

# CAM-SE–CSLAM: Consistent Coupling of a Conservative Semi-Lagrangian Finite-Volume Method with Spectral Element Dynamics

PETER HJORT LAURITZEN

*National Center for Atmospheric Research,<sup>a</sup> Boulder, Colorado*

MARK A. TAYLOR AND JAMES OVERFELT

*Sandia National Laboratories, Albuquerque, New Mexico*

PAUL A. ULLRICH

*Department of Land, Air, and Water Resources, University of California, Davis, Davis, California*

RAMACHANDRAN D. NAIR, STEVE GOLDHABER, AND RORY KELLY

*National Center for Atmospheric Research,<sup>a</sup> Boulder, Colorado*

(Manuscript received 7 July 2016, in final form 7 October 2016)

## ABSTRACT

An algorithm to consistently couple a conservative semi-Lagrangian finite-volume transport scheme with a spectral element (SE) dynamical core is presented. The semi-Lagrangian finite-volume scheme is the Conservative Semi-Lagrangian Multitracer (CSLAM), and the SE dynamical core is the National Center for Atmospheric Research (NCAR)'s Community Atmosphere Model–Spectral Elements (CAM-SE). The primary motivation for coupling CSLAM with CAM-SE is to accelerate tracer transport for multitracer applications. The coupling algorithm result is an inherently mass-conservative, shape-preserving, and consistent (for a constant mixing ratio, the CSLAM solution reduces to the SE solution for air mass) transport that is efficient and accurate. This is achieved by first deriving formulas for diagnosing SE air mass flux through the CSLAM control volume faces. Thereafter, the upstream Lagrangian CSLAM areas are iteratively perturbed to match the diagnosed SE air mass flux, resulting in an equivalent upstream Lagrangian grid that spans the sphere without gaps or overlaps (without using an expensive search algorithm). This new CSLAM algorithm is not specific to air mass fluxes provided by CAM-SE but applies to any air mass fluxes that satisfy the Lipschitz criterion and for which the Courant number is less than one.

## 1. Introduction

State-of-the-art atmosphere models prognose an increasing number of tracers, which involves solving the continuity equation of each tracer. Tracers in atmosphere models typically include different forms of water (humidity, cloud water, cloud ice, rain, snow, etc.). Models with a comprehensive treatment of aerosols explicitly solve a number of aerosol continuity equations, and

models with comprehensive chemistry prognose many chemical species. For example, NCAR's Community Atmosphere Model (CAM), version 5 (Neale et al. 2012), in its standard configuration solves 30 continuity equations in addition to the continuity equation for air and over 120 continuity equations when run with comprehensive chemistry. Hence, the number of continuity equations in state-of-the-art models easily outnumbers the remaining equations of motion [momentum equations and thermodynamic equation(s)]. Consequently, the computational cost of tracer transport may dominate the overall cost of the model even if tracer transport is supercycled (i.e., run with a larger time step than the dynamics).

Many tracer distributions in the atmosphere have complex spatial structures associated with them. For

---

<sup>a</sup>The National Center for Atmospheric Research is sponsored by the National Science Foundation.

*Corresponding author e-mail:* Peter Hjort Lauritzen, pel@ucar.edu

example, photolysis-driven chemical species exhibit large gradients near the solar terminator as they are destroyed or created by sunlight and moist processes; for example, convection can introduce grid-scale structure in the water variables. It is usually challenging for transport algorithms to accurately represent the transport process for tracer distributions with large gradients and grid-scale structure. It is important that the transport scheme is not a spurious source or sink of mass. For coupling with parameterizations, it is desirable that the transport scheme provides physically realizable solutions; in particular, it should not produce negative values for a positive definite tracer. To avoid overshoots near large gradients, it is desirable that the scheme is monotone (also loosely referred to as shape preserving or nonoscillatory). For the representation of aerosols and chemical species, it is important that the transport scheme does not spuriously perturb relations between tracers (if they are functionally related; e.g., see [Lin and Rood 1996](#); [Lauritzen and Thuburn 2012](#)). For a more complete discussion on desirable properties for transport schemes intended for climate and climate/chemistry applications, see, for example, [Lauritzen et al. \(2011\)](#).

To assess the transport scheme's ability to advect rough distributions and maintain nonlinear correlations between trace species, [Lauritzen et al. \(2012\)](#) created an idealized test case suite for modelers to assess desirable properties for their transport algorithm in an idealized setup. This test case suite used an idealized 2D wind field from [Nair and Lauritzen \(2010\)](#), with a range of initial condition distributions for tracers ranging from smooth to distributions with discontinuous gradients and a wide range of existing and new diagnostics. Results from over a dozen transport algorithms are presented in [Lauritzen et al. \(2014b\)](#).

To improve the scalability of CAM based on the finite-volume dynamical core (CAM-FV; [Lin 2004](#)), the spectral element (SE) dynamical core was imported into CAM from High-Order Methods Modeling Environment (HOMME; [Dennis et al. 2005](#); [Nair et al. 2009](#); [Taylor and Fournier 2010](#)) and is referred to as CAM-SE. CAM-SE is based on a continuous Galerkin finite-element (or spectral element) method ([Taylor et al. 1997](#)) and is discretized using a cubed-sphere tiling of the sphere. Petascale scalability of CAM-SE was demonstrated in [Dennis et al. \(2012\)](#). Also, CAM-SE provided refined mesh functionality in CAM (e.g., [St-Cyr et al. 2008](#); [Zarzycki et al. 2014](#)), improved accuracy of idealized baroclinic wave simulations ([Lauritzen et al. 2010a](#)), conserved total energy to time-truncation errors ([Taylor 2011](#)), and improved global axial angular momentum budgets compared to CAM-FV ([Lauritzen et al. 2014a](#)). That said, the tracer transport component of CAM-SE appeared to be less accurate than CAM-FV

for nonsmooth tracer distributions ([Lauritzen et al. 2014b](#); [Hall et al. 2016](#)). In an idealized terminator test in which two species react according to a reaction coefficient proportional to the solar terminator while being transported by an idealized 2D flow, CAM-SE produces errors larger than CAM-FV ([Lauritzen et al. 2015](#)). The errors are due to the limiter used for tracer transport that prevents oscillatory (and, in particular, negative) solutions for tracers ([Guba et al. 2014](#)). In terms of computational throughput, it was also found that CAM-SE was slower for large tracer counts (at least when run with core counts similar to the non-scalable CAM-FV).

It is the overall purpose of this paper to improve the accuracy and efficiency of tracer transport in CAM-SE. To improve the accuracy and efficiency of multitracer transport, the Conservative Semi-Lagrangian Multitracer (CSLAM; [Lauritzen et al. 2010b](#)) transport scheme was implemented in HOMME ([Erath et al. 2012, 2013](#)). Later, an alternative inherently conservative semi-Lagrangian scheme called spectral element Lagrangian transport (SPELT; [Erath and Nair 2014](#)) was implemented in HOMME, which easily supports mesh-refinement applications and has a simpler search algorithm compared to the original CSLAM (note that the new version of CSLAM presented here does not have a comprehensive search algorithm). Both SPELT and CSLAM are fully 2D finite-volume semi-Lagrangian schemes. Semi-Lagrangian schemes based on dimensional splitting along coordinate lines ([Putman and Lin 2007](#)) or Lagrangian translations of coordinate lines (conservative cascade methods; e.g., [Nair et al. 2002](#); [Zerroukat et al. 2002](#); [Shashkin and Tolstykh 2016](#)) are alternatives to fully 2D methods that are more easily extensible to 3D methods. Cascade schemes are computationally efficient and designed for structured orthogonal grid systems. It is unclear how to extend the cascade methods to nonorthogonal grids—in particular, the gnomonic cubed-sphere grid used here—as the grid lines are discontinuous near the cubed-sphere edges and corners. All of these schemes are stable for long time steps, inherently mass conservative, and shape preserving; some of them preserve linear relations between tracers even with a shape-preserving limiter (e.g., CSLAM) and are efficient for large tracer counts as the upstream Lagrangian grids can be reused for each additional tracer. Specific to this study, we note that in 2D idealized test cases, the CSLAM scheme is more accurate than CAM-SE in terms of the diagnostics used in the test case suite presented in [Lauritzen et al. \(2012\)](#).

For 3D applications in CAM-SE, the winds for CSLAM are provided by CAM-SE, which creates an inconsistency between the continuity equation for air (solved with the SE method) and tracers (solved with

CSLAM). In other words, if a tracer is uniformly one, then the solution for tracer mass, computed with CSLAM, does not reduce to the CAM-SE continuity equation for air. Inconsistency can lead to loss of tracer mass conservation and/or unphysical solutions (non-monotone). This problem of the discretization of the transport equation not being consistent with the air mass equation has been known for some time (Rood 1987; Lin and Rood 1996; Gross et al. 2002). Discretizations that are consistent are also referred to in the literature as free-stream preserving, constancy preserving, or compatible. Solutions have been proposed for shallow-water applications in which correction terms in the velocity fields have been introduced to enforce consistency (e.g., Deleersnijder and Lebon 2001) or using a corrective mass flux in the transport scheme (e.g., Dawson 1999).

In this paper, we present a CSLAM-based algorithm in which CSLAM is made consistent with the SE continuity equation for air so that the CSLAM solution to the tracer continuity equations is mass conservative, monotone, and consistent. This is achieved by diagnosing CAM-SE air mass fluxes through CSLAM control volume edges and using that information to construct CSLAM control volumes for which the air mass flux implied by CSLAM matches CAM-SE fluxes to round off. Consistency is effectively achieved by finding an upstream departure grid for which the CSLAM solution for air mass matches the CAM-SE solution for air mass to round off. Rather than manipulating velocity components or adjusting fluxes a posteriori, the consistency is enforced in Lagrangian space. Note that the algorithm for enforcing consistency in CSLAM is general. Any air mass fluxes (so not necessarily CAM-SE diagnosed fluxes) that satisfy the Lipschitz criterion (also known as the deformational Courant number by which the upstream trajectories do not cross intersect in a single time step; Pudykiewicz et al. 1985; Kuo and Williams 1990) and for which the Courant number is less than one can be used.

The paper is organized as follows: The continuity equation for CSLAM and SE are introduced in section 2 as well as the consistent coupling problem formulation. In section 3, the consistent SE-CSLAM algorithm is presented, and in section 4, we present results from idealized baroclinic wave simulations with inert and reactive tracers in order to assess accuracy and computational efficiency. A summary is provided in section 5.

## 2. Methods

### a. Finite-volume continuity equation

Assume a floating Lagrangian vertical coordinate that initially coincides with the hybrid-sigma coordinates

(Lin 2004; Starr 1945). The air mass in each layer is  $\Delta p \Delta A_{ij}$ , where  $\Delta p$  is the pressure-level thickness (for notational simplicity, the vertical index is dropped) and  $\Delta A_{ij}$  is the area of the regular (Eulerian) grid cell  $A_{ij}$ . Similar variable definitions for the tracer mass  $\Delta p Q \Delta A_{ij}$ , where  $Q$  is mixing ratio. The gnomonic coordinate locations of the vertices of  $A_{ij}$  associated with the cubed-sphere geometry (Rančić et al. 1996) are  $(x_i, y_j)$ ,  $(x_{i+1}, y_j)$ ,  $(x_{i+1}, y_{j+1})$ , and  $(x_i, y_{j+1})$ . After a number of time steps in which the vertical coordinate is floating, all variables are remapped back to an Eulerian reference vertical coordinate (Lin 2004).

Consider the time-discretized integral form of the continuity equation for air and tracer mass discretized in time using floating Lagrangian vertical coordinates:

$$\int_{A_{ij}} \psi_{ij}^{\tau+1} dA = \int_{a_{ij}} \psi_{ij}^{\tau} dA, \quad \psi = \Delta p, \Delta p Q, \quad (1)$$

(see Fig. 1) where  $\tau$  is the time-level index, and  $a_{ij}$  is the upstream Lagrangian area that, after one time step  $\Delta t$ , ends up at  $A_{ij}$ .

The Lagrangian form of the continuity equation (1) can also be written in flux form (Harris et al. 2011):

$$\int_{A_{ij}} \psi_{ij}^{\tau+1} dA = \int_{A_{ij}} \psi_{ij}^{\tau} dA + \sum_{\varepsilon=1}^4 \mathcal{H}_{ij}^{(\varepsilon)}, \quad \psi = \Delta p, \Delta p Q, \quad (2)$$

where the mass fluxes through the four edges of a CSLAM cell are given by

$$\mathcal{H}_{ij}^{(\varepsilon)} = \text{sgn}_{ij}^{(\varepsilon)} \int_{a_{ij}^{(\varepsilon)}} \psi_{ij}^{\tau} dA, \quad \varepsilon = 1, \dots, 4, \quad (3)$$

and  $\text{sgn}_{ij}^{(\varepsilon)}$  is  $-1$  for outflow and  $+1$  for inflow. Note that the union of all the flux areas  $a_{ij}^{(\varepsilon)}$  with appropriate signs yields the Lagrangian upstream area:

$$\int_{A_{ij}} \psi_{ij} dA + \sum_{\varepsilon=1}^4 \int_{a_{ij}^{(\varepsilon)}} \text{sgn}_{ij}^{(\varepsilon)} \psi_{ij} dA = \int_{a_{ij}} \psi_{ij} dA. \quad (4)$$

Hence, the two forms of the continuity equation are, as expected, equivalent in their continuous forms.

In the CSLAM discretization, the vertices of  $A_{ij}$  are traced upstream and connected with straight lines to form  $a_{ij}$ . To make the scheme third order, a subgrid-scale distribution of  $\psi$  is computed (fully 2D degree-two Lagrange polynomial; Ullrich et al. 2009) with mass conservation as a constraint; that is, the polynomial integrated over  $A_{ij}$  yields the gridcell average value  $\bar{\psi}_{ij}$ . The integral of the polynomial over the overlap areas between  $a_{ij}$  and the regular grid cells is computed

