Stabilized bases for high-order, interpolation semi-Lagrangian, element-based tracer transport[1]

Andrew M. Bradley[2]

Abstract

In a computational fluid model of the atmosphere, the advective transport of trace species, or *tracers*, can be computationally expensive. For efficiency, models often use semi-Lagrangian advection methods. High-order interpolation semi-Lagrangian (ISL) methods, in particular, can be extremely efficient, if the problem of property preservation specific to them can be addressed. Atmosphere models often use geometrically and logically nonuniform grids for efficiency and, as a result, element-based discretizations. Such grids and discretizations make stability a particular problem for ISL methods. Generally, high-order, element-based ISL methods that use the natural polynomial interpolant associated with a nodal finite-element discretization are unstable. We derive new bases having order of accuracy up to nine, with positive nodal weights, that stabilize the element-based ISL method. We use these bases to construct the linear advection operator in the property-preserving Interpolation Semi-Lagrangian Element-based Transport (Islet) method. Then we discuss key software implementation details. Finally, we show performance results for the Energy Exascale Earth System Model's atmosphere dynamical core, comparing the original and new transport methods. These simulations used up to 27,600 Graphical Processing Units (GPU) on the Oak Ridge Leadership Computing Facility's Summit supercomputer.

1 Introduction

An atmosphere model has three major subcomponents: first, the dynamical equations solver for resolved fluid flow and resolved thermodynamics, subsequently the *dynamics solver*; second, the resolved tracer *transport solver*; third, the subgrid *parameterizations*, which compute unresolved chemistry and physics processes, typically within vertical columns [Stensrud, 2009]. The first two subcomponents form the *dynamical core*. This article focuses on the second subcomponent, the transport solver.

In many earth system models, the parameterizations use many tracers, from a few to hundreds. The large number makes tracer transport computationally very expensive if the continuity equation solver from the dynamics solver is also used for tracer transport directly or with little modification. 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], which by default has 40 tracers, tracer transport takes approximately 75% of the dynamical core wall clock time and approximately 23% of the atmosphere model wall clock time on a typical computer cluster [Golaz et al., 2022, Fig. 3].

To speed up tracer transport, EAM version 2 (EAMv2) uses a new semi-Lagrangian (SL) tracer transport method [Bradley et al., 2022] that is 6.5 to over 8 times faster than the original Eulerian method in EAM version 1 in the cases of, respectively, low and high workload per computer node [Golaz et al., 2022, Fig. 3]. In addition, EAMv2 has separate grids for physics parameterizations and dynamics; the physics parameterizations run on a coarser grid and thus are 1.6 to 2.2 times faster in EAMv2 than in version 1 [Hannah et al., 2021, Golaz et al., 2022]. Together, these two

---

[2]Andrew M. Bradley (ambradl@sandia.gov), Sandia National Laboratories, Albuquerque, New Mexico, USA.
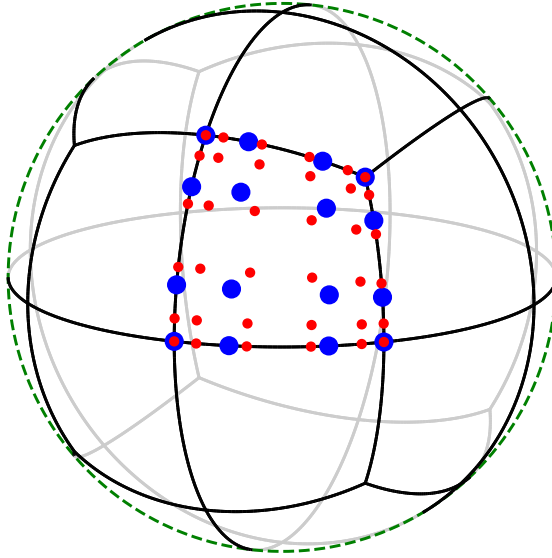
Figure 1: Cubed-sphere grid (black lines in foreground, gray lines in background) with 2×2 spectral elements per cube face. One element shows the subelement grids: $n_p^{\mathrm{v}} = 4$ dynamics (large blue circles) and $n_p^{\mathrm{t}} = 6$ transport (small red circles).

new features make EAMv2 approximately twice as fast as EAM version 1 on the same hardware [Golaz et al., 2022].

EAM uses the spectral element (SE) method, and the Gauss–Lobatto–Legendre (GLL) bases in particular, to discretize the equations of atmospheric flow, implemented in the High-Order Method Modeling Environment (HOMME) dynamical core [Dennis et al., 2005, 2012]. The *element grid* consists of horizontally unstructured quadrilateral elements extruded in the vertical dimension. Methods that can use unstructured element grids enable nonuniform discretization of a domain: for example, E3SM's regionally refined model (RRM) configurations [Tang et al., 2019]. We refer to such methods as *element based*.

Bradley et al. [2022], subsequently BBG22, describe and generalize the new transport and grid remap methods in EAMv2. In that work, we introduce the Interpolation Semi-Lagrangian Element-based Transport (Islet) method for extremely efficient, property-preserving tracer transport coupled to an EAM-like atmosphere model. In its most general form, the Islet method manages data on three grids, one grid for each of the three subcomponents of the atmosphere model: the dynamics, tracers, and parameterizations grids. The Islet method permits each subcomponent to run in its most efficient configuration, maximizing the overall atmosphere model's efficiency. Like EAMv2's two grids, the fully general Islet method's three grids share one element grid, and each grid has its own subelement grid. Thus, each subcomponent consists of element-based methods operating on the subcomponent's specific subelement degrees of freedom. The Islet method provides remap methods to transfer data among these grids. On the tracer grid, the method provides an efficient, property-preserving, element-based, interpolation semi-Lagrangian (ISL) transport method with a parameter, the basis size, to trade between transport speed and accuracy independently of the dynamics and parameterizations grids.

Figure 1 shows an example of a coarse cubed-sphere horizontal element grid on the sphere.

Each face of the cube has $n_e \times n_e$ elements; in this example, $n_e = 2$. The upper-right element of the front cubed-sphere face shows the subelement grids of the dynamics and transport solvers, each a tensor-product basis having $n_p^v = 4$ and $n_p^t = 6$, respectively, points in each dimension.

The Islet method's key enabling numerical technique is a new set of bases, the *Islet bases*. In the present article, which is a companion to BBG22, we explain why the standard high-order spectral element bases cannot be used in an ISL method. Then we derive the high-order Islet bases that the Islet method uses instead.

This article is structured as follows. Section 2 reviews technical background and related work. Section 3 describes a general method to find bases for element-based ISL methods and uses this method to derive the Islet bases. Section 4 describes the results of numerical experiments and a scaling study on the GPU-powered supercomputer Summit. Finally, Sect. 5 concludes.

2 Technical background

This article focuses on the source-less linear advection equation on the sphere:

$$\frac{D\bar{q}}{Dt} \equiv \frac{\partial \bar{q}}{\partial t} + \boldsymbol{u} \cdot \nabla \bar{q} = 0. \tag{1}$$

At position $\boldsymbol{x}$ and time t_1, the exact solution of (1) in terms of the solution at another time t_0 is

$$\bar{q}(\boldsymbol{x}, t_1) = \bar{q}(\boldsymbol{X}(t_0; \boldsymbol{x}, t_1), t_0). \tag{2}$$

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

$$\frac{d}{dt} \boldsymbol{X}(t; \boldsymbol{x}, t_1) = \boldsymbol{u}(\boldsymbol{X}(t; \boldsymbol{x}, t_1), t) \tag{3}$$

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.

A linear ISL advection operator can be specified by choosing a grid, providing an interpolation operator to evaluate the right hand side of (2) using the grid data, and providing a discretization of (3) to compute departure points for use in (2). Let a grid having grid points $\boldsymbol{x}_i$ discretize a domain, and let each grid point $\boldsymbol{x}_i$ have an associated global basis function $\psi_i(\boldsymbol{x})$. The scalar function $\bar{q}(\boldsymbol{x})$ is approximated as $\bar{q}(\boldsymbol{x}) \approx q(\boldsymbol{x}) \equiv \sum_i q_i \psi_i(\boldsymbol{x})$. For each grid point $\boldsymbol{x}_i$, an ISL method solves (3) with initial condition $\boldsymbol{X}(t_1; \boldsymbol{x}_i, t_1) = \boldsymbol{x}_i$ to find the approximate solution $\boldsymbol{x}_i^* \approx \boldsymbol{X}(t_0; \boldsymbol{x}_i, t_1)$. In general, $\boldsymbol{x}_i^*$ is not at a grid point. Then, following (2), an ISL method sets

$$q(\boldsymbol{x}_i, t_1) = q(\boldsymbol{x}_i^*, t_0) = \sum_j q_j(t_0) \psi_j(\boldsymbol{x}_i^*). \tag{4}$$

ISL methods are efficient; however, generally they are not mass conserving, and the stability of (4) depends on details of the approximations to (2) and especially (3). Efficient global property-preservation filters [Bradley et al., 2019] address conservation and other properties, leaving stability as the key challenge when designing an ISL method.

2.1 Related work

An alternative to ISL methods that can, in principle, address both shortcomings is the family of Lagrange-Galerkin (LG) SL methods [Douglas and Russell, 1982, Pironneau, 1982], sometimes also called characteristic Galerkin methods [Morton et al., 1988, Djoumna et al., 2008]. These methods require integration between basis functions. If exact integration is used, then an LG method is an L^2 projection, from which stability follows given conditions on the flow field [Ferretti and Mehrenberger, 2020]; in addition, the method is locally mass conserving. However, inexact quadrature can yield an unstable method [Morton et al., 1988]. Flux-form LG methods [Lauritzen et al., 2010, Lee et al., 2016, Erath and Nair, 2014] are automatically locally mass conserving regardless of exactness of quadrature; remap-form LG methods are not. However, the computational and communication costs of a remap-form method are very nearly independently of time step, while costs can grow linearly with time step in flux-form methods. For a given basis set, integration for either form makes the LG method computationally costlier than the interpolation method for two reasons: first, a larger computational cost due to sphere-to-reference point calculations and interpolant evaluations at many quadrature points; second, a larger communication volume because the discrete domain of dependence results from basis–basis interactions rather than basis–point interactions. Finally, it is a challenge to extend LG methods to arbitrarily high order.

The class of ISL methods contains three important subclasses: classical, meshless, and element-based. Classical and meshless ISL methods use centered, overlapping interpolants. In contrast, in an element-based ISL method, interpolants are nonoverlapping and, thus, not centered. The classical symmetric tensor-product interpolation (TPI) method generally cannot be applied to unstructured grids. In contrast, element-based TPI (EBTPI) methods, which include the Islet method, are well suited to this case. However, it is a challenge to stabilize high-order EBTPI methods, whereas centered, overlapping interpolants generally stabilize classical and meshless methods.

Meshless methods include, e.g., global and local radial basis function (RBF) [Flyer and Wright, 2007, Flyer and Lehto, 2010, Shankar and Wright, 2018] and generalized moving least squares (GMLS) [Mohammadi et al., 2021] methods, as well as other related kernel methods. These methods permit arbitrarily high order and, in some types, high degrees of smoothness. They are well suited to unstructured problems. In addition, because in most meshless methods interpolation at a departure point is roughly symmetric, stability is likely easier to obtain than with EBTPI methods. Meshless methods generally require more computation and greater communication volume per departure point than EBTPI methods. This cost difference is the primary motivation for pursuing the design of EBTPI methods; thus, we discuss the difference in some detail, first here and then in Sect. 4.2.

Consider the 2D transport problem. Let n be the number of nodes in the kernel method's support of a departure point and $m \leq n$ the number of basis functions in, e.g., a GMLS approximation. m is typically not much smaller than n, and in some methods $m = n$. n is for example in the range 17 to 85 in Shankar and Wright [2018]. m is perhaps 10 to 55 in the case of GMLS. Finally, let n_t be the number of tracers. We disregard one-time setup costs, which are unimportant, and work associated with computing departure-point data. In each time step, in the reconstruction phase, the GMLS method requires $O(nm^2)$ work per departure point. Other methods require a factorization only during one-time setup and $O(n^2)$ work per departure point in each reconstruction phase. All methods require $O(n)$ additional work per tracer in the interpolation phase. Our EBTPI method requires $O(n_p^2)$ work for reconstruction and then again per departure point and tracer, for n_p in the range 4 to 13 in this article and n_p^2 the equivalent of a kernel method's n. For n_t very large, computational costs are very roughly similar, but for n_t small, TPI methods can be much faster. Section 4.2 includes a detailed, but necessarily imprecise, comparison between our method

4

and that of Shankar and Wright [2018] to illustrate this point. Communication volume is generally higher in meshless methods because the interpolant supports are symmetric and overlapping. The communication halo for the reconstruction phase has a factor proportional to the support size, i.e., n. In contrast, in EBTPI methods, the equivalent factor is proportional to n_p rather than n_p^2, i.e., roughly $\sqrt{n}$. For a given order of accuracy (OOA), it is likely that $n < n_p^2$, but n is still much larger than n_p.

Bosler et al. [2017] and Taylor and Nave [2023] describe element-based ISL methods that use cubic Hermite interpolation on triangles, providing gradient data through reconstruction. Stability is not analyzed, but it is plausible the interpolant gives a stable ISL method because it is centered. The methods have orders of accuracy between 2 and 4, depending on details. Generalizing the methods to higher order is not straightforward. Note that Taylor and Nave [2023] implement only what Bosler et al. [2017] call the *indirect* method, making all errors in their results temporal and essentially none spatial because only the initial condition is ever sampled. The indirect method is not applicable to production models because remap of tracer fields must occur at least at the physics parameterizations time step, when two-way coupling occurs.

Finally, we review work on stability related to EBTPI methods. Stability analysis usually uses the standard *test problem* of periodic, uniform flow with no source terms, and requires a uniform (element) grid. Baptista and Oliveira [Baptista, 1987, Oliveira and Baptista, 1995] differentiate between compact and non-compact interpolants, essentially the same as our distinction between classical and element-based TPI methods. Let a *natural* interpolant be the unique polynomial interpolant that uses all available nodes in an element as its support, and similarly for a natural basis. For the specific subelement grid points they use, they find an ISL method using a compact natural interpolant is unstable on the test problem for elements having degree three and higher. In Sect. 3.8, we make a similar, but more general, statement.

More recently, Ferretti [2010], Ferretti [2013], and Ferretti and Mehrenberger [2020] prove the stability of the classical ISL method that uses overlapping, symmetric Lagrange interpolants of any odd (for symmetry) degree on a uniform grid in two cases. The first case is the test problem in any dimension; as Ferretti and Mehrenberger [2020] note, this case is also proven in Besse and Mehrenberger [2008]. The second case is nonuniform velocity in one dimension subject to a restriction on the time step related to the deformation gradient of the flow. Ferretti [2010] notes that the analysis approach used in this set of articles "cannot handle ... reconstructions performed by finite elements, which violate the assumptions of symmetry and translation invariance," while Ferretti and Mehrenberger [2020] write that "a partial result of stability for the case of piecewise quadratic finite element interpolations ... will be the object of a forthcoming study," but "a rigorous extension to the multidimensional case still seems out of reach." We conclude from this set of articles that analytical techniques are not available for the case of high-order EBTPI methods.

Others have developed high-order EBTPI methods, motivated by their efficiency and flexibility. Giraldo [1998] and Bochev et al. [2015] describe methods for unstructured, quadrilateral spectral elements with GLL collocation points, as we do. Giraldo [1998] uses analysis based on Fourier modes to conclude the method is unstable. However, Fourier modes are not eigenvectors in the case of high-order finite elements, and thus Fourier amplification factors are not eigenvalues. Our stability analysis implies these methods are *fully unstable*, as we define the term in Sect. 2.2.2.

Xiu and Karniadakis [2001] describe an EBTPI method for triangular elements using a high-order modal spectral element method but do not analyze stability. Natarajan and Jacobs [2020] describe an EBTPI method for use with a discontinuous spectral element grid using Chebyshev–Gauss collocation points, i.e., points not on the element edges. The characteristic departure point is a Gauss point, and they limit the time step so that no point leaves its element. Thus, while

the method is semi-Lagrangian, its purpose is not to take large time steps. They use the spectral element basis directly for interpolation. Stability analysis is not provided. Our results suggest that the linear operators in these methods are fully unstable.

