approximated with great-circle arcs. Biquadratic polynomial functions are used for approximating the density distribution in the cubed-sphere grid cells. The upstream surface integrals associated with the conservative semi-Lagrangian scheme are computed as line-integrals by employing the Gauss–Green theorem. The line-integrals are evaluated using a combination of exact integrals and high-order Gaussian quadrature. The upstream cell (trajectories) information and computation of weights of integrals can be reused for each additional tracer.

The CSLAM scheme is extensively tested with various standard benchmark test cases of solid-body rotation and deformational flow in both Cartesian and spherical geometry, and the results are compared with those of other published schemes. The CSLAM scheme is accurate, robust, and moreover, the edges and vertices of the cubed-sphere (discontinuities) do not affect the overall accuracy of the scheme. The CSLAM scheme exhibits excellent convergence properties and has an option for enforcing monotonicity. The advantages of introducing cross-terms in the fully two-dimensional biquadratic density distribution functions are also examined in the context of Cartesian as well as the cubed-sphere grid which has six local sub-domains with discontinuous edges and corners.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

The transport problem in computational fluid dynamics can either be cast in Lagrangian, Eulerian or in Arbitrary Lagrangian–Eulerian (ALE) form [1]. Lagrangian methods let the mesh travel and evolve with the fluid throughout the integration whereas Eulerian methods use a fixed mesh. Both methods have their strengths and weaknesses. The ALE method was developed in an attempt to combine the advantages of the Eulerian and the Lagrangian approaches by letting the mesh move in any prescribed manner as an extra independent degree of freedom. A popular choice of prescribed mesh movement is to run in Lagrangian mode for one time-step and then regrid (interpolate) back to the static and regular (Eulerian) mesh. In meteorological literature this approach is known as the semi-Lagrangian method [2]. A comprehensive review of conservative semi-Lagrangian methods are given in [3,4], and a stability analysis of these schemes is presented in [5].

☆ The National Center for Atmospheric Research is sponsored by the National Science Foundation.
* Corresponding author. Tel.: +1 303 497 1316; fax: +1 303 497 1324.
E-mail address: pel@ucar.edu (P.H. Lauritzen).
URL: http://www.cgd.ucar.edu/cms/pel/ (P.H. Lauritzen).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.036

At every time-step, the semi-Lagrangian approach involves regridding (interpolating) quantities from a distorted Lagrangian mesh to a regular Eulerian mesh or vice versa, depending on the trajectories. Hence the transport problem is reduced to a regridding problem if the Lagrangian mesh movement is prescribed. For a variety of reasons it is desirable that the regridding procedure is conservative and monotonic. Conservative regridding is often referred to as remapping or rezoning. The problem of remapping quantities between arbitrary grids, which involves integration over overlapping areas between the grids, has received considerable attention in the literature due to its many applications. In general direct integration over arbitrary overlap areas is not practical. Through the pioneering work of Dukowicz [6,7] and Ramshaw [8] the remapping problem has been made practical by the application of Gauss–Green's theorem which converts area-integrals into line-integrals. This approach has been applied for up to second-order static grid-to-grid remapping in [9] and later the method was extended to third-order and optimized for the regular latitude–longitude and cubed-sphere grids [10].

In most atmosphere and ocean modeling applications the continuity equation must be solved multiple times for fluid density as well as dozens of tracers (chemical species). For example, the chemistry version of NCAR's Community Atmospheric Model (CAM) model [11] uses on the order of 100 prognostic tracers [12]. Therefore it is highly desirable that the numerical algorithm used for tracer transport is efficient and adaptable for a large number of tracers. In [13] an incremental remapping algorithm based on the semi-Lagrangian technique has been introduced for multi-tracer transport. Although incremental remapping has a high startup cost associated with geometry calculations, each additional tracer adds only a relatively small cost. The CSLAM algorithm considered herein follows this strategy.

Traditionally the regular latitude–longitude grid has been the preferred choice for global atmospheric models. However, models based on such grid system may have scalability issues. The scalability problems are either rooted from the non-scalable global numerical methods or the application of non-local polar filters. To address these problems, the atmospheric modeling community is developing numerical models based on more isotropic spherical grid systems that are free from singularities or contain weaker singularities. Also these grid geometries are amenable to *local* numerical methods such as the finite-volume method or element-based high-order Galerkin methods. The cubed-sphere geometry introduced by Sadourny [14] offers many computationally attractive features. Recently the cubed-sphere (spherical cube or expanded cube) geometry has been reintroduced in [15,16] with additional desirable features such as the equi-angular grid-spacing or orthogonality. Here we consider cubed-sphere grids based on the central (gnomonic) projection.

In this paper we optimize the more general method of Dukowicz [6,7] for transport on the cubed-sphere grid in two ways. Firstly, instead of using constant cell densities as in [6,8] or linear reconstructions of cell densities as in [7,9,17–19], we use the fully two-dimensional biquadratic reconstruction functions with a monotone option. Secondly, we exploit that for the gnomonic cubed-sphere grid it is possible to evaluate line-integrals along coordinate lines exactly [10]. Contrary to the incremental remapping algorithm, CSLAM is designed to allow for long time-steps with Courant numbers exceeding unity.

This paper is organized as follows. In Section 2 we introduce the CSLAM algorithm in Cartesian geometry. This involves defining the transport problem and introduce the notation required to mathematically describe the Lagrangian grid, in particular, the overlap regions between the static mesh and the Lagrangian grid. The conversion of area-integrals into line-integrals using Gauss–Green's theorem is described with details including the analytic integration of two-dimensional polynomial reconstruction functions. In Section 3, CSLAM is extended to the cubed-sphere geometry. Section 4 show results for standard test cases in Cartesian and spherical geometry. We will summarize the findings in Section 5.

2. Cartesian geometry

The two-dimensional transport equation for a tracer, in the absence of sources or sinks, can be written as

$$\frac{d}{dt}\int_{A(t)}\psi\,dA=0, \tag{1}$$

(e.g., [13]) where ψ is the density (typically the product of the air density and the tracer concentration per unit mass), and the integration is over an arbitrary Lagrangian area $A(t)$ at time t, that is, an area that moves with the flow with no flux through its boundaries. A temporal discretization of (1) along the characteristics is

$$\int_{A(t+\Delta t)}\psi\,dA=\int_{A(t)}\psi\,dA, \tag{2}$$

where Δt is the time-step size.

In a semi-Lagrangian method either $A(t+\Delta t)$ or $A(t)$ is a static grid cell, or equivalently, either upstream (backward trajectories) or downstream (forward trajectories) cell tracking is used. Here we use the upstream approach so that $A(t+\Delta t)$ is a regular grid cell. Using the two-time level semi-Lagrangian terminology, $A(t+\Delta t)$ is referred to as the *arrival* (or Eulerian) cell and $A(t)$ the *departure* (or Lagrangian) cell.

In a two-dimensional Cartesian orthogonal grid system, let A_k be the kth (Eulerian) grid cell, where $k=1,\ldots,N$, such that N is the total number of cells in the domain Ω. The departure cell corresponding to the arrival cell A_k is denoted by a_k (see Fig. 1). Note that there exists a one-to-one correspondence between departure and arrival cells such that the departure cells span Ω without gaps or overlaps between them,

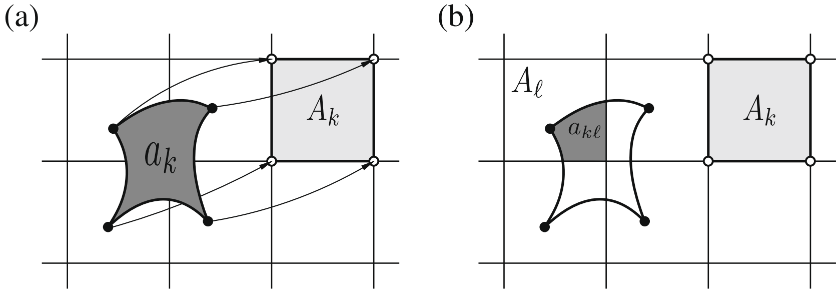


Fig. 1. A schematic illustration of concepts used in the semi-Lagrangian finite-volume scheme. (a) The deformed departure cell a_k (dark shaded area) ends up, after being transported by the flow for one time-step, at the regular arrival cell A_k (light shaded area). The trajectories for the cell vertices are shown with arrows, and the departure and arrival cell vertices are marked with filled and open circles, respectively. (b) Illustrates the overlap region between the grid cell A_ℓ and the departure cell a_k referred to as $a_{k\ell}$ used for the upstream integral computation given in Eq. (4).

$$\bigcup_{k=1}^{N} a_k = \Omega, \quad \text{and} \quad a_k \cap a_\ell = \emptyset, \quad \forall k \neq \ell,$$

where a_k is assumed to be a simply connected region on Ω. For a cell-integrated semi-Lagrangian method, as the one considered here, it is required that the trajectories should not cross during the time-step Δt [20].

For the discretization of (2) we need to define the overlap regions between the departure cell a_k and the Eulerian grid cells $A_\ell,\ \ell = 1, \ldots, N$. Let $a_{k\ell}$ be the non-empty overlap region between departure cell a_k and grid cell A_ℓ such that

$$a_{k\ell} = a_k \cap A_\ell, \quad a_{k\ell} \neq \emptyset; \quad \ell = 1, \ldots, L_k, \quad \text{and} \quad 1 \leqslant L_k \leqslant N,$$

where L_k is the number of non-empty overlap regions between departure cell a_k and the Eulerian grid cells. L_k depends on the characteristics of the flow and time-step size.

The semi-Lagrangian finite-volume version of the discretized transport Eq. (1) for ψ can be written as follows:

$$\bar{\psi}_k^{n+1} \Delta A_k = \overline{\psi_k^{*n}} \, \delta a_k \tag{3}$$

(e.g., [21]) where $\overline{\psi}_k^{n+1}$ is the average tracer density in cell k at time-level $n+1$ (i.e., $t = (n+1)\Delta t$); ΔA_k and δa_k is the area of the arrival and departure cell A_k and a_k, respectively, and $\overline{\psi_k^{*n}}$ is the average density in the departure cell.

To compute the mass in the departure cell from known cell average values $\bar{\psi}_\ell^n,\ \ell = 1, \ldots, N$, in a higher-order and conservative manner, one needs to construct a continuous sub-grid-scale representation of ψ within each Eulerian cell with mass-conservation as a constraint. The sub-grid-scale reconstruction in a cell ℓ is denoted $f_\ell(x, y)$. The integral over the departure cell can be broken up into the sum of integrals of $f_\ell(x, y)$ over non-empty overlap regions $a_{k\ell}$ as follows:

$$\overline{\psi_k^{*n}} = \frac{1}{\delta a_k} \sum_{\ell=1}^{L_k} \int\!\!\int_{a_{k\ell}} f_\ell(x, y)\, dA. \tag{4}$$

Note that no approximations have been made at this point.

Since the departure cells a_k span the integration domain Ω without gaps or overlaps global mass is conserved as long as the reconstructions $f_\ell(x, y)$ satisfy

$$\int\!\!\int_{A_\ell} f_\ell(x, y)\, dA = \bar{\psi}_\ell \, \Delta A_\ell \quad \text{for } \ell = 1, \ldots, N.$$

For general (smooth) flows the boundary of the departure cells are smooth curves rather than straight line segments, as is the case for the arrival cell walls. Only in simple cases such as for pure translational (non-divergent) wind fields the analytic departure cell boundaries consist of straight line segments but in general the departure cell sides must be approximated. To address this problem several approaches have been taken in the literature (see Fig. 2.10 in [3] and Fig. 2 in [22] for illustrations). Most methods track cell vertices moving with the flow and approximate the departure cell sides from the location of these vertices. Probably the most straight forward cell approximation results from connecting the cell vertices with straight lines (Fig. 2(a)).

To improve the representation of any particular Lagrangian cell edge one may approximate it with piecewise straight lines, that is, introduce more Lagrangian parcels along the cell sides and connect them with straight lines. By increasing the number of points tracked along each cell side one would converge towards the analytic departure cell (see Fig. 2). It is beyond the scope of this paper to investigate such an approach, that is, we simply approximate the cell sides with straight lines connecting the vertices of the departure cell. Hence the region a_k is a quadrilateral.

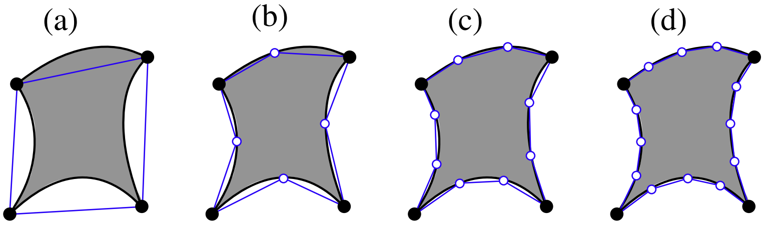


Fig. 2. Schematic illustrations of possible approximations to the analytical departure cell boundary (solid curved line) using different levels of refinement with piecewise straight lines. (a) The approach used in this paper connects the four vertices of the departure cell (filled circles) with straight lines. To improve the approximation to the departure cell one may introduce (b) one, (c) two or (d) three Lagrangian points along the cell sides (unfilled circles) and connect these by straight line segments to converge towards the exact departure cell boundary.

