Islet: Interpolation semi-Lagrangian element-based transport

Andrew M. Bradley[1], Peter A. Bosler[1], and Oksana Guba[1]

[1]Sandia National Laboratories, Albuquerque, New Mexico

Correspondence: Andrew M. Bradley (ambradl@sandia.gov)

Abstract. Advection of trace species, or *tracers*, also called tracer transport, in models of the atmosphere and other physical domains is an important and potentially computationally expensive part of a model's dynamical core. Semi-Lagrangian (SL) advection methods are efficient because they permit a time step much larger than the advective stability limit for explicit Eulerian methods, without requiring the solution of a globally coupled system of equations as implicit Eulerian methods do. Thus, to reduce the computational expense of tracer transport, dynamical cores often use SL methods to advect tracers. The class of interpolation semi-Lagrangian (ISL) methods contains potentially extremely efficient SL methods. We describe a finite-element ISL transport method that we call the Interpolation Semi-Lagrangian Element-based Transport (Islet) method, such as for use with atmosphere models discretized using the spectral element method. The *Islet method* uses three grids that share an element grid: a dynamics grid supporting, for example, the Gauss-Legendre-Lobatto basis of degree three; a physics parameterizations grid with a configurable number of finite-volume subcells per element; and a tracer grid supporting use of *Islet bases*, with particular basis again configurable. This method provides extremely accurate tracer transport and excellent diagnostic values in a number of verification problems.

1 Introduction

An atmosphere model has two parts: the dynamical core, which computes resolved fluid flow and resolved thermodynamics (Lauritzen, 2011); and the subgrid parameterizations, which compute unresolved chemistry and physics processes (Stensrud, 2009). In turn, the dynamical core also has two parts: first, the dynamics solver, which solves the equations of fluid motion; second, the tracer transport solver, which advects trace atmosphere species, or *tracers*, using the air density and flow fields from the dynamics solver.

Because of the large number of tracers in climate models, tracer transport can be computationally very expensive. For example, in the Dept. of Energy's Energy Exascale Earth System Model (E3SM) (E3SM Project, 2018) Atmosphere Model (EAM) version 1 (Golaz et al., 2019), configured with the default 40 tracers, tracer transport takes approximately 75% of the total dynamical core wall clock time and approximately 23% of the total atmosphere model wall clock time on a typical computer cluster (Golaz et al., 2022, Fig. 3). For EAM version 2, we developed a new tracer transport method that is 6.5 to over 8 times faster than in EAM version 1, in the cases of, respectively, low and high workload per computer node (Golaz et al., 2022, Fig. 3). In addition, we developed remap operators to remap data between separate grids for physics parameterizations

and dynamics, permitting the physics parameterization computations to run on a coarser grid and thus 1.6 to 2.2 times faster in version 2 than in version 1 (Hannah et al., 2021; Golaz et al., 2022).

In this article, we describe the Interpolation Semi-Lagrangian Element-based Transport method, with the acronym stylized as "Islet" rather than "ISLET" to minimize distracting all-capitalized text. The Islet method extends those we developed for EAM version 2 by providing the discretizations with natural parameters to trade between computational cost and accuracy in tracer transport and physics parameterizations for a given, fixed dynamical core configuration.

1.1 Problem statement

The mass continuity equation for the air density ρ is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{1}$$

where $\boldsymbol{u}$ is the flow velocity. The tracer transport equation in continuity form and with a source term for a tracer mixing ratio q and corresponding tracer density ρq is

$$\frac{\partial (\rho q)}{\partial t} + \nabla \cdot (\rho q \boldsymbol{u}) = \rho f, \tag{2}$$

where f is a source term for q. Combining Eqs. (1) and (2) gives the advective form of the tracer transport equation:

$$\frac{Dq}{Dt} \equiv \frac{\partial q}{\partial t} + \boldsymbol{u} \cdot \nabla q = f. \tag{3}$$

Because of time splitting in the atmosphere model, our focus is often on the sourceless advection equation,

$$\frac{Dq}{Dt} = 0. \tag{4}$$

At position $\boldsymbol{x}$ and time t_1, the exact solution of (4) in terms of the solution at another time t_0 is

$$q(\boldsymbol{x}, t_1) = q(\boldsymbol{X}(t_0; \boldsymbol{x}, t_1), t_0). \tag{5}$$

$\boldsymbol{X}(t_0; \boldsymbol{x}, t_1)$ is the solution of the ordinary differential equation

$$\frac{d}{dt} \boldsymbol{X}(t; \boldsymbol{x}, t_1) = \boldsymbol{u}(\boldsymbol{X}(t; \boldsymbol{x}, t_1), t), \tag{6}$$

with the initial condition $\boldsymbol{X}(t_1; \boldsymbol{x}, t_1) = \boldsymbol{x}$. In a semi-Lagrangian method, $\boldsymbol{X}(t_1; \boldsymbol{x}, t_1) = \boldsymbol{x}$ is called the *arrival* point and $\boldsymbol{X}(t_0; \boldsymbol{x}, t_1)$ is called the *departure* point. In this article, our focus is on the two-dimensional equations on the sphere. Thus, $\boldsymbol{X}$ and $\boldsymbol{u}$ each have two horizontal components and no vertical component. In various settings, the density variable ρ can be, for example, mass per unit area, layer thickness, or layer pressure increment.

An approximate numerical solution for q of Eq. (4) is said to be *property preserving* if (possibly just a subset of) properties that hold for the exact solution also hold for the approximate one. Equation (5) implies that advection cannot introduce new extrema in the mixing ratio; advection is said to be *shape preserving*. Equation (2) with $f = 0$ implies the global mass is

conserved. Although the focus of this article is not the continuity equation, we note that the Lagrangian form of the continuity equation, Eq. (B4) in Appendix B, implies that the total mass in a Lagrangian parcel, which is a parcel of fluid that moves with the flow, is constant. A final property that is a special case of the shape preserving property relates to coupling a solver for Eq. (4) to a dynamics solver: mass-tracer consistency. This property means that if q is constant in space at time t_0, then it remains constant in space at every other time. In other words, the dynamics solver and transport solver use the same air density. The methods in this article conserve global mass, do not introduce new nodal extrema, and provide mass-tracer consistency when coupled to a dynamics solver.

There is a large body of work on tracer transport methods. As a result, reseachers have developed a large number of accuracy, property preservation, and other diagnostics for use in comparisons of methods (Lauritzen et al., 2012, 2014; Lauritzen and Thuburn, 2012; Lauritzen, 2011; Lauritzen et al., 2015).

Another quality of a transport method is important: its computational efficiency. Computational efficiency is some measure of solution accuracy for a given set of computational resources.

Thus, when developing a new transport method, the objective is to obtain high efficiency, as measured by diagnostic values and computational cost, constrained by the need to couple to specific dynamics and physics grids. Our objective in this article is to extend our highly efficient tracer transport method for EAM version 2 by introducing, first, parameters whose values trade between solution accuracy and computational cost and, second, algorithms associated with these parameters.

Our motivation is to support the computationally efficient simulation of strongly tracer-dependent models such as those of aerosols by enabling extremely highly resolved tracer filamentary structure for a fixed dynamics resolution. A recent report from the National Academies of Sciences, Engineering, and Medicine (Field et al., 2021) includes the recommendation to "explore whether global aerosol optical depth (AOD) distribution is significantly affected by plume-scale effects" and asks: "Are nested grids needed to represent plume processes? What spatial resolution is needed to faithfully represent the radiative forcing and impact outcomes?" In this article, we present algorithms that can help to address these questions.

1.2 Overview of the E3SM Atmosphere Model

EAM uses the High-Order Method Modeling Environment (HOMME) dynamical core (Dennis et al., 2005, 2012). HOMME uses the spectral element method to discretize the equations of atmospheric flow. HOMME's grid consists of horizontally unstructured quadrilateral elements extruded as columns in the vertical dimension. *Element-based* methods permit extremely flexible discretization of domains: for example, E3SM's regionally refined model (RRM) configurations (Tang et al., 2019), in which the atmosphere element grid is refined in a particular region of the earth.

In the horizontal direction, each quadrilateral element has an $n_p \times n_p$ subgrid. HOMME uses the Gauss-Legendre-Lobatto (GLL) spectral element polynomial basis and nodes. The two-dimensional (2D) basis functions are tensor products of one-dimensional (1D) GLL basis functions. n_p is the number of subgrid nodes in a 1D basis function; the degree of the polynomial is $n_p - 1$. Figure 1 shows one spectral element, outlined in blue. The large black dots are the 4×4 tensor product grid of GLL nodes, with $n_p = 4$.
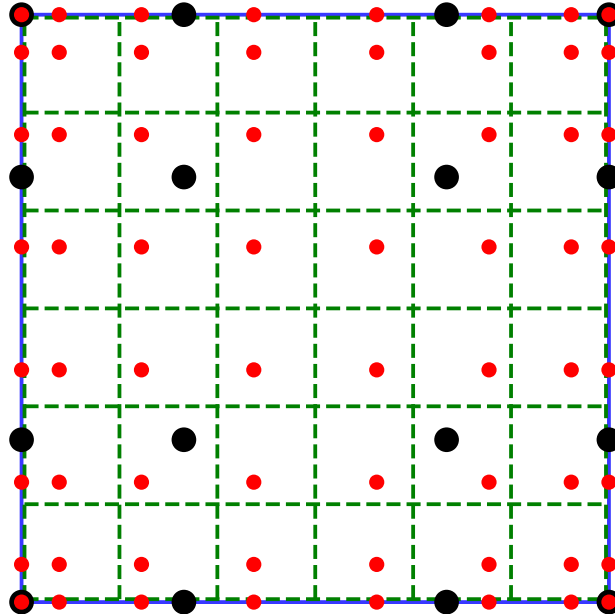
Figure 1. Diagram of subelement grids. One spectral element (blue solid line outlining the full square) with dynamics ($n_p^v = 4$, black large circles), tracer ($n_p^t = 8$, small red circles), and physics ($n_f = 6$, green dashed lines) subelement grids.

The CFL condition for the advection operator in spectral element fluid dynamics scales inversely with the minimum distance between GLL nodes. In turn, this minimum distance scales roughly quadratically in n_p. For example, nodes for $n_p = 6$, 8, and 12 have minimum distances respectively 2.4, 4.3, and 10.0 times smaller than nodes for $n_p = 4$. Thus, developers of a dynamics solver choose a value for n_p that is large enough to provide good accuracy and wave representation but small enough to permit a large time step, i.e., one determined by accuracy rather than stability considerations. EAM always sets $n_p^v = 4$, where we use v to refer specifically to the dynamics solver's value of n_p. We refer to the spectral element grid, including the spectral element nodes, as the *dynamics grid*, and the element grid excluding subgrid nodes as the *element grid*.

For efficiency, the EAM version 2 physics and chemistry parameterizations run on a different grid than the dynamics; we call this grid the *physics grid*. To couple efficiently to the dynamics grid, the physics grid has the same element grid as the dynamics grid but its own subelement grid (Herrington et al., 2019; Hannah et al., 2021). The physics grid divides each horizontal quadrilateral element into a regular $n_f \times n_f$ subgrid of finite-volume quadrilateral subcells. In Fig. 1, the dashed green lines outline the 6×6 physics subgrid, with $n_f = 6$. EAM version 2 uses $n_f = 2$. Compared with EAM version 1, in which the physics and dynamics grids were the same, EAM version 2 has 4/9 as many physics grid points, speeding up the parameterizations part of the model by up to 2.25 times, with a small cost for remapping between dynamics and physics grids. The shared element grid and the element-local remap operators minimize interprocess communication costs.

1.3 Overview of the Islet method

While our tracer transport method is based on spectral elements and, like the physics grid, uses the same element grid as the dynamics grid, it is not restricted by the CFL condition. This important fact motivates making n_p^t, where we use t to refer specifically to the transport solver's value of n_p, a parameter of the tracer transport module and independent of n_p^v, to trade between computational speed (maximum at $n_p^t = n_p^v$) and higher accuracy (increasing with n_p^t). In Fig. 1, the small red circles are GLL nodes for $n_p^t = 8$. We refer to this approach as *tracer transport p-refinement*. In the finite element method, p-refinement means increasing the basis polynomial degree. In the Islet method, we increase n_p^t relative to n_p^v to represent the mixing ratio fields at higher resolution.

In summary, we use one element grid, three subelement grids, and three corresponding parameters: the dynamics grid, n_p^v; the physics grid, n_f; and the tracer transport grid, n_p^t. For efficiency, EAM version 2 already uses two grids, one with $n_p^t = n_p^v = 4$ and the other with $n_f = 2$. This article considers the case $n_p^t > n_p^v$. Figure 1 shows the three subgrids for one spectral element.

The tracer transport solver couples to the dynamics solver through the flow velocity $\boldsymbol{u}$, the air density ρ, and possibly a small subset of tracers, such as specific humidity. It couples to the parameterizations through the tracers. Coupling requires remapping a field between grids. It is natural to give responsibility for these remap procedures to the tracer transport solver module, as the remap operators share many of the same requirements and computations as the transport solver. As in the case of remapping quantities between the physics grid and the dynamics grid in EAM version 2, the shared element grid and element-local remap procedures minimize communication when remapping among the three grids. For brevity, we refer to this overall spectral element, three-grid scheme as the *Islet method*.

1.4 Outline

The rest of this article is structured as follows. The Islet method depends on a number of lower-level algorithms. Section 2 and Appendix A describe these; mathematical details can be skipped on a first reading. Section 3 describes each step of the Islet method over the course of one time step. Descriptions are compact, relying on the details in Sect. 2. Section 4 describes and presents the results of a number of verification problems. Finally, Sect. 5 concludes.

2 Overview of algorithms

This section describes the details of low-level algorithms on which the Islet method depends, associated concepts, and related work for context. Our objective is to isolate the notation for, and details of, these low-level algorithms to this section, then focus on the higher-level details of the Islet method in Sect. 3 without lower-level clutter. Section 2.1 discusses semi-Lagrangian transport, the finite-element interpolation semi-Lagrangian method in particular, and the bases that the Islet method uses. Section 2.2 describes our approach to solving the property preservation problem. Section 2.3 describes algorithms to remap data between grids. Section 2.4 describes the spectral element direct stiffness summation operator.

2.1 Semi-Lagrangian transport

The Islet method uses semi-Lagrangian (SL) transport. In an SL method, following Eqs. (5) and (6), each Eulerian arrival grid point at time t_1 is advected backward or forward in time to its generally off-grid departure point at time t_0. The resulting Lagrangian departure grid then samples the field at time t_0, and by some means uses these data to construct the advected field at time t_1.

In the problem of tracer transport, for which the flow velocity $\boldsymbol{u}$ is provided, SL methods can take much longer time steps than Eulerian methods. Data dependencies are local to a small domain of dependence; thus, unlike an implicit method, SL tracer transport does not require the solution of a global linear equation. This long time step requires many fewer interprocess communication rounds, and sometimes fewer floating point operations, per unit of simulation time than Eulerian methods, making SL methods substantially more efficient than Eulerian methods for tracer transport.

Transport methods often are the composition of a number of operators. Two key operators are the linear advection operator and the nonlinear property preservation operator; we describe ours in Sects. 2.1.4 and 2.2, respectively.

2.1.1 Types of SL methods

For a broad survey of atmosphere tracer transport methods, including a large number of Lagrangian and semi-Lagrangian methods, we refer the reader to Lauritzen et al. (2014, Sect. 2). In addition, Giraldo et al. (2003); Natarajan and Jacobs (2020); Bradley et al. (2019) review types of SL methods. This article focuses on remap-form interpolation methods.

A method in *remap form* directly remaps the tracer field on the Eulerian grid at the previous time step to the Lagrangian grid, in contrast to flux-form methods that integrate over swept regions (Lauritzen et al., 2010; Lee et al., 2016) or characteristic curves (Erath and Nair, 2014). The computational and communication costs of a remap-form method are very nearly independent of time step, whereas a flux-form method's cost grows roughly linearly with the time step.

An interpolation SL (ISL) method directly discretizes the tracer transport equation, Eq. (5): $q(\boldsymbol{x}_i, t_1) = q(\bar{\boldsymbol{x}}_i, t_0)$, where $\boldsymbol{x}_i$ is an Eulerian arrival grid point and $\bar{\boldsymbol{x}}_i$ is its departure, generally off-grid, point at $t_0 < t_1$. q has exact values only at grid points $\boldsymbol{x}_i$; thus, evaluating $q(\bar{\boldsymbol{x}}_i, t_0)$ requires interpolation.

Interpolation is in contrast to *exactly cell-integrated* methods, which accurately integrate the basis of a target (e.g. Lagrangian) element against those of the source; see, e.g., Bosler et al. (2019). (In some cases, an inaccurate cell-integrated method can be interpreted as an interpolation method; see Appendix B for an example.) Exactly cell-integrated methods have substantially greater cost than interpolation methods for three reasons.

First, to obtain smoothness in the integrand, integration is over facets computed by geometric intersection of a target element against source elements; intersection calculations are not needed in interpolation methods. Typically, to minimize computational geometry complexity, departure cell edges are approximated by great arcs rather than flow-distorted curves, limiting the method to second-order accuracy; however, Ullrich et al. (2013) describe a higher-order edge reconstruction that yields a third-order accurate advection method. In contrast, achieving arbitrarily high order in an ISL method's linear advection operator does not entail any additional complexity.

Second, accurate integration has a larger computational cost because it requires sphere-to-reference point calculation and interpolant evaluations at many quadrature points.

Third, an exactly cell-integrated method requires a larger communication volume because all data from a source element are used to integrate against each target basis function.

In trade for these additional costs, exactly cell-integrated methods are locally mass conserving, and the fact that they are L^2 projections can be used to prove stability. Local mass conservation means that one can identify numerical, possibly Lagrangian, fluid parcels on the grid that have constant tracer mass. *Local* is in contrast to *global* mass conservation; the latter means that the mass of the tracer fluid is conserved over the whole domain but not necessarily in any identifiable parcels smaller than the domain. Although an exactly cell-integrated method is locally mass conserving, coupling it to a dynamics solver still generally requires additional measures to obtain mass-tracer consistency.