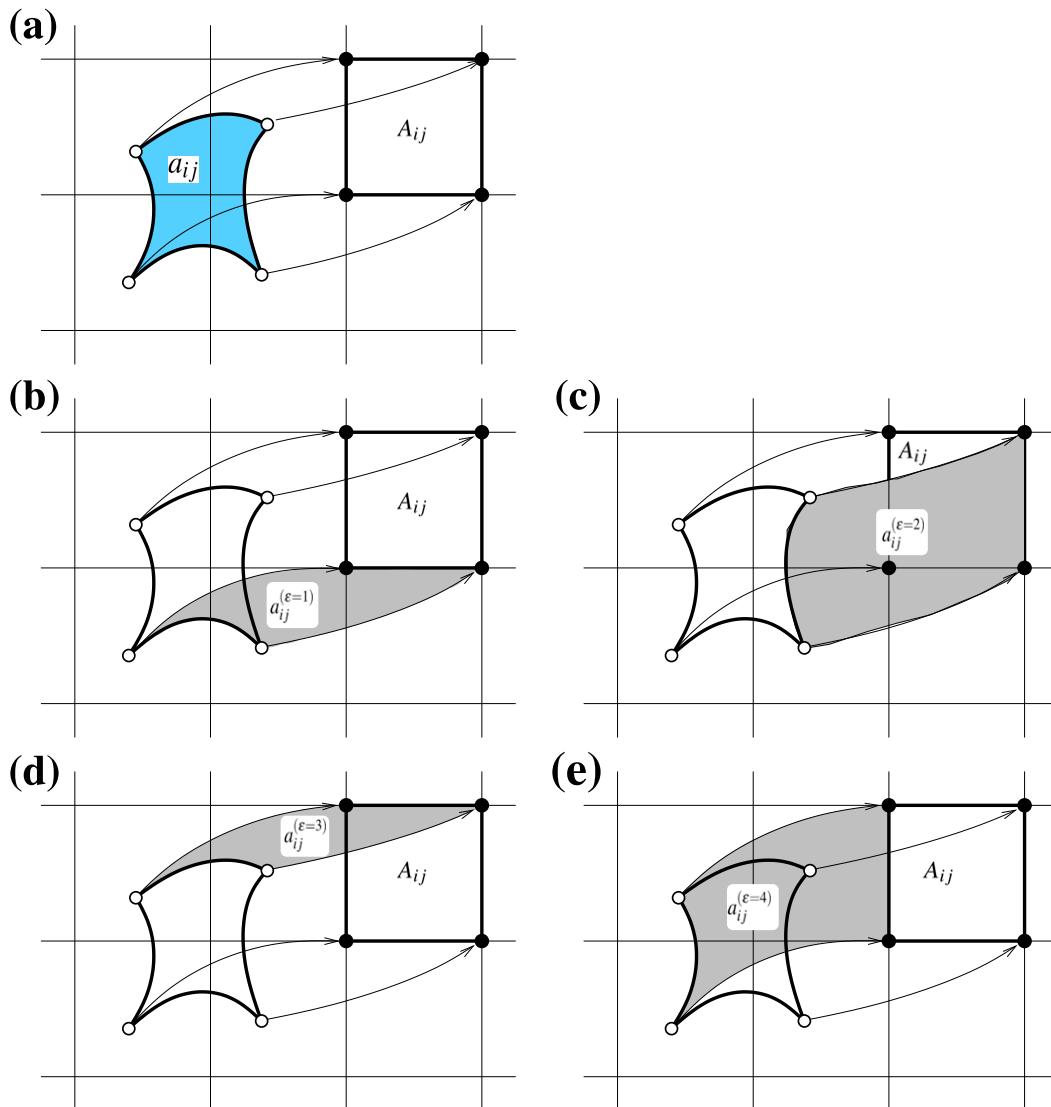


FIG. 1. A graphical illustration of (a) the finite-volume Lagrangian discretization of the continuity equation and (b)–(e) the Eulerian finite-volume flux-form discretization. In (a), the upstream Lagrangian area  $a_{ij}$  (light blue shading) ends up at the regular Eulerian grid cell  $A_{ij}$  after one time step. The filled circles mark the vertices of the Eulerian cell  $A_k$ , and the unfilled circles are the corresponding upstream vertices or departure points. In (b)–(e) the swept flux areas  $a_{ij}^{(\epsilon)}$  (gray shaded areas) are shown, where  $\epsilon = 1, 2, 3, 4$ , respectively.

via line integrals. Since the scheme is fully two-dimensional, it extends to the cubed sphere quite easily in the sense that it can accurately represent the deformation of the Lagrangian cells when they span different panels and hence different projections [see, e.g., Fig. 6 in Nair and Lauritzen (2010)]. For more details on the CSLAM scheme, see Lauritzen et al. (2010b). For the coupling between tracer mass and air mass, when both are solved with the CSLAM scheme, see appendix B in Nair and Lauritzen (2010). It may be shown that even in their discretized forms, the flux-form version of CSLAM based on (2) is equivalent to (1).

Monotone solutions for  $Q$  are obtained by applying a limiter/filter to the subgrid-scale reconstruction function for  $Q$  so that its extrema are within the range of the surrounding gridcell average values of  $Q$  (Barth and Jespersen 1989).

#### b. SE continuity equation

The SE continuity equation for air mass is based on the advection–diffusion equation

$$\frac{\partial \Delta p}{\partial t} = -\nabla \cdot (\mathbf{u} \Delta p) + \nu \nabla^\kappa \Delta p, \quad \kappa = 2, 4, \dots, \quad (5)$$

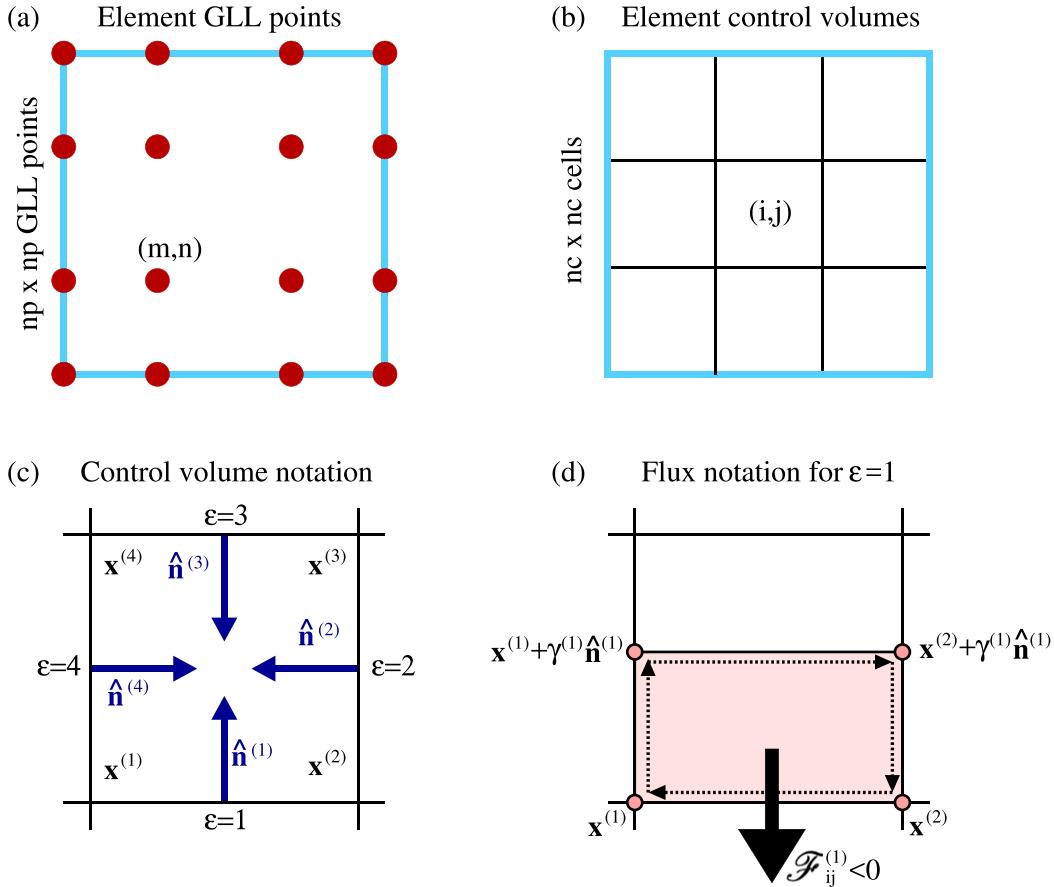


FIG. 2. (a) The  $np \times np$  GLL points (filled red circles) in an element (light blue boundary). (b) The  $nc \times nc$  CSLAM control volumes in an element. (c) Notation for the  $ij$ th CSLAM cell. Sides  $\varepsilon$ ,  $\varepsilon = 1, \dots, 4$ , are numbered counterclockwise around the cell. The vertices are located at gnomonic coordinates  $\mathbf{x}^{(\varepsilon)}$ ,  $\varepsilon = 1, \dots, 4$ , with counterclockwise orientation. Each cell side has an associated unit vector  $\hat{\mathbf{n}}^{(\varepsilon)}$  (dark blue arrows) pointing inwards. (d) Notation for the perpendicular swept flux area for side  $\varepsilon = 1$ . The flux area has vertices  $\mathbf{x}^{(\varepsilon)}$ ,  $\mathbf{x}^{(\varepsilon)} + \gamma^{(\varepsilon)} \hat{\mathbf{n}}^{(\varepsilon)}$ ,  $\mathbf{x}^{(\varepsilon+1)} + \gamma^{(\varepsilon)} \hat{\mathbf{n}}^{(\varepsilon)}$ , and  $\mathbf{x}^{(\varepsilon+1)}$  (marked with filled purple circles). The large black arrow points in the direction of the SE flux  $\mathcal{F}_{ij}^{(\varepsilon)}$  (outflow through side one). The dashed arrows show the orientation of the line integrals used to compute the perpendicular swept flux.

where  $\mathbf{u}$  is the velocity vector,  $\kappa$  is the order of the viscosity, and  $\nu$  is the (hyper)viscosity coefficient. For simplicity, the derivations below assume  $\kappa = 2$ . The SE method uses a weak formulation based on globally continuous test functions and integration over the entire computational domain. A suitable choice of Lagrange basis functions and a Gauss–Lobatto–Legendre (GLL) quadrature inner product leads to a diagonal mass matrix, and the SE method can then be formulated in terms of operations local to each element, followed by the direct stiffness summation (DSS) operation to maintain continuity at the element edges. For more details, see Deville et al. (2002).

For the SE method, we define the inner product via

$$\langle p, q \rangle = \sum_{m=1}^{np} \sum_{n=1}^{np} w_m w_n J(x_m, y_n) p(x_m, y_n) q(x_m, y_n), \quad (6)$$

where  $J$  is the Jacobian of the map from the quadrilateral spherical element to the reference element  $[-1, 1]^2$ ;  $\{x_m, y_n\}$  are the  $np \times np$  GLL quadrature points in each element (see Fig. 2a);  $p(x, y)$  and  $q(x, y)$  are polynomials; and  $w_m$  and  $w_n$  are the GLL quadrature weights [for more information on the HOMME grid system, see Nair et al. (2009)]. Note that the GLL quadrature point locations use subscript  $m$  and  $n$ , whereas the CSLAM control volume edges use subscript  $i$  and  $j$ .

The SE advection step may be written as

$$\left\langle \phi, \frac{\Delta p^{\tau+1} - \Delta p^\tau}{\Delta t} \right\rangle = \langle \phi, \nabla \cdot \mathbf{F} \rangle + \langle \nabla \phi, \mathbf{G} \rangle + \langle \phi, \mathbf{D} \rangle, \quad (7)$$

where  $\phi$  is a test function,  $\mathbf{F} = \mathbf{v} \Delta p$  and  $\mathbf{G} = \nu \nabla \Delta p$ , and  $\mathbf{D}$  is the nodal flux corresponding to the DSS operation (i.e., summation at element edges combined with

multiplication of the inverse mass matrix). Note that if  $\phi = 1$ , then the right-hand side of (7) is the mass change within each element.

It may be shown that there exists a test function  $\phi^{(ij)}$  that gives the change of mass over the  $ij$  th CSLAM control volume ( $A_{ij}$ ):

$$\frac{\langle \phi^{(ij)}, \Delta p^{\tau+1} \rangle - \langle \phi^{(ij)}, \Delta p^{\tau} \rangle}{\Delta t} = \langle \phi^{(ij)}, \nabla \cdot \mathbf{F} \rangle + \langle \nabla \phi^{(ij)}, \mathbf{G} \rangle + \langle \phi^{(ij)}, \mathbf{D} \rangle. \quad (8)$$

Moreover, the change of mass in each subcell ( $A_{ij}$ ) given in (8) is given by a sum of edge mass fluxes  $\mathcal{F}$ :

$$\langle \phi^{(ij)}, \Delta p^{\tau+1} \rangle - \langle \phi^{(ij)}, \Delta p^{\tau} \rangle = \sum_{\varepsilon} \mathcal{F}_{ij}^{(\varepsilon)}, \quad \varepsilon = 1, 2, 3, 4, \quad (9)$$

and for each face, there is a flux contribution from  $\mathbf{F}$ ,  $\mathbf{G}$ , and  $\mathbf{D}$ . In other words, the SE air mass flux can be diagnosed and is derived in detail in the [appendix](#). This SE air mass flux is essential for the coupling algorithm described in this paper. First, let us formulate the coupling problem.

### c. Problem formulation

The challenge is to couple the CSLAM continuity equation for tracers with the SE continuity equation for air without violating the following:

- Tracer mass conservation
- Consistency (aka compatibility, free-stream preservation, constancy preservation)
- Tracer shape preservation
- Linear correlations between tracers (e.g., [Lin and Rood 1996](#); [Lauritzen and Thuburn 2012](#))

Here, consistency refers to the property that the continuity equation for tracer mass reduces to the equation for air mass for  $Q = 1$  in discretized space. This is trivial if the tracer transport equation is solved with the same numerical method as the continuity equation for air. The consistency is violated if, for example, the continuity equation for tracers is forced with reanalysis data that prescribe winds and surface pressure or, as is the case here, if different numerical methods (on different grids) are used for the continuity equation for tracers and air mass.

If one can find an upstream Lagrangian grid  $a_{ij}$  that satisfies the following constraints, then the requirements listed above are automatically satisfied.

Constraint 1: the CSLAM fluxes must match the SE fluxes:

$$\forall ij \quad \text{and} \quad \varepsilon = 1, \dots, 4: \mathcal{H}_{ij}^{(\varepsilon)} = \mathcal{F}_{ij}^{(\varepsilon)}. \quad (10)$$

Constraint 2: the upstream Lagrangian cells  $a_{ij}$  span the sphere without gaps and/or overlaps:

$$\cup_{ij} a_{ij} = \Omega \quad \text{and} \quad \cap_{ij} a_{ij} = \emptyset, \quad (11)$$

where  $\Omega$  is the surface of the sphere. Note that the upstream grid constructed from using a standard semi-Lagrangian trajectory algorithm does not necessarily provide CSLAM fluxes that satisfy constraint 1 and therefore, the air mass consistency is violated. If the individual air mass fluxes from CSLAM are overwritten with SE fluxes, the tracer shape preservation is violated.

The problem of developing swept areas to match the SE fluxes may initially seem intractable, as the two methods appear very different (SE is often thought of as a pointwise method in contrast with the volumetric approach of finite-volume methods). However, local construction of the swept areas is made possible by the fact that, on the level of individual spectral elements, there is a well-defined notion of fluxes between adjacent elements. It is exactly this correspondence that allows SE methods to maintain global mass conservation on a local level.

## 3. Consistent CSLAM flux algorithm

The algorithm has been designed to avoid (as much as possible) conditional statements in the code for control volumes at the cubed-sphere edges. Hence, there is a substantial amount of notation that needs to be introduced. The basic algorithm is explained in [Fig. 3](#).

