A flux-form version of the Conservative Semi-Lagrangian Multi-tracer transport scheme (CSLAM) on the cubed sphere grid

Lucas M. Harris[a,*], Peter H. Lauritzen[b], Rashmi Mittal[c]

[a]*Department of Atmospheric Sciences, Box 351640, University of Washington, Seattle, WA, 98105; Current affiliation: NOAA/Geophysical Fluid Dynamics Laboratory, Princeton University, Princeton, NJ, 08540-6649*

[b]*Atmospheric Modeling and Predictability Section, Climate and Global Dynamics Division, NCAR Earth Systems Laboratory, National Center for Atmospheric Research, Boulder, Colorado*

[c]*Mesoscale and Microscale Meteorology Division, NCAR Earth Systems Laboratory, National Center for Atmospheric Research, Boulder, Colorado*

Abstract

A conservative semi-Lagrangian cell-integrated transport scheme (CSLAM) was recently introduced, which ensures global mass conservation and allows long timesteps, multi-tracer efficiency, and shape preservation through the use of reconstruction filtering. This method is fully two-dimensional so that it may be easily implemented on non-cartesian grids such as the cubed-sphere grid. We present a flux-form implementation, FF-CSLAM, which retains the advantages of CSLAM while also allowing the use of flux-limited monotonicity and positivity preservation and efficient tracer subcycling. The methods are equivalent in the absence of flux limiting or reconstruction filtering.

FF-CSLAM was found to be third-order accurate when an appropriately smooth

*Corresponding author. Tel: +1 609 452 5326, fax: +1 609 987 5063

Email addresses: lucas.harris@noaa.gov (Lucas M. Harris), pel@ucar.edu (Peter H. Lauritzen), mittal@ucar.edu (Rashmi Mittal)

[1]The National Center for Atmospheric Research is sponsored by the National Science Foundation.

initial mass distribution and flow field (with at least a continuous second derivative) was used. This was true even when using highly deformational flows and when the distribution is advected over the singularities in the cubed sphere, the latter a consequence of the full two-dimensionality of the method. Flux-limited monotonicity preservation, which is only available in a flux-form method, was found to be both less diffusive and more efficient than the monotone reconstruction filtering available to CSLAM. Despite the additional overhead of computing fluxes compared to CSLAM's cell integrations, the non-monotone FF-CSLAM was found to be at most only 40% slower than CSLAM for Courant numbers less than one, with greater overhead for successively larger Courant numbers.

Keywords:

2000 MSC: 2010 65M08, Advection, Conservation, Flux-form, Monotonicity, Cubed-sphere, Semi-Lagrangian, Piecewise Parabolic Method

---

1. Introduction

Inherently conservative semi-Lagrangian advection has been an active research subject since the pioneering work of [1] and the further development by [2], [3], [4], and [5, 6, 7]. Finite-volume semi-Lagrangian schemes enforce local conservation properties without *a posteriori* fixers [8], allows timesteps with Courant numbers greater than 1, and have been shown to be more accurate than traditional non-conservative semi-Lagrangian schemes in idealized tests [e.g. 9]. Recently, the conservative semi-Lagrangian schemes have been coupled with the momentum equations using semi-implicit timestepping to form shallow water models [10, 11, 12] and also extended to the three-dimensional hydrostatic equations of motion [13].

A locally conservative semi-Lagrangian scheme can be cast as either a dimensionally-split method or as a fully two-dimensional method. A dimensionally-split method

[c.f. 14] casts the two-dimensional interpolation problem into multiple applications of one-dimensional operators. Dimensionally split methods can either be cast as two one-dimensional sweeps in flow-dependent cascade directions [15], referred to as cascade interpolation, or in terms of fixed-direction splitting where the operators are applied along coordinate directions. [17] and [18] extended the non-conservative cascade interpolation method of [15] to an inherently conservative method. [16, 19] created a fully two-dimensional scheme from two one-dimensional operators composed in such a way as to eliminate the splitting error and directional bias typically resulting from dimensional splitting. This scheme is constructed to provide numerous desirable features, such as preserving a constant density field for nondivergent flows and preserving linear correlations between two scalars. Dimensionally split methods are easily extended to higher dimensions as only one-dimensional operators are needed, and can use any of a large number of one-dimensional shape-preservation methods (i.e. those preserving monotonicity or positive-definiteness) with relatively low computational cost. This includes recent schemes that are much less diffusive compared to conventional montonicity-preserving methods [e.g. 20]. The conservative splitting method of [18] has proven accurate for orthogonal meshes such as the regular latitude-longitude grid [e.g. 21]. However, extending dimensional splitting to geometries under consideration for weather and climate applications, such as icosahedral, cubed-sphere or completely unstructured meshes, is not obvious and fully two-dimensional semi-Lagrangian schemes may be more attractive.

Recently [22] (hereafter referred to as LNU) introduced a general fully two-dimensional semi-Lagrangian scheme referred to as CSLAM (Conservative Semi-LAgrangian Multi-tracer scheme), based on tracking Lagrangian cells moving with the flow and, at the end of each timestep, remapping (conservatively interpolating) variables back to the regular Eulerian mesh. The Lagrangian areas are simply

connected polygons over which area integrals can be directly evaluated, although this requires a complicated algorithm which takes every possible 2D shape into account [1]. [23] found that the area integrals can be converted into line integrals using Gauss-Green's theorem, reducing the problem to finding intersections between straight lines and so greatly simplifying the integration algorithm. The geometric computations for defining line segments only has to be done once per timestep and can be reused for each additional tracer, hence the scheme is a "multi-tracer scheme" similar to the incremental remapping scheme introduced by [24]. Both multi-tracer schemes and incremental remapping require more memory and computation time than a conventional advection scheme for a single tracer, but since the results of the line integrations can be re-used for other tracers the marginal cost for additional tracers is much less than for conventional schemes. Indeed, [25] found that for as few as seven tracer species their incremental remapping scheme was more efficient than a conventional Eulerian scheme. The efficiency introduced by multi-tracer schemes is becoming increasingly important in weather and climate applications as the number of prognostic tracers grows; for example, the chemistry version of NCAR's Community Atmospheric Model model [26] uses on the order of 100 prognostic tracers [27].

Fully two-dimensional transport schemes are more easily extended to non-cartesian grids compared to dimensionally-split schemes. CSLAM was implemented and tested for a particular cubed-sphere geometry but can, in theory, be extended to any spherical grid constructed from great-circle arcs such as is being done for icosahedral-type grids (Mittal et al., manuscript in preparation). CSLAM makes minimal approximations regarding the spherical geometry: The computational space is the gnomonic projection where the straight line connecting any two points is a great-circle arc on the sphere, and the integrals are computed exactly along lines parallel to coordinate

lines and by using Gaussian quadrature on other lines.

Flux-Form CSLAM, or FF-CSLAM, tracks the flux of mass through each Eulerian cell wall rather than tracking cells moving with the flow. FF-CSLAM is conceptually very similar to the incremental remapping scheme introduced by [24] and used on the sphere by [25], although incremental remapping uses 2D Gaussian quadrature to evaluate the area integrals instead of converting the area integrals to line integrals. Incremental remapping also does not allow timesteps for which the Courant number (defined in Sec. 4a) exceeds unity, and [24] only allowed a maximum Courant number of one-half to avoid evaluating integrals over non simply-connected regions.

Although CSLAM and FF-CSLAM will be shown to be formally equivalent, there are advantages of casting the scheme in flux form. In CSLAM shape-preservation is enforced by scaling the sub-grid-cell reconstruction functions (henceforth referred to as **filtering**), but a flux-form scheme may also use flux limiters or flux-corrected transport methods [28, referred to as **limiting**] as do Eulerian flux-form schemes[2]. Another advantage of a flux-form method is that it allows for timestep *sub-cycling* in models that solve the equations of motion with explicit time-differencing [e.g. 29]. Since the transport equation lacks fast wave modes which limit the timestep for the dynamics, tracer transport can use longer timesteps by accumulating each small timestep's fluxes over the long timestep. Sub-cycling is "free-stream" preserving since the tracer advection solution and the solution to the continuity equation for air become identical when the mixing ratio for the tracer is one. Models that sub-cycle air mass with respect to tracers can gain substantial computational savings [e.g. 26].

Other fully-two-dimensional methods have been implemented on the cubed-sphere.

[2]Reconstruction filtering is often called reconstruction limiting or slope limiting, although neither are "limiters" in the sense introduced here.

[30] introduced a scheme which prognoses several moments of the sub-grid-cell tracer distribution, which is fourth-order-accurate for both tracer advection and shallow-water problems. However, computing and storing multiple moments causes a substantial increase in overhead for each additional tracer species, and this method may become less competitive for large numbers of tracers. [31] uses a discontinuous Galerkin scheme on the cubed-sphere which is formally highly accurate, but imposes a severe Courant number restriction for stability—[31] perform some tracer advection tests with a Courant number as low as 0.005, two to three orders of magnitude smaller than that used by CSLAM, [30], and others.

This paper concerns the derivation of FF-CSLAM and its shape-preservation methods. We test several methods for sub-grid-cell reconstruction filtering as well as for flux limiting. In Section 2 CSLAM is introduced, FF-CSLAM is derived, shown to be equivalent to CSLAM, and its extension to the cubed-sphere geometry is briefly discussed. Section 3 describe the limiters and filters used in this paper. Results from standard as well as novel test cases using FF-CSLAM are presented in Section 4. Our findings are summarized in Section 5.

2. CSLAM and FF-CSLAM

2.1. Cartesian geometry

We first describe the CSLAM and FF-CSLAM methods in cartesian geometry. A cell-integrated Lagrangian form of the two-dimensional transport equation for a passive tracer can be written as

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

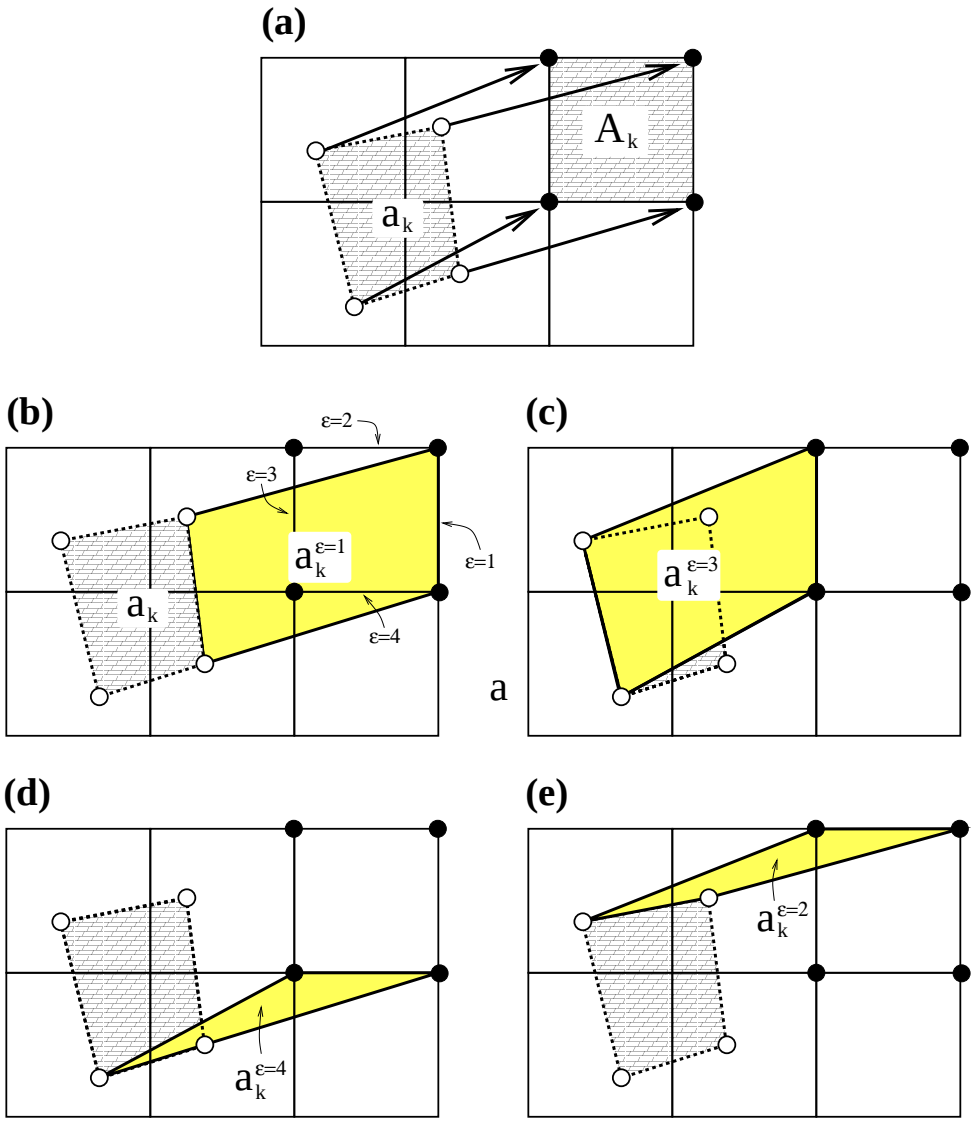


Figure 1: Illustration of the notation used for the (a) cell-integrated Lagrangian form of CSLAM and (b-e) flux-form version of CSLAM. (a) The Lagrangian area a_k ('brick' pattern) and the associated Eulerian area A_k ('brick pattern') at which a_k ends up after being transported by the flow as a material surface for one time step. The arrows show the trajectories for the vertices of a_k (unfilled circles) and A_k (filled circles). The cell face numbering ϵ used in the flux-form version of CSLAM is shown in (b) and the area swept through each Eulerian cell face during one timestep a_k^{ϵ}, $\epsilon = 1, 2, 3, 4$ is shown in (b), (c), (d) and (e) (shaded area), respectively.

[e.g. 24] where ψ is the continuous tracer density (typically the product of the air density and the tracer concentration per unit mass)[3], d/dt is the substantive (or Lagrangian) derivative, and the integration is over an arbitrary Lagrangian area $A(t)$—a material area that moves with the flow with no flux through its boundaries—at time t. A temporal discretization of (1) is given by

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

where Δt is the timestep length.