2.1. Upstream integrals

The sub-domains $a_{k\ell}$ over which must be integrated can have many possible shapes (Fig. 3). The practical difficulty in developing analytical integrals that cover all possible cases is, in general, somewhat complicated but not impossible [23]. Instead the problem can be greatly simplified by converting the area-integrals into line-integrals by appropriate use of the Gauss–Green theorem [6].

2.1.1. Lagrangian cell boundary computation (search algorithm)

Suppose the trajectories for the vertices of a_k are given. Finding the location of the vertices of $a_{k\ell}$ basically reduces to the computation of intersections between coordinate lines (sides of A_ℓ) and lines of arbitrary orientation (sides of $a_{k\ell}$). Only three intersection scenarios are possible when marching counter-clockwise along a side of $a_{k\ell}$: Intersection with a horizontal coordinate line (Fig. 4(a)), intersection with a vertical coordinate line (Fig. 4(b)) or intersection with a vertex of A_ℓ (Fig. 4(c)). The coordinates of the crossing are simply the location of the intersection between straight lines. Let N_h be the number of vertices of $a_{k\ell}$. The coordinates of the vertices of the polygon $a_{k\ell}$ are denoted $(x_{k\ell,h}, y_{k\ell,h})$, $h = 1, \ldots, N_h$, and are numbered counter-clockwise (Fig. 5). The first subscript k refers to the kth departure cell to which $a_{k\ell}$ belongs, ℓ refers to the fact that $(x_{k\ell,h}, y_{k\ell,h})$ is a vertex in the grid cell A_ℓ and h is the local index for the numbered vertices of $a_{k\ell}$.

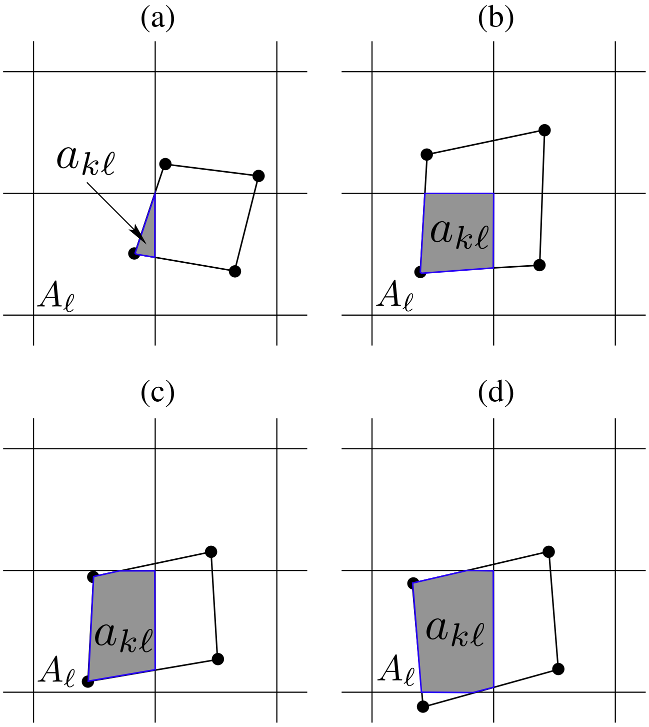


Fig. 3. A schematic illustration of some of the possible shapes the polygons $a_{k\ell}$ (shaded areas) may take depending on the location of the departure points (filled circles). The number of vertices can be (a) 3, (b) 4, (c) 5, (d) 6 and even more depending on the flow and time-step.

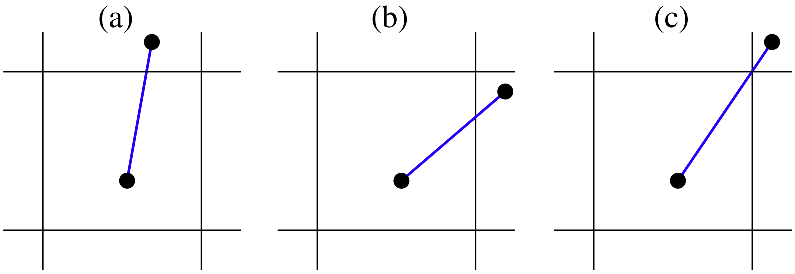


Fig. 4. A schematic illustration of the three possible intersections between a departure cell side and the coordinate lines. In (a) and (b) a horizontal and vertical grid line is intersected, respectively, and in (c) a vertex of an Eulerian cell is intersected. The resulting line segment is one of the sides of the polygon $a_{k\ell}$ that defines the overlap area between Eulerian cell A_k and departure cell a_k.

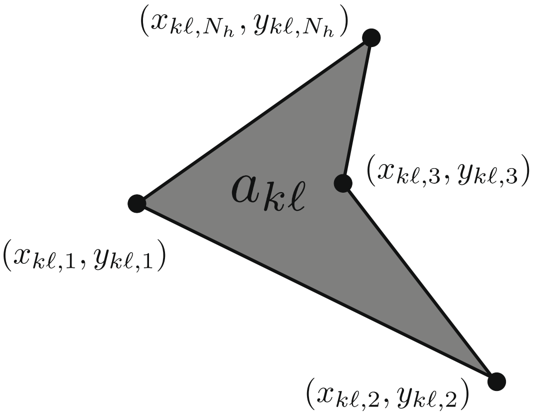

$(x_{k\ell,N_h}, y_{k\ell,N_h})$
$(x_{k\ell,3}, y_{k\ell,3})$
$a_{k\ell}$
$(x_{k\ell,1}, y_{k\ell,1})$
$(x_{k\ell,2}, y_{k\ell,2})$

Fig. 5. A schematical illustration of the coordinates and numbering of the boundary of $a_{k\ell}$ with four vertices ($N_h = 4$).

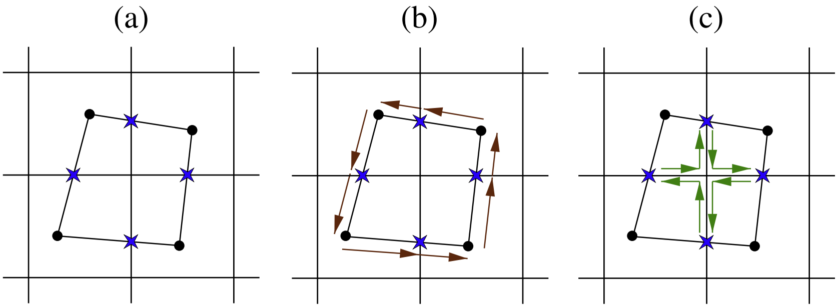


Fig. 6. A schematic illustration of the search algorithm used to define the overlap regions $a_{k\ell}$ between the departure cell (vertices marked with filled circles) and the Eulerian cells A_ℓ. (a) First the intersections between Eulerian grid lines and the sides of the departure cell a_k are computed (crossings marked with stars). (b) The line segments along the sides of the departure cell a_k, referred to at *outer* line segments (indicated with arrows), are stored. (c) Thereafter the *inner* line segments along the coordinate lines and enclosed in a_k are stored (indicated with arrows).

The algorithm works as follows:

1. Compute all intersections between grid lines and the four sides of the departure cell a_k by marching counter-clockwise along the sides of the departure cell (Fig. 6(a)).
2. All segment coordinates along the sides of the departure cell a_k are temporarily stored as well as the index of the Eulerian cell A_ℓ in which the line-segment is located. We refer to these segments as *outer* line-segments (Fig. 6(b)). Note that the computation of the index of the Eulerian cell A_ℓ in which the segment is located does not need an extensive search algorithm since when marching along the sides of the departure cell we move between adjacent cells (Fig. 6(b)).

3. The coordinate line crossings can also be used to define the line-segments along coordinate lines enclosed by a_k (Fig. 6(c)) by marching along the coordinate line on which a crossing occurs until the next crossing is encountered. We refer to these lines as *inner* line segments. Note that a_k is a quadrilateral so the departure cell sides can only cross a particular coordinate line once. Marching along all coordinate lines that are intersected by a side of a_k defines all the *inner* line segments. The line-segment coordinates are registered counter-clockwise so if a line-segment is oriented from left to right in a particular cell the line-segment is registered from right to left in the cell above. As for the *outer* segments the index of the Eulerian cell in which the segment belongs is temporarily stored.
4. The *outer* and *inner* line-segments complete the definition of $a_{k\ell}$. Special attention must be given to situations in which the *inner* and *outer* line segments coincide. In such a situation the segment must, of course, only be registered once.

This defines the boundary of all the overlap polygons $a_{k\ell}$ which will be used for computing the line-integrals.

2.1.2. Converting area-integrals into line-integrals (weights)

In order to evaluate the upstream integrals over the Lagrangian cells efficiently we employ Gauss–Green's theorem: for the simply connected regions $a_{k\ell}$ the following integral equation holds:

$$\int\int_{a_{k\ell}} f_\ell(x,y)\,dx\,dy = \oint_{\partial a_{k\ell}} [P\,dx + Q\,dy], \tag{5}$$

where $\partial a_{k\ell}$ is the boundary of $a_{k\ell}$. The functions $P = P(x,y)$ and $Q = Q(x,y)$ are chosen such that they satisfy:

$$-\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} = f_\ell(x,y).$$

In general, a third-order polynomial reconstruction function in Eulerian cell A_ℓ can be written as

$$f_\ell(x,y) = \sum_{i+j\leqslant 2} C_\ell^{(i,j)}(x - X_\ell)^i(y - Y_\ell)^j, \tag{6}$$

where $C^{(i,j)}$ are coefficients of the biquadratic polynomial (6) ensuring conservation (e.g., [23,24]), and (X_ℓ, Y_ℓ) is the centroid of cell ℓ. Henceforth $i,j \in \{0,1,2\}$. Collecting terms of the same order x^iy^j in (6) yields

$$f_\ell(x,y) = \sum_{i+j\leqslant 2} c_\ell^{(i,j)}x^iy^j, \tag{7}$$

where $c^{(i,j)}$ are derived coefficients. Then the integral of the polynomial reconstruction function $f_\ell(x,y)$ in (7) can be written as

$$\int\int_{a_{k\ell}} f_\ell(x,y)\,dx\,dy = \sum_{i+j\leqslant 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}, \tag{8}$$

where $w_{k\ell}^{(i,j)}$ is given by

$$w_{k\ell}^{(0,0)} = \frac{1}{2}\sum_{h=1}^{N_h}(x_{k\ell,h} + x_{k\ell,h-1})(y_{k\ell,h} - y_{k\ell,h-1}) \tag{9}$$

$$w_{k\ell}^{(1,0)} = \frac{1}{6}\sum_{h=1}^{N_h}\left(x_{k\ell,h}^2 + x_{k\ell,h}x_{k\ell,h-1} + x_{k\ell,h-1}^2\right)(y_{k\ell,h} - y_{k\ell,h-1}) \tag{10}$$

$$w_{k\ell}^{(0,1)} = -\frac{1}{6}\sum_{h=1}^{N_h}\left(y_{k\ell,h}^2 + y_{k\ell,h}y_{k\ell,h-1} + y_{k\ell,h-1}^2\right)(x_{k\ell,h} - x_{k\ell,h-1}) \tag{11}$$

$$w_{k\ell}^{(2,0)} = \frac{1}{12}\sum_{h=1}^{N_h}(x_{k\ell,h} + x_{k\ell,h-1})\left(x_{k\ell,h}^2 + x_{k\ell,h-1}^2\right)(y_{k\ell,h} - y_{k\ell,h-1}) \tag{12}$$

$$w_{k\ell}^{(0,2)} = -\frac{1}{12}\sum_{h=1}^{N_h}(y_{k\ell,h} + y_{k\ell,h-1})\left(y_{k\ell,h}^2 + y_{k\ell,h-1}^2\right)(x_{k\ell,h} - x_{k\ell,h-1}) \tag{13}$$

$$w_{k\ell}^{(1,1)} = \frac{1}{24}\sum_{h=1}^{N_h}\left\{\left[y_{k\ell,h}\left(3x_{k\ell,h}^2 + 2x_{k\ell,h}x_{k\ell,h-1} + x_{k\ell,h-1}^2\right) + y_{k\ell,h-1}\left(x_{k\ell,h}^2 + 2x_{k\ell,h}x_{k\ell,h-1} + 3x_{k\ell,h-1}^2\right)\right] \times \left(y_{k\ell,h} - y_{k\ell,h-1}\right)\right\}, \tag{14}$$

where $(x_{k\ell,h-1}, y_{k\ell,h-1})$ and $(x_{k\ell,h}, y_{k\ell,h})$ are contiguous points (defining a line segment) and the index h is cyclic so that $h = 0$ equals $h = N_h$. Note that after having computed the weights the detailed line-segments information $(x_{k\ell,h}, y_{k\ell,h})$ is no longer needed.