### 2.1.2 Spectral element ISL transport

Our objective in this work is to use the design freedom provided by giving up local, but not global, mass conservation to maximize computational efficiency. We develop an ISL method that uses the *Islet bases*, summarized in Appendix A and derived in Bradley et al. (2021, Sect. 2 and 3) and a forthcoming article based on that material, to satisfy a necessary condition for stability.

Figure 2(a) illustrates the linear advection step in the Islet method. First, the upper-left element is advected backward in time from time t_1 to time t_0, using velocity data on the $n_p^v = 4$ dynamics grid to advect each dynamics grid point. Second, the $n_p^v = 4$ isoparametric map interpolates the locations of the Lagrangian $n_p^t = 6$ tracer grid points from the Lagrangian $n_p^v = 4$ dynamics grid points. Third, the tracer data at time t_0 in the bottom-right Eulerian element is used to interpolate the tracer mixing ratio at t_0 to each green-starred Lagrangian point. Interpolation uses the 2D tensor-product $n_p^t = 6$ nodal basis. Fourth, these values are then the values in the upper-left Eulerian element at time t_1.

Figure 2(b) shows two bases, the GLL basis, called the *natural* basis when a distinguishing name is needed, and the Islet GLL basis. The ISL method using the natural GLL basis is unstable for the test problem of uniform flow on a uniform grid, but the ISL method using the Islet GLL basis is stable for this problem (Bradley et al., 2021). In this article, all of the algorithms accommodate either basis type, and we use the Islet GLL bases to obtain stability; see Appendix A for details.

Each basis is a *nodal* basis: a basis function has value 1 at one node and 0 at every other node. Thus, each basis function is associated with a node. For example, in Fig. 2(b), the blue basis function is associated with the third node of six. The basis is symmetric; basis function $k \in \{0, \dots, n_p - 1\}$ is the mirror image of basis function $n_p - k - 1$. Thus, the blue and cyan functions are mirror images around reference coordinate 0.

There are two conventional ways to enumerate finite element nodal degrees of freedom and basis functions, where for a scalar field, one degree of freedom is associated with each basis function. First, in this article, by a basis we always mean an *element* basis unless stated otherwise. A basis function in an element basis set has non-zero values only within an element. As a result, nodes in the interior of an element have single-valued field values, while nodes at an element boundary may have multi-valued field values.
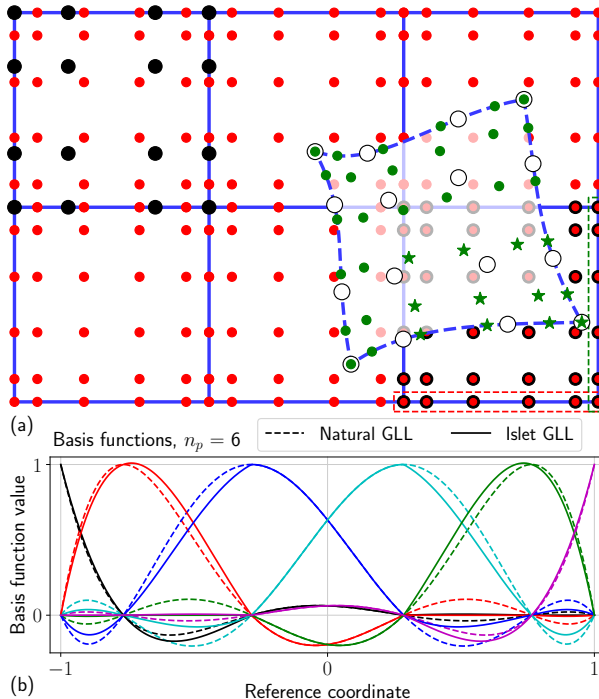
Figure 2. Diagram of the linear advection operation in a finite-element interpolation semi-Lagrangian time step. (a) Six spectral elements (blue solid lines) with an $n_p^t = 6$ tracer grid (red small circles). The black large circles in the upper-left element show the $n_p^v = 4$ dynamics grid. The upper-left Eulerian arrival element is advected backward in time to the right and down, resulting in the Lagrangian departure element (blue dashed outline, open-circle dynamics grid points, green tracer grid points). The domain of dependence of each green-starred tracer grid point is the bottom-right Eulerian element, with interpolation grid points (red small circles) associated with this domain of dependence outlined in black. (b) The six 1D $n_p^t = 6$ element basis functions, a different color for each, for the natural GLL (dashed curves) and Islet GLL (solid curves) bases. In the bottom-right Eulerian element, a set of 1D basis functions is conceptually associated with each of the red horizontal and green vertical rectangles outlining an element side, and a 2D basis function over the element is a tensor product of one 1D basis function from each set.

Second, the viewpoint of *global* continuous degrees of freedom and *global* basis functions always assigns only one degree of freedom of the field to each grid point. The basis function associated with a grid point in the interior of an element is the same as in the element basis. A grid point on the element boundary has a basis function that spans all the elements that share that grid point. This global basis function is the sum of the element basis functions associated with that grid point.

In the case of continuous finite element methods, the two viewpoints are equivalent, and convenience determines which viewpoint is adopted. In the case of a discontinuous finite element method, only the element viewpoint is possible.

The Islet method accommodates both continuous and discontinuous variants; in this article, we present only the continuous variant. We always take the element viewpoint, regardless of the continuity of a field. Thus, for a scalar field, there is one degree of freedom associated with each grid point in an element, and thus n degrees of freedom at a grid point shared by n elements.

Index i is associated with grid point location $\boldsymbol{x}_i$. Again, to be clear, for a grid point at an element edge, multiple index values are associated with the same location $\boldsymbol{x}$. Some steps in the Islet method introduce discontinuities in the field across elements. Section 2.4 describes the operator that restores continuity to a discontinuous field.

2.1.3 Definitions and notation

The tracer mixing ratio degrees of freedom are written q_i; the air density, ρ_i. The vector of all scalar values is bold-faced, e.g., $\boldsymbol{q}$. A subset of these, with the index set given by $\mathcal{E}$, is written $\boldsymbol{q}(\mathcal{E})$. When needed, a superscript letter indicates on which grid the field is approximated: t, tracer; v, dynamics; f, physics.

Arithmetic between vectors applies entry-wise. For two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we write $\boldsymbol{x} \odot \boldsymbol{y}$ for element-wise multiplication and $\boldsymbol{x} \oslash \boldsymbol{y}$ for element-wise division.

We need to define a map between a reference element with coordinates $\boldsymbol{r} \in [-1, 1]^2$ and element e on the manifold with coordinates $\boldsymbol{x}$, e.g., the sphere. Let this map be $\boldsymbol{x} = m_e(\boldsymbol{r})$ and its inverse be $\boldsymbol{r} = m_e^{-1}(\boldsymbol{x})$. Additionally, let $e = E(\boldsymbol{x})$ return the element index for the element that contains e. In this work, if $\boldsymbol{x}$ is on the boundary of multiple elements, $E(\boldsymbol{x})$ can be any of the corresponding element indices without affecting the solution.

Similarly, we need to map between (element-)local and global indices. Local indices are in the set $\{0, \ldots, (n_p^g)^2 - 1\}$, where $g \in \{v, t\}$. Global index i is in element $\mathrm{elm}(i; g) \equiv \lfloor i/(n_p^g)^2 \rfloor$. The corresponding local index is $\mathrm{lcl}(i; g) \equiv i \bmod (n_p^g)^2$. Given the local index j and element index e, the global index is $\mathrm{glb}(j; g) \equiv (n_p^g)^2 e + j$. Let $\mathcal{E}(e; g)$ be the set of global indices for the grid points in element e. We omit g when it is not necessary to specify a grid. For convenience, let $\boldsymbol{q}_e \equiv \boldsymbol{q}(\mathcal{E}(e))$.

Element basis functions and reference-coordinate nodes are indexed locally. Thus, global grid point i corresponds to the 2D tensor-product element basis function $\phi_{\mathrm{lcl}(i; g)}$ and reference-coordinate node $\boldsymbol{r}_{\mathrm{lcl}(i; g)}$.

Occasionally we write the tensor product in the 2D basis explicitly. Let a 1D element basis function set be $\psi_j, j \in 0, \ldots, n_p - 1$. Let $\boldsymbol{r} \equiv (r_1, r_2)$. Then $\phi_i(\boldsymbol{r}) \equiv \psi_{\lfloor i/n_p \rfloor}(r_1) \psi_{i \bmod n_p}(r_2)$. Figure 2(b) shows the $n_p^t = 6$ basis functions ψ_j.

The function $\mathrm{normalize}(\boldsymbol{x}) \equiv \boldsymbol{x}/\|\boldsymbol{x}\|_2$.

Finally, we write the particular map $\boldsymbol{x} = m_e(\boldsymbol{r})$ we use. The map uses just the corner nodes of an element; it is the isoparametric map for a bilinear basis. In element e, let $\mathrm{corner}_e(j; v)$ be the global index of corner j of the element, here referenced to the dynamics grid for concreteness. Let $\phi_j^{\mathrm{bilin}}(\boldsymbol{r})$ be the bilinear basis functions, i.e., the GLL basis with $n_p = 2$. Then

$$\boldsymbol{x}(\boldsymbol{r}) = \mathrm{normalize}\left(\sum_{j=0}^{3} \boldsymbol{x}_{\mathrm{corner}(j; v)}^v \phi_j^{\mathrm{bilin}}(\boldsymbol{r}) \right). \tag{7}$$

This is the map described in Guba et al. (2014, Appendix A).

2.1.4 The linear advection operator

Now we write the steps illustrated in Fig. 2. First, each grid point $\boldsymbol{x}_i^v$ is advected from time t_1 to time t_0 to give

$$\bar{\boldsymbol{x}}_i^v \equiv \boldsymbol{X}(t_0; \boldsymbol{x}_i^v, t_1). \tag{8}$$

In this article, we assume this step is performed nearly exactly, i.e., with numerical error substantially smaller than any other source of error, because this article focuses on the spatial part of the advection discretization. Second, the tracer grid points are computed from the dynamics grid points using the n_p^{v}-basis isoparametric map followed by normalization:

$$\bar{\boldsymbol{x}}_i^{\text{t}} \equiv \text{normalize}\left(\sum_{j \in \mathcal{E}^{\text{v}}(\text{elm}(i;\text{t}))} \bar{\boldsymbol{x}}_j^{\text{v}} \phi_{\text{lcl}(j;\text{v})}^{\text{v}}(\boldsymbol{r}_{\text{lcl}(i;\text{t})}^{\text{t}})\right). \tag{9}$$

Third, an interpolated value from time t_0 is assigned to each advected tracer grid point:

$$e \equiv E(\bar{\boldsymbol{x}}_i^{\text{t}}) \tag{10}$$

$$q_i^{\text{t}}(t_1) = \sum_{j \in \mathcal{E}^{\text{t}}(e)} q_j^{\text{t}}(t_0) \phi_{\text{lcl}(j;\text{t})}^{\text{t}}(m_e^{-1}(\bar{\boldsymbol{x}}_i^{\text{t}})). \tag{11}$$

2.2 Property preservation

Shape preservation and mass conservation on their own are not necessarily difficult to achieve. Rather, the hard problem in the development of an efficient transport method is the consistent coupling of the transport scheme to the dynamical core, or mass-tracer consistency. Mass-tracer consistency means that the transport solver and dynamics solver agree exactly on the air density field. Achieving both mass-tracer consistency and a long transport time step requires redistributing tracer mass.

It is possible to couple a tracer transport method to a dynamical core consistently and preserve properties using only local operations. Lauritzen et al. (2017) describe the coupling of a finite-volume, cell-integrated SL method to the HOMME spectral element dynamical core using Lagrangian grid adjustments, where each adjustment is a function of only data in the adjusted entity's local neighborhood. However, in their method, the time step is limited to a Courant number of at most 1, limiting efficiency.

Our approach to consistent coupling is to use what is sometimes called a *global fixer*, although often this term applies to only the problem of global mass conservation. In Bradley et al. (2019), we describe several classes of methods that we call Communication-Efficient Density Reconstructors (CEDR). Each CEDR can solve a variety of property preservation problems, including shape preservation and global mass conservation. Each CEDR provides a tight and practically meaningful bound on mass redistribution. A solution is assured if the input data meet necessary and sufficient conditions. Depending on the algorithm, a CEDR uses either one or two global reductions or reduction-like communication rounds. See Bradley et al. (2019, Sect. 1.2) for a discussion of how CEDRs relate to previous global fixers, in particular with respect to solution guarantees and number of communication rounds.

Because we use a CEDR for consistent coupling, we are then free to exploit it in other parts of our transport method to maximize efficiency. For example, we can use an interpolation SL method, which is not mass conserving, because the mass will be corrected at the end of each time step.

Exact shape preservation uses mixing ratio bounds computed from each arrival grid point's domain of dependence and does not permit any violation of a domain's nodal extrema. As a result, exact shape preservation clips smooth extrema, limiting the overall transport method to an order of accuracy generally not more than a little above two. Nonetheless, higher order linear

advection is still useful because it increases the absolute accuracy of the overall method even if not its order of accuracy, as examples in Sect. 4 will demonstrate.

2.2.1 CLIPANDASSUREDSUM

In this article, we use the simplest CEDR: CLIPANDASSUREDSUM, subsequently CAAS, Algorithm 3.1 in Bradley et al. (2019).

To explain CAAS, we need to define the weight associated with a grid point, since weights are needed to define the mass. Conceptually, a weight is an integral over the element of the product of the grid point's basis function and the Jacobian determinant of $\boldsymbol{x} = m(\boldsymbol{r})$:

$$\bar{\theta}_i \equiv \int\limits_{-1}^{1} \int\limits_{-1}^{1} \phi_{\mathrm{lcl}(i)}(\boldsymbol{r}) \left| \frac{\partial m_{\mathrm{elm}(i)}(\boldsymbol{r})}{\partial \boldsymbol{r}} \right| \, \mathrm{d}\boldsymbol{r}. \tag{12}$$

Details of the discretization modify this value. Our manifold is the sphere, and we use HOMME's definition of a weight. HOMME discretizes the weight integral using the n_p-basis GLL quadrature. Let

$$v_i \equiv \int\limits_{-1}^{1} \psi_i(r) \, \mathrm{d}r.$$

Because $\phi_{\mathrm{lcl}(i)}(\boldsymbol{r}_{\mathrm{lcl}(j)}) = 1$ when $j = i$ and is 0 for every other value of j, GLL quadrature applied to Eq. (12) gives

$$\theta_i \equiv \sum_{j \in \mathcal{E}(\mathrm{elm}(i))} \left[\phi_{\mathrm{lcl}(i)}(\boldsymbol{r}_{\mathrm{lcl}(j)}) \left| \frac{\partial m_{\mathrm{elm}(i)}(\boldsymbol{r}_{\mathrm{lcl}(j)})}{\partial \boldsymbol{r}} \right| \right] \times$$

$$v_{\lfloor \mathrm{lcl}(j)/n_p \rfloor} v_{\mathrm{lcl}(j) \bmod n_p}$$

$$= \left| \frac{\partial m_{\mathrm{elm}(i)}(\boldsymbol{r}_{\mathrm{lcl}(i)})}{\partial \boldsymbol{r}} \right| v_{\lfloor \mathrm{lcl}(i)/n_p \rfloor} v_{\mathrm{lcl}(i) \bmod n_p}$$

$$\equiv J_i w_i.$$

In the final line, we define

$$J_i \equiv \left| \frac{\partial m_{\mathrm{elm}(i)}(\boldsymbol{r}_{\mathrm{lcl}(i)})}{\partial \boldsymbol{r}} \right| \tag{13}$$

and w_i as the product of the two 1D basis function integrals. In practice, HOMME modifies each node's Jacobian determinant J_i by multiplying it by a constant very close to 1 such that the global sum $\sum_i J_i w_i = 4\pi$; that is, the sum of all the products of w_i and adjusted J_i gives exactly the area of the unit sphere. This is an optional and minor implementation detail, and we neglect this scaling factor subsequently. Note that in general $\boldsymbol{J}$ is discontinuous across elements.

Now we describe CAAS, independent of two sets of details: first, the means by which time-t_1 data are obtained; second, the index set $\mathcal{E}$ over which CAAS is applied. Later, we will specify these details for each application of CAAS. At time t_1 and grid point i, we are given air density ρ_i, preliminary tracer mixing ratio $\bar{q}_i$, and weight θ_i. In addition, we are given a target total

mass b over all $i \in \mathcal{E}$. Finally, we have lower and upper bounds, $q_i^{\min}$ and $q_i^{\max}$, on the target q_i. These are the inputs to CAAS.

The output is a set of modified values $\boldsymbol{q}(\mathcal{E})$ that solves the 1-norm minimization problem

$$\min \sum_{i \in \mathcal{E}} |q_i - \bar{q}_i|$$

$$\text{subject to} \sum_{i \in \mathcal{E}} \theta_i \rho_i q_i = b \qquad \qquad \text{(mass conservation)}$$

$$q_i^{\min} \le q_i \le q_i^{\max} \text{ for } i \in \mathcal{E} \qquad \qquad \text{(shape preservation)}.$$

This problem has a solution, i.e., the constraint set is non-empty, if and only if

$$\sum_{i \in \mathcal{E}} \theta_i \rho_i q_i^{\min} \le b \le \sum_{i \in \mathcal{E}} \theta_i \rho_i q_i^{\max}. \tag{14}$$

For details, see Bradley et al. (2019, Sect. 2).

When there is a solution, there are usually an infinite number, and CAAS efficiently finds one as follows. First, clip the mixing ratios for $i \in \mathcal{E}$:

$$\hat{q}_i \equiv \min(q_i^{\max}, \max(q_i^{\min}, \bar{q}_i)).$$

Let the preliminary mass be

$$\hat{m} \equiv \sum_{i \in \mathcal{E}} \theta_i \rho_i \hat{q}_i.$$

If $\hat{m} = b$, then set $q_i = \hat{q}_i$ for $i \in \mathcal{E}$ and terminate. Otherwise, suppose $\hat{m} < b$. Second, compute the *capacity*:

$$c \equiv \sum_{i \in \mathcal{E}} \theta_i \rho_i (q_i^{\max} - \hat{q}_i).$$

Third, compute the final values for $i \in \mathcal{E}$:

$$q_i = \hat{q}_i + (b - \hat{m}) \frac{q_i^{\max} - \hat{q}_i}{c}.$$

As a check, note that

$$\sum_{i \in \mathcal{E}} \theta_i \rho_i q_i = \sum_{i \in \mathcal{E}} \theta_i \rho_i \hat{q}_i + (b - \hat{m}) \frac{\sum_{i \in \mathcal{E}} \theta_i \rho_i (q_i^{\max} - \hat{q}_i)}{c}$$

$$= \hat{m} + (b - \hat{m}) \frac{c}{c}$$

$$= b.$$

If $b - \hat{m} \le c$, then $q_i \le q_i^{\max}$ for $i \in \mathcal{E}$; these conditions hold if and only if Eq. (14) does. The case $\hat{m} > b$ is similar. Let

$$\boldsymbol{q}(\mathcal{E}) = \text{CAAS}(\mathcal{E}, \boldsymbol{\rho}, \boldsymbol{\theta}, b, \boldsymbol{q}^{\min}, \boldsymbol{q}^{\max}, \bar{\boldsymbol{q}})$$

carry out these steps.