For a Lagrangian discretization in space, index the cells with subscript k with a total number of grid cells N. Denote the downstream cell $A_k = A(t + \Delta t)$ and the upstream cell $a_k = A(t)$. Here we consider upstream semi-Lagrangian schemes so A_k is a regular (Eulerian) grid cell and a_k is the upstream deformed (Lagrangian) cell transported to A_k after one timestep Δt. The discretized continuity equation for cell k can then be written in the *cell-integrated Lagrangian form*

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

where $\overline{(\cdot)}^{n+1}$ refers to the average value of ψ over A_k and $\overline{(\cdot)}^{n}$ is the average value of ψ over a_k. The areas ΔA_k and δa_k are the areas of A_k and a_k, respectively. Both CSLAM and FF-CSLAM assume the region a_k to be simply-connected. A graphical illustration of the scheme is given in Fig. 1a. In this paper we will assume a quadrilateral mesh although the method can be extended to more complicated meshes (such as hexagonal or unstructured grids) provided high-order subgrid-cell

[3]In this paper we will only consider nondivergent flow with constant, uniform air density, so we do not need to distinguish tracer mixing ratio and mass (cf. Appendix A and [32]).

reconstruction functions can be derived and the mesh is constructed from great-circle arcs. We also assume that the winds are available at the cell vertices, ie. defined on the Arakawa B-grid, although if they are not interpolated winds can be used.

Instead of integrating the equation of motion over a cell moving with the flow one may also discretize the continuity equation in Eulerian form, in which rather than tracking a cell moving with the flow the flux of mass through the Eulerian cell (A_k) walls are computed. This *flux-form* discretization can be written as

$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{n}\,\Delta A_k - \Delta t \sum_{\epsilon=1}^{4} \left(\overline{\overline{\langle \psi_k \rangle\, \vec{v}_k}} \cdot \vec{n} \Delta l \right)_\epsilon , \tag{4}$$

[e.g. 14] where $\overline{\overline{\cdot}}$ is the time-average over one timestep, $< \cdot >$ is the average in x or y direction of the cell, n_ϵ is the outward-directed unit vector normal to face ϵ, and Δl_ϵ is the length of face ϵ. Each term on the right-hand side of (4) represents the mass transported through one of the four Eulerian cell faces ϵ into the cell A_k during the time step. The area swept through face ϵ of cell A_k is denoted a_k^ϵ (see Fig. 1b–e for a graphical illustration). We will refer to a_k^ϵ as the "flux-area" for face ϵ. An x-flux area is a flux-area with $\epsilon = 1$ or 3, and a y-flux area is a flux-area with $\epsilon = 2$ or 4. Unlike the upstream region a_k the flux areas may not be simply connected, a possibility that must be considered by the solution algorithm.

The flux-form version of CSLAM is based on a semi-Lagrangian discretization of (4), which can be written as

$$\overline{\psi}_k^{n+1}\,\Delta A_k = \overline{\psi}_k^{n}\,\Delta A_k + \sum_{\epsilon=1}^{4} s^\epsilon F_k^\epsilon, \tag{5}$$

where

$$s^\epsilon = \begin{cases} +1 & \text{for inflow} \\ -1 & \text{for outflow.} \end{cases} \tag{6}$$

and the magnitude of the individual fluxes through each face is

$$F_k^\epsilon = \iint\limits_{a_k^\epsilon} \psi^n(x,y)\, dA, \tag{7}$$

where $\psi^n(x,y)$ is the continuous tracer density at timestep n. Note that $s^\epsilon F_k^\epsilon > 0$ indicates flux *into* cell A_k, and represents the increase in mass in that cell from flow through face ϵ during a timestep. The example depicted on Fig. 1 has $s^\epsilon = -1$ for $\epsilon = 1, 2$ and $s^\epsilon = 1$ for $\epsilon = 3, 4$. The notation used in (5) does not cover the special case where there is both inflow and outflow for a particular face, that is, a_k^ϵ overlaps a face, such as in Fig. 2c. In such a situation s^ϵ is multi-valued, and must be defined in terms of the individual overlap areas defined later in (9).

Assuming exact trajectories and an exact spatial distribution of ψ at time-level n, $\psi^n = \psi^n(x,y)$, the equations (3) and (5) are exact solutions to (1). The union of the areas used on the right-hand side of (4) with positive sign for inflow and negative sign for outflow equals the upstream Lagrangian area a_k:

$$\Delta A_k + \sum_{\epsilon=1}^{4} (s^\epsilon \cdot \delta a_k^\epsilon) = \delta a_k, \tag{8}$$

where δa_k^ϵ is the area of a_k^ϵ (see Fig. 1). Hence, as expected, both the flux-form and cell-integrated Lagrangian version of the discretized continuity equation are identical[4]. Guaranteeing mass conservation, whether local or global, in the cell-integrated

[4]A similar proof of the equivalence between flux-form and cell-integrated semi-Lagrangian

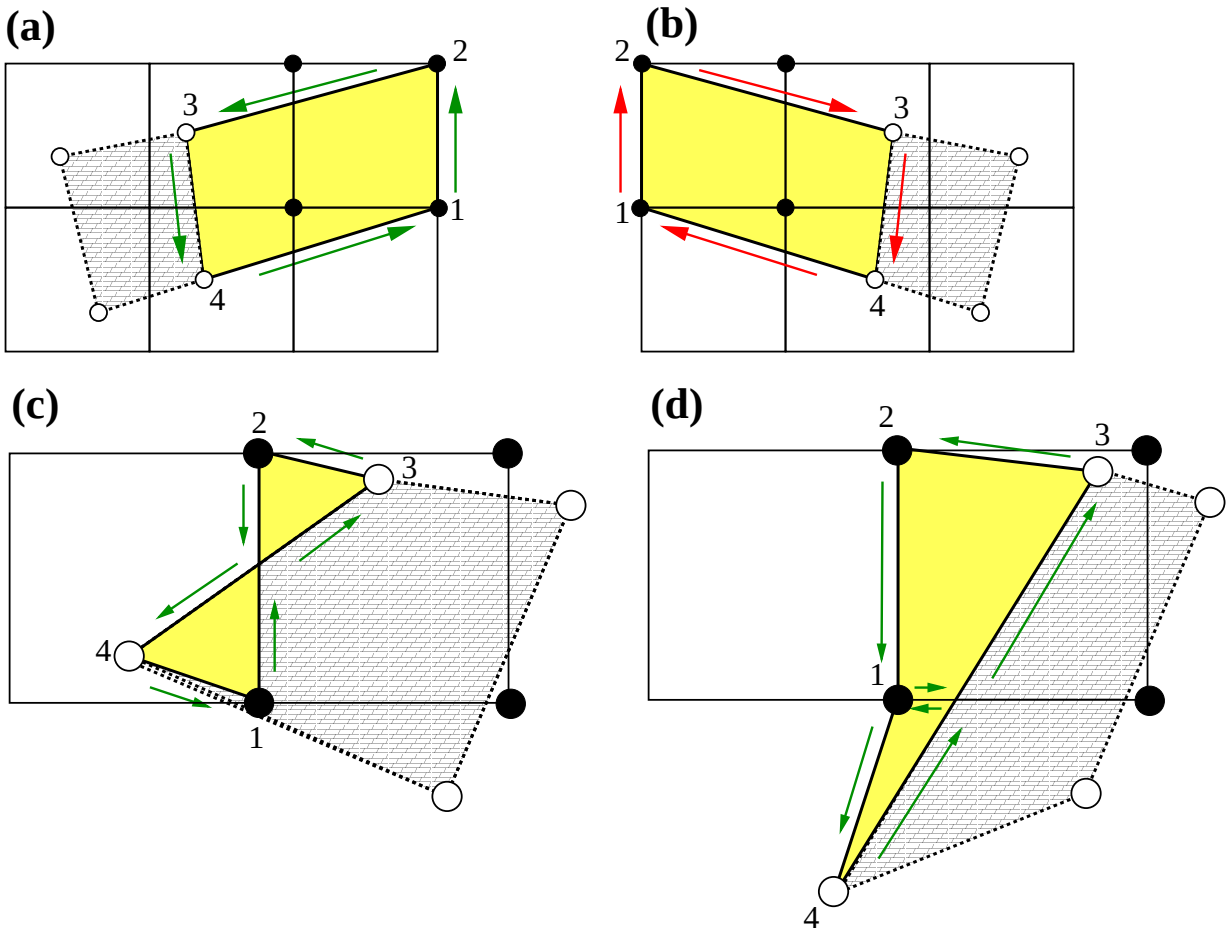


Figure 2: Examples of flux areas; green arrows represent properly oriented direction of integration, which in (c) and (d) is after the areas have been split up. Numbers on flux area vertices correspond to the default ordering of the vertices on the boundary. All other symbols are as in Fig. 1. Note that the Lagrangian area (brick pattern) in all four panels are simple quadrilaterals. (a) Counter-clockwise-oriented, convex flux area. (b) Clockwise-oriented, convex flux area; red arrows indicate incorrect orientation given by the default ordering of the vertices. (c) Non-simple flux area. (d) Concave flux area. Panels (c) and (d) have been enlarged for clarity.

Lagrangian version requires taking care that the upstream areas a_k span the domain without gaps or overlaps, while any flux-form scheme is inherently both locally and globally mass-conservative since the flux of mass into a cell along a face is equal to the flux out of the neighboring cell with which it shares the face.

Both the semi-Lagrangian (3) and flux-form (5) versions of CSLAM require integrating over areas a_k and a_k^ϵ, respectively. Since we will use local sub-grid-cell reconstructions within each cell (that are not necessarily continuous across cell boundaries), we need to integrate over the overlapping regions between a_k (or a_k^ϵ) and the Eulerian grid cells A_ℓ, for $\ell = 1, \ldots, N$. Define $a_{k\ell}$ as the non-empty overlap area between a_k and grid cell A_ℓ such that

$$a_{k\ell} = a_k \cap A_\ell, \text{ and } a_{k\ell} \neq \emptyset; \text{ for } \ell = 1, \ldots, L_k \text{ and } 1 \leq L_k \leq N, \tag{9}$$

where L_k is the number of non-empty overlap areas between cell a_k and the Eulerian grid cells A_k, which depends on the characteristics of the flow and timestep size. Similarly we define $a_{k\ell}^\epsilon$ as the overlap area between the flux-area a_k^ϵ and Eulerian cell A_ℓ. The number of overlap areas of a_k^ϵ are denoted L_k^ϵ.

If $f_\ell(x, y)$ is the sub-grid-cell distribution in Eulerian cell A_ℓ then CSLAM and FF-CSLAM can be written as

$$\overline{\psi}_k^{n+1} \, \Delta A_k = \begin{cases} \sum_{\ell=1}^{L_k} \iint\limits_{a_{k\ell}} f_\ell(x, y) \, dA & \text{CSLAM} \\ \\ \overline{\psi}_k^n \, \Delta A_k + \sum\limits_{\epsilon=1}^{4} \left[\sum\limits_{\ell=1}^{L_k^\epsilon} s_\ell^\epsilon F_{k\ell}^\epsilon \right] & \text{FF-CSLAM,} \end{cases} \tag{10}$$

schemes can be found in Section 2 of [14], and in [33].

respectively, where the individual overlap fluxes are

$$F_{k\ell}^{\epsilon} = \iint\limits_{a_{k\ell}^{\epsilon}} f_\ell(x,y)\, dA. \tag{11}$$

We recover (8) when $f_\ell = 1$ and the two right-hand-sides of (10) are equated. The sign-function s_ℓ^ϵ for inflow and outflow for each overlap area is defined as in (6), although unlike s^ϵ, s_ℓ^ϵ can only be single-valued.

The piecewise-parabolic subgrid cell reconstructions of the tracer field ψ are the same in FF-CSLAM as in CSLAM, which used fully two-dimensional biquadratic functions of the form

$$f_\ell(x,y) = \sum_{i+j\le 2} c_\ell^{(i,j)} x^i y^j = c_\ell^{(0,0)} + c_\ell^{(1,0)} x + c_\ell^{(0,1)} y + c_\ell^{(2,0)} x^2 + c_\ell^{(0,2)} y^2 + c_\ell^{(1,1)} xy, \tag{12}$$

in each Eulerian cell A_ℓ; here, the superscripts on x and y are exponents, not indices. The coefficients $c_\ell^{(i,j)}$ are chosen so that the average value of the reconstruction is equal to $\overline{\psi_\ell^n}$. (For more details, see LNU and [34].) Integrating these reconstructions over the overlap area $a_{k\ell}$ then yields

$$\iint\limits_{a_{k\ell}} f_\ell(x,y)\, dx\, dy = \sum_{i+j\le 2} c_\ell^{(i,j)} \iint\limits_{a_{k\ell}} x^i y^j dx\, dy = \sum_{i+j\le 2} c_\ell^{(i,j)} w_{k\ell}^{(i,j)}, \tag{13}$$

where each term in the sum is a product of a coefficient, which depends only on the values of the scalar variable, and an area weight $w_{k\ell}^{(i,j)}$ which depends only on the overlap area. (For notational simplicity, we have suppressed the superscript ϵ in the area weights.) Note that the area weights $w_{k\ell}^{(i,j)}$ are separated from the reconstruction coefficients $c_\ell^{(i,j)}$, implying that the weights can be re-used for each additional tracer; hence FF-CSLAM is a multi-tracer method.

The integral in (13) is evaluated not by directly integrating over $a_{k\ell}$ but instead converted into a more-easily evaluated line integral through Gauss-Green's theorem. Details are given in LNU; however, evaluating the line integrals of the flux areas $a^{\epsilon}_{k\ell}$ is more difficult than for Lagrangian areas a_ℓ. Line integrals are computed assuming a counter-clockwise orientation of the points, which by default is given by a list of successive vertices along the boundary of the region being integrated; the numbering of the vertices used by default CSLAM is shown for several example flux areas in Fig. 2. Simply-connected Lagrangian areas cannot have the orientations of their boundaries reversed from those of the corresponding Eulerian areas and will always be correctly oriented. This is not true for flux areas, for which the orientation depends on the flow direction. In Fig. 2a, a correctly-oriented flux area is shown: the order of the vertices on the flux area yields a counter-clockwise boundary. However, when the flow is reversed in Fig. 2b the resulting flux area is clockwise-oriented by default, and will need to be re-oriented by reversing the order of the vertices to be correctly integrated.