The weights $w_{k\ell}^{(i,j)}$ given in Eqs. (9)–(14) have been derived by using (5) with the following pairs $(P^{(i,j)}, Q^{(i,j)})$:

$$\left(P^{(0,0)} = 0, Q^{(0,0)} = x\right),$$
$$\left(P^{(1,0)} = 0, Q^{(1,0)} = \frac{x^2}{2}\right),$$
$$\left(P^{(0,1)} = -\frac{y^2}{2}, Q^{(0,1)} = 0\right),$$
$$\left(P^{(2,0)} = 0, Q^{(2,0)} = \frac{x^3}{3}\right),$$
$$\left(P^{(0,2)} = -\frac{y^3}{3}, Q^{(0,2)} = 0\right),$$
$$\left(P^{(1,1)} = 0, Q^{(1,1)} = \frac{x^2y}{2}\right).$$

Note that the choice of P and Q is not unique – here we haven chosen P and Q as in [25]. Clearly the method easily generalizes to high-order given the reconstruction coefficients $c_\ell^{(i,j)}$.

2.2. Final discretized transport equation

Using (3) and (4) the discretization scheme corresponding to conservative semi-Lagrangian transport can now be written as

$$\bar{\psi}_k^{n+1}\Delta A_k = \sum_{\ell=1}^{L_k} \int\int_{a_{k\ell}} f_\ell(x,y)\,dx\,dy = \sum_{\ell=1}^{L_k}\left[\sum_{i+j\leqslant 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}\right]. \tag{15}$$

The reconstruction coefficients $c_\ell^{(i,j)}$ are derived from known cell average values $\bar{\psi}_\ell^n$. Here we use the piecewise-parabolic reconstruction method in each coordinate direction as in [21] to obtain $c_\ell^{(0,0)}$, $c_\ell^{(1,0)}$, $c_\ell^{(0,1)}$, $c_\ell^{(2,0)}$, and $c_\ell^{(0,2)}$ and the cross term $c_\ell^{(1,1)}$ is computed as in [26].

It is worth noting the separation of the weights $w_{k\ell}^{(i,j)}$ from the reconstruction coefficients $c_\ell^{(i,j)}$ in (15). In practice this separation implies that once the weights have been computed they can be reused for the integral of each additional tracer distribution at a given time-step. Hence the transport of additional tracers reduces to the multiplication of precomputed weights and reconstruction coefficients similarly to the incremental remapping algorithm and traditional non-conservative semi-Lagrangian schemes.

3. Extension to the sphere

3.1. Gnomonic cubed-sphere grid

For the present study we consider cubed-sphere grids resulting from equi-angular gnomonic (central) projection

$$x = r\tan\alpha \quad\text{and}\quad y = r\tan\beta; \quad \alpha,\beta \in \left[-\frac{\pi}{4},\frac{\pi}{4}\right], \tag{16}$$

where α and β are central angles in each coordinate direction, $r = R/\sqrt{3}$ and R is the radius of the Earth [15]. Without loss of generality we assume $r = 1$. For a schematic illustration of the gnomonic projection/coordinates see Fig. 7. A point on the sphere is identified with the three-element vector (x, y, v) where v is the panel index (Fig. 8). Hence the physical domain S (sphere) is represented by the gnomonic (central) projection of the cubed-sphere faces, $\Omega^{(v)} = [-1,1]^2$, $v = 1,2,\ldots,6$, and

$$S = \bigcup_{v=1}^{6} \Omega^{(v)},$$

where the panel domains $\Omega^{(v)}$ are non-overlapping and the cube edges are discontinuous. Note that any straight line on the gnomonic projection (x, y, v) corresponds to a great-circle arc on the sphere. In the discretized scheme we let the number of cells along a coordinate axis be N_c so that the total number of cells in the global domain is $6 \times N_c^2$. Note that the equi-angular cubed-sphere grid is orthogonal only at the center of each panel (6 points) elsewhere it is a non-orthogonal curvilinear coordinate.

3.2. Patch boundaries

One advantage of the cubed-sphere geometry is that the interior of panels can be treated as in Cartesian geometry. However, it is required to consistently couple the panel discretizations for the global domain.

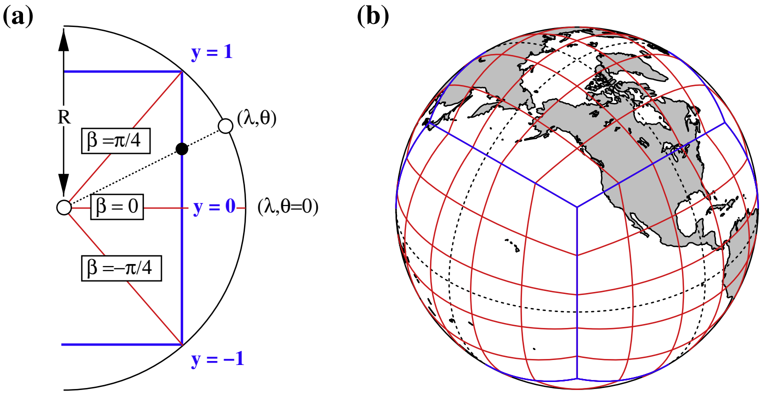


Fig. 7. (a) A schematic illustration of the gnomonic coordinate. For simplicity only the $y = r \tan \beta$ and β-coordinates of the gnomonic projection are shown for one of the equatorial panels of the cubed-sphere ($v = 1, 2, 3, 4$). A point on the sphere (λ, θ) has the gnomonic y coordinate given by the intersection (filled circle) between face v of the inscribed cube (thick lines) and the straight line (dashed line) connecting the point on the sphere and the center of the sphere (unfilled circles). The central angle β for (λ, θ) is given by the angle between the normal vector for the face and the dashed line. Solid straight lines show coordinates for the center ($\beta = y = 0$) and edges of the panel ($y = \pm 1, \beta = \pm \pi/4$). (b) shows the control-volumes on the surface of the sphere for the equi-angular cubed-sphere grid with $N_c = 5$. Equi-angular refers to the fact that the increment in α (and β) for adjacent coordinate lines is constant ($\Delta\alpha = \Delta\beta$).

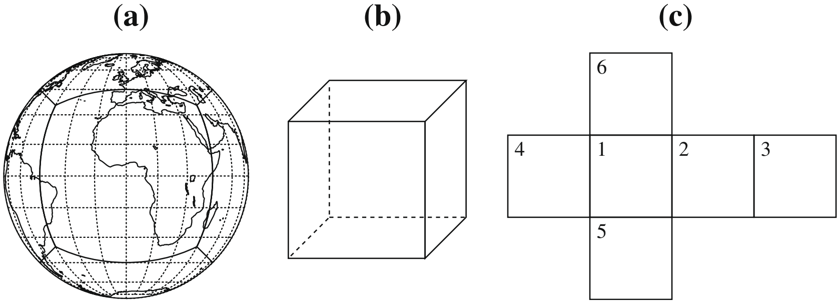


Fig. 8. (a) The edges of the cubed-sphere grid plotted on the sphere (solid lines). Dashed lines show latitudes and longitudes. (b) The inscribed cube and (c) the numbering convention for the panels of the cube used in this paper. The Greenwich meridian line ($\lambda = 0$) divides panel $v = 1$ in two.

In this Section we first discuss how the panel boundaries are treated in CSLAM. The mechanism for mass flux exchange between panels is then presented and finally we derive the spherical line integral formulae.

3.2.1. Departure cells

All computations are performed on the gnomonic projection in (x, y, v)-coordinates so that the algorithm for Cartesian geometry described earlier can be employed. As in the Cartesian case we connect the departure points with straight line segments. As mentioned previously, by doing so in the gnomonic projection the sides of the departure cells are great-circle arcs on the sphere. For cells that stay completely on a panel when being transported by the flow (for one time-step) the overlap areas $a_{k\ell}$ are defined exactly as in the Cartesian case. The question then becomes how to deal with the cells that traverse the edges of the cube. Since the CSLAM scheme is fully two-dimensional it is possible to treat cells that cross panel edges in a rigorous two-dimensional manner that adds a minor complexity to the algorithm as compared to the Cartesian case.

For a particular panel v we introduce a halo zone around the panel and treat the halo cells on the same projection as panel v (Fig. 9). An algorithm for identifying indices of neighboring cells across panel sides is, for example, given in [27]. As an illustrative example consider a 1-cell halo zone and a resolution of $N_c = 9$. Fig. 9(a) shows the Eulerian cells on the gnomonic projection for panel v (solid lines) as well as the halo cells (dashed lines). Since the sides of any grid cell on the cubed-sphere are great-circle arcs also the halo cell sides are straight lines on panel v's gnomonic projection. The halo cell sides are, however, not necessarily aligned with panel v grid lines.

We compute the departure points for the grid cell vertices on panel v as well as for the grid cell vertices of the halo zone cells. The departure points connected by straight lines are shown on Fig. 9(b).

Next we restrict the overlap areas $a_{k\ell}$ to panel v:

$$a^{(v)}_{k\ell} = a_{k\ell} \cap \Omega^{(v)}, \tag{17}$$

so that the panel v restricted departure area is given by

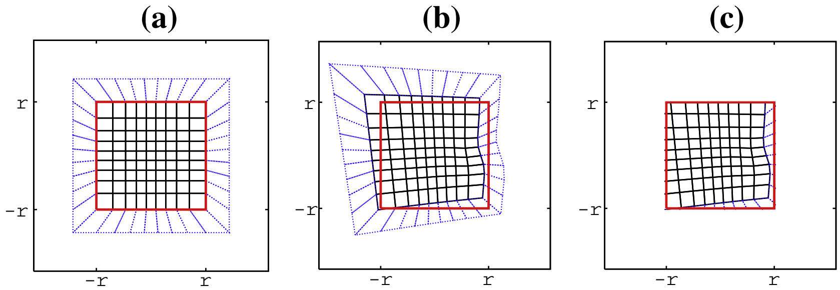


Fig. 9. (a) The grid lines for a panel v as projected onto a plane (solid black lines) and the halo zone grid lines from adjacent faces plotted on panel v's projection (dashed lines). The boundary of panel v is marked with the thick solid (red) line. (b) The departure grid corresponding to the arrival grid shown on (a) for the moving vortices test case using a time-step of 5 h. (c) The departure grid 'clipped' so it is limited to panel v. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

$$a_k^{(v)} = \left(\bigcup_{\ell=1}^{L_k} a_{k\ell}\right) \cap \Omega^{(v)}. \tag{18}$$

The superscript (v) on the left-hand side of (17) and (18) refers to the fact that $a_{k\ell}$ and a_k have been restricted to panel v. The 'clipping' procedure is graphically shown on Fig. 9(c). Note that by 'clipping' the parts of departure areas that are not located on panel v the departure cell is no longer guaranteed to be a quadrilateral but can be a simply connected polygon. This does, however, not add any particular complexity to the algorithm as compared to the Cartesian case. It is noted that $a_k^{(v)}$ with an appropriate width of the halo zone spans panel v without overlaps.

3.2.2. Panel mass-exchanges

The procedure of mass-exchanges between panels is described by example. Consider the situation when a departure cell is located over the edge of the cube, for example, as shown in Fig. 10, where the departure cell corresponding to the arrival cell in the upper-right corner of panel 1 span panels 1, 2 and 6 marked with patterns 'hatched', 'hexagon' and 'zig-zagged', respectively, on Fig. 10(a). The mass in $a_k^{(1)} = a_k \cap \Omega^{(1)}$ is computed by integration on panel 1 ('hatched pattern' on Fig. 10(a)). The masses in the parts of the departure area that overlap panels 2 and 6 will correspond to masses over areas that 'entered' from halo cells of these panels (Fig. 10(b) and (d)) and are computed on those panels. When updating the amount of mass ending up in the arrival cell in question the masses computed on neighboring panels 2 and 6 must be added to the mass over the 'hatched pattern' cell on panel 1. This is done similarly to the index association used to identify neighbors to Eulerian cells on the panel sides. Note this procedure for handling the sides of the panels allows for large CFL numbers as long as the halo zone is chosen wide enough.

3.3. Line-integrals on the cubed-sphere

Let $\mathbf{\Psi}$ be a vector field with contravariant components Ψ_x and Ψ_y in the direction of the unit basis vectors $(\mathbf{e}_x, \mathbf{e}_y)$, i.e., $\mathbf{\Psi} = \Psi_x\mathbf{e}_x + \Psi_y\mathbf{e}_y$. Following [10] Gauss–Green's theorem for the vector field $\mathbf{\Psi}$ in gnomonic coordinates can be written as

$$\int_{a_{k\ell}} \nabla \cdot \mathbf{\Psi} dV = -\oint_{\partial a_{k\ell}} [\widetilde{\Psi}_x dy + \widetilde{\Psi}_y dx], \tag{19}$$

where

$$\widetilde{\Psi}_x = \frac{\Psi_x}{\rho\sqrt{1+y^2}} \quad \text{and} \quad \widetilde{\Psi}_y = \frac{\Psi_y}{\rho\sqrt{1+x^2}},$$

with $\rho = \sqrt{1+x^2+y^2}$. Here, the divergence operator is given by

$$\nabla \cdot \mathbf{\Psi} = \rho^3 \left[\frac{\partial \widetilde{\Psi}_x}{\partial x} + \frac{\partial \widetilde{\Psi}_y}{\partial y}\right]. \tag{20}$$

As usual, the contour integral is taken in the counter-clockwise direction around the boundary of a given overlap area $a_{k\ell}$.