### a. Departure point algorithm

In this section, subscript  $ij$  has been dropped for notational simplicity. The departure points are defined in terms of perpendicular swept areas,  $\delta^{\perp} a^{(\varepsilon)}$  ([Fig. 2d](#)), for which the swept CSLAM mass flux  $\mathcal{H}$  matches the SE mass flux  $\mathcal{F}$  to round off. The mathematical formulation is as follows: for CSLAM control volume ( $i, j$ ), find perpendicular swept area for side  $\varepsilon = 1, 2, 3, 4$ , so that

$$\mathcal{H}^{(\varepsilon)} = \int_{\delta^{\perp} a^{(\varepsilon)}} \Delta p^{\tau} dA, \quad (12)$$

where  $\delta^{\perp} a^{(\varepsilon)}$  is the perpendicular swept area (a rectangle) with vertices  $\mathbf{x}^{(\varepsilon)}$ ,  $\mathbf{x}^{(\varepsilon)} + \gamma^{(\varepsilon)} \hat{\mathbf{n}}^{(\varepsilon)}$ ,  $\mathbf{x}^{(\varepsilon+1)} + \gamma^{(\varepsilon)} \hat{\mathbf{n}}^{(\varepsilon)}$ , and  $\mathbf{x}^{(\varepsilon+1)}$ , where  $\gamma^{(\varepsilon)}$  is the displacement parameter ([Fig. 2d](#)), and  $\hat{\mathbf{n}}^{(\varepsilon)}$  is the unit vector perpendicular to side  $\varepsilon$  ([Fig. 2c](#)). The integral on the right-hand side of (12) is converted to line integrals using the divergence theorem, and the line integrals are computed using Gauss quadrature [for details, see [Lauritzen et al. \(2010b\)](#)]. Note that the potentials for the line integrals are chosen

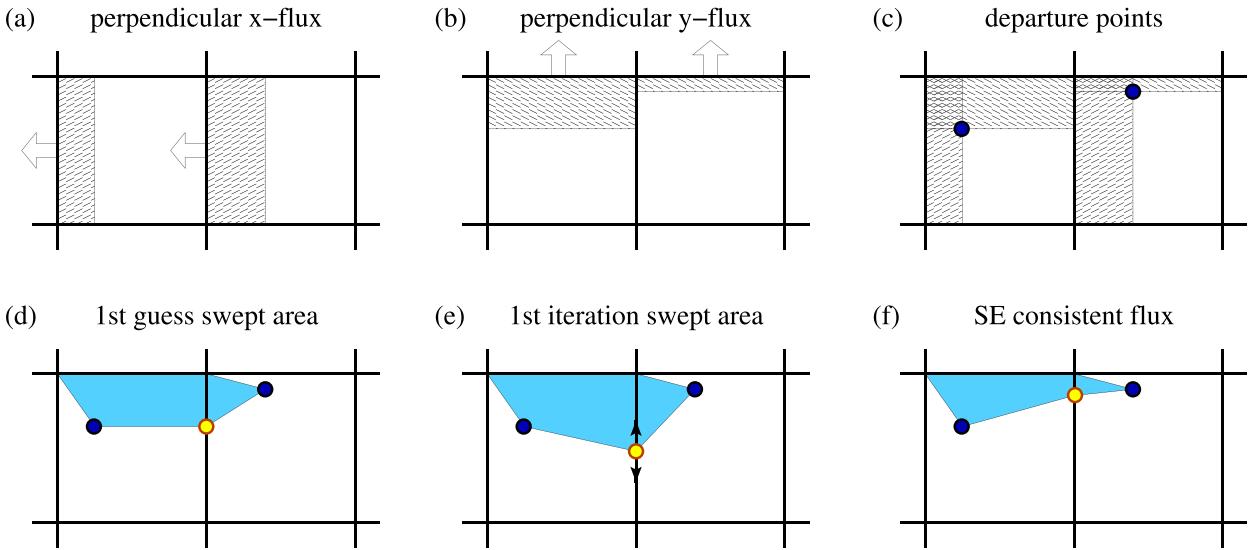


FIG. 3. Step-by-step example of the consistent SE-CSLAM algorithm for face 3. (a),(b) Compute the CSLAM flux perpendicular to each face by iteration. (c) The intersection between the upstream perpendicular flux faces defines the departure points for CSLAM (dark blue filled circles). (d) A degree of freedom for the swept area is introduced at the intersection between the perpendicular swept y-flux area and the coordinate line (red circle filled with yellow). (e),(f) By iteration, the degree of freedom is moved up and down along the coordinate axis until the flux area exactly matches the SE flux.

such that lines with constant  $x$  coordinates have zero line integrals.

The swept area  $\delta^\perp a^{(\varepsilon)}$  is defined in terms of one independent variable  $\gamma^{(\varepsilon)}$ , and the integral on the right-hand side of (12) is monotonically increasing with increasing  $\gamma^{(\varepsilon)}$ . This constitutes a well-posed iteration problem that is solved with the Newton iteration algorithm. For notational simplicity, we have dropped the side index  $\varepsilon$  in the pseudoalgorithm below.

Let  $\Gamma[\gamma(\vartheta)]$  be the difference between the perpendicular CSLAM swept flux (which is a function of  $\gamma$ ) and the SE flux  $\Gamma(\gamma)$ :

$$\Gamma[\gamma(\vartheta)] = \int_{\delta^\perp a[\gamma(\vartheta)]} \Delta p^\tau dA - \mathcal{F}, \quad (13)$$

where  $\vartheta$  is the iteration index. The iteration procedure is as follows:

- 1) Set  $\gamma(1) = 0, \Gamma[\gamma(1)] = 0$ .
- 2) Set  $\gamma(2) = 0.5$  and compute  $\Gamma[\gamma(2)]$ .
- 3) Compute

$$d\gamma = \frac{[\gamma(2) - \gamma(1)]\Gamma[\gamma(2)]}{\Gamma[\gamma(2)] - \Gamma[\gamma(1)]}.$$

- 4) Set  $\gamma(3) = \gamma(2) - d\gamma$  and compute  $\Gamma[\gamma(3)]$ .
- 5) If  $|\Gamma[\gamma(3)]|$  is less than a given tolerance (e.g.,  $10^{-14}$ ), the iteration has converged; else set  $\gamma(1) = \gamma(2)$  and  $\gamma(2) = \gamma(3)$  and repeat steps 3–5.
- 6) Done: set  $\gamma = \gamma(3)$ .

Note that when computing the line integrals for  $\Gamma(\gamma)$ , line segments that are  $x$  isolines are zero and only the line segments that are a function of  $\gamma$  need to be recomputed during each Newton iteration. Hence, for sides  $\varepsilon = 2, 4$ , there are only two line segments that need to be recomputed for each iteration, and for sides  $\varepsilon = 1, 3$ , there is one. The iteration algorithm is therefore efficient.

Obviously, the fluxes are equal with opposite sign for sides shared between control volumes. Hence, the displacements,  $\gamma^{(\varepsilon)}$ , are only computed for outflow ( $\mathcal{F}^{(\varepsilon)} < 0$ ) and set to zero for inflow. For sides located in nonexistent cells in physical space [in other words, the halo cells located at the cubed-sphere corners; Fig. 5 in Erath et al. (2012)], the displacements are set to zero. The departure point corresponding to arrival point  $x_{ij}^{(\varepsilon)}$  is given by

$$x_{ij}^{(\varepsilon)*} = x_{ij}^{(\varepsilon)} + \sum_{i'=i-1}^i \sum_{j'=j-1}^j \gamma_{i'j'}^{(\varepsilon)} \hat{\mathbf{n}}_{i'j'}^{(\varepsilon)} + \gamma_{ij}^{(\varepsilon)} \hat{\mathbf{n}}_{ij}^{(\varepsilon)}. \quad (14)$$

To make sure the iterative algorithm for the final swept area is well conditioned and simple, we set the displacements to zero for circular flow and “degenerate” flow conditions (described in Fig. 4). If this is not done, the iterative algorithm for the final swept areas (described in the next section) becomes more complicated and expensive. If we assume that the SE fluxes satisfy a flow-deformation criterion similar to the Lipschitz criterion (also referred to as deformational Courant number), the circular and degenerate flow cases are limited in terms of the size of the swept fluxes.

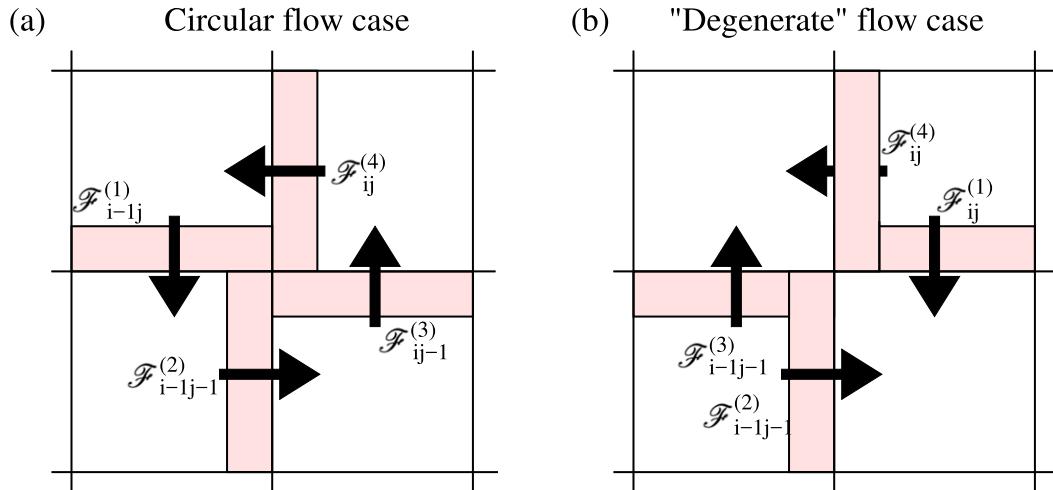


FIG. 4. Perpendicular swept flux situations in which the displacement  $\gamma$  for the center vertex ( $\mathbf{x}_{ij}^{(1)}$ ,  $\mathbf{x}_{ij-1}^{(4)}$ ,  $\mathbf{x}_{i-1j}^{(2)}$ , and  $\mathbf{x}_{i-1j-1}^{(3)}$ ) is set to zero. (a) For flow situations in which the perpendicular swept fluxes follow a pattern of circular flow, the displacement is set to zero. Similarly for circular flow in the reverse direction. (b) When the perpendicular fluxes are negative (outflow) for sides  $\varepsilon = 2, 3$  in cell  $(i - 1, j - 1)$  as well as sides  $\varepsilon = 1, 4$  in cell  $(i, j)$ , then the displacement is set to zero. Similarly when  $\mathcal{F}_{i-1j}^{(\varepsilon)} < 0$  for  $\varepsilon = 1, 2$  and  $\mathcal{F}_{ij-1}^{(\varepsilon)} < 0$  for  $\varepsilon = 3, 4$ .

*b. Final swept flux area*

The algorithm for calculating the final swept areas is the same for each side with appropriate index change. Hence, we describe the algorithm only for side 1,  $\varepsilon = 1$ , in cell  $(i, j)$ . Define displacement variables  $d_k$  (defined as the sum of the displacements on either side of a cell wall) and associated unit/basis vectors  $\mathbf{b}_k$  (defined in Fig. 5). The definitions of  $d$  and  $\mathbf{b}$  for sides 2, 3, and 4 are obtained by rotating the figure  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ , respectively, about the center for cell  $(i, j)$ . The subscript  $k$  is chosen quite arbitrarily. The purpose of introducing this notation is so that after the association of  $d_k$  and  $\mathbf{b}_k$  with the relevant displacements and face unit vectors, the algorithm is the same for all faces. The notation used here closely follows the FORTRAN source code.

For the categorization of the flow cases (Fig. 6), the logical variables  $L_1, L_2, L_4$ , and  $L_5$  are defined as follows:

- $L_1 = d_6 < 0 \wedge d_7 > 0$ , i.e.  $\mathbf{x}_{ij*}^{(1)}$  in cell  $(i - 1, j - 1)$ ,
- $L_2 = d_3 < 0 \wedge d_6 \geq 0$ , i.e.  $\mathbf{x}_{ij*}^{(1)}$  in cell  $(i - 1, j)$ ,
- $L_4 = d_2 < 0 \wedge d_4 \geq 0$ , i.e.  $\mathbf{x}_{ij*}^{(2)}$  in cell  $(i + 1, j)$ ,
- $L_5 = d_4 < 0 \wedge d_5 > 0$ , i.e.  $\mathbf{x}_{ij*}^{(2)}$  in cell  $(i + 1, j - 1)$ ,

and similarly (by appropriate rotation) for sides 2, 3, and 4. The variable  $L_r$  is true if a departure point is located in area  $r$  (see Fig. 5b for the area  $r$  definition). The challenges in the categorization are the pathological cases

where some displacements are zero. The categorization into flow cases depicted in Fig. 6 is given by

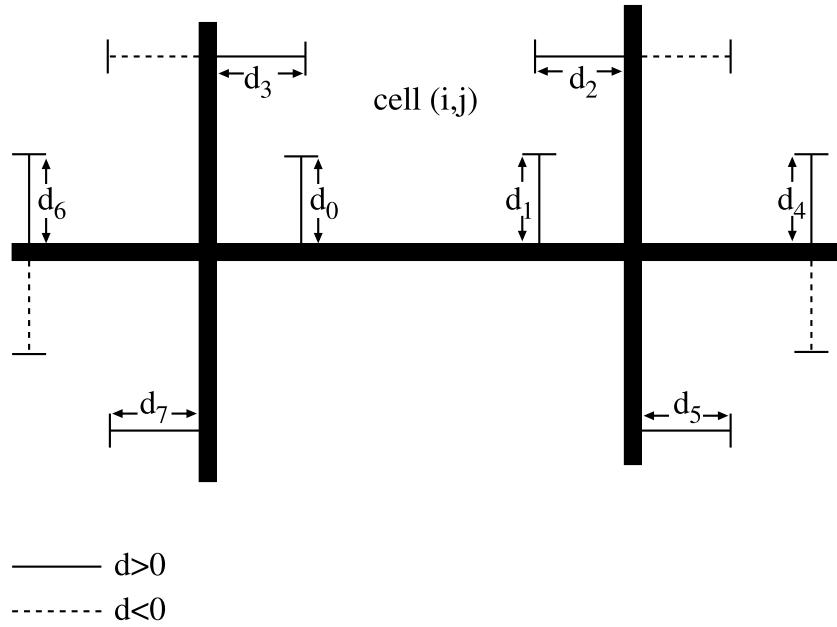
$$\text{case} = \begin{cases} 0, & (\neg L_1) \wedge (\neg L_2) \wedge (\neg L_4) \wedge (\neg L_5), \\ 1, & L_2 \wedge L_4, \\ 2, & L_2 \wedge (\neg L_4) \wedge (\neg L_5), \\ 3, & L_4 \wedge (\neg L_2) \wedge (\neg L_1), \\ 4, & L_1 \wedge (\neg L_4) \wedge (\neg L_5), \\ 5, & L_5 \wedge (\neg L_1) \wedge (\neg L_2), \\ 6, & L_1 \wedge L_4 \wedge (\neg L_5), \\ 7, & L_5 \wedge L_2 \wedge (\neg L_1) \\ 8, & L_1 \wedge L_5. \end{cases} \quad (15)$$

Having defined the flow cases, a well-posed and simple iteration algorithm is constructed to make the CSLAM swept fluxes identical to the prescribed SE fluxes  $\mathcal{F}$ . For that, a swept area that decreases and increases monotonically as a function of one parameter,  $\gamma$ , is defined. We then iteratively compute  $\gamma$  by minimizing the difference between the swept CSLAM flux and SE flux:

$$\Gamma(\gamma) = \left( \sum_{k=1}^5 \int_{\delta a_k} \Delta p_k^r \right) - \mathcal{F}, \quad (16)$$

where  $\delta a_k$  is the overlap area between the swept flux area and Eulerian cell  $k$ . For side 1, the Eulerian areas are  $k = \{(i - 1, j - 1), (i - 1, j), (i, j), (i + 1, j), (i + 1, j - 1)\}$ . A well-posed iteration problem is obtained by defining the CSLAM swept areas using the following criteria:

(a) Definition of displacement variable  $d$  for side 1 (similarly for other sides by rotation)



(b) Basis/unit vector  $\mathbf{b}$  and vertex  $\mathbf{v}$  notation for side 1 (similarly for other sides by rotation)