The proper orientation is determined by the location of the upstream points relative to the cell face in question. An x-flux (y-flux) area is counter-clockwise if both upstream points are leftward (downward) of the Eulerian cell's face and clockwise if both are rightward (upward) of the Eulerian cell's face. This can be seen for an x-flux area in Fig. 2ab. This method assumes that the flux area is a simple (one whose edges do not cross) convex quadrilateral; cases in which either assumption is violated must be handled separately. Fig. 2c depicts a non-simple flux area in which the flow direction normal to the cell face changes, which must be broken into two triangular flux areas which are separately oriented. The integration for a triangular area is no different than for a simply-connected convex quadrilateral. A similar procedure is carried out when the trajectories cross, which also creates a non-simple flux area.

Fig. 2d depicts a concave flux area, which is broken into a triangular flux area and the remaining quadrilateral flux area, both of which are again separately oriented and integrated. These four cases—counter-clockwise and clockwise convex simply-connected, concave simply-connected, and non-simple—cover all possible shapes of a quadrilateral region[5]. Regions with more sides would require the consideration of correspondingly more cases.

Of major importance for chemistry models [16] is the ability of an advection scheme to preserve linear correlations between tracer species. Appendix A presents a proof that CSLAM and FF-CSLAM satisfy this property, even when monotonicity or selective monotonicity preservation is used (but not when our positivity preservation methods are used). A consequence of preserving linear correlations is that a constant tracer mixing ratio field is also preserved.

2.2. Extension to cubed-sphere geometry

We consider cubed-sphere grids resulting from equi-angular gnomonic (central) projection

$$x = r\,\tan\alpha \quad \text{and} \quad y = r\,\tan\beta; \quad \alpha,\beta \in \left[-\frac{\pi}{4},\frac{\pi}{4}\right], \tag{14}$$

[35] where α and β are central angles in each coordinate direction, $r = R/\sqrt{3}$ and R is the radius of the Earth. The physical domain Ω (sphere) is represented by the gnomonic (central) projection of the cubed-sphere faces, $\Omega^{(\nu)} = [-1,1]^2$, $\nu = 1,2,\ldots,6$, and

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

[5]The degenerate case, a flux area with zero area, is simply handled by setting the value of the corresponding integral to zero.

where the panel domains $\Omega^{(\nu)}$ are non-overlapping and the cube edges are discontinuous. The gnomonic grid on each panel $\Omega^{(\nu)}$ constitutes a non-orthogonal curvilinear coordinate system that is only orthogonal at the center of each panel (6 points). In the discretized scheme the number of cells along a coordinate axis is denoted N_c so that the total number of cells in the global domain is $6 \times N_c^2$. Any straight line in the gnomonic projection corresponds to a great-circle arc on the sphere.

The extension of FF-CSLAM to the sphere is exactly as for CSLAM except for the handling of flux orientations between panels. Most of the Cartesian FF-CSLAM algorithm extends trivially to spherical cubed-sphere geometry. The flux areas a_k^ϵ are defined as polygons in gnomonic (x, y) coordinates which are spherical polygons on the sphere, and the straight line sides of the flux areas are now great-circle arcs on the sphere. However, Gauss-Green's theorem must be extended to gnomonic coordinates, the reconstruction functions must be computed on the sphere in gnomonic coordinates and the six cubed-sphere panels must be coupled. See LNU for further details.

The coupling of the panels is straightforward since FF-CSLAM is a fully two-dimensional scheme. The algorithm divides any flux-area a_k^ϵ into overlap areas $a_{k\ell}^\epsilon$. If a flux-area a_k^ϵ spans more than one panel the division of the area into overlap areas will divide the flux-area into overlapping areas that belong to different panels (Fig. 10 of LNU). The line-integrals are then performed on the panel that is local to the overlap area and thereafter the contributions to the flux-areas are collected across each panel. The collection process is slightly different than that described in LNU. The direction of the fluxes must be preserved when a flux overlap area is computed on one cubed-sphere face for a cell face on another cubed-sphere face, since the direction of basis vectors for each coordinate can change between faces. For example, in Fig. 3, flux A is in the positive y-direction in the front panel's coordinate

system, but in the *negative* x-direction on the top panel's coordinate system, and so a positive y-flux on the front panel becomes a negative x-flux on the top panel and vice versa. Likewise, flux C is a positive y-flux on the right-hand side panel but a negative y-flux on the top panel. Flux B needs no such revision as the positive x-direction is the same on the adjacent panels.

3. Monotonicity and positivity preservation

3.1. Limiters and Filters

A number of techniques exist to enforce monotonicity for solutions of the advection equation [36, 37, Ch. 5]. These can be classified into those which act *a posteriori* on the computed fluxes, and those which act *a priori* on a subgrid reconstruction function prior to computing the fluxes or performing an update step. The former will be referred to as "limiters", which includes both true flux limiters which act on a single flux and flux-corrected transport methods which combine a non-monotone high-order flux and a monotone flux. The *a priori* methods will be referred to as "filters", which have the advantage that they can be applied to both semi-Lagrangian and flux-form methods, whereas limiters are by definition restricted to flux-form methods. (Again see Footnote 2 in Sec. 1 about our definitions.)

Since CSLAM is fully two-dimensional, the methods used for monotonicity preservation must also be fully two-dimensional; however, few such methods exist for reconstruction polynomials of greater than second order. LNU used the filter of [38], which simply scales the subgrid reconstruction in each cell so that its extreme values are no greater than the maximum and minimum of the surrounding cells' average values; such a filter can be applied to reconstruction functions of any order. The filter first defines $\overline{\psi_\ell^{\min}}$, $\overline{\psi_\ell^{\max}}$ by $\min_{\ell\in K} \overline{\psi_\ell}$ and $\max_{\ell\in K} \overline{\psi_\ell}$, respectively, where K is the set

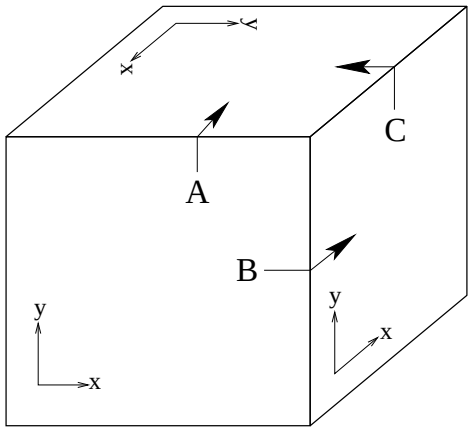


Figure 3: Three panels of the cubed sphere, with local coordinate systems and example fluxes A, B, and C. The front, right, and top panels shown here correspond to panels numbered 2, 3, and 6, respectively, in LNU.

of cells surrounding and including cell ℓ. The filter then replaces the sub-grid-cell distribution $f_\ell(x, y)$ in cell A_ℓ with:

$$f_\ell^{\text{mono}}(x, y) = \sigma_\ell^{\text{mono}} \left(f_\ell(x, y) - \overline{\psi_\ell} \right) + \overline{\psi_\ell} \tag{15}$$

where $\sigma_\ell^{\text{mono}}$ is chosen to be the largest value in $[0, 1]$ such that $\overline{\psi_\ell^{\min}} \leq f_\ell^{\text{mono}}(x, y) \leq \overline{\psi_\ell^{\max}}$ for all $(x, y) \in A_\ell$; if this cannot be satisfied, $\sigma_\ell^{\text{mono}}$ is set to zero, replacing the reconstruction with the constant value $\overline{\psi_\ell}$. This filter is computationally intensive, as it must check along each cell face and each corner as well as in the interior of A_ℓ for extreme values. LNU found this filter to be strongly diffusive. The effect of this filter is illustrated on a sample set of 1D data in Fig 4a; note in particular how the filter replaces the parabolic reconstruction in the cell containing the global maximum with a piecewise-constant reconstruction, substantially decreasing the order of accuracy of the method near that cell. The authors are not aware of any other fully two-dimensional filters for higher-order reconstruction functions in the literature.

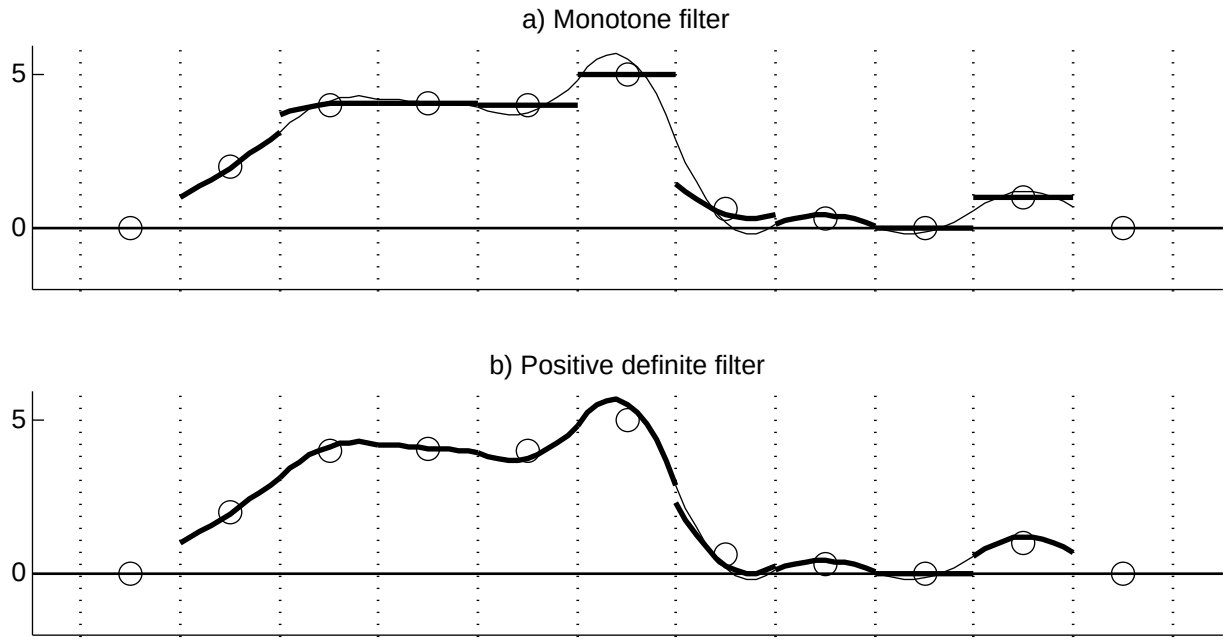


Figure 4: The effects of various filters (solid lines) on 1D piecewise-parabolic reconstructions (thin lines) from a set of cell-average values (open circles). (a) Monotone filter (15); (b) Positive-definite filter (24).

A few fully two-dimensional limiters exist [e.g. 39, 40]. In this paper, we will use the limiter of Zalesak [28], which is both easy to implement and is flexible enough to allow changes which may improve the method [cf. 41, 28, pg. 346–347]. Zalesak's limiter uses two fluxes, FF-CSLAM's unlimited high-order flux

$$F_k^\epsilon = \sum_{\ell=1}^{L_k^\epsilon} \iint_{a_{k\ell}^\epsilon} f_\ell(x,y)\, dA \tag{16}$$

and a monotone flux using piecewise-constant reconstructions:

$$F_k^{\epsilon,\text{mono}} = \sum_{\ell=1}^{L_k^\epsilon} \overline{\psi_\ell}^n \iint_{a_{k\ell}^\epsilon} dA. \tag{17}$$

A monotone solution is then computed using the low-order fluxes:

$$\overline{\psi}_k^{\text{mono}}\, \Delta A_k = \overline{\psi}_k^n\, \Delta A_k + \sum_{\epsilon=1}^{4} s^\epsilon F_k^{\epsilon,\text{mono}}, \tag{18}$$

which is then used to compute the flux-corrected solution:

$$\overline{\psi}_k^{n+1}\, \Delta A_k = \overline{\psi}_k^{\epsilon,\text{mono}}\, \Delta A_k + \sum_{\epsilon=1}^{4} s^\epsilon C_k^{\epsilon,\text{mono}} \left(F_k^\epsilon - F_k^{\epsilon,\text{mono}}\right), \tag{19}$$

in which the correction factor $C_k^{\epsilon,\text{mono}}$ is defined in Appendix C. In FF-CSLAM, the weights $\omega_{k\ell}$ have already been computed with the higher-order fluxes, and so computing the low-order flux requires little additional overhead.

3.2. Selective Limiting and Filtering

While the limiters and filters described in the previous section do avoid overshoots and undershoots in the solution, they all will incorrectly damp well-resolved

extrema. This motivated [42] to perform "selective" limiting or filtering, in which a limiter or filter is applied only where a smoothness metric exceeds a certain threshold, indicating either a steep gradient or discontinuity in the solution, and thus where a monotonicity preserving method is needed. [42] found that this procedure could greatly improve the solution near smooth extrema while causing only miniscule violations of monotonicity. Most notably, selective methods were found to be nearly as accurate as the piecewise-cubic monotonic scheme of [20] but about a third faster; and selective methods were as accurate as the Weighted Essentially Non-Oscillatory scheme of [43], but more efficient by a factor of 2.5.

The original selectivity metric first computed a smoothness parameter $\tilde{\Lambda}$ at each cell face, which is an analogue of a 1D Weighted Essentially Non-Oscillatory smoothness metric [44]. Monotonicity preservation was only applied at cell faces where $\tilde{\Lambda} > \Lambda^{\max}$, in which $\Lambda^{\max}$ is a parameter chosen so as to yield a favorable result in a range of test cases. [42] only applied selectivity to dimensionally-split methods; for FF-CSLAM, we present a two-dimensional extension of their smoothness metric. Define:

$$\gamma_k = \frac{1}{2}\left[\left(2\Delta x_k \frac{\partial f}{\partial x}_k\right)^2 + \left(\Delta x_k^2 \frac{\partial^2 f}{\partial x^2}_k\right)^2 + \left(2\Delta y_k \frac{\partial f}{\partial y}_k\right)^2 + \left(\Delta y_k^2 \frac{\partial^2 f}{\partial y^2}_k\right)^2 + \left(\Delta x_k \Delta y_k \frac{\partial^2 f}{\partial x \partial y}_k\right)^2\right] \tag{20}$$

for cell A_k with dimensions Δx_k by Δy_k. Since CSLAM computes each of the derivatives as part of the subgrid reconstruction process, they can be used directly in determining γ_k, without further discretizing (20).

Selectivity is applied differently depending on whether it is applied to a filter or a limiter. For the selective filter, the smoothness metric is computed for each cell,

rather than cell face:

$$\Lambda_k = \frac{\max_{k\in K}\gamma_k}{\min_{k\in K}\gamma_k + \varepsilon}. \tag{21}$$

where K is the set of indices of the nine cells surrounding and including the cell A_k, and ε is a small parameter preventing division by zero. If $\Lambda_k > \Lambda^{\max}$ in a cell, the filter is applied to the reconstruction; otherwise the filtering step is simply skipped.