Again we consider sub-grid-cell reconstructions of up to third-order of the form (6) but now X_ℓ and Y_ℓ refer to the x and y components of the Eulerian cell centroids defined by

$$X_\ell = \frac{1}{A_\ell}\int_{A_\ell} x\,dA, \quad Y_\ell = \frac{1}{A_\ell}\int_{A_\ell} y\,dA.$$

In practice, these quantities are computed by transforming the area integrals to line-integrals via Gauss–Green's theorem.

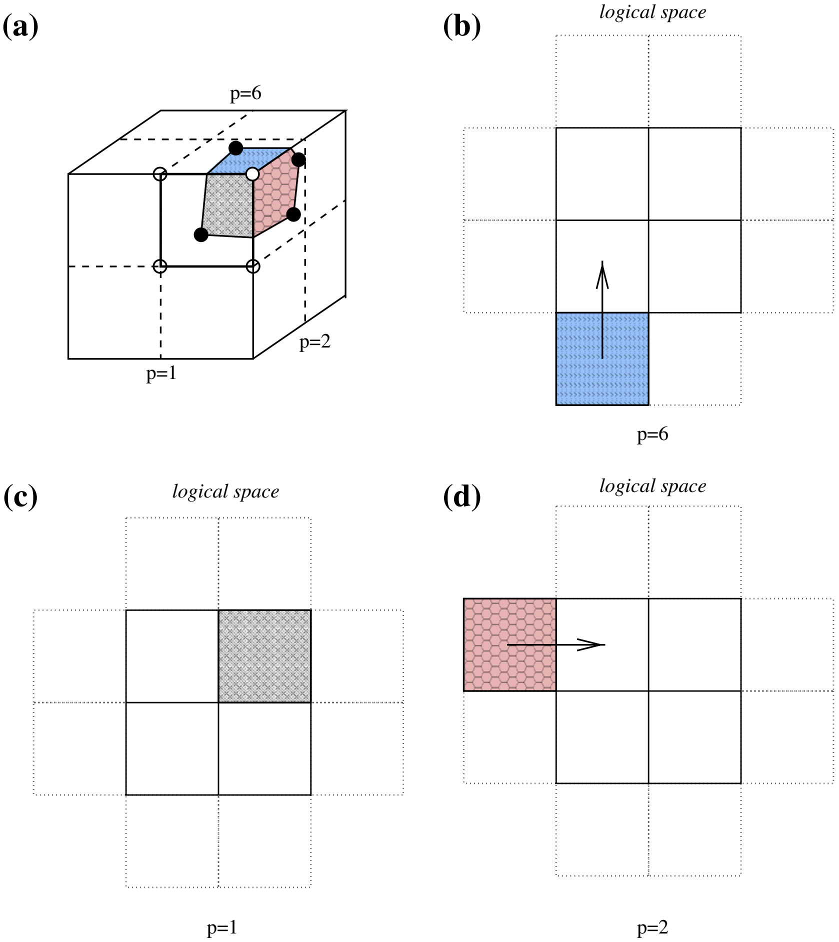


Fig. 10. A schematical illustration of how mass is exchanged between panels. (a) shows the gnomonic projection on the inscribed cube of the Eulerian cells (dashed lines) and a departure cell on the edge of the inscribed cube (deformed cell with vertices marked with filled circles) corresponding to an arrival cell located in the upper-right corner of panel 1 (unfilled circles connected with thick lines). (b–d) Illustrate how the mass-exchange between the panels is handled in logical space for panels 6, 1 and 2, respectively. See text for more details.

In order to apply the Gauss–Green theorem to compute the integral of $f_\ell(x,y)$ over $a_{k\ell}$ we need to determine a $\mathbf{\Psi}^{(i,j)}$ so that

$$\nabla \cdot \mathbf{\Psi}^{(i,j)} = x^i y^j. \tag{21}$$

By choosing $\widetilde{\Psi}_x^{(i,j)} = 0$ this reduces to solving the equation

$$\frac{\partial}{\partial y}\left(\widetilde{\Psi}_y^{(i,j)}\right) = \frac{x^i y^j}{\rho^3}, \tag{22}$$

Note that even with the aforementioned simplification $\widetilde{\Psi}_y$ is still not unique as there is a family of potentials $\mathbf{\Psi}$ that satisfy (22), that is, we may freely choose an additive constant that will play no role in the final calculation. Also, one may solve (22) in terms of either central angle or gnomonic coordinates, which are connected via the relation (16). In either case one will obtain identical expressions for the potentials:

$$\widetilde{\Psi}_y^{(0,0)}(x,y) = \frac{1}{1+x^2}\,\frac{y}{\rho}, \tag{23}$$

$$\widetilde{\Psi}_y^{(1,0)}(x,y) = \frac{1}{1+x^2}\,\frac{xy}{\rho}, \tag{24}$$

$$\widetilde{\Psi}_y^{(0,1)}(x,y) = -\frac{1}{\rho}, \tag{25}$$

$$\widetilde{\Psi}_y^{(2,0)}(x,y) = \frac{1}{1+x^2}\,\frac{x^2 y}{\rho}, \tag{26}$$

$$\widetilde{\Psi}_y^{(0,2)}(x,y) = -\frac{y}{\rho} + \operatorname{arcsinh}\left(\frac{y}{\sqrt{1+x^2}}\right), \tag{27}$$

$$\widetilde{\Psi}_y^{(1,1)}(x,y) = -\frac{x}{\rho}, \tag{28}$$

(see [10], for the corresponding formulas given in terms of central angle coordinates). Substituting the expressions (23)–(28) into (19) and taking the line-integrals yields the area-integrals of the terms $x^i y^j$. For lines that are parallel to the coordinate axis the line-integrals can be computed exactly (see [10]), that is, for lines of constant y:

$$I^{(0,0)}(x,y) = -\arctan\left(\frac{xy}{\rho}\right), \tag{29}$$

$$I^{(1,0)}(x,y) = \operatorname{arcsinh}\left(\frac{y}{\sqrt{1+x^2}}\right), \tag{30}$$

$$I^{(0,1)}(x,y) = \operatorname{arcsinh}\left(\frac{x}{\sqrt{1+y^2}}\right), \tag{31}$$

$$I^{(2,0)}(x,y) = -y\operatorname{arcsinh}\left(\frac{x}{\sqrt{1+y^2}}\right) - \arccos\left(\frac{x}{\sqrt{1+x^2}}\frac{y}{\sqrt{1+y^2}}\right), \tag{32}$$

$$I^{(0,2)}(x,y) = -x\operatorname{arcsinh}\left(\frac{y}{\sqrt{1+x^2}}\right) - \arccos\left(\frac{x}{\sqrt{1+x^2}}\frac{y}{\sqrt{1+y^2}}\right), \tag{33}$$

$$I^{(1,1)}(x,y) = \rho, \tag{34}$$

where

$$I^{(i,j)}(x,y) = -\int \widetilde{\Psi}_y \, dx.$$

Since we have chosen $\Psi_x = 0$ lines of constant x give zero contribution which, in general, results in having to compute half the number of inner line-integrals compared to a non-zero choice of Ψ_x.

For arbitrarily oriented lines closed form line-integration is not straight forward and it is most likely computationally cheaper to evaluate the line-integrals with Gaussian quadrature. Herein we compute lines parallel to the coordinate lines using exact integration (as with all inner integrals) and all other line-integrals are approximated using Gaussian quadrature. As (23)–(28) are rather smooth and slowly varying functions over a cell side, relatively low order quadrature is sufficient (see results in Section 4).

Note that summing all outer line-integrals (except when departure cell sides coincide with grid lines) yields zero since a line-integral along a particular side of a cell is exactly equal to the line integral along the same side shared with the adjacent cell but with opposite sign. All inner line-integrals do, however, not cancel since the sub-grid-cell distribution is discontinuous across grid cell sides. The inner line-integrals collectively yield the total mass in all grid cells (to machine precision).

3.4. Sub-grid-cell reconstructions on the sphere

The coefficients for the mass-conservative reconstruction polynomial $c_\ell^{(i,j)}$ are described in detail in [10] so only a brief overview will be given here. Basically the coefficients are defined in terms of a Taylor expansion

$$c_\ell^{(i,j)} = \left(\frac{\partial^{i+j} f}{\partial x^i \partial y^j}\right)_\ell, \quad (i,j) \neq (0,0), \tag{35}$$

and by choosing the constant term so that mass is conserved

$$c^{(0,0)} = \bar{\psi}_\ell + \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_\ell \left(X_\ell^2 - m_\ell^{(2,0)}\right) + \frac{1}{2}\left(\frac{\partial^2 f}{\partial y^2}\right)_\ell \left(Y_\ell^2 - m_\ell^{(0,2)}\right) + \left(\frac{\partial^2 f}{\partial x \partial y}\right)_\ell \left(X_\ell Y_\ell - m_\ell^{(1,1)}\right), \tag{36}$$

where $m_\ell^{(i,j)}$ are the area-averaged moments defined by

$$m_\ell^{(i,j)} = \frac{1}{\Delta A_\ell}\int_{A_\ell} x^i y^j \, dA. \tag{37}$$

Obviously the derivatives in (35) must be estimated and several options were explored in [10]. We use a non-equidistant parabolic interpolation procedure in gnomonic (x,y)-coordinates to estimate the gradients (see [10] for details).

In order to compute coefficients near the panel boundaries information from neighboring panels is needed, however, the average values on neighboring panels are not aligned with the cells on the panel in question (see Fig. 9). To apply the discretization formulas the values in the halo cells that would be obtained by extending the panel in question outwards are needed. Here we use a one-dimensional non-conservative fourth-order non-conservative interpolation to obtain halo cell values from the neighboring cell average values [10]. Note that this interpolation procedure does not need to be conservative since any choice of coefficients $c_\ell^{(i,j)}$ will yield mass-conservation through (36).

3.5. Discretized transport equation on the cubed-sphere

The CSLAM transport scheme on the cubed-sphere in analogous to the Cartesian version (15):

$$\bar{\psi}_k^{n+1}\Delta A_k = \sum_{\ell=1}^{L_k}\left[\sum_{i+j\leqslant 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}\right], \tag{38}$$

where the area of the regular (Eulerian) grid cells ΔA_k can be computed using the formulas in Appendix C of [28] (note that inverse cosine is missing on the right-hand side of equation C3 in [28]

4.1. Experiments in Cartesian geometry

Here we consider two standard advection tests which are solid-body rotation and deformational flow. The results for the experiments in Cartesian geometry are compared to published conservative semi-Lagrangian schemes: SLICE [29], CCS [30] and CISL [21]. The SLICE and CCS schemes are cascade schemes in which the two-dimensional transport problem is cast into two one-dimensional sweeps; one in a coordinate direction and the other along the deformed Lagrangian coordinate that is initially orthogonal with the first sweep. The CISL scheme uses a fully two-dimensional cell approximation but does not connect the departure vertex points with straight lines but the cell is approximated with line-segments that are parallel to the coordinate axis (see Fig. 2 in [22]). For this flow the CSLAM scheme uses an exact approximation to the departure cells. For each sweep in the cascade schemes and in each coordinate direction of the CISL and CSLAM schemes, the piecewise parabolic method is used [24]. There are different versions of the SLICE scheme depending on which sub-grid-scale reconstruction method is used for the cascade sweeps: The piecewise cubic method, SLICE(PCM), using cubic polynomials, and SLICE(PPM) and SLICE(PSM) using the piecewise parabolic method and piecewise spline method [31], respectively. Contrary to CISL the CSLAM scheme also includes a cross term which is approximated as in [26].

4.1.1. Solid-body advection of a slotted cylinder and cosine hill

For the solid-body rotation tests the Zelasak's slotted cylinder [32] and cosine hill (see, e.g., [29]) are used. The flow rotates about the center of the domain with an angular velocity so that one revolution is completed in 96 and 71 time-steps for the slotted cylinder and cosine hill test cases, respectively. A domain of 100×100 grid cells is used with grid-spacing $\Delta x = \Delta y = 1$ $(\Delta A_k = 1)$ for the slotted cylinder test case. For the cosine hill a much coarser resolution of 32×32 grid cells is used. The specific parameters for the analytic solution (including the initial distribution) are given in [29].

Fig. 11 shows surface plots of the CSLAM solution after one revolution for the slotted cylinder and cosine hill test cases, respectively. Standard error measures are given in Tables 1 and 2 as well as the performance measures for other published mass-conservative semi-Lagrangian schemes. All schemes use analytical trajectories.

First of all it is noted that the relative performance of the schemes is similar when comparing error measures based on the same $\bar{\psi}^e$ (either point values or cell-averages) and that the error measures decrease when using the cell average for $\bar{\psi}^e$ compared to using the point value. So the conclusions are independent of the choice of $\bar{\psi}^e$ in the error norms as long as the error norms are computed consistently. CSLAM performs better than CISL in all test cases and error measures. Compared to the cascade schemes CSLAM performs better or worse depending on the test case and type of reconstruction function used. Note that the cascade schemes may in certain cases have a more accurate representation of the diagonal variation since the second remapping is along the Lagrangian coordinate lines and not along coordinate lines [22]. This might give a better representation of diagonal variation than the cross-term in the fully two-dimensional reconstruction used in CSLAM.

