

CONSERVATIVE MULTIMOMENT TRANSPORT ALONG CHARACTERISTICS FOR DISCONTINUOUS GALERKIN METHODS*

PETER A. BOSLER[†], ANDREW M. BRADLEY[†], AND MARK A. TAYLOR[†]

Abstract. A set of algorithms based on characteristic discontinuous Galerkin methods is presented for tracer transport on the sphere. The algorithms are designed to reduce message passing interface communication volume per unit of simulated time relative to current methods generally, and to the spectral element scheme employed by the U.S. Department of Energy's Exascale Earth System Model (E3SM) specifically. Two methods are developed to enforce discrete mass conservation when the transport schemes are coupled to a separate dynamics solver; constrained transport and Jacobian-combined transport. A communication-efficient method is introduced to enforce tracer consistency between the transport scheme and dynamics solver; this method also provides the transport scheme's shape preservation capability. A subset of the algorithms derived here is implemented in E3SM and shown to improve transport performance by a factor of 2.2 for the model's standard configuration with 40 tracers at the strong scaling limit of one element per core.

Key words. discontinuous Galerkin, multimoment, transport, conservative method, semi-Lagrangian

AMS subject classifications. 76M10, 86-08

DOI. 10.1137/18M1165943

1. Introduction. Numerical methods used to compute tracer transport must capture the dynamics of transport accurately and in an efficient manner. In the atmospheric components of modern climate models, for example, it is common to track 40 or more separate tracers [2, 29, 44] that represent trace gases, chemical species, suspended liquid droplets, ice crystals, or other aerosols such as pollutants and volcanic ash. Motivated by the U.S. Department of Energy's Exascale Earth System Model (E3SM) [2], our goal is to develop a computationally efficient transport scheme suitable for use with climate models running on new, heterogeneous parallel computing architectures. In these computing architectures performance is primarily limited by data movement. We therefore seek a transport scheme with improved data-locality properties relative to current schemes.

A *transport scheme* is a numerical method that solves the transport equation,

$$(1) \quad \frac{\partial(\rho q_i)}{\partial t} + \nabla \cdot (\rho q_i \mathbf{u}) = 0,$$

*Submitted to the journal's Computational Methods in Science and Engineering section January 19, 2018; accepted for publication (in revised form) May 28, 2019; published electronically August 29, 2019.

<https://doi.org/10.1137/18M1165943>

Funding: This work was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) Program and Biological and Environmental Research (BER) Program under a Scientific Discovery through Advanced Computing (SciDAC 4) BER partnership pilot project, the BER Launching an Extreme-scale ACME Prototype for Transport (LEAP-T) project, as well as a John von Neumann Postdoctoral Fellowship at Sandia National Laboratories. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by the National Technology and Engineering Solutions of Sandia, L.L.C., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. SAND NO. 2019-8793J.

[†]Center for Computing Research, Sandia National Laboratories, Albuquerque, NM 87185 (pabosle@sandia.gov, ambradl@sandia.gov, mataylo@sandia.gov).

or, equivalently, $Dq_i/Dt = \partial q_i/\partial t + \mathbf{u} \cdot \nabla q_i = 0$, where $q_i \in [0, 1]$ is the tracer mixing ratio, or concentration, of the i th tracer species; subsequently we omit i . Velocity $\mathbf{u}(\mathbf{x}, t)$ is known and $\rho > 0$ is the fluid mass density, which satisfies the continuity equation,

$$(2) \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0.$$

In this work, (1) and (2) are solved on the surface of a sphere; in other domains boundary conditions for ρ , q , and \mathbf{u} may also be necessary.

In standalone test problems, both the mass density $\rho(\mathbf{x}, t)$ and the tracer mixing ratio $q(\mathbf{x}, t)$ are subject to initial conditions, $\rho(\mathbf{x}, 0) = \rho_0(\mathbf{x})$ and $q(\mathbf{x}, 0) = q_0(\mathbf{x})$, and dynamics are simulated with known functions $\mathbf{u}(\mathbf{x}, t)$ defined by test cases relevant to global atmospheric circulation models. The test cases provide a convenient environment for error calculation and a basis for comparison with other schemes.

In practice, however, $\mathbf{u}(\mathbf{x}, t)$ and $\rho(\mathbf{x}, t)$ are numerical solutions to an underlying set of PDEs (e.g., Navier–Stokes) that include (2) and are provided by a model dynamics solver. This solver is coupled to a transport scheme, which solves (1) for mixing ratio $q(\mathbf{x}, t)$. Within the atmospheric modeling community, a dynamics solver and coupled transport scheme together are known as a model's *dynamical core*.

The numerical methods employed by the dynamics solver are not necessarily the same as the numerical methods employed by the transport scheme; in fact, computational efficiency concerns strongly motivate the use of different discretizations and time steps for the two dynamical core components. An inconsistent coupling between the two dynamical core components can cause several problems in the model solutions [39, 57]. This issue is known by several names in the literature, including “consistency with continuity” [22] and “tracer-air mass consistency” [33]. In this work we use the term *tracer consistency* to distinguish this issue from the familiar notion of consistency in numerical analysis. A transport scheme is said to be tracer consistent if its solution of (1) for the special case of $q = 1$ exactly matches the dynamics solver's solution of (2). Tracer consistency has consequences for mass conservation, since each dynamical core component may use a different discrete approximation of the continuous mass integral.

Tracer transport is an inherent component of any multiphysics fluid solver, and many methods have been developed as the field of computational fluid dynamics has matured. In the climate modeling community alone, there are traditional Eulerian transport schemes based on a discretization of (1) by finite difference methods [43], finite volume methods [35, 41], spectral element methods [38, 51], and approaches based on radial basis functions [18, 19, 20]. Equation (1) is particularly well-suited to Lagrangian formulations, and semi-Lagrangian methods have been successful [4, 30, 46, 48, 56]. Arbitrary-Lagrangian–Eulerian methods [24], Lagrangian particle methods [6, 12], contour dynamics [15], and similar hybrid [27] methods have also been proposed. In this work, we make two algorithm design choices intended to improve computational performance relative to current methods in general, and to the spectral element scheme currently employed by E3SM specifically.

Performance may be understood as the time to compute a solution, on a particular computer, at a requested accuracy. We choose a semi-Lagrangian time stepping strategy because its ability to use a long time step reduces the total amount of communication relative to a CFL-constrained Eulerian method, and its fixed mesh facilitates coupling to other model components more efficiently than Lagrangian methods. To further reduce communication we choose the compact high-order stencils of a discon-

tinuous Galerkin spatial discretization. The resulting schemes belong to a family of methods known as Lagrange–Galerkin or characteristic Galerkin methods.

Characteristic Galerkin methods combine finite elements with semi-Lagrangian time stepping and have been used since the 1980s, e.g., [1, 14, 34, 40], to solve advection-diffusion equations. They have been generalized to local adjoint methods, e.g., [9], and applied to more complex PDEs [10, 42, 47]. Their application to tracer transport on the sphere has been investigated previously in [21] using the triangular elements of spherical geodesic grids. Here, we present a transport scheme for use with unstructured grids of spherical convex polygons, with a focus in our implementation on arbitrarily-sized quadrilaterals derived from a cubed sphere mesh.

Five key transport scheme traits have become significant in recent years as atmospheric models have improved and the demands placed on them have increased: (a) conservation, (b) accuracy, (c) shape preservation, (d) tracer consistency, and (e) efficiency. Due to the long time scales involved, mass conservation (a) is particularly important in climate modeling applications. Accuracy and order of accuracy (b) are fundamental concerns of any numerical method. Shape preservation (c) provides assurances about a numerical solution’s physical realism. One or more quantities or inequalities (generally, *properties*) are required to be maintained essentially exactly (to machine precision) in an otherwise approximate numerical solution. Example properties include mass conservation, density positivity, and global or local bounds on mixing ratio extrema. The need for tracer consistency (d) arises when a dynamics solver is coupled to a separate transport scheme, as discussed above. Finally, efficiency (e) is a function of data communication, number of computations per unit of simulated time, and parallel efficiency. The algorithms presented here require fewer communication rounds and lower communication volumes per unit of simulated time than current operational methods. Our goal in this work is a transport scheme that performs well with respect to each of these five traits.

1.1. Outline. In the following sections we derive a transport scheme designed to combine large time steps with data locality. We examine the consequences of these design choices for accuracy, conservation, quasi-monotonicity, and tracer consistency. Section 2 introduces the Lagrangian flow map and the accompanying notation we employ to describe its action on points and regions over a discrete time interval.

Section 3 is the bulk of the paper and presents the derivation of our scheme. The derivation is spread over several subsections, each devoted to a fundamental component of the scheme. The overlap mesh, or *common refinement*, is introduced in subsection 3.1 and the accompanying element-local matrix formulation of the scheme is given in subsection 3.2. The close relationship between Lagrange–Galerkin methods and transport by incremental remap is examined in subsection 3.3. Semi-Lagrangian methods enable a flow-dependent MPI communication pattern, which is introduced in subsection 3.4. Subsection 3.5 defines the scheme’s isoparametric mappings; these mappings in turn define our discrete approximation of the flow map, whose consequences on the scheme’s accuracy and order of accuracy are the subject of subsection 3.6. Mass conservation in the operational setting, where the transport scheme is coupled to a separate dynamics solver, is addressed in subsection 3.7. Subsection 3.8 introduces the shape preservation filter, which also provides tracer consistency. Subsection 3.9 discusses some specific implementation details of these algorithms in code.

Section 4 presents results from a collection of well-known test cases related to atmospheric models and documents the numerical properties of the scheme. In section 5 the scheme is coupled to the full three-dimensional (3D) E3SM atmosphere model,

and its performance advantage relative to the model's current method is presented. Section 6 concludes.

2. Flow map. The Lagrangian flow map $\mathbf{x}^*(\mathbf{a}, t)$ is defined by

$$(3) \quad \frac{d\mathbf{x}^*}{dt}(\mathbf{a}, t) = \mathbf{u}(\mathbf{x}^*(\mathbf{a}, t), t)$$

and the initial condition $\mathbf{x}^*(\mathbf{a}, t_0) = \mathbf{a}$, where \mathbf{a} is a Lagrangian parameter [11]. The parameter is defined such that at a particular moment in time, $t = t_0$, the trajectories \mathbf{x}^* coincide with the Lagrangian parameter \mathbf{a} . The flow map solution $\mathbf{x}^*(\mathbf{a}, t)$ is a trajectory that advances a fluid particle from its position \mathbf{a} at time t_0 to its position \mathbf{x}^* at time t . In semi-Lagrangian schemes, the flow map is only considered in small increments of time, $t \in [t_n, t_{n+1}]$. In this work we set $t_0 = t_n$ so that

$$(4) \quad \mathbf{x}^*(\mathbf{a}, t_n) = \mathbf{a}.$$

The solution of (3) is a mapping,

$$(5) \quad \mathbf{x}^* : \mathbf{a} \mapsto \mathbf{x}^*(\mathbf{a}, t_{n+1}),$$

of a set of points at locations \mathbf{a} at $t = t_n$ to their new positions \mathbf{x}^* at $t = t_{n+1}$. The flow map (5) is invertible on the sphere S if the velocity field $\mathbf{u}(\mathbf{x}, t)$ is Lipschitz continuous in \mathbf{x} for all fixed t [37]. In this work we assume all $\mathbf{u}(\mathbf{x}, t)$ satisfy this requirement; we omit shocks. The inverse flow map is written as

$$(6) \quad \mathbf{a}^* : \mathbf{x} \mapsto \mathbf{a}^*(\mathbf{x}, t_n),$$

and denotes the mapping of a set of points at locations \mathbf{x} at $t = t_{n+1}$ to their previous positions \mathbf{a}^* at $t = t_n$. In semi-Lagrangian parlance, (5) acts forward-in-time, mapping *departure* points \mathbf{a} at $t = t_n$ along flow trajectories to corresponding *arrival* points \mathbf{x}^* at $t = t_{n+1}$, while (6) is the backward-in-time mapping of arrival points \mathbf{x} at t_{n+1} to their corresponding departure points \mathbf{a}^* at t_n .

The rest of this work follows the notational convention introduced by (5) and (6). Arbitrary physical space coordinates are denoted as either \mathbf{a} or \mathbf{x} corresponding to times t_n or t_{n+1} , respectively. Asterisks are added to emphasize trajectories defined by (3) where appropriate.

As mentioned previously, we employ a discontinuous Galerkin spatial discretization. The sphere S is discretized by a set of N_e nonoverlapping elements E_k such that $S = \cup_{k=1}^{N_e} E_k$. For later reference, we define the mesh size $\Delta\lambda$ as the average node spacing at the equator, and $h = p\Delta\lambda$ as the average element width, where p is the element degree.

We use notation from (5) and (6) to define the Lagrangian images and preimages of mesh elements. For element E_k at $t = t_n$,

$$(7) \quad E_k(t_{n+1}) = \{\mathbf{x}^*(\mathbf{a}, t_{n+1}) : \mathbf{a} \in E_k\}$$

is the Lagrangian image at $t = t_{n+1}$ of the set of points $\mathbf{a} \in E_k$. Analogously, for element E_k at $t = t_{n+1}$,

$$(8) \quad E_k(t_n) = \{\mathbf{a}^*(\mathbf{x}, t_n) : \mathbf{x} \in E_k\}$$

is the Lagrangian preimage of points $\mathbf{x} \in E_k$. Throughout this paper we have $k, l \in [1, N_e]$ index mesh elements. Indices $i, j \in [1, N_p]$ denote the i th or j th node within an element, where $N_p = (p + 1)^2$ is the number of nodes per element.

3. Characteristic Galerkin transport. Our derivation begins with a generalized statement of mass conservation that follows from the transport theorem [11].

THEOREM 3.1 (transport theorem). *For a Lagrangian volume $V(t)$,*

$$(9) \quad \frac{d}{dt} \int_{V(t)} \rho(\mathbf{x}, t) q(\mathbf{x}, t) \beta(\mathbf{x}, t) d\mathbf{x} = 0$$

for all functions $\beta(\mathbf{x}, t)$ that satisfy

$$(10) \quad \frac{\partial \beta}{\partial t} + \mathbf{u} \cdot \nabla \beta = 0.$$

Equations (9) and (10) state that the moments of ρq with respect to β are constant in time over Lagrangian volumes provided that β is transported as a constant along flow trajectories. Substituting $\beta = 1$ recovers the Lagrangian statement of mass conservation.

For notational convenience, we define tracer mass density $Q(\mathbf{x}, t) = \rho(\mathbf{x}, t)q(\mathbf{x}, t)$. From a semi-Lagrangian point of view, (9) implies that

$$(11) \quad \int_{V(t_{n+1})} Q(\mathbf{x}, t_{n+1}) \beta(\mathbf{x}, t_{n+1}) d\mathbf{x} = \int_{V(t_n)} Q(\mathbf{a}, t_n) \beta(\mathbf{a}, t_n) d\mathbf{a},$$

where the region $V(t_{n+1})$ is the Lagrangian image at $t = t_{n+1}$ of volume $V(t_n)$. The next step, common to all semi-Lagrangian schemes, is to relate the Lagrangian volume $V(t)$ to an Eulerian mesh.

We consider Eulerian mesh element E_k as a semi-Lagrangian *arrival element*, i.e., $E_k = V(t_{n+1})$ in (11). Within each element E_k there are N_p nodes with positions \mathbf{x}_{ki} , $i = 1, \dots, N_p$, each of which correspond to a nodal basis function $\eta_{ki}(\mathbf{x})$. The basis functions are used to approximate functions on S ; since we use a Gauss–Lobatto–Legendre (GLL) nodal scheme, they are also cardinal functions within each element.