When applying selectivity to the limiter, the smoothness metric is computed at each cell face:

$$\Lambda_k^{\epsilon} = \frac{\max_{k\in K\epsilon}\gamma_k}{\min_{k\in K\epsilon}\gamma_k + \varepsilon}. \tag{22}$$

in which K_ϵ is the set of indices of the cells surrounding the cell immediately upstream of face ϵ of cell A_k; the γ_k are all still defined on the cells and are not interpolated to the cell faces. The correction factor $C_k^{\epsilon,\text{mono}}$ in (19) is then replaced by

$$C_k^{\epsilon,\text{sel}} = \begin{cases} C_k^{\epsilon,\text{mono}} & \text{if } \Lambda_k^{\epsilon} > \Lambda^{\max} \\ 1 & \text{otherwise.} \end{cases} \tag{23}$$

Unlike [42], we use $\Lambda^{\max} = 60$. For higher-resolution simulations, and in particular higher-resolution deformational flow cases, the original value of 20 was found to yield a solution which was too diffusive, while using either 60 or 100 yielded nearly identical results. We expect the fully 2D Λ to be larger than $\tilde{\Lambda}$ due to the additional terms in (20) and in particular the two additional second derivative terms.

3.3. Positive definite schemes

Selectively-limited or filtered schemes guarantee neither strict monotonicity nor positivity. For applications in which even small negative values are undesirable, a positive-definite filter or limiter can be applied. A positive-definite limiter is created by replacing the correction factor $C_k^{\epsilon,\text{mono}}$ in the Zalesak limiter (20) by a second

correction factor $C_k^{\epsilon,\mathrm{PD}}$, defined in Appendix C. The monotone filter (15) can be made into a positive-definite filter by replacing $\sigma_\ell^{\mathrm{mono}}$ with

$$\sigma_\ell^{\mathrm{PD}} = \begin{cases} 1 & \text{if } f_\ell^{\min} \geq 0 \\ \frac{\overline{\psi_\ell}}{\overline{\psi_\ell} - f_\ell^{\min}} & \text{otherwise,} \end{cases} \tag{24}$$

in which $f_\ell^{\min} = \min_{(x,y)\in A_\ell} f_\ell(x,y)$. The positive-definite filter is illustrated in Fig. 4b. Unless otherwise noted, we will always apply the positive-definite limiter when using either the selective limiter or filter.

4. Results

4.1. Solid-body advection of a cosine bell

A steady, nondivergent wind field that will transport a tracer field around a great circle without distortion so that one revolution is completed in 12 days, at which time the exact solution is identical to the initial condition, is given by LNU and [45]:

$$\begin{aligned} u &= u_0 \cos\theta \left(\cos\varphi + \sin\varphi \cos\lambda\right) \\ v &= -u_0 \sin\varphi \sin\lambda, \end{aligned} \tag{25}$$

where λ is the longitude, θ the latitude, u, v the zonal and meridional components of the wind, respectively, $u_0 = 2\pi R/(12 \text{ days})$, φ is the rotation angle of the flow (0 for pure zonal flow, $\pi/2$ for over-the-pole flow), and R is the radius of the sphere. Details for the computation of the trajectories used to determine the locations of the upstream cells and flux areas are given in LNU. Unless otherwise noted, advection is performed with exact, analytic trajectories, so all of the error is due to remapping.

The initial tracer fields ϕ all consist of localized disturbances in the center of

one of the cubed sphere's equatorial panels. When combined with a flow field using $\varphi = \pi/4$ the disturbances will be advected through four of the corners and along two of the edges of the cubed sphere, yielding the most difficult test using solid-body advection. The first initial condition (IC) is a cosine bell, defined as:

$$\psi(\lambda,\theta) = \begin{cases} \frac{\psi_0}{4}\left[1+\cos\left(\frac{\pi R_g}{R_c}\right)\right]^2 & \text{if} R_g \leq R_c \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

where $R_c = R/3$ is the radius of the cone, $\psi_0 = 1000$, and R_g is the great-circle distance between (λ,θ) and the center of the distribution (λ_c,θ_c),

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

and $(\lambda_c,\theta_c) = (3\pi/2, 0)$, the center of an equatorial panel. This IC has continuous second and third derivatives (i.e. is $\mathcal{C}^3$), which differs from the standard cosine bell defined in [45] and used in LNU, among others. Unless otherwise noted, all simulations are set up so that the number of cells across each panel scales inversely with the timestep, as in Table 1, so the Courant number is the same as in a simulation with $N_c = 48$ and $\Delta t = 1800$ s. Here, the Courant number is defined as the trajectory length in a coordinate direction divided by the cell width in that coordinate direction, yielding an x-Courant number and a y-Courant number; we will always refer to the greater of the two, particularly when discussing the maximum CFL on the domain. For $N_c = 48$ and $\Delta t = 1800$ s, the maximum Courant number is about 0.38. All simulations will advect the IC once around the sphere.

First, a series of $N_c = 48$ simulations was performed with and without mono-

N_c	Δt	$\Delta\lambda$
24	3600 s	3.75°
48	1800 s	1.87°
96	900 s	0.937°
192	450 s	0.469°
384	225 s	0.234°

Table 1: Resolutions and timesteps used for simulations, excepting those otherwise noted. Rightmost column gives approximate equatorial cell width in degrees of longitude.

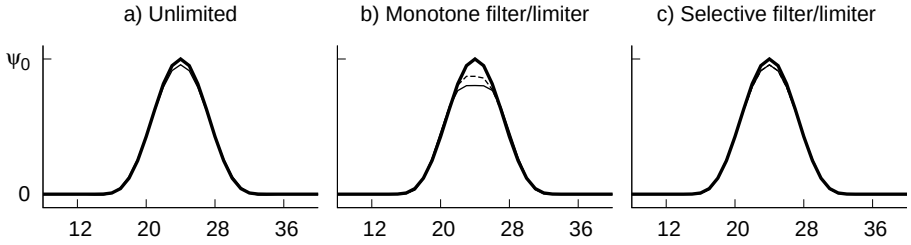


Figure 5: East-west cross-sections through the center of the $\mathcal{C}^3$ cosine bell with $N_c = 48$ after one revolution. Thin solid or dashed line denotes numerical solution, heavy solid line denotes exact solution. In (b) thin solid line is the solution with the monotone filter, and the thin dashed line is the solution with the monotone limiter.

	E_2	E_∞	max	min
Unlimited	0.036	0.042	0.96	-0.0066
Monotone filter	0.088	0.19	0.81	0
Monotone limiter	0.055	0.13	0.87	0
Selective filter	0.027	0.043	0.96	0
Selective limiter	0.030	0.043	0.96	0

Table 2: Error norms E_2 (27) and E_∞ (28) for the $\mathcal{C}^3$ cosine bell simulations, with $N_c = 48$. Maximum and minimum solution values are given in terms of ψ/ψ_0.

	E_2			
N_c	24	48	96	192
Unlimited	0.264	0.0357	3.89×10^{-3}	3.76×10^{-4}
Monotone filter	0.422	0.0882	0.0163	3.55×10^{-3}
Montone limiter	0.325	0.0554	8.24×10^{-3}	1.40×10^{-3}
Selective filter	0.243	0.0296	3.92×10^{-3}	4.84×10^{-4}
Selective limiter	0.236	0.0309	3.65×10^{-3}	4.28×10^{-4}

	E_∞			
N_c	24	48	96	192
Unlimited	0.305	0.0419	3.93×10^{-3}	4.43×10^{-4}
Monotone filter	0.550	0.194	0.0654	0.0238
Montone limiter	0.447	0.128	0.0361	0.0108
Selective filter	0.311	0.0433	3.93×10^{-3}	4.50×10^{-4}
Selective limiter	0.304	0.0430	3.93×10^{-3}	4.43×10^{-4}

Table 3: Error norms E_2 (27) and E_∞ (28) for the $\mathcal{C}^3$ cosine bell simulations plotted in Fig. 6. (Numerical values for the error norms in other cases in this paper are available upon request to the author.)

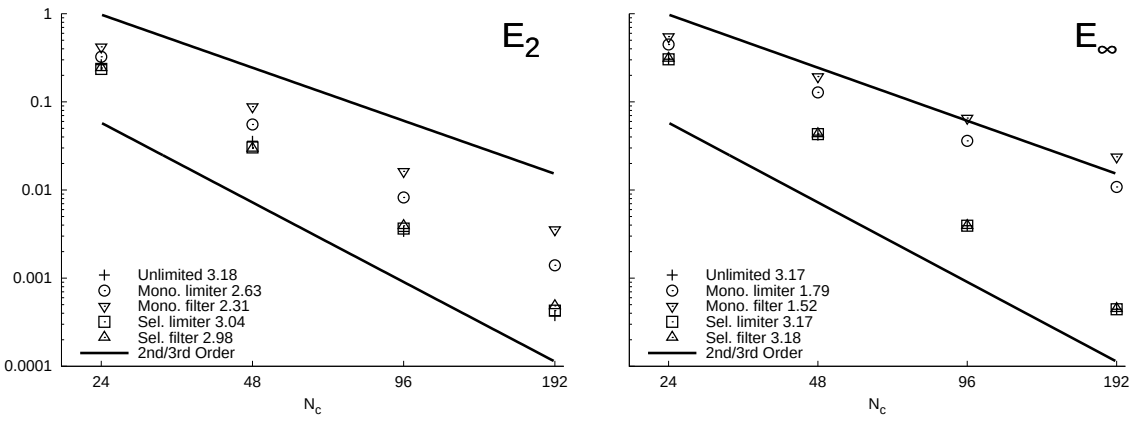


Figure 6: Error norms E_2 (27) and E_∞ (28) for the $\mathcal{C}^3$ bell as a function of the number of cells across a cube face. Numbers in legends correspond to the empirically-determined convergence rates $\mathcal{K}_2$ and $\mathcal{K}_\infty$ (29). Heavy lines correspond to the slopes of second and third-order convergence rates. Numerical values for the error norms are given in Table. 3.

tonicity preservation, and their errors compared (Table 2). The error norms are

$$E_2 = \sqrt{\frac{\sum_{\ell=1}^{N} \left[\left(\overline{\psi_\ell} - \overline{\psi_\ell^e} \right) \Delta A_\ell \right]^2}{\sum_{\ell=1}^{N} \left[\overline{\psi_\ell^e} \Delta A_\ell \right]^2}} \tag{27}$$

and

$$E_\infty = \frac{\max \left(\overline{\psi_\ell} - \overline{\psi_\ell^e} \right) \Delta A_\ell}{\max \overline{\psi_\ell^e} \Delta A_\ell}, \tag{28}$$

where $\overline{\psi_\ell^e}$ is the exact solution in grid cell k, and N is the total number of grid cells[6]. It is immediately apparent that the unlimited scheme (with no monotonicity preservation applied)[7] preserves more of the amplitude of the original cosine bell than do either of the monotone schemes, with the monotone filter being more diffusive than the monotone limiter. Cross-sections through the center of the bell (Fig. 5) clearly show greater damping when using the monotone methods compared to when no monotonicity preservation was used. The unlimited scheme does however cause oscillations outside of the bell, which are not large enough to be apparent in Fig. 5a but are indicated by the negative minimum value in Table 2. Using the selective methods (with the positive-definite limiter) not only eliminates the negative values, but since the peak of the bell is sufficiently well-resolved that Λ_k never exceeds $\Lambda_{\max}$, the selective methods do not apply any monotonicity preservation at that point, and the amplitude of the bell is preserved as well as when using the unlimited scheme (Fig. 5c). Note that eliminating the oscillations while preserving the bell's amplitude

[6]The E_1 error norm, defined by eqn. 41 in LNU, did not yield greatly different results than did the E_2 norm, except that unlimited schemes tended to have higher values of E_1 compared to the selective schemes due to the presence of small oscillations in the unlimited solution, which are more heavily weighted in the E_1 norm than in the E_2 norm.

[7]A solution using the unlimited cell-integrated CSLAM differed from the equivalent flux-form solution by no more than $2 \times 10^{-10} \psi_0$ for this case, demonstrating the equivalence (8) between the two methods.

causes the selective methods to yield smaller E_2 errors than the unlimited scheme.

To determine if our numerical scheme converges at the expected third-order accuracy, the errors for a series of model runs in which the width of the grid cells is successively halved (i.e. N_c successively doubled) are presented in Fig. 6. Empirical convergence rates $\mathcal{K}_2$ and $\mathcal{K}_\infty$, for E_2 and E_∞, respectively, are determined by performing a least-squares linear regression of the form

$$\begin{aligned} \log E_2 &= \mathcal{A}_2 - \mathcal{K}_2 \log N_c \\ \log E_\infty &= \mathcal{A}_\infty - \mathcal{K}_\infty \log N_c \end{aligned} \tag{29}$$

to the data; here, the $\mathcal{A}$'s are constants and are not used in this study. As expected, the unlimited scheme converges at third order in both error norms (Fig. 6), in spite of the bell being advected over the singularities at the corners of the cubed sphere. Tests in which the xy "cross term" in (12) is not used show that when the bell is advected over the singularities the errors are larger (Table 4) and the convergence rate slower (2.93 and 2.88 for E_2 and E_∞, respectively). The decrease in the convergence rate is easily seen when comparing different resolutions: disregarding the cross-term increases the error by only about a third at $N_c = 48$ (Table 4), but the error is doubled in an $N_c = 192$ simulation by neglecting the cross term. If over-the-pole advection is used instead ($\varphi = 90°$ in (26)) so that the solution does not encounter the singularities, using the cross term makes comparatively little difference in the error (Table 4) and convergence rates; even at $N_c = 192$ neglecting the cross-term only increases the error by six percent. The selective methods both converge at third order, corroborating the result of [42]; there is little difference in the convergence rate between selectively filtered and selectively limited solutions. The monotone methods both converge more slowly, as is the case for any monotone scheme, with

the monotone filter converging at the slowest rate.