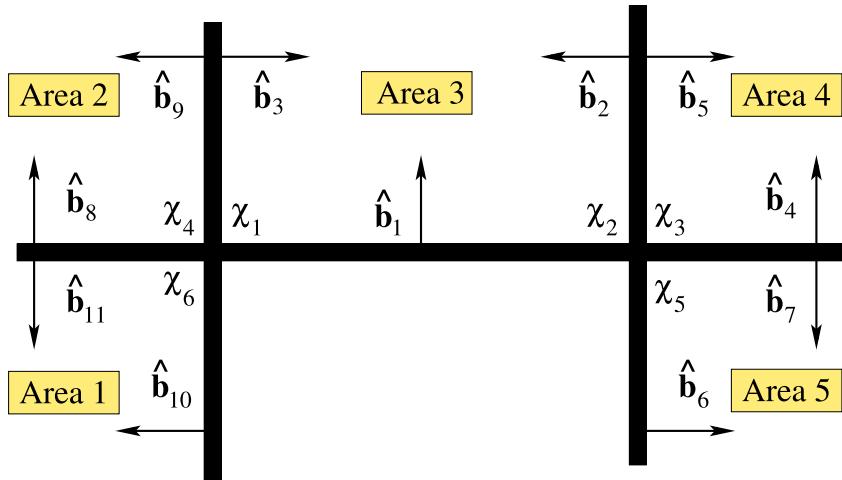


FIG. 5. (a) Definition of displacement variable  $d$  for side 1 in cell  $(i, j)$ :  $d_0 = \gamma_{ij}^{(1)}$ ,  $d_1 = \gamma_{ij}^{(1)}$ ,  $d_2 = \gamma_{ij}^{(2)} - \gamma_{i+1j}^{(4)}$ ,  $d_3 = \gamma_{ij}^{(4)} - \gamma_{i-1j}^{(2)}$ ,  $d_4 = \gamma_{i+1j-1}^{(1)} - \gamma_{i+1j-1}^{(3)}$ ,  $d_5 = \gamma_{i+1j-1}^{(4)} - \gamma_{ij-1}^{(2)}$ ,  $d_6 = \gamma_{i-1j}^{(1)} - \gamma_{i-1j-1}^{(3)}$ , and  $d_7 = \gamma_{i-1j-1}^{(4)} - \gamma_{ij-1}^{(2)}$ . The reason for defining  $d_0$  and  $d_1$  is that if there is a circular or degenerate flow case at vertex  $\mathbf{x}_{ij}^{(1)}$  or  $\mathbf{x}_{i+1j}^{(1)}$ , then the displacements  $d_0$  and  $d_1$  are no longer identical. (b) The associated basis/unit vectors for each side:  $\hat{\mathbf{b}}_0 = \hat{\mathbf{b}}_1 = \hat{\mathbf{n}}_{ij}^{(1)}$ ,  $\hat{\mathbf{b}}_2 = \hat{\mathbf{n}}_{ij}^{(2)}$ ,  $\hat{\mathbf{b}}_3 = \hat{\mathbf{n}}_{ij}^{(4)}$ ,  $\hat{\mathbf{b}}_4 = \hat{\mathbf{n}}_{i+1j}^{(1)}$ ,  $\hat{\mathbf{b}}_5 = \hat{\mathbf{n}}_{i+1j}^{(4)}$ ,  $\hat{\mathbf{b}}_6 = \hat{\mathbf{n}}_{i+1j-1}^{(4)}$ ,  $\hat{\mathbf{b}}_7 = \hat{\mathbf{n}}_{i+1j-1}^{(3)}$ ,  $\hat{\mathbf{b}}_8 = \hat{\mathbf{n}}_{i-1j}^{(1)}$ ,  $\hat{\mathbf{b}}_9 = \hat{\mathbf{n}}_{i-1j}^{(2)}$ ,  $\hat{\mathbf{b}}_{10} = \hat{\mathbf{n}}_{i-1j-1}^{(2)}$ , and  $\hat{\mathbf{b}}_{11} = \hat{\mathbf{n}}_{i-1j-1}^{(3)}$ . The vertex notation is as follows:  $\chi_1 = \mathbf{x}_{ij}^{(1)}$ ,  $\chi_2 = \mathbf{x}_{ij}^{(2)}$ ,  $\chi_3 = \mathbf{x}_{i+1j}^{(1)}$ ,  $\chi_4 = \mathbf{x}_{i-1j}^{(2)}$ ,  $\chi_5 = \mathbf{x}_{i+1j-1}^{(1)}$ , and  $\chi_6 = \mathbf{x}_{i-1j-1}^{(5)}$ . For cells in the interior of a panel,  $\chi_1 = \chi_4 = \chi_6$  and  $\chi_2 = \chi_3 = \chi_5$ ; however, for cells at cubed-sphere corners and sides, this is no longer the case. In yellow boxes, the area index is written; that is, areas 1, 2, 3, 4, and 5 correspond to cells  $(i-1, j-1)$ ,  $(i-1, j)$ ,  $(i, j)$ ,  $(i+1, j)$ , and  $(i+1, j-1)$ , respectively, for side  $\varepsilon = 1$ .

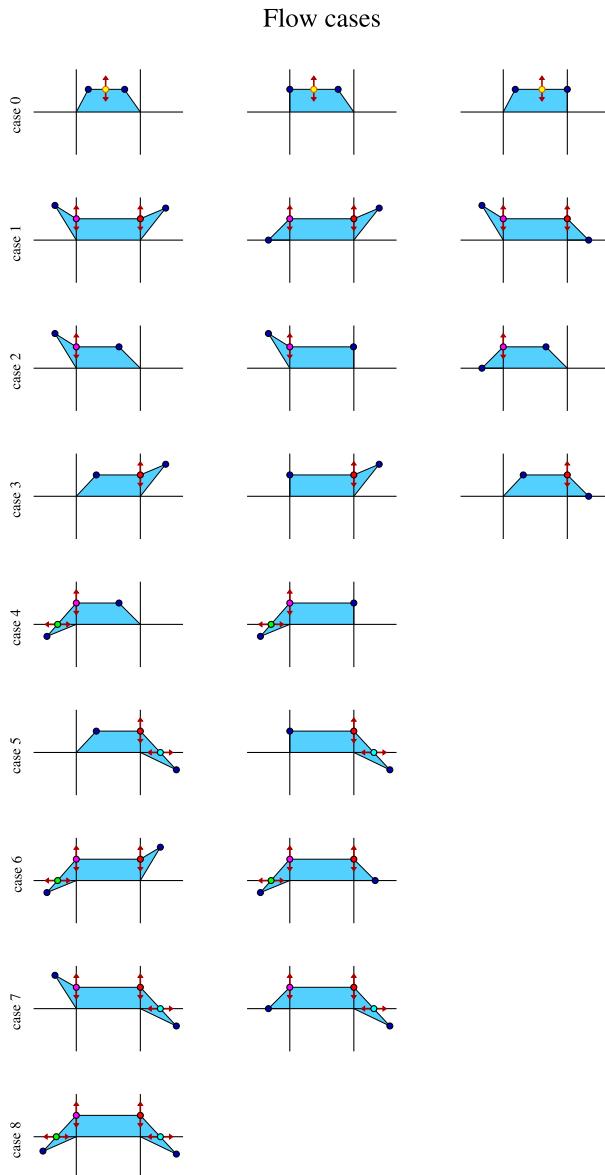


FIG. 6. A depiction of the categorization of possible flow cases defined in (15). The light blue shaded area is the first-guess swept flux. The dark blue filled circles are the departure points (that remain fixed) and the yellow filled circles are moved along the coordinate lines (according to red arrows) until the swept air mass flux matches the SE mass flux to round off. The middle and right columns showcase where one or two departure points coincide with a grid line.

- If the departure point  $\mathbf{x}_{ij^*}^{(1)}$  is located in area 1 or 2 (cases 1, 2, 4, 6, 7, and 8), then the first-guess crossing ( $\gamma = 1$ ) with side 4 is given by  $\chi_1 + \gamma d_0 \hat{\mathbf{b}}^{(1)}$  for the line integrals in area 3 and  $\chi_4 + \gamma d_6 \hat{\mathbf{b}}^{(8)}$  in area 2 (see magenta filled circles in Fig. 6). The coordinate crossings are the same if side 4 is not coinciding with a cubed-sphere panel side. Similarly, if the departure point  $\mathbf{x}_{ij^*}^{(2)}$  is located in area 3 or 4 (cases 1, 3, 5, 7, and

8), then the first-guess crossing ( $\gamma = 1$ ) with side 2 is given by  $\chi_2 + \gamma d_1 \hat{\mathbf{b}}^{(1)}$  for the line integrals in area 3 and  $\chi_3 + \gamma d_4 \hat{\mathbf{b}}^{(4)}$  for the line integrals in area 4 (see red filled circles in Fig. 6). Note that if  $\gamma$  is decreased, the swept area decreases.

- If the departure point  $\mathbf{x}_{ij^*}^{(1)}$  is located in area 1, the first-guess crossing with side 1 of cell  $(i - 1, j)$  is computed as the crossing between the line connecting the departure point  $\mathbf{x}_{ij^*}^{(1)}$  (blue filled circle in area 1) and the crossing computed above  $\chi_1 + \gamma d_0 \hat{\mathbf{b}}^{(1)}$  (magenta filled circle) given by

$$\chi_4 - \gamma x_{\text{cross}}^{(\text{left})} \hat{\mathbf{b}}^{(9)} \quad (17)$$

(green filled circle), where

$$x_{\text{cross}}^{(\text{left})} = \frac{d_0 d_7}{(d_0 - d_6)} \quad (18)$$

and  $\gamma = 1$  for the initial iteration. Similarly, if the departure point  $\mathbf{x}_{ij^*}^{(2)}$  is located in area 5, the first-guess crossing with side 1 of cell  $(i + 1, j)$  is given by

$$\chi_3 + \gamma x_{\text{cross}}^{(\text{right})} \hat{\mathbf{b}}^{(5)} \quad (19)$$

(green filled circle), where

$$x_{\text{cross}}^{(\text{right})} = \frac{d_1 d_5}{(d_1 - d_4)} \quad (20)$$

(cyan filled circle).

- If both departure points,  $\mathbf{x}_{ij^*}^{(1)}$  and  $\mathbf{x}_{ij^*}^{(2)}$ , are located in area 3, then they are given by

$$\mathbf{x}_{ij^*}^{(1)} = \chi_1 + d_0 \mathbf{b}_1 + d_3 \mathbf{b}_3, \quad (21)$$

$$\mathbf{x}_{ij^*}^{(2)} = \chi_2 + d_1 \mathbf{b}_1 + d_2 \mathbf{b}_2. \quad (22)$$

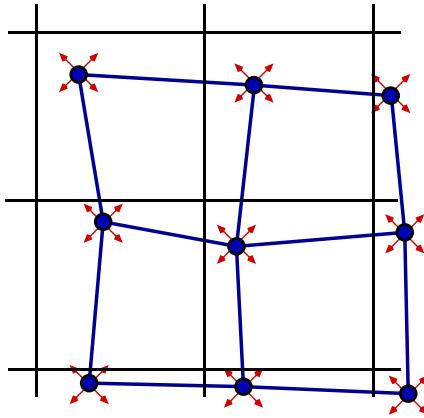
There is no crossing between the upstream flux side and the coordinate lines. We introduce a point halfway between the departure points:

$$\frac{1}{2} (\mathbf{x}_{ij^*}^{(1)} + \mathbf{x}_{ij^*}^{(2)}), \quad (23)$$

and it is perturbed along a vector perpendicular to the flux side—that is, a  $90^\circ$  counterclockwise rotation of the vector  $\mathbf{x}_{ij^*}^{(2)} - \mathbf{x}_{ij^*}^{(1)}$  (see Fig. 6; case 0).

Defining the swept flux overlap areas as a function of  $\gamma$  as described above guarantees that the swept flux area is simply connected and that the iteration problem is well posed. Also, it provides an accurate approximation of flux areas across cubed-sphere panel sides. The vertices for the overlap areas  $\delta a_k$  are defined in terms of the displacement variables  $d$ , unit vectors  $\mathbf{b}$ , coordinate

(a) Global iteration problem



(b) Local iteration problem

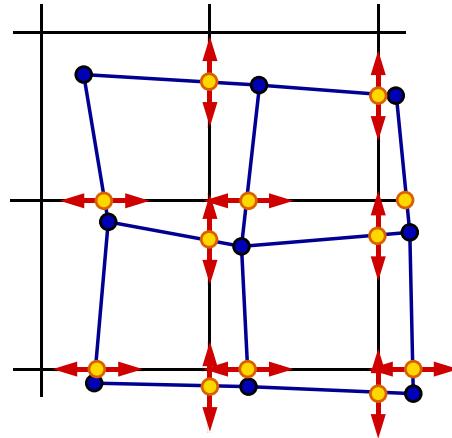


FIG. 7. A graphical illustration of two different ways to make CSLAM and SE consistent. The dark blue filled circles are the departure points, the dark blue lines are the sides of the Lagrangian CSLAM cells, and the black lines are the CSLAM control volumes. The arrows indicate the points that are perturbed to make SE and CSLAM consistent. (a) SE and CSLAM are made consistent by perturbing the departure points. Note that this problem is a global iteration problem requiring computation of overlap areas during each iteration. (b) The approach chosen in this paper. The intersections between the sides of the Lagrangian cells with the grid lines are perturbed. This constitutes a local iteration problem as fluxes are matched with SE fluxes for each flux side, whereas the departure points stay fixed.

vertices  $\chi$ , and the crossings (if applicable) as defined above. The departure point  $\mathbf{x}_{ij*}^{(1)}$  and  $\mathbf{x}_{ij*}^{(2)}$  are given by

$$\mathbf{x}_{ij*}^{(1)} = \begin{cases} \chi_6 + d_7 \mathbf{b}_{10} - d_6 \mathbf{b}_{11}, & \mathbf{x}_{ij*}^{(1)} \text{ in area 1,} \\ \chi_4 - d_3 \mathbf{b}_9 + d_6 \mathbf{b}_8, & \mathbf{x}_{ij*}^{(1)} \text{ in area 2,} \\ \chi_1 + d_0 \mathbf{b}_1 + d_3 \mathbf{b}_3, & \mathbf{x}_{ij*}^{(1)} \text{ in area 3,} \end{cases} \quad (24)$$

and

$$\mathbf{x}_{ij*}^{(2)} = \begin{cases} \chi_1 + d_1 \mathbf{b}_1 + d_2 \mathbf{b}_2, & \mathbf{x}_{ij*}^{(2)} \text{ in area 3,} \\ \chi_3 - d_2 \mathbf{b}_5 + d_4 \mathbf{b}_4, & \mathbf{x}_{ij*}^{(2)} \text{ in area 4,} \\ \chi_5 + d_5 \mathbf{b}_6 - d_4 \mathbf{b}_7, & \mathbf{x}_{ij*}^{(2)} \text{ in area 5.} \end{cases} \quad (25)$$

respectively. If areas 1 and 2 are on a panel different than areas 3, 4, and 5, the formulas still apply; hence, there is no special treatment (if-else logic) for swept flux areas spanning panel edges. The iteration problem is well posed if the flow deformation prescribed by the SE fluxes satisfies the Lipschitz criterion. It is solved using the same Newton algorithm as described in section 3a. Note again that certain line integrals are static (e.g., the line integrals from the Eulerian vertices to the departure points) and do not need to be recomputed during each Newton iteration. Also, only at initialization it is necessary to compute intersections between line segments when  $\delta a_1$  and/or  $\delta a_5$  are nonempty. This is a major simplification compared to the “traditional” CSLAM algorithm.