CAAS is a CEDR because the global version, i.e., the problem in which $\mathcal{E}$ is over all grid points, can be implemented using a single global reduction. For details, see Bradley et al. (2019)[Sect. 6].

2.2.2 Infeasible problems

In some situations, the constraint set is empty. There is a sequence of up to two relaxations to the constraints to produce a non-empty constraint set. Algorithm 3.4 in Bradley et al. (2019), RECONSTRUCTSAFELY, which wraps a lower-level algorithm like CAAS, formalizes this sequence. Suppose the original constraint set is empty and $\hat{m} < b$. RECONSTRUCTSAFELY first tries to compute a one-norm-minimal relaxation to $q^{\max}$ such that the maximum value $\max_{i \in \mathcal{E}} q_i^{\max}$ is not exceeded:

$$q(\mathcal{E}) = q^{\max}(\mathcal{E}) +$$
$$\text{CAAS}(\mathcal{E}, \, \rho, \, \theta, \, b - \sum_{i \in \mathcal{E}} \theta_i \rho_i q_i^{\max}, \, \mathbf{0},$$
$$\max_{i \in \mathcal{E}} q_i^{\max} - q, \, q^{\max}).$$

This relaxed problem provides mass-tracer consistency and mass conservation but does not solve the exact shape preservation problem. If this relaxation has an empty constraint set, then RECONSTRUCTSAFELY returns $q(\mathcal{E})$ with uniform values that satisfy the mass conservation constraint.

2.2.3 Local and global problems

We use CAAS to solve problems at two levels: first, within each element separately; second, over the whole grid. In addition, at the global level, there are two natural choices for scalar values in the vectors passed to CAAS: grid point values and integrals over elements. In practice, some local problems may not have assuredly non-empty constraint sets, but the global problem always has a non-empty constraint set.

In the first approach to the global problem, which we refer to as *CAAS-point*, the inputs to CAAS are simply the grid point values. Global and local applications of CAAS are then identical except for the index set $\mathcal{E}$.

In the alternative approach, which we refer to as *CAAS-CAAS*, the global problem is solved over element-level scalar values, and then a second step applies CAAS separately to each element. The global step's task is to provide each element with a sufficient mass adjustment so that each element's local CAAS step has a non-empty constraint set. The global problem's inputs are formulated as follows. Weight e is element e's area: $A_e \equiv \sum_{i \in \mathcal{E}(e)} w_i J_i$. Air density ρ_e is the element's average density. Let the element's mass be $m_e \equiv \sum_{i \in \mathcal{E}(e)} w_i J_i \rho_i$. Then $\rho_e = m_e / A_e$. The minimum mixing ratio is the minimum tracer mass over the element divided by the total mass: $q_e^{\min} \equiv \sum_{i \in \mathcal{E}(e)} w_i J_i \rho_i q_i^{\min} / m_e$, and similarly for $q_e^{\max}$. Finally, the initial value $\bar{q}_e$ is $\bar{q}_e \equiv \sum_{i \in \mathcal{E}(e)} w_i J_i \rho_i q_i / m_e$. This global CAAS step returns element-level mass adjustments. The adjusted mass in an element is then b in the input to the local CAAS step.

2.2.4 Relaxed problems

In the Islet method, multiple local, and one global, property preservation problems are solved in each time step. Not every problem must enforce strict property preservation; strictness is required only in certain outputs, e.g., at the end of a time step. Relaxing the extremal bounds on the mixing ratio grid point values reduces the amount of mass redistribution. More subtly,

solving relaxed element-local problems before solving the strict global problem increases intra-element, and decreases inter-element, mass redistribution, thus increasing the locality of mass redistribution. Finally, problems such as the toy chemistry problem described in Sect. 4.3.2 that are sensitive to finite-precision round-off errors benefit from relaxed bounds because the perturbation to the mixing ratio is decreased. Thus, in some element-local CAAS applications in a time step, for each index i, we widen each pair of bounds by 1% of $q_i^{\max} - q_i^{\min}$ in each direction. The global CAAS-point application uses strict bounds, assuring the tracer field is property preserving at the end of a time step.

2.3 Grid remap

Remapping data among multiple component grids is common in many applications of PDE-based modeling because it is a direct means to permit each component to run in its most efficient configuration. For example, many whole-earth, fully coupled earth system models use different grids for ocean, land, and atmosphere components. Remapping among subcomponent grids is less common. Recent examples include separate physics parameterizations and dynamics grids in the atmosphere (Herrington et al., 2019; Hannah et al., 2021); adaptive mesh refinement (AMR) of a tracer (Chen et al., 2021; Semakin and Rastigejev, 2020); and local vertical refinement in physics parameterizations relative to the shared background vertical grid, the Framework for Improvement by Vertical Enhancement (FIVE) (Yamaguchi et al., 2017; Lee et al., 2020).

In the Islet method, grid remap operators transfer data among the dynamics, tracer, and physics grids. In all of these grid transfers, there is one key property: the linear remap operators use only element-local data. Thus, remap between grids is extremely efficient.

Remapping a field between a GLL grid, dynamics or tracer, and the physics grid is described in detail in Hannah et al. (2021, Sect. 2), and these details are not reproduced here. Remapping from the dynamics grid to the physics grid is simple: a physics grid subcell's value is assigned the average value of the density over the subcell. Remapping from the physics grid to the dynamics grid requires a high-order reconstruction of the data on the physics grid, in addition to maintaining an element-local mass-conservation constraint, and most of the details in Hannah et al. (2021, Sect. 2) focus on this high-order reconstruction.

Remapping a field between the dynamics and tracer grids, in either direction, is simple; each grid hosts a field already in a high-order representation, and thus no high-order reconstruction is needed. The first step is interpolation. Let $\mathcal{I}^{\mathrm{v}\to\mathrm{t}}$ interpolate a field in an element on the dynamics grid, represented by the natural GLL n_p^{v}-basis, to one on the tracer grid, represented by the Islet n_p^{t}-basis, using the natural n_p^{v}-basis functions to interpolate. Let $\mathcal{I}^{\mathrm{t}\to\mathrm{v}}$ do the opposite, using the Islet n_p^{t}-basis functions. We can write these linear operators explicitly using the notation introduced in Sect. 2.1.2. Let $\boldsymbol{f}$ be the grid-point values of a scalar field. Consider element e. Then for local index $i \in \{0, \dots, (n_p^{\mathrm{t}})^2 - 1\}$,

$$(\boldsymbol{f}_e^{\mathrm{t}})_i = \left(\mathcal{I}^{\mathrm{v}\to\mathrm{t}} \boldsymbol{f}_e^{\mathrm{v}}\right)_i \equiv \sum_{j \in \mathcal{E}^{\mathrm{v}}(e)} f_j^{\mathrm{v}} \phi_{\mathrm{lcl}(j;\mathrm{v})}^{\mathrm{v}}(\boldsymbol{r}_i^{\mathrm{t}}),$$

where we have used the notation $\boldsymbol{f}_e \equiv \boldsymbol{f}(\mathcal{E}(e))$. $\mathcal{I}^{\mathrm{t}\to\mathrm{v}}$ is written the same way but with t and v switched.

380　The interpolation of a field f from the dynamics grid to the tracer grid has the useful property that it is conservative within each element e despite not being in all cases an L^2 projection. Consider $\boldsymbol{f}_e^{\mathrm{t}} = \mathcal{I}^{\mathrm{v}\to\mathrm{t}}\boldsymbol{f}_e^{\mathrm{v}}$. Then

$$\sum_{i\in\mathcal{E}^{\mathrm{t}}(e)} w_i^{\mathrm{t}} f_i^{\mathrm{t}} = \sum_{i\in\mathcal{E}^{\mathrm{v}}(e)} w_i^{\mathrm{v}} f_i^{\mathrm{v}}. \tag{15}$$

See Appendix A2 for a proof of Eq. (15).

　　Remapping between grids requires, as a second step, applying CAAS for property preservation; thus, we need grid point
385　weight data on each grid. We define $\boldsymbol{J}^{\mathrm{t}}$ using Eq. (13). On the tracer grid, rather than use Eq. (13), we interpolate the Jacobian determinant values from the dynamics grid,

$$\boldsymbol{J}_e^{\mathrm{t}} = \mathcal{I}^{\mathrm{v}\to\mathrm{t}}\boldsymbol{J}_e^{\mathrm{v}}, \tag{16}$$

for two reasons. First, this operation conserves the area of the element,

$$\sum_{j\in\mathcal{E}^{\mathrm{t}}(e)} w_i^{\mathrm{t}} J_i^{\mathrm{t}} = \sum_{j\in\mathcal{E}^{\mathrm{v}}(e)} w_i^{\mathrm{v}} J_i^{\mathrm{v}},$$

390　by Eq. (15). We describe the second reason when discussing ρ in Sect. 3.1. On the physics grid, we follow a similar procedure, explained in Hannah et al. (2021, Sect. 2.2.1), that is specialized to the finite-volume physics grid. Here again, the area of an element on the physics grid is the same as on the dynamics grid. Thus, all three subgrid definitions of the element area agree on element area values.

2.4　Direct stiffness summation

395　In a spectral element method, most operations are performed independently in each element, often leading to discontinuities in a field across element boundaries. The global direct stiffness summation (DSS) operator (Fischer and Patera, 1989; Dennis et al., 2012) restores continuity. At each grid point on an edge of an element, the multi-valued solution is restored to a single value by weighted summation of contributions from each element sharing the grid point. An element's weight at grid point i is the value $w_i J_i$ normalized by the sum of all contributing elements' values. We will need a generalization of this operator in
400　which an extra factor σ_i is absorbed into the weight in the weighted sum. Let $\delta_{ij}^{\mathrm{grid}}$ be 1 if global indices i and j are associated with the same grid point and 0 otherwise. For global index i, the generalized DSS applied to a preliminary field $\bar{\boldsymbol{y}}$ is written

$$y_i = \mathrm{DSS}(\boldsymbol{\theta},\boldsymbol{\sigma},\bar{\boldsymbol{y}},i) \equiv \frac{\sum_e \sum_{j\in\mathcal{E}(e)} \delta_{ij}^{\mathrm{grid}} \theta_j \sigma_j \bar{y}_j}{\sum_e \sum_{j\in\mathcal{E}(e)} \delta_{ij}^{\mathrm{grid}} \theta_j \sigma_j}, \tag{17}$$

where $\sum_e$ is the sum over all elements. In our use of the generalized DSS, $\boldsymbol{\sigma} = \boldsymbol{\rho}$ and $\bar{\boldsymbol{y}} = \bar{\boldsymbol{q}}$. In the standard DSS, $\boldsymbol{\sigma}$ is all one and $\bar{\boldsymbol{y}} = \boldsymbol{\rho} \odot \bar{\boldsymbol{q}}$. These two use cases give the same value for q_i if $\boldsymbol{\rho}$ is continuous across elements. We use the generalized DSS
405　when we want to make $\boldsymbol{q}$ continuous but leave $\boldsymbol{\rho}$ discontinuous and, in particular, unmodified from its original value.

3 The Islet method

Now that we have described the problem setting and the core algorithms we use, we describe each algorithm that runs in one time step of the Islet method, advancing the simulation from time step n to $n+1$.

As part of an earth system model, the advection equation Eq. (3) has a source term. f is a function of possibly all the variables in a simulation. The physics parameterizations compute f on the physics grid. In practice there are multiple equations of the form Eq. 3 to solve, e.g., 40 in the E3SM version 2 water cycle simulations (Golaz et al., 2022); because they decouple during one transport time step, we can focus on just one in this section.

In summary, first, air density ρ and trajectory data are remapped from the dynamics grid to the tracer grid. Second, $\Delta q \equiv f \Delta t$ is remapped from the physics grid to the tracer grid, where Δt is the physics parameterization time step. Either of f or Δq is sometimes called a *tendency*. Third, the remapped tracer tendencies are added to the tracer-grid tracers. Fourth, the tracers are advected on the tracer grid. Fifth, tracer states are remapped to the physics and, optionally, dynamics grids.

3.1 Algorithms

There are a number of algorithmic steps in a time step. To organize these algorithms, we name each algorithm using the format **Step::algorithm-name** and sometimes **Step::algorithm-name::sub-algorithm-name**.

Step::density-d2t. Interpolate $\boldsymbol{J} \odot \rho^{n+1}$ from the dynamics grid to the tracer grid. In each element e, compute

$$(\boldsymbol{\rho}_e^{\mathrm{t}})^{n+1} = (\mathcal{I}^{\mathrm{v}\to\mathrm{t}}(\boldsymbol{J}_e^{\mathrm{v}} \odot (\boldsymbol{\rho}_e^{\mathrm{v}})^{n+1})) \oslash \boldsymbol{J}_e^{\mathrm{t}}.$$

This grid transfer conserves $\boldsymbol{J}_e \odot \rho_e^{n+1}$ by Eq. (15). $\boldsymbol{J}^{\mathrm{t}} \odot (\boldsymbol{\rho}^{\mathrm{t}})^{n+1}$ is not continuous across element boundaries because $\boldsymbol{J}^{\mathrm{t}}$ is not, but continuity is not needed. Equation (16) implies that if $\boldsymbol{\rho}_e$ is constant on the dynamics grid, then it is constant on the tracer grid, too. Mapping a constant air density exactly is not necessary, but since it is possible, we do; this property is the second reason to define $\boldsymbol{J}^{\mathrm{t}}$ according to Eq. (16).

Step::tendency-f2t. Map the tracer tendency Δq^n from the physics grid to the tracer grid. This step involves multiple sub-algorithms.

Step::tendency-f2t::bounds. In each element e, compute and store the minimum and maximum values of $(q_e^{\mathrm{f}})^n$, subsequently *extrema*. Let an element's neighborhood contain itself and every other element that shares a vertex with it. Augment an element's extrema with the extrema over its neighborhood. Finally, augment element e's extrema again with the extremal values of the tracer-grid mixing ratio state $(q_e^{\mathrm{t}})^n$. This final extrema update assures that if $\Delta(q_e^{\mathrm{f}})^n = 0$, then $(\bar{q}_e^{\mathrm{t}})^n$ is unmodified in **Step::tendency-f2t::CAAS**. These final extrema are the bounds used in subsequent property preservation corrections.

Step::tendency-f2t::linear-remap. In each element e, apply the linear, element-local, conservative, panel-reconstruction remap operator described in Hannah et al. (2021, Sect. 2.2.3) to map the tendency from the physics grid to the tracer grid. In the Islet method, but unlike in Hannah et al. (2021), the basis used in the mass matrix of the L^2 projection is the Islet basis. From the previous step's application of **Step::density-d2t**, we have ρ_e^{t}. The operator uses this quantity and $\rho_e^{\mathrm{f}} \odot \Delta(q_e^{\mathrm{f}})^n$ to compute $\rho_e^{\mathrm{t}} \odot \Delta(q_e^{\mathrm{t}})^n$.

Step::tendency-f2t::CAAS. In each element e, apply CAAS, described in Sect. 2.2.1, to $(\bar{q}_e^{\mathrm{t}})^n \equiv (q_e^{\mathrm{t}})^n + \Delta(q_e^{\mathrm{t}})^n$, using the weight vector $\boldsymbol{\theta}_e^{\mathrm{t}} \equiv \boldsymbol{w}_e^{\mathrm{t}} \odot \boldsymbol{J}_e^{\mathrm{t}}$ and air density $(\boldsymbol{\rho}_e^{\mathrm{t}})^n$. Target mass b is set to the current total element mass because **Step::tendency-f2t::linear-remap** is conservative; thus, no adjustment to the total element mass is needed. The same upper and lower bounds apply to each GLL node in the element. In this step, the bounds that **Step::tendency-f2t::bounds** computed are relaxed in each direction by 1% of the difference between upper and lower bounds, as discussed in Sect. 2.2.4. The exact bounds will be enforced in **Step::CEDR::global**. The constraint set of mass conservation and bounds nonviolation is non-empty because the constant mixing ratio is a solution, as explained in Hannah et al. (2021, Sect. 2.3).

Step::tendency-f2t::DSS. At this point, $(\bar{q}^{\mathrm{t}})^n$ is discontinuous across element boundaries. Neither order of accuracy nor property preservation requires continuity, but we find that continuity improves the solution quality. $(\rho^{\mathrm{t}})^n$ can remain discontinuous. Thus, we apply the generalized DSS, Eq. (17), to $(\bar{q}^{\mathrm{t}})^n$, using $(\rho^{\mathrm{t}})^n$, to obtain the continuous field $(\hat{q}^{\mathrm{t}})^n$.

Step::advect-interp. Compute and apply the linear advection ISL operator described in Sect. 2.1.4. This step involves two substeps: computing the grid point trajectories and computing the interpolants.

Step::advect-interp::trajectory. Compute Eq. (8). The dynamics component supplies velocity data at the dynamics GLL grid points. Any of a number of algorithms can compute departure points at time n backward in time from dynamics-grid arrival GLL points at time $n+1$. The Islet method takes as input these departure points as 3D Cartesian departure points. Next, compute Eq. (9). In each element, the natural GLL interpolant $\mathcal{I}^{\mathrm{v}\to\mathrm{t}}$ is applied to each of the Cartesian components separately to provide departure points on the tracer grid. This procedure implies that adjacent elements compute identical departure points at shared boundaries. Finally, for simplicity in the subsequent sphere-to-reference map computations, the 3D Cartesian points are normalized to the sphere. For $n_p^{\mathrm{v}} = 4$, the n_p^{t}-basis departure points are obtained at order of accuracy 4.