APPROXIMATION 3.2. *A function f is replaced with its Galerkin expansion,*

$$(12) \quad f(\mathbf{x}, t) = \sum_{k,j} f_{kj}(t) \eta_{kj}(\mathbf{x}).$$

Using Approximation 3.2 for tracer density Q , (11) becomes

$$(13) \quad \sum_{l,j} Q_{lj}(t_{n+1}) \int_{E_k} \eta_{lj}(\mathbf{x}) \beta(\mathbf{x}, t_{n+1}) d\mathbf{x} = \sum_{l,j} Q_{lj}(t_n) \int_{E_k(t_n)} \eta_{lj}(\mathbf{a}) \beta(\mathbf{a}, t_n) d\mathbf{a}.$$

Due to the discontinuous Galerkin discretization, basis functions $\eta_{lj}(\mathbf{x}) = 0$ for all $\mathbf{x} \notin E_l$. Hence, the case $l = k$ on the left-hand side may be treated independently for each arrival element E_k . To determine the domains of integration whose integrands are smooth on the right-hand side, we require an additional tool: an overlap mesh, or *common refinement* [26], to relate arrival elements E_k to their preimages $E_k(t_n)$.

3.1. Common refinement. For each Eulerian element E_l , the flow map defines an image, $E_l(t_{n+1})$, according to (7). The images $E_l(t_{n+1})$ are flow-dependent and will generally be distorted relative to the Eulerian mesh, as illustrated in Figure 1(a). Four elements (solid edges, light gray shading, vertices \bullet) are advected in a velocity field (small gray arrows); their distorted images at t_{n+1} have dashed edges, vertices \circ , and are shaded medium gray. Vertex trajectories are indicated by large black

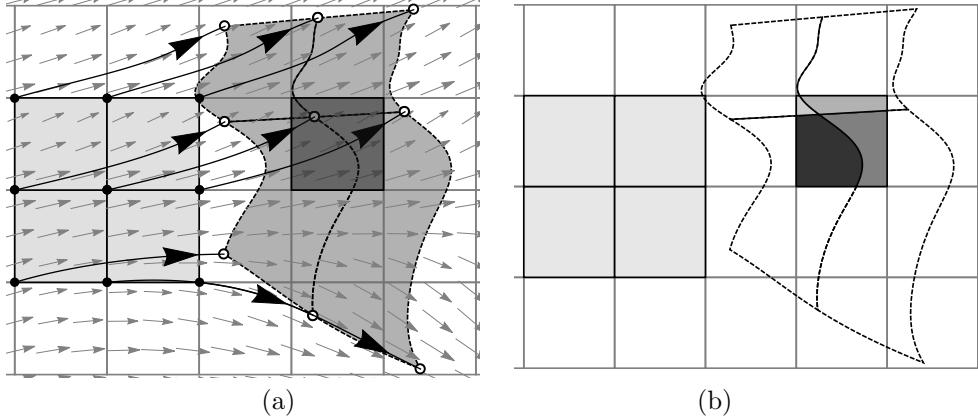


FIG. 1. Illustration scheme. (a) Eulerian elements E_l at $t = t_n$ (light gray, vertices \bullet) are advected by a velocity field (small gray arrows) to their images $E_l(t_{n+1})$ at $t = t_{n+1}$ (medium gray, vertices \circ , dashed edges). Vertex trajectories are indicated by large black arrows; an example target element is shaded dark gray. (b) The common refinement of the target element has four overlap regions V_{kl} (varied shading).

arrows. The dark gray element is highlighted for emphasis; in the following discussion it represents arrival element E_k .

The common refinement defines the intersection of each target element E_k with the distorted elements $E_l(t_{n+1})$, $l = 1, \dots, N_e$ of the advected mesh.

DEFINITION 3.3 (common refinement). *The common refinement of element E_k is a list, L_k , containing indices of elements $E_l(t_{n+1})$ that intersect E_k ,*

$$(14a) \quad L_k = \{l \in [1, N_e] : E_l(t_{n+1}) \cap E_k \neq \emptyset\}.$$

Each $l \in L_k$ is associated with an overlap region,

$$(14b) \quad V_{kl} = \{\mathbf{x} : \mathbf{x} \in E_k \text{ and } \mathbf{x} \in E_l(t_{n+1})\}.$$

The list L_k corresponding to the target element in Figure 1(a) contains the indices of the light gray elements. Overlap regions V_{kl} are shown in Figure 1(b) as filled shaded regions.

Flow map equation (8) relates the arrival element E_k to its preimage $E_k(t_n)$. Rather than solve for the preimage, we use (8) to change variables of integration in each term on the right-hand side of (13),

$$(15) \quad \int_{E_k(t_n)} \eta_{lj}(\mathbf{a}) \beta(\mathbf{a}, t_n) d\mathbf{a} = \int_{E_k} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \beta(\mathbf{a}^*(\mathbf{x}, t_n), t_n) J_x^a(\mathbf{x}) d\mathbf{x},$$

where $J_x^a(\mathbf{x}) = \left| \frac{\partial \mathbf{a}^*}{\partial \mathbf{x}}(\mathbf{x}, t_n) \right|$ is the Jacobian determinant of (6). Since basis function $\eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) = 0$ for all $\mathbf{x} \notin E_l(t_{n+1})$, the integrand on the right-hand side of (15) is nonzero only for $\mathbf{x} \in V_{kl}$. We next define β in terms of the Galerkin basis functions η_{ki} .

3.2. Mass matrices. Inspection of the left-hand side of (13) motivates a particular choice of β . Defining $\beta(\mathbf{x}, t)$ such that

$$(16) \quad \beta(\mathbf{x}, t_{n+1}) = \eta_{ki}(\mathbf{x}), \quad i = 1, \dots, N_p,$$

causes the constant Eulerian mass matrix of element E_k to appear in (13) alongside the unknown coefficient vector $Q_{ki}(t_{n+1})$. As a constant matrix, it (and its factorization) may be computed once and stored to reduce per-time step computational work.

Since β is constant along flow trajectories by (10), the choice of (16) also defines $\beta(\mathbf{a}^*(\mathbf{x}, t_n), t_n)$ in (15), as follows. We consider each $\mathbf{x} \in V_{kl}$ to be an arrival point $\mathbf{x} = \mathbf{x}^*(\mathbf{a}, t_{n+1})$ such that $\mathbf{a} \in E_l$. Substituting flow map trajectories (5) into (16) we get

$$(17) \quad \eta_{ki}(\mathbf{x}) = \beta(\mathbf{x}^*(\mathbf{a}, t_{n+1}), t_{n+1}).$$

Following the flow map trajectory backwards to $t = t_n$, we have

$$(18) \quad \beta(\mathbf{x}^*(\mathbf{a}, t_{n+1}), t_{n+1}) = \beta(\mathbf{x}^*(\mathbf{a}, t_n), t_n)$$

as a consequence of (10). The initial condition for the flow map (4) asserts that $\mathbf{x}^*(\mathbf{a}, t_n) = \mathbf{a}$; hence,

$$(19) \quad \beta(\mathbf{x}^*(\mathbf{a}, t_n), t_n) = \beta(\mathbf{a}^*(\mathbf{x}, t_n), t_n).$$

Within each overlap region V_{kl} we combine (17), (18), and (19) to get

$$(20) \quad \beta(\mathbf{a}^*(\mathbf{x}, t_n), t_n) = \eta_{ki}(\mathbf{x}) \quad \forall \mathbf{x} \in V_{kl}.$$

With (15), (16), and (20), the scheme (13) becomes

$$(21) \quad \sum_{j=1}^{N_p} Q_{kj}(t_{n+1}) \int_{E_k} \eta_{kj}(\mathbf{x}) \eta_{ki}(\mathbf{x}) d\mathbf{x} \\ = \sum_{l \in L_k} \left(\sum_{j=1}^{N_p} Q_{lj}(t_n) \int_{V_{kl}} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \eta_{ki}(\mathbf{x}) J_x^a(\mathbf{x}) d\mathbf{x} \right).$$

Equation (21) may be viewed as an element-local matrix equation,

$$(22) \quad A_k \mathbf{Q}_k(t_{n+1}) = \sum_{l \in L_k} B_{kl} \mathbf{Q}_l(t_n),$$

where $\mathbf{Q}_k(t) = [Q_{ki}(t)]^T$ and $\mathbf{Q}_l(t) = [Q_{lj}(t)]^T$ for $i, j = 1, \dots, N_p$. Matrix A_k is the symmetric $N_p \times N_p$ mass matrix associated with element E_k . Its entries are

$$(23) \quad (A_k)_{ij} = \int_{E_k} \eta_{kj}(\mathbf{x}) \eta_{ki}(\mathbf{x}) d\mathbf{x}.$$

Borrowing terminology from [17], B_{kl} is a *mixed mass matrix* that corresponds to an interaction between arrival element E_k and distorted element $E_l(t_{n+1})$. Its entries are given by

$$(24) \quad (B_{kl})_{ij} = \int_{V_{kl}} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \eta_{ki}(\mathbf{x}) J_x^a(\mathbf{x}) d\mathbf{x}.$$

Numerical consistency (not to be confused with tracer consistency) requires that the scheme (22) reduce to the identity in the limit $\Delta t \rightarrow 0$, where $\Delta t = t_{n+1} - t_n$. While the mass matrix A_k is constant, the mixed mass matrix B_{kl} is dependent upon the time step Δt via the common refinement and the flow map. Hence, if

$$(25) \quad \lim_{\Delta t \rightarrow 0} B_{kl} = \begin{cases} A_k, & l = k, \\ 0, & l \neq k, \end{cases}$$

then the scheme is consistent. B_{kl} is necessarily dense, which implies that consistency requires the full (not lumped) mass matrix.

Morton et. al. [37] prove that numerical stability is independent of $|\mathbf{u}|$ for CDG schemes, including the present work. Since the scheme is linear, convergence follows from the Lax equivalence theorem. As is desirable, the time step may be determined by accuracy considerations rather than stability.

3.3. Transport by incremental remap. The close relationship between transport schemes and remapping algorithms is well known, e.g., [5, 12, 16]. We now show that an incremental remap (IR) algorithm combined with a discontinuous Galerkin discretization is nearly equivalent to characteristic Galerkin transport.

Incremental remap is a two-step algorithm. First, the tracer density Q and the GLL mesh are transported by the flow map for one time step. Second, the transported densities are remapped from the advected mesh to the Eulerian mesh. In Figure 1(a), forward-in-time advection is illustrated by the distorted gray shaded region. Following this transport step, the solution at each mesh node is updated by remapping the distorted mesh data to the constant Eulerian mesh. Both meshes in the remapping problem exist at $t = t_{n+1}$; the distorted mesh is the data *source* and the Eulerian mesh is the *target*.

Since we are advecting tracer densities rather than tracer mixing ratios, the data change during transport. On both meshes there exists a Galerkin expansion of $Q(\mathbf{x}, t)$. On the Eulerian mesh, we have, from (12),

$$(26) \quad Q(\mathbf{a}, t_n) = \sum_{l,j} Q_{lj}(t_n) \eta_{lj}(\mathbf{a}),$$

and on the advected mesh we have

$$(27) \quad Q(\mathbf{x}, t_{n+1}) = \sum_{l,j} \tilde{Q}_{lj}(t_{n+1}) \xi_{lj}(\mathbf{x}),$$

where $\xi_{lj}(\mathbf{x})$ are the set of discontinuous basis functions on distorted mesh elements $E_l(t_{n+1})$ and $\tilde{Q}_{lj}(t)$ are the associated coefficients.

Conservation of mass can be achieved if, for each basis function ξ_{lj} , $l = 1, \dots, N_e$ and $j = 1, \dots, N_p$,

$$(28) \quad \tilde{Q}_{lj}(t_{n+1}) \int_{E_l(t_{n+1})} \xi_{lj}(\mathbf{x}) d\mathbf{x} = Q_{lj}(t_n) \int_{E_l} \eta_{lj}(\mathbf{a}) d\mathbf{a}.$$

Using the flow map, we have

$$(29) \quad \int_{E_l(t_{n+1})} \xi_{lj}(\mathbf{x}) d\mathbf{x} = \int_{E_l} \xi_{lj}(\mathbf{x}^*(\mathbf{a}, t_{n+1})) J_a^x(\mathbf{a}) d\mathbf{a}.$$

Hence, defining ξ_{lj} such that

$$(30) \quad \xi_{lj}(\mathbf{x}) = \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \quad \forall \mathbf{x} \in E_l(t_{n+1})$$

satisfies (28). Using (30) in (28), we derive an expression for $\tilde{Q}_{lj}(t_{n+1})$,

$$(31) \quad \tilde{Q}_{lj}(t_{n+1}) = Q_{lj}(t_n) \frac{\int_{E_l} \eta_{lj}(\mathbf{a}) d\mathbf{a}}{\int_{E_l(t_{n+1})} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) d\mathbf{x}}.$$

We see that as nodal values $Q_{lj}(t)$ are advected by the flow, their value changes by the multiplicative factor

$$(32) \quad R_{lj}^n = \frac{\int_{E_l} \eta_{lj}(\mathbf{a}) d\mathbf{a}}{\int_{E_l(t_{n+1})} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) d\mathbf{x}},$$

the *density factor* for basis function η_{lj} . The density factor accounts for the change in mass due to velocity divergence; $R_{lj}^n = 1$ if $\nabla \cdot \mathbf{u} = 0$. Density factors may be computed independently and reused for every tracer.

With transport complete, the remap problem sets up the equations that project $Q(\mathbf{x}, t_{n+1})$ as represented on the distorted mesh by the Galerkin expansion (27) onto the basis functions of the Eulerian mesh (26). It may be written as a standard L_2 projection (e.g., [17, 26, 28]) from one set of basis functions to another,

$$(33) \quad \sum_{j=1}^{N_p} Q_{kj}(t_{n+1}) \int_{E_k} \eta_{kj}(\mathbf{x}) \eta_{ki}(\mathbf{x}) d\mathbf{x} \\ = \sum_{l \in L_k} \left(\sum_{j=1}^{N_p} Q_{lj}(t_n) R_{lj}^n \int_{V_{kl}} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \eta_{ki}(\mathbf{x}) d\mathbf{x} \right).$$

This equation can be represented in matrix form, similarly to (22), as

$$(34) \quad A_k \mathbf{Q}_k(t_{n+1}) = \sum_{l \in L_k} B_{kl} R_l \mathbf{Q}_l(t_n),$$

where the mass matrix A_k and solution vector $\mathbf{Q}_k(t_{n+1})$ are identical to their definitions in the previous section. Mixed mass matrix B_{kl} has entries defined by

$$(35) \quad (B_{kl})_{ij} = \int_{V_{kl}} \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \eta_{ki}(\mathbf{x}) d\mathbf{x},$$

and $R_l = \text{diag}(R_{lj}^n)$ is the diagonal matrix of density factors (32).

In both schemes the term $\eta_{lj}(\mathbf{a}^*(\mathbf{x}, t))$ appears; it represents a *transported basis function*, which is a common feature of characteristic Galerkin methods. The mass matrices A_k and common refinement are identical between the two schemes; the only differences lie in the entries of the mixed mass matrix. Comparing (35) with (24), we note that the Jacobian determinant $J_x^a(\mathbf{x})$ does not appear in (35); its effects are now accounted for by the density factors (32). This is indicative of the primary difference between CDG and IR. With CDG, transport occurs coincidentally with the definition of the linear system. In IR, transport occurs prior to the definition of the linear system. Taylor series expansions of (21) and (33) about a point $\mathbf{x}_0 \in V_{kl} \subseteq E_l(t_{n+1})$ reveal a difference of $O(h^3)$ between the two schemes as mesh element width $h \rightarrow 0$.