What if we were to use the IC of Williamson et al. [45]? This IC is only C^1, and since determining the order of accuracy [cf. 46, for analysis of a similar 1D method] assumes a solution with at least a smooth second derivative, we cannot expect that this C^1 cosine bell will yield third-order convergence. Indeed, if we perform the same convergence tests using this IC, even the unlimited simulations do not converge at third order (Fig. 7) in either error norm. For these simulations, the oscillations created near the second deriviative's discontinuity at the base of the bell eventually become larger than the damping of the bell's peak as resolution increases, and dominate the error in both norms. A similar problem is seen for the selective methods. Similar sub-third-order convergence with this IC was found in the piecewise-parabolic schemes used by LNU and [47] and by the discontinuous Galerkin scheme of [31].

4.2. Solid-body advection of a slotted cylinder

A much more difficult test uses an IC with a slotted cylinder [28]. The formulation we use is

$$\psi(\lambda,\theta) = \begin{cases} \psi_0 & \text{if } R_g \le R_c \text{ and } |\lambda - \lambda_c| \ge R_c/6 \\ \psi_0 & \begin{array}{l}\text{if } R_g \le R_c \text{ and } |\lambda - \lambda_c| < R_c/6 \\ \quad \text{and } \theta - \theta_c < -\frac{5}{12} R_c\end{array} \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

where $\psi_0 = 1$ and $R_c = R/3$ is the radius of the cylinder. The width of the slot is one-sixth the diameter of the cylinder. The cylinder is advected over the corners of the cubed sphere in the same manner as the cosine bell ($\varphi = \pi/4$ in (26)). As expected, the unlimited scheme causes severe oscillations with this initial condition

	With cross term		Without cross term	
N_c	48	192	48	192
$\varphi = 45°$	0.036	3.76×10^{-4}	0.051	7.02×10^{-4}
$\varphi = 90°$	0.047	5.45×10^{-4}	0.048	5.80×10^{-4}

Table 4: E_2 error for unlimited $\mathcal{C}^3$ cosine bell simulations, with and without the cross term in (12) and at different advection angles.

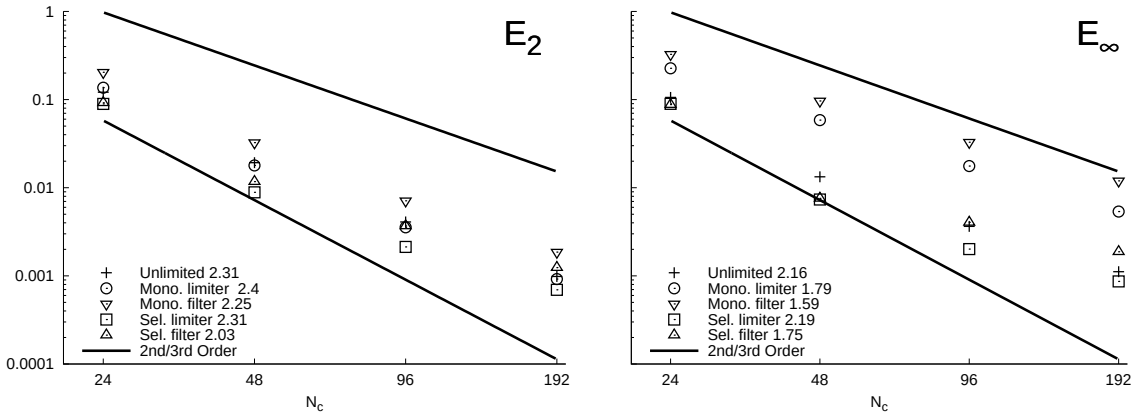


Figure 7: As in Fig. 6, but for the C^1 cosine bell.

(Fig. 8a) when using $N_c = 48$. Using any of the monotonicity-preserving methods eliminates the oscillations (Fig. 8b). The selective methods yield roughly the same solutions as their strictly-monotone counterparts (Fig. 8c), except that there are some overshoots at the top of the cylinder. Overshoots appearing in the selective simulations remain smaller than in the unlimited simulation (Table 5).

Since the solution is discontinuous, we do not expect even first-order convergence in space for any of our methods. This is borne out by the error norms listed in Table 5: doubling the resolution to $N_c = 96$ only decreases the E_2 norm by about one-third, and causes little difference in the E_∞ norm. Oscillations are still present in the unlimited case (Fig. 8d); the monotone (Fig. 8e) methods are able to retain most of the amplitude of the cylinder without any apparent oscillations. The selective methods (Fig. 8f) also retain the cylinder's amplitude but again the oscillations are smaller than for the unlimited simulation.

4.3. Advection by a deformational flow

FF-CSLAM is now tested by moving deformational vortices [48] advected through the corners of the cubed sphere. This test, identical to that in LNU, begins with a smooth tracer field that is deformed into narrow vortical filaments by the flow (Fig. 9), and poses a more difficult test for the numerical schemes than the solid-body rotation cases. Full details of the flow field, initial condition and exact solution in this test case are given in LNU, although we have added a constant background field so that the mean value of the IC is 1, and so that the solution should never approach zero; as a consequence we do not apply positivity preservation when using the selective methods in this case. Again, exact trajectories are used. Errors are primarily due to difficulty resolving the filaments near the end of the simulation and by the discontinuity in the flow field at the center of the vortex; as a result we do

	$N_c = 48$				$N_c = 96$			
	E_2	E_∞	max	min	E_2	E_∞	max	min
Unlimited	0.331	0.643	1.201	-0.0755	0.241	0.625	1.141	-0.166
Monotone filter	0.352	0.677	0.959	0	0.243	0.672	0.995	0
Monotone limiter	0.344	0.660	0.994	0	0.241	0.664	0.996	0
Selective filter	0.340	0.676	1.143	0	0.243	0.676	1.002	0
Selective limiter	0.338	0.672	1.129	0	0.243	0.668	1.114	0

Table 5: Errors for the slotted cylinder simulations.

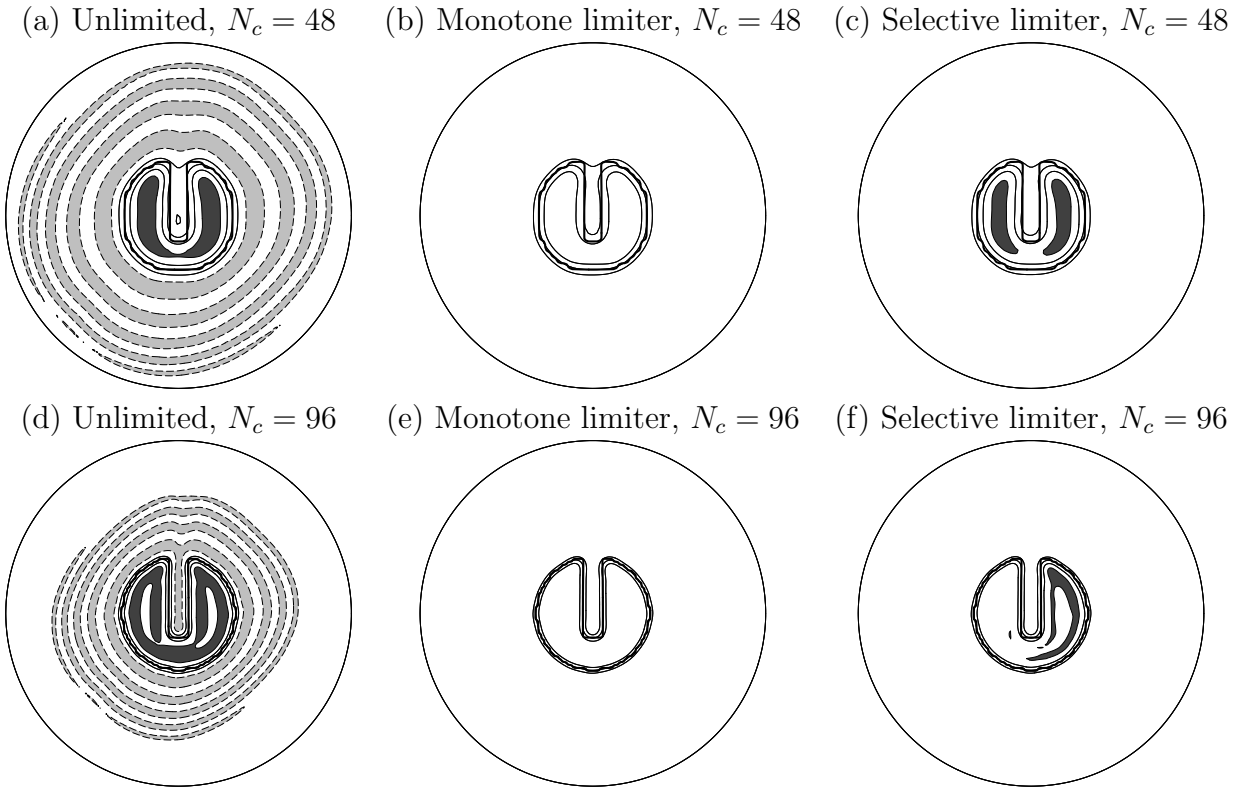


Figure 8: Solutions (plotted on the sphere) for the slotted cylinder simulations. Contour interval is $0.25\psi_0$; negative values are shaded light gray, values greater than ψ_0 are shaded dark gray. Heavy black line is the exact solution.

not expect third-order convergence from this test.

The error in the unlimited simulation (Fig. 10a) is nearly identical to that in the selectively-limited simulations (not shown), which is expected as the field away from the vortex center remains smooth throughout the simulation and the smoothness parameter (21) remains small. The errors become larger when the monotone limiter is used (Fig. 10b) and larger still when the monotone filter is used (Fig. 10c). Fig. 11 demonstrates the convergence rates for these tests; $N_c = 24$ is not shown, as by the end of the simulation the filaments in the vortex are no longer resolved for such a large cell size. The unlimited and the selectively-limited cases converge at the same rate in both error norms but fall short of third-order convergence, likely because of the flow field's discontinuity. Again, the monotone filtered and limited solutions are more diffused than either the unlimited or the selectively-limited solutions, and in particular have higher errors in the E_∞ norm.

Few other tests of a deformational flow on the cubed-sphere exist in the literature. [19] used a monotone Eulerian finite-volume scheme on this test and found a convergence rate of 1.53, similar to that of our solution with the monotone limiter in the ℓ_∞ norm (and somewhat lower than our solutions using the monotone filter), but found a convergence rate of 1.67 in the ℓ_2 norm, substantially lower than the convergence rates of either of our monotone simulations. That our scheme improves upon that of [19] in the ℓ_2 norm may indicate smaller phase errors, possibly due to our scheme's use of fully two-dimensional operators instead of two one-dimensional operators. [31] perform a similar test using a non-monotonic discontinuous Galerkin scheme; although they do not quote a convergence rate, their Fig. 11c indicates a convergence rate in the ℓ_2 norm of at most second order and certainly well short of third order.

Such a heavily deformational flow could create a heavily deformed flux area which

may not be closely approximated by a quadrilateral. Instead, a flux area constructed from a higher-order polygon can be used as suggested by LNU. To do this, consider an upstream cell with additional points along each of its sides (Fig. 12a), which are the upstream points from evenly-spaced points along the sides of the Eulerian cell. We then create flux areas by connecting the corners of the Eulerian cell to the same points of the upstream cell with straight lines (Fig. 12b), as was done when going from CSLAM to FF-CSLAM for a quadrilateral. Since CSLAM and FF-CSLAM are equivalent, any increase in accuracy gained by adding points to the upstream area will carry over to adding points *to the upstream end of the flux area alone*. Conversely, if we were to add points to the sides of the flux areas joining the upstream and Eulerian areas, there would be no change in the accuracy of the unlimited scheme since the additional area would be canceled out by the opposing flux area when the flux divergence is calculated upon taking a timestep. This equivalence of the flux-form and semi-Lagrangian methods allows us to use more points to define the flux area without having to consider every possible non-simple shape for a higher-order polygon.

Error norms when using additional upstream points for the deformational flow case and FF-CSLAM simulations with no monotonicity preservation are shown in Fig. 13. The additional points show only minor improvement in either the absolute error norms or in the convergence rate, and little to no improvement when going beyond one additional point. We conclude that it is not the error from approximating the flux area with a quadrilateral that is preventing us from attaining third-order convergence in our deformational flow simulations but instead the discontinuity in the flow.

Can third-order convergence be attained if a smooth deformational flow is used? Towards this end, tests are performed using a recently developed test case [32, test

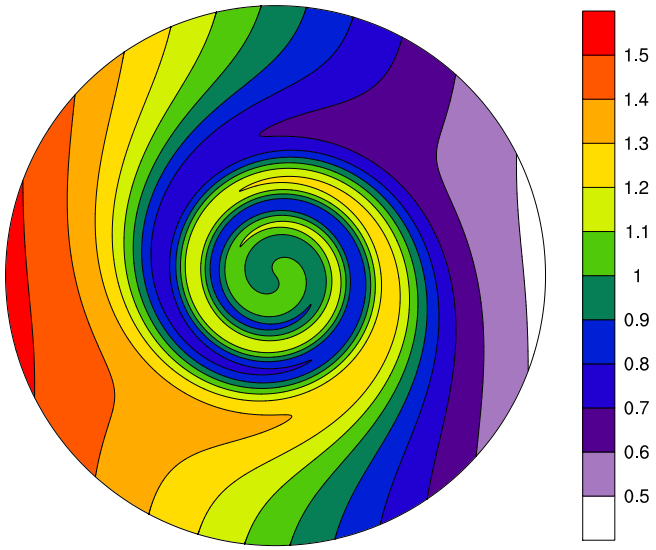


Figure 9: Exact solution (plotted on the sphere) for the deformational flow test at t =12 days, the end of the simulation. Extreme values are $1 \pm \tanh(3/5) \approx 1 \pm 0.54$.

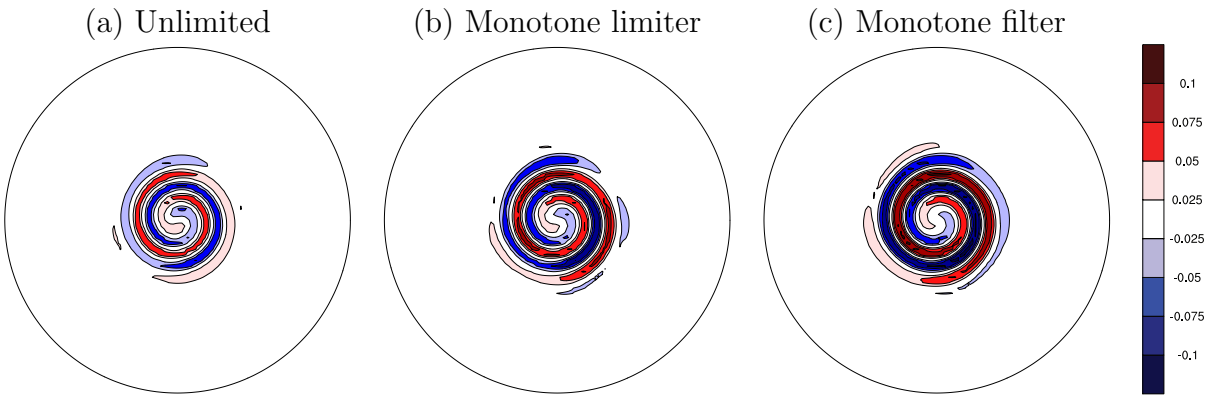


Figure 10: As in Fig. 9, but for the errors in three $N_c = 48$ simulations.