Step::advect-interp::mixing-ratio. Compute Eq. (10). Each departure tracer-grid GLL node is mapped to the containing element, subsequently the *source* element. Details of finding the source element depend on host-model implementation details and are omitted here; possibilities include octree search, $O(1)$ arithmetic for quasiuniform cubed-sphere element grids having certain reference-to-sphere maps, and search within a predefined element neighborhood whose size is proportional to maximum wind speed times advection time step. Then the corresponding reference coordinates within the source element are computed using Newton's method. Next, compute Eq. (11). The mixing ratio value is computed at the departure point using the Islet basis interpolant and the source element's $(\hat{q}^{\mathrm{t}})^n$ values. In addition, the source element's stored extrema are associated to this point as bounds. Finally, the mixing ratio value and bounds are assigned to the target GLL node on the arrival tracer grid. Departure points and interpolant weights are calculated once and then are reused for each tracer.

Step::CEDR. Apply Communication-Efficient Density Reconstructors (CEDR) on the tracer grid, first to each element and then globally. In this work we use CAAS, described in Sect. 2.2.1. Node weights are $\boldsymbol{w}^{\mathrm{t}} \odot \boldsymbol{J}^{\mathrm{t}}$.

Step::CEDR::local. First, apply the element-local CAAS algorithm. This step is not required, but it reduces the amount of global mass redistribution in **Step::CEDR::global**, is computationally inexpensive, involves no interprocess communication, and thus is worth including. As in **Step::tendency-f2t::CAAS**, we relax bounds in each direction by 1% of the difference between upper and lower bounds. Unlike in **Step::tendency-f2t::CAAS**, in this application of the element-local CAAS algorithm, any two target GLL nodes in an element may have different bounds; the bounds depend on the source element for the

target GLL node, as detailed in **Step::advect-interp::mixing-ratio**. At this point, the global mass has not yet been corrected; thus, this local CAAS application's mass constraint is to maintain the element's current tracer mass. The constraint set is not assuredly non-empty. Thus, RECONSTRUCTSAFELY, described in Sect. 2.2.2, wraps the call to CAAS.

Step::CEDR::global. Apply CAAS-point, described in Sect. 2.2.3, on the global tracer grid. The exact, rather than relaxed, bounds are applied to each node. The inputs to CAAS-point are as follows: the air density $(\boldsymbol{\rho}^{\mathrm{t}})^{n+1}$; the weights $\boldsymbol{\theta} \equiv \boldsymbol{w}^{\mathrm{t}} \odot \boldsymbol{J}^{\mathrm{t}}$; $b = \sum_i w_i^{\mathrm{t}} J_i^{\mathrm{t}} (\rho_i^{\mathrm{t}})^n (\hat{q}^{\mathrm{t}})^n$, the global tracer mass after the tendency update; the mixing ratio bounds computed in **Step::tendency-f2t::bounds**; and the current mixing ratio values from **Step::CEDR::local**. Let the output values be $(\boldsymbol{q}^{\mathrm{t}})^{n+1}$.

Continuity across element boundaries was restored to the mixing ratio field in **Step::tendency-f2t::DSS**. Subsequent steps maintain it in exact arithmetic. However, in finite precision, continuity in $(\boldsymbol{q}^{\mathrm{t}})^{n+1}$ holds only to a little above machine precision and not exactly. This numerical discontinuity does not grow in time because, in each time step, both **Step::tendency-f2t::DSS** and **Step::advect-interp::mixing-ratio** restore exact continuity in finite precision, although only one restoration per time step is necessary to prevent growth of the numerical discontinuity. No step of the overall algorithm is sensitive to this small and roughly temporally constant numerical discontinuity.

At this point, the time step is complete; the remaining computations remap the tracer grid data between grids.

Step::state-t2f. Remap the mixing ratio state to the physics grid. This is a purely element-local operation. First, the linear operator described in Hannah et al. (2021, Sect. 2.2.1) is applied to the tracer density. Second, the element-local CAAS algorithm is applied on the physics grid, with the extremal mixing ratio values in the element on the tracer grid as the bounds. For the same reason as in **Step::tendency-f2t::CAAS**, the constraint set is assuredly non-empty.

Step::state-t2v. The dynamics solver needs one or more mixing ratios on the dynamics grid, e.g., specific humidity. In addition, in our numerical results in Sect. 4, we compute all errors, except as indicated, on the dynamics grid, so we use this step to obtain those errors. First, in an element, $\mathcal{I}^{\mathrm{t} \to \mathrm{v}}$ interpolates the tracer-grid mixing ratio to the n_p^{v}-basis. Second, the element-local CAAS algorithm is applied on the dynamics grid to preserve shape and conserve mass; the details are as in **Step::state-t2f** but with GLL nodes instead of finite-volume subcells. The result is $(\boldsymbol{q}^{\mathrm{v}})^{n+1}$. Third, the standard DSS is applied to $(\boldsymbol{\rho}^{\mathrm{v}})^{n+1} \odot (\boldsymbol{q}^{\mathrm{v}})^{n+1}$, where $(\boldsymbol{\rho}^{\mathrm{v}})^{n+1}$ is the continuous air density from the dynamical core, to obtain continuous tracer density and mixing ratio fields.

Step::state-v2t. In verification problems in Sect. 4, we need to remap a mixing ratio initial condition from the dynamics grid to the tracer grid. The algorithm is the same as **Step::state-t2v** except that the DSS follows the procedure in **Step::tendency-f2t::DSS** because the air density on the tracer grid is and remains discontinuous at element boundaries.

If $n_p^{\mathrm{t}} = n_p^{\mathrm{v}}$, then tracer transport p-refinement (TTPR) is not enabled. In this case, identity maps replace a subset of the algorithms described in this subsection: **Step::density-d2t**, the interpolation part of **Step::advect-interp::trajectory**, **Step::state-t2v**, and **Step::state-v2t**. When describing numerical experiments, we indicate when TTPR is not enabled.

4 Numerical results

This section presents results for a number of verification problems. Except in Sect. 4.3, the equation is the sourceless advection equation, Eq. (4). Two-dimensional, time-dependent flow, $\boldsymbol{u}(\boldsymbol{x}, t)$, is prescribed on the sphere.

In most figures, we show results for $n_p^{\mathrm{t}} = 4, 6, 8, 9, 12$. The $n_p^{\mathrm{t}} = n_p^{\mathrm{v}} = 4$ case provides a reference because it does not use the TTPR algorithms. The $n_p^{\mathrm{t}} = 6$ case has the smallest value of n_p^{t} providing order of accuracy (OOA) greater than 2, in this case 4. The $n_p^{\mathrm{t}} = 8$ case provides OOA 5 and has four times as many nodes as the $n_p^{\mathrm{t}} = 4$ basis in two dimensions. The $n_p^{\mathrm{t}} = 9$ case provides OOA 6. Finally, the $n_p^{\mathrm{t}} = 12$ case provides OOA 8 and has four times as many nodes as the $n_p^{\mathrm{t}} = 6$ basis.

Tests follow the procedures detailed in Lauritzen et al. (2012). Results can be compared with those from many models described in Lauritzen et al. (2014). We refer to these articles frequently and thus abbreviate them as TS12 ("test suite") and TR14 ("test results"), respectively. Initial conditions are generated on the dynamics grid. Similarly, error diagnostics are computed on the dynamics grid in most cases; we state the exceptions when they occur. In most cases we omit results for simulations without property preservation, as tracer transport modules in earth system models are expected to be property preserving. We have not attempted to make this section self-contained, as describing details of the large number of verification problems would take too much space. We recommend the reader not familiar with these problems read TS12. In addition, we refer to specific figures in TR14 and sometimes TS12 so the reader can compare our results with those from previously documented methods.

We briefly summarize the key characteristics of the verification problems. There are two prescribed flows: a nondivergent one and a divergent one. Each prescribes a flow that lasts for $T = 12$ days and such that at time T, the exact solution is the same as the initial condition. This 12-day prescribed flow can be run for multiple cycles to lengthen the simulation. The nondivergent flow creates a filament of maximum aspect ratio at time $T/2$. The divergent flow tests treatment of divergence. There are four initial conditions that share the feature of placing two circular shapes at two points along the equator: the C^∞ Gaussian hills, the C^1 cosine bells, the correlated cosine bells used in the mixing diagnostic, and the discontinuous slotted cylinders. Assessing the behavior of a transport method on tracers having various degrees of smoothness is important because atmosphere tracers can be smooth or nonsmooth.

This article does not study time integration methods to generate the dynamics-grid departure points. Thus, to remove temporal errors due to time integration algorithms, we use an adaptive Runge-Kutta method (Dormand and Prince, 1980; Shampine and Reichelt, 1997) with a tight tolerance (10^{-8}, with an exception noted later) to integrate trajectories very accurately. All tests for $n_p^{\mathrm{t}} > 4$ use TTPR with $n_p^{\mathrm{v}} = 4$ unless we state otherwise. Because $n_p^{\mathrm{v}} = 4$ in these verification problems, the $n_p^{\mathrm{t}} = 4$ configuration does not use TTPR.

As we discussed in Sect. 2.2.3, there are two natural approaches when applying CAAS globally, which we refer to as CAAS-CAAS and CAAS-point. CAAS-CAAS tends to give slightly more accuracy than CAAS-point. However, it does not permit the bound relaxations in the element-local CAAS applications that improve the solution quality when simulating source terms sensitive to round-off errors, as discussed in Sect. 2.2.4. However, because it provides slightly more accurate results, we use

CAAS-CAAS for $n_p^t = 4$, except where noted, while using CAAS-point for all other n_p^t values, thus maximizing the accuracy obtained with $n_p^t = 4$ to provide the best baseline ($n_p^t = n_p^v = 4$) accuracy.

The air density on the dynamics grid, $\boldsymbol{\rho}^v$, is not needed for the advection of mixing ratios, but it and its remapped versions are needed for property preservation. In these verification problems, we have no independent computation of density $\boldsymbol{\rho}^v$, as occurs when transport is coupled to a full dynamical core. For the nondivergent flow problem, we could set $\boldsymbol{\rho}^v$ to a constant, but for the divergent flow problem, a constant is incorrect. Thus, within our standalone verification code, we need a means to compute $\boldsymbol{\rho}^v$ as a surrogate for a dynamics solver. Appendix B describes the linear advection part of computing $\boldsymbol{\rho}^v$. After the linear advection operator is applied, because it is not mass conserving, the density is corrected by adding $\Delta m / a_{\text{total}}$ to each grid point, where Δm is the global mass discrepancy after advection and a_{total} is the total area of the grid. Negative density does not occur in these verification problems. The algorithm for $\boldsymbol{\rho}^v$ has OOA 2 for $n_p^v = 4$.

We use a quasiuniform, equiangular cubed-sphere element grid. Let a cube face of the cubed-sphere element grid have $n_e \times n_e$ elements. Each element has an $n_p^v \times n_p^v$ tensor grid of GLL nodes and thus $n_p^v - 1$ intervals between GLL nodes along each direction of an element. GLL nodes are mapped to the sphere using the isoparametric map for a bilinear element, as described in Appendix A of Guba et al. (2014) and Eq. (7). Long tracer time steps correspond to $6n_e$ steps per T; short, $30n_e$. For the test flows, these correspond to, respectively, approximately 5.5 and 1 times the maximum Courant number. These are the same time step settings as the two CSLAM (Lauritzen et al., 2010) model configurations used in TR14. Essentially all SL methods, particularly when given exact trajectory data, exhibit greater error with smaller time step, e.g., CSLAM in TR14. This is because the only source of error, given exact trajectories, is the remap error. Smaller time steps correspond to more remaps to reach a fixed simulation time.

Figures 3–9 and Fig. 15 show convergence plots, and we explain the format of these figures here. A curve's marker corresponds to n_p^t, as listed in the legend. To maximize font size and minimize notational clutter in figures, in figures we use n_p rather than n_p^t and omit "="; e.g., "$n_p^t = 8$" is written "n_p 8". Additionally, the legend omits "n_p" entirely. The x-axis is the average dynamics-grid node spacing at the equator in degrees for a cubed-sphere grid with north and south cube faces centered at the poles. Thus, for example, $n_e = 5$ and $n_p^v = 4$ correspond to the resolution

$$\frac{360°}{4 \text{ cube faces}} \cdot \frac{1 \text{ cube face}}{5 \text{ elements}} \cdot \frac{1 \text{ element}}{(4-1) \text{ intervals}} = \frac{6°}{\text{interval}}.$$

The y-axis is $\log_{10}$ of the relative error, with the norm or norms indicated in each figure. The title of a plot lists details of the configuration: the test flow, the initial condition (IC), the time step (short or long), and, if property preservation is disabled, an extra line stating that. Many plots have a dotted line showing the slope for OOA 2.
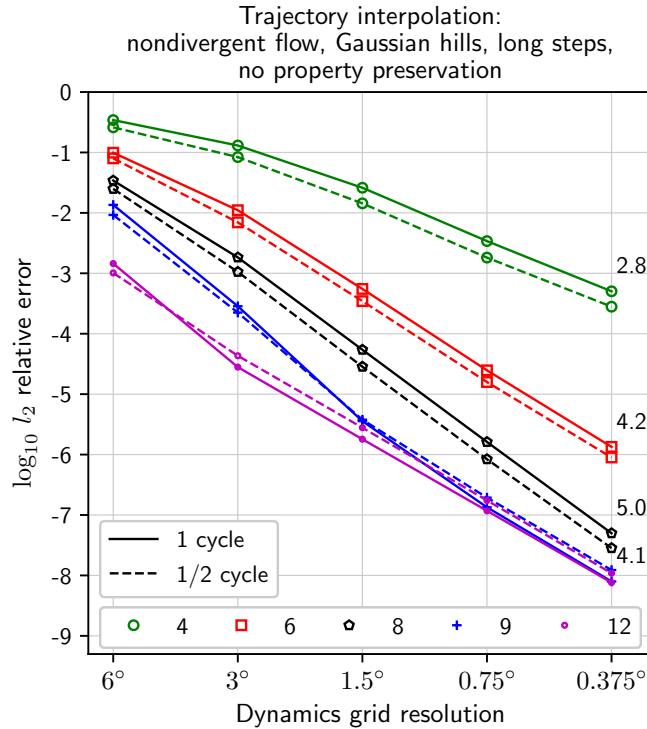
Figure 3. Comparison of relative errors calculated at the test simulation's midpoint time of 6 days (1/2 cycle, dashed lines) and endpoint time of 12 days (1 cycle, solid lines). Each number at the right side of the plot is the empirical OOA computed using the final two points of the 1-cycle result.

4.1 Accuracy for C^∞ and C^1 tracers

4.1.1 Accuracy limited only by trajectories

The first experiment tests the OOA limit due to computing trajectories on the tracer grid using velocities from the dynamics grid. Because the dynamics grid uses $n_p^v = 4$ and $\mathcal{I}^{v \to t}$ uses the natural GLL basis, we expect this OOA limit to be 4. Property preservation is turned off to expose this OOA limit.

Sometimes verification problems have unexpected features that interact with a method to produce higher-accuracy solutions than would occur in a more realistic problem. Of particular concern is the symmetry of the flow in time around the midpoint time, 6 days. To be sure our trajectory interpolation procedure is not interacting with this symmetry to produce artificially higher accuracy, in this experiment we compute the solution error at the midpoint time as well as the final time.

The test uses the nondivergent flow, the Gaussian hills IC, and the long time step. The midpoint reference solution is computed using one 6-day step and the natural GLL basis. The time integrator's relative error tolerance is set to 10^{-14} rather

than its usual 10^{-8} for this step. One remap step with the natural GLL basis provides a midpoint solution much more accurate than the time-stepped case, thus serving as an appropriate reference.

Figure 3 shows the results. The dashed curves show the errors measured at the midpoint, half of a cycle of the 12-day problem; the solid curves, the usual endpoint. The numbers at the right side of the plot show empirical OOA computed using the final two points of the one-cycle results. Numbers for the half-cycle curves are omitted because each pair of curves has almost exactly parallel lines for resolution at least as fine as $0.75°$. In addition, the empirical OOA for $n_p^t = 12$ is omitted because the curves nearly overlap those for $n_p^t = 9$. For fine enough resolution, all curves should have empirical OOA limited to 4, but for $n_p^t = 8$, the full convergence regime is not reached in this plot, thus giving OOA 5 for $n_p^t = 8$. The $n_p^t = 4$ curves have OOA less than 4 due to a spatial OOA limit of 2 in the full convergence regime. We see OOA at about 4 for $n_p^t = 9$ and 12. The $n_p^t = 9$ curve shows at coarse resolution higher OOA, governed by the spatial error, and then a drop to 4 as the temporal error becomes dominant. Importantly, the half- and full-cycle errors are very close for each value of n_p^t, demonstrating that the endpoint error metrics are valid measurements for the Islet method.