In this article, we stabilize a class of high-order EBTPI methods by modifying basis functions. A common stabilization method is hyperdiffusion, as, for example, in the flux-form SE SL advection scheme (FF-SLSE) of Ullrich and Norman [2014]. The hyperdiffusion operator requires additional communication rounds per time step. Additionally, as the basis order increases, the hyperdiffusion-limited time step roughly decreases: in terms of the advective CFL number, from 2.44 for cubic to below 1 for quintic [Ullrich and Norman, 2014, Table 1]. For this reason, they write that "the FF-SLSE method may not be desirable for schemes requiring greater than third-order accuracy" [Ullrich and Norman, 2014, Sect. 3].

2.2 Stability

We use the standard *test problem* of source-less, periodic, uniform flow. Let $\boldsymbol{A}$ be the space–time evolution matrix for a linear discretization applied to the test problem; that is, $\boldsymbol{q}^{n+1} = \boldsymbol{A}\boldsymbol{q}^n$ for time steps n and $n+1$ and grid values $\boldsymbol{q}$. The key tool in the analysis of linear discretizations is the Lax-Richtmyer equivalence theorem, which states that a consistent linear discretization is convergent if and only if it is stable [e.g., LeVeque, 2002, Sect. 8.2]. Here, consistency means that the local truncation error converges to 0 as the time step and grid point spacing converge to 0. A necessary and sufficient condition for the stability of a linear discretization is that $\boldsymbol{A}$ is uniformly power bounded (e.g., LeVeque [2002, Sect. 8.3.2] and Strikwerda [2004, Sects. 9.4 and 11.5]): $\|\boldsymbol{A}^n\| \leq C$ for constant C and with $n\Delta t = T$, independently of (i.e., "uniform in") discretization parameters. A necessary condition for the bound on $\|\boldsymbol{A}^n\|$ to hold is that the maximum eigenvalue magnitude

$$\lambda_{\max}(\boldsymbol{A}) \equiv \max_i |\lambda_i(\boldsymbol{A})| \leq 1 \tag{5}$$

independently of discretization parameters [Strikwerda, 2004, Sect. 11.5]. Discretizations of (1) lead to matrices having at least one eigenvalue of magnitude at least one, and thus a maximum magnitude of exactly one if the discretization is stable.

2.2.1 Instability in the classical cubic ISL method

We now demonstrate two examples of settings more general than the test problem on a uniform grid in which the classical cubic TPI SL method does not satisfy the necessary condition for stability (5). Based on these examples, we conclude that it is extremely unlikely that there exists a high-order, EBTPI method that satisfies the condition in all settings.

To find each example, we construct a parameterized problem and discretization, then run a randomized search over parameter values to find an unstable instance. Finally, to simplify reproducibility, we round the parameter values to not more than five digits while still obtaining a maximum eigenvalue magnitude greater than one.

First, the classical ISL method applied to the test problem on a uniform grid with a symmetric Lagrange interpolant is stable [Ferretti and Mehrenberger, 2020, Besse and Mehrenberger, 2008]. However, on a nonuniform grid, it is not: "it is known that using [symmetric Lagrange interpolation] with unevenly spaced nodes results in general in an unstable scheme" [Ferretti, 2013, Sect. 5]. We can demonstrate this fact as follows. Let the periodic grid have the five grid points

$\{0, 0.11242, 0.44817, 0.78392, 0.88737\}$, where 0 is identified with 1 by periodicity. The cubic interpolant uses the two nearest Eulerian grid points on each side of a departure point as its support. Let the flow speed and time step be such that in each time step, the field translates by distance 0.33575. The space–time operator for this configuration has maximum eigenvalue magnitude exceeding $1 + 10^{-3}$, violating the necessary condition for stability.

Second, consider a doubly periodic, two-dimensional grid $[0, 1]^2$, where again 0 is identified with 1 by periodicity. The grid consists of uniformly sized rectangles. The flow, while still nondivergent and constant in time, is shear in space. Its velocity is $(u, v) = (s, s)$, where $s \equiv 1 + \cos(2\pi(0.342 + x - y))$. With 15 unique grid points in the x direction and 13 in the y and time step 0.276, the space–time operator has maximum eigenvalue magnitude exceeding $1 + 10^{-2}$. For time steps 0.273 and 0.279, the maximum eigenvalue is 1. Now, the deformational Courant number is $\max_{x,y} \max(|u_x|, |u_y|, |v_x|, |v_y|)\Delta t = 2\pi\Delta t$ [e.g., Smolarkiewicz and Pudykiewicz, 1992]; thus, with $\Delta t = 0.276$, the number exceeds 1. However, trajectories in this flow cannot intersect, and a departure quadrilateral cannot self-intersect or have an orientation opposite its arrival quadrilateral.

2.2.2 Sparse instability

Instability in SL methods tends to have different qualitative and quantitative characteristics than instability in explicit-in-time Eulerian methods. For many schemes of the latter type, the maximum eigenvalue magnitude grows, often roughly linearly, with the quotient of the time and space steps. Thus, forcing a scheme to exhibit instability is achieved simply by increasing the time step beyond the stable time step by a factor not much larger than 1, e.g., 1.5. In contrast, unstable SL methods for the advection equation often have maximum eigenvalue magnitudes of the form $1 + \epsilon$ with $\epsilon \lessapprox 10^{-2}$, with no particular dependence on time step. This type of instability can be divided into two types, which we now describe.

Figure 3, described in Sect. 3.7, plots $\log_{10}(\lambda_{\max} - 1)$ as a function of step size for a set of bases applied to the test problem with a uniform grid (solid lines) and a nonuniform grid (markers). We need to proceed further in this article before we can describe this figure in detail. However, one qualitative feature is apparent now. For one basis, with corresponding line pattern red with small circles, $\lambda_{\max}$ is substantially above 1 for almost all step sizes; thus, the corresponding ISL method is essentially unconditionally unstable even in the case of a uniform grid. For the other two, green with $\times$ markers and black with large circle markers, $\lambda_{\max} - 1$ is at machine precision for almost all values, with one or a small number of spikes to larger values in the case of a nonuniform grid. We refer to this second pattern of $\lambda_{\max} > 1$ values as *sparse*. In the two examples in Sect. 2.2.1, the classical cubic ISL method's $\lambda_{\max} > 1$ pattern is sparse with respect to step size and other parameters, in the cases of a nonuniform grid or flow field.

We speculate that all high-order ISL methods have at least a sparse $\lambda_{\max} > 1$ pattern when applied to some problems, as a function of discretization or problem parameters. To categorize those methods that are usable in practice, we write that a method has *sparse instability* or is *sparsely unstable* if it satisfies (5) for the test problem with a uniform grid and has no worse than a sparse pattern when applied to most problems. We write "most" because, in the absence of theoretical analysis proving otherwise, we must assume some problems can be found on which sparsely unstable methods are unstable to a greater degree. We refer to a method that does not satisfy these requirements as *unstable* or, for clarity, *fully unstable*.

Based on this discussion, our objective in the remainder of this article is to design highly efficient, high-order, element-based, tensor-product interpolation semi-Lagrangian methods that

are necessarily sparsely unstable but are not fully unstable.

3 The Islet bases

In this section, we develop a method to search for bases that meet a stability criterion, ranked according to additional quality heuristics, and then present a set of bases up to order of accuracy (OOA) 9. Section 3.1 lists notation and definitions. Section 3.2 defines element-based interpolation bases. Section 3.3 defines our stability criterion and summarizes technical details to compute the criterion; A provides details. Section 3.4 constructs the search space of potential bases. Section 3.5 defines accuracy and stability heuristics to augment the stability criterion in the search. Section 3.6 briefly summarizes the search procedure; C provides details. Section 3.7 enumerates the Islet bases resulting from the search procedure. Finally, Sect. 3.8 examines the stability of natural bases on arbitrary, up to symmetry, subelement grid points to complete the argument that the natural basis is not viable for high-order, element-based ISL methods.

3.1 Notation and definitions

Vectors and matrices are written in boldface; continuum fields are not. Thus, $\boldsymbol{f}$ could be the nodal values that, combined with a basis, represent the continuum field f. Indexing starts at 0. To minimize nested subscripting, in many cases we denote entries of a vector parenthetically, and we use colon notation to indicate an index list. For example, $\boldsymbol{x}(i)$ is element i of $\boldsymbol{x}$, and $\boldsymbol{x}(i:i+2)$ produces the 3-element vector containing elements i through $i+2$ inclusively. The list $n-1:-1:0$ decrements the index; thus, $\boldsymbol{x}(n-1:-1:0)$ is a vector with elements in the opposite order of those in $\boldsymbol{x}$. It is sometimes useful to annotate a matrix or vector with its size; for example, each of the matrix of zeros $\boldsymbol{0}^{m\times n}$ and the identity matrix $\boldsymbol{I}^{m\times n}$ has size $m\times n$, and $\boldsymbol{x}^n$ is an n-vector. In equations, we often annotate only the first instance of the vector or matrix to provide useful data while minimizing clutter. $\boldsymbol{e}_i$ is column i of the identity matrix.

We use polynomial interpolants in Lagrange form extensively. We write the polynomial interpolant through values $\boldsymbol{y}^n$ at points $\boldsymbol{x}^n$, evaluated at x, in Lagrange form as follows:

$$\mathcal{L}(x;\boldsymbol{y}^n,\boldsymbol{x}^n) \equiv \sum_{i=0}^{n-1}\boldsymbol{y}(i)\prod_{j=0,j\neq i}^{n-1}\frac{x-\boldsymbol{x}^n(j)}{\boldsymbol{x}^n(i)-\boldsymbol{x}^n(j)}.$$

If the second argument to $\mathcal{L}$ is a matrix $\boldsymbol{Y}^{n\times k}$ rather than a vector $\boldsymbol{y}^n$, then the output is a vector whose element j corresponds to column j of $\boldsymbol{Y}^{n\times k}$. If, in addition, the first argument is a vector $\boldsymbol{z}^m$, then the output is an $m\times k$ matrix whose ith row corresponds to $\boldsymbol{z}^m(i)$ and jth column corresponds to column j of $\boldsymbol{Y}^{n\times k}$.

3.2 Element-based tensor-product interpolation bases

Now we define (element-)*local* bases. We use ϕ to denote a local basis function; in (4), we used ψ to denote a global basis function. A global basis assembles local basis functions so that the resulting global basis functions are one-to-one with global degrees of freedom. In this section, the local basis is the more convenient to use because we can construct and analyze a basis with respect to a single *reference element.*

We use *nodal* bases. This means, first, that the basis function ϕ_i is one-to-one with node i in the reference element. Second, $\phi_i(\boldsymbol{x}^n(j)) = 1$ if $i = j$ and is 0 otherwise.

Let a reference element contain $n = d + 1$ subelement grid points $\boldsymbol{x}^n$. We can write the natural basis for degree-d polynomials over this element using $\mathcal{L}$. The natural basis functions are $\phi_i^n(x) \equiv \mathcal{L}(x; \boldsymbol{e}_i^n, \boldsymbol{x}^n)$ or, using a matrix input to $\mathcal{L}$, $\boldsymbol{\phi}^n(x) \equiv \boldsymbol{\mathcal{L}}(x; \boldsymbol{I}^{n \times n}, \boldsymbol{x}^n)$.

In this work, we take $\boldsymbol{x}^n$ to be points in the reference-element domain $[-1, 1]$. Usually, $\boldsymbol{x}^n = \boldsymbol{x}_G^n$, the GLL points, or a subset of these. *Region r* of the domain $[-1, 1]$ is $[\boldsymbol{x}^n(r), \boldsymbol{x}^n(r+1)]$. A *middle region* is region $r = n/2 - 1$ if n is even; if n is odd, then there is no middle region.

Figure 2 shows two local nodal bases for a reference element having $n = 6$ nodes. The natural basis is plotted with dashed lines; the other, solid. Because there are six nodes, there are also six nodal basis functions in a nodal basis. They take values in the region $[-1, 1]$. Each basis function has value 1 at one node and 0 at the others. We describe further details of this figure in Sect. 3.4.2.

The *nodal weight w_i* is the integral of basis function ϕ_i over $[-1, 1]$: $w_i \equiv \int_{-1}^{1} \phi_i(x) \, \mathrm{d}x$. It can be thought of, roughly, as the mass or volume of the node. It is standard to require that all nodal weights be positive, and usually we want the largest weight to be not too much larger than the smallest weight.

3.3   Test-problem stability

The review in Sect. 2.2 of the stability analyses of Ferretti and collaborators suggests that there are no stability analysis methods available for the class of discretizations we develop in this article, even in the restricted case of the test problem discretized on a uniform element grid. Thus, we take a semi-analytical approach. We describe a standard method to efficiently numerically compute the eigenvalues of our operators, and then we apply this method to numerically test methods at a large but finite set of points in the discretization's parameter space and with finite-precision results. We write *semi-analytical* because our approach is not simply testing the method on a set of validation problems, which one might call *empirical*. The eigenvalue analysis is complete up to the limits of finite precision and a finite lattice of parameter values.

Let $\lambda_{\max}$ denote the maximum eigenvalue magnitude we seek. The necessary condition implies $\lambda_{\max} \leq 1$ for the test problem. In finite precision, we expect error in the computation and require $\lambda_{\max} \approx \tilde{\lambda}_{\max} \leq 1 + \varepsilon$, where $\varepsilon > 0$ is a tolerance near machine precision and $\tilde{\lambda}_{\max}$ is a numerical approximation to the true $\lambda_{\max}$.

We say that a discretization is *test-problem stable* (t.p.s.) if, for every sample in a large but finite sampling of the parameter space of the discretization, the corresponding operator has $\tilde{\lambda}_{\max} \leq 1 + \varepsilon$. We refer to this numerical test as the *stability criterion*. For brevity, we also say that a basis is or is not t.p.s., by which we mean that the ISL method that uses the basis is or is not t.p.s.

The ISL space–time operator for an element-based method applied to the test problem on a uniform element grid having N elements has the following structure. Let a matrix $\boldsymbol{B}^{d \times (d+1)} \equiv \left(\bar{\boldsymbol{B}}^{d \times d} \quad \boldsymbol{b}^{d \times 1} \right)$. Let $\boldsymbol{C}_k$ be the permutation matrix that circularly shifts its left operand k columns to the right and its right operand k rows up. Then

$$\boldsymbol{A} \equiv \boldsymbol{C}_{-r'} \left[\left(\boldsymbol{I}^{N \times N} \otimes \bar{\boldsymbol{B}} \right) + \left(\boldsymbol{I}^{N \times N} \otimes (\boldsymbol{b} \boldsymbol{e}_{d-1}^T) \right) \boldsymbol{C}_1 \right], \tag{6}$$

where $\otimes$ denotes the Kronecker product, T denotes the matrix transpose, and translation distance determines the number of rows r' blocks are circularly shifted down (up if $r' < 0$).

9

$\boldsymbol{A}$ is a block circulant matrix. The Fourier transform block-diagonalizes it [Vichnevetsky and Bowles, 1982, Idelsohn et al., 1995, Cohen, 2001, Ainsworth, 2004]. Thus, the spectrum of $\boldsymbol{A}$ can be obtained by solving eigenvalue problems associated with just $\boldsymbol{B}^{d \times (d+1)}$. For $\boldsymbol{A}$ corresponding to N elements, one must solve N independent eigenvalue problems of size $d \times d$. Thus, they can be solved in parallel, and each has a cost that is a function of d rather than Nd. A derives this procedure.

Now we complete the definition of test-problem stability. Let $\mathcal{M}$ be the collection of data describing a basis. A discretization of the test problem on a uniform element grid using a particular basis has two parameters. The first parameter is N, the number of elements. The second is the *translation distance* Δx, which is the product of the scalar flow speed and the time step. $\boldsymbol{B}$ is a function of $\mathcal{M}$ and Δx, and $\boldsymbol{A}$ is a function of $\boldsymbol{B}$, Δx, and N.

Let $\lambda_{\max}(\mathcal{M})$ be the maximum eigenvalue magnitude corresponding to basis $\mathcal{M}$. Its approximation, $\tilde{\lambda}_{\max}(\mathcal{M})$, is the maximum numerically computed eigenvalue magnitude of $\boldsymbol{A}$ among a large sample of possible values of N and Δx. We can use two symmetries to reduce the translation distance set. First, on a uniform element grid having element size one, any translation $\Delta x + n$ for integer n yields the same maximum eigenvalue magnitude as Δx. Integer increments to Δx correspond to row shifts of the space–time operator, to which the maximum eigenvalue magnitude is invariant. Second, we consider only symmetric bases. Basis symmetry implies that only $\Delta x \in (0, 1/2]$ needs to be searched, for the following reasons: $\lambda_{\max}$ is the same for translations $\pm \Delta x$ by basis symmetry, and then it is the same for $-\Delta x + 1$ by the first point. Thus, computing $\tilde{\lambda}_{\max}(\mathcal{M})$ requires solving $d \times d$ eigenvalue problems for a finite subset of the two-dimensional parameter space $\Delta x \in (0, 1/2]$, $\theta \in [0, 2\pi)$, where, as explained in A, θ is the parameter in the eigenvalue computation corresponding to N.