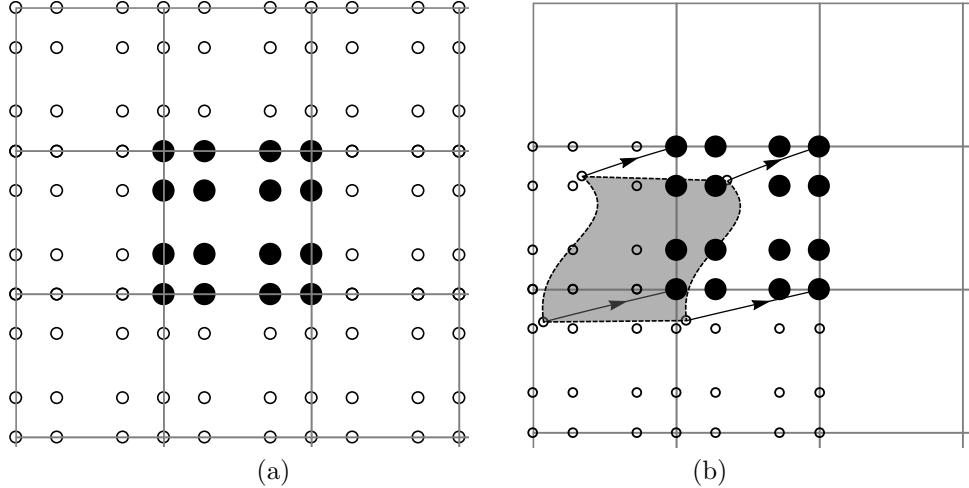


FIG. 2. *MPI communication patterns. Target element (center, nodes •) receives data from its neighbors (nodes ○); (a) full halo exchange, (b) upwind scheme.*

3.4. MPI communication patterns. In a distributed computing environment the data required by the target element to construct its mixed mass matrix and the vector $\mathbf{Q}_l(t_n)$ may not be available—it may reside on other processors. These data must be transferred to the target element, typically using a parallel MPI.

Figure 2(a) shows a typical communication pattern for a case at the strong scaling limit of 1 element per core. The target element in the center of the illustration, with mesh nodes shown as filled circles, receives data from each of its neighbors, whose nodes are depicted as open circles. The adjacent neighbors are commonly referred to as the target element’s *halo*. The figure depicts a 1-element halo; if the time step is large relative to the mesh size as in Figure 1(a), larger halos are required. For simplicity, in this section we assume the time step is restricted to ensure the target element’s domain of dependence lies entirely within a 1-element halo.

In our application each node represents a vertical column of the model atmosphere, and each column is discretized by a user-defined number of levels. In E3SM version 1, the default is 72 levels per column. The target element requires data from all levels of each column from each element in its domain of dependence.

Full halo exchanges are independent of the flow and are straightforward to implement; however, they often result in more communication than necessary. For each target element E_k , the list L_k (14) defines the discrete domain of dependence and may contain fewer elements than the halo. A typical case is shown in Figure 2(b).

In a typical implementation of a full halo exchange, each element of the halo independently sends all data to the target element. For the case shown in Figure 2(a) with eight elements in the halo, each with 16 nodes, this requires sending the data for $8 \times 16 = 128$ columns. An optimal implementation would only send each column once, 84 columns in this case.

Semi-Lagrangian schemes can benefit from *upwind* communication patterns. For the case illustrated by Figure 2(b), the target element requires data from only four elements, or 48 columns if each of those elements sends all data, 33 columns in an optimal implementation. Compared to the full halo exchange, an upwind communication scheme requires an additional step for the target element to communicate a Boolean

value to each element in the halo—true if that element is included in L_k , false if not. After this “handshake” step, only the elements in the domain of dependence of the target send their data. The cost of this handshake step is negligible relative to the cost of sending the column data. For the case illustrated by Figure 2, which is typical for our applications, the upwind communication pattern requires approximately 40% of the data movement required by the full halo exchange.

In practice, the advantage of an upwind scheme over a full halo exchange is dependent on the flow, computer architecture, MPI distribution, and number of tracers. Preliminary work has focused on traditional, interpolation-based semi-Lagrangian methods, for which the communication volume can be reduced by an even larger factor. End-to-end speedup of the transport component using the upwind pattern is observed to be approximately $2\times$ at 40 tracers relative to the full halo exchange.

3.5. Reference element maps. Details of the discretizations of both (21) and (33) depend on a set of invertible maps from a mathematical reference space to physical space. Each element has a local mapping to and from a quadrilateral reference element R_Q .

DEFINITION 3.4 (reference element). *Quadrilateral reference element R_Q is defined as*

$$R_Q = \{\mathbf{r} = (r_1, r_2) : -1 \leq r_1, r_2 \leq 1\}$$

and contains GLL quadrature points \mathbf{r}_i , with corresponding quadrature weights w_i , $i = 1, \dots, N_p$.

The bijective map from R_Q to element E_k is denoted by $m_{R_Q}^{E_k}(\mathbf{r})$. In this notation, maps are denoted by the symbol m with a subscript corresponding to the domain and superscript corresponding to the range. Hence the inverse of $m_{R_Q}^{E_k}(\mathbf{r})$ is written as $m_{E_k}^{R_Q}(\mathbf{x})$. The maps’ Jacobian determinants follow the same convention: the Jacobian determinant of map $m_{R_Q}^{E_k}(\mathbf{r})$ is $J_{R_Q}^{E_k}(\mathbf{r})$.

Reference quadrature points \mathbf{r}_i are mapped by $m_{R_Q}^{E_k}(\mathbf{r})$ to define Eulerian mesh nodes \mathbf{x}_{ki} such that

$$(36) \quad \mathbf{x}_{ki} = m_{R_Q}^{E_k}(\mathbf{r}_i).$$

Each quadrature point \mathbf{r}_i is associated with a GLL basis function $\phi_i(\mathbf{r})$ in the reference space. The mapping also relates the reference element basis ϕ_i to the spherical basis η_{ki} ,

$$(37) \quad \eta_{ki}(\mathbf{x}) = \phi_i(m_{E_k}^{R_Q}(\mathbf{x})) \quad \forall \mathbf{x} \in E_k \quad \text{and} \quad \phi_i(\mathbf{r}) = \eta_{ki}(m_{R_Q}^{E_k}(\mathbf{r})) \quad \forall \mathbf{r} \in R_Q.$$

The map employed by E3SM’s atmosphere model consists of two steps [23, Appendix A]. First, $\tilde{m}_{R_Q}^{E_k}(\mathbf{r})$ maps the corners of R_Q to the sphere. Using the vertices $\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C$, and \mathbf{x}_D of quadrilateral element $E_k \subset S$,

$$\begin{aligned} \tilde{m}_{R_Q}^{E_k}(\mathbf{r}) = \frac{1}{4} & \left((1 - r_1)(1 - r_2)\mathbf{x}_A + (1 + r_1)(1 - r_2)\mathbf{x}_B \right. \\ & \left. + (1 + r_1)(1 + r_2)\mathbf{x}_C + (1 - r_1)(1 + r_2)\mathbf{x}_D \right). \end{aligned}$$

The mapped vertices define a bilinear surface in \mathbb{R}^3 , circumscribed by S , whose corners are coincident with the corners of element E_k . The second step radially projects the

surface to the sphere,

$$(38) \quad m_{R_Q}^{E_k}(\mathbf{r}) = \frac{\tilde{m}_{R_Q}^{E_k}(\mathbf{r})}{\|\tilde{m}_{R_Q}^{E_k}(\mathbf{r})\|}.$$

While the map $m_{R_Q}^{E_k}(\mathbf{r})$ and its Jacobian determinant $J_{R_Q}^{E_k}(\mathbf{r})$ are known analytically, the inverse map $m_{E_k}^{R_Q}(\mathbf{x})$ must be solved numerically by a nonlinear solver, such as an iterative Newton's method. Such nonlinear solves can be computationally expensive; we return to this issue in section 4.

3.5.1. Flow map approximation. The flow map (3) defines pointwise trajectories, but the common refinement, upon which our method is based, is defined over regions. In this section we describe the construction of the local polygonal approximation of the flow map that we use to build the common refinement.

Distorted elements $E_l(t_{n+1})$ are approximated as spherical polygons $P_l(t_{n+1})$ in two steps. First, vertices \mathbf{a}_I , $I = 1, \dots, N_I$, of element E_l are advected forward in time by solving (3) to find arrival points $\mathbf{x}^*(\mathbf{a}_I, t_{n+1})$. The number of advected vertices N_I is determined by the polynomial degree of the edge reconstruction: for linear edges, $N_I = 4$ (two points per edge); for quadratic edges, $N_I = 8$ (three points per edge), etc. We employ a linear edge reconstruction in this work. Vertex advection is illustrated by the large black arrows in Figure 1(a). Second, the vertices' arrival points are connected to form the quadrilateral $P_l(t_{n+1})$. The flow map is known only pointwise, at the vertices where (3) has been solved. However, the maps $m_{R_Q}^{E_l}$ and $m_{R_Q}^{P_l(t_{n+1})}$ are known for all $\mathbf{r} \in R_Q$ after the advection step. We use these maps to approximate the action of the flow map on each element.

APPROXIMATION 3.5 (flow map). *The action of the flow map on element E_l over the interval $t \in [t_n, t_{n+1}]$,*

$$(39a) \quad \mathbf{a} \mapsto \mathbf{x}^*(\mathbf{a}, t_{n+1}) \quad \forall \mathbf{a} \in E_l,$$

is approximated by reference element maps $m_{R_Q}^{E_l}(\mathbf{r})$ and $m_{R_Q}^{P_l(t_{n+1})}(\mathbf{r})$ as

$$(39b) \quad \mathbf{x}^*(\mathbf{a}, t_{n+1}) = m_{R_Q}^{P_l(t_{n+1})}(m_{E_l}^{R_Q}(\mathbf{a})) \quad \forall \mathbf{a} \in E_l.$$

Similarly, the inverse flow map is approximated as

$$(39c) \quad \mathbf{a}^*(\mathbf{x}, t_n) = m_{R_Q}^{E_l}(m_{P_l(t_{n+1})}^{R_Q}(\mathbf{x})) \quad \forall \mathbf{x} \in P_l(t_{n+1}).$$

The Jacobian determinant of the inverse flow map, $J_x^a(\mathbf{x})$, is approximated by the Jacobian determinant of (39c).

The set of quadrilaterals $P_l(t_{n+1})$ is overlayed on the Eulerian mesh to define the common refinement. The resulting overlap regions V_{kl} are spherical polygons with between three and eight edges inclusive.

Figure 3(a) provides an illustration. Arrival element E_k is shaded light gray. The vertices of element E_l have been moved forward in time, and distorted element $E_l(t_{n+1})$ is approximated by spherical quadrilateral $P_l(t_{n+1})$, shaded dark gray. The common refinement locates the intersection points (\bullet) of E_k with $P_l(t_{n+1})$. Intersection points are connected by great-circle arcs to form the spherical polygon V_{kl} (medium gray, with hatching).

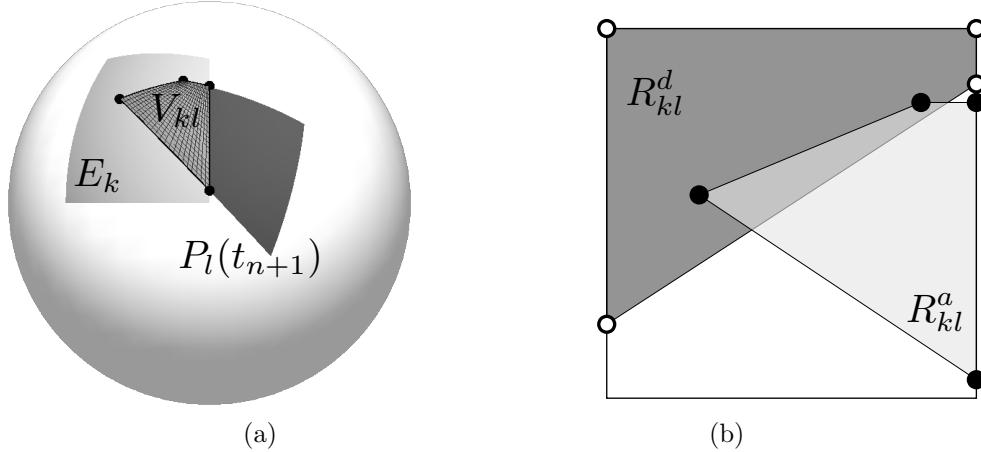


FIG. 3. *Polygonal overlap regions.* Physical space (a): spherical quadrilateral $P_l(t_{n+1})$ (dark gray) intersects target element E_k (light gray) to form spherical polygon V_{kl} (medium gray, hatched) with vertices \bullet connected by great circle arcs. Reference space (b): overlap region V_{kl} maps to departure region $R_{kl}^d \subseteq R_Q$ (dark gray, vertices \circ) and arrival region $R_{kl}^a \subseteq R_Q$ (light gray, vertices \bullet).

3.6. Accuracy. For both the CDG and IR schemes, (21) and (33), respectively, accuracy depends on (a) the computed trajectories (3), (b) the polynomial degree p of the Galerkin representation (12), (c) the choice of numerical quadrature scheme employed to evaluate (23) and either (24) (CDG) or (35) (IR), and (d) the geometric reconstruction used for Approximation 3.5.

We find that the scheme is insensitive to the time discretization error (a) in the trajectory approximation (3); the authors of [42] note a similar finding. The degree p of the Galerkin basis (b) is the user's choice; E3SM defaults to $p = 3$. We must use a quadrature scheme (c) of sufficiently high order to capture the nonzero off-diagonal entries of the mass matrix (23). Conservation, discussed in subsection 3.7, motivates a quadrature scheme that is “exact” with respect to machine precision.

The remaining source of error is (d), the geometric reconstruction of Approximation 3.5, which depends on the degree \tilde{p} of the distorted quadrilaterals $P_l(t_{n+1})$. Standard results from the analysis of isoparametric finite element methods [13, 49] show that this approximation adds error at the $O(h^{\tilde{p}+1})$ level to the scheme as $h \rightarrow 0$. In this work polygons $P_l(t_{n+1})$ in Approximation 3.5 have edges defined by great circles and $\tilde{p} = 1$. Our implementation is, therefore, second order accurate overall in general flow regimes, which is sufficient for E3SM.

Higher orders of accuracy are possible, but require higher order geometric reconstructions of the deformed elements' edges [3, 45]. The authors of [54] use quadratic edges to build a second order edge reconstruction on the sphere, which yields third order accuracy for their semi-Lagrangian flux form transport scheme, CSLAM. Motivated by their result, we have added the capability to use quadratic edge reconstructions to our common refinement software; however, we have not yet integrated this capability into a full transport scheme. Instead we demonstrate the capability for our schemes to achieve high order accuracy in section 4 using a solid body rotation test case where the linear edge reconstruction is exact.

3.7. Mass conservation. While transport schemes (22) and (34) were derived from a statement of mass conservation, discrete conservation is not assured—it depends on the definition of the discrete mass integral. Even though exact quadrature is impossible due to the nonlinear mapping (38), a sufficiently high order quadrature scheme can achieve arbitrarily high precision and conserve the total *spherical* mass,

$$(40) \quad M_S(t) = \sum_{k=1}^{N_e} \sum_{i=1}^{N_p} Q_{ki}(t) \int_{E_k} \eta_{ki}(\mathbf{x}) d\mathbf{x},$$

to machine precision. Using a high order triangular quadrature rule, our scheme is conservative with respect to this definition of mass. For a standalone transport scheme, this is sufficient.