The above algorithm constitutes a local iteration problem since a point along the side of the swept area is perturbed to obtain CSLAM fluxes consistent with the diagnosed SE fluxes. In other words, the vertex locations of the swept fluxes are determined by the orthogonal swept fluxes (illustrated in Figs. 3a–c). The upstream vertex locations stay fixed. We then introduce an extra degree of freedom along the flux side (which coincides with a coordinate line crossing else it is halfway in between the upstream vertices) and move that point (along the coordinate line) to make the SE and CSLAM fluxes match. Since the vertex locations stay fixed, the iteration only affects the swept flux area for the side in question. Therefore, the iteration algorithm is local and constrained on each side by the prescribed SE flux. As illustrated in Fig. 7, the union of the swept flux areas and the Eulerian cell define the equivalent upstream Lagrangian area. This equivalent upstream Lagrangian grid is shown in Fig. 7b for the new CSLAM algorithm. Note that had we chosen to perturb the departure points (Fig. 7a), then a global iteration problem would result since the departure points are shared between Lagrangian cells (or equivalently swept flux areas). This global iteration problem would also require intersections between Eulerian and Lagrangian cells to be computed during each iteration, which is an expensive operation. For example, if the global iteration problem would converge in 10 iterations, the cost (in the current implementation of the original CSLAM) would be

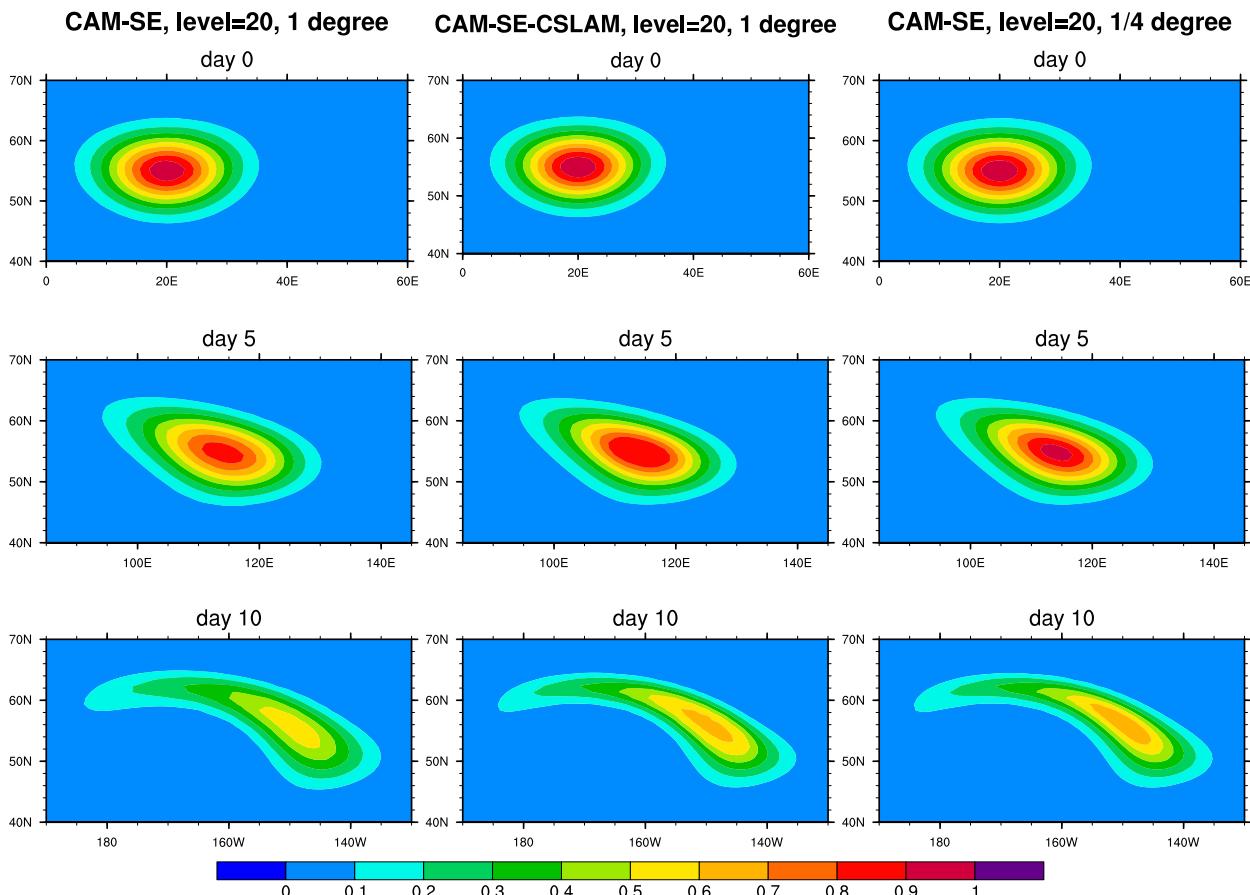


FIG. 8. (columns from left to right)  $1^\circ$  SE,  $1^\circ$  CSLAM, and  $1/4^\circ$  SE solutions for tracer one  $Q_1$  at (top) day 0, (middle) day 5, and (bottom) day 10 at level 20 (at approximately 610 hPa).

equivalent to transporting approximately 90–140 tracers with the SE method (not shown).

#### 4. Results

To assess the accuracy and efficiency of the CAM-SE-CSLAM algorithm, idealized baroclinic wave simulations have been performed with inert tracer distributions as well as idealized reactive chemical species. We compare the SE advection algorithm with the CSLAM advection algorithm. Note that the two transport schemes are driven by the same winds and air mass fields from the CAM-SE solution to the equations of motion. (The tests have been performed with the <https://svn-homme-model.cgd.ucar.edu/branches/cslam@5168> code base.) This model version uses a floating Lagrangian vertical coordinate (Starr 1945; Lin 2004) but is otherwise equivalent to the CAM-SE version documented in Neale et al. (2012).

The model is run with 30 vertical levels (level locations as in CAM5), and the horizontal resolution is approximately  $1^\circ$  ( $30 \times 30$  elements on each cubed-sphere

panel and  $4 \times 4$  GLL quadrature points in each element). The time step for solving the equations of motion (including SE tracer transport) is 300 s, the hyperviscosity time step is 100 s, and the hyperviscosity coefficient is  $10^{15} \text{ m}^4 \text{ s}^{-1}$ . High-resolution reference solutions are computed with a horizontal resolution of approximately  $1/4^\circ$  ( $120 \times 120$  elements on each cubed-sphere panel) with a dynamics time step of 75 s, a hyperviscosity time step of 12.5 s, and a hyperviscosity coefficient of  $10^{13} \text{ m}^4 \text{ s}^{-1}$ . The vertical remapping is performed every third dynamics time step.

The CSLAM scheme has been implemented within the element structure (Erath et al. 2012) where each element has been subdivided into  $3 \times 3$  control volumes (using equiangular partitioning). Hence, the CSLAM grid resolution is also approximately  $1^\circ$  when using  $30 \times 30$  elements on each panel. Contrary to Lauritzen et al. (2010b), we use quadratic interpolation (instead of cubic) in the halo regions to interpolate cell average values on neighboring panels to halo cells resulting from extending the panel in question during the reconstruction process.

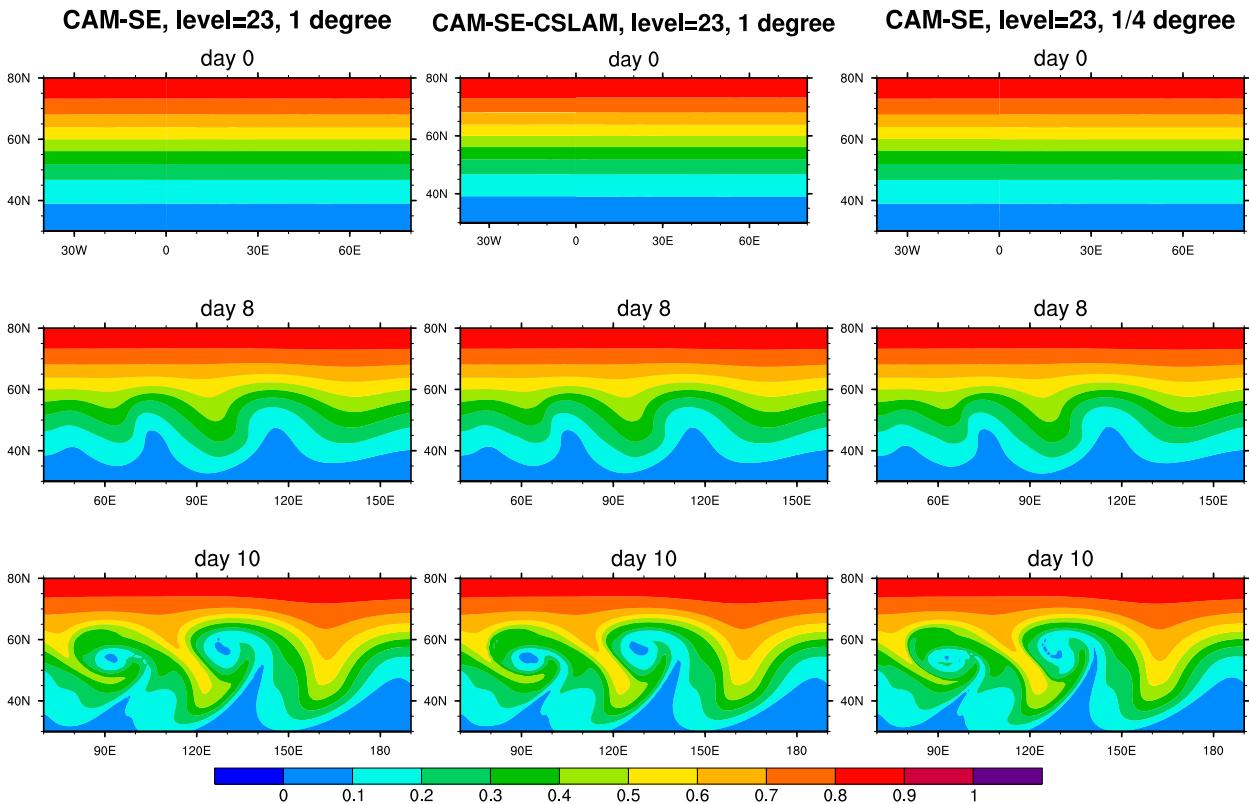


FIG. 9. As in Fig. 8, but for tracer two  $Q_2$  and (top) day 0, (middle) day 8, and (bottom) day 10.

CSLAM has been implemented for Courant numbers less than unity; hence, CSLAM is run with the remapping time step (i.e., 3 times longer time step than SE advection).

The idealized baroclinic wave used here is defined in Ullrich et al. (2014). This wave is similar to the Jablonowski and Williamson (2006) wave but with analytic support for deep atmosphere approximations. Both waves consist of a steady-state solution to the equations of motion with a perturbation that triggers the growth of a baroclinic wave. For this study, either of the baroclinic waves could have been used. In separate sections below, we describe the inert tracer transport test and reactive chemical species test, both being driven by the same SE solution to the Ullrich et al. (2014) baroclinic wave. For simplicity, we assume that the tracers are dimensionless.

*a. Idealized baroclinic wave with inert tracers*

Three ( $Q_1, Q_2, Q_3$ ) inert tracer distributions are used. The first tracer is a smooth ‘‘Gaussian ball’’ distribution:

$$Q_1(\lambda, \theta, \eta) = \begin{cases} 0, & \text{if } |\tilde{Q}_1| < 10^{-8}, \\ \tilde{Q}_1, & \text{if } |\tilde{Q}_1| \geq 10^{-8}, \end{cases} \quad (26)$$

where  $\eta$  is the hybrid vertical coordinate,

$$\tilde{Q}_1(\lambda, \theta, \eta) = \exp \left\{ - \left[ \left( \frac{r}{R} \right)^2 + \left( \frac{\eta - \eta_c}{\eta_{hw}} \right)^2 \right] \right\}, \quad (27)$$

$\lambda$  is longitude,  $\theta$  is latitude, maximum amplitude in the vertical is placed at  $\eta_c = 0.1$  (which is near the surface where strong gradients occur during the baroclinic wave evolution), horizontal half-width  $\tilde{R} = R/10$  (where  $R$  is the mean radius of Earth), and vertical half-width  $\eta_{hw}$  is 0.1. The great-circle distance  $r$  is measured as the distance between  $(\lambda, \theta)$  and the center of the initial distribution  $(\lambda_c, \theta_c) = (\pi/9, 11\pi/18) = (40^\circ\text{N}, 20^\circ\text{E})$ :

$$r = R \arccos[\sin\theta_c \sin\theta + \cos\theta_c \cos\theta \cos(\lambda - \lambda_c)]. \quad (28)$$

The initial condition is shown in row 1 in Fig. 8.

The second tracer distribution structurally resembles the initial condition temperature field, which is zonally symmetric (see Fig. 9):

$$Q_2(\lambda, \theta, \eta) = \frac{1}{2} [\tanh(3|\theta| - \pi) + 1]. \quad (29)$$

The third distribution is the slotted cylinder with discontinuous gradients (see Fig. 10):



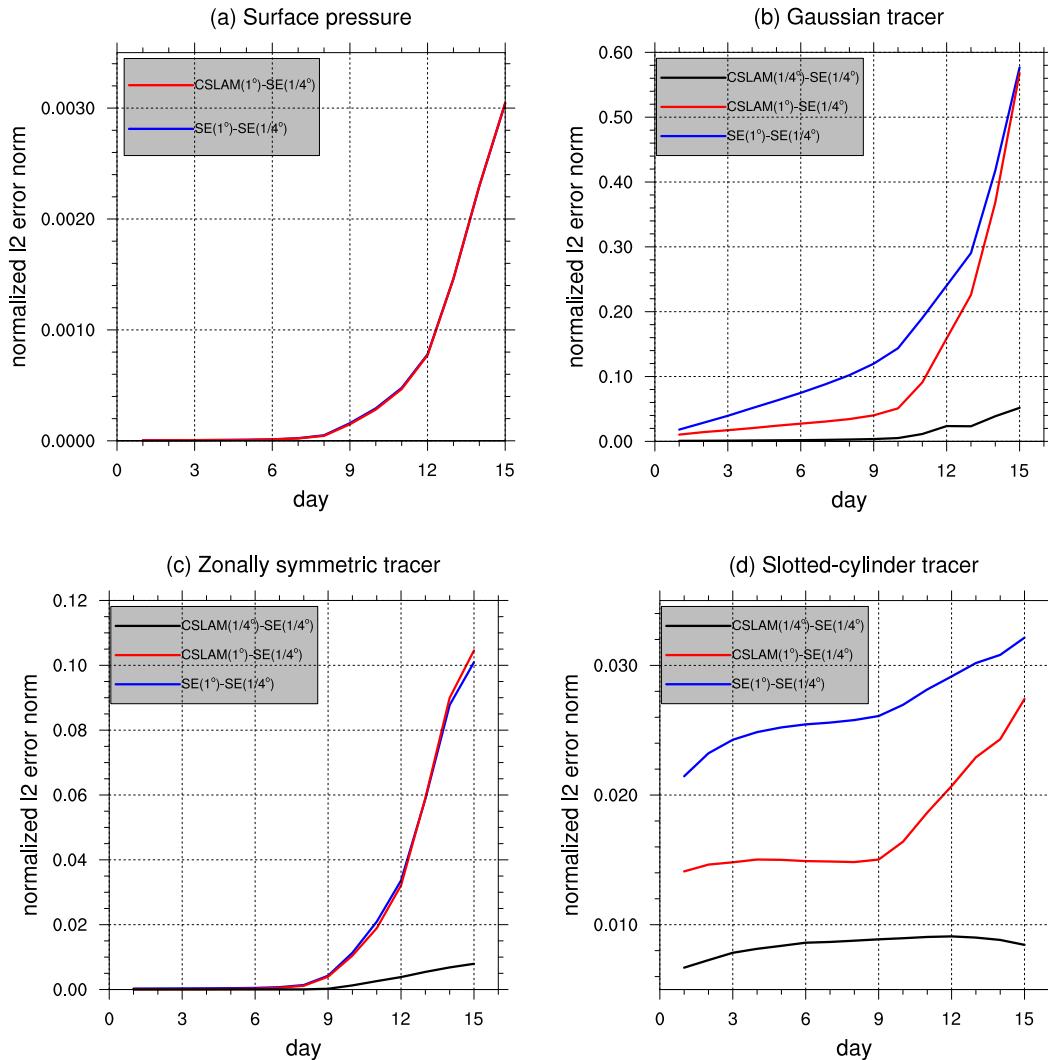


FIG. 11. Standard  $\ell_2$  error norm for (a)  $p_s$ , (b)  $Q_1$ , (c)  $Q_2$ , and (d)  $Q_3$  using SE and CSLAM. The  $1/4^\circ$  SE solution is used as the “truth” for  $\ell_2$  computations.

high-resolution solutions computed with CSLAM ( $\psi$ ) and SE ( $\Psi$ ). As expected, the high-resolution solutions differ a lot less (approximately an order of magnitude difference in  $\ell_2$ ) than the low-resolution solutions (cf. red and blue lines with black lines). For the relatively smooth Gaussian initial condition  $Q_1$ , the CSLAM  $1^\circ$  solution is about twice as accurate compared to the SE solution until the baroclinic wave starts growing rapidly. This can also be seen in the instantaneous fields shown in Fig. 8.