If a basis is t.p.s. for the one-dimensional test problem and has tensor-product structure in multiple dimensions, then the basis is t.p.s. for the multidimensional test problem, too, because the tensor-product structure makes the discretization of the multidimensional test problem decouple by dimension.

3.4 The sets of potential bases

In this subsection, we construct sets of bases that we then search for t.p.s. ones.

3.4.1 Constraints on potential bases

We study nodal bases over $\boldsymbol{x}^n$ that are more complicated than the natural basis. The element grid points $\boldsymbol{x}^n$ are part of the basis specification and thus can vary. The coefficients of the ISL space–time operator correspond to interpolation at departure points. We consider piecewise-polynomial interpolants having the following constraints. In B we state these constraints precisely; here, we summarize them sufficiently to motivate the subsets we choose to study in this article.

1. Given a departure point x, the interpolant is a linear operator in the field values $\boldsymbol{y}^n$ at grid points $\boldsymbol{x}^n$. This constraint makes the discretization linear and is standard in ISL methods.

2. In each region $[\boldsymbol{x}^n(r), \boldsymbol{x}^n(r+1)]$, the interpolant is a polynomial.

3. The interpolant must recover a degree-p polynomial with specified $p = s - 1$ and $2 \leq s \leq n$. We call this the *order constraint*.

4. In each region $[\boldsymbol{x}^n(r), \boldsymbol{x}^n(r+1)]$, the interpolant must interpolate nodes r and $r+1$; that is, it must have values $\boldsymbol{y}(r)$ and $\boldsymbol{y}(r+1)$, respectively, at these nodes. We call this the *region interpolation*

10

constraint. This interpolation property is standard for nodal ISL methods. We emphasize *region* here because, later, it will help to clarify how we apply this constraint.

5. The nodal basis function associated with node i of n must have mirror symmetry around $x = 0$ with the basis function associated with node $n - i - 1$. Thus, the points in $\boldsymbol{x}^n$ must be symmetric around 0. Symmetric bases are standard in many methods.

3.4.2 Nodal subset bases

We focus on two nested subsets of the infinite set of all bases that satisfy constraints 1–5: *nodal subset bases* (n.s. bases) and a subset of these, *offset nodal subset bases* (o.n.s. bases). Furthermore, we restrict these to the finite sets that use $\boldsymbol{x}^{n_p} = \boldsymbol{x}_G^{n_p}$; we sometimes qualify these as the GLL (offset) nodal subset bases.

A nodal subset n_p-basis $\mathcal{M}$ over an element is specified by two data sets, $\mathcal{M}(r)$ and $\mathcal{M}_{\boldsymbol{x}}$. First, it has $n_p - 1$ sets of support nodes, $\mathcal{I}_r \equiv \mathcal{M}(r)$, one for each region $r \in \{0, \ldots, n_p - 2\}$. For brevity, we call a set of support nodes $\mathcal{I}_r$ a *support*. Each support is a subset of the full set of nodes: $\mathcal{I}_r \subseteq \{0, \ldots, n_p - 1\}$. The number of nodes in region r's support is $(n_p^{\text{sub}})_r \equiv |\mathcal{I}_r|$. The region interpolation constraint requires $r, r+1 \in \mathcal{I}_r$. Second, a nodal subset n_p-basis has the points $\boldsymbol{x}^{n_p} \equiv \mathcal{M}_{\boldsymbol{x}}$.

We write the vector of reference domain coordinates of the support nodes for a region, the region's *support points*, as $\boldsymbol{x}^{n_p}(\mathcal{I}_r)$. Given these data, the basis functions take the values $\boldsymbol{\phi}(x) = \mathcal{L}(x; \boldsymbol{I}^{n_p \times n_p}(\mathcal{I}_r, :), \boldsymbol{x}^{n_p}(\mathcal{I}_r))$ in region r, i.e., for $x \in [\boldsymbol{x}^{n_p}(r), \boldsymbol{x}^{n_p}(r + 1)]$.

Offset nodal subset bases in addition have consecutive indices in each $\mathcal{I}_r$, and thus a region's support can be defined by its number of support nodes and an offset index giving the smallest index node in the support.

So far, these definitions satisfy the first four constraints. Using the symmetry constraint, basis data for regions $r \geq \lfloor n_p/2 \rfloor$ are constructed by symmetry of the basis set from regions $r < \lfloor n_p/2 \rfloor$. In addition, when n_p is even, the symmetry constraint implies that the middle region support, $\mathcal{I}_r$ with $r = n_p/2 - 1$, must correspond to support points that are symmetric around $x = 0$.

Figure 2 shows an example of a GLL n.s. basis for $n_p = 6$. Each color corresponds to a basis function. Dashed curves are the natural GLL basis functions; the natural basis is also an n.s. basis. Solid curves are the Islet n.s. basis functions; this basis corresponds to the entry for $n_p = 6$ in Table 1. There are five regions. The green span shows region 1. The red arrows point to the nodes in the support of region 1; the red $\times$ is beneath the one node not in region 1's support. Referring now to Table 1, regions 0 to 2 have $(n_p^{\text{sub}})_r$ values, in region order, of $\{5, 5, 6\}$, implying the basis provides OOA 4. Region 1 has support $\mathcal{I}_1 = \{0, 1, 2, 3, 5\}$. Because this support omits node 4 and includes nodes 3 and 5, this basis is not an o.n.s. basis; $\mathcal{I}_1$ does not contain fully consecutive nodes. By symmetry, region 3 has support $\mathcal{I}_3 = \{0, 2, 3, 4, 5\}$. Region 2 has support $\mathcal{I}_2 = \{0, 1, 2, 3, 4, 5\}$ and thus uses all available nodes. The support nodes for region 2 are symmetric around the element center, thus satisfying the symmetry constraint. Consider now the basis function values as they pass through region 1. Each basis function can be identified with the node at which it takes value 1. Basis functions 0, 1, 2, 3, and 5 all have nonzero values in region 1 because these nodes are in $\mathcal{I}_1$. Basis function 4 has function values 0 throughout region 1 because node 4 is not in $\mathcal{I}_1$. As an example of one basis function, basis function 2 has the values $\mathcal{L}(x; (0\ 0\ 1\ 0\ 0)^T, \boldsymbol{x}_G^6([0, 1, 2, 3, 5]))$ in region 1.

Let $r = \text{region}(x; \boldsymbol{x}^n)$ be the region containing x. If $x = \boldsymbol{x}^n(r)$, $\text{region}(x; \boldsymbol{x}^n)$ can return r or $r - 1$; the value does not affect calculations because the interpolant is continuous everywhere.
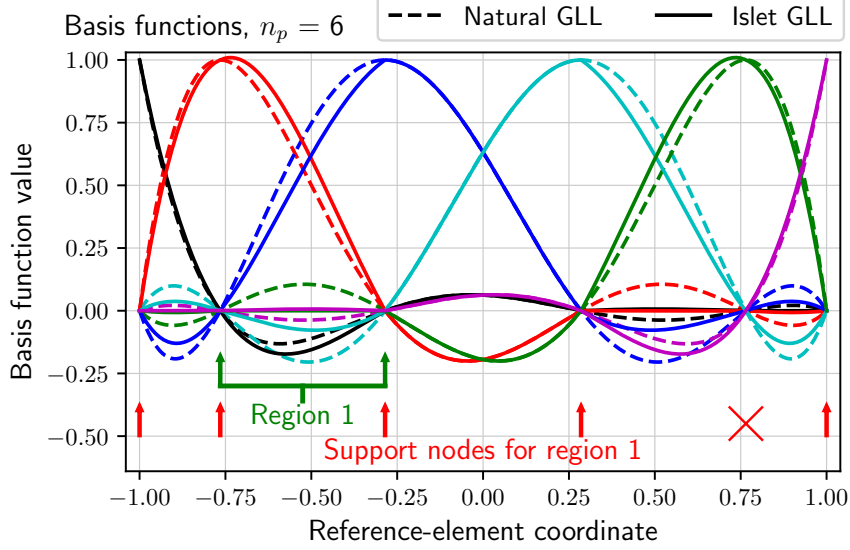
Figure 2: Element-local basis functions for the Islet $n_p = 6$ GLL nodal subset basis listed in Table 1. Each curve's color corresponds to a basis function. Each line pattern corresponds to a basis type, as listed in the legend. The green span shows region 1. The red arrows point to the nodes in the support of region 1; the red $\times$ is beneath the one node not in region 1's support.

Let n_p^{submin} be the minimum value of $|\mathcal{I}_r|$ over r. When present, a subscript $\mathcal{M}_{n_p}$ gives the total number of unique support nodes for the basis, and a superscript $\mathcal{M}_{n_p}^{n_p^{\text{submin}}}$ gives the minimum value of $|\mathcal{M}(r)|$ in the basis. Thus, for example, $\mathcal{M}_4^4$ denotes the natural GLL basis.

3.5 Heuristics

We find that for each value of n_p, especially larger values, the set of potential bases contains a large number of t.p.s. bases sharing the same maximum available OOA. Thus, to choose the best t.p.s. basis, we develop two heuristics to assess quality: one to approximately quantify accuracy and the other to approximately quantify instability on other than uniform grids. We want our heuristics to use nothing more than the test problem. For example, we do not want to compare the accuracy of various bases by looking at their results for validation problems, a procedure that might not be sufficiently general.

3.5.1 Accuracy

We start by describing the accuracy heuristics. Let $p_d(x)$ be a degree-d polynomial interpolant defined by support points $\boldsymbol{x}^n$, $n = d + 1$, in $[-1, 1]$ and including the endpoints, and nodal values $f(\boldsymbol{x}^n)$, with $f(x) \in C^{d+1}$ for $x \in [-1, 1]$. A standard result [e.g., Lozier, 2003, Sect. 3.3(i)] is that the approximation error is bounded by

$$|p_d(x) - f(x)| \leq \frac{1}{n!} \left| \prod_{i=0}^{n-1} (x - \boldsymbol{x}^n(i)) \right| \max_{\xi \in [-1,1]} \left| f^{(n)} \right|. \tag{7}$$

12

We define the following accuracy heuristics for the l_1, l_2, and l_∞ norms by dropping the f-dependent $f^{(n)}$ factor, leaving expressions that depend only on the supports and $\boldsymbol{x}^n$. Let

$$e(x; \mathcal{M}) \equiv \frac{1}{n!} \left| \prod_{i=0}^{n-1} (x - \boldsymbol{x}(\mathcal{I}_r(i))) \right|, \text{ where } r = \text{region}(x; \boldsymbol{x}^n), \ n = |\mathcal{I}_r|.$$

The error heuristics for the basis are then

$$a_1(\mathcal{M}) \equiv \int_{-1}^{1} e(x; \mathcal{M}) \, \mathrm{d}x, \quad a_2(\mathcal{M}) \equiv \left(\int_{-1}^{1} e(x; \mathcal{M})^2 \, \mathrm{d}x \right)^{1/2}, \quad a_\infty(\mathcal{M}) \equiv \max_{x \in [-1,1]} e(x; \mathcal{M}).$$

We refer to these collectively as $a_{1,2,\infty}$ for brevity. In Sect. 4.4, we illustrate the correlation between a_2 and the accuracy the corresponding methods achieve on particular validation problems.

3.5.2 Instability on a perturbed uniform grid

Based on the results in Sect. 2.2.1, for any high-order, t.p.s. $\mathcal{M}$, we expect $\tilde{\lambda}_{\max}(\mathcal{M}) > 1 + \varepsilon$ for some nonuniform element grids. We expect the method still to be usable in practice because, first, we expect $\tilde{\lambda}_{\max}(\mathcal{M}) - 1 > 0$ to be small and, second, we expect the deviation from stability to have a sparse pattern. To quantify this instability, we construct the following *perturbed uniform grid heuristic*, denoted $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\mathcal{M})$.

Conceptually, we want to maximize $\lambda_{\max}(\boldsymbol{A})$ over all space–time evolution operators $\boldsymbol{A}$ for the test problem using basis $\mathcal{M}$, for all translation distances and over the set of grids that are perturbed slightly and randomly from uniform. The grid perturbation could be implemented by, e.g., independent and identically distributed perturbations to each element boundary node, with the perturbations bounded by, e.g., 10% of the uniform element size. The result of this maximization is the value $\lambda_{\max}^{\mathrm{PUG}}(\mathcal{M})$. The t.p.s. basis having the smallest $\lambda_{\max}^{\mathrm{PUG}}(\mathcal{M})$ is the best, according to this heuristic. The heuristic we develop is a tractable approximation to this maximization problem.

To compute $\tilde{\lambda}_{\max}(\mathcal{M})$ for the test problem, recall that we construct the matrix $\boldsymbol{B}$ corresponding to $\mathcal{M}$ and a particular Δx and use the methods described in Sect. 3.3 and A. In this case, $\boldsymbol{B}$ represents the action of the ISL operator due to one source element. To develop the approximation $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\mathcal{M}) \approx \lambda_{\max}^{\mathrm{PUG}}(\mathcal{M})$, we construct $\boldsymbol{B}$ to represent the action of the ISL operator due to multiple contiguous nonuniform source elements, as follows.

First, choose an element patch having n_e^{sub} subelements. Second, perturb each element boundary inside the patch by $U(-\delta/2, \delta/2)$, where $U(a, b)$ is the uniform probability distribution over $[a, b]$ and δ is fraction of the subelement size. Thus, a subelement has size $U(1 - \delta, 1 + \delta)$ relative to an unperturbed size of 1. Third, compute $\boldsymbol{B}$ as one matrix for the whole patch. The result corresponds to periodic translation on a nonuniform element grid tiled by the patch. Fourth, approximate $\lambda_{\max}(\boldsymbol{B})$ almost the same as usual. The one modification is the set of Δx values; we explain this modification in Sect. 3.7. To compute $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\mathcal{M})$, repeat these four steps multiple times for a set of n_e^{sub} values and take the maximum over the $\tilde{\lambda}_{\max}$ values. In practice we choose $n_e^{\mathrm{sub}} \in \{3, \ldots, 15\}$. The other parameter we must determine is δ. In practice we choose a fixed $\delta = 0.01$, the choice of and insensitivity to which we explain in Sect. 3.7.

3.6 The search procedure

We developed software to search for t.p.s. n.s. bases. Because there are many of these, especially as n_p grows, we use the heuristics described in Sect. 3.5 to choose among the t.p.s. ones. In addition,

n_p	OOA	n_p^{sub}	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 1: Islet GLL nodal subset bases. Each row provides the data for a basis. The columns are n_p; order of accuracy; the support size n_p^{sub} for each region, ordered from the left region to the middle; and the support for each region, again ordered from left to middle. For offset nodal subset bases, supports are given by offsets. For general nodal subset bases, supports are given by nodal subsets. The basis for $n_p = 4$ is described in Sect. 3.7.2. For all bases, the support points are GLL points.

we require that each nodal weight be positive so that a positive nodal value is associated with a positive mass. C describes the search procedure in detail.

3.7 The Islet bases

3.7.1 Nodal subset bases

Table 1 lists the resulting set of GLL n.s. bases, one for each value of n_p between 5 and 13. Each row provides a formula for the row's n_p value. Columns are n_p, order of accuracy $n_p^{\text{submin}} - 1$, the support sizes n_p^{sub} for each region ordered left to middle, and the supports. For o.n.s. bases, supports are given by offsets. For general n.s. bases, supports are given by nodal subsets, again ordered from left region to middle. Supports for the right half of the reference element are determined by the symmetry constraint. The bases for $n_p = 6$ and 9 are n.s. but not o.n.s. Section 3.7.2 describes the basis for $n_p = 4$, which is not an n.s. basis.