For coupled transport schemes, however, the definition of mass must be consistent with the dynamics solver. In our case, E3SM dynamics are solved with a nodal GLL spectral element method [51]; hence, the mass we should conserve is the GLL *spectral approximation* of the spherical mass integral,

$$(41) \quad M_p(t) = \sum_{k=1}^{N_e} \sum_{i=1}^{N_p} Q_{ki}(t) J_{R_Q}^{E_k}(\mathbf{r}_i) w_i,$$

where \mathbf{r}_i and w_i are defined on R_Q by Definition 3.4.

To achieve conservation of $M_p(t)$ we must make adjustments to our scheme. We have developed two strategies. In the first we augment the linear system, either (22) or (34), with a set of constraints. The resulting linear equality constrained least-squares problem is consistent with (41). We refer to this as *constrained transport*.

The second strategy results from considering the source of this mass inconsistency, i.e., the fact that $M_S(t) \neq M_p(t)$, which is fundamentally due to the truncation error discrepancy between (40) and (41). If the two truncation errors could be matched exactly, the scheme would be conservative.

Matching truncation error can be achieved by defining a new variable that combines the nonlinear Jacobian determinant of the spherical quadrilateral map (38) and tracer mass density. The scheme is then formulated within a flat reference space, where standard Gaussian quadrature rules are able to exactly integrate polynomials. We refer to it as *Jacobian-combined transport*, or JCT. Each conservation strategy is described below in more detail.

3.7.1. Constrained transport. Tracer mass within element E_k is defined as $\int_{E_k} Q(\mathbf{x}, t) d\mathbf{x}$. Its discrete representation is inherited from the GLL discretization, $M_k(t) = \sum_{i=1}^{N_p} Q_{ki}(t) J_{R_Q}^{E_k}(\mathbf{r}_i) w_i$. Previously, in each element we solved either (21) for CDG or (33) for IR,

$$(42) \quad A_k \mathbf{Q}_k(t_{n+1}) = \sum_{l \in L_k} B_{kl} R_l \mathbf{Q}_l(t_n),$$

where $R_l = I$ for CDG. This system provides conservation of $M_S(t)$. Equation (42) can be interpreted as the first order optimality condition to the linear least squares problem

$$(43) \quad \min_{\mathbf{Q}_k(t_{n+1})} \left[\frac{1}{2} \mathbf{Q}_k(t_{n+1})^T A_k \mathbf{Q}_k(t_{n+1}) - \mathbf{Q}_k^T(t_{n+1}) \sum_{l \in L_k} B_{kl} R_l \mathbf{Q}_l(t_n) \right].$$

Now we augment this objective with the linear equality constraint

$$(44) \quad \sum_{i=1}^{N_p} Q_{ki}(t_{n+1}) J_{R_Q}^{E_k}(\mathbf{r}_i) w_i = \sum_{l=1}^{N_e} \sum_{j=1}^{N_p} \frac{\sum_{i=1}^{N_p} (B_{kl})_{ij}}{\sum_{m=1}^{N_e} \sum_{i=1}^{N_p} (B_{ml})_{ij}} Q_{lj}(t_n) J_{R_Q}^{E_k}(\mathbf{r}_j) w_j,$$

or, briefly,

$$(45) \quad \mathbf{c}^T \mathbf{Q}_k(t_{n+1}) = d.$$

Summing each side over k yields the global conservation of $M_p(t)$. The Karush–Kuhn–Tucker (KKT) system for this linear-equality-constrained least-squares problem is

$$(46) \quad \begin{pmatrix} A_k & \mathbf{c} \\ \mathbf{c}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Q}_k(t_{n+1}) \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \sum_{l \in L_k} B_{kl} R_l \mathbf{Q}_l(t_n) \\ d \end{pmatrix}.$$

Its solution requires only a small (and, in practice, negligible) additional amount of computation relative to the original scheme (42). We show in section 4 that the unconstrained schemes recover the optimal order of accuracy, $p+1$, for the solid-body rotation test case. Numerical evidence shows that order of accuracy reduces to p when the constraint (44) is added; hence, we expect that constrained transport is capable of high order accuracy if combined with a high order geometric reconstruction.

3.7.2. JCT. The discrete tracer mass $M_k(t)$ in element E_k is derived by rewriting element E_k 's continuous mass integral over reference element R_Q and applying the GLL quadrature rule. It follows from the properties of GLL quadrature that, while $M_k(t)$ is not an exact representation of the continuous mass integral, it is an exact representation of the related integral

$$(47) \quad M_k(t) = \int_{R_Q} I_p \left[Q(m_{R_Q}^{E_k}(\mathbf{r}), t) J_{R_Q}^{E_k}(\mathbf{r}) \right] d\mathbf{r},$$

where $I_p[f(\mathbf{r})]$ is the degree p interpolating polynomial of f on R_Q .

Recalling the GLL quadrature points \mathbf{r}_i from Definition 3.4, the interpolating polynomial's coefficients are the nodal values $Q(m_{R_Q}^{E_k}(\mathbf{r}_i), t) J_{R_Q}^{E_k}(\mathbf{r}_i)$,

$$(48) \quad I_p \left[Q(m_{R_Q}^{E_k}(\mathbf{r}), t) J_{R_Q}^{E_k}(\mathbf{r}) \right] = \sum_{j=1}^{N_p} Q(m_{R_Q}^{E_k}(\mathbf{r}_j), t) J_{R_Q}^{E_k}(\mathbf{r}_j) \phi_j(\mathbf{r}).$$

This polynomial may alternatively be viewed as the nodal Galerkin expansion on R_Q of the product $\bar{Q}_k(\mathbf{r}, t) = Q(m_{R_Q}^{E_k}(\mathbf{r}), t) J_{R_Q}^{E_k}(\mathbf{r})$, which combines tracer mass and the local spherical map's Jacobian determinant. We refer to $\bar{Q}_k(\mathbf{r}, t)$ as the *combined variable*. We define time-dependent coefficient functions $\bar{Q}_{ki}(t) = Q(m_{R_Q}^{E_k}(\mathbf{r}_i), t) J_{R_Q}^{E_k}(\mathbf{r}_i)$ and replace $\bar{Q}_k(\mathbf{r}, t)$ with its interpolating polynomial so that

$$(49) \quad \bar{Q}_k(\mathbf{r}, t) = \sum_{i=1}^{N_p} \bar{Q}_{ki}(t) \phi_i(\mathbf{r}).$$

This polynomial on R_Q can be integrated exactly by standard quadrature rules. With JCT, we preserve this property by using only linear maps. The resulting mass matrix and mixed mass matrix integrands are polynomial, allowing exact integration.

The scheme will automatically be conservative because the exact computation of (47) matches the truncation error in the spectral approximation of $\int_{E_k} Q(\mathbf{x}, t) d\mathbf{x}$. Summing over index k in (47) recovers (41).

To evaluate mass matrix entries $(A_k)_{ij}$ in a manner that preserves (47), we write the left-hand side of (21) over reference element R_Q ,

$$(50) \quad \begin{aligned} \int_{E_k} \sum_{j=1}^{N_p} Q_{kj}(t_{n+1}) \eta_{kj}(\mathbf{x}) \eta_{ki}(\mathbf{x}) d\mathbf{x} \\ = \int_{R_Q} \left(\sum_{j=1}^{N_p} Q_{kj}(t_{n+1}) J_{R_Q}^{E_k}(\mathbf{r}) \phi_j(\mathbf{r}) \right) \phi_i(\mathbf{r}) d\mathbf{r}. \end{aligned}$$

If the terms inside the parentheses are replaced with their interpolating polynomial (49), we arrive at an integral that can be computed exactly. This substitution defines the mass matrices A_k for JCT, whose entries are

$$(51) \quad (A_k)_{ij} = \int_{R_Q} \phi_j(\mathbf{r}) \phi_i(\mathbf{r}) d\mathbf{r}.$$

To evaluate the mixed mass matrix, we consider both the flow map and the overlap region from the point of view of reference element R_Q .

Overlap region V_{kl} corresponds to two regions of reference space: $R_{kl}^d \subseteq R_Q$ is the subset of R_Q that maps to departure region $\{\mathbf{a} \in E_l : \mathbf{x}^*(\mathbf{a}, t_{n+1}) \in E_k\}$, and $R_{kl}^a \subseteq R_Q$ is the subset of R_Q that maps to arrival region $\{\mathbf{x} \in E_k : \mathbf{a}^*(\mathbf{x}, t_n) \in E_l\}$. The action of the flow map for overlap region V_{kl} over the interval $t \in [t_n, t_{n+1}]$ may therefore be viewed as a mapping within reference space from R_{kl}^d to R_{kl}^a , denoted by $m_d^a(\mathbf{r})$. The flow map takes the departure region into the arrival region; each $\mathbf{r} \in R_{kl}^d$ corresponds to a point $m_d^a(\mathbf{r}) \in R_{kl}^a$, where

$$(52) \quad m_d^a(\mathbf{r}) = m_{E_k}^{R_Q}(m_{R_Q}^{P_l(t_{n+1})}(\mathbf{r})).$$

The inverse flow map takes the arrival region into the departure region; each $\mathbf{r} \in R_{kl}^a$ corresponds to a point $m_a^d(\mathbf{r}) \in R_{kl}^d$ such that

$$(53) \quad m_a^d(\mathbf{r}) = m_{P_l(t_{n+1})}^{R_Q}(m_{R_Q}^{E_k}(\mathbf{r})).$$

From the reference element's perspective, (39b) and (39c) are represented by (52) and (53), respectively.

CLAIM 3.6 (inverse flow map Jacobian determinant). *Within arrival region $R_{kl}^a \subseteq R_Q$, the Jacobian determinant of the inverse flow map is given by*

$$(54) \quad J_x^a(m_{R_Q}^{E_k}(\mathbf{r})) = \frac{J_a^d(\mathbf{r}) J_{R_Q}^{E_l}(m_a^d(\mathbf{r}))}{J_{R_Q}^{E_k}(\mathbf{r})},$$

where $J_a^d(\mathbf{r})$ is the Jacobian determinant of (53).

Proof. By definition,

$$(55) \quad \int_{V_{kl}} J_x^a(\mathbf{x}) d\mathbf{x} = \int_{V_{kl}(t_n)} d\mathbf{a},$$

where $V_{kl}(t_n)$ is the Lagrangian preimage of V_{kl} at $t = t_n$. Writing the left-hand side over R_{kl}^a and the right-hand side over R_{kl}^d , we get

$$(56) \quad \int_{R_{kl}^a} J_x^a(m_{R_Q}^{E_k}(\mathbf{r})) J_{R_Q}^{E_k}(\mathbf{r}) d\mathbf{r} = \int_{R_{kl}^d} J_{R_Q}^{E_l}(\mathbf{r}) d\mathbf{r}.$$

Using (53) we change variables on the right-hand side to write both integrals over the arrival region R_{kl}^a . Combining them, we have

$$(57) \quad \int_{R_{kl}^a} J_x^a(m_{R_Q}^{E_k}(\mathbf{r})) J_{R_Q}^{E_k}(\mathbf{r}) - J_{R_Q}^{E_l}(m_a^d(\mathbf{r})) J_a^d(\mathbf{r}) d\mathbf{r} = 0.$$

We conclude that the integrand must vanish and the result follows. \square

With (53) and Claim 3.6, the right-hand side of (21) may now be written over $R_{kl}^a \subseteq R_Q$,

$$(58) \quad \sum_{l \in L_k} \int_{V_{kl}} \sum_{j=1}^{N_p} Q_{lj}(t_n) \eta_{lj}(\mathbf{a}^*(\mathbf{x}, t_n)) \eta_{ki}(\mathbf{x}) J_x^a(\mathbf{x}) d\mathbf{x} \\ = \sum_{l \in L_k} \int_{R_{kl}^a} \left(\sum_{j=1}^{N_p} Q_{lj}(t_n) J_{R_Q}^{E_l}(m_a^d(\mathbf{r})) \phi_j(m_a^d(\mathbf{r})) \right) \phi_i(\mathbf{r}) J_a^d(\mathbf{r}) d\mathbf{r},$$

where we have reordered the terms in the integrand so that the terms inside the parentheses may be approximated by (49), the interpolating polynomial $\bar{Q}_l(m_a^d(\mathbf{r}), t_n)$.

Mappings (52) and (53) both use (38); hence, they are still nonpolynomial. To ensure that the right-hand side of (58) can be integrated exactly with standard quadrature rules, JCT approximates nonlinear maps on S with piecewise linear maps in R_Q ; as a consequence we cannot expect more than second order accuracy.

APPROXIMATION 3.7 (flow map approximation, reference element). *Given vertices $\{\mathbf{x}_{vI}\} \in S$ of overlap region V_{kl} , departure and arrival subsets R_{kl}^d and R_{kl}^a are approximated as polygons in R_Q . Region R_{kl}^d is approximated by defining vertices*

$$(59) \quad \mathbf{r}_{vI}^d = m_{P_l(t_{n+1})}^{R_Q}(\mathbf{x}_{vI})$$

and connecting them with straight lines in R_Q . Region R_{kl}^a is approximated by defining vertices

$$(60) \quad \mathbf{r}_{vI}^a = m_{E_k}^{R_Q}(\mathbf{x}_{vI})$$

and connecting them with straight lines in R_Q . The polygons are triangulated to build piecewise linear maps

$$(61) \quad \tilde{m}_d^a(\mathbf{r}) : R_{kl}^d \mapsto R_{kl}^a \quad \text{and} \quad \tilde{m}_a^d(\mathbf{r}) : R_{kl}^a \mapsto R_{kl}^d,$$

with piecewise constant Jacobian determinants $\tilde{J}_d^a(\mathbf{r})$ and $\tilde{J}_a^d(\mathbf{r})$.

The triangulation must be done in a consistent manner between the departure and arrival regions. If the triangulation of R_{kl}^d contains an edge connecting vertices \mathbf{r}_{vI}^d and \mathbf{r}_{vJ}^d , then the triangulation of R_{kl}^a must have an edge connecting vertices \mathbf{r}_{vI}^a and \mathbf{r}_{vJ}^a .

Figure 3 provides an illustration of Approximation 3.7. Vertices \mathbf{x}_{vI} , the filled black circles (\bullet) in Figure 3(a), define the overlap region V_{kl} . Vertices \mathbf{r}_{vI}^d , defined by (59), are indicated in Figure 3(b) by vertices (\circ). Departure region R_{kl}^d is defined by connecting these vertices by straight lines, which is indicated by dark gray shading. Vertices of the arrival region \mathbf{r}_{vI}^a , defined by (60), are plotted as (\bullet) in Figure 3(b), and the lines connecting them define R_{kl}^a , shaded light gray.

We note that R_{kl}^a and R_{kl}^d do not necessarily intersect each other. While the reference element is independent of time, departure points and arrival points exist at times t_n and t_{n+1} , respectively, so the regions R_{kl}^d and R_{kl}^a should be thought of separately. Similarly, while arrival regions satisfy $R_Q = \cup_{l \in L_k} R_{kl}^a$ for each k , the same is not true of departure regions R_{kl}^d .

Approximation 3.7 completes the derivation of the JCT method. Substituting the interpolating polynomial (49) into (58) and using (51), the JCT scheme is

$$(62) \quad \sum_{j=1}^{N_p} \bar{Q}_{kj}(t_{n+1}) \int_{R_Q} \phi_j(\mathbf{r}) \phi_i(\mathbf{r}) d\mathbf{r} \\ = \sum_{l \in L_k} \left(\sum_{j=1}^{N_p} \bar{Q}_{lj}(t_n) \int_{R_{kl}^a} \phi_j(\tilde{m}_a^d(\mathbf{r})) \phi_i(\mathbf{r}) \tilde{J}_a^d(\mathbf{r}) d\mathbf{r} \right).$$

Its mixed mass matrix B_{kl} is defined by

$$(63) \quad (B_{kl})_{ij} = \int_{R_{kl}^a} \phi_j(\tilde{m}_a^d(\mathbf{r})) \phi_i(\mathbf{r}) \tilde{J}_a^d(\mathbf{r}) d\mathbf{r}.$$

The integrand is piecewise polynomial, since $\tilde{m}_a^d(\mathbf{r})$ is piecewise linear and $\tilde{J}_a^d(\mathbf{r})$ is piecewise constant. It can, therefore, be computed exactly using Gaussian quadrature, and the resulting scheme is conservative.