4.1.2 Accuracy data for other standard configurations

The next figures show accuracy for standard configurations used in TR14. Although we explain how each figure can be compared with corresponding ones in TR14, these figures also stand on their own as simply convergence plots for various test cases.
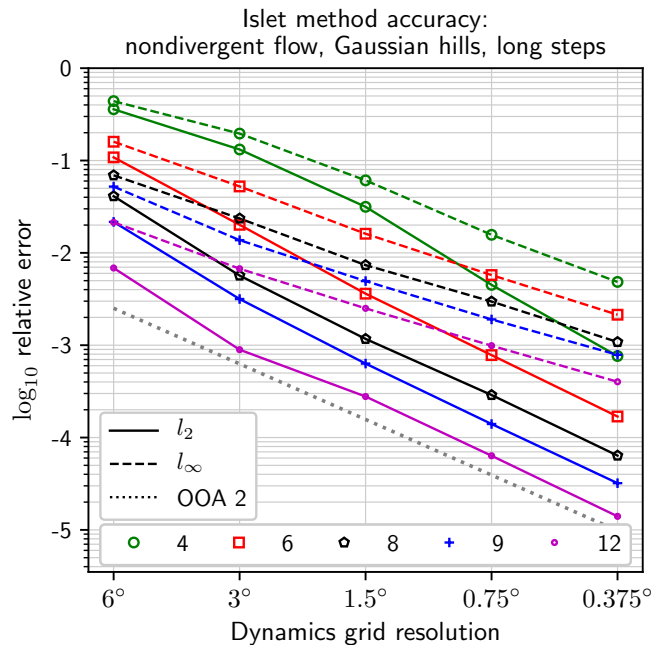
Figure 4. Accuracy diagnostic. Compare with Figs. 1, 2 in TR14.
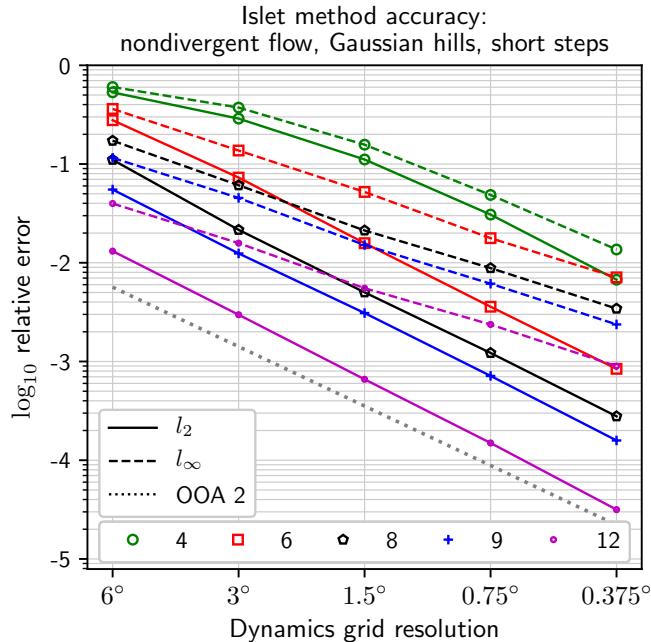
Figure 5. Accuracy diagnostic. Compare with Figs. 1, 2 in TR14.

Figures 4 and 5 can be compared with Figs. 1 and 2 in TR14. They evaluate error on an infinitely smooth IC. The Islet method with a high-order basis compares extremely favorably with the methods in TR14. For example, the most accurate shape-preserving method in TR14 for the nondivergent flow with the Gaussian hills IC is HEL-ND-CN1.0, by a substantial

595   margin (cyan curves in Fig. 1, bottom right, of TR14). The $n_p^t = 12$ Islet scheme with the long time step, Fig. 4, is approximately three times more accurate than HEL-ND-CN1.0 in the l_2 norm at resolution $0.375°$ and approximately twice as accurate at resolution $3°$. Yet HEL-ND is, quoting TR14, an "unphysical" method. It is run for comparison with the practically useful HEL scheme. After HEL-ND, the next most accurate method in the l_2 norm at $0.375°$ resolution is CSLAM-CN5.0. The Islet method with the long time step, the same as that of CSLAM-CN5.0, is at least as accurate for $n_p^t \geq 8$. With the short time step,

600   the same as that of CSLAM-CN1.0, the Islet method is at least as accurate as CSLAM-CN1.0 for $n_p^t \geq 6$. At $3°$ resolution, no method other than HEL-ND-CN1.0 provides l_2 norm below 10^{-2}; the Islet method does for $n_p^t \geq 8$ with the long time step and $n_p^t \geq 10$ (only $n_p^t = 12$ is shown) with the short time step.

Now, we must remind the reader that, in this article, we use nearly exact trajectories on the dynamics grid because the description of practical trajectory methods is outside the scope of this article. However, first, many of the schemes in TR14 use

605   nearly exact trajectories, as well, e.g., CSLAM. Second, when using a practical trajectory algorithm, highly accurate, even if not exact, trajectories are possible because a trajectory over a tracer time step can be computed from multiple dynamics time steps. Thus, the use of exact trajectories is only slightly unrealistic. Third, the diagnostic values in Sect. 4.1.3, 4.1.4, and 4.3.2 are roughly independent of temporal errors.

23
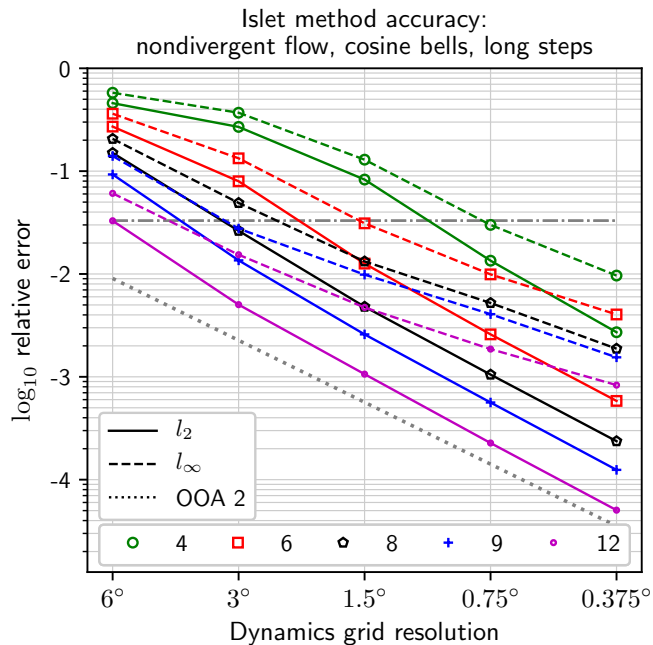
Figure 6. Accuracy diagnostic. Compare with Fig. 3 in TR14.

Figures 6 and 7 provide data that can be compared with the top panel of Fig. 3 in TR14. They evaluate error on a C^1 IC. The horizontal dash-dotted line provides the relative l_2-error-norm value of 0.033 by which the "minimal resolution" diagnostic value is determined; the coordinate of the intersection between the l_2-norm curve and this reference line is the value. A larger value is better. For example, with a long time step, for $n_p^t = 8$, this value is a little coarser than $3°$; for $n_p^t = 12$, approximately $6°$. For comparison, no model in TR14 reports a value larger than $2.5°$.

Figures 8 and 9 provide data that can be compared with the top two panels in Fig. 16 of TR14, given the minimum resolutions plotted in Fig. 3 of TR14. They are like Figs. 6 and 7, but here the divergent flow is used.

4.1.3   Filament diagnostic

Figure 10 shows results for the filament diagnostic described in Sect. 3.3 of TS12, to compare with Fig. 5 in TR14. The two dynamics grid resolutions are as prescribed in TR14. We used the code distributed with TS12 to compute the results. The diagnostic uses the nondivergent flow and the cosine bells IC. In this test, the midpoint solution is analyzed to determine the quality of the filamentary structure. See Fig. 13 for an illustration of the filamentary structure at the simulation midpoint, although with the slotted cylinders IC. For each possible value τ of the tracer mixing ratio at the initial time, the area over which the mixing ratio is at least τ at the midpoint time is computed. For the cosine bells IC, $\tau \in [0.1, 1]$. The diagnostic is then this area divided by the correct area, which for any nondivergent flow is the area at the initial time. The perfect diagnostic value is 100% for all $\tau \in [0.1, 1]$ and 0 otherwise. In each plot in Fig. 10, the x-axis is τ and the y-axis is the diagnostic value.
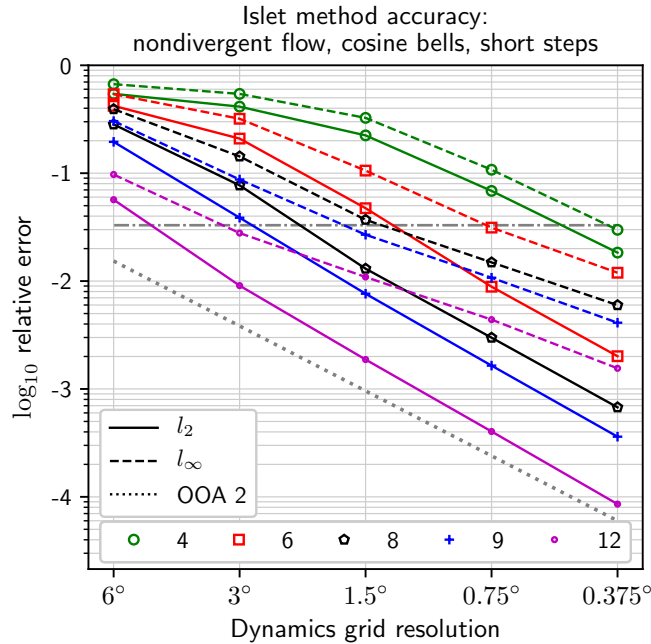
Figure 7. Accuracy diagnostic. Compare with Fig. 3 in TR14.

The diagnostic is computed for the dynamics-grid resolutions and time step lengths listed in the legend. Note that the y-axis limits are tighter with increasing n_p^t.

A subtlety with this diagnostic is that the area calculation must use the quadrature method of the discretization. On the dynamics grid, the resulting curve is noisier than is implied by the underlying solution on the tracer grid. Thus, in Fig. 10, we show the diagnostic as computed on the dynamics grid in the top row of plots; on the tracer grid, in the bottom row.

There is no summary number that can be compared directly with the results in Fig. 5 of TR14, but, visually, the curves for $n_p^t \geq 8$, the resolution 1.5°, and on the tracer grid are among the best of those in TR14 at the resolution 1.5°.

4.1.4 Mixing diagnostic

Figures 11 and 12 show results for the mixing diagnostics described in Sect. 3.5 of TS12, to compare with Figs. 11–14 in TR14. Again the two dynamics grid resolutions are as prescribed in TR14.

The test uses the nondivergent flow with two ICs. The diagnostics assess preservation of the nonlinear correlation of two tracers; one mixing ratio is a nonlinear function of the other. The mixing ratio of each tracer is on an axis, cosine bells on the x-axis and the correlated cosine bells field on the y-axis. Each dot is a grid-point sample from the dynamics grid of the two mixing ratios. Like the filament diagnostic, the analysis is done at the solution midpoint rather than the endpoint. Also as for the filament diagnostic, we used the code distributed with TS12 to compute the results. n_p^t and the time step length are printed
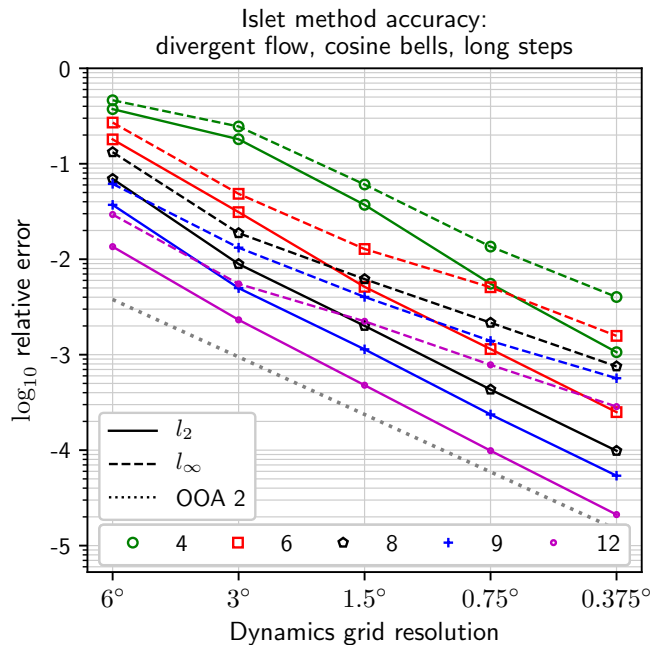
Figure 8. Accuracy diagnostic. Compare with Fig. 16 in TR14.

in each plot. The diagnostic values l_r and l_u, explained in a moment, are also printed in each plot; the parenthesized "(v)" suffix means the value is measured on the dynamics grid, while the un-suffixed values are measured on the tracer grid.

At the initial time, all points are on the curve. Perfect nonlinear correlation corresponds to staying on the upper curve as the simulation proceeds. Points that drop into the convex hull of the starting points can be interpreted as physical mixing of air parcels, since such mixing results in a linear combination of two points on the curve. The diagnostic value l_r measures this type of error; smaller is better. In Figs. 11–14 in TR14, the smallest value of l_r at $1.5°$ among the property-preserving methods is 2.15×10^{-4}, by the UCISOM-CN5.5 method, except for a value of 0 by HEL-ND, which, again, cannot be used in practice. For the long time step, the Islet method gives at least as small a value for $n_p^{\mathrm{t}} \geq 6$; for the short time step, $n_p^{\mathrm{t}} \geq 8$.

Values outside the triangle are overshoots, possible only if the method is not strictly shape preserving. Since our method is, this diagnostic value, l_o, is always 0 and is not displayed in the figures.

Values outside the convex hull cannot be described as physical mixing of parcels and thus are purely numerical artifacts of a method. The corresponding diagnostic is l_u, and again smaller is better. This diagnostic is more difficult to compare than l_r because very dissipative methods tend to have a large value of l_r and consequently a very small value for l_u. In contrast, a very accurate method, for which l_r is small, can have a larger l_u value than a very dissipative method. One means of comparison is to consider the best l_u values among methods that obtain, say, $l_r \leq 5 \times 10^{-4}$. In Figs. 11–14 in TR14, the smallest value of l_u at $1.5°$ under this restriction is 0, obtained by the HEL-CN1.0 and HEL-CN5.5 methods. These HEL variants are practically usable, unlike HEL-ND, and are designed to preserve tracer correlations exactly. Other than the HEL methods, the next best
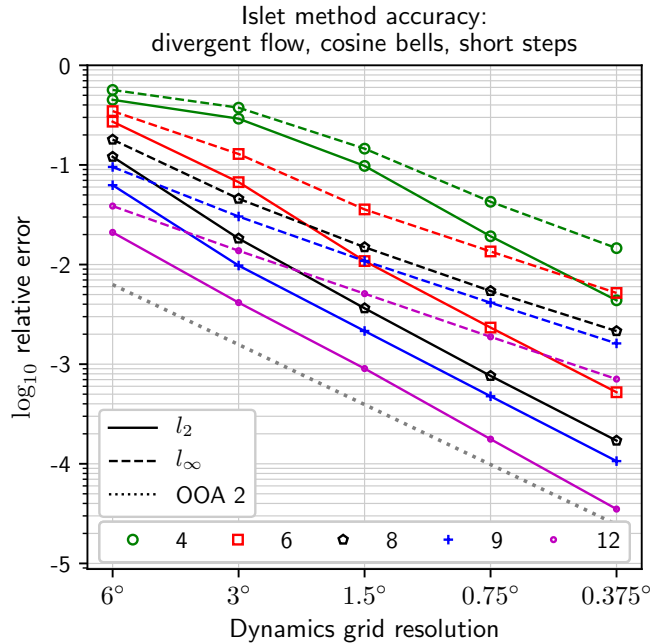
**Figure 9.** Accuracy diagnostic. Compare with Fig. 16 in TR14.

value is 4.80×10^{-5}, again by the UCISOM-CN5.5 method. For both the long and short time steps, at $1.5°$, the Islet method gives at least as small a value for $n_p^t \geq 8$. However, even with the constraint on l_r, comparison is not straightforward, as the UCISOM methods are not strictly shape-preserving and so have $l_o > 0$.

4.2 Slotted cylinders

We observe that in both the filament and mixing diagnostics of TS12, $n_p^t \geq 6$ gives excellent results; $n_p^t = 12$, nearly perfect. Figures 13 and 14 show solution quality using latitude-longitude images and further underscore these observations. The problem is the nondivergent flow with the slotted cylinders IC, at resolutions $1.5°$ and $0.75°$, with the long and short time steps. The text in the individual images in Fig. 14 provides normwise accuracy at the end of one cycle and deviation from the initial extrema. $\phi_{\min} \geq 0$ and $\phi_{\max} \leq 0$ are consistent with no global extrema violation. Compare Fig. 13 with Figs. 7–10 in TR14 and both figures with Fig. 7 in TS12. Although TR14 does not provide error norm values for this problem, those in Fig. 7 of TS12 can be compared with the Islet method's values at $1.5°$ resolution and the long time step (left column of Fig. 14). The Islet method's values of l_2, $l_{\inf}$, $\phi_{\min}$, and $\phi_{\max}$ are at least as good as those in Fig. 7 of TS12 for $n_p^t \geq 6$.

4.3 Source term

Now we move to verification problems that include a source term: $f \neq 0$ in Eq. (3).