As an example of an o.n.s. basis, consider the $n_p = 8$ basis. Regions 0 to 3 have $(n_p^{\text{sub}})_r$ values, in region order, of {6, 6, 7, 6}. There are seven regions total, and region 3 is a middle region because n_p is even. By symmetry, the $(n_p^{\text{sub}})_r$ values for regions 0 to 6 are {6, 6, 7, 6, 7, 6, 6}. Because n_p^{submin} is 6, corresponding to a degree-5 interpolating polynomial, the basis yields an ISL method having OOA 5. The offsets for regions 0 to 3 are {0, 0, 0, 1}. Thus, region 0 has support {0, 1, 2, 3, 4, 5}, and the middle region, region 3, has support {1, 2, 3, 4, 5, 6}. The support nodes for region 3 are

14

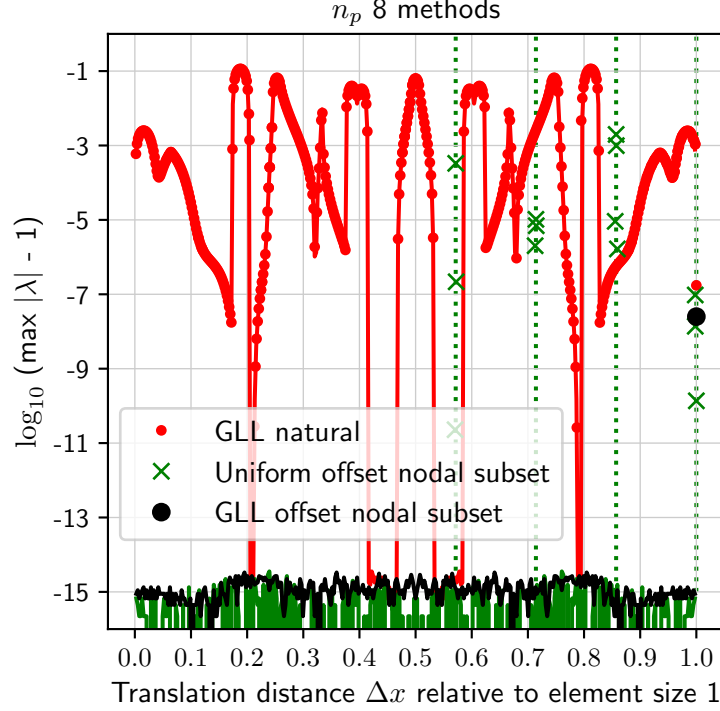Figure 3: $\tilde{\lambda}_{\max}(\Delta x) - 1$ (solid lines) and $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\Delta x) - 1$ (markers) for the natural GLL (red, small circles), uniform-points offset nodal subset (green, $\times$), and Islet GLL nodal subset (black, large circle) $n_p = 8$ bases. Green dotted vertical lines mark multiples of $1/(n_p - 1) = 1/7$.

symmetric around the element center to satisfy the symmetry constraint. By symmetry with region 0, region 6 has support $\{2, 3, 4, 5, 6, 7\}$.

Figure 3 illustrates $\tilde{\lambda}_{\max}$ and $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ for three $n_p = 8$ bases. Each color corresponds to a basis: natural GLL, red; an o.n.s. basis with uniform points, green; Islet, black. The uniform-points basis is an o.n.s. basis with n_p^{sub} values $\{4, 4, 4, 4\}$ and offsets $\{0, 0, 1, 2\}$. A solid line shows the maximum eigenvalue magnitude of the space–time operator as a function of translation distance $\Delta x \in (0, 1]$, where $\Delta x = 1$ corresponds to one full element. Note the symmetry around $\Delta x = 1/2$ of these curves. Markers show $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ as a function of Δx, with $\delta = 0.01$.

The natural GLL basis is not t.p.s.; thus, $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ values do not add new information and agree closely with $\tilde{\lambda}_{\max}$ values except at $\Delta x = 1$, where $\tilde{\lambda}_{\max} \approx 1$ but $\tilde{\lambda}_{\max}^{\mathrm{PUG}} > 1 + 10^{-7}$. The $\tilde{\lambda}_{\max}$ curve has values almost entirely above $1 + 10^{-7}$, with values as large as at least $1 + 10^{-1}$. For the Islet basis, the only $\tilde{\lambda}_{\max}^{\mathrm{PUG}} > 1 + \varepsilon$, where $\varepsilon < 10^{-13}$, occurs at $\Delta x = 1$. In the case of the t.p.s. basis having uniform points, $\tilde{\lambda}_{\max}^{\mathrm{PUG}} > 1 + \varepsilon$ at translations that are clustered around multiples of $1/7$ and larger than $1/2$. In general, we observe $\tilde{\lambda}_{\max}^{\mathrm{PUG}} > 1 + \varepsilon$ clustered around values of Δx at which most of the translated nodes nearly overlie the Eulerian nodes in a different element. For the Islet bases, these are integer values of Δx other than 0. For a t.p.s. uniform-points basis, these are multiples of $1/(n_p - 1) \geq 1/2$. These $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ values are an example of sparse instability: $\tilde{\lambda}_{\max}^{\mathrm{PUG}} > 1 + \varepsilon$ in only small parts of the parameter space Δx.

In our basis search procedure, we compute $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\mathcal{M})$ at Δx values that follow these observations, rather than the values $\Delta x \in (0, 1/2]$ used when computing $\tilde{\lambda}_{\max}(\mathcal{M})$. Since the discrete set size is much smaller than for the computation of $\tilde{\lambda}_{\max}(\mathcal{M})$, we can run more trials of randomly perturbed
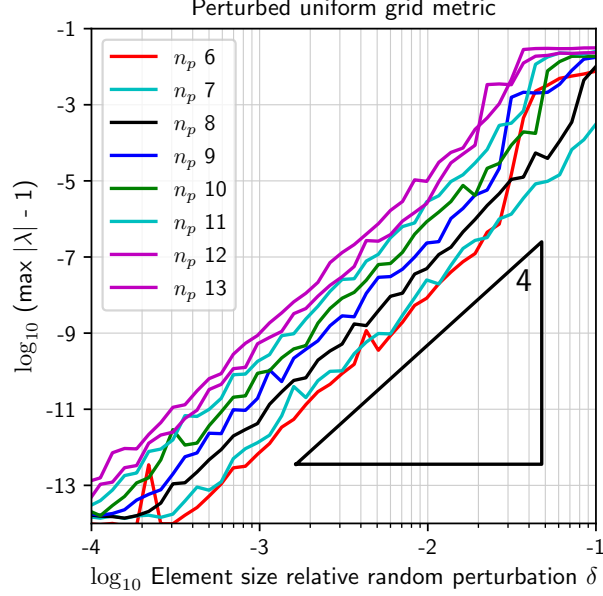
Figure 4: $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\delta) - 1$ for the bases in Table 1 with $n_p \geq 6$. The triangle provides a δ^4 reference slope.

subelements in each test.

A search with our software shows that t.p.s. uniform-points n.s. bases having OOA at least 3 exist for $n_p \geq 8$, and in terms of accuracy in practice, these are potentially useful. However, because uniform-points n.s. bases exhibit much larger $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ than GLL-node n.s. bases and are limited to OOA not more than 3 for $n_p \leq 13$ (the largest n_p we have checked), we do not consider them further in this article.

Figure 4 plots $\tilde{\lambda}_{\max}^{\mathrm{PUG}} - 1$ vs. subelement perturbation δ for the bases in Table 1, excluding those for $n_p \leq 5$ because those curves have a number of drops to machine precision values that would clutter the plot. Empirically, $\tilde{\lambda}_{\max}^{\mathrm{PUG}} - 1$ is proportional to δ^k, $k = 4$, for these bases, and for a fixed value of δ, it tends to increase with n_p. Not all t.p.s. GLL n.s. bases fall off at this rate $k = 4$. Some fall off more slowly, $k < 4$, indicating greater instability on nonuniform grids. We have not specifically focused a search on attempting to find a t.p.s. basis with rate $k > 4$, but we have not observed any. Based on this plot, when computing $\tilde{\lambda}_{\max}^{\mathrm{PUG}}(\mathcal{M})$ in our basis search procedure, we choose the specific value $\delta = 0.01$, with search results extremely insensitive to this value over a large range of δ.

3.7.2   An optimized basis for $n_p = 4$

For the $n_p = 4$ GLL nodes, the t.p.s. n.s. basis has n_p^{sub} values $\{3, 4\}$ with offsets $\{0, 0\}$, referred to subsequently as $\mathcal{M}_4^3$. In this subsection, we find a t.p.s. basis more accurate than this n.s. one, although with the same OOA.

The middle region already has $n_p^{\mathrm{sub}} = n_p$ and, thus, is not modified. In the left region, a convex combination $\alpha(x)$ is sought so that the basis functions are

$$(1 - \alpha(x)) \, \boldsymbol{\mathcal{L}}(x; \boldsymbol{I}^{4\times 4}(0\!:\!2, :), \boldsymbol{x}_G^4(0\!:\!2)) + \alpha(x) \, \boldsymbol{\mathcal{L}}(x; \boldsymbol{I}^{4\times 4}, \boldsymbol{x}_G^4) \tag{8}$$
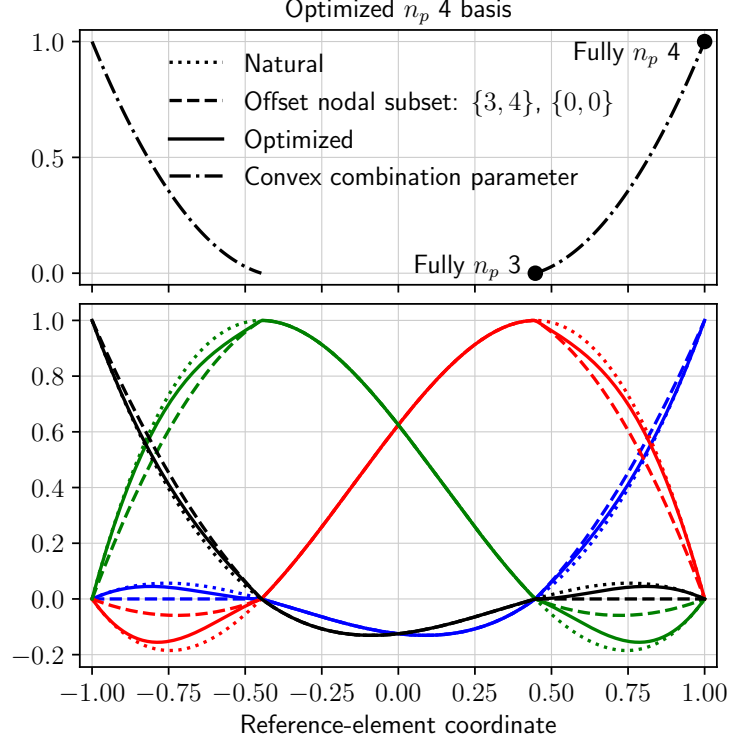
16

Figure 5: Optimized Islet GLL $n_p = 4$ basis (solid line) compared with the natural (dotted) and the best nodal subset (dashed) $n_p = 4$ bases. The top panel shows the convex combination parameter value as a function of the reference coordinate that is used to combine the natural and best nodal subset bases to form the optimized basis.

for $x \in [\boldsymbol{x}_G^4(0), \boldsymbol{x}_G^4(1)]$. The right region is symmetric to the left, as usual.

We set $\alpha(x)$, $x \in [\boldsymbol{x}_G^4(0), \boldsymbol{x}_G^4(1)]$, to be a quadratic polynomial having values $\boldsymbol{c} \in [0, 1]^3$ at left, middle, and right points of the region. Then, as usual, we use the stability criterion and the heuristics described in Sect. 3.5 to search for the optimal $\boldsymbol{c}$. To match the functional form in (8), for the left region, the accuracy heuristic is modified to use $\bar{e}(x) \equiv (1 - \alpha(x)) e(x; \mathcal{M}_4^3) + \alpha(x) e(x; \mathcal{M}_4^4)$, where $\mathcal{M}_4^4$ is the standard GLL 4-basis, and similarly for the right region. Results are insensitive to small perturbations to $\boldsymbol{c}$, so the final numbers are rounded to $\boldsymbol{c} = (1, 0.306, 0)$.

Figure 5 illustrates this basis. The top panel shows $\alpha(x)$ in the left region and its mirror image in the right region. The bottom panel shows the natural (dotted), o.n.s. (dashed), and optimized (solid) basis functions.

Future work could construct similar optimized bases for $n_p > 4$.

3.8 Natural bases

The natural basis over support points $\boldsymbol{x}^{n_p}$ that are arbitrary up to the constraints $\boldsymbol{x}^{n_p}(0) = -1$, symmetry, $\boldsymbol{x}^{n_p}(i) > \boldsymbol{x}^{n_p}(i-1)$, and all-positive nodal weights, is an important basis subset to consider. It contains all bases providing polynomial interpolation over a whole element, a continuous global interpolant, and the maximum OOA possible for a given value of n_p, again subject to all-positive nodal weights. For $n_p = 2$ and 3, the unique $\boldsymbol{x}^{n_p}$ satisfying these constraints give
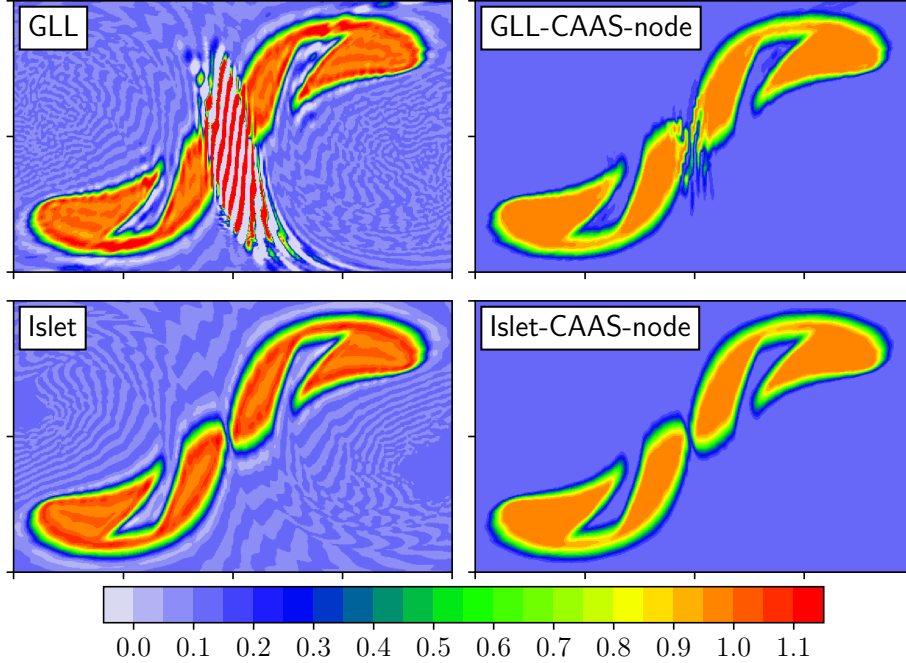
Figure 6: Images for unstable (top) and stable (bottom) ISL transport. The problem is nondivergent flow with the slotted cylinders IC, with $n_e = 20$, $n_p^{\mathrm{v}} = n_p^{\mathrm{t}} = 6$, $n_{\mathrm{step}} = 120$. The snapshot is at the end of day 11 of the first cycle; the images are zoomed to just the region containing the slotted cylinders. The color range is [-0.05, 1.15], which clips the top-left image's range of [-30.4, 32.7]. The bases are GLL (top) and Islet (bottom), without (left) and with (right) property preservation.

t.p.s. bases. (Hence our focus in this article on $n_p \geq 4$.) We searched over $\boldsymbol{x}^{n_p}$, subject to these constraints, for $4 \leq n_p \leq 12$ and found that only $n_p = 4$ has a t.p.s. basis in this subset. However, this $n_p = 4$ basis has a ratio of largest nodal weight to smallest of approximately 449, compared with 5 for both the GLL and Islet bases for $n_p = 4$. Thus, we conclude that the natural basis, regardless of support points $\boldsymbol{x}^{n_p}$, is not viable for high-order, element-based ISL methods.

4   Numerical and computational performance results

This section describes stability, accuracy, and computational performance results. These require a variety of configurations. The objective of the stability and accuracy experiments is to characterize the Islet bases in isolation from other parts of a production-configured transport method. Thus, one configuration is what we shall refer to as the *simple ISL method*. The simple ISL method is a direct implementation of (4), with no property preservation or remap between grids. Velocity data are provided at the tracer grid's points: i.e., $n_p^{\mathrm{v}} = n_p^{\mathrm{t}}$. A second configuration is the *Islet method*, which provides operators for property preservation, adding physics source terms, and coupling among separate dynamics, physics, and tracer grids; generally, $n_p^{\mathrm{t}} > n_p^{\mathrm{v}} = 4$. The companion article BBG22 documents numerical results for the Islet method. The final configuration is the *current E3SM configuration*, which we use for the computational performance measurements.