3.8. Shape preservation and tracer consistency. Shape preservation and tracer consistency are fundamentally linked. Shape preservation constrains a nodal mixing ratio to lie within the extremal values of its discrete domain of dependence; we refer to this property as *local bounds preservation*. A shape preservation algorithm that ensures local bounds preservation is sufficient to also ensure tracer consistency. For if a mixing ratio field is uniform (such as $q = 1$), local bounds preservation maintains this uniformity. Recall that the total mass density is provided by the dynamics solver's solution of (2). A transport scheme capable of preserving a constant mixing ratio in (1), despite the use of an externally provided total mass density, therefore yields tracer consistency.

Preserving local bounds while conserving global mass may require mass exchange between elements; thus, it is a globally coupled problem that requires global communication. We call an algorithm that solves this problem a constrained density reconstruction (CDR) algorithm. In the context of our discretization, the problem is as follows.

Elements that intersect the discrete domain of dependence of element E_k are given by the list L_k from the common refinement. Let $\boldsymbol{\rho}(t)$ be the vector of coefficients from

the dynamics solver's density solution at time t . We define local bounds

$$(64a) \quad q_k^{\min}(t_{n+1}) = \min_{l \in L_k, i \in 1, \dots, N_p} \frac{Q_{li}(t_n)}{\rho_{li}(t_n)},$$

$$(64b) \quad q_k^{\max}(t_{n+1}) = \max_{l \in L_k, i \in 1, \dots, N_p} \frac{Q_{li}(t_n)}{\rho_{li}(t_n)}.$$

Let \mathbf{w} be the vector of GLL quadrature weights. The shape preservation algorithm finds $\mathbf{Q}(t_{n+1})$ close to $\mathbf{Q}^*(t_{n+1})$, the solution of either (46) or (62), subject to the constraints

$$(65a) \quad q_k^{\min}(t_{n+1}) \leq \frac{Q_{ki}(t_{n+1})}{\rho_{ki}(t_{n+1})} \leq q_k^{\max}(t_{n+1}) \quad \forall k, i,$$

$$(65b) \quad \sum_{k,i} w_{ki} Q_{ki}(t_{n+1}) = \sum_{k,i} w_{ki} Q_{ki}(t_n).$$

In these constraints, k ranges over $1, \dots, N_e$ and i ranges over $1, \dots, N_p$.

Our companion paper [7] discusses conditions required for the existence of solutions to (65). It describes three families of CDR algorithms to find $\mathbf{Q}(t_{n+1})$ close to $\mathbf{Q}^*(t_{n+1})$; each algorithm has a definition of closeness. One method, QLT (for quasi-local, tree-based CDR), redistributes mass approximately locally, where locality is defined by its tree over the mesh, with the minimum possible number of global communication steps. We refer the reader to [7] for details.

A reduction in order of accuracy is common when shape preservation filters are used, and we find this is the case with our schemes as well. With QLT turned on, our schemes are second order accurate for all p even in nondeformational flow, where the isoparametric mapping approximation is exact. Significantly, the second order limit on order of accuracy imposed by QLT is asymptotically equivalent to both our choice of geometric reconstruction and the mapping approximations in JCT. The overall order of accuracy of the scheme in general deformational flow is therefore unaffected by the shape preservation filter. In fact, since E3SM requires shape preserving transport, JCT was designed to be a cell-integrated semi-Lagrangian algorithm whose computational efficiency is optimized for second order accuracy, even when coupled to a dynamics solver whose GLL scheme uses $p \geq 3$.

3.9. Implementation. In each variant of our scheme, we tested several time integrators to compute flow trajectories (3) and found no significant differences between them. The standalone tests shown below use the standard fourth order Runge–Kutta scheme. For the E3SM demonstration in section 5, trajectories are computed with a second order Taylor series method [36]. Vertex connectivities from the Eulerian mesh are reused to define distorted elements $P_l(t_{n+1})$.

Each element E_k independently computes its common refinement using a standard polygonal intersection procedure [50] adapted for the surface of the sphere. Overlap regions V_{kl} are triangulated, and both CDG and IR use a high order Gaussian quadrature rule to construct the mass matrix and mixed mass matrix. With cubic elements, we have found our best results using an 18th order quadrature rule with 66 points per triangle [52]. For JCT with cubic elements a 12th order quadrature rule with 32 points is sufficient for exact integration. We discuss this difference in the number of quadrature points further in section 4. The element-local problem (21) (CDG) or (33) (IR) is solved using the precomputed Cholesky factorization of mass matrix A_k , where A_k is defined by (23), or (51) for the case of JCT. Finally, the QLT shape preservation procedure is applied.

Parallelism. By design, our scheme exposes a significant amount of parallelism that can be exploited in code. The majority of work is element-local and reused for each tracer, e.g., mass matrix construction and factorization, the common refinement, solving the inverse problem (38), mixed mass matrix construction. Since $\mathbf{u}(\mathbf{x}, t)$ is either provided or computed separately, trajectory computations are embarrassingly parallel over the mesh vertices. Only the shape preservation algorithm requires a global reduction; performing this efficiently is the subject of [7]. Furthermore, much of the element-local work is rich in the fine-grained parallelism appropriate for a GPU implementation. The nonlinear solves and interpolations at quadrature points are embarrassingly parallel across quadrature points within an element. The Cholesky factorization, or even the inverse, of the target mass matrix is computed just once at simulation initialization; thus, element-local projections at each time step again are easily parallelized.

4. Standalone tests. In this section we demonstrate our scheme using several standalone test cases from the literature. Each test is defined by a flow velocity $\mathbf{u}(\mathbf{x}, t)$ and a pair of initial conditions, $q_0(\mathbf{x})$ and $\rho_0(\mathbf{x})$. These initial conditions and velocity functions are defined below. For some quantities it may be more convenient to use Cartesian, rather than spherical, coordinates. In such cases, it is understood that $x = a \cos \lambda \cos \theta$, $y = a \sin \lambda \cos \theta$, $z = a \sin \theta$, where a is the radius of the sphere, λ is longitude, and θ is latitude. Velocities are given in terms of the spherical coordinate unit vectors, $\mathbf{u}(\lambda, \theta, t) = u(\lambda, \theta, t)\hat{\mathbf{e}}_\lambda + v(\lambda, \theta, t)\hat{\mathbf{e}}_\theta$, where u, v are the zonal and meridional velocity components, respectively.

For each standalone test, the initial condition for the tracer mixing ratio is given by [31, eq. (7)], restated here for convenience. A single Gaussian is defined on the equator as

$$(66) \quad g_i(\mathbf{x}) = g_{\max} \exp \left[-b \left((x - x_i)^2 + (y - y_i)^2 + z^2 \right) \right], \quad i = 1, 2,$$

with $b = 5$ and $g_{\max} = 0.95$. The initial tracer mixing ratio is a pair of these Gaussians,

$$(67) \quad q_0(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x}),$$

where $x_1 = a \cos(5\pi/6)$, $y_1 = a \sin(5\pi/6)$, $x_2 = a \cos(7\pi/6)$, and $y_2 = a \sin(7\pi/6)$. The initial condition for density is $\rho_0(\mathbf{x}) = 1$.

Tracer error is quantified by $l_\infty(t)$ and $l_2(t)$, standard error norms defined in [31, 55] for the mixing ratio q . We define two additional error norms, the relative change in the global mass integral for each of the two mass definitions, (40) and (41). We denote these by $l_{M_S}(t)$ and $l_{M_p}(t)$,

$$(68) \quad l_{M_S}(t) = \frac{|M_S(t) - M_S(0)|}{M_S(0)}, \quad l_{M_p}(t) = \frac{|M_p(t) - M_p(0)|}{M_p(0)}.$$

Below we show results for both the characteristic discontinuous Galerkin (CDG, (21)) and incremental remap (IR, (33)) schemes, and their variants (46) and (62), with three element types (linear, quadratic, and cubic) and a variety of element widths h . Table 1 shows the mesh sizes $\Delta\lambda$ for each combination of h and p tested. Unless otherwise noted, the results are computed at constant Courant number, $Cr = 5$, where $Cr = \Delta t \max |\mathbf{u}(\mathbf{x}, t)| / (a\Delta\lambda)$.

Test Case 1 is defined by a velocity field $\mathbf{u}(\mathbf{x}, t)$ that corresponds to solid-body rotation about the z -axis with period $T = 12$ days; its components are given by [55, eqs. (75), (76) with $\alpha = 0$]:

$$(69) \quad u(\lambda, \theta, t) = u_0 \cos \theta, \quad v(\lambda, \theta, t) = 0,$$

TABLE 1

Mesh sizes $\Delta\lambda$ (degrees) used for Test Cases 1, 2, and 3 for given element types with degree p and widths h .

		h (degrees)					
		9.0	5.625	3.0	1.5	0.75	0.375
	p						
Linear	1	9.0	5.625	3.0	1.5	0.75	0.375
Quadratic	2	4.5	2.8125	1.5	0.75	0.375	0.1875
Cubic	3	3.0	1.875	1.0	0.5	0.25	0.125

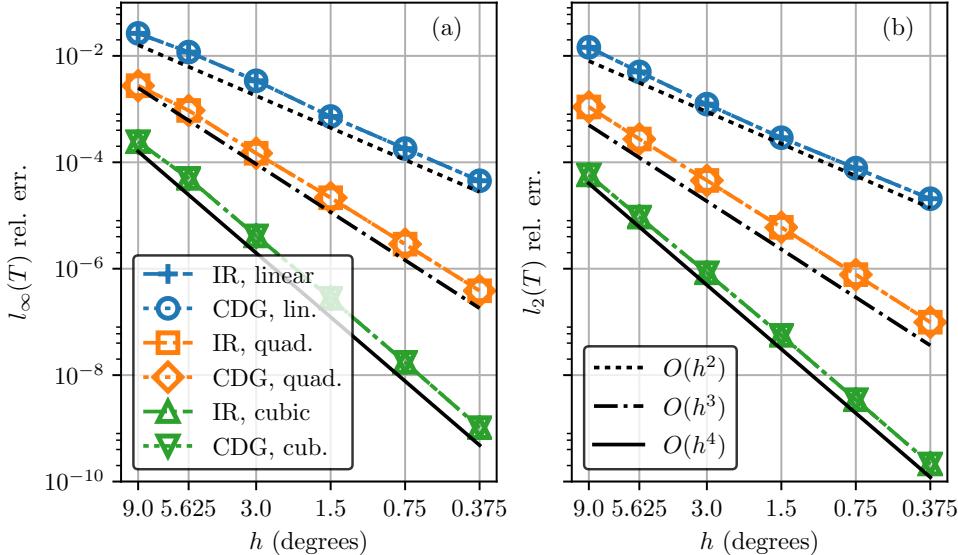


FIG. 4. Test Case 1, tracer error versus spatial resolution at constant Courant number $Cr = 5$ for CDG (dotted lines) and IR schemes (dash-dot lines). Linear elements (+), (\circ); quadratic elements (\square), (\diamond); cubic elements (\triangle), (∇); (a) $l_\infty(T)$, (b) $l_2(T)$.

where $u_0 = 2\pi a/(12 \text{ days})$.

With this simple velocity field the exact solution $q(\mathbf{x}, t)$ is known for all $t > 0$, and element edges do not deform. Only time discretization error and spatial discretization error are present. Isoparametric mapping error is absent for this test case; due to the lack of edge deformation, a linear edge reconstruction is exact. Test Case 1 is used to verify the fundamental accuracy, order of accuracy, and conservation properties of our scheme.

Figure 4 presents tracer error norms for test case 1 after one revolution, $l_\infty(T)$, (a), and $l_2(T)$, (b). For degree p elements both error norms converge at the expected theoretical rate, $O(h^{p+1})$, as $h \rightarrow 0$. In addition to verifying the methods' convergence rates, these results suggest that both CDG and IR schemes are capable of providing high order accuracy in deformational flow when they are combined with a high order geometric reconstruction such as [54]. Differences between the two schemes are not visually evident.

A suite of tests proposed by [31] is designed to challenge transport schemes in more realistic deformational flows. We derive our next test cases from this collection. The velocity fields are reversible in time with period $T = 12$ days. From $t = 0$ to $t = T/2$, the flow deforms and initially large scale features are transported into

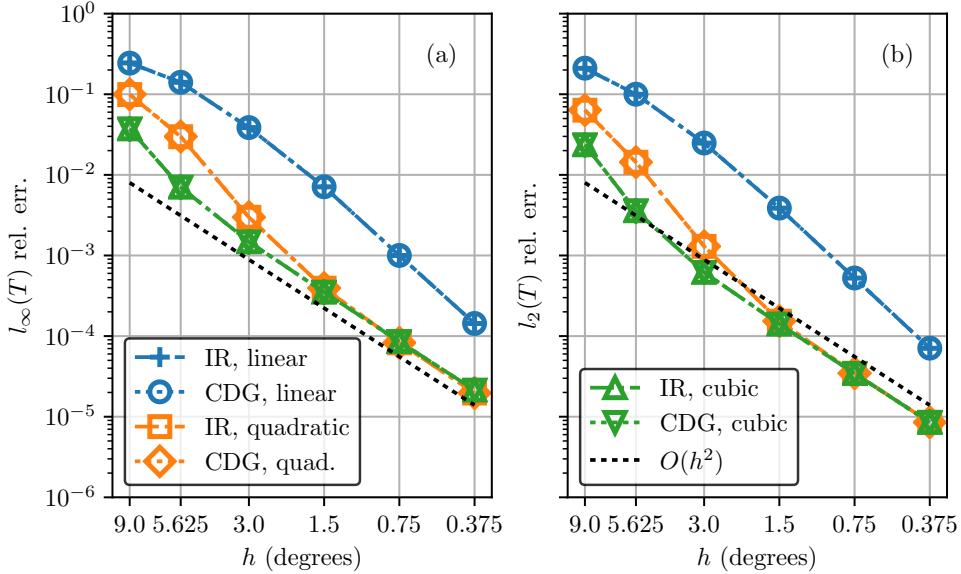


FIG. 5. *Test Case 2, tracer error versus spatial resolution at $Cr = 5$. $l_\infty(T)$, (a); $l_2(T)$, (b).* Curves labeled as in Figure 4.

filaments. From $t = T/2$ to $t = T$, the flow unwinds itself so that the exact solution at $t = T$ is identical to the initial condition. Flow visualizations are provided by [6, 31, 32] but, for the sake of brevity, are not included here. Results from a collection of numerical transport schemes are presented in [32].

Test Case 2 introduces edge deformation and the associated isoparametric mapping error. Its velocity components are defined by [31, eqs. (18) and (19)]

$$(70a) \quad u(\lambda, \theta, t) = \frac{10a}{T} \sin^2 \left(\lambda - \frac{2\pi t}{T} \right) \sin(2\theta) + \frac{2\pi a}{T} \cos \left(\frac{\pi t}{T} \right) + \frac{2\pi a}{T} \cos(\theta),$$

$$(70b) \quad v(\lambda, \theta, t) = \frac{10a}{T} \sin \left(2 \left(\lambda - \frac{2\pi t}{T} \right) \right) \cos(\theta) \cos \left(\frac{\pi t}{T} \right).$$

This velocity field satisfies $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$ analytically. Results are shown in Figure 5. Tracer errors $l_\infty(T)$ (a) and $l_2(T)$ (b) are plotted for both the CDG and IR schemes for the same element types and widths as shown for Test Case 1. As was the case for Test Case 1, differences between the two schemes are not visually evident. In Test Case 2, however, both schemes show second order convergence regardless of element type. This is a result of the piecewise linear geometric reconstruction employed by the common refinement, which imposes a second order bound on the scheme's order of accuracy in deformational flow, as discussed in subsection 3.6.