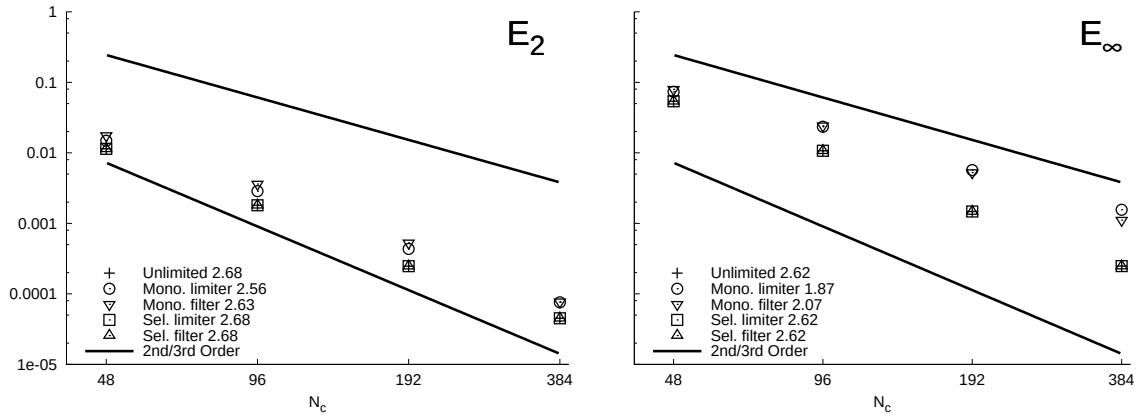


Figure 11: As in Fig. 7, but for the deformational flow test case.

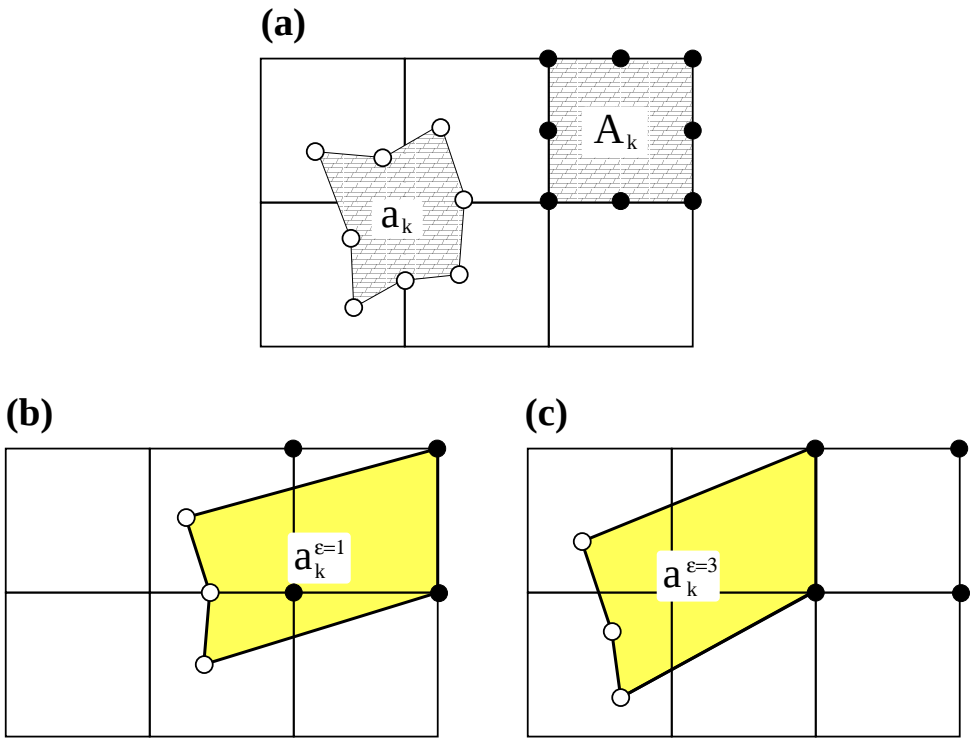


Figure 12: (a) Example of an upstream area defined using additional points. Flux areas defined for this area are shown for (b) $\epsilon = 1$ and (c) $\epsilon = 2$. Shading is as in Fig. 1.

2], which uses a strongly deformational nondivergent flow similar to that of [39] but extended to the sphere. The zonal and meridional wind components are

$$u(\lambda, \theta, t) = u_0 \cos\theta \sin^2\lambda \sin 2\theta \cos(\pi t/T) \quad (31)$$

$$v(\lambda, \theta, t) = u_0 \sin 2\lambda \cos\theta \cos(\pi t/T) \quad (32)$$

respectively, where the period T is 12 days and $u_0 = 10R/T$, in which R is again the radius of the sphere. The IC is two cosine bells (26) initially centered at $(\lambda_c, \theta_c) = (5\pi/6, 0)$ and $(7\pi/6, 0)$ (ICs are depicted in Fig 15 by red contours). The trajectories for this test case are non-trivial and are computed using high-order Taylor Series expansions [see 32, for details]. The timestep is $\Delta t = T/600$ for $N_c = 48$ and is halved for each successive doubling of N_c so as the Courant number remains constant across simulations, with a maximum of about 0.6.

The flow is constructed so that flow deforms the tracers into filaments by $t = T/2$ (Fig 15a), at which time the flow reverses and returns the tracer field to its initial position at $t = T$ (Fig 15bc). Errors are characterized by deviations from the IC at the end of the simulation.

The $\mathcal{C}^\infty$ flow and the $\mathcal{C}^3$ IC are sufficiently smooth as to test for third-order convergence. Indeed, a convergence rate greater than three is found in the absence of montonicity preservation (Fig. 14), although again the unlimited simulations cause oscillations in the solution (Fig 15). Some phase error is seen in lower-resolution simulations (Fig 15b), a consequence of the strong deformation of the flow at $t = T/2$ (Fig 15a); this is less apparent at higher resolutions (Fig 15c). The monotone filter and limiter yield particularly poor convergence rates in this test. The selective methods (used here with positivity preservation) exhibit better convergence rates than the monotone methods, and better than for the earlier deformational flow case

(Fig. 11), but fall short of third-order convergence.

4.4. Timing

The various methods presented in this paper can also be tested for their computational efficiency. Table 6 presents the time needed to complete the $\mathcal{C}^3$ cosine bell simulation for a variety of methods. First, the unlimited flux-form method is only at most about 40% slower than the unlimited semi-Lagrangian method, in spite of the fact that the flux-form scheme has to integrate over twice as many flux-areas as the semi-Lagrangian method does Lagrangian areas. This is presumably because each flux area tends to have fewer overlap areas than does the Lagrangian area when the Courant number is as small as in these simulations (0.38). It is also apparent that the monotone filter is much less efficient than the monotone limiter, as anticipated in Section 3a; in particular the additional overhead introduced by the flux-form method is comparatively small compared to that added by monotone filtering. The positive-definite filter is less efficient than the positive-definite limiter for the same reasons that the monotone filter is less efficient than the monotone limiter. This severely slows down selective schemes which use the filter to enforce positive definiteness, as would be the case in any simulation using the semi-Lagrangian scheme. Note that if the positive-definite limiter is used, selective filtering is in fact faster than selective limiting. This is because the time-consuming filtering is only done in the few cells where $\Lambda_k > \Lambda_{\max}$, while the behavior of selective limiting is not changed until (23) is applied at the last step of the limiter. This loss of efficiency is however only due to the particular implementation of selectivity, and it is possible to optimize the selective limiter by checking the smoothness parameter first, and then only performing the calculations if the threshold is exceeded.

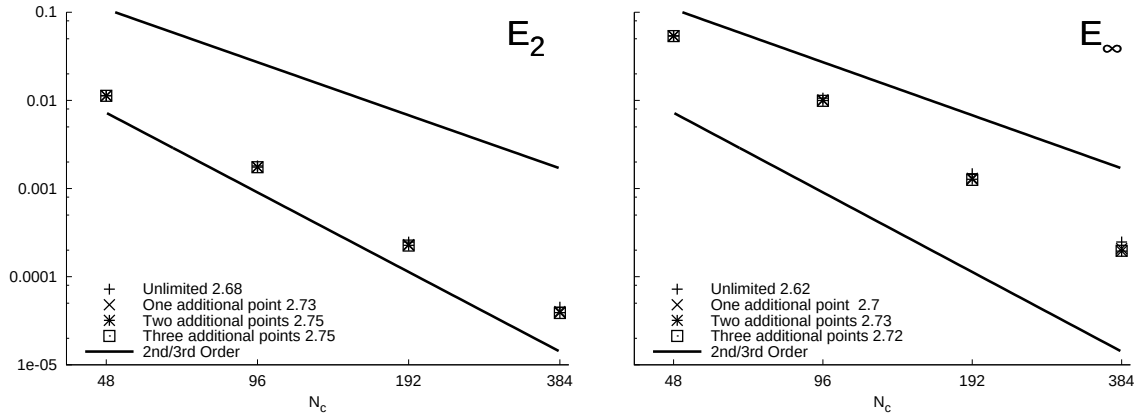


Figure 13: As in Fig. 7, but for unlimited simulations of the deformational flow test case with additional points used to define the upstream end of the flux areas (as in Fig. 12).

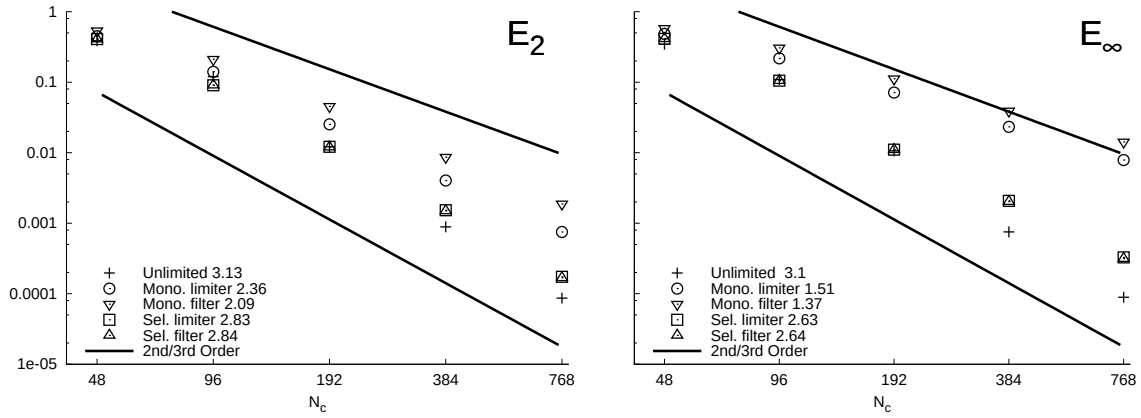
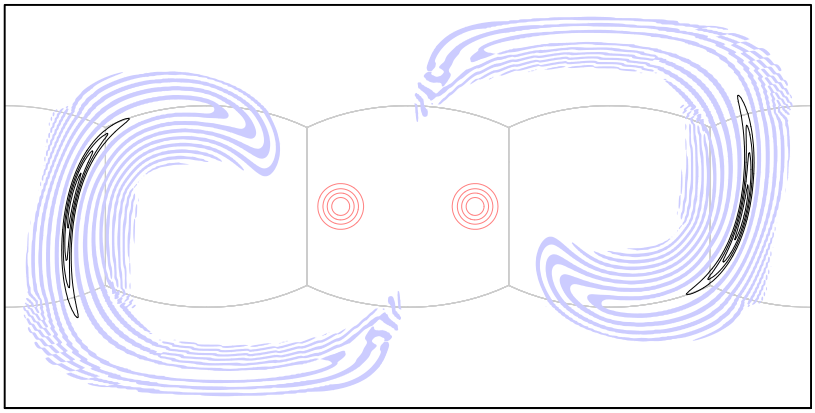
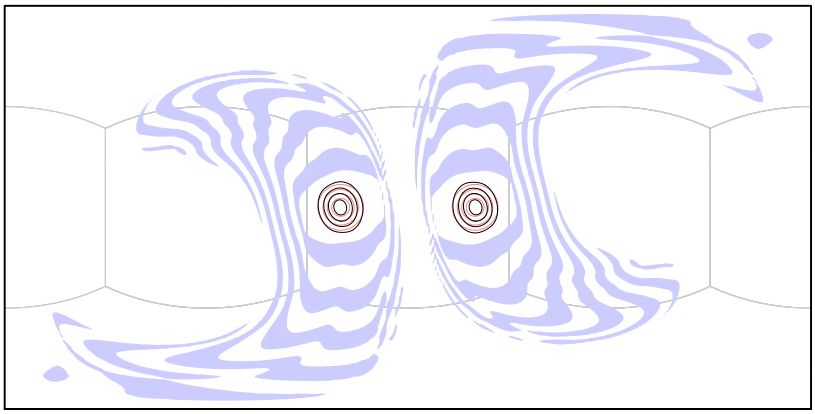


Figure 14: As in Fig. 7, but for the deformational flow test of [32].