To assess the importance of the cross-term in CSLAM the test case was also run without it $(\text{CSLAM}^{\star})$. As a results the error measures worsen by a few percent for the slotted cylinder test case and by approximately 15–19% depending on the error measure for the cosine hill test case.

4.1.2. Idealized cyclogenesis

The idealized cyclogenesis problem introduced by [34] is used as a standard scalar advection test [23]. The flow is highly deformational challenging other aspects of the scheme than the ability to transport distributions as solid bodies. The test case consists of a circular vortex that forces the initial condition to curl up into thin filaments with steep gradients (Fig. 12). A complete test case description is, for example, given in [29]. Standard settings are used: domain size is 128×128 cells, $\Delta t = 0.3125$, $\Delta x = \Delta y = 0.078125$ and the test is run for 16 time-steps (corresponding to 5 time units).

Standard error norms are given in Table 3. The fully two-dimensional CSLAM scheme is slightly superior followed by CISL and SLICE in terms of standard error measures. Excluding the cross term in CSLAM only has a minor effect on the accuracy.

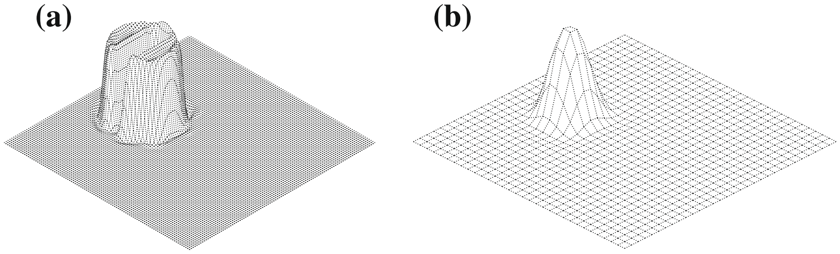
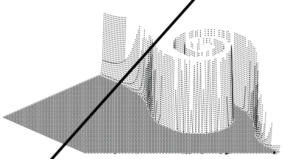


Fig. 11. Surface plots of the CSLAM solution to the solid-body advection of (a) a slotted cylinder and (b) a cosine hill, respectively, after one revolution.



4.2. Test cases on the sphere

We show results from three test cases in spherical geometry commonly used in the meteorological literature. Unfortunately, various authors choose different parameters for the same test cases. Therefore we run each case with different parameters to facilitate the comparison with published schemes. The test cases are defined below.

Table 3
Same as Table 1 but for the cyclogenesis test case.

	$\bar{\psi}^e$ = point value at center				$\bar{\psi}^e$ = approximate cell average			
Scheme	RMS	l_1	l_2	l_∞	RMS	l_1	l_2	l_∞
Idealized cyclogenesis in Cartesian geometry								
CSLAM	**0.0642**	**0.0113**	**0.0646**	**0.8802**	**0.0272**	**0.0059**	**0.0275**	**0.3262**
CSLAM*	**0.0653**	**0.0116**	**0.0656**	**0.8777**	**0.0285**	**0.0062**	**0.0287**	**0.3363**
SLICE(PPM)	0.0701	–	–	–	0.0317	–	–	–
SLICE(PCM)	0.0693	–	–	–	0.0311	–	–	–
CISL	0.0666	0.0119	0.0670	0.8737	0.0304	0.0065	0.0307	0.3598

4.2.1. Solid-body advection of a cosine hill

The wind field that will transport a distribution along a great-circle without distorting it is given by [35]

$$u = u_0(\cos\varphi\cos\theta + \sin\varphi\cos\lambda\sin\theta), \tag{44}$$

$$v = -u_0\sin\varphi\sin\lambda, \tag{45}$$

where u and v are the velocity components in the longitudinal (λ) and latitudinal (θ) directions, respectively, $u_0 = 2\pi R/(12\text{ days})$ and φ is the rotation angle (flow orientation parameter), and R is the radius of the sphere. When $\varphi = 0$ the flow is oriented along the equator, and when $\varphi = \pi/2$ the flow is along the pole-to-pole direction resulting in cross-polar advection. The flow field is non-divergent and translates the solid-body along a great-circle without incurring any deformation so that the final solution after 12 model days should exactly match the initial condition. The initial condition for the solid-body (or scalar field ψ) is a 'cosine hill' defined as follows [35]:

$$\psi(\lambda,\theta) = \begin{cases} \frac{\psi_0}{2}\left[1+\cos\left(\frac{\pi R_g}{R_c}\right)\right], & \text{if } R_g \leqslant R_c \\ 0, & \text{if } R_g > R_c \end{cases}$$

where R_c is the radius of the hill and R_g is the great-circle distance between (λ, θ) and the center of the distribution (λ_c, θ_c),

$$R_g = \arccos[\sin\theta_c\sin\theta + \cos\theta_c\cos\theta\cos(\lambda-\lambda_c)],$$

and $(\lambda_c, \theta_c) = (3\pi/2, 0)$ so that initially the cosine hill is located at the center of an equatorial panel $(\nu = 4)$. When transported with $\varphi = \pi/4$, the cosine hill passes through the discontinuous regions containing two edges and four vertices of the cubed-sphere, and this is a more challenging parameter setting for advection on the cubed-sphere than, for example, $\varphi = 0$ [36]. Note that the cosine hill is only C^0 at the base of the hill.

Note that for the spherical semi-Lagrangian solid-body advection, the analytic trajectory origins (or departure point positions) (λ_d, θ_d) can be determined without the knowledge of wind fields (u, v) if the angular velocity ω_s of solid-body rotation is known. In order to compute exact trajectory origins, however, a flow dependent rotated spherical coordinate system (λ', θ') with respect to the regular (λ, θ)-sphere is required. The exact trajectory origins are then given by $(\lambda' - \omega_s\Delta t, \theta')$ on the rotated sphere which corresponds to the exact upstream position (λ_d, θ_d) on the regular sphere (see [37] for details).

Several different parameters for this test case are used in the literature. We will use the following:

1. The cosine hill with dimensions $R_c = R/3$ and $\psi_0 = 1000$. The time-step is $\Delta t = 1800$ s so that one full revolution is completed in 576 time-steps (12 days). This setting is, for example, used by [38]. The initial condition is shown on Fig. 15(d).
2. In the literature pertaining to semi-Lagrangian schemes (e.g., [21,39]) the cosine hill dimensions are typically $R_c = R7\pi/64$, and $\psi_0 = 1$ and the time-step is either $\Delta t = 4050$ s or $\Delta t = 14{,}400$ s so that in 12 days one full revolution is completed in 256 or 72 time-steps.

4.2.2. Deformational flow tests on the sphere

Recently Nair and Jablonowski [37] introduced a new deformational benchmark test for advection schemes on the sphere. The test consists of two deforming and moving vortices located at diametrically opposite sides of the sphere such that the flow is time dependent, non-divergent, and the analytic solution is known at any time. This test combines the solid-body rotation test [35] and the static deformational test [21]. It is referred to as the moving vortices test case and it is gaining popularity in the literature [38].

The exact solution at any t is given by [37]

$$\psi(\lambda',\theta',t) = 1 - \tanh\left[\frac{d}{\gamma}\sin(\lambda' - \omega_r(\theta')t)\right], \tag{46}$$

where (λ', θ') is the rotated coordinate system with respect to the regular (λ, θ) coordinates, $d = d_0\cos\theta'$ is the radial distance from the vortex center and ω_r is the angular velocity of the vortices. For a *smooth* deformational flow, the parameters $\gamma = 5$, $d_0 = 3$ are used [37]. The scaled tangential velocity V_t of the rotational motion is defined to be

$$V_t = u_0 \frac{3\sqrt{3}}{2} \operatorname{sech}^2(d) \tanh(d), \tag{47}$$

where the scale factor $u_0 = 2\pi R/(12 \text{ days})$, which indicates 12 model days are required for the full evolution of the vortices – the same time taken for a complete revolution around the sphere. The angular velocity $\omega_r(\theta')$ varies with the vortex *radial distance* Rd, and is defined by

$$\omega_r(\theta') = \begin{cases} V_t/(Rd) & \text{if} \quad d \neq 0 \\ 0 & \text{if} \quad d = 0. \end{cases} \tag{48}$$

which has the physical unit radians/seconds. The time dependent wind vector $(u,\ v)$ for the moving vortex is given by

$$u(t) = u_0(\cos\theta\cos\varphi + \sin\theta\cos\lambda\sin\varphi) + R\omega_r[\sin\theta_c(t)\cos\theta - \cos\theta_c(t)\cos(\lambda - \lambda_c(t))\sin\theta] \tag{49}$$

$$v(t) = -u_0\sin\lambda\sin\varphi + R\omega_r[\cos\theta_c(t)\sin(\lambda - \lambda_c(t))], \tag{50}$$

where φ is the flow orientation parameter as in the case of solid-body rotation test and $(\lambda_c(t),\ \theta_c(t))$ is the center of a moving vortex. The initial vortex center is located at $(\lambda_c(t=0),\ \theta_c(t=0)) = (\lambda_0,\ \theta_0)$ and the initial conditions for the vortex field is $\psi(\lambda', \theta', t = 0)$. Note that the static vortices [21] are a special case of the moving vortices and can be obtained by ignoring the solid-body rotation part (Eq. (49) and (50) with $u_0 = 0$). The detailed procedure for finding the exact departure point positions $(\lambda'_d,\ \theta'_d)$ for the moving vortices is described in [37], and will not be discussed herein. We consider both static and dynamic vortices for evaluating the CSLAM algorithm.

4.3. Results

4.3.1. Solid-body advection of a cosine hill

First we investigate the impact on accuracy, in terms of the standard error norms defined in (41)–(43), using different orders of Gaussian quadrature for the outer integrals as well as varying the order of the reconstruction functions. For that we use the solid-body advection of a cosine hill test case at resolution $N_c = 32$ and $\varphi = \pi/4, \Delta t = 4050$ s and a total of 256 time-steps (one revolution). If N_g is the number of Gaussian quadrature points the integration is exact for polynomials

Table 4
Standard error measures for the solid-body advection of a cosine hill for different orders of Gaussian quadrature (N_g is the number of Gaussian quadrature points) for the outer line-integrals as well as a version of CSLAM not including the cross term in the reconstruction polynomial (CSLAM*). The trailing N refers to the non-monotone (unlimited) version of CSLAM. The dimensions and parameters used here are: $N_c = 32$ (2.8125° resolution at equator), $\varphi = \pi/4, \Delta t = 4050$ s, 256 time-steps are used (one revolution), $\psi_0 = 1, R_c = R7\pi/64$.

Scheme	N_g	l_1	l_2	l_∞
Solid-body rotation of cosine hill on the sphere				
CSLAM*-N	2	0.0949	0.0536	0.0332
CSLAM-N	2	0.0764	0.0414	0.0254
CSLAM-N	3	0.0765	0.0414	0.0255
CSLAM-N	4	0.0765	0.0414	0.0255
CSLAM-N	5	0.0765	0.0414	0.0255

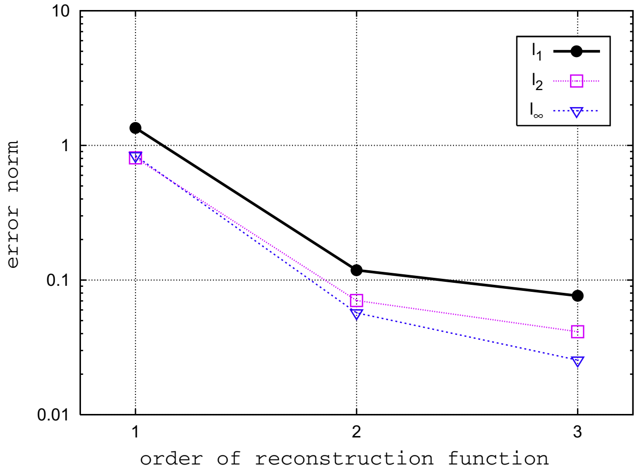


Fig. 13. Standard error norms for the solid-body advection of a cosine hill for different orders of reconstruction function. Settings are as in Table 4.