27
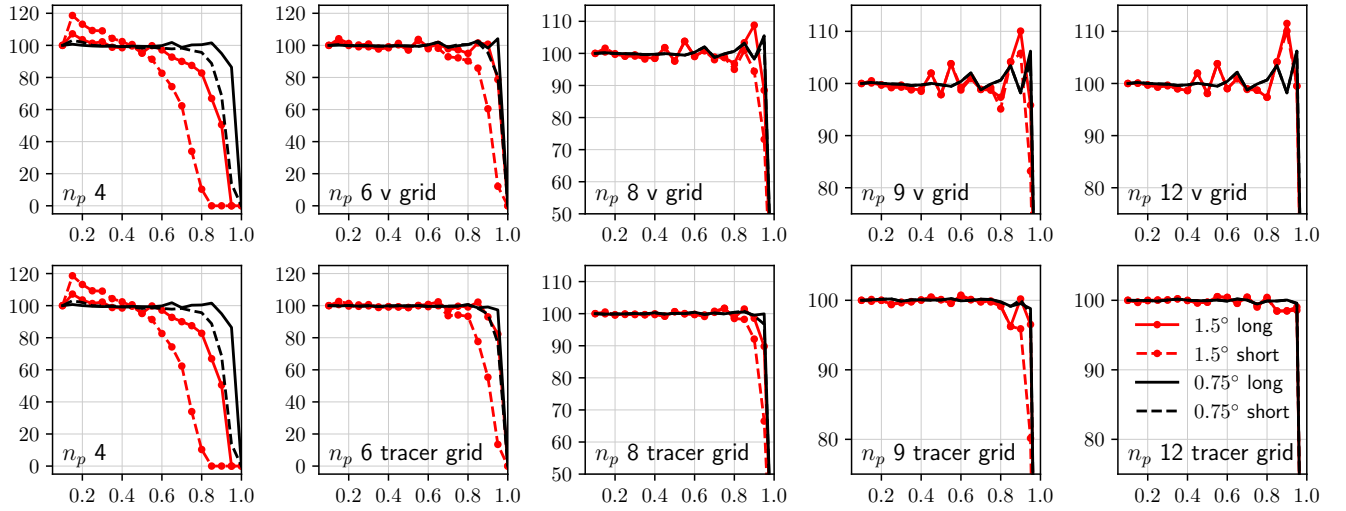
Figure 10. Filament diagnostic, following Sect. 3.3 of TS12. Compare with Fig. 5 in TR14. The top row shows the diagnostic measured on the $n_p^v = 4$ dynamics grid; the bottom row, on the tracer grid. The legend describes the dynamics-grid resolution and the time step length. The prescribed verification problem is the nondivergent flow with cosine bells IC. Property preservation is on. The x-axis is τ, the mixing ratio threshold. The y-axis is the percent area having mixing ratio at least τ relative to that at the initial time.
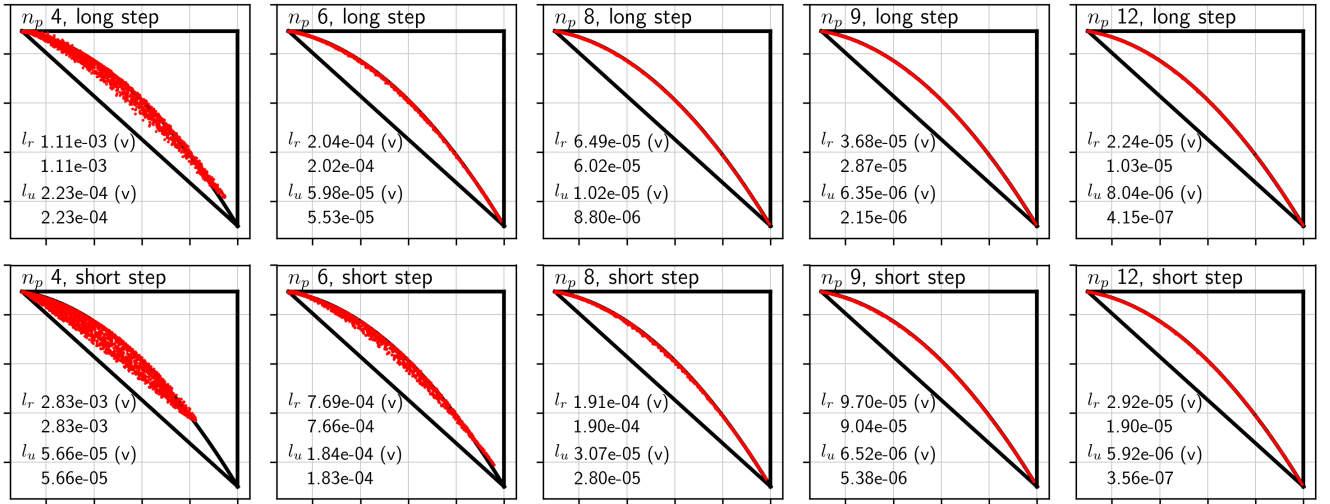
Figure 11. Mixing diagnostic, following Sect. 3.5 of TS12. Compare with Figs. 11–14 in TR14. This figure shows results for dynamics-grid resolution of $1.5°$. l_o is exactly 0 in all cases because shape preservation is on, and so is not shown. See text for further details.
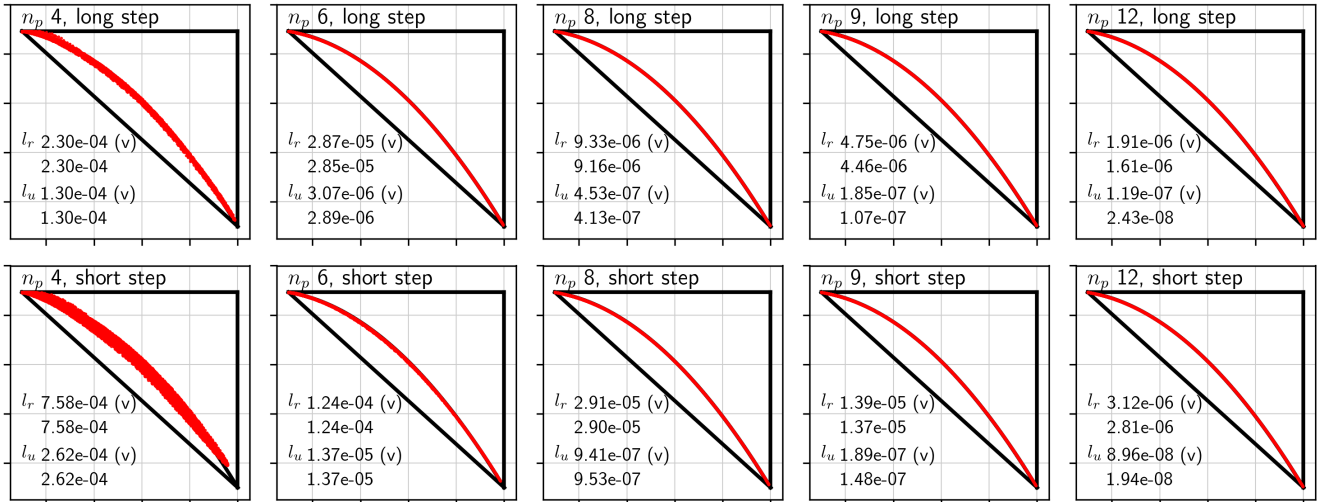
Figure 12. Same as Fig. 11 but with dynamics-grid resolution $0.75°$.

4.3.1 Accuracy

The first test verifies the property-preserving remaps between the physics and tracer grids. The test is constructed as follows. Two tracer mixing ratios, a source $q_1 = s$ and a manufactured tracer $q_2 = m$, are paired. At time $t = 0$, $m(\boldsymbol{x}, t)$ is set to 0, where $\boldsymbol{x}$ is position on the sphere. A tendency Δm is applied to m on the physics grid: $\Delta m(\boldsymbol{x}, t) \equiv -[\cos(2\pi(t + \Delta t)/T) - \cos(2\pi t/T)]s(\boldsymbol{x}, t)/2$, so that the exact solution is $m(\boldsymbol{x}, t) = (1 - \cos(2\pi t/T))s(\boldsymbol{x}, t)/2$ and, in particular, $m(\boldsymbol{x}, T/2) = s(\boldsymbol{x}, T/2)$. To compute the tendency, the state $s(\boldsymbol{x}, t)$ must be remapped to the physics grid. Thus, the results for this test depend on both grid-transfer directions. We measure the error at time $T/2$, on the dynamics grid as usual, as in Fig. 3. Figure 15 shows the results. We run this test with $n_f = n_p^t$ (dash-dotted lines) and $n_f = 2$ (dashed lines). The solid lines show the error in $s(\boldsymbol{x}, T/2)$ as a reference. The dotted line provides the OOA-2 reference. We see that when $n_f = n_p^t$, the errors are nearly the same as those for $s(\boldsymbol{x}, T/2)$; for $n_p^t = n_f = 4$, the curves overlap at the resolution of the plot. As one expects, when $n_f = 2$, the error in $m(\boldsymbol{x}, t)$ is much larger than in $s(\boldsymbol{x}, T/2)$, but the OOA remains 2.

4.3.2 Toy chemistry diagnostic

The toy chemistry verification problem is described in Lauritzen et al. (2015), subsequently TC15. The problem consists of two tracer mixing ratios, $q_i = X_i$, $i = 1, 2$, that interact according to chemical kinetic equations that are nonlinear in one of them: $DX_i(\boldsymbol{y}, t)/Dt = f_i(\boldsymbol{y}, X_1, X_2)$, where $\boldsymbol{y}$ is the spatial coordinate on the sphere. (We use $\boldsymbol{y}$ here to avoid overlap with the species symbol, X, that is used in TC15.) The tracers are composed of a monatomic and a diatomic molecule, respectively, of the same atomic species. The reactions are extremely sensitive to solar insolation. The sun's position is held fixed with respect to the grid. As a result, the largest-scale spatial pattern one sees in the fields is the boundary dividing nonzero (day) and zero
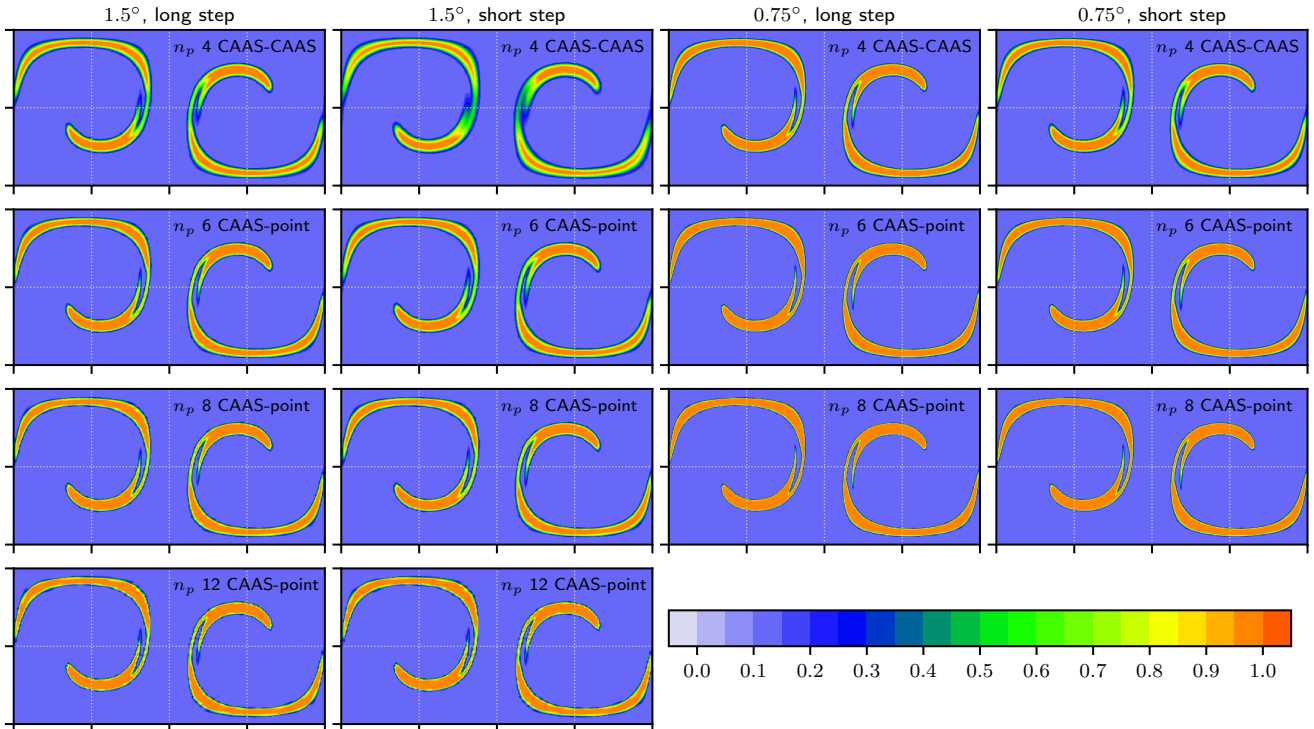
Figure 13. Images of the slotted cylinders IC advected by the nondivergent flow at the simulation's midpoint. Each column corresponds to a spatial resolution and time step length configuration, as stated at the top of each column. Each row corresponds to a particular value of n_p^t, as stated in the text at the top-right of each image. We omit $n_p^t = 12$ results for the $0.75°$ resolution because they are essentially identical at the resolution of the figure to the $n_p^t = 8$ images.

(night) solar insolation, the solar *terminator*; this boundary is particularly visible in the right image of Fig. 17, a figure that we will describe subsequently. The ICs are designed so that the sum over the atomic mixing ratio at each point in space is a constant, $\bar{X}_T$. The source terms have this property, too, since they model chemical reactions. Thus, in the exact solution, $\bar{X}_T$ is maintained at every point in space and time. Let X_T, without a bar, be the corresponding measured quantity. The toy chemistry diagnostics are then $c_2(t)$, the l_2 norm of $X_T - \bar{X}_T$ at time t normalized by $\bar{X}_T$, and $c_\infty(t)$, the same but for the l_∞ norm.

As explained in the context of equation 14 in TC15, any advection operator that is *semi-linear* will produce a perfect diagnostic value of 0 when using exact arithmetic. Linear operators are semi-linear; the CEDR algorithms we use are, as well, as explained in Bradley et al. (2019); and a composition of semi-linear operators is, too. Thus, the Islet method is semi-linear, and deviation from 0 in the diagnostic values is wholly due to the effects of finite precision arithmetic.

We compute the diagnostics, as usual, on the dynamics grid. Following the Islet tracer transport method described in Sect. 3.1, the source terms are computed on the physics grid using states remapped from the tracer grid, and then the computed tendencies are remapped to the tracer grid.
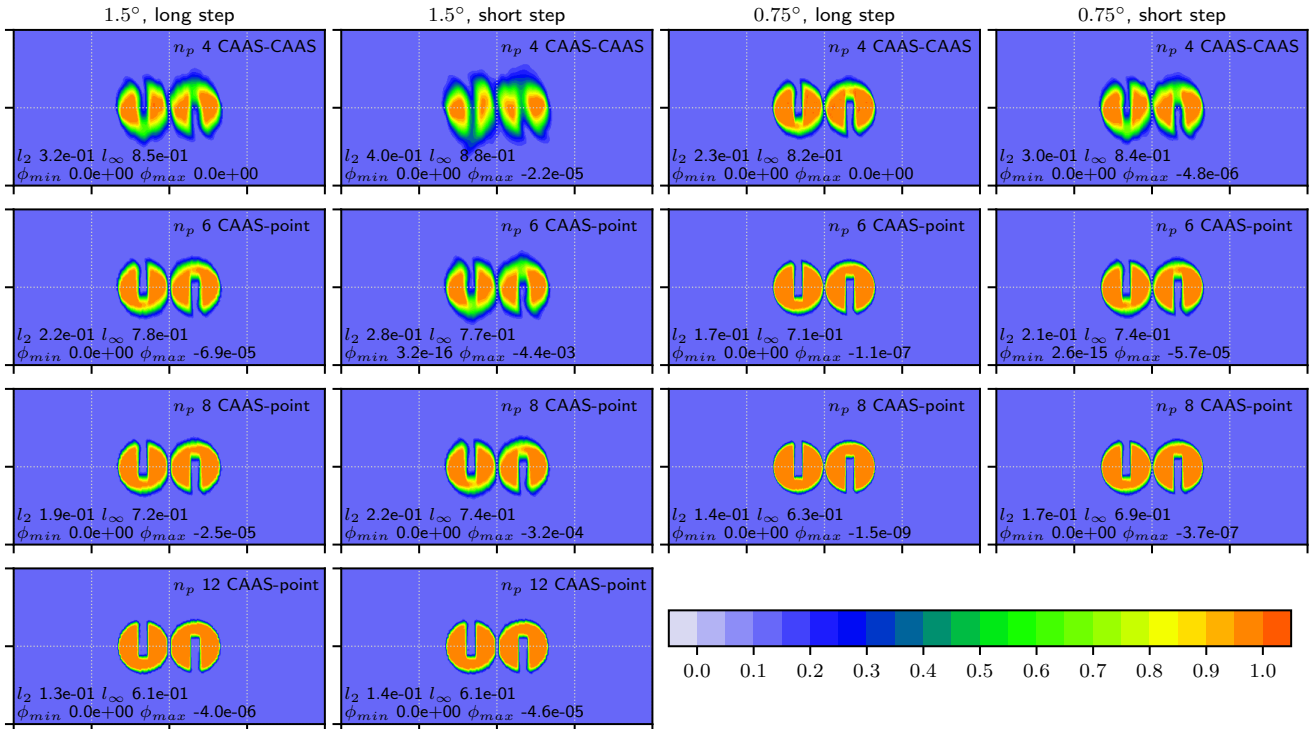
Figure 14. Same as Fig. 13 but for the simulation final point. Error measures are printed at the bottom-left of each image; see text for details.

It is already known that the Eulerian spectral element tracer transport method yields poor values for this diagnostic due to finite-precision effects of the limiter (Lauritzen et al., 2017). The Islet method with property preservation using CAAS-CAAS does, as well. Again, in exact arithmetic, each of these methods would produce perfect values. The poor diagnostic values are due to quickly accumulating machine-precision round-off errors that break semi-linearity in finite precision. The interaction of the chemistry source term with exact bounds in element-local limiter applications is responsible for this fast accumulation. In contrast, the Islet method using CAAS-point and relaxed-bound, element-local CAAS applications produces good diagnostic values. The relaxed bounds in the element-local part make unnecessary many of the mixing ratio adjustments that lead to loss of semi-linearity in finite precision. Recall that CAAS-point, applied at the end of the step, imposes the exact bounds, so at the end of a time step, shape preservation still holds to machine precision. Clips to bounds must still occur, but adjustments to other grid points to compensate are smaller because the adjustments are spread over many more grid points.

Figure 16 shows the diagnostic values for the case of nondivergent flow, $1°$ dynamics-grid resolution, and a 30-minute time step, where these configuration details are prescribed in TC15. For the $n_p^t = 4$ case, we use CAAS-point rather than CAAS-CAAS as previously, since we already know that CAAS-CAAS will produce poor values. The diagnostic is usually plotted over the course of one cycle (12 days) of the flow, but it is useful to view it over multiple cycles. Figure 16 shows ten cycles on the x-axis, for a total of 120 days. The y-axis is the diagnostic value. Solid lines plot $c_2(t)$; dashed, $c_\infty(t)$. Markers are placed
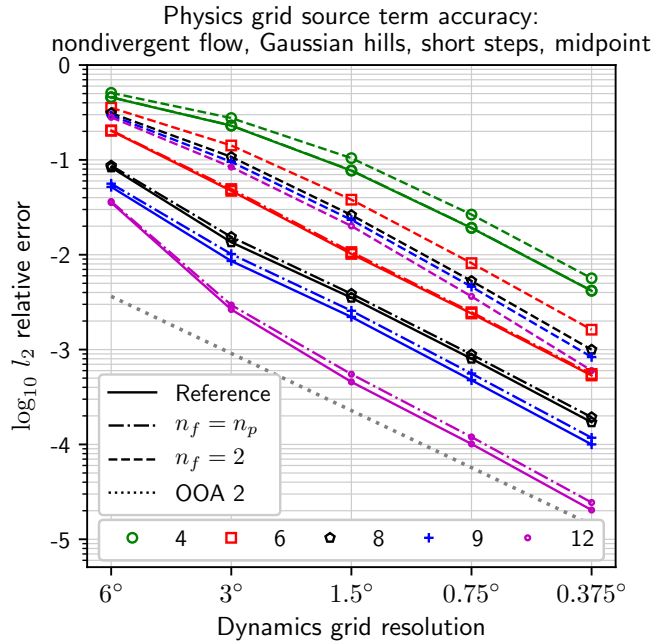
Figure 15. Verification of the remap of tendencies from physics to tracer grids and state from tracer to dynamics grids. See Sect. 4.3 for a description of the problem.

on the curves at the start of each cycle to differentiate the curves. In each case, $n_f = n_p^t$. We choose this value of n_f because the toy chemistry source term has an extremely large gradient at the terminator, and thus it makes sense to compute the physics tendencies at high spatial resolution. The $n_p^t = 4$ case with CAAS-point is greatly improved relative to the Eulerian spectral element results shown in Fig. 7 of TC15, even after ten cycles instead of the one cycle shown in that figure. In Fig. 7 of TC15, $c_2 \approx 10^{-2}$ and $c_\infty > 10^{-1}$ at the end of one cycle, compared with approximately 10^{-7} and 10^{-5}, respectively, for $n_p^t = 4$ at the end of ten cycles. For $n_p^t > 4$, the growth in error is very small, with $c_2 < 10^{-10}$ through ten cycles.