As noted in subsection 3.3, the difference between the CDG and IR schemes is the replacement of the flow map's Jacobian determinant by the density factor (32). This suggests that discrepancies between the IR and CDG schemes may be more obvious with nonzero velocity divergence, and motivates our third test—a compressible flow field.

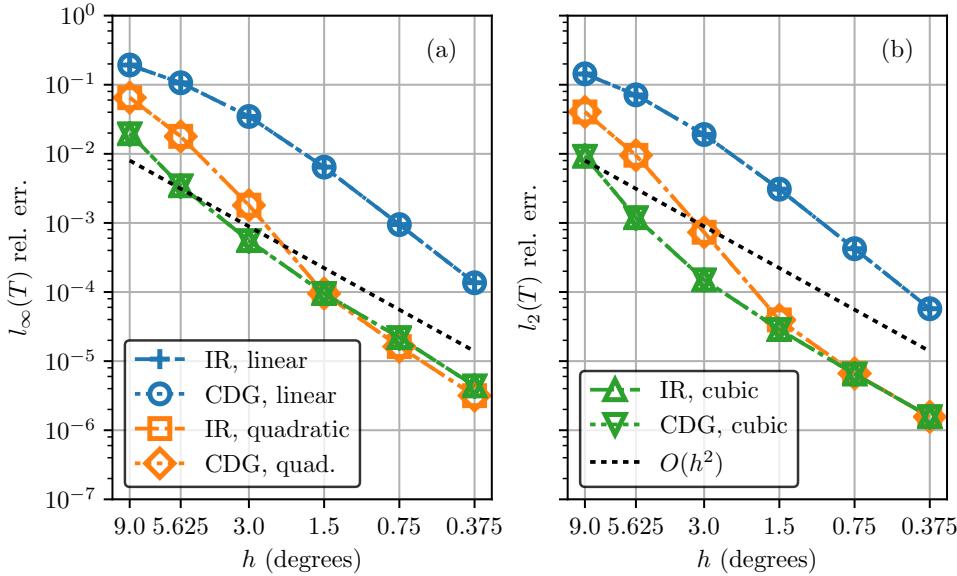


FIG. 6. *Test Case 3, tracer error versus spatial resolution at $Cr = 5$. $l_\infty(T)$, (a); $l_2(T)$, (b).*
Curves labeled as in Figures 4 and 5.

The velocity field for *Test Case 3* is given by [31, eqs. (21) and (22)]

$$(71a) \quad u(\lambda, \theta, t) = -\frac{5a}{T} \sin^2\left(\frac{1}{2}\left(\lambda - \frac{2\pi t}{T}\right)\right) \sin(2\theta) \cos^2(\theta) \cos\left(\frac{\pi t}{T}\right) + \frac{2\pi a}{T} \cos(\theta),$$

$$(71b) \quad v(\lambda, \theta, t) = \frac{5a}{2T} \sin\left(\lambda - \frac{2\pi t}{T}\right) \cos^3(\theta) \cos\left(\frac{\pi t}{T}\right).$$

This provides a standard test for divergent flow fields ($\nabla \cdot \mathbf{u} \neq 0$). Figure 6 shows results. Tracer errors $l_\infty(T)$ (a) and $l_2(T)$ (b) are plotted as in Figures 4 and 5. Convergence rates for each element type are second order due to the deformational flow field. As in the previous two tests, differences between the two schemes appear negligible. To examine their differences more closely, we plot the difference between CDG error and IR error for Test Cases 1–3 in Figure 7.

Figure 7(a) shows that, for the rigid rotation test case, the two schemes are equivalent numerically (to near machine precision). Figure 7(b) plots the difference of the two schemes' tracer errors for Test Case 2, whose velocity field's analytic definition has zero divergence. While the two schemes are mathematically equivalent for this flow, in their discrete forms the actual flow may exhibit nonzero divergence at the level of truncation error. The difference between the two schemes decays at an approximately cubic rate, confirming our prediction from the Taylor series analysis mentioned in subsection 3.3. Figure 7(c) shows a similar result for Test Case 3, whose velocity field is divergent by design. In this test case too, the two schemes' difference decays at a rate of $O(h^3)$ as $h \rightarrow 0$ for each element type. While these differences are nonzero, they are too small to be noticeable in Figures 4–6.

Our results so far have established the accuracy and order of accuracy traits of our scheme. We now present its conservation properties. Figure 8(a) presents $l_{MS}(t)$ (a) and $l_{MP}(t)$ (b) for $t \in [0, T]$ in Test Case 2. Figure 8(a) shows that the spherical definition of mass, (40), is conserved nearly to machine precision. The coarsest grid,

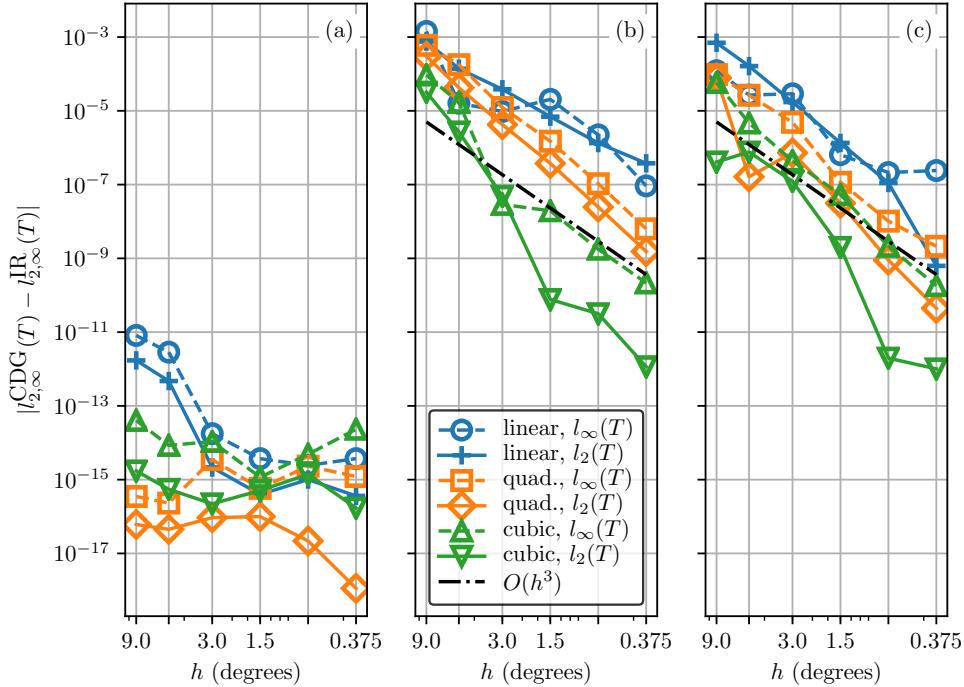


FIG. 7. Difference in tracer error between CDG and IR versus spatial resolution at $Cr = 5$ for (a) Test Case 1, (b) Test Case 2, (c) Test Case 3. Dashed lines, $l_\infty(T)$; solid lines $l_2(T)$; linear elements ($\circ, +$), quadratic elements (\square, \diamond), cubic elements (\triangle, ∇).

$h = 2^\circ$ with linear elements, has relative mass error $l_{M_S}(T) < 1.0E-13$. Decreasing h or using higher order elements (or both) reduces the error to less than $1.0E-14$. Figure 8(b) suggests that $l_{M_P}(t) \rightarrow 0$ in both limits $h \rightarrow 0$ and $p \rightarrow \infty$, but for fixed $h > 0$ and finite p the scheme does not conserve the spectral approximation of mass (41). This motivates the use of either constrained transport or JCT.

We discuss Constrained Transport (46) first. Figure 9(a) shows the relative change of the spectral mass integral, $l_{M_p}(t)$, for the same element types and resolutions as Figure 8(b). These results indicate that constrained transport achieves tracer mass conservation with respect to $M_p(t)$, i.e., $l_{M_p}(T) \sim O(1.0E-15)$ for all $h > 0$ and $p < \infty$. Next we examine the effects of the constraints on the scheme's accuracy.

Figure 10(a,c) shows tracer error for Test Case 1 with constrained transport; a reduction in order of accuracy is apparent with linear and quadratic elements. We find that, for nondeformational flow, error decays as $O(h^p)$ as $h \rightarrow 0$ for degree p elements, instead of the order $p + 1$ convergence rate of the unconstrained scheme. Cubic elements also decrease from fourth to third order convergence, but this is not obvious from the plot since, for these elements, this effect is only noticeable at very high resolutions. High-resolution results are reported for cubic elements in Table 2; the approximate rate of convergence approaches third order as spatial resolution increases. Table 2 reports the time step, Δt , since for these experiments $Cr \neq 5$. Based on these results and the discussion in [54] regarding high order geometric reconstructions, we conclude that constrained transport is a viable candidate for future development into a high order (i.e., third order or higher) conservative transport scheme.

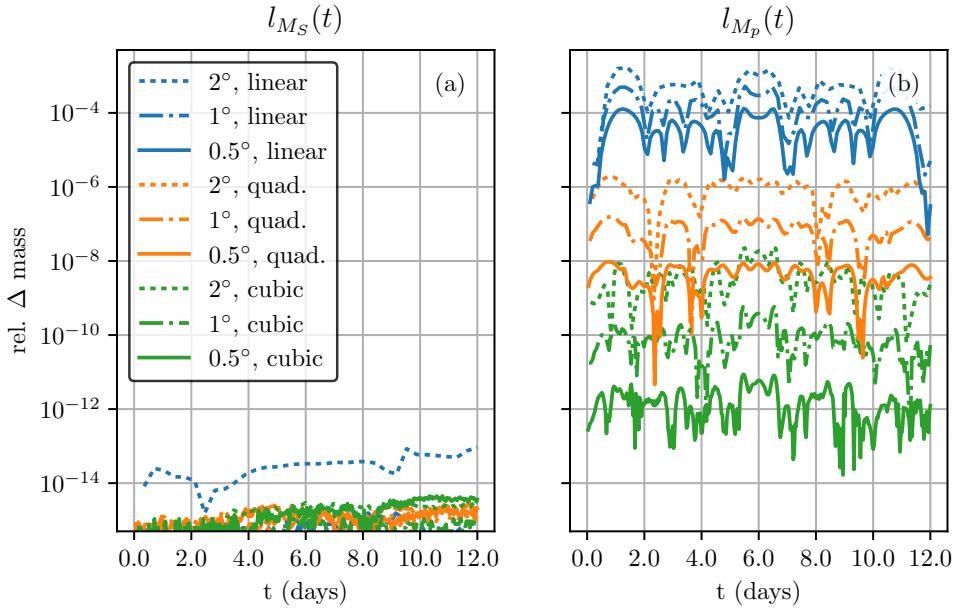


FIG. 8. *Test Case 2, relative change in tracer mass versus time for (a) spherical mass, (b) its spectral approximation. Colors distinguish element types: blue, linear; orange, quadratic; green, cubic. Line styles distinguish element size h : dotted, 2° ; dash-dot, 1° ; solid, 0.5° . (Figure in color online.)*

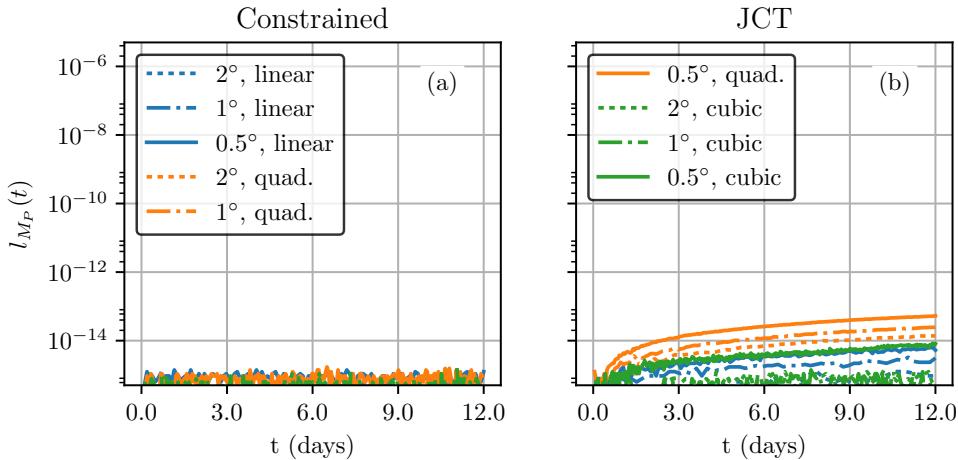


FIG. 9. *Test Case 2, relative change in spectral tracer mass approximation versus time for constrained transport (a), Jacobian-combined transport (b). Curves are labeled as in Figure 8.*

We repeat the above experiments now for JCT, with the timestep again defined so that $Cr = 5$. Figure 9(b) plots $l_{M_p}(t)$, the relative change in the tracer mass integral, for Test Case 2 as computed by JCT. The results show that tracer mass error is approximately $O(1.0E-14)$, i.e., the scheme is conservative to near machine precision with a small accumulation of floating point error in time. Recall that JCT relies on piecewise linear mappings in a flat reference space and therefore cannot achieve

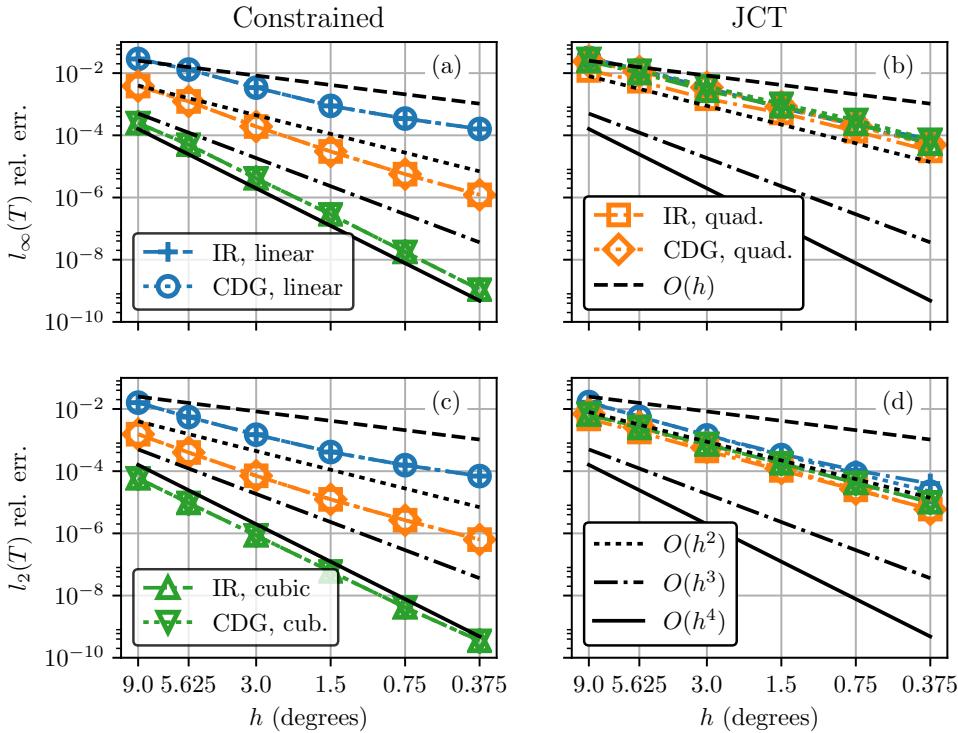


FIG. 10. Test Case 1, tracer error for constrained transport (a,c); JCT (b,d). Curves labeled as in Figures 4–6; legends apply to all panels.