(a) Unlimited $N_c = 96$ simulation, $t = T/2$



(b) Unlimited $N_c = 96$ simulation, $t = T$



(c) Unlimited $N_c = 192$ simulation, $t = T$

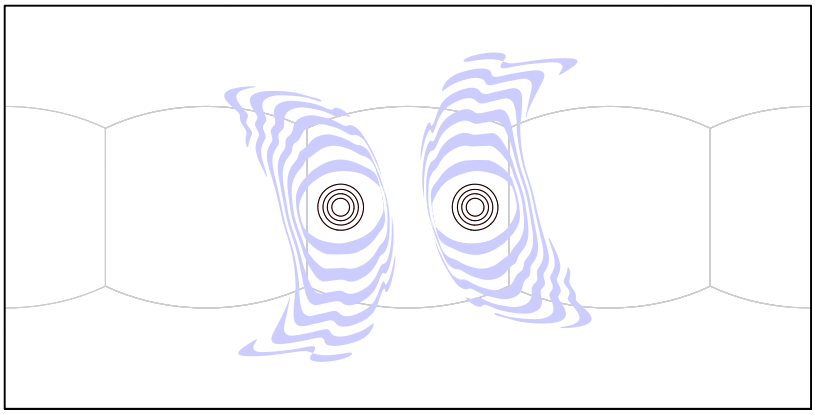


Figure 15: Example solutions (on a planar projection of the sphere) for the test case of [32]. Numerical solution given in black with negative values shaded, initial condition/exact solution at $t = T$ given in red. Contour interval is $0.2\psi_0$ for positive values. Lines in background are outlines of the cubed sphere panels.

N_c	24	48	96
Unlimited (SL)	0.35	2.35	19.2
Unlimited	0.45	3.25	25.5
PD limiter	0.61	4.64	38.4
Monotone limiter	0.85	6.16	53.9
Selective filter, PD limiter	1.30	7.63	53.7
Selective limiter, PD limiter	1.11	9.03	75.6
PD filter (SL)	2.13	12.9	90.3
Monotone filter (SL)	2.08	13.2	90.7
PD filter	2.13	13.8	98.5
Monotone filter	2.14	13.8	101
Selective filter, PD filter (SL)	2.81	15.7	106
Selective filter, PD filter	2.90	16.7	112

Table 6: Time (in seconds) required for the reconstruction, remapping, and limiting (if applied) steps for the $\mathcal{C}^3$ cosine bell solid-body rotation test case. All simulations are flux-form unless otherwise specified; here, "SL" refers to a semi-Lagrangian simulation and "unlimited" refers to a simulation with neither filtering nor limiting. Each simulation using the selective methods is performed with a positive definite (PD) method, either the filter or the limiter. Times are averages of five identical simulations using compiler-optimized Fortran 90 code on a 2.27 GHz Intel Xeon processor. Simulations are listed in order of increasing average time for the $N_c = 48$ case.

4.5. Efficiency and accuracy as a function of timestep

While both CSLAM and FF-CSLAM allow us to take longer timesteps than are allowed for traditional Eulerian methods, longer timesteps result in larger flux areas with more overlap areas, and computing the integrals may become progressively more expensive with increasing timestep. To test this, a series of unlimited simulations of the $\mathcal{C}^3$ cosine bell solid-body rotation test case with fixed $N_c = 48$ were performed (Table 7). It is apparent in both methods that for Courant numbers less than unity, doubling the timestep halved the time to complete the simulations, and the flux-form method is still at most 40% slower than the semi-Lagrangian method. However, as the Courant number grows beyond unity, the comparative efficiency of the flux-form method begins to decrease, and by $\Delta t = 28800$ s the flux-form method is actually slower than for a timestep of 14400 s. As the Courant number exceeds unity the flux-area will extend beyond the cell adjacent to the original Eulerian cell, increasing the number of overlap areas which need to be integrated. This decrease in efficiency at large Courant number does not occur for the semi-Lagrangian method, because the upstream areas do not enlarge with increasing Courant number. It may be possible to increase FF-CSLAM's efficiency by translating the downstream end of the flux areas upstream an integer number of cells in each direction, similar to the methods of [16] and [42], among others; further, it is possible to optimize the line integral computations for Courant numbers smaller than unity since only the nearest cell upstream of a particular Eulerian cell need to be considered. Both are held for further research.

Note that as the timestep is increased beyond 1800 s the errors successively decrease due to the smaller number of remappings that need to be performed, as was found by LNU. Alternately, for smaller timesteps the error begins to decrease again. [49] analyzed errors in cell-integrated finite-volume schemes and found that the

errors were least for integer Courant numbers and greatest for half-integer Courant numbers. Since at $\Delta t = 1800$ s the Courant number is close to 1/2 (Table 7) while for smaller timesteps it approaches 0, the decrease in errors with decreasing timesteps shorter than 1800 s is in line with the results of [49]. For $\Delta t \geq 7200$ s crossing trajectories are found in the solid-body advection flow, but this does not adversely affect either method as the error continues to decrease with timestep in spite of the crossing trajectories. This is because the in semi-Lagrangian method the upstream Lagrangian cell is not altered by the crossing of trajectories, and the flux-form method specifically accounts for crossing trajectories (Sec. 2a).

	Time (seconds)					
Δt	Semi-Lagrangian	Flux-form	% Increase	Max. Courant	E_2	E_∞
450 s	9.26	12.3	33%	0.087	0.0272	0.0304
900 s	4.74	6.12	29%	0.173	0.0324	0.0374
1800 s	2.35	3.25	38%	0.38	0.0357	0.0419
3600 s	1.27	1.68	32%	0.70	0.0251	0.0287
7200 s	0.752	1.07	42%	1.41	0.00512	0.00527
14400 s	0.450	0.938	110%	2.87	0.00369	0.00397
28800 s	0.386	1.25	1500%	5.89	0.00262	0.00240

Table 7: As in Table 6, but for unlimited simulations with $N_c = 48$ and variable timestep length. "% Increase" is the increase in time required for the flux-form scheme compared to the semi-Lagrangian scheme.

5. Conclusion

A flux-form version of the Conservative Semi-Lagrangian Multi-tracer transport scheme (CSLAM) has been presented on the cubed-sphere grid. This scheme, Flux-Form CSLAM (FF-CSLAM), retains the advantages of CSLAM such as full two-dimensionality, multi-tracer efficiency, global mass conservation, and long timesteps, while adding the advantages of flux-form schemes such as the use of flux-limited

monotonicity and positivity preservation as well as tracer sub-cycling to increase the efficiency of time-explicit dynamical solvers. FF-CSLAM yields identical solutions to CSLAM in the absence of flux limiting.

FF-CSLAM has been tested on a number of problems, including solid-body rotation advection and more stringent deformational flows. It was found that if a sufficiently-smooth initial condition and flow is used, the method is third-order accurate in space. The full two-dimensionality of the scheme allows third-order accuracy even when the tracer disturbance is advected over the singularities of the cubed-sphere grid or when a highly deformational flow is used.

Shape preserving methods in both CSLAM and FF-CSLAM are limited to those which are fully two-dimensional and can act on reconstructions of arbitrary order. While CSLAM had only a crude *a priori* reconstruction filter available to it, several fully two-dimensional flux limiters are available to FF-CSLAM. Monotone limiting—which requires a flux-form method—was found to yield lower errors and to better preserve extrema than CSLAM's monotone filter. Other methods could also be implemented: selective limiting and filtering were found to both preserve monotonicity as well as well-resolved extrema while remaining third-order accurate.

Despite the additional overhead required by computing twice as many flux areas in FF-CSLAM as Lagrangian areas in CSLAM, in the absence of filtering or limiting the flux-form scheme required at most 40% more computation time than CSLAM for Courant numbers less than unity. For larger Courant numbers the increasing size of the flux areas and their greater number of overlap areas can increase this overhead. This decrease in efficiency over CSLAM is acceptable since flux-form schemes can be used to perform sub-cycling, allowing the tracer timestep to be as much as several times longer than the dynamical timestep, compensating for this overhead. The flux limiters (both monotonicity- and positivity-preserving) available only to a flux-form

method were also found to be much more efficient than the monotone and positive-definite filters available to CSLAM.

Acknowledgements: This manuscript was greatly improved by comments from Dale Durran, David L. Williamson, Peter Blossey, Shian-Jiann Lin, and two anonymous reviewers. The authors also thank Paul Ullrich for his help understanding the remapping code and how the reconstructions are computed, and Rich Loft and NCAR SIParCS for funding LMH's participation. RM's participation was supported by NCAR Advanced Study Program. This work was partially supported by NSF grant ATM-0836316 and the DOE BER Program under award DE-SC0001658.

Appendix A. Preservation of linear correlations

Preserving linear correlations between two tracer species is an important property for advection schemes intended for chemistry models. This is discussed at length by [16] who present examples of methods which do or do not preserve correlations. An advection scheme preserves linear correlations between the mixing ratios of two tracer species, p and q, initially related by

$$q^n = \alpha p^n + \beta \tag{A.1}$$

if at the next timestep $q^{n+1} = \alpha p^{n+1} + \beta$, where α and β are constants. FF-CSLAM (and therefore CSLAM) satisfy this property as a consequence of its piecewise-parabolic reconstructions and the lack of 1D operator splitting which can violate correlation preservation if not properly constructed [16]. The proof will be for a model which separately prognoses the air density π and does not assume a nondivergent flow.

The flux-form air density equation

$$\frac{\partial \pi}{\partial t} + \nabla \cdot (\pi \vec{v}) = 0, \tag{A.2}$$

is discretized by FF-CSLAM as

$$\overline{\pi}_k^{n+1} \;=\; \overline{\pi}_k^n + \frac{1}{\Delta A} \sum_{\epsilon=1}^{4} \left\{ \sum_{\ell=1}^{L_k} \sum_{i,j} s_\ell^\epsilon c_\ell^{(i,j)}(\pi)\, \omega_{k\ell}^{(i,j)} \right\}. \tag{A.3}$$

Here, the $c_\ell^{(i,j)}(\pi)$ are the reconstruction coefficients for π, and other notations are as in Sec. 2.1. Define the air density flux through face ϵ by

$$\overline{\pi}_k^\epsilon = s_k^\epsilon \frac{\sum_\ell^{L_k} \sum_{i,j} c_\ell^{(i,j)}(\pi)\, \omega_{k\ell}^{(i,j)}}{\delta a_k^\epsilon}, \tag{A.4}$$

for which (A.3) becomes

$$\overline{\pi}_k^{n+1} = \overline{\pi}_k^n + \frac{1}{\Delta A} \sum_{\epsilon=1}^{4} \overline{\pi}_k^\epsilon \delta a_k^\epsilon. \tag{A.5}$$

The flux-form tracer continuity equation

$$\frac{\partial (\pi q)}{\partial t} + \nabla \cdot (\pi q \vec{v}) = 0, \tag{A.6}$$

is discretized by assuming $\overline{\pi}_k^\epsilon$ is uniform along each face as in [16, pg. 2063], for which FF-CSLAM becomes

$$\overline{\pi}_k^{n+1}\, \overline{q}_k^{n+1} = \overline{\pi}_k^n\, \overline{q}_k^n + \frac{1}{\Delta A} \sum_{\epsilon}^{4} \overline{\pi}_k^\epsilon \left\{ \sum_{\ell}^{L_k} \sum_{i,j} c_\ell^{(i,j)}(q^n) \omega_{k\ell}^{(i,j)} \right\}, \tag{A.7}$$

using (10), (11) and (13). Recalling that by (13) $\delta a_k = \sum_{\ell=1}^{L_k} \sum_{i,j} c_\ell^{(i,j)}(1)\, \omega_{k\ell}^{(i,j)}$, (A.7) reduces to (A.5) for $q = 1$, demonstrating consistency in the sense of [16]: the discretization of a spatially uniform tracer mixing ratio reduces to that for air density, regardless of whether the flow field is nondivergent.

The reconstruction coefficients satisfy

$$c_\ell^{(i,j)}(\alpha p^n + \beta) = \alpha\, c_\ell^{(i,j)}(p^n) + \beta c_\ell^{(i,j)}(1), \tag{A.8}$$

which is a general property true for piecewise-parabolic and other reconstructions which are linear in the cell-averaged values, but is not true for schemes such as the piecewise rational method of [50]. Equation (A.8) holds even if the monotonicity preserving methods described in this paper are used, since neither the scaling factor $\sigma_\ell^{\text{mono}}$ or the correction factor $C_k^{\epsilon,\text{mono}}$ are altered by replacing ψ with $\alpha\psi + \beta$ (see Appendix C for a proof of the latter). The selectivity parameter (21), (22) is also trivially unchanged between a pair of linearly-correlated flows.

Equation (A.7) can be written

$$\begin{aligned}
\overline{\pi}_k^{n+1}\, \overline{q}_k^{n+1} &= \overline{\pi}_k^n\, (\alpha \overline{p}_k^n + \beta) + \frac{1}{\Delta A} \sum_\epsilon^4 \overline{\pi}_k^\epsilon \left(\sum_\ell^{L_k} \sum_{i,j} \left(\alpha c_\ell^{(i,j)}(p^n) + \beta c_\ell^{(i,j)}(1) \right) \omega_{k\ell}^{(i,j)} \right) \\
&= \alpha \left(\overline{\pi}_k^n\, \overline{p}_k^n + \frac{1}{\Delta A} \sum_\epsilon^4 \overline{\pi}_k^\epsilon \sum_\ell^{L_k} \sum_{i,j} c_\ell^{(i,j)}(p^n)\, \omega_{k\ell}^{(i,j)} \right) \\
&+ \beta \left(\overline{\pi}_k^n + \frac{1}{\Delta A} \sum_\epsilon^4 \overline{\pi}_k^\epsilon \sum_\ell^{L_k} \sum_{i,j} c_\ell^{(i,j)}(1)\, \omega_{k\ell}^{(i,j)} \right) \\
&= \alpha \overline{\pi}_k^{n+1}\, \overline{p}_k^{n+1} + \overline{\pi}_k^{n+1}\, \beta = \overline{\pi}_k^{n+1} \left(\alpha \overline{p}_k^{n+1} + \beta \right)
\end{aligned} \tag{A.9}$$

where we have twice again used (A.7) substituting either p or 1 for q as necessary. Therefore FF-CSLAM preserves linear correlations *even when our monotonicity pre-*

serving methods are used. Numerical tests of solid-body rotation of the $\mathcal{C}^3$ cosine bell with $N_c = 48$ reveal preservation of initially correlated fields better than to one part in 10^4 in unlimited, monotone limited, and selectively limited simulations after one rotation around the sphere.

As a consequence of preserving linear correlations, a constant tracer field ($\alpha = 0$) is also preserved. Note that we have made no assumption of nondivergence upon the flow field; see Appendix B for a brief discussion of the meaning of nondivergence in a discretized system.

Our positive-definite methods do not preserve linear correlations, but the positive-definite limiter can be modified to do so (see Eqn. 43 of [42]). Note that neither cell-integrated schemes such as FF-CSLAM or Eulerian schemes such as that of [16] preserve nonlinear correlations, which is also important in modeling chemical reactions.