To illustrate what these diagnostics measure, Figure 17 shows latitude-longitude images of the monatomic tracer at the end of the first cycle for $n_p^t = n_f = 4$ with CAAS-CAAS (left) and $n_p^t = n_f = 8$ with CAAS-point (right). Note that the images in TS12 and TR14 are plotted with longitude ranging from 0 to 2π; those in TC15, from $-\pi$ to π. In our latitude-longitude figures so far, we have chosen the convention used in TS12 and TR14, and we continue to use it in these toy-chemistry images. Thus, these images are circularly shifted horizontally by half the image width relative to those in TC15. The globally extremal tracer values are printed in the upper-right quadrant of each image. The correct maximum is 4×10^{-6} and the correct minimum is at least 0. The right image is free of noise and satisfies these bounds. The left image shows substantial noise, as we expect when using exact bounds in the local property preservation problems, and consistent with previous observations about spectral element transport. Other than noise and some filaments that grow from the noise, the two images are qualitatively similar. Figure 18 shows images in the same format, but the quantity is now $(X_T - \bar{X}_T)/\bar{X}_T$ at the end of the first cycle. The correct

32
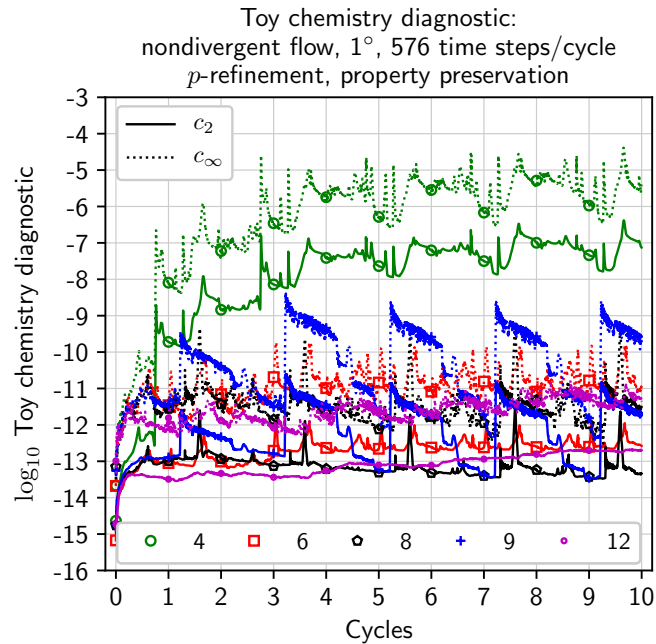
Figure 16. Toy chemistry diagnostic values as a function of time for ten cycles of the nondivergent flow. Time is on the x-axis and measured in cycles. Diagnostic values c_2 (solid lines) and c_∞ (dashed lines) are on the y-axis. Markers as listed in the bottom legend are placed at the start of each cycle to differentiate the curves.
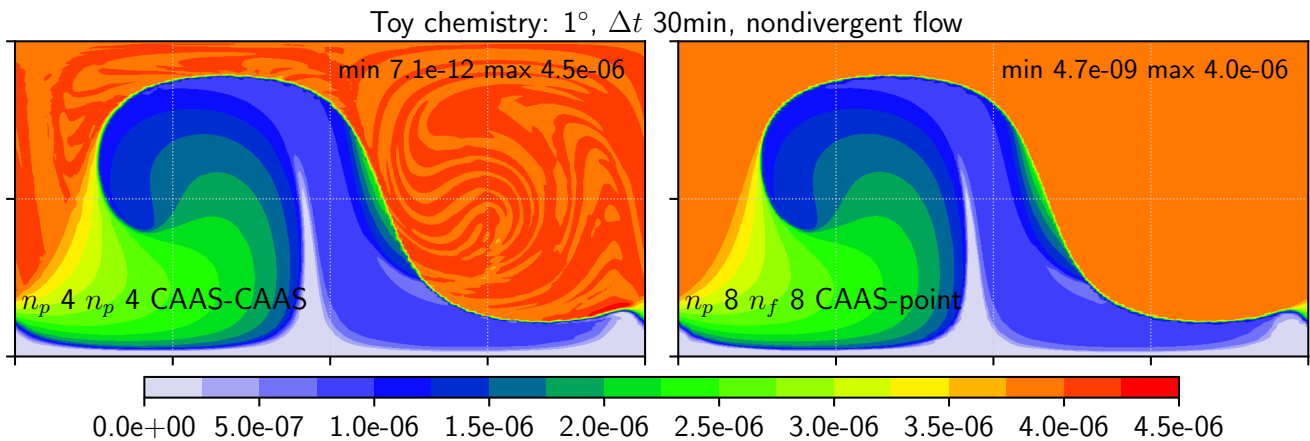
Figure 17. Images of the monatomic tracer at the end of the first cycle. Text at the lower left of each image states the configuration. Text at the upper right reports global extremal values.
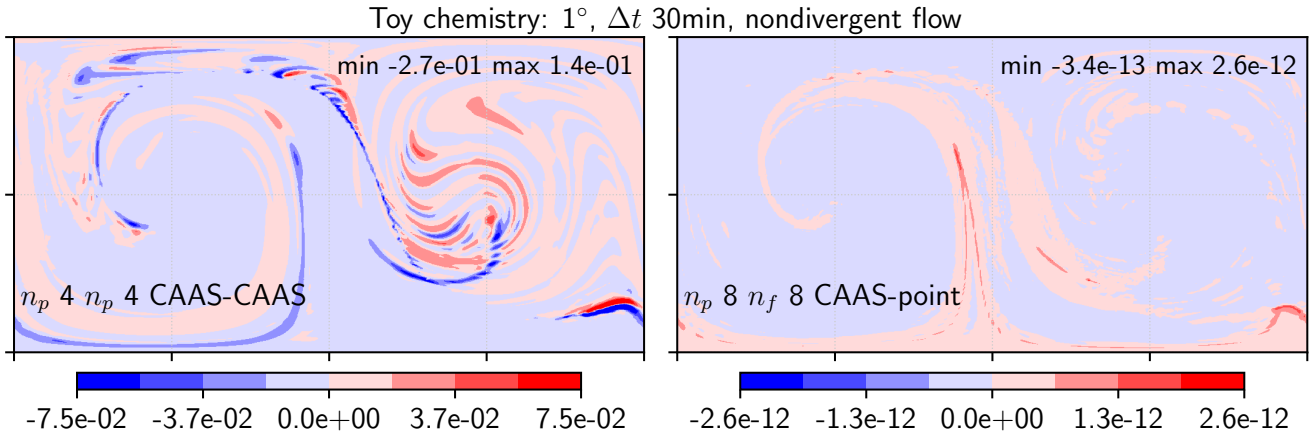
Figure 18. Same as Fig. 17, but now the images are of $(X_T - \bar{X}_T)/\bar{X}_T$.

value is 0 everywhere. In the right image, the pointwise relative error is a little better than 10^{-11}, consistent with the l_∞-norm diagnostic value for $n_p^t = 8$ at the end of the first cycle in Fig. 16.

5 Conclusions

735 We described the Islet method, a property preserving tracer transport method to support a three-grid atmosphere model, one with a shared element grid but separate subelement grids for physics parameterizations, dynamics, and tracer transport. This configuration permits the modeler to create a dynamics grid with a tolerable CFL-limited time step independent of the other two subcomponents, while physics parameterizations and tracer transport can run at resolutions potentially substantially higher. The shared element grid minimizes communication during remaps between grids, and almost all operations are local to each
740 element, making the Islet method extremely efficient.

Section 4 presented a number of verification problems with diagnostics assessing accuracy, order of accuracy, numerical mixing, filament preservation, and nonlinear correlation preservation. The corresponding figures and diagnostic values can be compared directly with those of several other methods. The Islet method performs well in a number of detailed comparisons.

Possible future work includes the following. First, because all operations except the DSS are either local to the element or
745 act on element-level scalars, both the tracer and the physics grids permit various types and subsets of spatially and temporally adaptive and tracer-dependent refinement and derefinement, possibly in combination with already existing E3SM regionally refined models (RRM). Future work could develop procedures to permit spatially and temporally varying n_p^t and n_f values and incorporate these into the Islet method.

Second, this article focuses on the horizontal dimensions or, more generally, 2D Lagrangian levels in a three-dimensional
750 discretization. However, it is necessary to increase simultaneously both the horizontal and vertical resolutions of a plume to

34

capture and maintain its structure (Eastham and Jacob, 2017). Future work using methods similar to those in Sect. 3 should address the vertical dimension.

Third, it is possible to recover local mass conservation in an ISL method by modifying the linear advection operator's coefficients so that the sum of the mass over all the target points associated with a source element is consistent with the source element's total mass (Kaas, 2008). This coefficient modification step might be compatible with the Islet method and should be explored.

Finally, we predict that applications related to, in particular, aerosols will benefit from the Islet method. In future work, we intend to integrate the Islet method into the E3SM Atmosphere Model to investigate its impact on science applications.

Appendix A: Islet bases

The GLL basis functions yield an unstable ISL method for $n_p \geq 4$. We developed a set of bases on GLL nodes that yield ISL methods that meet a necessary condition for stability on the test problem of uniform periodic flow. We refer to these as the *Islet bases* and use these bases in this article. Bradley et al. (2021, Sect. 2 and 3) and a forthcoming article based on that material detail the derivation of the Islet bases, while this appendix summarizes the results of that derivation for completeness. This article can be understood equally well by assuming the standard GLL bases are used; only the numerical results depend on the details of the basis functions.

A 1D reference element has domain $[-1, 1]$. Higher-dimensional basis functions are tensor products of 1D basis functions. The bases are nodal, meaning a basis function has the value 1 at one node and 0 at all others. The nodes are GLL. Let $x_G^{n_p}(i)$, $i \in \{0, \ldots, n_p - 1\}$, be the reference coordinates of the GLL nodes. Let *region* $r \in \{0, \ldots, n_p - 2\}$ be the segment $[x_G^{n_p}(r), x_G^{n_p}(r+1)]$. A 1D basis function is a continuous piecewise polynomial over $[-1, 1]$; in each region, it is a polynomial. Each region r has an associated ordered list of *support nodes*, denoted $\mathcal{I}_r^{n_p}$. Each list $\mathcal{I}_r^{n_p}$ is a subset of $\{0, \ldots, n_p - 1\}$. Let $|\mathcal{I}_r^{n_p}|$ be the number of elements in $\mathcal{I}_r^{n_p}$, let $n_p^{\text{submin}} \equiv \min_r |\mathcal{I}_r^{n_p}|$, and let $\mathcal{I}_r^{n_p}(j)$ be the jth element.

In region r and thus for $x \in [x_G^{n_p}(r), x_G^{n_p}(r+1)]$, basis function k is 0 if $k \notin \mathcal{I}_r^{n_p}$ and otherwise has the value given by the Lagrange polynomial

$$\phi_k^{n_p}(x) \equiv \prod_{\substack{j \in \{0, \ldots, |\mathcal{I}_r^{n_p}| - 1\}, \\ \mathcal{I}_r^{n_p}(j) \neq k}} \frac{x - x_G^{n_p}(\mathcal{I}_r^{n_p}(j))}{x_G^{n_p}(k) - x_G^{n_p}(\mathcal{I}_r^{n_p}(j))}.$$

We construct a basis for each $n_p \in \{4, \ldots, 13\}$. Each basis is described by a set of support node lists, one $\mathcal{I}_r^{n_p}$ per region. For $n_p = 4$, there are additional details that we describe in Appendix A1. There are two types of bases, each corresponding to a method of describing the support node lists.

The first type of basis is the *offset nodal basis*. Each list $\mathcal{I}_r^{n_p}$ is given by an *offset*, which is the first index in the list, and its size, $|\mathcal{I}_r^{n_p}|$. For example, $n_p = 7$, offset 2, and size 4 correspond to support nodes $\mathcal{I}_r^7 = \{2, 3, 4, 5\}$. In addition, the basis is symmetric, meaning basis function $\phi_i^{n_p}(x) = \phi_{n_p-1-i}^{n_p}(-x)$. Thus, first, support nodes are specified for regions 0 through $\lfloor n_p/2 \rfloor - 1$, and the support nodes for the remaining regions are determined by symmetry. Second, if n_p is even, then the middle region, $r = n_p/2 - 1$, has support nodes $\mathcal{I}_r^{n_p}$ that are symmetric around reference coordinate 0.

The second type of basis is the *nodal basis*. An offset nodal basis is a nodal basis whose support nodes all permit the more compact description of offset and size. A general nodal basis has at least one support nodal list that cannot be described by just an offset and a size. Instead, the list is stated explicitly. Again, all nodal bases are symmetric.

The order of accuracy of the ISL method using an Islet GLL basis is $n_p^{\text{submin}} - 1$ if there is no property preservation step and the initial condition is $C^{n_p^{\text{submin}}-2}$.

If $n_p^{\text{submin}} = n_p$, then the basis is the natural GLL basis. Thus, any statement about an Islet GLL basis also holds for the natural GLL basis.

A search procedure searches for nodal bases that provide ISL methods that are stable on the test problem of uniform flow on a uniform grid. An accuracy heuristic is used to find the most accurate basis satisfying this stability condition. Table A1 lists nodal bases, most of them offset nodal bases, for $n_p \in \{5, \ldots, 13\}$. In addition, a special basis is constructed for $n_p = 4$.

A1 $n_p = 4$ Islet basis

For $n_p = 4$, we boost the accuracy of the basis by combining an offset nodal basis and the GLL basis. The middle region has support nodes $\mathcal{I}_1 = \{0, 1, 2, 3\}$, i.e., all four nodes. Region 0 convexly combines basis function k given by $\mathcal{I}_0^4 = \{0, 1, 2\}$ with GLL basis function k over the region $x \in [x_G^4(0), x_G^4(1)]$, with convex combination function $\alpha(x)$. $\alpha(x)$ is a quadratic polynomial having values 1, 0.306, and 0 at, respectively the left side, middle, and right side of region 0. The combination is defined so that at the left side, where $\alpha(-1) = 1$, only the GLL basis function k is used; at the right side, where $\alpha(x_G^4(1)) = 0$, only the offset nodal basis function k is used. Region 2 is constructed by symmetry. $\alpha(x)$ is determined to maximize the accuracy of the resulting ISL method subject to the constraint of stability on the test problem.

A2   Conservation when interpolating

The element-local conservation expressed in Eq. (15) holds for the following reasons. First, for any Islet basis for which $(n_p^{\text{t}})^{\text{submin}} \geq n_p^{\text{v}}$, the continuum field f is the same on each grid, since the tracer grid can exactly represent polynomials of degree $n_p^{\text{v}} - 1$. Second, for n_p^{v} even, such as the standard $n_p^{\text{v}} = 4$, if $(n_p^{\text{t}})^{\text{submin}} = n_p^{\text{v}} - 1$, the continuum field f is in general different on each grid, but the integral of each over an element has the same value. This is because for polynomial degree d odd and $f(x) \equiv \sum_{i=0}^d a_i x^i$, $\int_{-1}^1 f(x)\, dx = \int_{-1}^1 (f(-x) + f(x))/2\, dx$, and $g(x) \equiv (f(-x) + f(x))/2$ has degree at least one less than $f(x)$. $g(x)$ is exactly represented on the tracer grid, and thus the integral of f, which is the same as the integral of g,

n_p	OOA	$|\mathcal{I}_r|$	Supports
4	2	see text	see text
5	2	{3, 4}	offsets {0, 0}
6	4	{5, 5, 6}	nodal subsets {{0, 1, 2, 3, 4}, {0, 1, 2, 3, 5}, {0, 1, 2, 3, 4, 5}}
7	4	{5, 5, 6}	offsets {0, 0, 0}
8	5	{6, 6, 7, 6}	offsets {0, 0, 0, 1}
9	6	{7, 8, 8, 7}	nodal subsets {{0, 1, 2, 3, 4, 5, 8}, {0, 1, 2, 3, 4, 5, 7, 8}, {0, 1, 2, 3, 4, 5, 6, 8}, {1, 2, 3, 4, 5, 6, 7}}
10	6	{7, 7, 7, 8, 8}	offsets {0, 0, 0, 0, 1}
11	7	{8, 9, 8, 9, 8}	offsets {0, 0, 0, 0, 1}
12	8	{9, 9, 10, 10, 9, 10}	offsets {0, 0, 0, 0, 1, 1}
13	9	{10, 10, 10, 10, 11, 10}	offsets {0, 0, 0, 0, 0, 1}

Table A1. Islet GLL nodal subset bases. Each row provides a formula for the row's n_p value. Columns are n_p, order of accuracy (OOA), the support sizes $|\mathcal{I}_r|$ for each region ordered left to middle, and the supports. For offset nodal subset bases, supports are given by offsets. For general nodal subset bases, supports are given by nodal subsets, again ordered from left region to middle. The case $n_p = 4$ is described in Appendix A1. In all cases, the support points are GLL nodes.

is the same on the tracer grid as on the dynamics grid. These two reasons assure that $\mathcal{I}^{v \to t}$ is conservative for the Islet bases in Table A1 when $n_p^v = 4$.