Two references provide the validation tests we use in this article, and we compare results against multiple references. All simulations are two-dimensional transport on the sphere. The
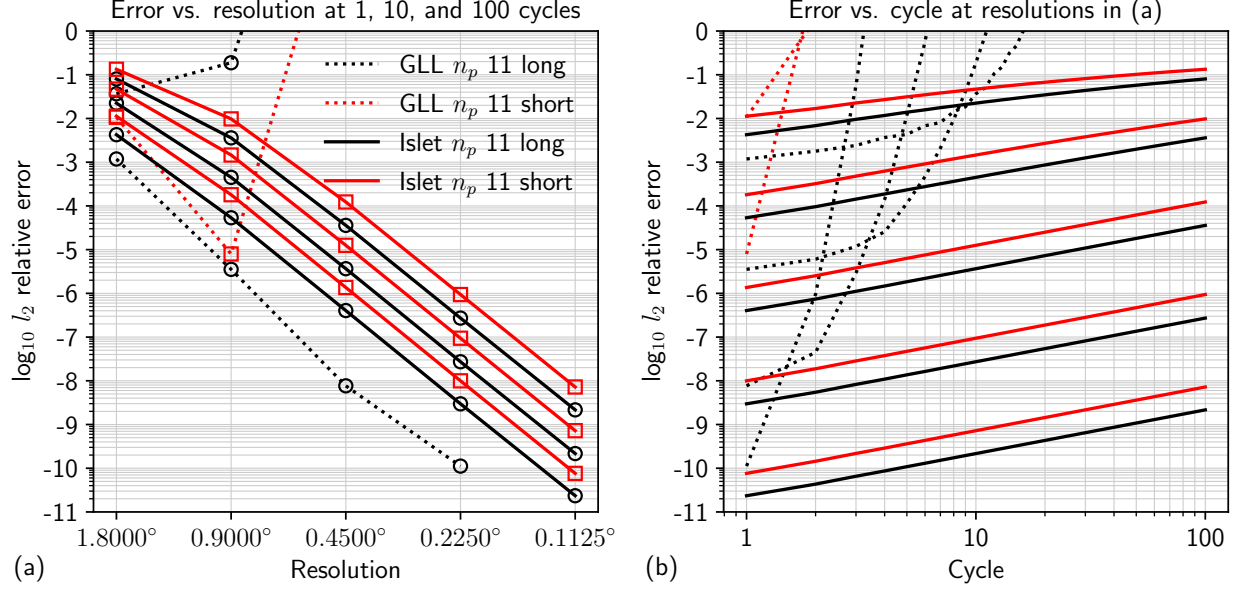
Figure 7: Results of long stability tests. (a) Error vs. resolution for long (30 time steps per cycle at 1.8° resolution; linear growth with refinement) and short (150 time steps per cycle at 1.8°) time steps, GLL and Islet $n_p = 11$ bases, and 1, 10, and 100 cycles of the nondivergent flow problem. (b) Error vs. cycle for the same simulations shown in (a).

first reference, Lauritzen et al. [2012], subsequently TS12, describes the *nondivergent* and *divergent* test flow fields; the C^∞ *Gaussian hills*, the C^1 *cosine bells*, the C^2 *correlated cosine bells*, and the discontinuous *slotted cylinders* initial conditions (ICs); the *mixing* and *filament* diagnostics; and the grid resolutions used for particular validation tests. In this article, we use the nondivergent flow field but not the divergent; see BBG22 for results using the divergent flow field. The flow field is defined for $T = 12$ days, but it can be run for multiple cycles of 12 days each. At times that are a multiple of T, the exact solution is the same as the initial condition. All four ICs place two circular shapes at two points along the equator. Results for a large set of methods are reported in Lauritzen et al. [2014], subsequently TR14. The second reference, Nair and Jablonowski [2008], describes the *moving vortices* flow field and the associated initial condition. Our implementation of this test problem follows the formulas in Bosler et al. [2017, Sect. 3].

It is standard when applying a method to these validation tests to integrate in time with sufficient accuracy to make the spatial error dominate. We use an adaptive Runge-Kutta method [Dormand and Prince, 1980, Shampine and Reichelt, 1997] with a tight tolerance (10^{-8}) to integrate the semi-Lagrangian departure-point trajectories. In some tests, we refer to the Islet method's property-preservation and p-refinement configuration settings to provide a comprehensive view of results. The precise details related to these settings are numerous and unrelated to the focus of this article; thus, we omit description of these settings here. See BBG22 Sect. 4 for details.

We use the equiangular cubed-sphere element grid. Each face of the cube has $n_e \times n_e$ elements. Figure 1 shows an example of this type of grid. In a practical application of the Islet method, the rest of the simulation determines n_e and the physics time step, and n_p^t is a free parameter. Higher values of n_p^t enrich the solution relative to what the dynamics grid provides. Thus, in BBG22, n_e alone determines the resolution. In contrast, because our primary focus here is the simple ISL

method, we fix n_p and then choose n_e so that $n_e(n_p-1)$ matches a desired resolution. The resolution is computed as a function of n_e and n_p as follows:

$$\frac{360°}{4 \text{ cube faces}} \cdot \frac{1 \text{ cube face}}{n_e \text{ elements}} \cdot \frac{1 \text{ element}}{(n_p - 1) \text{ intervals}} = \frac{90}{n_e \ (n_p - 1)} \frac{\text{degrees}}{\text{interval}}.$$

This measure of resolution trades between n_e and n_p. For example, the pairs ($n_p = 4$, $n_e = 20$) and ($n_p = 13$, $n_e = 5$) have the same resolution.

The test flows have a background uniform rotation with axis through the poles and a period equal to the 12-day cycle. If the rotation axis intersects an element corner, a mildly unstable mode can grow over the course of multiple cycles. This is true even for the $n_p = 3$ basis. In practice, this mode is not excited because in real simulations, a rotational flow center does not stay fixed to a grid point for hundreds of time steps. To put the grid in a general orientation, we rotate it to a general position with respect to the flow's background rotation axis, with results insensitive to the particular grid orientation. This grid orientation is the same for all validation tests in this section, except the moving vortices case as we explain later.

Two time steps are used, the *short* and *long* steps. These correspond to standard time step selections in TR14 for semi-Lagrangian methods, e.g., CSLAM Lauritzen et al. [2010]. In this article, the number of time steps per cycle of a problem is usually

$$n_{\text{step}} = 2fn_e(n_p - 1), \tag{9}$$

where $f = 1$ for the long step and $f = 5$ for the short step. In BBG22, time step selection is different because both resolution and time step are a function of n_e alone. We note when time step selection differs from (9). For a semi-Lagrangian method with very accurate trajectory time integration, a shorter time step generally yields less accuracy because the data are remapped more frequently per unit of time. Thus, given the freedom, an SL method will take as long a time step as possible. In practice, constraints from the application limit the time step. The strictest constraint is the physics time step, at which coupling between the dynamics and physics parameterizations occurs. Comparing with a typical production application's time steps, the short time step is shorter than the physics time step, while the long time step is approximately the same as or slightly longer than the physics time step.

Norm-wise relative errors are computed using the integral operator associated with the natural GLL basis.

4.1 Stability

We start by examining stability. Figure 6 shows snapshots for four simulations using $n_p = 6$ from the end of day 11. The top row are for the natural GLL basis; the bottom, the Islet basis. The left column shows the simulations with no property preservation. The right column uses the Islet method's property-preservation method; see BBG22 Sects. 2.2 and 3.1 for details. In the top row, the GLL basis makes the simulation unstable. Without a limiter, the instability quickly grows to a magnitude approximately 30 times the initial value; the color range is clipped for clarity. With a limiter, the instability in the linear advection operator does not develop as quickly, but it is still readily apparent. In the bottom row, the Islet basis yields a stabilized ISL method; the nonlinear property preservation step now just controls mass conservation and extrema, as intended.

Next, we run a large set of stability tests to compare the natural and Islet bases. The simple ISL method is used, thereby testing the Islet method's linear advection operator in isolation. The set
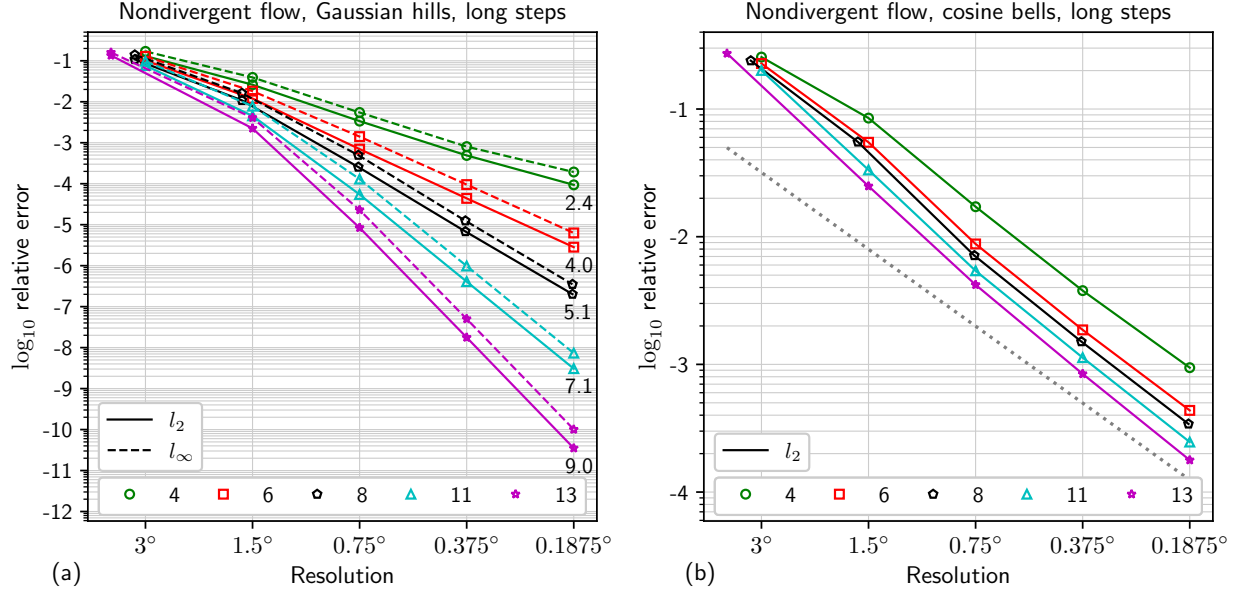
Figure 8: (a) Empirical verification of the order of accuracy of the Islet GLL bases. Each number at the right side of the plot is empirical OOA computed using the final two points of the l_∞ curve. (b) Cosine bells accuracy diagnostic. The dotted line is the OOA-2 reference.

is constructed as follows: $n_p^v = n_p^t$ with values 4 to 13, long ($n_{step} = 6n_e$) and short ($n_{step} = 30n_e$) time steps, and five resolutions. Each simulation runs 100 cycles of the nondivergent flow problem.

Figure 7 shows results for the $n_p = 11$ natural and Islet bases and the Gaussian hills IC. Figure 7(a) plots the l_2 relative errors at the ends of cycles 1, 10, and 100 as a function of resolution. As the cycle number increases, the corresponding curve shifts upward. For the Islet basis, the slopes of the curves are consistent with the expected OOA 7 from Table 1; the curve for the short time step at 100 cycles shows an empirical OOA of 7.02, using the two most refined grids. For the GLL basis, instability makes l_2 errors grow quickly, even in the first cycle. Figure 7(b) plots these same data in a different format, showing the growth in error as a function of flow cycle. As we expect for a stable simulation, at high resolution the errors for the Islet basis grow linearly with cycle number, with a slope of 1. The errors for the GLL basis diverge with increasing cycle count.

Similar results hold for all other values of n_p in Table 1, as well as other ICs and problem configurations. We show $n_p = 11$ and this particular problem configuration because these settings best show all the qualitative aspects of this test. At higher values of n_p, accuracy saturates at higher resolutions due to finite-precision arithmetic. At lower values of n_p, the 100-cycle, low-resolution, short-time-step l_2 errors are larger, increasingly obscuring the trends in the curves with decreasing n_p. Similarly, only the Gaussian hills IC is sufficiently smooth to reveal the full OOA.

4.2 Accuracy

Figure 8(a) empirically verifies the order of accuracy of the Islet bases using the simple ISL method. The test uses the nondivergent flow, Gaussian hills IC, and long time steps. The legend shows the marker type for each value of n_p. We show results for a subset of the bases in this and other figures to reduce clutter; results for those bases not shown are similar. Relative errors are reported in the
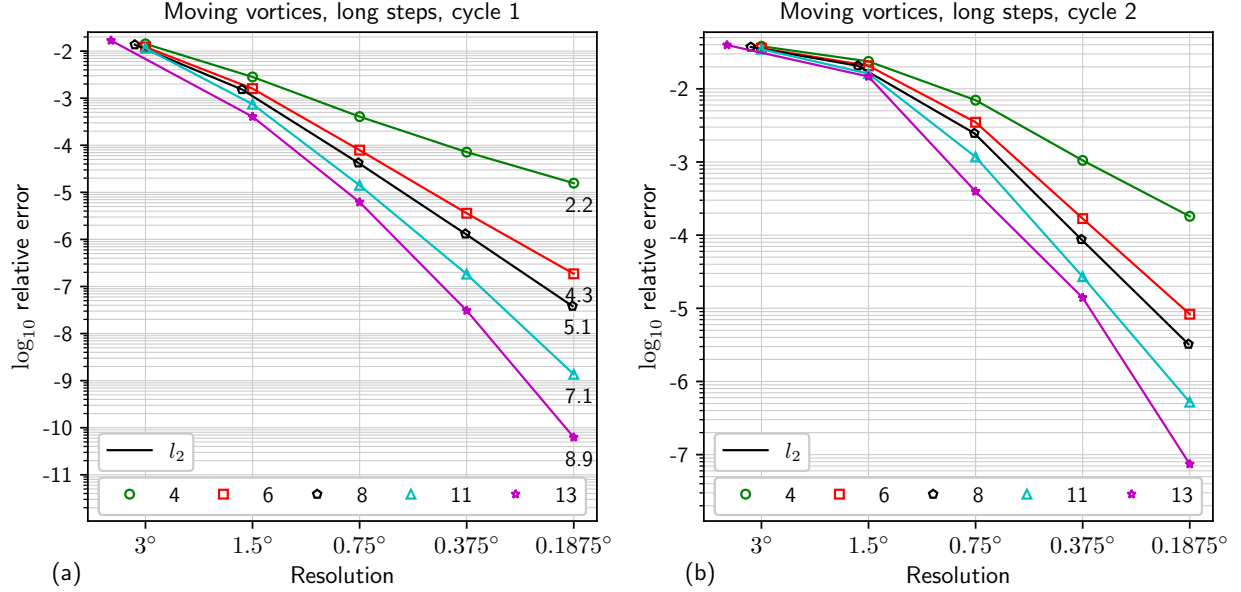
21

Figure 9: Error vs. resolution for the moving vortices problem at the end of (a) cycle 1 and (b) cycle 2. In (a), the empirical OOA is shown.

l_2 (solid lines) and l_∞ (dashed lines) norms. The numbers on the right side of the plot are empirical OOA calculated using the final two points of the l_∞-norm curves; the values for the l_2-norm curves are about the same. Each empirical OOA is at least as large as, but not much larger than, the expected OOA listed in Table 1. Comparing with the left column of Figs. 1, 2 in TR14, we see that the $n_p = 13$ accuracy is greater than that of every method in TR14.

Figure 8(b) shows results for the same configuration except with the cosine bells IC. Because the IC is C^1, at most OOA 2 is possible. Despite this limit on OOA, accuracy increases with increasing n_p at every resolution. At any resolution at least $1.5°$, the $n_p = 13$ case is roughly four times or more accurate than the $n_p = 4$ case.