TABLE 2

Test Case 1, constrained transport with $p = 3$ elements, width h and timestep Δt . Tracer errors $l_2(T)$ and $l_\infty(T)$, their approximate convergence rates, spectral mass error $l_{M_p}(T)$.

h	Δt (s)	$l_2(T)$	Rate	$l_\infty(T)$	Rate	$l_{M_p}(T)$
9°	4320	1.20E-04	—	3.86E-04	—	4.09E-15
4.5°	2160	7.78E-06	3.94	2.52E-05	3.94	9.49E-15
2.25°	1080	5.39E-07	3.85	1.52E-06	4.06	2.34E-14
1.125°	540	4.59E-08	3.55	1.01E-07	3.91	3.24E-14
0.5625°	270	4.93E-09	3.22	8.70E-09	3.53	3.31E-14
0.28125°	135	5.95E-10	3.05	9.80E-10	3.15	4.93E-14

orders of accuracy greater than 2, even in Test Case 1's nondeformational flow. Test Case 1 tracer errors $l_\infty(T)$ and $l_2(T)$ are shown in Figure 10(b,d). The JCT scheme is approximately second order accurate for element degrees $p \geq 1$ as $h \rightarrow 0$. Convergence rates for the deformational flows of Test Cases 2 and 3 are discussed below with shape preservation.

Remarks. The results shown in Figures 4–7 and 10 show that the differences between CDG and IR are insignificant, particularly for the E3SM standard configuration which uses cubic elements. Figure 9 shows that both conservation strategies succeed in conserving mass as defined by the coupled dynamics solver. We have found that the implementation of the above algorithms in code most closely follows the mathematics of the schemes' derivations if either (a) incremental remap is combined with constrained transport (IR/constrained) or (b) characteristic discontinu-

ous Galerkin is combined with Jacobian-combined transport (CDG/JCT). While we have tested each combination of the above options—CDG, IR, constrained transport, JCT—and our code supports any of these combinations, our two favored groupings are IR/constrained and CDG/JCT.

Computational cost. Between the two favored methods, the number of required computations is nearly the same except for the construction of the mixed mass matrix B_{kl} . In IR/constrained transport, integration occurs within the spherical triangle $\tau_S \subseteq V_{kl}$, and each quadrature point $\mathbf{x}_J \in V_{kl}$ must be mapped to R_Q . Hence, for each \mathbf{x}_J the method requires the nonlinear solution of (38) to define \mathbf{r}_J such that $\mathbf{x}_J = m_{R_Q}^{E_k}(\mathbf{r}_J)$, then performs two interpolations to evaluate basis function values using (37) and Approximation 3.5. For 66 quadrature points, this sums to 66 solves and 132 interpolations per triangle. In CDG/JCT, integration occurs in the planar triangle $\tau_R \subseteq R_{kl}^a$. Only the vertices of V_{kl} need to be mapped to R_Q . Basis function values again require two interpolations per quadrature point. For 32 quadrature points, since vertices may be shared between adjacent triangles, JCT requires ≤ 3 solves and 64 interpolations per triangle. This implies that for each time step, CDG/JCT requires $\approx 22\times$ fewer nonlinear solves and $\approx 2\times$ fewer interpolations than the IR/constrained scheme. In wallclock time, for the test cases described above, we find that CDG/JCT is approximately five to ten times faster than IR/constrained transport, depending on the particular problem and the particular machine. We therefore choose CDG/JCT over IR/constrained transport to demonstrate our scheme for the remainder of this section and in the full E3SM model.

Shape preservation. In practice we can expect the requirement of shape preservation for which we use the QLT algorithm to solve (65). Figure 11 plots tracer error norms $l_\infty(T)$ (a,b) and $l_2(T)$ (c,d) versus element width h for Test Case 2 (a,c) and Test Case 3 (b,d) using CDG/JCT with QLT. Empirical convergence rates as $h \rightarrow 0$ are slightly less than second order for $l_\infty(T)$. The reduction in accuracy is a common side effect of shape preservation—local extrema tend to be reduced. The $l_2(T)$ error is second order for both deformational flow test cases. While the convergence rates for $p = 1$ elements appear superquadratic, they return to near second order ($l_\infty(T)$) and second order ($l_2(T)$) asymptotically as resolution increases beyond $h < 0.375^\circ$. This confirms our expectation that the combination of JCT and QLT is approximately second order accurate; the shape preservation algorithm does not significantly degrade the order of accuracy of the overall scheme.

5. E3SM demonstration. In this section, the computational performance of CDG/JCT is compared to the current E3SM spectral element (SE) transport scheme. We use the baroclinic instability problem defined by [25] since it is already implemented as a test case within E3SM. The model is run with $h = 4.5^\circ$ and cubic elements (mesh size $\Delta\lambda = 1.5^\circ$) and 26 vertical levels. The transport scheme solves a two-dimensional transport problem on each level and a separate set of algorithms, coupled to both SE and CDG/JCT, are used for the vertical discretization. The SE transport scheme uses the same time step, $\delta t = 100$ seconds, as the E3SM dynamics solver. CDG/JCT uses a time step of $\Delta t = 8\delta t$. The multiplier 8 is the maximum integer multiplier that guarantees trajectories do not depart a 1-element halo. Since the dynamics solver is subcycled relative to CDG/JCT, sufficient data are available for the intermediate stages of trajectory approximation. Both schemes have shape preservation filters turned on; for CDG/JCT, this implies the QLT density reconstruction [7]. The CDG/JCT solutions are converted to the continuous representation expected by the dynamics solver using a direct stiffness summation procedure [51], and this step is

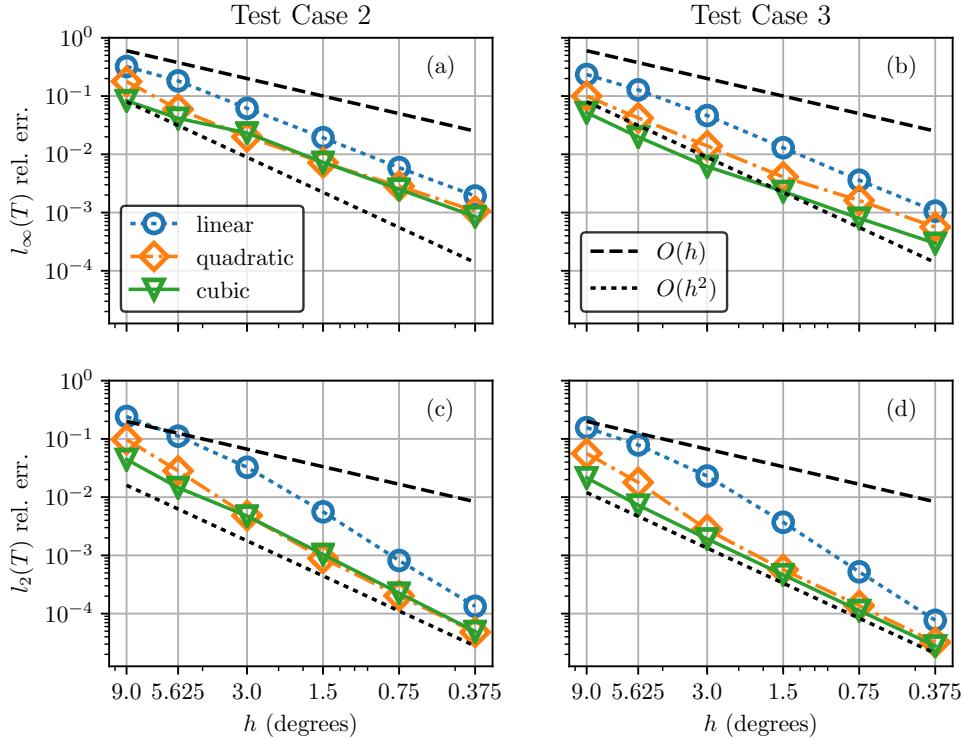


FIG. 11. Tracer error $l_\infty(T)$ (a,b); $l_2(T)$ (c,d); versus h for CDG/JCT with QLT and shape preservation filter. Test Case 2 (a, c); Test Case 3 (b,d). (Figure in color online.)

included in the timing experiment described below. The test case was run with 2400 MPI ranks on 75 Intel Haswell nodes (128GB RAM, 32 cores per node) of the Sandia National Laboratories' Mutriño high-performance computing cluster. This setup achieves the strong scaling limit of one element per core. Both SE and CDG/JCT methods use the same full halo exchange communication pattern.

We now introduce some notation to facilitate our discussion of performance. Let $T_{SE}(\Delta\lambda, N_Q)$ represent the compute time required by SE transport to advance one timestep, from $t = t_n$ to $t = t_n + \delta t$, at a given resolution $\Delta\lambda$ with N_Q tracers. $T_{JCT}(\Delta\lambda, N_Q)$ represents the compute time for the CDG/JCT scheme to advance its solution from $t = t_n$ to $t = t_n + \Delta t$. The total computing time required by the current E3SM dynamical core is

$$(72) \quad C_{SE}(\Delta\lambda, N_Q) = n_s (D(\Delta\lambda) + T_{SE}(\Delta\lambda, N_Q)),$$

where $n_s = t_f/\delta t$ is the total number of dynamics steps required to reach the simulation's final time, t_f , and $D(\Delta\lambda)$ represents the time required by the dynamics solver to advance the model state from $t = t_n$ to $t = t_n + \delta t$. The total time required by the E3SM dynamics solver coupled to CDG/JCT is

$$(73) \quad C_{JCT}(\Delta\lambda, N_Q) = n_s \left(D(\Delta\lambda) + \frac{\delta t}{\Delta t} T_{JCT}(\Delta\lambda, N_Q) \right).$$

We perform several experiments, each using a different number of tracers $N_Q \in \{0, 1, 2, 4, 8, 12, 16, 20, 25, 30, 40\}$, with both SE and CDG/JCT. The model's primary

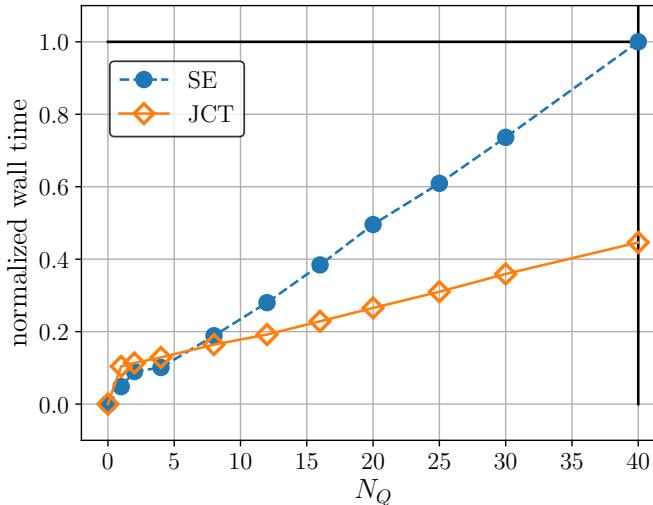


FIG. 12. E3SM atmosphere test case, normalized transport scheme wallclock time versus number of tracers N_Q . SE transport (blue circles, dashed line); CDG/JCT (orange diamonds, solid line). (Figure in color online.)

main loop timer provides values for $C_{SE}(1.5^\circ, N_Q)$ and $C_{JCT}(1.5^\circ, N_Q)$. The case $N_Q = 0$ provides a value for $D(\Delta\lambda) = C_{SE}(\Delta\lambda, 0)/n_s$. This value is then used to compute $T_{SE}(1.5^\circ, N_Q)$ and $T_{JCT}(1.5^\circ, N_Q)$ for $N_Q > 0$ using (72) and (73), respectively. Finally, we normalize the transport time so that the E3SM version 1 configuration, with $N_Q = 40$ and SE transport, corresponds to a value of 1.

Figure 12 shows normalized transport times $T_{SE}(1.5^\circ, N_Q)/T_{SE}(1.5^\circ, 40)$ and $T_{CDG}(1.5^\circ, N_Q)/T_{SE}(1.5^\circ, 40)$ as functions of N_Q . For $N_Q > 7$ the performance of CDG/JCT is better than SE, and for $N_Q = 40$, JCT is a factor 2.24 times faster than SE. The ratios n_s/C_{SE} and n_s/C_{JCT} represent the average number of dynamics steps computed per second by each dynamical core. With $N_Q = 40$, we find that the current dynamical core is able to compute approximately 204 steps per second, while the new scheme achieves a rate of approximately 389 steps per second, a $1.91 \times$ improvement in overall dynamical core performance.

Discussion. The speedup of CDG/JCT relative to SE is due to two factors. First, due to a larger time step the semi-Lagrangian multimoment algorithms (SLMM) presented in this work require both fewer communication rounds and reduced communication volume per unit of simulated time than the SE method, even though both methods transfer data via full halo exchange in these experiments. Second, the SLMM schemes have greater multitracer efficiency. With both CDG/JCT and IR/constrained transport, flow map trajectories, the common refinement, mass matrices, and mixed mass matrices are independent of q and may be reused for each tracer.

Since both schemes used the same MPI communication pattern, dynamics solver, and vertical discretization, the above reported speedups are due nearly exclusively to the algorithmic differences between the transport schemes. Consequently, the normalized ratios plotted in Figure 12 are intended to be broadly representative of what may be expected at other resolutions. However, the ratios n_s/C_{SE} and n_s/C_{JCT} are more closely tied to this particular machine, resolution, and parallel domain decomposition.

In the above analysis, we have assumed that the time required by the dynamics

solver is the same for both transport schemes. In practice we find that some of the dynamics solver's main subroutines proceed faster when coupled to a semi-Lagrangian transport scheme rather than the SE scheme. As the ratio $\delta t/\Delta t$ decreases, two things happen. First, the dynamics variables can remain in fast memory (e.g., L3 cache) for longer periods of time before that memory is needed by the transport scheme. Second, there are fewer point-to-point communications required for transport, which reduces the amount of time processors spend waiting for data to arrive. Together, these lead to small but noticeable improvements in the dynamics solver's performance.

6. Summary. Two SLMMs for tracer transport on the sphere are presented, the Characteristic Discontinuous Galerkin (CDG) method (22) and the Incremental Remap (IR) method (34). The methods were designed with the goal of minimized MPI communication between compute nodes as a primary motivation. Each method relies on a locally constructed overlap mesh, or common refinement (Definition 3.3), to relate distorted Lagrangian elements to the Eulerian mesh. In our current software implementations of these algorithms, we use a linear geometric reconstruction to define the common refinement and its associated isoparametric mappings. The CDG and IR methods differ only in the construction of their linear operator; their accuracies are nearly identical (Figure 4). In general deformational flows, both schemes are second order accurate (Figures 5 and 6) due to their isoparametric maps.

We have developed two strategies, constrained transport (46) and Jacobian-combined transport (JCT) (62), to conserve the definition of the discrete mass integral used by the E3SM dynamics solver. Both strategies maintain second order accuracy overall (Figure 10). JCT relies on more approximations than the constrained scheme but requires less computational work.

A constrained density reconstruction algorithm, QLT, is introduced; it provides both tracer consistency and shape preservation and is compatible with each combination of the above methods. QLT is one choice from a family of constrained density reconstruction algorithms which are discussed in detail in a companion paper [7]. For cubic elements, adding QLT to the CDG/JCT scheme does not degrade its second order accuracy (Figure 11).