A3 Unstable and stable integration

815 Figure A1, which has the format of the figures described in Sect. 4, compares accuracy and stability between the natural and Islet bases. The divergent flow, Gaussian hills IC, property preservation, TTPR, and long time steps are used. To test the stability of the ISL linear advection operator using the Islet bases, we run the verification problem for both 1 cycle and 100 cycles. A curve's line pattern corresponds to basis type and number of cycles: solid, Islet basis for 1 cycle; dashed, Islet basis for 100 cycles (or $12 \times 100 = 1200$ days); dash-dotted, natural basis for 1 cycle. For each n_p^t, the l_2 norm of the solution using 820 the natural basis diverges with increasing resolution within the first cycle, demonstrating that the basis leads to an unstable ISL

37

linear advection operator. In the case of $n_p^t = 4$, we see the start of the curve's divergence, but further element-grid refinement is needed to see the curve fully diverge. In contrast, the curves for the Islet method with the Islet bases converge at order of accuracy 2.
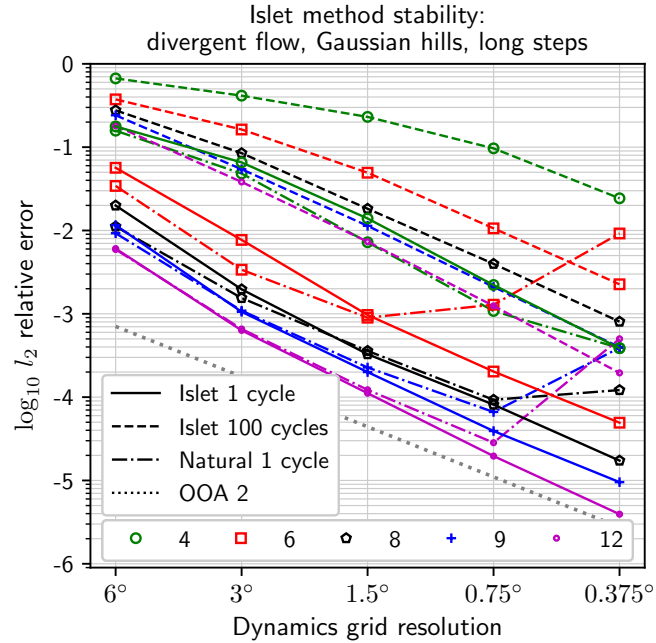
Figure A1. Stability of the Islet method with the Islet GLL bases, compared with the instability of the method with the natural GLL bases. The x-axis is the average dynamics-grid grid point spacing at the equator in degrees for the quasiuniform cubed-sphere grid. The y-axis is $\log_{10} l_2$ relative error. A curve's line pattern corresponds to the basis type and number of cycles, as listed in the top legend. A curve's marker corresponds to n_p^t, as listed in the bottom legend.

Appendix B: Discretization of the continuity equation

825 Our primary interest is the advection equation and not the continuity equation. However, to run tests with divergent flow and property preservation enabled, we need to solve the continuity equation for the air density to provide the ρ argument to CAAS. This appendix describes a discretization of the continuity equation that differs in only one factor from the discretization of the advection equation.

We start by following, e.g., Giraldo (1997, Sect. 3.3) and Bosler et al. (2019, Sect. 3). The Reynolds transport theorem for a
830 Lagrangian fluid parcel $\Omega(t)$ containing a fluid having density $\rho(\boldsymbol{x}, t)$ is

$$\frac{d}{dt} \int_{\Omega(t)} \rho \, d\boldsymbol{x} = \int_{\Omega(t)} \rho_t + \nabla \cdot (\rho \boldsymbol{u}) \, d\boldsymbol{x}. \tag{B1}$$

To balance clarity and concision, we include or omit the position and time arguments depending on the context. Suppose ρ satisfies the continuity equation,

$$\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{B2}$$

835   In addition, consider a function $\phi(\boldsymbol{x}, t)$ that satisfies the advection equation:

$$\frac{D\phi}{Dt} = 0. \tag{B3}$$

Then by Eqs. (B1), (B2), and (B3),

$$\frac{d}{dt} \int_{\Omega(t)} \rho\phi \, \mathrm{d}\boldsymbol{x} = \int_{\Omega(t)} (\rho\phi)_t + \nabla \cdot (\rho\phi\boldsymbol{u}) \, \mathrm{d}\boldsymbol{x}$$

$$= \int_{\Omega(t)} \phi[\rho_t + \nabla \cdot (\rho\boldsymbol{u})] + \rho\frac{D\phi}{Dt} \, \mathrm{d}\boldsymbol{x}$$

840 $$= 0,$$

which in turn implies

$$\int_{\Omega(t_1)} \rho\phi \, \mathrm{d}\boldsymbol{x} = \int_{\Omega(t_0)} \rho\phi \, \mathrm{d}\boldsymbol{x} \tag{B4}$$

for all times t_0, t_1.

     Now we discretize Eq. (B4) in space. First, we write an integral over a scalar f in terms of the element reference coordinate

845   system. Let $\Omega(t)$ be a quadrilateral element, with $\Omega(t_1)$ the arrival Eulerian element. Then

$$\int_{\Omega(t)} f(\boldsymbol{x}, t) \, \mathrm{d}\boldsymbol{x} = \int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{x}(\boldsymbol{r}, t), t) \, J(\boldsymbol{r}, t) \, \mathrm{d}\boldsymbol{r}. \tag{B5}$$

Here, $\boldsymbol{x}(\boldsymbol{r}, t) \equiv \boldsymbol{X}(t; \boldsymbol{x}(\boldsymbol{r}, t_1), t_1)$ is the time-dependent map from the reference element to the manifold and satisfies Eq. (6). $J(\boldsymbol{r}, t)$ is the absolute value of the Jacobian determinant of this map, $|\det(\partial \boldsymbol{x}(\boldsymbol{r}, t)/\partial \boldsymbol{r})|$.

     Second, let the element have a tensor-product grid of GLL nodes, with n_p the number of 1D nodes. We apply tensor-product

850   GLL quadrature with this value of n_p to the right hand side of Eq. (B5). Then

$$\int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{x}(\boldsymbol{r}, t), t) \, J(\boldsymbol{r}, t) \, \mathrm{d}\boldsymbol{r}$$

$$\approx \sum_{i=0}^{n_p^2-1} f(\boldsymbol{x}(\boldsymbol{r}_i, t), t) \, J(\boldsymbol{r}_i, t) \, w_i,$$

where i iterates over nodes in the element, $\boldsymbol{r}_i$ is the ith GLL node, and w_i is the ith GLL weight. Let $\boldsymbol{x}_i(t) \equiv \boldsymbol{x}(\boldsymbol{r}_i, t)$.

Third, we combine the quadrature approximation with a specific form of f, $f = \rho\phi$, where ϕ satisfies Eq. (B3). Let $\phi(\boldsymbol{x}, t_1)$

855 be an element, GLL nodal, tensor-product n_p-basis function on an Eulerian arrival element. Because ϕ is a nodal basis function with the same nodes as the element, $\phi = 1$ at one node, say node k, and is 0 at all other nodes $i \neq k$. ϕ need not be a natural GLL basis function; for example, it can be an Islet GLL basis function. Because $D\phi/Dt = 0$, $\phi(\boldsymbol{x}_i(t), t) = \phi(\boldsymbol{x}_i(t_1), t_1)$ for any time t. Thus, the quadrature sum can be simplified:

$$\sum_{i=0}^{n_p^2-1} \rho(\boldsymbol{x}_i(t), t) \, \phi(\boldsymbol{x}_i(t), t) \, J(\boldsymbol{r}_i, t) \, w_i$$

860
$$= \rho(\boldsymbol{x}_k(t), t) \, J(\boldsymbol{r}_k, t) \, w_k.$$

Fourth, we apply quadrature to each side of Eq. (B4) to obtain the discretization

$$\rho(\boldsymbol{x}_k(t_1), t_1) \, J(\boldsymbol{r}_k, t_1) \, w_k = \rho(\boldsymbol{x}_k(t_0), t_0) \, J(\boldsymbol{r}_k, t_0) \, w_k,$$

or

$$\rho(\boldsymbol{x}_k(t_1), t_1) = \frac{J(\boldsymbol{r}_k, t_0)}{J(\boldsymbol{r}_k, t_1)} \, \rho(\boldsymbol{x}_k(t_0), t_0). \tag{B6}$$

865 Eq. (B6) differs from Eq. (5) in the appearance of the *density factor*, the quotient of the two Jacobian determinants. In this equation, both ρ values and the denominator of the density factor are grid point values. Note that this simple form depends on three different node sets being the same: those for the element, the GLL quadrature, and the basis function ϕ.

Fifth, to complete the discretization, the discretization steps in Sect. 2.1.4 are applied to Eq. (B6) as they were to Eq. (5), but now with the additional factor present.

870 Since Eq. (B6) is not the focus of this article, we omit detailed analysis of the discretization. However, we note some basic facts. Importantly, the Jacobian determinant of the Lagrangian element appears in the numerator rather than the denominator of the density factor; thus, the denominator is always well behaved if the Eulerian grid has reasonable quality. Second, for certain simple flows such as uniform translational flow on the plane, J is constant in time and Eq. (B6) reduces to the discretization of the advection equation, Eq. (5). For general nondivergent flow, J is time dependent because of the spatial discretization

875 error. Third, only the density factor can make the order of accuracy of the discretization Eq. (B6) differ from that of Eq. (5). Details concerning the manifold on which flow occurs and the map from the reference element to the manifold affect the order of accuracy. In the verification problems in this article, the density ρ is approximated using the Islet $n_p = 4$ basis, the manifold is the sphere, and the map from the reference element to the sphere is the isoparametric map of the finite element method for the n_p-basis. The discretization of the continuity equation for ρ then has order of accuracy 2.

References

Bosler, P. A., Bradley, A. M., and Taylor, M. A.: Conservative multimoment transport along characteristics for discontinuous Galerkin methods, SIAM J. on Sci. Comput., 41, B870–B902, 2019.

Bradley, A. M., Bosler, P. A., Guba, O., Taylor, M. A., and Barnett, G. A.: Communication-efficient property preservation in tracer transport, SIAM J. on Sci. Comput., 41, C161–C193, 2019.

Bradley, A. M., Bosler, P. A., and Guba, O.: Islet: Interpolation semi-Lagrangian element-based transport, Geosci. Model Dev. Discuss., p. [preprint], https://doi.org/10.5194/gmd-2021-296, 2021.

Chen, Y., Simon, K., and Behrens, J.: Extending legacy climate models by adaptive mesh refinement for single-component tracer transport: a case study with ECHAM6-HAMMOZ (ECHAM6. 3-HAM2. 3-MOZ1. 0), Geosci. Model Dev., 14, 2289–2316, 2021.

Dennis, J., Fournier, A., Spotz, W. F., St-Cyr, A., Taylor, M. A., Thomas, S. J., and Tufo, H.: High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core, Int. J. High Perform C., 19, 225–235, 2005.

Dennis, J., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P., Mirin, A. A., St-Cyr, A., Taylor, M. A., and Worley, P. H.: CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model, Int. J. High Perform C., 26, 74–89, 2012.

Dormand, J. R. and Prince, P. J.: A family of embedded Runge-Kutta formulae, J. of Comput. and Appl. Math., 6, 19–26, 1980.

E3SM Project: Energy Exascale Earth System Model (E3SM), [Computer Software] https://dx.doi.org/10.11578/E3SM/dc.20180418.36, 2018.

Eastham, S. D. and Jacob, D. J.: Limits on the ability of global Eulerian models to resolve intercontinental transport of chemical plumes, Atmos. Chem. Phys., 17, 2543–2553, 2017.

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

Field, C., Cheung, W., Dilling, L., Frumhoff, P., Greely, H., Hordequin, M., Hurrell, J., Light, A., Lin, A., MacMartin, D., et al.: Reflecting Sunlight: Recommendations for Solar Geoengineering Research and Research Governance, The National Academies Press, Washington, DC, https://doi.org/10.17226/25762, 2021.

Fischer, P. F. and Patera, A. T.: Parallel spectral element methods for the incompressible Navier-Stokes equations, in: Solution of Superlarge Problems in Computational Mechanics, pp. 49–65, Springer, 1989.

Giraldo, F. X.: Lagrange–Galerkin methods on spherical geodesic grids, Journal of Computational Physics, 136, 197–213, 1997.

Giraldo, F. X., Perot, J. B., and Fischer, P. F.: A spectral element semi-Lagrangian (SESL) method for the spherical shallow water equations, J. Comput. Phys., 190, 623–650, 2003.

Golaz, J.-C., Caldwell, P. M., Van Roekel, L. P., Petersen, M. R., Tang, Q., Wolfe, J. D., Abeshu, G., Anantharaj, V., Asay-Davis, X. S., Bader, D. C., et al.: The DOE E3SM coupled model version 1: Overview and evaluation at standard resolution, J. Adv. Model Earth Sy., 11, 2089–2129, 2019.

Golaz, J.-C., Van Roekel, L. P., Zheng, X., Roberts, A., Wolfe, J. D., Lin, W., Bradley, A., Tang, Q., Maltrud, M. E., Forsyth, R. M., et al.: The DOE E3SM Model Version 2: Overview of the physical model, Earth and Space Science Open Archive [preprint], https://doi.org/10.1002/essoar.10511174.1, 2022.

Guba, O., Taylor, M. A., Ullrich, P. A., Overfelt, J. R., and Levy, M. N.: The spectral element method (SEM) on variable-resolution grids: Evaluating grid sensitivity and resolution-aware numerical viscosity, Geosci. Model Dev., 7, 2803–2816, 2014.

Hannah, W. M., Bradley, A. M., Guba, O., Q., T., Golaz, J.-C., and Wolfe, J.: Separating Physics and Dynamics grids for Improved Computational Efficiency in Spectral Element Earth System Models, J. Adv. Model Earth Sy., 13, e2020MS002 419, 2021.

Herrington, A. R., Lauritzen, P. H., Reed, K. A., Goldhaber, S., and Eaton, B. E.: Exploring a lower resolution physics grid in CAM-SE-CSLAM, J. Adv. Model Earth Sy., p. 2019MS001684, https://doi.org/10.1029/2019MS001684, 2019.

Kaas, E.: A simple and efficient locally mass conserving semi-Lagrangian transport scheme, Tellus A, 60, 305–320, 2008.

Lauritzen, P. H.: Atmospheric Transport Schemes: Desirable Properties and a Semi-Lagrangian View on Finite-Volume Discretizations, in: Numerical Techniques for Global Atmospheric Models, edited by Lauritzen, P. H., Jablonowski, C., Taylor, M. A., and Nair, R. D., vol. 80 of *Lecture Notes in Computational Science and Engineering*, pp. 185–250, Springer: Berlin, 2011.

Lauritzen, P. H. and Thuburn, J.: Evaluating advection/transport schemes using interrelated tracers, scatter plots and numerical mixing diagnostics, Q. J. Roy. Meteor. Soc., 138, 906–918, 2012.

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

Lauritzen, P. H., Skamarock, W. C., Prather, M. J., and Taylor, M. A.: A standard test case suite for two-dimensional linear transport on the sphere, Geosci. Model Dev., 5, 2012.

Lauritzen, P. H., Ullrich, P. A., Jablonowski, C., Bosler, P. A., Calhoun, D., Conley, A. J., Enomoto, T., Dong, L., Dubey, S., Guba, O., et al.: A standard test case suite for two-dimensional linear transport on the sphere: results from a collection of state-of-the-art schemes, Geosci. Model Dev., 7, 105–145, 2014.

Lauritzen, P. H., Conley, A. J., Lamarque, J.-F., Vitt, F., and Taylor, M. A.: The terminator "toy" chemistry test: a simple tool to assess errors in transport schemes, Geosci. Model Dev., 8, 1299–1313, 2015.

Lauritzen, P. H., Taylor, M. A., Overfelt, J., Ullrich, P. A., Nair, R. D., Goldhaber, S., and Kelly, R.: CAM-SE–CSLAM: Consistent Coupling of a Conservative Semi-Lagrangian Finite-Volume Method with Spectral Element Dynamics, Mon. Weather Rev., 145, 833–855, 2017.

Lee, D., Lowrie, R., Petersen, M., Ringler, T., and Hecht, M.: A high order characteristic discontinuous Galerkin scheme for advection on unstructured meshes, J. Comput. Phys., 324, 289–302, 2016.

Lee, H.-H., Bogenschutz, P., and Yamaguchi, T.: The Implementation of Framework for Improvement by Vertical Enhancement (FIVE) into Energy Exascale Earth System Model (E3SM), J. Adv. Model Earth Sy., p. e2020MS002240, 2020.

Natarajan, H. and Jacobs, G. B.: An explicit semi-Lagrangian, spectral method for solution of Lagrangian transport equations in Eulerian-Lagrangian formulations, Comput. Fluids, 207, 104 526, 2020.

Semakin, A. N. and Rastigejev, Y.: Optimized wavelet-based adaptive mesh refinement algorithm for numerical modeling of three-dimensional global-scale atmospheric chemical transport, Q. J. Roy. Meteor. Soc., 146, 1564–1574, 2020.

Shampine, L. F. and Reichelt, M. W.: The Matlab ODE suite, SIAM J. on Sci. Comput., 18, 1–22, 1997.

Stensrud, D. J.: Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models, Cambridge University Press, 2009.

Tang, Q., Klein, S. A., Xie, S., Lin, W., Golaz, J.-C., Roesler, E. L., Taylor, M. A., Rasch, P. J., Bader, D. C., Berg, L. K., et al.: Regionally refined test bed in E3SM atmosphere model version 1 (EAMv1) and applications for high-resolution modeling, Geosci. Model Dev., 12, 2679–2706, 2019.

Ullrich, P. A., Lauritzen, P. H., and Jablonowski, C.: Some considerations for high-order 'incremental remap'-based transport schemes: edges, reconstructions, and area integration, Int. J. for Numer. Meth. in Fl., 71, 1131–1151, 2013.

Yamaguchi, T., Feingold, G., and Larson, V. E.: Framework for improvement by vertical enhancement: A simple approach to improve representation of low and high-level clouds in large-scale models, J. Adv. Model Earth Sy., 9, 627–646, 2017.