Because the primary motivation for EBTPI methods compared with meshless ones is efficiency, we carry out a detailed, although necessarily imprecise, comparison of our simple ISL method with the local-RBF (simple) ISL methods of Shankar and Wright [2018]. In the local-RBF methods, the work at each interpolation step is $O(Nn(n+n_t))$, where N is the number of nodes, n is the support size, and n_t is the number of tracers. A difference between our numerical experiments and theirs is the time step. They fix the time step at 50 per cycle independently of resolution. In contrast, we increase ours linearly with resolution, following usage (due to the physics time step) in a production model and common practice in, e.g., TR14. At their highest resolution, we use 240 time steps. First, our Fig. 8(a) can be compared with their Fig. 11. At their highest resolution, $N = 304^2$, the RBF-PU method with $n = 84$ has l_2 error norm 5e-7. Our $n_p = 13$ method matches that accuracy at a little coarser than $0.53°$ resolution, or $\bar{N} \approx 173015$. The factor difference in computation is very roughly $(Nn(n+n_t))/(\bar{N}n_p^2(1+n_t))$, although of course constants and details of the problem setup make this estimate extremely imprecise. Nonetheless, as examples, for $n_t = 1$, the estimate suggests our method is 11.3 times more efficient; for $n_t = 10$, 2.27 times; for $n_t = 100$, 0.48, i.e., half as efficient. A similar comparison can be made between our Fig. 8(b) and their Fig. 7. Here, lower-order methods tend to be more efficient and configuration details matter a lot more because
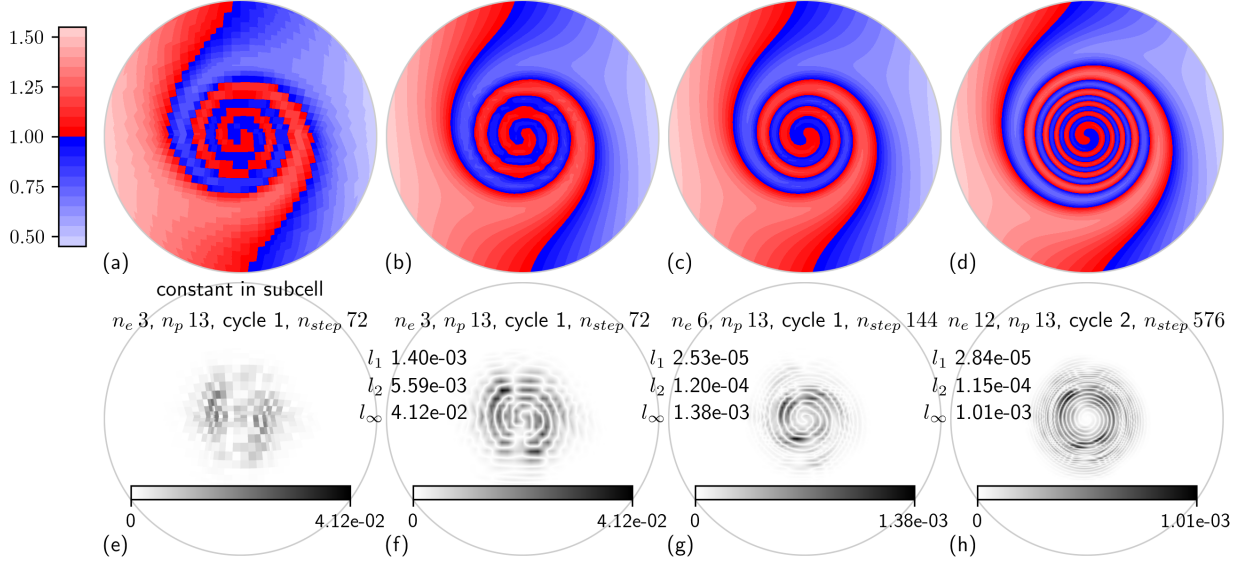
22

Figure 10: (a-d) Images of one of the two vortices in the moving vortices problem at the end of (a-c) cycle 1 and (d) cycle 2. Parameters in each column report the test configuration. The first two columns (a, b, e, f) show results for the same problem. In (a), values in each subcell are the average of values at the subcell's supporting nodes, which is constant in each subcell. In (b-d), bilinear interpolation is computed in each subcell. (e-h) Corresponding relative errors. Relative error norms are reported in text; images show pointwise error divided by the l_∞-norm of the reference solution.

the IC is only C^1. At $N = 304^2$, the local RBF methods $n = 31$ give approximately 1e-3 accuracy. Because configuration details matter more in this case, we determined n_e with 50 time steps and $n_p = 6$ such that our method gives the same accuracy: $n_e = 38$, with $\bar{N} = 6(n_e(n_p - 1))^2 + 2$. The efficiency factors are then 5.9 for 1 tracer, 1.37 for 10, and 0.48 for 100. Finally, we repeat this more precise approach for the Gaussian hills IC to illustrate the effects of configuration details like the number of time steps. For 50 time steps, we find $n_p = 13$ and $n_e = 11$ give 5.51e-7 accuracy, about the same as the more accurate of the local RBF methods at $N = 304^2$. This search gives revised efficiency factors of 18.7 for 1 tracer, up from 11.3; 3.75 for 10; and 0.80 for 100. Again, we emphasize these factors are extremely imprecise. We also omit an analysis of communication costs, which scale with n_t and generally favor element-based methods over overlapping-stencil methods for any value of n_t. We conclude that an element-based tensor-product interpolation method is competitive with a contemporary meshless method and perhaps substantially more efficient when the number of tracers is small.

Similar comparison of the Gaussian hills case can be made with results for the direct method in Bosler et al. [2017, Fig. 6]. Because their method is limited to OOA a little under 4, the high-order Islet bases give greater accuracy for a given resolution. An efficiency comparison would require substantial analysis and is not attempted.

Figure 9 shows similar results for the moving vortices case, at the ends of cycles 1 and 2. The $n_p = 13$ curve is not quite fully converged to the expected OOA 9, but the other curves are. Figure 10 show images for three $n_p = 13$ resolutions of interest. The lowest-resolution simulation is comparable to that in Nair and Jablonowski [2008, Fig. 2] from an Eulerian discontinuous Galerkin spectral element method on the cubed sphere with $n_e = 5$, $n_p = 8$. We obtain very slightly greater
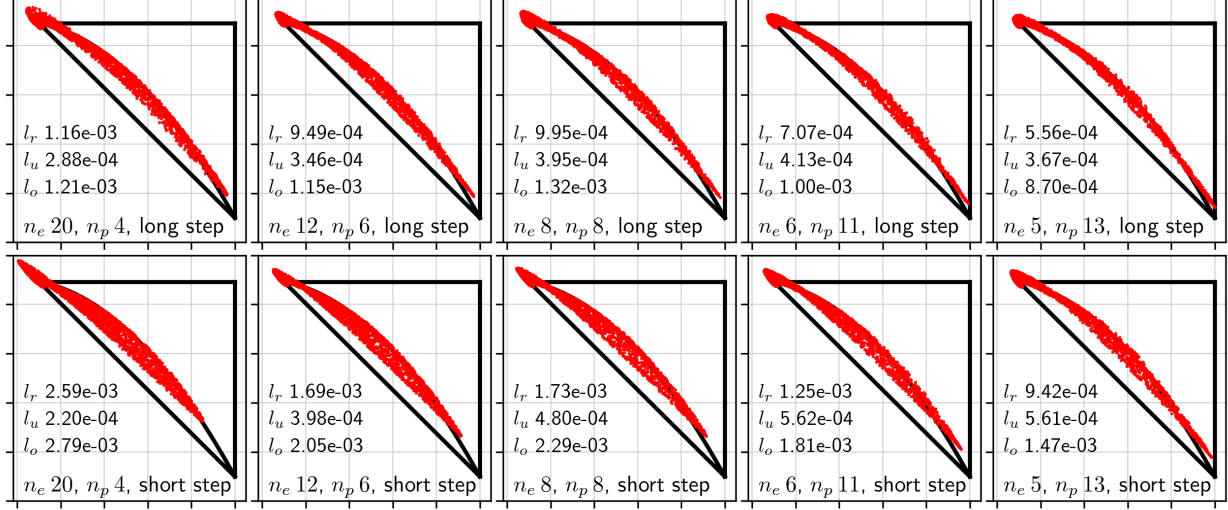
Figure 11: Mixing diagnostic, following TS12 Sect. 3.5. Configurations details are reported at the bottom of each diagram.

accuracy with $n_e = 3$, $n_p = 13$, but this would change if we were to use somewhat shorter time steps. As in Nair and Jablonowski [2008, Fig. 2], and departing from the other tests, we orient the grid so that the vortices move over four cube corners, as cube corners are sometimes considered challenging. Figure 9(a) plots the data with constant values in each element to show the underlying grid. We do not see any grid-related artifacts other than those due to the grid's coarseness. The middle-resolution simulation can be compared with the right column of Bosler et al. [2017, Fig. 3]. The highest-resolution simulation can be compared with the right column of Taylor and Nave [2023, Fig. 4], although we use approximately 16 times fewer degrees of freedom.

4.3 Other diagnostics

The mixing diagnostic described in TS12 and TR14 measures the persistence of correlation in two nonlinearly correlated ICs, the C^1 cosine bells and the C^2 correlated cosine bells, using the same flow field and configurations as in Fig. 8. It is measured at the simulation midpoint. In Figs. 11 and 12, each point represents a pair of values of the two tracers at a point in space. The perfect solution is essentially like in the top-right corner of Fig. 12. Loss of correlation spreads these points away from the curve. Figure 11 shows results for $1.5°$ resolution and 12 for $0.75°$. The top row is for the long time step and the bottom for the short. We expect and observe that increasing the time step and, separately, increasing the order of the basis, increase persistence of correlation. The labeled errors in each panel follow the diagnostic's protocol. They are useful when comparing results with TR14 Figs. 11–14. Figure 11 corresponds to the left of the four columns and Fig. 12 to the third column. The simple ISL method is competitive with those in TR14, but not particularly great, because high order is only marginally useful for doing well on this diagnostic.

The filament diagnostic described in TS12 and TR14 measures the persistence of filamentary structure. It is measured at the same time as the mixing diagnostic and using the cosine bells IC. At a value τ on the x axis of a panel, the y axis is the area of tracer that is at least τ, relative to this value in the IC and expressed as a percent. Thus, perfect persistence corresponds to a horizontal
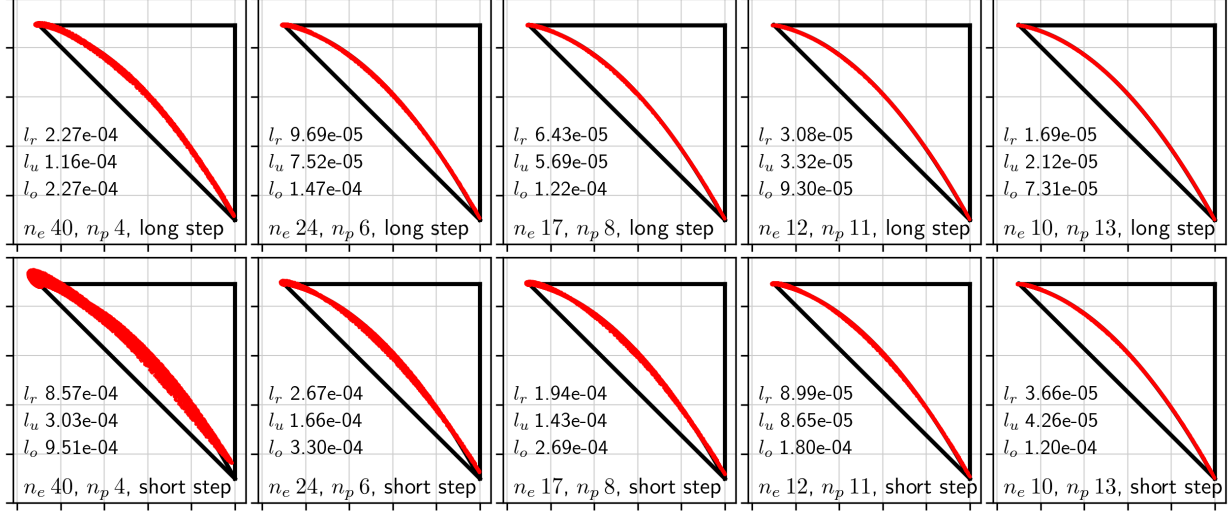
24

Figure 12: As Fig. 11 but for higher resolution.

line at 100. Figure 13 shows results for the simple ISL method. The top row, 1.5° resolution, can be compared with the unlimited subset of the left column of Fig. 5 in TR14, and the bottom row, 0.75° resolution, with the right column, again the unlimited subset. The results improve with increasing order. The 1.5°, $n_p = 13$ results are competitive with the best methods in TR14 because, despite the IC being the C^1 cosine bells, the lower dissipation in high-order methods is useful for this diagnostic. For this same reason, as order increases, the difference in results between the two time steps decreases.

4.4 The performance of the accuracy heuristic

To select Islet bases for each value of n_p^{t}, recall that we use the accuracy heuristics $a_{1,2,\infty}$ to choose accurate bases and the perturbed uniform grid metric to filter out bases that might be unstable in practice. The accuracy heuristics are functions of the basis data only, not any specific problem. To assess how well the accuracy heuristic predicts actual performance, Fig. 14 plots accuracy on a validation problem vs. the value of the a_2 heuristic for a large number of t.p.s. bases. Each point in a scatter plot corresponds to a basis. The y-axis is accuracy of bases when used to simulate the nondivergent flow with $n_e = 20$, in two different cases: Gaussian hills with no property preservation (Fig. 14(a)) and cosine bells with property preservation (Fig. 14(b)). The settings for the second case are those used in practice for the full Islet method, and both cases set $n_p^{\mathrm{v}} = 4$. The x-axis is the value of a_2 for the basis. The marker pattern corresponds to n_p^{t}. For each n_p^{t}, a red circle outlines the basis that appears in Table 1. We observe that the a_2 accuracy heuristic correlates well with the relative accuracy on the validation problems. As a result, the final set of selected bases provide the most accurate or nearly the most accurate simulation results among all the viable bases.

4.5 Computational performance

The E3SM Atmosphere Model's (EAM) nonhydrostatic dynamical core, HOMME [Dennis et al., 2005, 2012, Taylor et al., 2020], has been ported to C++ to run on GPU-based supercomputers, and
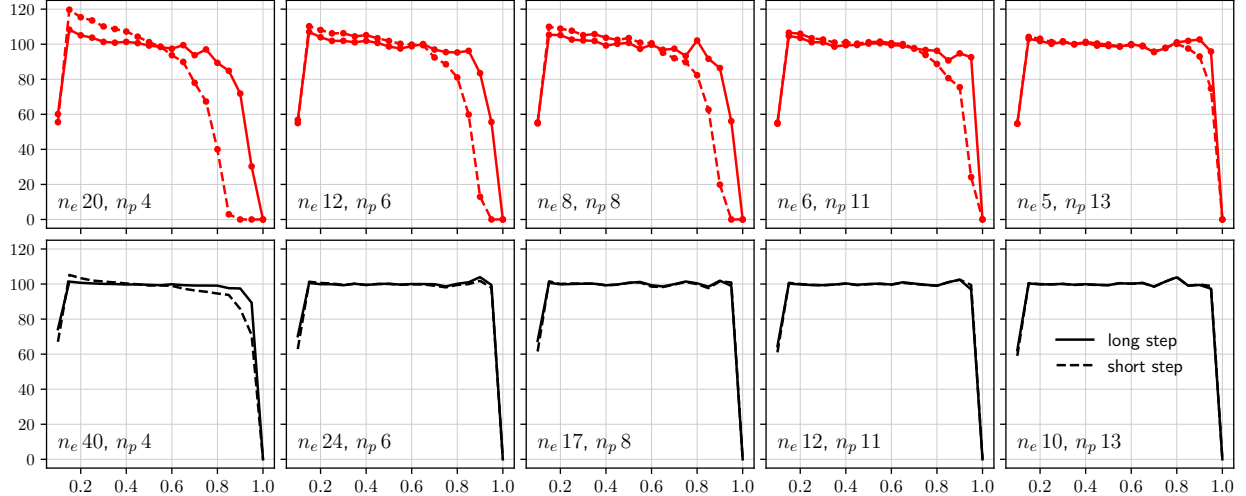
25

Figure 13: Filament diagnostic, following TS12 Sect. 3.3.

performance results are reported in Bertagna et al. [2020] for the NGGPS benchmark [Whitaker, 2016, Michalakes et al., 2016] run on the Oak Ridge Leadership Computing Facility's Summit supercomputer. At the time that article was written, only the original Eulerian tracer transport solver was available in the C++ dynamical core. In related work, a quasiuniform 3.25km convection-permitting configuration of EAM, called SCREAM, has been developed [Caldwell et al., 2021]; this work uses the Islet transport method with $n_p^v = n_p^t = 4$. Other than this difference, Bertagna et al. [2020] and Caldwell et al. [2021] use the same dynamical core configurations. We omit description of these configuration details because they are not relevant to this article; see the cited articles for details.

We re-ran the benchmark in Bertagna et al. [2020] and show the results in Fig. 15. The x-axis is number of Summit NVIDIA V100 GPUs used in a run. Summit has just over 27,600 V100 GPUs. The y-axis is dynamical core throughput reported in simulated years per wallclock day (SYPD). SYPD text labels annotate a subset of the data points.