Appendix B. Nondivergent flows in cell-integrated semi-Lagrangian schemes

We briefly discuss the conditions under which a flow is nondivergent, i.e. under which a constant, uniform air density field is preserved. For $\overline{\pi}_k^\epsilon = s_k^\epsilon \pi_0$ (A.5) becomes

$$\begin{aligned} \overline{\pi}_k^{n+1} &= \pi_0 + \frac{\pi_0}{\Delta A} \sum_{\epsilon=1}^{4} s_k^\epsilon \delta a_k^\epsilon \\ &= \pi_0 + \frac{\pi_0}{\Delta A} \left(\delta a - \Delta A \right) \\ &= \pi_0 \frac{\delta a_k}{\Delta A}. \end{aligned} \tag{B.1}$$

in which (8) has been used. Therefore, a constant density field is preserved in a cell-integrated method if the flow preserves areas, ie. is nondivergent:

$$\delta a_k = \Delta A. \tag{B.2}$$

This condition is analogous to Eq. 2.11 in [16] for their Eulerian scheme, which is simply that the first-order finite-difference discretization of the flow divergence is zero. Note that point values from a continuous nondivergent flow need not satisfy the discrete Eulerian nondivergence condition although such a flow can be easily constructed by taking finite differences of a streamfunction.

Appendix C. Correction factor for the monotone and positive-definite limiters

The correction factor $C_k^{\epsilon,\text{mono}}$ for the monotone limiter (19), as defined in [28], is derived by first defining the antidiffusive fluxes

$$F_k^{\epsilon,A} = F_k^{\epsilon} - F_k^{\epsilon,\text{mono}}.$$

The net incoming antidiffusive flux P_k^+, and outgoing antidiffusive flux P_k^-, for cell A_k are computed:

$$\begin{aligned} P_k^+ &= \sum_{\epsilon=1}^{4} \max\left(0, s^{\epsilon} F_k^{\epsilon,A}\right) \\ P_k^- &= -\sum_{\epsilon=1}^{4} \min\left(0, s^{\epsilon} F_k^{\epsilon,A}\right). \end{aligned}$$

We then wish to correct the antidiffusive flux so that the solution (19) for the cells adjacent to the face ϵ remain in a range of values such that the solution is monotone. If we define the extreme permissible values $\overline{\psi_k^{\min}}$, $\overline{\psi_k^{\max}}$ for $\overline{\psi_k}^{n+1}$ as $\min_{k\in K}\left(\overline{\psi_k}^{n}, \overline{\psi_k^{\text{mono}}}\right)$ and $\max_{k\in K}\left(\overline{\psi_k}^{n}, \overline{\psi_k^{\text{mono}}}\right)$, respectively, where K is the set of cells surrounding and

including cell A_k, then we define the maximum antidiffusive flux into, and minimum antidiffusive flux out of, cell A_k so as to preserve monotonicty as:

$$\begin{aligned} Q_k^+ &= \left(\overline{\psi_k^{\max}} - \overline{\psi_k^{\text{mono}}}\right) \Delta A_k \\ Q_k^- &= \left(\overline{\psi_k^{\text{mono}}} - \overline{\psi_k^{\min}}\right) \Delta A_k. \end{aligned}$$

If we then define

$$R_k^+ = \begin{cases} \min\left(1, Q_k^+ / P_k^+\right) & \text{if } P_k^+ > 0 \\ 0 & \text{if } P_k^+ = 0 \end{cases}$$

$$R_k^- = \begin{cases} \min\left(1, Q_k^- / P_k^-\right) & \text{if } P_k^- > 0 \\ 0 & \text{if } P_k^- = 0, \end{cases}$$

and then denote k^ϵ as the index of the cell sharing face ϵ with cell k, the correction factor $C_k^{\epsilon,\text{mono}}$ for face ϵ is given by

$$C_k^{\epsilon,\text{mono}} = \begin{cases} \min\left(R_{k^\epsilon}^+, R_k^-\right) & \text{if } s^\epsilon F^{\epsilon,A} \leq 0 \\ \min\left(R_{k^\epsilon}^-, R_k^+\right) & \text{if } s^\epsilon F^{\epsilon,A} > 0. \end{cases}$$

How does the correction factor change between a pair of linearly correlated tracers? Let $F(p)$ and $F^{\text{mono}}(p)$ represent the piecewise-parabolic and monotone fluxes, respectively, along some cell face for a tracer mixing ratio p. Let $q = \alpha p + \beta$ as in Appendix A with $\alpha \neq 0$. Since both the piecewise-parabolic and monotone fluxes are linear in the cell-averaged values, and since $F(\beta) = F^{\text{mono}}(\beta)$ for any constant

β, the antidiffusive flux is

$$\begin{aligned} F^A(q) &= F(\alpha p+\beta) - F^{\text{mono}}(\alpha p+\beta) && \text{(C.1)} \\ &= \alpha\left[F(p) - F^{\text{mono}}(p)\right] + F(\beta) - F^{\text{mono}}(\beta) && \text{(C.2)} \\ &= \alpha F^A(p). && \text{(C.3)} \end{aligned}$$

If $\alpha > 0$, $P^+(q) = \alpha P^+(p)$ and similarly for P^-; if $\alpha < 0$ then $P^+(q) = \alpha P^-(p)$ and vice versa. Trivially, the same is true for Q^+ and Q^-. Finally, in the case $P^+(q) > 0$ and $\alpha > 0$ we have

$$R^+(q) = \min\left(1, Q^+(q)/P^+(q)\right) = \min\left(1, Q^+(p)/P^+(p)\right) = R^+(p). \quad \text{(C.4)}$$

The same result holds for $R^-(q)$; in turn, $C_k^{\epsilon,\text{mono}}$ is the same for both p and q and therefore our monotone limiter preserves linear correlations. A similar analysis can be performed for the monotone filter.

The positive-definite correction factor $C_k^{\epsilon,\text{PD}}$ is formulated as a modification [37] of the monotone correction factor which does not use the antidiffusive fluxes. Define the total outward flux by

$$P_k^{\text{PD}} = -\sum_{\epsilon=1}^{4} \min\left(0, s^\epsilon F_k^\epsilon\right).$$

The maximum outward flux permitted, so as to keep the solution positive definite, is given by

$$Q_k^{\text{PD}} = \overline{\psi_k^n}\Delta A_k.$$

Then,

$$R_k^{\mathrm{PD}} = \begin{cases} \min\left(1, Q_k^{\mathrm{PD}}/P_k^{\mathrm{PD}}\right) & \text{if } P_k^{\mathrm{PD}} > 0 \\ 0 & \text{if } P_k^{\mathrm{PD}} = 0, \end{cases} \tag{C.5}$$

and the correction factor becomes:

$$C_k^{\epsilon,\mathrm{PD}} = \begin{cases} R_k^{\mathrm{PD}} & \text{if } s^\epsilon F^\epsilon \leq 0 \\ R_{k^\epsilon}^{\mathrm{PD}} & \text{if } s^\epsilon F^\epsilon > 0. \end{cases}$$

References

[1] M. Rančić, Semi-Lagrangian piecewise biparabolic scheme for two-dimensional horizontal advection of a passive scalar, Mon. Wea. Rev. 120 (1992) 1394–1405.

[2] L. M. Leslie, J. R. Purser, Three-dimensional mass-conserving semi-Lagrangian scheme employing forward trajectories, Mon. Wea. Rev. 123 (1995) 2551–2566.

[3] J. Laprise, A. Plante, A class of semi-Lagrangian integrated-mass (SLIM) numerical transport algorithms, Mon. Wea. Rev. 123 (1995) 553–565.

[4] M. Rančić, An efficient, conservative, monotonic remapping for semi-Lagrangian transport algorithms, Mon. Wea. Rev. 123 (1995) 1213–1217.

[5] B. Machenhauer, M. Olk, On the development of a cell-integrated semi-lagrangian shallow water model on the sphere, ECMWF workshop proceedings: Semi-Lagrangian methods (1996) 213–228.

[6] B. Machenhauer, M. Olk, The implementation of the semi-implicit scheme in cell-integrated semi-lagrangian models., in: C. Lin, R. Laprise, H. Ritchie

(Eds.), Numerical Methods in Atmospheric and Oceanic Modelling - The André J. Robert Memorial Volume, CMOS/NRC Research Press, 1997, pp. 103–126.

[7] B. Machenhauer, M. Olk, Design of a semi-implicit cell-integrated semi-Lagrangian model, Max Planck Institute for Meteorology Tech. Rep. 265, Hamburg, Germany, 76-85. (1998).

[8] A. Priestley, A quasi-conservative version of the semi-Lagrangian advection scheme, Mon. Wea. Rev. 121 (1993) 621–632.

[9] R. D. Nair, B. Machenhauer, The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, Mon. Wea. Rev. 130 (2002) 649–667.

[10] P. H. Lauritzen, E. Kaas, B. Machenhauer, A mass-conservative semi-implicit semi-Lagrangian limited area shallow water model on the sphere, Mon. Wea. Rev. 134 (2006) 1205–1221.

[11] M. Zerroukat, N. Wood, A. Staniforth, A. White, J. Thuburn, An inherently mass-conserving semi-implicit semi-lagrangian discretisation of the shallow-water equations on the sphere, Quart. J. Roy. Meteor. Soc. 135 (2009) 1104–1116.

[12] J. Thuburn, M. Zerroukat, N. Wood, A. Staniforth, Coupling a mass conserving semi-Lagrangian scheme (SLICE) to a semi-implicit discretization of the shallow-water equations: Minimizing the dependence on a reference atmosphere, Q. J. R. Meteorol. Soc. 136 (2009) 146–154.

[13] P. H. Lauritzen, E. Kaas, B. Machenhauer, K. Lindberg, A mass-conservative

version of the semi-implicit semi-Lagrangian HIRLAM, Q.J.R. Meteorol. Soc. 134 (2008) 1583–1595.

[14] 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 14 (2009). Elsevier, 2009, pp.3-120.

[15] R. Purser, L. Leslie, An efficient interpolation procedure for high-order three-dimensional semi-Lagrangian models, Mon. Wea. Rev. 119 (1991) 2492–2498.

[16] S. Lin, R. Rood, Multidimensional flux-form semi-Lagrangian transport schemes, Mon. Wea. Rev. 124 (1996) 2046–2070.

[17] R. D. Nair, J. S. Scroggs, F. H. M. Semazzi, Efficient conservative global transport schemes for climate and atmospheric chemistry models., Mon. Wea. Rev. 130 (2002) 2059–2073.

[18] M. Zerroukat, N. Wood, A. Staniforth, SLICE: A semi-Lagrangian inherently conserving and efficient scheme for transport problems, Q. J. R. Meteorol. Soc. 128 (2002) 2801–2820.

[19] W. M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, J. Comput. Phys. 227 (2007) 55–78.

[20] 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. Meteorol. Soc. 131 (2005) 2923–2936.

[21] M. Zerroukat, N. Wood, A. Staniforth, Application of the parabolic spline method (PSM) to a multi-dimensional conservative semi-lagrangian transport scheme (SLICE), J. Comp. Phys. 225 (2007) 935–948.

[22] P. H. Lauritzen, R. D. Nair, P. A. Ullrich, A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid., J. Comput. Phys. (2009).

[23] J. Dukowicz, Conservative rezoning (remapping) for general quadrilateral meshes, J. Comput. Phys. 54 (1984) 411–424.

[24] J. K. Dukowicz, J. R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.

[25] W. H. Lipscomb, T. D. Ringler, An incremental remapping transport scheme on a spherical geodesic grid., Mon. Wea. Rev. 133 (2005) 2335–2350.

[26] 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 (2004).

[27] 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).

[28] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.

[29] S.-J. Lin, A 'vertically Lagrangian' finite-volume dynamical core for global models, Mon. Wea. Rev. 132 (2004) 2293–2307.

[30] C. Chen, F. Xiao, Shallow water model on cubed-sphere by multi-moment finite volume method, J. Comput. Phys. 227 (2008) 5019–5044.

[31] R. D. Nair, S. J. Thomas, R. D. Loft, A discontinuous galerkin transport scheme on the cubed sphere., Mon.Wea.Rev. 133 (2005) 814–828.

[32] R. D. Nair, P. H. Lauritzen, A class of deformational flow test-cases for the advection problems on the sphere, J. Comp. Phys. In review (2010).

[33] P. H. Lauritzen, P. A. Ullrich, R. D. Nair, Atmospheric transport schemes: Desirable properties and a semi-lagrangian view on finite-volume discretizations, in: P. Lauritzen, R. Nair, C. Jablonowski, M. Taylor (Eds.), Numerical Techniques for Global Atmospheric Models, Lecture Notes in Computational Science and Engineering, Springer, Submitted for publication.

[34] P. A. Ullrich, P. H. Lauritzen, C. Jablonowski, Geometrically exact conservative remapping (GECoRe): Regular latitude-longitude and cubed-sphere grids., Mon. Wea. Rev. 137 (2009) 1721–1741.

[35] 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 (1996) 93–114.

[36] R. LeVeque, Finite volume methods for hyperbolic problems, Cambridge university press, 2002.

[37] D. D. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer-Verlag, 1998.

[38] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes., Proc. AIAA 27th Aerospace Sciences Meeting, Reno (1989).

[39] R. J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (1996) 627–665.

[40] J. Thuburn, Multidimensional flux-limited advection schemes, J. Comp. Phys. 123 (1996) 74–83.

[41] C. Schär, P. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations, J. Comp. Phys. 128 (1996) 101–120.

[42] P. Blossey, D. Durran, Selective monotonicity preservation in scalar advection, J. Comp. Phys. 227 (2008) 5160–5183.

[43] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys 126 (1996) 202–228.

[44] D. Hill, D. Pullin, Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shocks, J. Comp. Phys. 194 (2004) 435–450.

[45] 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.

[46] P. Colella, P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.

[47] W. C. Skamarock, Positive-definite and monotonic limiters for unrestricted-time-step transport schemes., Mon. Wea. Rev. 134 (2006) 2241–2250.

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

[49] P. H. Lauritzen, A stability analysis of finite-volume advection schemes permitting long time steps, Mon. Wea. Rev. 135 (2007) 2658–2673.

[50] F. Xiao, T. Yabe, X. Peng, H. Kobayashi, Conservative and oscillation-less atmospheric transport schemes based on rational functions, J. Geophys. Res. 107 (2002) 4609, doi:10.1029/2001JD001532.