For the zonally symmetric distribution  $Q_2$  (see Fig. 9), the errors for SE and CSLAM are completely dominated by the differences in the winds and pressure fields, which is expected for initially smooth and well-resolved tracer distributions (Fig. 11c). SE transport is formally fourth-order accurate (the addition of hyperviscosity, which

scales as  $dx^{3.2}$ , implies that the scheme is 3.2-order accurate), and CSLAM is third order. For this smooth tracer, both transport schemes provide almost indistinguishable solutions consistent with both schemes being high order for smooth problems. Both schemes drop in formal accuracy (approximately second order; Lauritzen et al. 2014b) when a shape-preserving limiter is used.

In the case of the slotted-cylinder initial distribution  $Q_3$  (see Fig. 10), errors develop immediately because of the models inability to represent a distribution with discontinuous gradients. The CSLAM solution, however, produces errors that are approximately half that of SE until  $\ell_2$  starts saturating by the driving flow differences. This is also visible in the instantaneous solutions shown in Fig. 10, where CSLAM resolves the sharp gradients more accurately than SE.

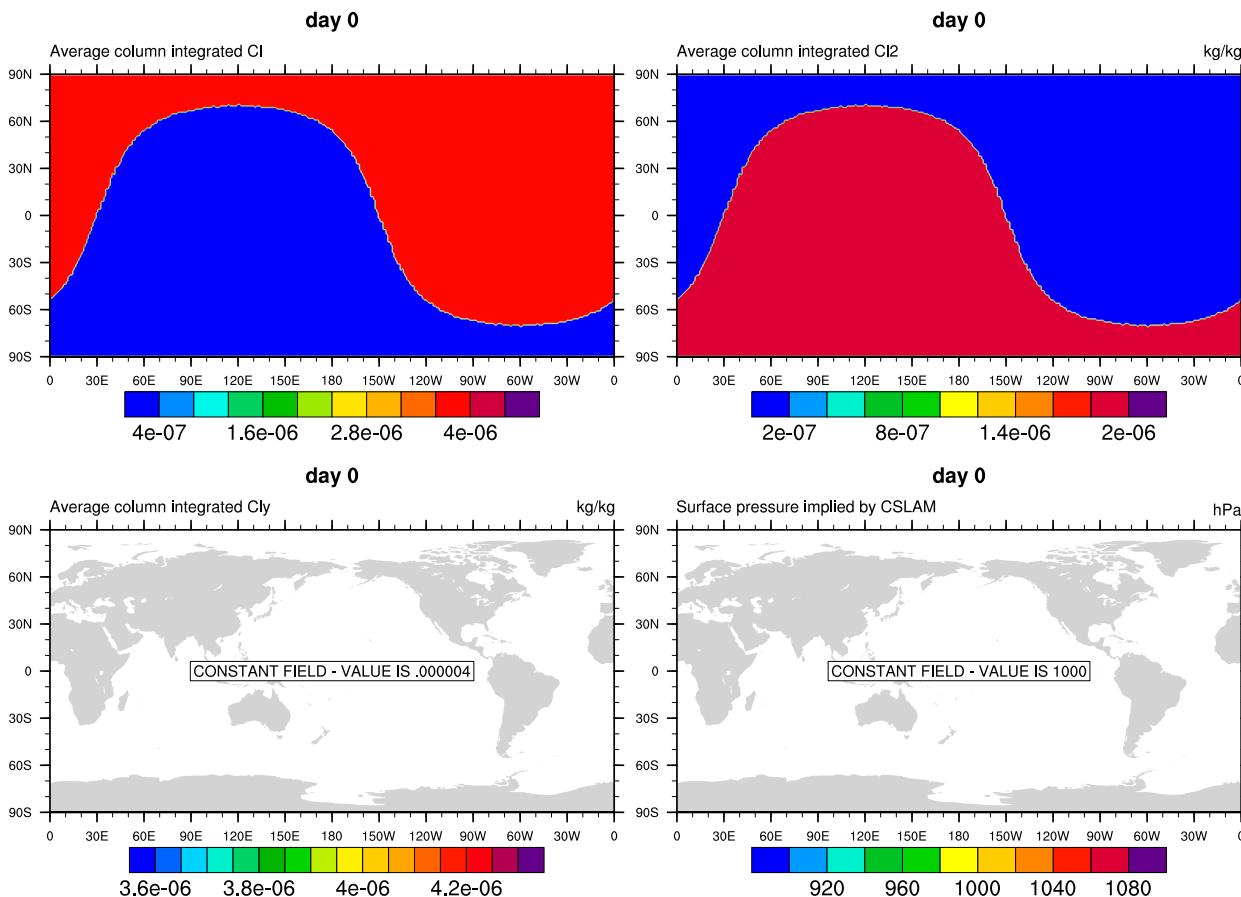


FIG. 12. Initial condition for (top left)  $\langle \text{Cl} \rangle$ , (top right)  $\langle \text{Cl}_2 \rangle$ , (bottom left)  $\langle \text{Cl}_y \rangle$ , and (bottom right)  $p_s$  used in the “toy” terminator chemistry test.

### b. Idealized baroclinic wave with reactive chemical species

To assess the performance of SE and CSLAM in the presence of subgrid-scale forcing, we use a three-dimensional extension of the Lauritzen et al. (2015) idealized terminator test referred to as “the toy terminator chemistry test.” It consists of two reactive species  $Q_4 = \text{Cl}$  and  $Q_5 = \text{Cl}_2$  that initially add up to a constant

$$\text{Cl}_y = \text{Cl} + 2\text{Cl}_2 = 4 \times 10^{-6}, \quad (32)$$

and that react nonlinearly with each other. The chemical reactions are given in Lauritzen et al. (2015). We use the same initial conditions in each level for Cl and Cl<sub>2</sub> as was used in Lauritzen et al. (2015). An advantage of this test is that no matter what flow the chemical species are placed in, the weighted sum (32) must always be conserved. So instead of using a prescribed analytical flow field as in Lauritzen et al. (2015), we use the idealized baroclinic wave flow field. The tendencies from the

chemical reactions are computed every 900 s for the 1<sup>o</sup> computations. The tendencies are used to update the state every 900 s [the ftype = 1 configuration is explained in appendix E in Lauritzen et al. (2015)].

As diagnostics, we used the averaged column integrated quantities:

$$\langle \psi \rangle = \frac{\int_{p=p_s}^{p_{\text{top}}} \psi dp}{\int_{p=p_s}^{p_{\text{top}}} dp}, \quad \psi = \text{Cl}, \text{Cl}_2, \text{Cl}_y. \quad (33)$$

The initial condition for  $\langle \text{Cl} \rangle$ ,  $\langle \text{Cl}_2 \rangle$ ,  $\langle \text{Cl}_y \rangle$ , and  $p_s$  are shown in Fig. 12. The same fields at day 15 are shown in Fig. 13 for CSLAM and Fig. 14 for SE. Note that  $p_s$  for SE and  $p_s$  implied by CSLAM are identical. The SE transport scheme maintains a constant  $\text{Cl}_y$  until day 11, after which errors start growing. In Lauritzen et al. (2015), this error growth was shown to be due to the monotone limiter from Guba et al. (2014). In Lopez and Taylor (2015), it was shown that this limiter preserves

### CAM-SE-CSLAM

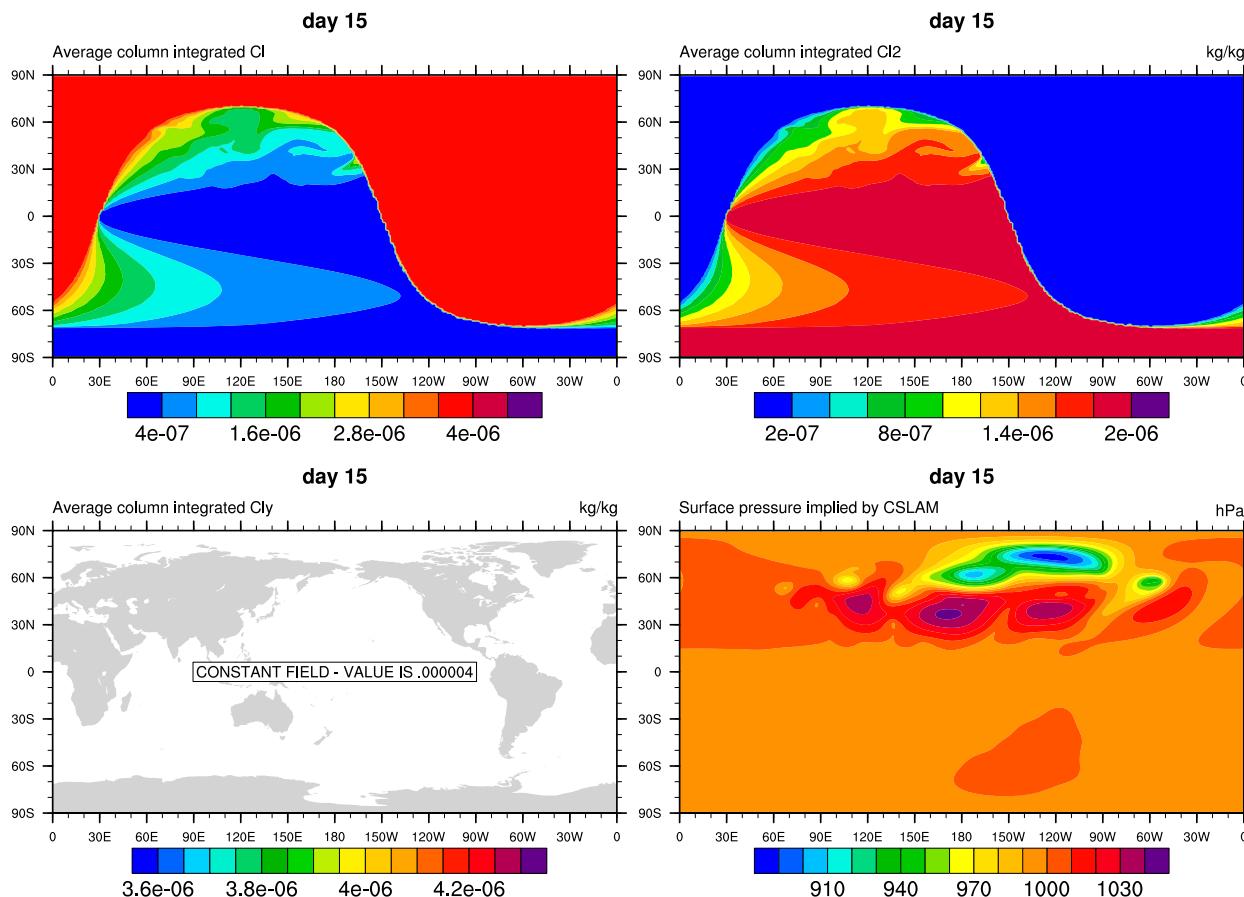


FIG. 13. As in Fig. 12, but for day 15 using CSLAM.

constant  $CI_y$  in exact arithmetic but not in the presence of round-off error. The CSLAM algorithm maintains the constant  $CI_y$  throughout the simulation.

#### c. Computational performance

Computational performance experiments are performed on the NCAR Yellowstone cluster (Computational and Information Systems Laboratory 2012) with standard computer optimization flags. We do not make use of threading. The amount of information that needs to be communicated between elements (if residing on different tasks) is very different between SE and CSLAM—similarly for the frequency of communication between elements. While the SE method only needs to share values at quadrature points located on the boundary of an element, communication is performed at every Runge–Kutta stage and every subcycling of the hyperviscosity operator. For the  $1^\circ$  model, that amounts to 21 communications per remapping time step (900s), 5 communications per 300-s tracer time step, and 2 communications for hyperviscosity per tracer time step.

CSLAM only needs one communication per 900-s tracer time step. That said, CSLAM needs all finite-volume values for each element surrounding the element being updated—that is, 72 values per tracer in the horizontal compared to 20 values for SE (four points on each edge plus the corner points). Figure 15a shows throughput for tracers for a 1-day simulation using SE (purple and black) and CSLAM (green) as a function of the number of tracers. The throughput for SE is shown for the CESM1 release code (purple line). The CAM-SE tracer advection code is currently being optimized further (J. Dennis 2016, personal communication). At a relatively low tracer count (at least for climate and climate-chemistry modeling), CSLAM is faster than SE. Note that the new CSLAM algorithm does not make use of an extensive search algorithm as the original formulation of CSLAM. The crossover between SE and CSLAM with the old formulation of flux-form CSLAM in HOMME (Harris et al. 2011) is over 20 tracers (not shown) and with the new consistent CSLAM algorithm, it is closer to 10 tracers. Figure 15a also shows the breakdown of

## CAM-SE

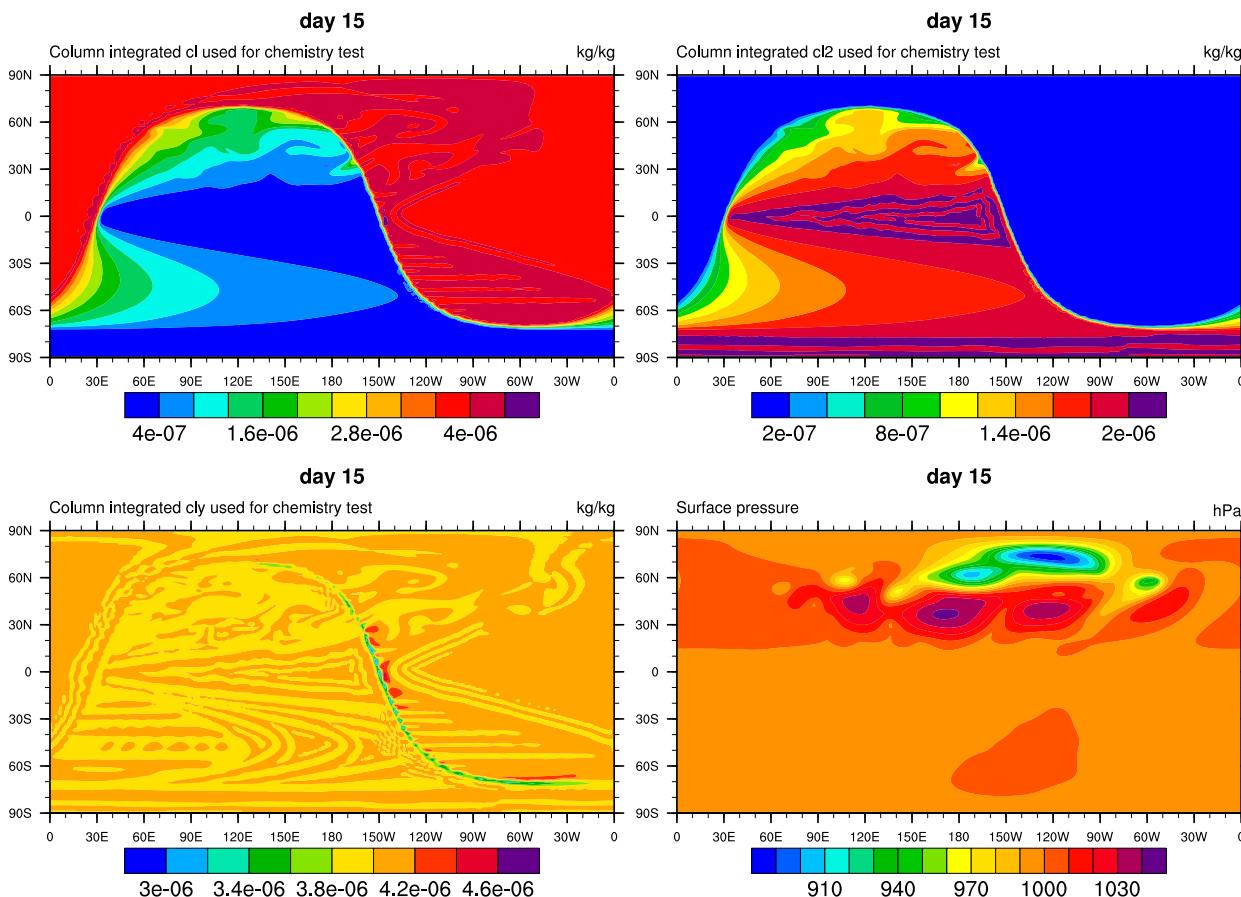


FIG. 14. As in Fig. 12, but for day 15 using SE.

computational cost for the different stages of the CSLAM algorithm. The departure point algorithm (orange line) and computation of high-order weights (yellow line) remains constant as a function of tracers and makes up a small cost even at low tracer counts. The communication (light blue), reconstruction (dark blue), and remapping (multiplication of reconstruction coefficients and precomputed weights; red) are all a function of tracers. The computation of the reconstruction coefficients is the most expensive step, mostly because of the reconstruction limiter that involves if – else logic. About half of the cost of reconstruction is the boundary exchange (communication).