First, we established that we could roughly reproduce the results in Bertagna et al. [2020]. The black dotted line shows the data we obtained using Eulerian transport; compare this curve with the data from Bertagna et al. [2020], plotted as the black solid line with circular markers. Our results are a little less than 10% slower, which is acceptable given different compiler and MPI versions, different job profiles running at the same time, and a slightly different code base. We then switched to using SL transport and obtained the solid red line with square markers. These simulations have ten tracers. We repeated them with 40 tracers, the number used in E3SMv1 and v2 models; the corresponding curves are dashed. Comparing the new SL data with the data in Bertagna et al. [2020], SL transport speeds up the nonhydrostatic dynamical core by a factor of over 1.4 when 10 tracers are transported. Bertagna et al. [2020] do not provide data for the case of 40 tracers. Thus, for the case of 40 tracers, we use just our data and find that the dynamical core speedup factor is 2.8.
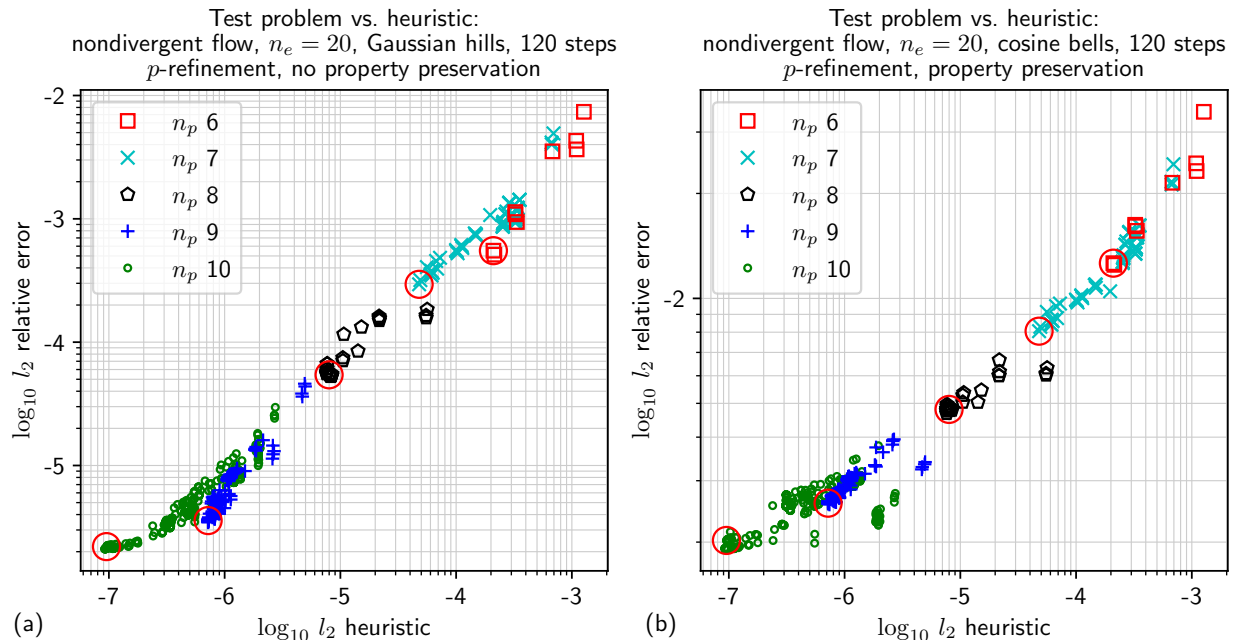
Figure 14: l_2 errors for the nondivergent flow problem using basis $\mathcal{M}_{n_p}$ vs. $a_2(\mathcal{M}_{n_p})$, for a large number of t.p.s. bases and $n_p = 6$ to 10. The legends list the marker type for each n_p. Large red circles outline the points corresponding to the bases in Table 1. (a) With the Gaussian hills IC and no property preservation. (b) With the cosine bells IC and property preservation.

5   Conclusions

Element-based, tensor-product interpolation semi-Lagrangian methods are a compelling class of ISL methods for tracer transport because they are highly efficient, easily extend to arbitrarily high order, and work on nonuniform and unstructured element grids. Efficient property-preservation methods can optionally provide shape preservation and mass conservation [Bradley et al., 2019]. They can couple particularly efficiently to other model components that share the same element grid [Bradley et al., 2022]. Their primary drawback is the challenge of designing stable methods. A secondary drawback is the challenge of proving a method to be stable in some useful sense, a challenge shared with all but the most structured, simplest interpolants. This challenge is exacerbated by the fact that even the simplest classical ISL methods are not unconditionally stable on all problems.

We defined *sparse instability* as a stability criterion to select usable ISL methods, then used this definition to construct quantitative criteria to assess a potential basis. Next, we constructed a large set of potential EBTPI bases and an efficient search procedure to find the best basis, according to these criteria, for each value of n_p from 4 to 13.

Numerical results show that the resulting methods are competitive with a number of contemporary methods. The Dept. of Energy's Energy Exascale Earth System Model version 2 uses the lowest-order variant of the Islet method and is substantially faster than version 1. The ultra-high-resolution configuration [Caldwell et al., 2021] of the E3SM Atmosphere Model uses the method, as well. The implementation in the latter shows excellent computational performance in a strong-scaling study that uses most of the Summit supercomputer.
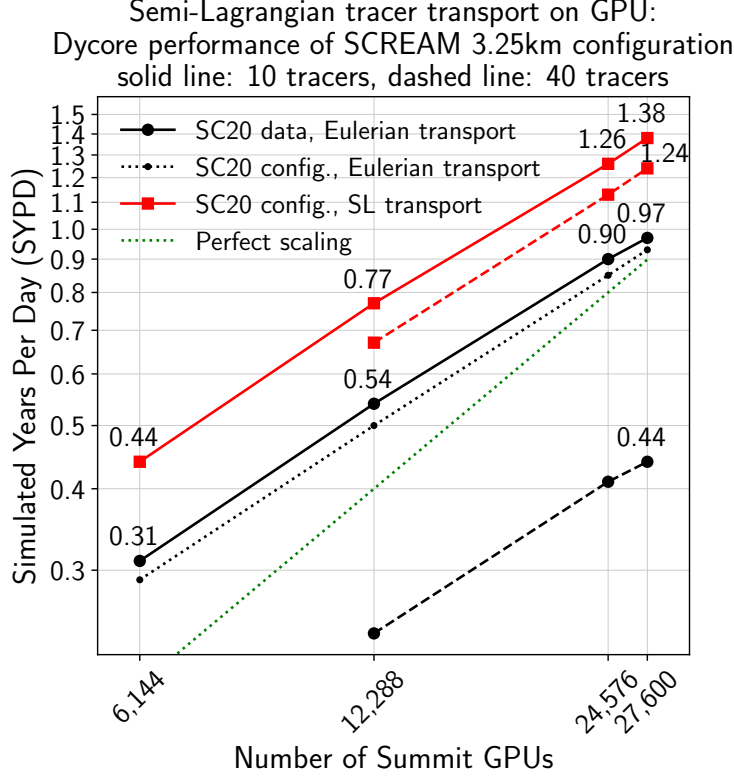
Figure 15: Performance comparison of SL transport with $n_p^{\mathrm{v}} = n_p^{\mathrm{t}} = 4$ vs. Eulerian transport in the E3SM Atmosphere Model's dynamical core on the Summit supercomputer. The x-axis is the number of NVIDIA V100 GPUs on Summit used in a run; the y-axis is the dynamical core (dycore) throughput reported in simulated years per wallclock day (SYPD). The black curves are for Eulerian transport; the red, for SL. Dashed lines are for 40 tracers; solid and the dotted black line, for 10. A number above a data point reports the y-value of that point.

The core of the Islet method is a set of bases found by computational search. While we developed methods to make the search procedure efficient, it still requires substantial computational work. Thus, we restricted the space of bases to search to a finite subset of the set of all potential bases. Future work could search larger sets and, in particular, include element reference grid point locations in the search. In addition, the basis optimization in Sect. 3.7.2 for $n_p = 4$ could be generalized to a procedure applicable to any value of n_p. Finally, the stability criterion used as one filter in the search could be enhanced to check for uniform power boundedness rather than just the maximum eigenvalue magnitude.

A   The maximum eigenvalue magnitude

This appendix explains how to compute $\tilde{\lambda}_{\max}(\boldsymbol{A})$ for $\boldsymbol{A}$ defined in (6). Let an eigenvector of $\boldsymbol{A}$ be $\boldsymbol{V}$, and write $\boldsymbol{V} \equiv \begin{pmatrix} \boldsymbol{v}_0^T & \cdots & \boldsymbol{v}_{N-1}^T \end{pmatrix}^T$, where d-vector $\boldsymbol{v}_k$ is d-length block row k of $\boldsymbol{V}$, $k \in \{0, \ldots, N-1\}$. We suppose $\boldsymbol{v}_k = \mu^k \boldsymbol{q}$ with $\mu \equiv e^{i2\pi j/N}$, $j \in \{0, \ldots, N-1\}$, where i is the imaginary unit and $\boldsymbol{q}$ is an unknown d-vector. Thus, we suppose an eigenvector is periodic at the block level, and each block is the same up to a unit-magnitude complex scalar. This form for $\boldsymbol{V}$

makes sense because each block row of $\boldsymbol{A}$ is identical except for a column shift; thus, μ should have unit magnitude. Because $\boldsymbol{A}$ is a uniform-element discretization of uniform flow on a periodic domain, μ should be an Nth root of unity to make $\boldsymbol{V}_j$ periodic.

Let $r \equiv (r' + \rho d) \mod (Nd)$ for $\rho \in \{0, \ldots, N-1\}$. We need to establish some indices and permutations. Let $\bar{r} \equiv \lfloor r/d \rfloor$, $\hat{r} \equiv r \mod d$, and

$$\boldsymbol{S}_r^{\mu} \equiv \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I}^{(d-\hat{r}) \times (d-\hat{r})} \\ \mu \boldsymbol{I}^{\hat{r} \times \hat{r}} & \boldsymbol{0} \end{pmatrix}.$$

Consider rows $(r + (0 : d-1)) \mod (Nd)$ of the eigenvalue problem $\boldsymbol{A}\boldsymbol{V} = \lambda\boldsymbol{V}$, and assume $\boldsymbol{v}_k = \mu^k \boldsymbol{q}$ as described above. The left hand side can be written

$$\boldsymbol{A}((r + (0:d-1)) \mod (Nd), :)\boldsymbol{V} = \boldsymbol{B}\boldsymbol{V}((\rho d + (0:d)) \mod (Nd)) = \boldsymbol{B}\mu^{\rho} \begin{pmatrix} \boldsymbol{I}^{d \times d} \\ \mu \boldsymbol{e}_0^T \end{pmatrix} \boldsymbol{q}. \tag{10}$$

The right hand side can be written

$$\lambda\boldsymbol{V}((r + (0:d-1)) \mod (Nd)) = \lambda\mu^{\bar{r}} \begin{pmatrix} \boldsymbol{q}(\hat{r} : d-1) \\ \mu\boldsymbol{q}(0 : \hat{r}-1) \end{pmatrix} = \lambda\mu^{\bar{r}} \boldsymbol{S}_r^{\mu} \boldsymbol{q}.$$

Together, these give the $d \times d$ eigenvalue problem

$$\mu^{\rho - \bar{r}} (\boldsymbol{S}_r^{\mu})^H \boldsymbol{B} \begin{pmatrix} \boldsymbol{I}^{d \times d} \\ \mu \boldsymbol{e}_0^T \end{pmatrix} \boldsymbol{q} = \lambda\boldsymbol{q}, \tag{11}$$

where H is the matrix conjugate transpose. The eigenvalues of $\boldsymbol{A}$ are the union of those obtained from solving (11) for each $\mu = e^{i2\pi j/N}$, $j = 0, \ldots, N-1$.

If every value of N is of interest, then $\mu = e^{i\theta}$, $\theta \in [0, 2\pi)$. Thus, to compute the maximum eigenvalue of $\boldsymbol{A}$ independently of N, we solve the problem

$$\lambda_{\max}(\boldsymbol{B}) \equiv \max_{\substack{\theta \in [0, 2\pi) \\ i \in \{0, \ldots, d-1\}}} |\lambda_i|$$

$$\text{subject to } (\boldsymbol{S}_r^{e^{i\theta}})^H \boldsymbol{B} \begin{pmatrix} \boldsymbol{I}^{d \times d} \\ e^{i\theta} \boldsymbol{e}_0^T \end{pmatrix} \boldsymbol{Q} = \boldsymbol{\Lambda}\boldsymbol{Q},$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues λ_i, $i \in \{0, \ldots, d-1\}$. $\mu^{\rho - \bar{r}}$ is omitted from the eigenvalue problem because it does not affect an eigenvalue's magnitude.

In practice, we must limit the search to a large, finite subset of $[0, 2\pi)$ and compute the eigenvalues numerically, giving the approximation $\tilde{\lambda}_{\max}(\boldsymbol{B}) \approx \lambda_{\max}(\boldsymbol{B})$. This problem requires solving many separate $d \times d$ eigenvalue problems instead of many $(Nd) \times (Nd)$ problems. Thus, first, one can program a work- and parallel-efficient method to check a particular discretization for semi-analytical satisfaction of the necessary condition; second, we avoid the superlinear growth in the cost of the eigenvalue computation with N.

B   Interpolants

In this appendix, we construct expressions for the most general set of potential bases, stating the five constraints listed in Sect. 3.2 more precisely.

1. Given a departure point x, the interpolant is a linear operator in the values $\boldsymbol{y}^n$ at basis points $\boldsymbol{x}^n$.

2. Within each region, the interpolant is a polynomial.

Combining constraints 1 and 2, the interpolant has the form

$$f(\boldsymbol{y}^n; x) \equiv \mathcal{L}(x; \boldsymbol{M}^{n \times n}(x)\boldsymbol{y}^n, \boldsymbol{x}^n), \tag{12}$$

where $M(x)$ is constant in each region,

$$\boldsymbol{M}(x) = \sum_{r=0}^{d-1} \text{Hat}(x; \boldsymbol{x}^n(r), \boldsymbol{x}^n(r+1)) \ \boldsymbol{A}_r^{n \times n}, \tag{13}$$

and

$$\text{Hat}(x; a, b) \equiv \begin{cases} 1 & \text{if } a \leq x \leq b \\ 0 & \text{else.} \end{cases}$$

We call each matrix $\boldsymbol{A}_r^{n \times n}$ a *region operator*.

3. f must recover a degree-p polynomial with specified $p = s - 1$ and $2 \leq s \leq n$. This *order constraint* implies

$$\boldsymbol{A}_r^{n \times n} \boldsymbol{Y}^{n \times s} = \boldsymbol{Y}^{n \times s}$$

for each region r, where $\boldsymbol{Y}^{n \times s}$ provides a basis for degree-p polynomials, $\mathcal{L}(x; \boldsymbol{Y}^{n \times s}, \boldsymbol{x}^n)$. An example is

$$\boldsymbol{Y}^{n \times s} = \begin{pmatrix} \boldsymbol{I}^{s \times s} \\ \boldsymbol{B}^{(n-s) \times s} \end{pmatrix},$$

where $\boldsymbol{B}^{(n-s) \times s} \equiv \mathcal{L}(\boldsymbol{x}^n(s{:}n-1); \boldsymbol{I}^{s \times s}, \boldsymbol{x}^n(0{:}s-1))$.

4. f must interpolate nodes r, $r+1$. This *region interpolation constraint* implies

$$\boldsymbol{A}_r([r, r+1], {:}) = \begin{pmatrix} \boldsymbol{0}^{2 \times r} & \boldsymbol{I}^{2 \times 2} & \boldsymbol{0}^{2 \times (n-r-2)} \end{pmatrix}$$

for each region r.

5. Consider the n basis functions

$$\phi_i(x) \equiv f(\boldsymbol{e}_i; x).$$

These must have the symmetry

$$\phi_i(x) = \phi_{n-1-i}(-x).$$

This *symmetry constraint* implies, first, the points in $\boldsymbol{x}^n$ must be symmetric around $x = 0$; second,

$$\boldsymbol{A}_r = \boldsymbol{A}_{n-1-r}(n-1{:}-1{:}0, n-1{:}-1{:}0)$$

for each region r.

For a non-middle-region operator, $\boldsymbol{A}_r$ starts with n^2 degrees of freedom. The order constraint removes ns. The region interpolation constraint removes $2(n-s)$ more; the $-2s$ term is because $2s$ degrees of freedom are shared with the order constraint. This leaves $n^2 - ns - 2(n-s) = (n-2)(n-s)$ independent degrees of freedom. The symmetry constraint implies that a middle-region operator $\boldsymbol{A}_r$ has half of these: $(n-2)(n-s)/2$.