The CDG/JCT scheme with QLT is shown to reduce by a factor of 2.24 the time required by tracer transport in the full E3SM atmosphere model in its standard configuration at the strong scaling limit of one element per core (Figure 12) using the full halo exchange communication pattern. Its performance advantage results from its large time step and multitracer efficiency. Speedup over the current E3SM transport scheme increases linearly with the number of tracers. Additional speedup is likely if CDG/JCT is combined with an upwind MPI communication pattern (subsection 3.4).

Software. We developed two C++ libraries that encapsulate the main subroutines required by our algorithms. Spherical Intersection and Quadrature with Kokkos computes the common refinement and provides methods for quadrature on arbitrary spherical and planar polygons. A second library, Communication-Efficient Density Reconstruction, provides the Quasi-Local Tree-based (QLT) density reconstruction used for tracer consistency and the local shape preservation filter. Both libraries use the Kokkos programming model [53] for performance-portability and run on various computer architectures, including GPUs and Intel's KNL processors. They are open-source and publicly available [8]. Our software and algorithms will be integrated with E3SM version 2.

REFERENCES

- [1] T. ARBOGAST AND C.-S. HUANG, *A fully mass and volume conserving implementation of a characteristic method for transport problems*, SIAM J. Sci. Comput., 28 (2006), pp. 2001–2022, <https://doi.org/10.1137/040621077>.
- [2] D. BADER, W. COLLINS, R. JACOB, P. JONES, P. RASCH, M. TAYLOR, P. THORNTON, AND D. WILLIAMS, *Accelerated Climate Modeling for Energy: Project Strategy and Initial Implementation Plan*, Tech. report, U. S. Dept. of Energy Office of Biological and Environmental Research, Washington, DC, 2014.
- [3] F. BASSI AND S. REBAY, *High-order accurate discontinuous finite element solution of the 2D Euler equations*, J. Comput. Phys., 138 (1997), pp. 251–285.
- [4] J. BEHRENS AND A. ISKE, *Grid-free adaptive semi-Lagrangian advection using radial basis functions*, Comput. Math. Appl., 43 (2002), pp. 319–327.
- [5] P. BOCHEV, D. RIDZAL, AND K. PETERSON, *Optimization-based remap for transport: A divide and conquer strategy for feature-preserving discretizations*, J. Comput. Phys., 257 (2014), pp. 1113–1139, <https://doi.org/10.1016/j.jcp.2013.03.057>.
- [6] P. A. BOSLER, J. KENT, R. KRASNÝ, AND C. JABLONOWSKI, *A Lagrangian particle method with remeshing for tracer transport on the sphere*, J. Comput. Phys., 340 (2017), pp. 639–654, <https://doi.org/10.1016/j.jcp.2017.03.052>.
- [7] A. M. BRADLEY, P. A. BOSLER, O. GUBA, M. A. TAYLOR, AND G. A. BARNETT, *Communication-efficient property preservation in tracer transport*, SIAM J. Sci. Comput., 41 (2019), pp. C161–C193, <https://doi.org/10.1137/18M1165414>.
- [8] A. M. BRADLEY, O. GUBA, P. A. BOSLER, AND M. A. TAYLOR, *COMPOSE: Library for Communication-Efficient, Property-Preserving, Semi-Lagrangian Tracer Transport*, 2019, <https://doi.org/10.5281/zenodo.2552888>, <https://github.com/E3SM-Project/COMPOSE>.
- [9] M. CELIA, *An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation*, Adv. Water Resour., 13 (1990), pp. 187–206.
- [10] P. N. CHILDS AND K. W. MORTON, *Characteristic Galerkin methods for scalar conservation laws in one dimension*, SIAM J. Numer. Anal., 27 (1990), pp. 553–594, <https://doi.org/10.1137/0727035>.
- [11] A. J. CHORIN AND J. E. MARSDEN, *A Mathematical Introduction to Fluid Dynamics*, 3rd ed., Springer-Verlag, New York, 1993.
- [12] C. COTTER, J. FRANK, AND S. REICH, *The remapped particle-mesh semi-Lagrangian advection scheme*, Q. J. Royal Meterol. Soc., 133 (2007), pp. 251–260.
- [13] S. DEY, M. S. SHEPHARD, AND J. E. FLAHERTY, *Geometry representation issues associated with p-version finite element computations*, Comput. Methods Appl. Mech. Engrg., 150 (1997), pp. 39–55.
- [14] J. DOUGLAS, JR., AND T. F. RUSSELL, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Anal., 19 (1982), pp. 871–885, <https://doi.org/10.1137/0719063>.
- [15] D. DRITSCHEL AND M. H. P. AMBAUM, *A contour-adveected semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields*, Q. J. Royal Meterol. Soc., 123 (1997), pp. 1097–1130.
- [16] J. K. DUKOWICZ AND J. R. BAUMGARDNER, *Incremental remapping as a transport/advection algorithm*, J. Comput. Phys., 160 (2000), pp. 318–335, <https://doi.org/10.1006/jcph.2000.6465>.
- [17] P. E. FARRELL, M. D. PIGGOT, C. C. PAIN, G. J. GORMAN, AND C. R. WILSON, *Conservative interpolation between unstructured meshes via supermesh construction*, Comput. Methods Appl. Mech. Eng., 198 (2009), pp. 2632–2642, <https://doi.org/10.1016/j.cma.2009.03.004>.
- [18] N. FLYER AND E. LEHTO, *Rotational transport on a sphere: local node refinement with radial basis functions*, J. Comput. Phys., 229 (2010), pp. 1954–1969.
- [19] N. FLYER AND G. B. WRIGHT, *Transport schemes on a sphere using radial basis functions*, J. Comput. Phys., 226 (2007), pp. 1059–1084.
- [20] B. FORNBERG AND E. LEHTO, *Stabilization of RBF-generated finite-difference methods for convective PDEs*, J. Comput. Phys., 230 (2011), pp. 2270–2285.
- [21] F. X. GIRALDO, *Lagrange-Galerkin methods on spherical geodesic grids*, J. Comput. Phys., 136 (1997), pp. 197–213.
- [22] E. S. GROSS, L. BONAVENTURA, AND G. ROSATTI, *Consistency with continuity in conservative advection schemes for free-surface models*, Int. J. Numer. Methods Fluids, 38 (2002), pp. 307–327.

- [23] O. GUBA, M. A. TAYLOR, P. A. ULLRICH, J. R. OVERFELT, AND M. N. LEVY, *The spectral element method (SEM) on variable-resolution grids: evaluating grid sensitivity and resolution-aware numerical viscosity*, Geosci. Model Dev., 7 (2014), pp. 2803–2816, <https://doi.org/10.5194/gmd-7-2803-2014>.
- [24] C. W. HIRT, A. A. AMSDEN, AND J. L. COOK, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys., 14 (1974), pp. 227–253.
- [25] C. JABLONOWSKI AND D. L. WILLIAMSON, *A baroclinic instability test case for atmospheric model dynamical cores*, Q. J. Royal Meterol. Soc., 132 (2006), pp. 2943–2975, <https://doi.org/10.1256/qj.06.12>.
- [26] X. JIAO AND M. T. HEATH, *Common-refinement-based data transfer between non-matching meshes in multiphysics simulations*, Int. J. Numer. Meth. Eng., 61 (2004), pp. 2402–2427, <https://doi.org/10.1002/nme.1147>.
- [27] E. KAAS, B. SORENSEN, P. H. LAURITZEN, AND A. B. HANSEN, *A hybrid Eulerian-Lagrangian numerical scheme for solving prognostic equations in fluid dynamics*, Geosci. Model Dev., 6 (2013), pp. 2023–2047.
- [28] R. KRAUSE AND P. ZULIAN, *A parallel approach to the variational transfer of discrete fields between arbitrarily distributed unstructured finite element meshes*, SIAM J. Sci. Comput., 38 (2016), pp. C307–C333, <https://doi.org/10.1137/15M1008361>.
- [29] J.-F. LAMARQUE, L. K. EMMONS, P. G. HESS, D. E. KINNISON, S. TILMES, F. VITT, C. L. HEALD, E. A. HOLLAND, P. H. LAURITZEN, J. NEU, J. J. ORLANDO, P. J. RASCH, AND G. K. TYNDALL, *CAM-chem: description and evaluation of interactive atmospheric chemistry in the Community Earth System Model*, Geosci. Model Dev., 5 (2012), pp. 369–411.
- [30] P. H. LAURITZEN, R. D. NAIR, AND P. A. ULLRICH, *A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid*, J. Comput. Phys., 229 (2010), pp. 1401–1424.
- [31] P. H. LAURITZEN, W. C. SKAMAROCK, M. J. PRATHER, AND M. A. TAYLOR, *A standard test case suite for two-dimensional linear transport on the sphere*, Geosci. Model Dev., 5 (2012), pp. 887–901, <https://doi.org/10.5194/gmd-5-887-2012>.
- [32] P. H. LAURITZEN, P. A. ULLRICH, C. JABLONOWSKI, P. A. BOSLER, D. CALHOUN, A. J. CONLEY, T. ENOMOTO, L. DONG, S. DUBEY, O. GUBA, A. B. HANSEN, E. KAAS, J. KENT, J.-F. LAMARQUE, M. J. PRATHER, D. REINERT, V. V. SHASHKIN, W. C. SKAMAROCK, B. SORENSEN, M. A. TAYLOR, AND M. A. TOLSTYKH, *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 (2014), pp. 105–145, <https://doi.org/10.5194/gmd-7-105-2014>.
- [33] P. H. LAURITZEN, P. A. ULLRICH, AND R. D. NAIR, *Atmospheric transport schemes: Desirable properties and a semi-Lagrangian view on finite-volume discretizations*, in Numerical Techniques for Global Atmospheric Models, P. H. Lauritzen, C. Jablonowski, M. A. Taylor, and R. D. Nair, eds., Lect. Notes Comput. Sci. Eng. 80, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 188–250.
- [34] J. H. W. LEE, J. PERAIRE, AND O. C. ZIENKIEWICZ, *The characteristic-Galerkin method for advection-dominated problems—An assessment*, Comput. Methods Appl. Mech. Eng., 61 (1987), pp. 359–369.
- [35] S. J. LIN AND R. B. ROOD, *Multidimensional flux-form semi-Lagrangian transport schemes*, Mon. Weather Rev., 124 (1996), 2046–2070.
- [36] J. L. MCGREGOR, *Economical determination of departure points for semi-Lagrangian models*, Mon. Weather Rev., 121 (1993), pp. 221–230.
- [37] K. W. MORTON, A. PRIESTLY, AND E. SULLI, *Stability of the Lagrange-Galerkin method with non-exact integration*, RAIRO Model. Math. Anal. Numer., 22 (1988), pp. 625–653.
- [38] R. D. NAIR, S. J. THOMAS, AND R. D. LOFT, *A discontinuous Galerkin transport scheme on the cubed sphere*, Mon. Weather Rev., 133 (2005), pp. 814–828.
- [39] P. JÖCKEL, R. VON KUHLMANN, M. G. LAWRENCE, B. STEIL, C. A. M. BRENNINKMEIJER, P. J. CRUTZEN, P. J. RASCH, AND B. EATON, *On a fundamental problem in implementing flux-form advection schemes for tracer transport in 3-dimensional general circulation and chemistry transport models*, Q. J. Royal Meterol. Soc., 127 (2001), pp. 1035–1052.
- [40] A. PRIESTLY, *Exact projections and the Lagrange-Galerkin method: A realistic alternative to quadrature*, J. Comput. Phys., 112 (1994), pp. 316–333.
- [41] W. M. PUTNAM AND S.-J. LIN, *Finite-volume transport on various cubed-sphere grids*, J. Comput. Phys., 227 (2007), pp. 55–78.
- [42] M. RESTELLI, L. BONAVENTURA, AND R. SACCO, *A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flow*, J. Comput. Phys., 216 (2006), pp. 195–215.

- [43] R. B. ROOD, *Numerical advection algorithms and their role in atmospheric transport and chemistry models*, Rev. Geophysics, 25 (1987), pp. 71–100.
- [44] D. A. ROTMAN, C. S. ATHERTON, D. J. BERGMANN, P. J. CAMERON-SMITH, C. CHUANG, P. S. CONNELL, J. E. DIGNON, A. FRANZ, K. E. GRANT, D. E. KINNISON, C. R. MOLENKAMMP, D. D. PROCTOR, AND J. R. TANNAHILL, *IMPACT, the LLNL 3-D global atmospheric chemical transport model for the combined troposphere and stratosphere: Model description and analysis of ozone and other trace gases*, J. Geophys. Res., 109 (2004), D04303, <https://doi.org/10.1029/2002JD003155>.
- [45] R. SEVILLA, S. FERNÁNDEZ-MÉNDEZ, AND A. HUERTA, *Comparison of high-order curved finite elements*, Int. J. Numer. Meth. Eng., 87 (2011), pp. 719–734.
- [46] V. SHANKAR AND G. B. WRIGHT, *Mesh-free semi-Lagrangian methods for transport on a sphere using radial basis functions*, J. Comput. Phys., 366 (2018), pp. 170–190.
- [47] Z. SI, J. WANG, AND W. SUN, *Unconditional stability and error estimates of modified characteristics FEMs for the Navier–Stokes equations*, Numer. Math., 134 (2016), pp. 139–161, <https://doi.org/10.1007/s00211-015-0767-9>.
- [48] A. STANIFORTH AND J. COTÉ, *Semi-Lagrangian integration schemes for atmospheric models: A review*, Mon. Weather Rev., 119 (1991), pp. 2206–2223.
- [49] G. STRANG AND G. FIX, *An Analysis of the Finite Element Method*, 2nd ed., Wellesley-Cambridge Press, Wellesley, MA, 2008.
- [50] I. SUTHERLAND AND G. W. HODGMAN, *Reentrant polygon clipping*, Comm. ACM, 17 (1974), pp. 32–42.
- [51] M. A. TAYLOR AND A. FOURNIER, *A compatible and conservative spectral element method on unstructured grids*, J. Comput. Phys., 229 (2010), pp. 5879–5895.
- [52] M. A. TAYLOR, B. A. WINGATE, AND L. P. BOS, *A cardinal function algorithm for computing multivariate quadrature points*, SIAM J. Numer. Anal., 45 (2007), pp. 193–205, <https://doi.org/10.1137/050625801>.
- [53] C. R. TROTT, M. HOEMMEN, S. D. HAMMOND, AND H. C. EDWARDS, *Kokkos: The programming guide*, Tech. Report SAND2015-4178 O, Sandia National Laboratories, Albuquerque, NM, 2015, <https://github.com/kokkos/kokkos>.
- [54] P. A. ULLRICH, P. H. LAURITZEN, AND C. JABLONOWSKI, *Some considerations for high-order ‘incremental remap’-based transport schemes: edges, reconstructions, and area integration*, Int. J. Numer. Meth. Fluids, 71 (2013), pp. 1131–1151, <https://doi.org/10.1002/fld.3703>.
- [55] D. L. WILLIAMSON, J. B. DRAKE, J. J. HACK, R. JAKOB, AND P. N. SWARZTRAUBER, *A standard test set for numerical approximations to the shallow water equations in spherical geometry*, J. Comput. Phys., 102 (1992), pp. 211–224.
- [56] M. ZERROUKAT, N. WOOD, AND A. STANIFORTH, *A monotonic and positive-definite filter for a Semi-Lagrangian Inherently Conserving and Efficient (SLICE) scheme*, Q. J. Royal Meteorol. Soc., 131 (2005), pp. 2923–2936, <https://doi.org/10.1256/qj.04.97>.
- [57] K. ZHANG, H. WAN, B. WANG, AND M. ZHANG, *Consistency problem with tracer advection in the atmospheric model GAMIL*, Adv. Atmos. Sci., 25 (2008), pp. 306–318.