Especially for lower-resolution modeling, strong scaling is important. Figure 15b shows throughput as a function of tasks/processors. Note that the last data point on the right-hand side of the plot (5400 processors) is the throughput where there is only one element in the horizontal per processor. It is well known that SE exhibits strong scaling; however, CSLAM also scales to the limit of scalability despite the large halo.

Compared to the original formulation of CSLAM, the new CSLAM algorithm does not make use of an extensive search algorithm. It has been replaced with a local iterative algorithm to enforce consistency with SE. The communication structure (amount of data communicated and the frequency of communication), reconstruction, shape-preserving limiter, and remapping algorithm are essentially the same as the original CSLAM scheme. Hence, the detailed performance analysis as a function of cores and tracers presented in Erath et al. (2012) is also relevant for the algorithm presented here.

CAM-SE-CSLAM has been presented in the context of a dry dynamical core. The extension to a full physics setup requires several modifications to the existing modeling system. First of all, one needs to make choices about which grid is used to run the physics parameterizations (quadrature grid or CSLAM control volumes). In either case, tendencies from the parameterizations must be mapped between grids. Second, the treatment of moisture in the dynamical core can complicate the

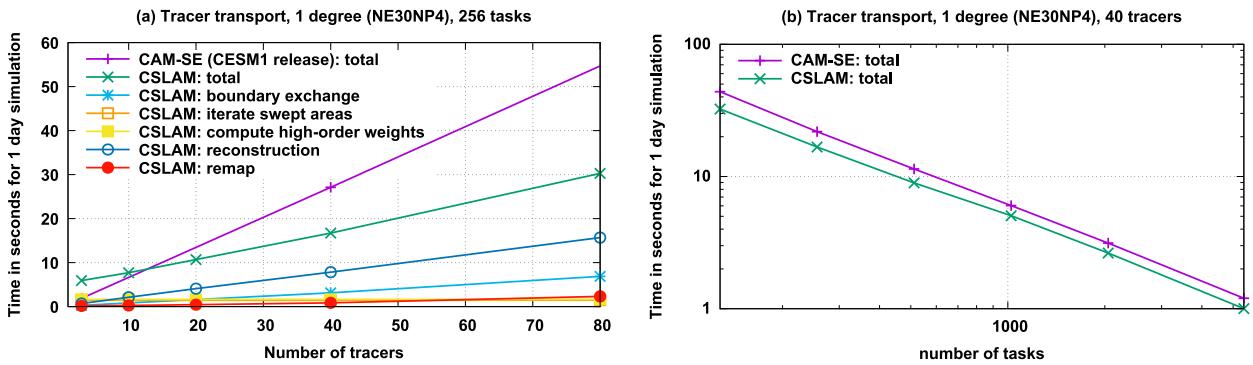


FIG. 15. (a) Time (s) spent in various parts of the CSLAM transport algorithm for a 1-day simulation using 256 tasks as a function of number of tracers (the yellow line is on top of the orange line). The throughput for SE transport based on the CESM1 release (CAM-SE) is also shown. See text for details. (b) As in (a) but for a fixed number of tracers (40) and as a function of tasks. Note that for the rightmost data point (5400 tasks), there is only one element in the horizontal per processor. This is the limit of scalability for CAM-SE and CAM-SE–CSLAM without threading.

coupling between CAM-SE and CSLAM. Currently, CAM-SE is based on a vertical coordinate based on moist (full) surface pressure, which implies that the model levels move when there are changes in the moisture field from parameterized sources/sinks of moisture. These sources/sinks need to be accounted for both in SE and CSLAM. To avoid this, dry-mass vertical coordinates, in which the model levels are determined by dry surface pressure (as well as the usual hybrid coefficients), can be used. With dry-mass vertical coordinates, the model levels do not change because of parameterized water vapor sources/sinks. The infrastructure to run physics, tracers, and dynamics on different grids has been added to CAM, and a dry-mass vertical coordinate version of CAM-SE has been derived and implemented. This version also easily accounts for condensate loading. Details on these developments are beyond the scope of this paper.

### 5. Summary

In this paper, it has been shown that it is possible to consistently couple CAM-SE with the CSLAM transport algorithm without violating desirable properties such as shape preservation, mass conservation, linear correlation preservation, and consistency (aka compatibility, free-stream preservation). This is achieved by first deriving formulas for diagnosing SE air mass flux through the CSLAM control volume faces. These mass fluxes are used to construct CSLAM swept areas for which the air mass fluxes computed with CSLAM match the SE air mass fluxes to round off. The iterative algorithm is local and results in an equivalent upstream Lagrangian CSLAM grid that spans the sphere without gaps or overlaps. This new consistent SE-CSLAM

algorithm avoids the expensive search for overlap areas used in the original formulations of CSLAM. The source code for computing weights has almost been cut in half with the new algorithm compared to the original CSLAM. This is mainly due to the new algorithm being formulated more generally, so special case statements for the edges and corners of the sphere have been reduced to a minimum.

It has been shown in idealized tracer experiments that CSLAM is more accurate than SE for challenging tracer distributions and at the same time is more efficient than SE if one is using more than approximately 12–15 tracers. Also, the CAM-SE–CSLAM model retains the excellent strong scaling properties of CAM-SE. The algorithm for making the CSLAM Lagrangian grid consistent with SE is general in the sense that CSLAM can be made consistent with any air mass fluxes as long as the fluxes satisfy the Lipschitz criterion.

*Acknowledgments.* Steve Goldhaber, James Overfelt, Paul Ullrich, and Mark Taylor were supported by the U.S. Department of Energy Office of Biological and Environmental Research, Work Package 12-015334 “Multiscale Methods for Accurate, Efficient, and Scale-Aware Models of the Earth System.”

### APPENDIX

#### Spectral Element Subcell Flux

Our goal is to write each of the three terms on the right-hand side of (8) as a sum of edge fluxes in such a way that the fluxes along an edge shared by two subcells will be equal in magnitude and of opposite sign.

a. *Strong form divergence:*  $\langle \phi^{(ij)}, \nabla \cdot \mathbf{F} \rangle$

The divergence in curvilinear coordinates is given by

$$\nabla \cdot \mathbf{F} = \frac{1}{J} \left[ \frac{\partial}{\partial x} (JF^1) + \frac{\partial}{\partial y} (JF^2) \right], \quad (\text{A1})$$

where  $J$  is the Jacobian and  $(F^1, F^2)$  are the contravariant components of the flux vector  $\mathbf{F} = \mathbf{v}\Delta p$ .

To compute the implied spectral element flux over the  $ij$  th CSLAM control volume, we construct special test functions  $\phi^{(ij)}$  such that

$$\langle \phi^{(ij)}, p \rangle = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} Jp(x, y) dx dy \quad (\text{A2})$$

for all polynomials  $p(x, y)$  of degree  $np - 1$ . Such test functions must exist since the integral is a linear functional of  $p$ , and  $p$  is uniquely determined by its GLL nodal values. One can further show that  $\phi^{(ij)}$  is a tensor product of one-dimensional test functions  $\phi^{(ij)} = \phi^{(i)}(x)\phi^{(j)}(y)$ . Hence, the flux for the  $ij$ th CSLAM control volume is

$$\begin{aligned} \langle \phi^{(ij)}, \nabla \cdot \mathbf{F} \rangle &= \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \left[ \frac{\partial}{\partial x} (JF^1) + \frac{\partial}{\partial y} (JF^2) \right] dx dy \\ &= \int_{y_j}^{y_{j+1}} JF^1(x_{i+1}, y) dy - \int_{y_j}^{y_{j+1}} JF^1(x_i, y) dy \\ &\quad + \int_{x_i}^{x_{i+1}} JF^2(x, y_{j+1}) dx - \int_{x_i}^{x_{i+1}} JF^2(x, y_j) dx. \end{aligned} \quad (\text{A3})$$

Thus, the flux naturally decomposes into the integral of  $\mathbf{F} \cdot \hat{\mathbf{n}}$  along each edge. These can be computed numerically based on the GLL nodal values via

$$\begin{aligned} \langle \phi^{(ij)}, \nabla \cdot \mathbf{F} \rangle &= - \sum_{m=1}^{np} \sum_{n=1}^{np} w_m \phi_m^{(i)} I_n^{(j)} JF^2(x_m, y_n) \\ &\quad (\text{bottom edge flux}) \\ &\quad + \sum_{m=1}^{np} \sum_{n=1}^{np} w_n \phi_n^{(j)} I_m^{(i+1)} JF^1(x_m, y_n) \\ &\quad (\text{right edge flux}) \\ &\quad + \sum_{m=1}^{np} \sum_{n=1}^{np} w_m \phi_m^{(i)} I_n^{(j+1)} JF^2(x_m, y_n) \\ &\quad (\text{top edge flux}) \\ &\quad - \sum_{m=1}^{np} \sum_{n=1}^{np} w_n \phi_n^{(j)} I_m^{(i)} JF^1(x_m, y_n) \\ &\quad (\text{left edge flux}), \end{aligned} \quad (\text{A4})$$

where  $\mathbf{I}$  is a polynomial interpolation matrix defined so that

$$\Delta p(x_i, y_j) = \sum_{m=1}^{np} \sum_{n=1}^{np} I_m^{(i)} I_n^{(j)} \Delta p(x_m, y_n). \quad (\text{A5})$$

Each term on the right-hand side of (A4) constitutes the contribution to  $\mathcal{F}_{ij}^{(\varepsilon)}$ ,  $\varepsilon = 1, 2, 3, 4$  from the strong flux divergence.

We note that in the SE method, at an edge shared by neighboring elements, both  $J$ ,  $F^1$ , and  $F^2$  will not necessarily be continuous because of deformed elements and different contravariant coordinates used within each element. However, the flux  $JF^1 = \mathbf{F} \cdot \hat{\mathbf{n}}$  is continuous at element boundaries if  $\mathbf{F}$  is continuous (Taylor and Fournier 2010).

b. *Weak form diffusion:*  $\langle \nabla \phi^{(ij)}, \mathbf{G} \rangle$

This term, where  $\mathbf{G} = \nu \nabla \Delta p$ , does not change the mass within an element and thus has zero net flux at element boundaries. It does induce mixing between subcells and will change the nodal values of  $\Delta p$ . The DSS operation can then redistribute some of this mass into neighboring cells. However, that flux is accounted for in the DSS operation below, not here.

For each subinterval, we need to decompose the change in mass into left, right, top, and bottom fluxes:  $G_L^{(i)}$ ,  $G_R^{(i)}$ ,  $G_T^{(i)}$ , and  $G_B^{(i)}$ , respectively. The requirement, that the flux at the sides of the element is zero and that the fluxes at CSLAM control volume edges are equal and with opposite signs, is sufficient to uniquely determine all the fluxes.

In each coordinate direction, we have

$$\left\langle \frac{\partial \phi^{(i)}}{\partial x}, G \right\rangle = G_L^{(i)} + G_R^{(i)} \quad (\text{A6})$$

(similarly for the top and bottom). Starting with the leftmost ( $i = 1$ ) subinterval, we know that  $G_L^{(1)} = 0$ , and then (A7) can be used to determine  $G_R^{(1)}$ . Since the fluxes at adjacent subelement boundaries must be equal and opposite, this determines  $G_L^{(2)} = -G_R^{(1)}$ , and as before, once  $G_L^{(2)}$  is determined, (A6) determines  $G_R^{(2)}$ .

Since  $\phi^{(i)}$  is a partition of unity, it can be shown that this procedure will end with the flux at the interval endpoint of  $G_R^{(i)} = 0$ , as required.

Thus these fluxes are given by

$$G_L^{(i)} = \begin{cases} 0 & \text{if } i = 1 \\ -G_R^{(i-1)} & \text{if } i > 1 \end{cases}, \quad (\text{A7})$$

$$G_R^{(i)} = \begin{cases} \left\langle \frac{\partial}{\partial x} \phi^{(i)}, G \right\rangle - G_L^{(i)} & \text{if } i < nc \\ 0 & \text{if } i = nc \end{cases}. \quad (\text{A8})$$

In two dimensions, we first integrate by parts (weak form):

$$\begin{aligned} \langle \nabla \phi, \mathbf{G} \rangle &= \sum_{m=1}^{np} \sum_{n=1}^{np} w_m w_n \\ &\times \left( \phi_n^{(j)} \frac{\partial \phi_m^{(i)}}{\partial x} JG^1 + \phi_m^{(i)} \frac{\partial \phi_n^{(j)}}{\partial y} JG^2 \right) \\ &+ \sum_j w_n \phi_n^{(j)} \left( \sum_{m=1}^{np} w_m \frac{\partial \phi_m^{(i)}}{\partial x} JG^1 \right) \\ &+ \sum_i w_m \phi_m^{(i)} \left( \sum_{n=1}^{np} w_n \frac{\partial \phi_n^{(j)}}{\partial y} JG^2 \right). \quad (\text{A9}) \end{aligned}$$

Then using the results from above, we get

$$\sum_{m=1}^{np} w_m \frac{\partial \phi_m^{(i)}}{\partial x} JG^1 = G_L^{(i)}(y_n) + G_R^{(i)}(y_n), \quad (\text{A10})$$

$$\sum_{n=1}^{np} w_n \frac{\partial \phi_n^{(j)}}{\partial y} JG^2 = G_T^{(j)}(x_m) + G_B^{(j)}(x_m), \quad (\text{A11})$$

and we can write the diffusion term as a sum of four edge fluxes:

$$\begin{aligned} \langle \nabla \phi, \mathbf{G} \rangle &= \sum_i w_m \phi_m^{(i)} G_B^{(i)}(x_m) + \sum_j w_n \phi_n^{(j)} G_R^{(j)}(y_n) \\ &+ \sum_i w_m \phi_m^{(i)} G_T^{(i)}(x_m) + \sum_j w_n \phi_n^{(j)} G_L^{(j)}(y_n). \quad (\text{A12}) \end{aligned}$$

Each term on the right-hand side of (A12) constitutes the contribution to  $\mathcal{F}_{ij}^{(\varepsilon)}$ ,  $\varepsilon = 1, 2, 3, 4$  from the diffusion operator.

*c. DSS operation:  $\langle \phi^{(ij)}, \mathbf{D} \rangle$*

We now treat the edge flux term coming from the DSS operation. First, consider the DSS term along a coordinate line: In one dimension, the net change in mass in the entire element is given by

$$\langle 1, D \rangle = w_1 D(x_1) + w_{np} D(x_{np}), \quad (\text{A13})$$

where we have used the fact that  $D = 0$  at element interior points. This term naturally decomposes into a flux at the left and right interval endpoints

$$D_L^{(1)} = w_1 D(x_1), \quad (\text{A14})$$

$$D_R^{(i)} = w_{np} D(x_{np}), \quad (\text{A15})$$

with

$$\langle 1, D \rangle = D_L^{(1)} + D_R^{(i)}. \quad (\text{A16})$$

We first note that this flux is conservative at the element level since the flux on the right edge of one element will be equal and opposite to the flux on the left edge of its adjacent element.