We can devise a basis $\mathcal{M}_{n_p}$ by selecting values for node points $\boldsymbol{x}^{n_p}$ and the region operators $\boldsymbol{A}_r$, subject to constraints 1–5. Subsets of bases are defined by adding additional constraints or construction rules. For example, for nodal subset bases, each $\boldsymbol{A}_r$ has rows of the identity matrix corresponding to nodes in $\mathcal{I}_r$, zero columns for nodes not in $\mathcal{I}_r$, and the remaining entries determined by the order constraint. In offset nodal subset bases, the rows of the identity matrix are, in addition, contiguous.

C The search procedure

The search procedure combines the stability criterion and the accuracy and instability heuristics to find one t.p.s. basis in the set of potential bases for each value of n_p.

For a given n_p, the procedure first enumerates all possible o.n.s. bases, grouped in sets by decreasing n_p^{submin}. All bases use GLL node points. Once it finds a value of n_p^{submin} for which there are t.p.s. bases and has tested all bases in this set, the procedure switches to enumerating all possible n.s. bases having this or larger value of n_p^{submin}.

In both phases, bases are filtered as follows. Three sets of uniformly spaced bins are constructed over a range of $\log_{10}(\tilde{\lambda}_{\max}^{\mathrm{PUG}} - 1)$ values, roughly 0 down to $\log_{10}$ of machine precision, one set of bins for each of $a_{1,2,\infty}$. Each bin starts empty and holds up to one basis. Consider a candidate basis. A sequence of analysis steps is performed. If a step fails, the analysis of the basis stops and the procedure moves to the next candidate. Each step is categorized as fast or slow. For efficiency, we delay slow steps as long as possible so that a failure in a fast step lets the search move to the next candidate basis as quickly as possible. The steps are, in sequence, as follows.

1. The basis's nodal weights, where a node's weight is the integral of its associated basis function, are computed. A non-positive weight is a failure because we want each basis function to integrate to a positive number. This step is fast.

2. The basis's $a_{1,2,\infty}$ values are computed. If, in each bin, no value is better than the best, the step fails. (Thus, if there is an empty bin, the step succeeds.) This step is fast.

3. The basis is tested for test-problem stability. As we described previously, many small eigenvalue problems are solved in parallel. If any has $\tilde{\lambda}_{\max} > 1 + \varepsilon$, the step terminates and fails. A speedup is to search Δx so that the range $(0, 1/2]$ is covered quickly, coarsely at first, and then filled in at a roughly uniform rate everywhere, and the same for θ. This enumeration of these sets encourages fast failures if a basis is not t.p.s. This step is slow if it succeeds but almost always is fast if it fails.

4. The largest $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ value that would permit acceptance of the candidate based on its $a_{1,2,\infty}$ values is computed using the current state of the bins. This step is fast and cannot fail.

5. $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ is computed, given the threshold from step 4. If a value is found that exceeds this threshold, this step terminates and fails. This step is usually slow.

6. For each a_i, $i \in \{1, 2, \infty\}$, the basis is associated with a bin, replacing any previous basis associated with the bin, if, first, the bin's upper $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ is at least as large as the value for the basis and, second, the basis's a_i value is better than the bin's current value. The bin's a_i value is then updated. Thus, a bin tracks the most accurate basis found up to the bin's upper $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ for each accuracy heuristic value. This step is fast and cannot fail.

The procedure ends when all candidate bases have been analyzed. The output is the set of t.p.s. bases in the bins at the procedure end and their values for each heuristic.

As the procedure runs, it can filter candidate bases more quickly because the bins are populated with increasingly accurate bases, leading to more frequent failures in step 2 and thus before any slow steps. In addition, by starting with o.n.s. bases, the bins get populated quickly with accurate bases, making the search through the much more numerous general n.s. bases faster. For this reason, future work that considers larger sets of potential bases should order the search as a sequence of supersets of the previous sets.

We have completed this procedure for bases through $n_p = 10$. For bases having $n_p > 10$, we have completed this procedure for o.n.s. bases but only incompletely for general n.s. bases.

After the procedure ends, we must still select a basis from the set of bases in the bins. We have not developed a strict rule to select the basis because there is a trade-off between $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ and accuracy. Instead, we follow a few guidelines to select the final one to recommend. First, we always choose a basis having the maximum n_p^{submin} we have found. Second, within that set, we prefer small $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ values. Within this preference, a factor of roughly three difference in $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ values does not matter, but a factor of roughly ten does. We then choose the most accurate basis within the tight cluster of $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ values we have selected, weighting a_2 more heavily than $a_{1,\infty}$. For $n_p \leq 10$, only $n_p = 6$ and 9 have significantly better non-offset than offset bases. In these two cases, $\tilde{\lambda}_{\max}^{\mathrm{PUG}}$ for the best o.n.s. basis is substantially higher than for the best non-offset one.

Code availability

Code and scripts for the algorithms and figures presented in this paper are available at https://github.com/E3SM-Project/COMPOSE/releases/tag/v1.1.4. See the directory `methods/islet/-readme.txt` for further instructions.

Acknowledgments

References

M. Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *J. Comput. Phys.*, 198(1):106–130, 2004.

A. M. Baptista. *Solution of advection-dominated transport by Eulerian-Lagrangian methods using the backwards method of characteristics.* PhD thesis, Massachusetts Institute of Technology, 1987.

L. Bertagna, O. Guba, M. Taylor, J. Foucar, J. Larkin, A. Bradley, S. Rajamanickam, and A. Salinger. A performance-portable nonhydrostatic atmospheric dycore for the Energy Exascale Earth System Model running at cloud-resolving resolutions. In *SC20: Int. Conf. High Perfor.*, pages 1304–1317. IEEE Computer Society, 2020.

Nicolas Besse and Michel Mehrenberger. Convergence of classes of high-order semi-Lagrangian schemes for the Vlasov-Poisson system. *Math. Comput.*, 77(261):93–123, 2008.

P. Bochev, S. Moe, K. Peterson, and D. Ridzal. A conservative, optimization-based semi-Lagrangian spectral element method for passive tracer transport. In B. Schrefler, E. Onate, and Papadrakakis M., editors, *Coupled Problems*, pages 23–34. CIMNE, 2015.

Peter A Bosler, James Kent, Robert Krasny, and Christiane Jablonowski. A Lagrangian particle method with remeshing for tracer transport on the sphere. *J. Comput. Phys.*, 340:639–654, 2017.

A. M. Bradley, P. A. Bosler, O. Guba, M. A. Taylor, and G. A. Barnett. Communication-efficient property preservation in tracer transport. *SIAM J. Sci. Comput.*, 41(3):C161–C193, 2019. doi: https://doi.org/10.1137/18m1165414.

Andrew M Bradley, Peter A Bosler, and Oksana Guba. Islet: interpolation semi-Lagrangian element-based transport. *Geosci. Model Dev.*, 15(16):6285–6310, 2022. doi: https://doi.org/10.5194/gmd-15-6285-2022.

P. M. Caldwell, C. R. Terai, B. R. Hillman, N. D. Keen, P. A Bogenschutz, W. Lin, H. Beydoun, M. A. Taylor, L. Bertagna, A. M. Bradley, T. C. Clevenger, A. S. Donahue, C. Eldred, J. Foucar, J.-C. Golaz, O. Guba, R. Jacob, J. Johnson, J. Krishna, W. Liu, K. Pressel, A. G. Salinger, B. Singh, A. Steyer, P. Ullrich, D. Wu, X. Yuan, J. Shpund, H.-Y. Ma, and C. S. Zender. Convection-permitting simulations with the e3sm global atmosphere model. *J. Adv. Model Earth Syst.*, page e2021MS002544, 2021.

G. Cohen. *Higher-Order Numerical Methods for Transient Wave Equations.* Springer Science & Business Media, 2001.

J. Dennis, A. Fournier, W. F. Spotz, A. St-Cyr, M. A. Taylor, S. J. Thomas, and H. Tufo. High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core. *Int. J. High Perform. Comput. Appl.*, 19(3):225–235, 2005.

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

G Djoumna, R Pierre, and DY Le Roux. High-order C^1 finite-element interpolating schemes—Part I: Semi-Lagrangian linear advection. *Int. J. Numer. Methods Fluids*, 57(11):1603–1627, 2008.

J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. of Comput. and Appl. Math.*, 6(1):19–26, 1980.

Jim Douglas, Jr and Thomas F Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19(5):871–885, 1982.

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

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

Roberto Ferretti. Equivalence of semi-Lagrangian and Lagrange-Galerkin schemes under constant advection speed. *J. Comput. Math.*, pages 461–473, 2010.

Roberto Ferretti. On the relationship between semi-Lagrangian and Lagrange-Galerkin schemes. *Numer. Math.*, 124(1):31–56, 2013.

Roberto Ferretti and Michel Mehrenberger. Stability of semi-Lagrangian schemes of arbitrary odd degree under constant and variable advection speed. *Math. Comput.*, 89(324):1783–1805, 2020.

Natasha Flyer and Erik Lehto. Rotational transport on a sphere: Local node refinement with radial basis functions. *J. Comput. Phys.*, 229(6):1954–1969, 2010.

Natasha Flyer and Grady B Wright. Transport schemes on a sphere using radial basis functions. *J. Comput. Phys.*, 226(1):1059–1084, 2007.

Francis X Giraldo. The Lagrange–Galerkin spectral element method on unstructured quadrilateral grids. *J. Comput. Phys.*, 147(1):114–146, 1998. doi: https://doi.org/10.1006/jcph.1998.6078.

J.-C. Golaz, P. M. Caldwell, L. P. Van Roekel, M. R. Petersen, Q. Tang, J. D. Wolfe, G. Abeshu, V. Anantharaj, X. S. Asay-Davis, D. C. Bader, et al. The DOE E3SM coupled model version 1: Overview and evaluation at standard resolution. *J. Adv. Model Earth Syst.*, 11(7):2089–2129, 2019.

Jean-Christophe Golaz, Luke P. Van Roekel, Xue Zheng, Andrew Roberts, Jonathan D Wolfe, Wuyin Lin, Andrew Bradley, Qi Tang, Mathew E Maltrud, Ryan M Forsyth, Chengzhu Zhang, Tian Zhou, Kai Zhang, Charles Sutton Zender, Mingxuan Wu, Hailong Wang, Adrian K Turner, Balwinder Singh, Jadwiga H. Richter, Yi Qin, Mark R. Petersen, Azamat Mametjanov, Po-Lun Ma, Vincent E Larson, Jayesh Krishna, Noel D. Keen, Nicole Jeffery, Elizabeth C Hunke, Walter M. Hannah, Oksana Guba, Brian M Griffin, Yan Feng, Darren Engwirda, Alan V. Di Vittorio, Cheng Dang, LeAnn M. Conlon, Chih-Chieh-Jack Chen, Michael A. Brunke, Gautam Bisht, James J. Benedict, Xylar S. Asay-Davis, Yuying Zhang, Meng Zhang, Xubin Zeng, Shaocheng Xie, Phillip J. Wolfram, Tom Vo, Milena Veneziani, Teklu K. Tesfa, Sarat Sreepathi, Andrew G. Salinger, J. E. Jack Reeves Eyre, Michael J. Prather, Salil Mahajan, Qing Li, Philip W. Jones, Robert L. Jacob, Gunther W. Huebler, Xianglei Huang, Benjamin R. Hillman, Bryce E. Harrop, James G. Foucar, Yilin Fang, Darin S. Comeau, Peter M. Caldwell, Tony Bartoletti, Karthik Balaguru, Mark A. Taylor, Renata B. McCoy, L. Ruby Leung, and David C. Bader. The DOE E3SM Model Version 2: Overview of the physical model and initial model evaluation. *J. Adv. Model Earth Syst.*, 14(12), 2022. doi: https://doi.org/10.1029/2022MS003156.

W. M. Hannah, A. M. Bradley, O. Guba, Q. Tang, J.-C. Golaz, and J. Wolfe. Separating physics and dynamics grids for improved computational efficiency in spectral element earth system models. *J. Adv. Model Earth Syst.*, 13(7):e2020MS002419, 2021.

S. Idelsohn, M. Storti, and N. Nigro. Stability analysis of mixed finite element formulations with special mention of equal-order interpolations. *Int. J. Numer. Methods Fluids*, 20(8-9):1003–1022, 1995.

P. H. Lauritzen, R. D. Nair, and P. A. Ullrich. A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *J. Comput. Phys.*, 229(5):1401–1424, 2010.

P. H. Lauritzen, W. C. Skamarock, M. J. Prather, and M. A. Taylor. A standard test case suite for two-dimensional linear transport on the sphere. *Geosci. Model Dev.*, 5(3), 2012.

P. H. Lauritzen, P. A. Ullrich, C. Jablonowski, P. A. Bosler, D. Calhoun, A. J. Conley, T. Enomoto, L. Dong, S. Dubey, O. Guba, 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(1): 105–145, 2014. doi: https://doi.org/10.5194/gmd-7-105-2014.

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

Randall J LeVeque. *Finite Volume Methods for Hyperbolic Problems*, volume 31. Cambridge University Press, 2002.

D. W. Lozier. NIST digital library of mathematical functions. *Ann. Math. Artif. Intel.*, 38(1): 105–119, 2003.

J. Michalakes, M. Govett, R. Benson, T. Black, H. Juang, A. Reinecke, B. Skamarock, M. Duda, T. Henderson, P. Madden, G. Mozdzynski, and R. Vasic. AVEC report: NGGPS level-1 benchmarks and software evaluation. Technical report, NOAA, 2016. URL https://repository.library.noaa.gov/view/noaa/18654.

Vahid Mohammadi, Mehdi Dehghan, Amirreza Khodadadian, and Thomas Wick. Numerical investigation on the transport equation in spherical coordinates via generalized moving least squares and moving kriging least squares approximations. *Eng. Comput.*, 37:1231–1249, 2021.

KW Morton, Andrew Priestley, and Endre Suli. Stability of the Lagrange-Galerkin method with non-exact integration. *ESAIM: Math. Model. Numer. Anal.*, 22(4):625–653, 1988.

Ramachandran D Nair and Christiane Jablonowski. Moving vortices on the sphere: A test case for horizontal advection problems. *Mon. Weather Rev.*, 136(2):699–711, 2008.

H. Natarajan and G. B. Jacobs. An explicit semi-Lagrangian, spectral method for solution of Lagrangian transport equations in Eulerian-Lagrangian formulations. *Comput. Fluids*, 207:104526, 2020.

A. Oliveira and A. M. Baptista. A comparison of integration and interpolation Eulerian-Lagrangian methods. *Int. J. Numer. Methods Fluids*, 21(3):183–204, 1995.

Olivier Pironneau. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numer. Math.*, 38(3):309–332, 1982.

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

Varun Shankar and Grady B Wright. Mesh-free semi-Lagrangian methods for transport on a sphere using radial basis functions. *J. Comput. Phys.*, 366:170–190, 2018.

Piotr K. Smolarkiewicz and Janusz A. Pudykiewicz. A class of semi-Lagrangian approximations for fluids. *J. Atmos. Sci.*, 49(22):2082–2096, 1992.

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

J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* SIAM, 2004.

Q. Tang, S. A. Klein, S Xie, W. Lin, J.-C. Golaz, E. L. Roesler, M. A. Taylor, P. J. Rasch, D. C. Bader, L. K. Berg, et al. Regionally refined test bed in E3SM atmosphere model version 1 (EAMv1) and applications for high-resolution modeling. *Geosci. Model Dev.*, 12(7):2679–2706, 2019.

M. A. Taylor, O. Guba, A. Steyer, P. A. Ullrich, D. M. Hall, and C. Eldrid. An energy consistent discretization of the nonhydrostatic equations in primitive variables. *J. Adv. Model Earth Syst.*, 12(1), 2020.

Seth Taylor and Jean-Christophe Nave. A projection-based characteristic mapping method for tracer transport on the sphere. *J. Comput. Phys.*, 477:111905, 2023.

Paul A. Ullrich and Matthew R. Norman. The flux-form semi-Lagrangian spectral element (FF-SLSE) method for tracer transport. *Q. J. Roy. Meteor. Soc.*, 140(680):1069–1085, 2014.

R. Vichnevetsky and J. B. Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations.* SIAM, 1982.

J. Whitaker. HIWPP non-hydrostatic dynamical core tests: Results from idealized test cases. Technical report, NOAA, 2016. URL https://www.weather.gov/media/sti/nggps/HIWPP_idealized_tests-v8%20revised%2005212015.pdf.

Dongbin Xiu and George Em Karniadakis. A semi-Lagrangian high-order method for Navier–Stokes equations. *J. Comput. Phys.*, 172(2):658–684, 2001. doi: https://doi.org/10.1006/jcph.2001.6847.