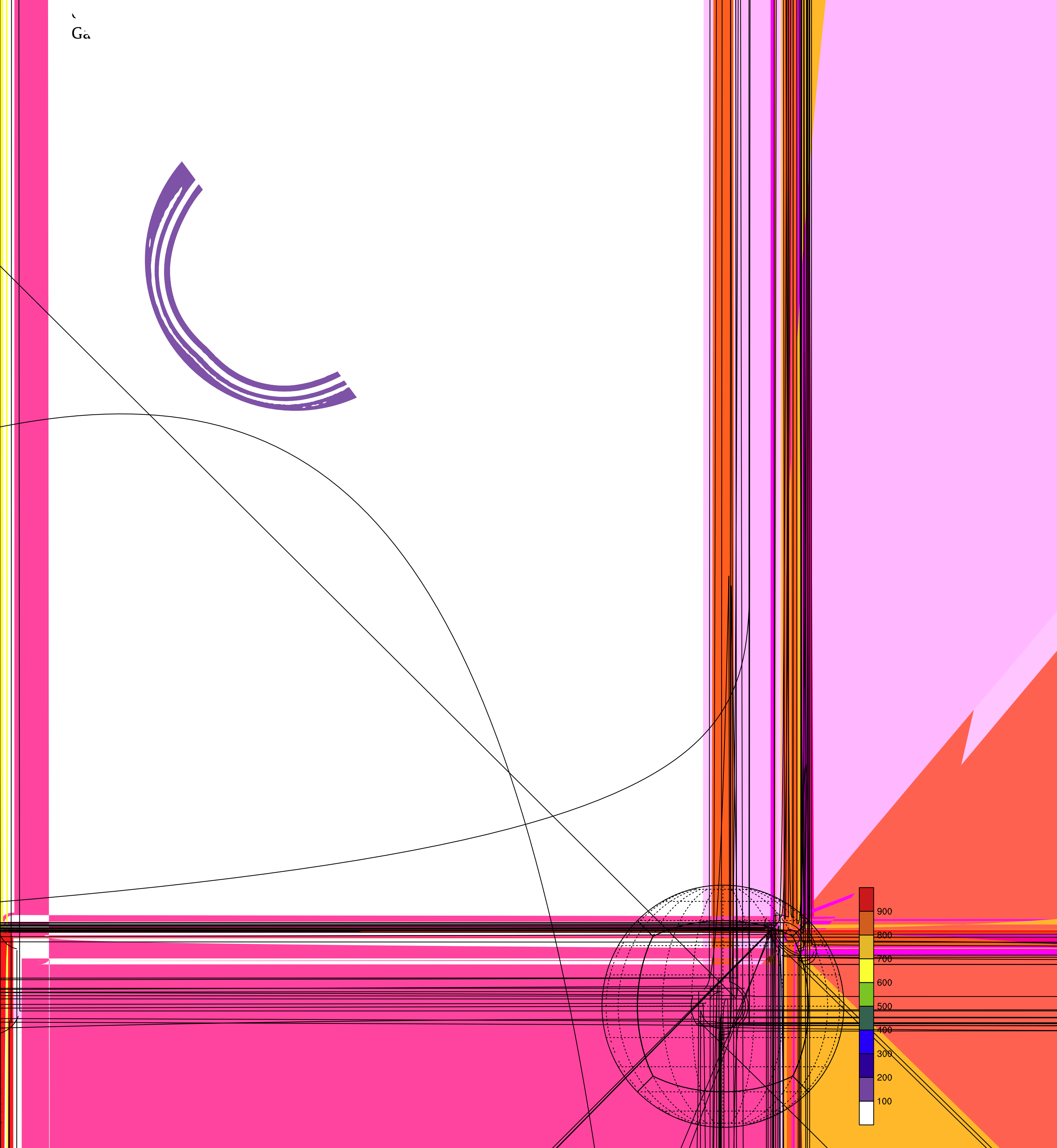


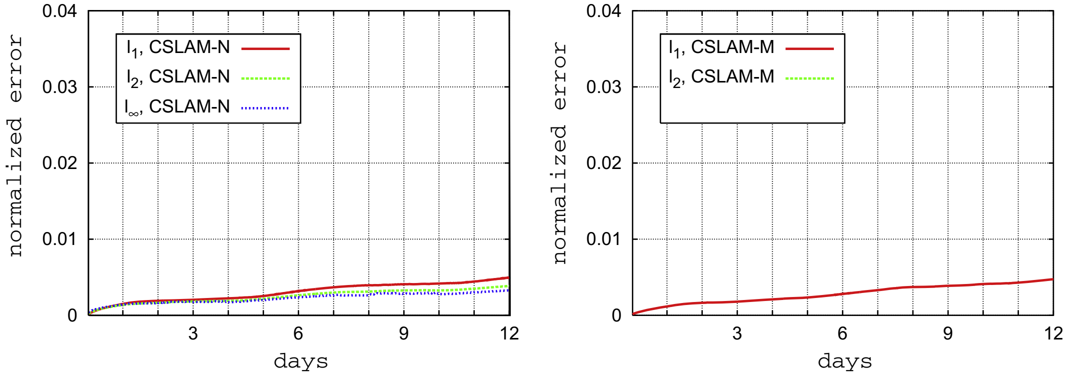
Fig. 15. (a–c) Normalized error norms l_1, l_2 and l_∞, respectively, as a function of resolution for the solid body advection of a cosine hill ($R_c = R/3, \psi_0 = 1000, \varphi = \pi/4$) after one full revolution completed using the non-monotone scheme with at time-step of 1800 s (triangles) and 4050 s (unfilled squares) as well as the monotone scheme with $\Delta t = 1800$ s. The solid line below the other error norm curves is a reference line with slope corresponding to the average convergence rate. (d) shows the initial condition (cosine hill in center of panel) and the exact solution after 44 h (cosine hile over inscribed cube edge).

use two-point Gaussian quadrature as default. Convergence with increasing order of sub-grid-scale reconstruction function is shown in Fig. 13. In the remainder of this paper we will use third-order reconstruction functions.

Table 5
Comparison of error measure norms from published conservative semi-Lagrangian schemes (CISL of [21], CCS of [30] and [41], and SLICE of [39,44,31]) for the solid-body rotation of a cosine hill with maximum amplitude $\psi_0 = 1$ for different values of rotation angle φ after one complete rotation ($\Delta t = 4050$ s, 256 time-steps). The unlimited schemes use appended acronym 'N' and monotone schemes use 'M' . All other than the CSLAM scheme use a regular latitude-longitude grid with a 2.8125° resolution. CSLAM uses $N_c = 32$ corresponding to a 2.8125° resolution at the equator for the cubed-sphere. Obviously the number of degrees of freedom is not identical for the latitude–longitude schemes compared to the cubed-sphere CSLAM scheme (the ratio between the two is 4/3).

Scheme	$\varphi = 0$			$\varphi = \pi/4$		
	l_1	l_2	l_∞	l_1	l_2	l_∞
Solid-body rotation of cosine hill on the sphere						
SLICE(PSM)-N	0.050	0.034	0.027	0.059	0.033	0.022
SLICE(PSM)-M	0.027	0.019	0.018	0.029	0.020	0.025
SLICE(PPM)-N	0.070	0.045	0.035	0.083	0.046	0.031
SLICE(PPM)-M	0.101	0.095	0.115	0.078	0.086	0.159
SLICE(PCM)-N	0.046	0.029	0.022	0.067	0.035	0.024
SLICE(PCM)-M	0.038	0.024	0.017	–	–	–
CISL-N	0.051	0.035	0.032	–	–	–
CISL-M	0.094	0.091	0.108	–	–	–
CSLAM-N	0.079	0.046	0.034	0.076	0.041	0.025
CSLAM-M	0.075	0.075	0.141	0.048	0.060	0.130
Scheme	$\varphi = \pi/2$			$\varphi = \pi/2 - 0.05$		
	l_1	l_2	l_∞	l_1	l_2	l_∞
SLICE(PSM)-N	0.079	0.054	0.049	0.077	0.052	0.043
SLICE(PSM)-M	0.057	0.046	0.043	0.056	0.045	0.044
SLICE(PPM)-N	0.103	0.065	0.055	0.111	0.067	0.050
SLICE(PPM)-M	0.109	0.102	0.118	0.109	0.102	0.124
SLICE(PCM)-N	0.079	0.049	0.042	0.079	0.048	0.039
SLICE(PCM)-M	0.058	0.040	0.037	0.056	0.039	0.041
CISL-N	0.063	0.046	0.048	–	–	–
CISL-M	0.084	0.084	0.109	–	–	–
CCS-N	0.054	0.042	0.065	–	–	–
CCS-M	0.076	0.082	0.129	–	–	–
CSLAM-N	0.079	0.046	0.034	0.079	0.046	0.034
CSLAM-M	0.075	0.075	0.141	0.070	0.069	0.133
Scheme	$\varphi = \pi/4 - 0.05$			$\varphi = \pi/4 + 0.05$		
	l_1	l_2	l_∞	l_1	l_2	l_∞
CSLAM-N	0.077	0.041	0.026	0.077	0.041	0.026
CSLAM-M	0.048	0.060	0.131	0.048	0.060	0.131

Parameters: $\psi_0 = 1,\ N_c = 32$ (2.8125° at equator), $R_c = R7\pi/64, \Delta t = 4050$ s.



Also illustrated in Table 4 is the importance of the cross-term in the sub-grid-scale reconstruction. In Cartesian geometry the cross-term only had a minor impact on the accuracy of the CSLAM scheme for the slotted cylinder (Table 1) and deformational flow test case (Table 3). For the cosine hill test case the impact was larger (approximately 15–19%). Not including the cross term (for this particular test setting) in the reconstruction function (CSLAM*) in the cubed-sphere version of CSLAM increases error measures $l_1,\ l_2$ and l_∞ by approximately 20%, 23% and 30%, respectively. This result is due to the fact that even for solid-body rotation the cells entering from neighboring panels are naturally skewed (Fig. 9) even though the Lagrangian cells on the sphere are simply translated (and not deformed or rotated).

For the solid-body rotation of a cosine hill negative undershoots appear without the application of monotone or positive definite filters as shown in Fig. 14(a). Following [10] we ensure monotonicity in the reconstruction function by employing the simple monotone filter of [40] that scales the sub-grid-scale reconstruction function so that its minimum and maximum values do not exceed the cell-averages of the neighboring cells. The unlimited versions for the CSLAM scheme use appended acronym 'N' and monotone versions use 'M'. When applying the monotone filter the negative undershoots disappear completely (Fig. 14b(b)) demonstrating that the monotone limiter is 'clipping' the spurious oscillations. The accuracy is, in general, slightly decreased in terms of standard error measures (Fig. 15(a)–(c)) with l_∞ being degraded the most.

A concern of every scheme implemented on a spherical grid is if the grids strong or weak singularities influence the solution adversely. For schemes defined on a regular latitude–longitude grid it is standard practice to let a distribution be advected across the strong singularities (poles) and look for possible spurious effects introduced by the converging meridians. For the cubed-sphere there are 8 weak singularities at the vertices of the inscribed cube. Therefore solid-body advection is performed with $\varphi = \pi/4$ and $\varphi = \pi/4 \pm 0.05$ so that a distribution initially located in the center of a panel passes over the vertices of the inscribed cube. The latter rotation angles are to avoid symmetry (Table 5). The time evolution of the standard error norms as the cosine hill passes over the vertices of the inscribed cube ($\varphi = \pi/4$) are shown on Fig. 16 for the unlimited and monotone scheme. No obvious noise is generated by the weak singularities of the cubed-sphere grid.

Since the scheme being described herein is in the category of finite-volume semi-Lagrangian algorithms its performance compared to published schemes in this category is discussed. Here we compare with CISL of [21], CCS of [30] and [41], and SLICE of [39] which are all implemented on a regular latitude–longitude grid. Table 5 shows that the schemes implemented on a regular latitude–longitude grid are almost twice as accurate in terms of $l_1,\ l_2$ and l_∞ when performing solid-body advection along the equator compared to advection across the poles. In other words, there is a large dependence on the rotation angle φ. For the cubed-sphere scheme there is little dependence on φ and the accuracy when using the same resolution at the equator is comparable to the CISL, SLICE and CCS schemes. It should be noted, however, that for a pure zonal flow the CISL, CCS and SLICE schemes defined on a regular latitude–longitude grid reduce to one dimension and there will be no mass-transport in the θ-direction. For a scheme defined on the cubed-sphere grid, however, the coordinate lines are not aligned with small circles and consequently this grid (as well as other non-traditional grids such as icosahedral grids) will not preserve a zonal flow exactly since there is a spurious transport in the θ-direction (see, e.g., [42]).

Also note that the number of degrees of freedom is less for the cubed-sphere scheme compared to the schemes defined on a regular latitude–longitude grid when both grids have the same resolution at the equator. More precisely, the ratio between the number of grid points on the regular latitude–longitude grid and the cubed-sphere grid both having the same resolution at the equator is 4/3.

The standard error measures as a function of resolution are shown on Fig. 15 for the solid-body rotation using the same settings as [38] ($\Delta t = 1800$ s) and [43] ($\Delta t = 4050$ s), respectively. The average convergence rates (e.g., [30]) for the unlim-

Table 6
Same as Table 5 but with a larger time-step $\Delta t = 14,400$ s (one revolution in 72 time-steps).