As for the diffusion term, given  $D_L^{(1)}$ , the remaining fluxes are determined by requiring that adjacent fluxes are equal and opposite,  $D_R^{(i)} = -D_L^{(i+1)}$ , and that the change in mass from the DSS term in each subinterval is given by the sum of the subinterval fluxes:

$$\langle \phi^{(i)}, D \rangle = D_L^{(i)} + D_R^{(i)}. \quad (\text{A17})$$

The fact that  $\phi^{(i)}$  is a partition of unity ensures that the final flux computed by this procedure,  $G_R^{(i)}$ , will satisfy (A15).

The result can be written as

$$D_L^{(i)} = \begin{cases} w_1 D(x_1) & \text{if } i = 1 \\ -D_R^{(i-1)} & \text{if } i > 1 \end{cases}, \quad (\text{A18})$$

$$D_R^{(i)} = \begin{cases} \langle \phi^{(i)}, D \rangle - D_L^{(i)} & \text{if } i < nc \\ w_{np} D(x_{np}) & \text{if } i = nc \end{cases}. \quad (\text{A19})$$

In two dimensions, we have

$$\begin{aligned} \langle \phi^{(ij)}, D \rangle &= \sum_{m=1}^{np} \sum_{n=1}^{np} w_m w_n \phi_m^{(i)} \phi_n^{(j)} (JD)(x_m, y_n) \\ &= \frac{1}{2} \sum_i w_m \phi_m^{(i)} \left[ \sum_{n=1}^{np} w_n \phi_n^{(j)} (JD)(x_m, y_n) \right] \\ &\quad + \frac{1}{2} \sum_{n=1}^{np} w_n \phi_n^{(j)} \left[ \sum_i w_m \phi_m^{(i)} (JD)(x_m, y_n) \right]. \quad (\text{A20}) \end{aligned}$$

Applying the one-dimensional (1D) procedure from above to this term and denoting the resulting fluxes (which are now a function of  $x_m$ ) at the top and bottom edges by  $D_T^{(j)}$  and  $D_B^{(j)}$ , respectively, we get

$$\sum_{n=1}^{np} w_n \phi_n^{(j)} (JD)(x_m, y_n) = D_T^{(j)}(x_m) + D_B^{(j)}(x_m), \quad (\text{A21})$$

$$\sum_{m=1}^{np} w_m \phi_m^{(i)} (JD)(x_m, y_n) = D_L^{(i)}(y_n) + D_R^{(i)}(y_n), \quad (\text{A22})$$

so that

$$\begin{aligned} \langle \phi^{(ij)}, D \rangle &= \frac{1}{2} \sum_{m=1}^{np} w_m \phi_m^{(i)} D_B^{(j)}(x_m) + \frac{1}{2} \sum_{n=1}^{np} w_n \phi_n^{(j)} D_R^{(i)}(y_n) \\ &\quad + \frac{1}{2} \sum_{m=1}^{np} w_m \phi_m^{(i)} D_T^{(j)}(x_m) + \frac{1}{2} \sum_{n=1}^{np} w_n \phi_n^{(j)} D_L^{(i)}(y_n). \quad (\text{A23}) \end{aligned}$$

Each term on the right-hand side of (A23) constitute the contribution to  $\mathcal{F}_{ij}^{(e)}$ ,  $\varepsilon = 1, 2, 3, 4$  from the DSS operator.

The definition of  $D_T^{(j)}$  and  $D_B^{(j)}$  is a simple extension of  $D_L^{(j)}$  and  $D_R^{(j)}$  as defined in (A18) and (A19). The only difficulty is at the corners,  $(x_1, y_1)$ ,  $(x_1, y_{np})$ ,  $(x_{np}, y_1)$ ,  $(x_{np}, y_{np})$ , where some ambiguity exists in how  $D$  is conservatively distributed among the element edges that share the corner node—that is,  $D(x_1, y_1) = D_T^{(1)}(x_1) + D_L^{(1)}(y_1)$ . We use the fact that all the DSS terms from all the elements that share a corner must sum to zero,  $\sum D^e(x_0) = 0$ , where the sum is taken over the three or four elements,  $e$ , that share a single corner point,  $x_0$ . We can then divide the contributions among each of the three or four elements, noting that even elements that share only a single node and not an edge will have flux contributions that flow through intermediate elements. For three elements sharing a point, the flux is based on one-third the difference of the DSS terms across neighboring faces. The formula for four elements is similar.

#### REFERENCES

- Barth, T. J., and D. C. Jespersen, 1989: The design and application of upwind schemes on unstructured meshes. *Proc. AIAA 27th Aerospace Sciences Meeting*, Reno, NV, American Institute of Aeronautics and Astronautics, 1–12, doi:10.2514/6.1989-366.
- Computational and Information Systems Laboratory, 2012: Yellowstone: IBM iDataPlex system. NCAR/UCAR. [Available online at <http://n2t.net/ark:/85065/d7wd3xhc>.]
- Dawson, C., 1999: Conservative, shock-capturing transport methods with nonconservative velocity approximations. *Comput. Geosci.*, **3**, 205–227, doi:10.1023/A:101153931766.
- Deleersnijder, E., and G. Lebon, 2001: Enforcing the continuity equation in numerical models of geophysical fluid flows. *Appl. Math. Lett.*, **14**, 867–873, doi:10.1016/S0893-9659(01)00057-X.
- Dennis, J., A. Fournier, W. Spitz, A. St-Cyr, M. A. Taylor, S. J. Thomas, and H. Tufo, 2005: High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core. *Int. J. High Perform. Comput. Appl.*, **19**, 225–235, doi:10.1177/1094342005056108.
- , and Coauthors, 2012: CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model. *Int. J. High. Perform. Comput. Appl.*, **26**, 74–89, doi:10.1177/1094342011428142.
- Deville, M. O., P. F. Fischer, and E. H. Mund, 2002: *High-Order Methods for Incompressible Fluid Flow*. Cambridge Monographs on Applied and Computational Mathematics, No. 9, Cambridge University Press, 528 pp.
- Erath, C., and R. D. Nair, 2014: A conservative multi-tracer transport scheme for spectral-element spherical grids. *J. Comput. Phys.*, **256**, 118–134, doi:10.1016/j.jcp.2013.08.050.
- , P. H. Lauritzen, J. H. Garcia, and H. M. Tufo, 2012: Integrating a scalable and efficient semi-Lagrangian multi-tracer transport scheme in HOMME. *Procedia Comput. Sci.*, **9**, 994–1003, doi:10.1016/j.procs.2012.04.106.
- , —, and H. M. Tufo, 2013: On mass conservation in high-order high-resolution rigorous remapping schemes on the sphere. *Mon. Wea. Rev.*, **141**, 2128–2133, doi:10.1175/MWR-D-13-00002.1.
- Gross, E. S., L. Bonaventura, and G. Rosatti, 2002: Consistency with continuity in conservative advection schemes for free-surface models. *Int. J. Numer. Methods Fluids*, **38**, 307–327, doi:10.1002/flid.222.
- Guba, O., M. Taylor, and A. St-Cyr, 2014: Optimization-based limiters for the spectral element method. *J. Comput. Phys.*, **267**, 176–195, doi:10.1016/j.jcp.2014.02.029.
- Hall, D. M., P. A. Ullrich, K. A. Reed, C. Jablonowski, R. D. Nair, and H. M. Tufo, 2016: Dynamical Core Model Intercomparison Project (DCMIP) tracer transport test results for CAM-SE. *Quart. J. Roy. Meteor. Soc.*, **142**, 1672–1684, doi:10.1002/qj.2761.
- Harris, L. M., P. H. Lauritzen, and R. Mittal, 2011: A flux-form version of the Conservative Semi-Lagrangian Multi-tracer transport scheme (CSLAM) on the cubed sphere grid. *J. Comput. Phys.*, **230**, 1215–1237, doi:10.1016/j.jcp.2010.11.001.
- Jablonowski, C., and D. L. Williamson, 2006: A baroclinic instability test case for atmospheric model dynamical cores. *Quart. J. Roy. Meteor. Soc.*, **132**, 2943–2975, doi:10.1256/qj.06.12.
- Kuo, H.-C., and R. T. Williams, 1990: Semi-Lagrangian solutions to the inviscid Burgers equation. *Mon. Wea. Rev.*, **118**, 1278–1288, doi:10.1175/1520-0493(1990)118<1278:SLSTTI>2.0.CO;2.
- Lauritzen, P. H., and J. Thuburn, 2012: Evaluating advection/transport schemes using interrelated tracers, scatter plots and numerical mixing diagnostics. *Quart. J. Roy. Meteor. Soc.*, **138**, 906–918, doi:10.1002/qj.986.
- , C. Jablonowski, M. A. Taylor, and R. D. Nair, 2010a: Rotated versions of the Jablonowski steady-state and baroclinic wave test cases: A dynamical core intercomparison. *J. Adv. Model. Earth Syst.*, **2**, 15, doi:10.3894/JAMES.2010.2.15.
- , R. D. Nair, and P. A. Ullrich, 2010b: A Conservative Semi-Lagrangian Multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *J. Comput. Phys.*, **229**, 1401–1424, doi:10.1016/j.jcp.2009.10.036.
- , P. A. Ullrich, and R. D. Nair, 2011: Atmospheric transport schemes: Desirable properties and a semi-Lagrangian view on finite-volume discretizations. *Numerical Techniques for Global Atmospheric Models*, P. H. Lauritzen et al., Eds., Lecture Notes in Computational Science and Engineering, Vol. 80, Springer, 185–250, doi:10.1007/978-3-642-11640-7\_8.
- , W. C. Skamarock, M. J. Prather, and M. A. Taylor, 2012: A standard test case suite for two-dimensional linear transport on the sphere. *Geosci. Model Dev.*, **5**, 887–901, doi:10.5194/gmd-5-887-2012.
- , J. T. Bacmeister, T. Dubos, S. Lebonnois, and M. A. Taylor, 2014a: Held-Suarez simulations with the Community Atmosphere Model spectral element (CAM-SE) dynamical core: A global axial angular momentum analysis using Eulerian and floating Lagrangian vertical coordinates. *J. Adv. Model. Earth Syst.*, **6**, 129–140, doi:10.1002/2013MS000268.
- , and Coauthors, 2014b: A standard test case suite for two-dimensional linear transport on the sphere: Results from a collection of state-of-the-art schemes. *Geosci. Model Dev.*, **7**, 105–145, doi:10.5194/gmd-7-105-2014.
- , A. J. Conley, J.-F. Lamarque, F. Vitt, and M. A. Taylor, 2015: The terminator “toy” chemistry test: A simple tool to assess errors in transport schemes. *Geosci. Model Dev.*, **8**, 1299–1313, doi:10.5194/gmd-8-1299-2015.
- Lin, S.-J., 2004: A “vertically Lagrangian” finite-volume dynamical core for global models. *Mon. Wea. Rev.*, **132**, 2293–2307, doi:10.1175/1520-0493(2004)132<2293:AVLFDC>2.0.CO;2.

- , and R. B. Rood, 1996: Multidimensional flux-form semi-Lagrangian transport schemes. *Mon. Wea. Rev.*, **124**, 2046–2070, doi:10.1175/1520-0493(1996)124<2046:MFFSLT>2.0.CO;2.
- Lopez, N. A., and M. A. Taylor, 2015: Improving the tracer correlation problem in a spectral element dynamical core. *Center for Computing Research Summer Proc. 2015*, Albuquerque, NM, Sandia National Laboratories, 106–117. [Available online at <https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/proceedings/ccr15.pdf#page=112>.]
- Nair, R. D., and P. H. Lauritzen, 2010: A class of deformational flow test cases for linear transport problems on the sphere. *J. Comput. Phys.*, **229**, 8868–8887, doi:10.1016/j.jcp.2010.08.014.
- , J. S. Scroggs, and F. H. M. Semazzi, 2002: Efficient conservative global transport schemes for climate and atmospheric chemistry models. *Mon. Wea. Rev.*, **130**, 2059–2073, doi:10.1175/1520-0493(2002)130<2059:ECGTSF>2.0.CO;2.
- , H.-W. Choi, and H. M. Tufo, 2009: Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core. *Comput. Fluids*, **38**, 309–319, doi:10.1016/j.compfluid.2008.04.006.
- Neale, R. B., and Coauthors, 2012: Description of the NCAR Community Atmosphere Model (CAM 5.0). NCAR Tech. Note NCAR/TN-486+STR, 274 pp. [Available online at [http://www.cesm.ucar.edu/models/cesm1.0/cam/docs/description/cam5\\_desc.pdf](http://www.cesm.ucar.edu/models/cesm1.0/cam/docs/description/cam5_desc.pdf).]
- Pudykiewicz, J., R. Benoit, and A. Staniforth, 1985: Preliminary results from a partial LRTAP model based on an existing meteorological forecast model. *Atmos.–Ocean*, **23**, 267–303, doi:10.1080/07055900.1985.9649229.
- Putman, W. M., and S.-J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comput. Phys.*, **227**, 55–78, doi:10.1016/j.jcp.2007.07.022.
- Rančić, M., R. J. Purser, and F. Mesinger, 1996: A global shallow-water model using an expanded spherical cube: Gnomonic versus conformal coordinates. *Quart. J. Roy. Meteor. Soc.*, **122**, 959–982, doi:10.1002/qj.49712253209.
- Rood, R. B., 1987: Numerical advection algorithms and their role in atmospheric transport and chemistry models. *Rev. Geophys.*, **25**, 71–100, doi:10.1029/RG025i001p00071.
- Shashkin, V. V., and M. A. Tolstykh, 2016: Parallel implementation of the cascade mass-conserving semi-Lagrangian transport scheme. *Russ. J. Numer. Anal. Math. Modell.*, **31**, 17–28, doi:10.1515/rnam-2016-0002.
- Starr, V. P., 1945: A quasi-Lagrangian system of hydrodynamical equations. *J. Meteor.*, **2**, 227–237, doi:10.1175/1520-0469(1945)002<0227:AQLSOH>2.0.CO;2.
- St-Cyr, A., C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas, 2008: A comparison of two shallow-water models with nonconforming adaptive grids. *Mon. Wea. Rev.*, **136**, 1898–1922, doi:10.1175/2007MWR2108.1.
- Taylor, M. A., 2011: Conservation of mass and energy for the moist atmospheric primitive equations on unstructured grids. *Numerical Techniques for Global Atmospheric Models*, P. H. Lauritzen et al. Eds., Lecture Notes in Computational Science and Engineering, Vol. 80, Springer, 357–380, doi:10.1007/978-3-642-11640-7\_12.
- , and A. Fournier, 2010: A compatible and conservative spectral element method on unstructured grids. *J. Comput. Phys.*, **229**, 5879–5895, doi:10.1016/j.jcp.2010.04.008.
- , J. Tribbia, and M. Iskandarani, 1997: The spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.*, **130**, 92–108, doi:10.1006/jcph.1996.5554.
- Ullrich, P. A., P. H. Lauritzen, and C. Jablonowski, 2009: Geometrically exact conservative remapping (GECORE): Regular latitude–longitude and cubed-sphere grids. *Mon. Wea. Rev.*, **137**, 1721–1741, doi:10.1175/2008MWR2817.1.
- , T. Melvin, C. Jablonowski, and A. Staniforth, 2014: A proposed baroclinic wave test case for deep- and shallow-atmosphere dynamical cores. *Quart. J. Roy. Meteor. Soc.*, **140**, 1590–1602, doi:10.1002/qj.2241.
- Zarzycki, C. M., M. N. Levy, C. Jablonowski, J. R. Overfelt, M. A. Taylor, and P. A. Ullrich, 2014: Aquaplanet experiments using CAM’s variable-resolution dynamical core. *J. Climate*, **27**, 5481–5503, doi:10.1175/JCLI-D-14-00004.1.
- Zerroukat, M., N. Wood, and A. Staniforth, 2002: SLICE: A semi-Lagrangian inherently conserving and efficient scheme for transport problems. *Quart. J. Roy. Meteor. Soc.*, **128**, 2801–2820, doi:10.1256/qj.02.69.