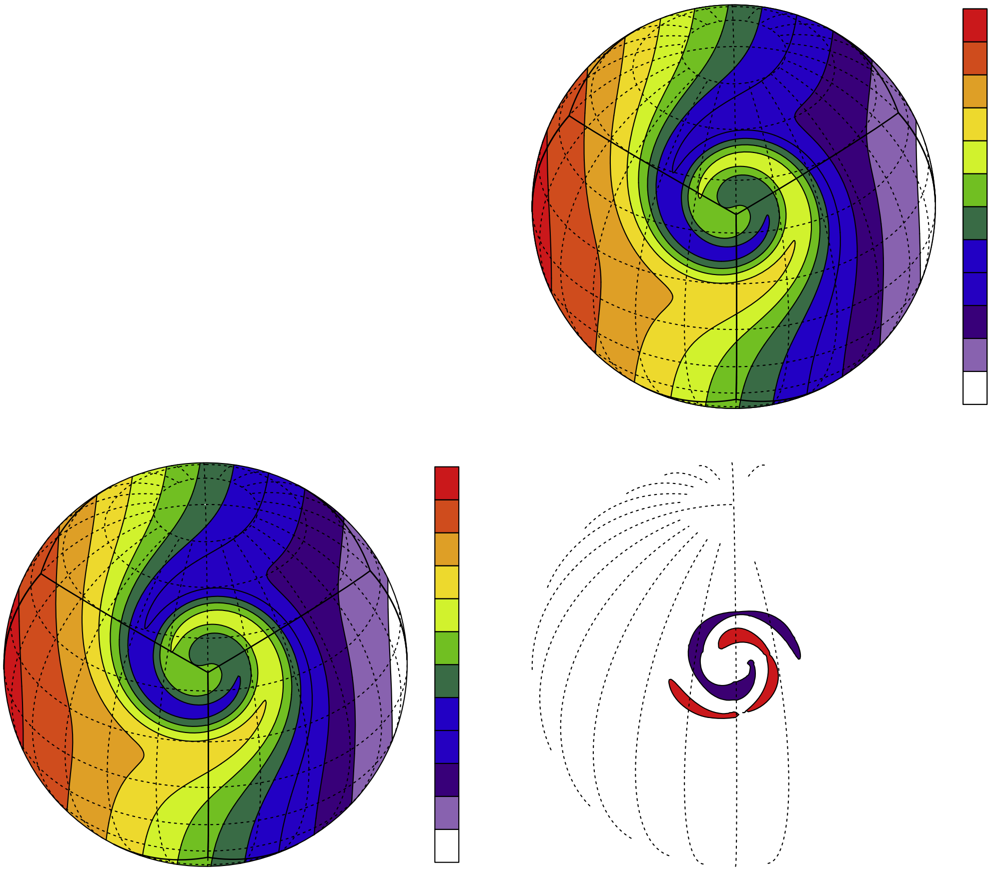
	$\varphi = \pi/2$		
Scheme	l_1	l_2	l_∞
CISL-N	0.037	0.031	0.033
CISL-M	0.040	0.042	0.068
CSLAM-N	0.031	0.018	0.012
CSLAM-M	0.029	0.033	0.070

Table 7
Error measures for the smooth deformational flow test case with the center of the vortex near the pole for CISL, CCS and CSLAM. Parameters used are: $t = 3$, 32 time-steps ($\Delta t = 4.5$ h), ($\lambda_0 = \pi + 0.025, \theta_0 = \pi/2.2$), $N_c = 32$.

Scheme	l_1	l_2	l_∞
CISL-N	0.0011	0.0025	0.0144
CISL-M	0.0013	0.0031	0.0211
CCS-M	0.0013	0.0033	0.0220
CSLAM-N	0.0013	0.0026	0.0137
CSLAM-M	0.0013	0.0027	0.0141

ited scheme are better than second-order (2.85, 2.48 and 2.25 for l_1, l_2 and l_∞ with $\Delta t = 1800$ s, respectively), but with the application of the monotone filter the l_∞ convergence rate drops slightly below second-order (1.87) while l_1 and l_2 are not affected and improved slightly (2.85, 2.55), respectively. Obviously the accuracy is better when using a longer time-step since less remappings are needed to complete one revolution.

To demonstrate the schemes ability to transport with large Courant numbers, the solid-body rotation test case is run with a much larger time-step than used above. Following [21] a time-step of 14,400 s is used so that one revolution is completed in 72 time-steps. To compare with literature a rotation angle of $\varphi = \pi/2$ is used. The error measures for CISL and CSLAM are comparable (Table 6).



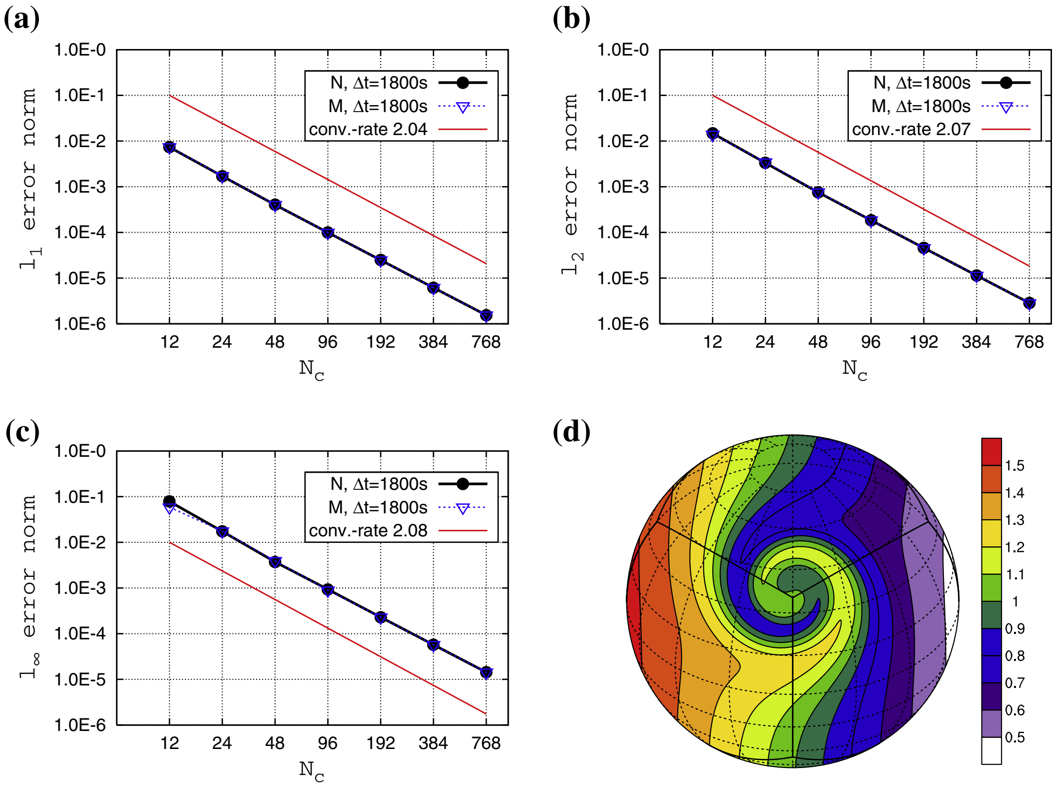


Fig. 18. Normalized error norms (a) l_1, (b) l_2 and (c) l_∞ as a function of resolution for the static vortices test case at day 6 (completed in 144 time-steps of 3600 s each) and (d) the exact solution at day 6. The vortices are situated near two edges of the cubed-sphere (offset by approximately 2°), that is, the origin of the rotated coordinate is located at $(\lambda_0, \theta_0) = (\pi - 0.8, \pi/4.8)$. These settings are identical to those used in [38]. The solid line with filled circles along it on plot (a–c) is the error norm for the unlimited scheme whereas the dashed line with unfilled triangles is for the monotone version of the scheme. The solid line below the other curves is a reference line with slope corresponding to the average convergence rate of the unlimited scheme which is (a) 2.04, (b) 2.07 and (c) 2.08, respectively.

4.3.2. Deformation flow test cases

The solid-body rotation test case only addresses the ability of a scheme to translate a distribution along a great-circle. The vortices test cases are formulated to address the schemes performance under strong deformational flow conditions. First the static vortices test case results are discussed. To compare with CISL and CCS the vortex is centered near the pole $(\lambda_0 = \pi + 0.025,\ \theta_0 = \pi/2.2)$ and a long time-step of 4.5 h is used at a 2.8125° resolution at the equator $(N_c = 32)$. Again the error measures for all schemes are comparable although the cubed-sphere scheme uses fewer cells in the polar areas (Table 7).

To challenge the cubed-sphere scheme the center of the vortex is placed near the corner of the inscribed cube $(\lambda_0,\ \theta_0) = (\pi - 0.8,\ \pi/4.8)$ (Fig. 17). The closest corner of the inscribed cube is located at $\lambda = 3\pi/4$ and $\theta = \mathrm{atan}(\sqrt{2}/2) \approx 0.615$ so the center of the vortex is offset by approximately 0.8° in longitude and 2.2° in latitude. Again, we look for any noise introduced by the weaker singularities of the cubed-sphere grid. To compare with literature, the solution and differences between the analytic and numeric solution at day 6 are shown on Fig. 17(c)–(f) at resolutions $N_c = 80$ (as used in [38]) and $N_c = 32$ (as used in [43]). No visible noise is generated by the scheme as the error is clearly related to gradient errors in the sub-grid-scale reconstruction function rather than the patch boundaries. The convergence rates as computed in [38] are shown on Fig. 18. The CSLAM scheme achieved better than second-order convergence with respect to all error measures.

Combining solid-body advection and the deformational flow field defines the moving vortex test case considered last. With a rotation angle of $\varphi = \pi/4$ the vortex passes over the corners of the cubed-sphere. Again we show convergence plots (Fig. 19), difference plots (Fig. 20) and the time evolution of the error measures (Fig. 21). We find better than second-order convergence for CSLAM-N (2.51, 2.59 and 2.53, respectively) and a degradation in accuracy when applying the monotone filter (Fig. 19). To compare with [38] and investigate the impact of the monotone filter on accuracy we show difference plots (Fig. 20) and time evolution of error measures (Fig. 19) at resolution $N_c = 80$. Errors are mainly associated with the sub-grid-cell reconstructions and the edges/corners of the cubed-sphere are not visible in the plots.

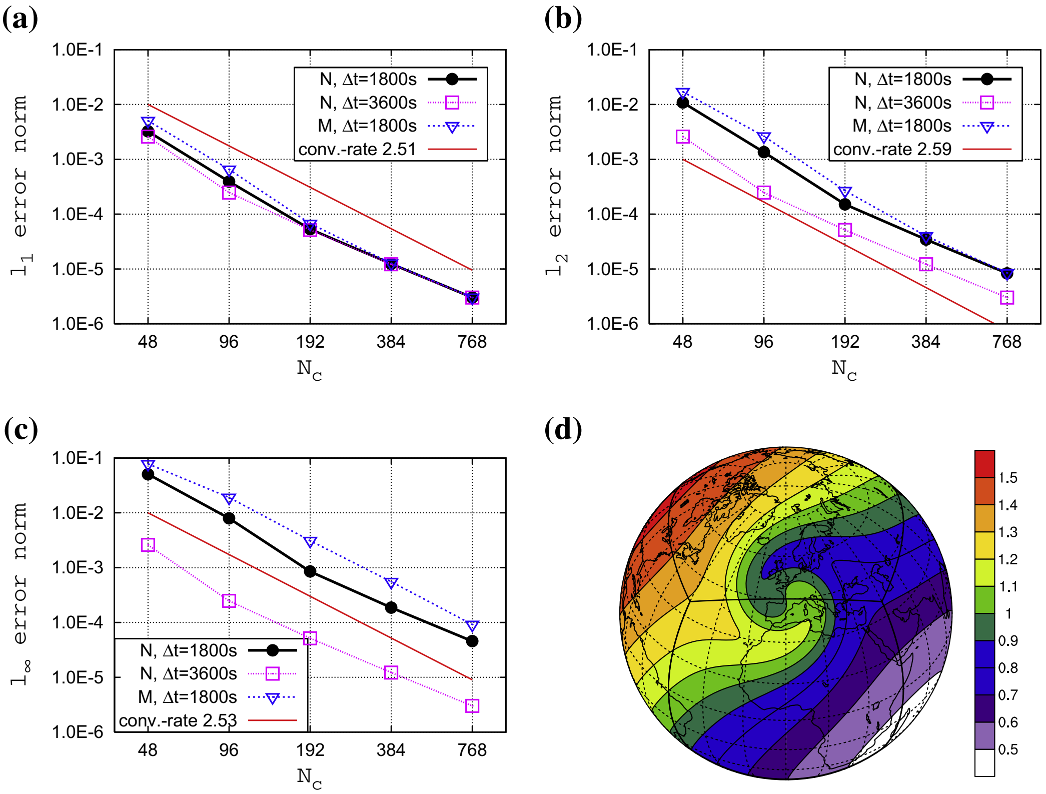


Fig. 19. Same as Fig. 18 but for the moving vortices test case. Settings are as in [38] and the traditional semi-Lagrangian tests in [37], that is, $\Delta t = 1800$ s and $\Delta t = 3600$ s (one revolution completed in 576 and 288 time-steps), respectively, and $\varphi = \pi/4$. (d) shows the analytical solution after a quarter revolution.

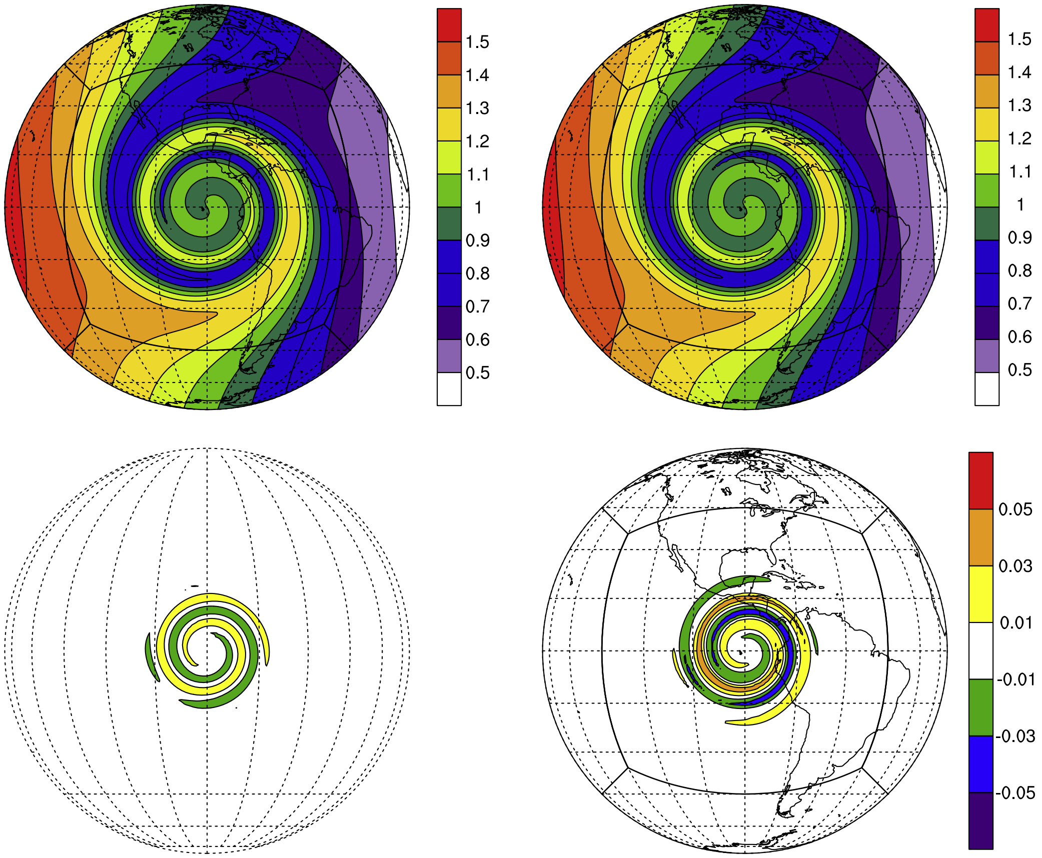
5. Summary and conclusions

In this paper we have developed a conservative semi-Lagrangian finite-volume multi-tracer transport scheme on the cubed-sphere grid. It is based on semi-Lagrangian upstream tracking of grid cells; more precisely, we compute upstream trajectories for cell vertices and the upstream Lagrangian cells are defined by connecting the vertex points with great-circle arcs. Having defined the upstream Lagrangian cells the transport problem is effectively reduced to a remapping problem, that is, to compute the mass enclosed in the deformed Lagrangian cells given the cell average values in the regular Eulerian cells. The remapping problem is solved by converting the area-integrals into line-integrals through the application of Gauss–Green's theorem. Line-integrals along grid lines are computed exactly while integrals along arbitrary lines are computed using Gaussian quadrature. Two quadrature points were found to be sufficient when using biquadratic polynomial reconstruction functions.

The fully two-dimensional nature of this scheme permits a rigorous treatment of the cubed-sphere edges and corners. An upstream cell located over an edge of the cubed-sphere is partitioned into parts overlapping just one face of the cube and mass enclosed in these sub-cells are computed separately on each panel.

The new scheme has been extensively tested in both Cartesian and spherical geometry using standard test cases. It has been demonstrated that the scheme allows for long time-steps and is competitive with other published schemes in terms of standard error norms. For example, CSLAM was found to be competitive with the recently developed cubed-sphere transport scheme presented in [38]. The importance of the cross term in the third-order reconstruction polynomials was also investigated. Even for solid-body advection on the sphere, where the Lagrangian cells are simply translated (and not deformed or rotated), the CSLAM scheme error measures could be improved by up to approximately 30% when including the cross term in the reconstruction function. The convergence rates for the scheme was shown to be better than second order with respect to all error norms and test cases. A monotone option is also available for the CSLAM scheme, and a flux-form version of CSLAM and improved limiters will be presented in a forthcoming paper.

For one tracer, the computational cost per degree of freedom is higher for CSLAM than for the schemes based on a regular latitude–longitude grid discussed herein due to the geometric complexity of CSLAM (computation of weights). However, the weights can be reused for each additional tracer and hence it is expected that for a certain number of tracers CSLAM will become computationally competitive. The virtue of CSLAM is its flexibility and generality at affordable cost (given the number of prognostic variables to be transported is large enough) rather than efficiency per degree of freedom.



Acknowledgments

The authors thank Dr. David Williamson (NCAR) for valuable comments made during internal review and Prof. Eigil Kaas (University of Copenhagen) for fruitful discussions during the development process. The authors gratefully acknowledge the careful review by Dr. Andrew Staniforth (UK Met Office) and an anonymous reviewer. The authors were partially supported by the DOE BER Program under award DE-SC0001658.

References

[1] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[2] A. Staniforth, J. Côté, Semi-Lagrangian integration schemes for atmospheric models – a review, Mon. Weather Rev. 119 (1991) 2206–2223.
[3] B. Machenhauer, E. Kaas, P.H. Lauritzen, Finite volume methods in meteorology, in: R. Temam, J. Tribbia, P. Ciarlet (Eds.), Computational methods for the atmosphere and the oceans, Handbook of Numerical Analysis, vol. 14, Elsevier, 2009, pp. 3–120.
[4] P.H. Lauritzen, P.A. Ullrich, R.D. Nair, Atmospheric mass schemes: desirable properties and a semi-Lagrangian view on finite-volume discretizations, in: P.H. Lauritzen, R.D. Nair, C. Jablonowski, M. Taylor (Eds.), Numerical techniques for global atmospheric models, Lecture Notes in Computational Science and Engineering, Springer, submitted for publication.
[5] P.H. Lauritzen, A stability analysis of finite-volume advection schemes permitting long time steps, Mon. Weather Rev. 135 (2007) 2658–2673.
[6] J. Dukowicz, Conservative rezoning (remapping) for general quadrilateral meshes, J. Comput. Phys. 54 (1984) 411–424.
[7] J.K. Dukowicz, J.W. Kodis, Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations, SIAM J. Sci. Stat. Comput. 8 (3) (1987) 305–321.
[8] J.D. Ramshaw, Conservative rezoning algorithm for generalized two-dimensional meshes, J. Comput. Phys. 59 (2) (1985) 193–199.
[9] P.W. Jones, First- and second-order conservative remapping schemes for grids in spherical coordinates, Mon. Weather Rev. 127 (1999) 2204–2210.
[10] P.A. Ullrich, P.H. Lauritzen, C. Jablonowski, Geometrically exact conservative remapping (GECoRe): regular latitude-longitude and cubed-sphere grids, Mon. Weather Rev. 137 (6) (2009) 1721–1741.
[11] W.D. Collins, P.J. Rasch, B.A. Boville, J.J. Hack, J.R. McCaa, D.L. Williamson, J.T. Kiehl, B. Briegleb, Description of the NCAR Community Atmosphere Model (CAM 3.0), NCAR Tech. Note, NCAR/TN-464+STR.
[12] J.-F. Lamarque, D.E. Kinnison, P. Hess, F. Vitt, Simulated lower stratospheric trends between 1970 and 2005: identifying the role of climate and composition changes, J. Geophys. Res. 113 (2008) D12301.
[13] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.
[14] R. Sadourny, Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids, Mon. Weather Rev. 100 (1972) 136–144.
[15] C. Ronchi, R. Iacono, P.S. Paolucci, The 'cubed sphere': a new method for the solution of partial differential equations in spherical geometry, J. Comput. Phys. 124 (1) (1996) 93–114.
[16] M. Rančić, J. Purser, F. Mesinger, A global shallow water model using an expanded spherical cube, Q. J. R. Meteor. Soc. 122 (1996) 959–982.
[17] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (2003) 266–298.
[18] W.H. Lipscomb, T.D. Ringler, An incremental remapping transport scheme on a spherical geodesic grid, Mon. Weather Rev. 133 (2005) 2335–2350.
[19] K.-S. Yeh, The streamline subgrid integration method: I. Quasi-monotonic second-order transport schemes, J. Comput. Phys. 225 (2007) 1632–1652.
[20] P.K. Smolarkiewicz, J.A. Pudykiewicz, A class of semi-Lagrangian approximations for fluids, J. Atmos. Sci. 49 (1992) 2082–2096.
[21] R.D. Nair, B. Machenhauer, The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, Mon. Weather Rev. 130 (3) (2002) 649–667.
[22] P.H. Lauritzen, E. Kaas, B. Machenhauer, A mass-conservative semi-implicit semi-Lagrangian limited area shallow water model on the sphere, Mon. Weather Rev. 134 (2006) 1205–1221.
[23] M. Rančić, Semi-Lagrangian piecewise biparabolic scheme for two-dimensional horizontal advection of a passive scalar, Mon. Weather Rev. 120 (1992) 1394–1405.
[24] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.
[25] S.F. Bockman, Generalizing the formula for areas of polygons to moments, Am. Math. Mon. 96 (1989) 131–132.
[26] C. Jablonowski, Adaptive grids in weather and climate modeling, Ph.D. Thesis, University of Michigan, 2004.
[27] J.L. McGregor, Semi-Lagrangian advection on conformal-cubic grids, Mon. Weather Rev. 124 (6) (1996) 1311–1322.
[28] P.H. Lauritzen, R.D. Nair, Monotone and conservative cascade remapping between spherical grids (CaRS): regular latitude–longitude and cubed-sphere grids, Mon. Weather Rev. 136 (2008) 1416–1432.
[29] M. Zerroukat, N. Wood, A. Staniforth, SLICE: a semi-Lagrangian inherently conserving and efficient scheme for transport problems, Q. J. R. Meteor. Soc. 128 (2002) 2801–2820.
[30] R.D. Nair, J.S. Scroggs, F.H.M. Semazzi, Efficient conservative global transport schemes for climate and atmospheric chemistry models, Mon. Weather Rev. 130 (8) (2002) 2059–2073.
[31] M. Zerroukat, N. Wood, A. Staniforth, Application of the parabolic spline method (PSM) to a multi-dimensional conservative semi-lagrangian transport scheme (SLICE), J. Comput. Phys. 225 (2007) 935–948.
[32] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.
[33] M.R. Norman, R.D. Nair, Inherently conservative nonpolynomial-based remapping schemes: application to semi-lagrangian transport, Mon. Weather Rev. 126 (2008) 5044–5061.
[34] C.A.I. Doswell, A kinematic analysis of frontogenesis associated with a nondivergent vortex, J. Atmos. Sci. (1984) 1242–1248.
[35] D. Williamson, J. Drake, J. Hack, R. Jakob, P. Swarztrauber, A standard test set for numerical approximations to the shallow water equations in spherical geometry, J. Comput. Phys. 102 (1992) 211–224.
[36] R.D. Nair, S.J. Thomas, R.D. Loft, A discontinuous Galerkin transport scheme on the cubed sphere, Mon.Weather Rev. 133 (2005) 814–828.
[37] R.D. Nair, C. Jablonowski, Moving vortices on the sphere: a test case for horizontal advection problems, Mon. Weather Rev. 136 (2008) 699–711.
[38] W.M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, J. Comput. Phys. 227 (1) (2007) 55–78.
[39] M. Zerroukat, N. Wood, A. Staniforth, SLICE-S: a semi-Lagrangian inherently conserving and efficient scheme for transport problems on the sphere, Q. J. R. Meteor. Soc. 130 (2004) 2649–2664.
[40] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, in: AIAA Paper No. 89-0366, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.
[41] R.D. Nair, J.S. Scroggs, F.H.M. Semazzi, A forward-trajectory global semi-Lagrangian transport scheme, J. Comput. Phys. 190 (2003) 275–294.
[42] P. Lauritzen, C. Jablonowski, M. Taylor, R. Nair, Rotated versions of the jablonowski steady-state and baroclinic wave test cases: a dynamical core intercomparison, J. Adv. Model. Earth Syst., Revising: See <http://adv-model-earth-syst.org/>.
[43] C. Chen, F. Xiao, Shallow water model on cubed-sphere by multi-moment finite volume method, J. Comput. Phys. 227 (10) (2008) 5019–5044.
[44] M. Zerroukat, N. Wood, A. Staniforth, A monotonic and positive-definite filter for a semi-Lagrangian inherently conserving and efficient (SLICE) scheme, Q. J. R. Meteor. Soc. 131 (611) (2005) 2923–2936.