A spectral element semi-Lagrangian (SESL) method for the spherical shallow water equations

F.X. Giraldo [a,*], J.B. Perot [b], P.F. Fischer [c]

[a] *Naval Research Laboratory, Monterey, CA 93943, USA*
[b] *University of Massachusetts, Amherst, MA 01003, USA*
[c] *Argonne National Laboratory, Argonne, IL 60439, USA*

Abstract

A spectral element semi-Lagrangian (SESL) method for the shallow water equations on the sphere is presented. The sphere is discretized using a hexahedral grid although any grid imaginable can be used as long as it is comprised of quadrilaterals. The equations are written in Cartesian coordinates to eliminate the pole singularity which plagues the equations in spherical coordinates. In a previous paper [Int. J. Numer. Methods Fluids 35 (2001) 869] we showed how to construct an explicit Eulerian spectral element (SE) model on the sphere; we now extend this work to a semi-Lagrangian formulation. The novelty of the Lagrangian formulation presented is that the high order SE basis functions are used as the interpolation functions for evaluating the values at the Lagrangian departure points. This makes the method not only high order accurate but quite general and thus applicable to unstructured grids and portable to distributed memory computers. The equations are discretized fully implicitly in time in order to avoid having to interpolate derivatives at departure points. By incorporating the Coriolis terms into the Lagrangian derivative, the block LU decomposition of the equations results in a symmetric positive-definite pseudo-Helmholtz operator which we solve using the generalized minimum residual method (GMRES) with a fast projection method [Comput. Methods Appl. Mech. Eng. 163 (1998) 193]. Results for eight test cases are presented to confirm the accuracy and stability of the method. These results show that SESL yields the same accuracy as an Eulerian spectral element semi-implicit (SESI) while allowing for time-steps 10 times as large and being up to 70% more efficient.
© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Cubic gnomonic; Flux-corrected transport; Hexahedral grid; Shallow water equations; Semi-Lagrangian; Semi-implicit; Spectral element method; Spherical geometry

1. Introduction

In [9] a method obtained by fusing the the spectral element method with the semi-Lagrangian method was introduced and applied to the advection–diffusion equation. A stability analysis revealed that this

---

hybrid method yields exponential convergence for increasing basis function order while experiencing neither dispersion nor dissipation errors; that is, provided that the solution is smooth. In that work, however, we only showed results for the 1D and 2D advection–diffusion equations. In a more recent paper [11], we showed how to apply the spectral element method to the shallow water equations on the sphere using Cartesian rather than spherical coordinates. Although using Cartesian coordinates seems counter-intuitive this coordinate system avoids the singularities at the poles encountered by spherical coordinates. This formulation of the equations is not dependent on the grid as is the case with all currently existing models [15,17,20,32]. In that paper [11] we showed that this approach yields exponential convergence for the shallow water equations; however, in that work the time integration scheme used was an explicit Eulerian method (third order Adams–Bashforth).

Although explicit time-integration methods for atmospheric simulations are quite efficient, their main disadvantage is that small time-steps must be observed in order to maintain stability. The reason for this prohibitively small time-step is due to the fast moving gravity waves. These waves require a small time-step while only carrying a very small percent of the total energy in the system. In order to ameliorate this rather stringent time-step restriction researchers have tried various approaches such as using a larger differencing stencil for the gravity wave terms thereby effectively reducing the Courant number [24,35], and discretizing the gravity wave terms implicitly in time [19]. The first strategy, that is using a larger differencing stencil for the gravity wave terms, is typically done to avoid the inf–sup (Babuska–Brezzi) condition [18] associated with the Stokes problem. However, discretizing the gravity wave terms implicitly in time is a much more effective way of increasing the time-step; this is the approach we follow in this paper.

After the gravity wave terms have been successfully discretized the next set of terms responsible for controlling the maximum time-step are the advection terms. In order to use increasingly larger time-steps researchers have turned to Lagrangian methods for treating these recalcitrant terms [26,33]. By rewriting the equations in terms of the Lagrangian derivative the troublesome advection terms are absorbed into the Lagrangian derivative. Thus the equations in this form are now discretized in time along characteristics which results in a much more stable numerical method due to the *virtual* disappearance of the Courant–Friedrichs–Lewy (CFL) condition. We say *virtual* because any time-step is possible provided that the Lipschitz condition is not violated and that the departure points of the semi-Lagrangian method are computed accurately.

In the current paper we combine a semi-Lagrangian method with high order basis functions and extend this hybrid method to the solution of the spherical shallow water equations. Thus we are essentially extracting the highlights of our previous papers [9,11] and fusing them into a powerful technique for solving the shallow water equations on the sphere. The allure of this method is that it achieves the same order of accuracy obtained with exponentially high order Eulerian methods [11] while using much larger time-steps. It should be mentioned that this is not the only possible characteristic method. There are other variants that are certainly worth considering especially the characteristic method of Maday et al. [22] which we hope to compare with SESL in future work. In this paper we only consider SESL because it represents the only true characteristic high order method which fuses both space and time into a single formulation.

The remainder of this paper is organized as follows. Section 2 describes the governing equations of motion used to test our numerical method. In Section 3 we describe the discretization of the governing equations. This includes the spatial discretization by the spectral element method, the time discretization by the semi-Lagrangian method, and the solution of the resulting implicit equations. In Section 4 we describe the tessellation of the sphere into quadrilateral elements which are used to construct the discrete operators. Finally, in Section 5 we present convergence rates of the spectral element semi-Lagrangian (SESL) method and compare it with an Eulerian spectral element semi-implicit (SESI) method. This then leads to some conclusions about the feasibility of this approach for constructing future atmospheric models and a discussion on the direction of future work.

2. Shallow water equations

The shallow water equations are a system of first order nonlinear hyperbolic equations which govern the motion of an inviscid incompressible fluid in a shallow depth. The predominant feature of this type of fluid is that the characteristic length of the fluid is far greater than its depth which is analogous to the motion of air in the atmosphere. For this reason these equations are typically used as a first step toward the construction of either numerical weather prediction (NWP) or climate models; the former type of model is the final goal of our research.

The spherical shallow water equations in Cartesian advection form are

$$\frac{\partial \varphi}{\partial t} + \boldsymbol{u} \cdot \nabla \varphi \boldsymbol{u} = -\varphi \nabla \cdot \boldsymbol{u}, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{2\omega z}{a^2}(\boldsymbol{x} \times \boldsymbol{u}) - \nabla(\varphi + \varphi^s) - \mu \boldsymbol{x}, \tag{2}$$

where φ is the geopotential height ($\varphi = gh$, where g is the gravitational constant and h is the vertical height of the fluid), φ^s is the surface topography (e.g., mountains), $\boldsymbol{u}$ is the Cartesian wind velocity vector (u, v, w), and (ω, a) represent the rotation of the earth and its radius, respectively.

The term $\mu \boldsymbol{x}$, where $\boldsymbol{x} = (x, y, z)$ is the position vector of the grid points, is a fictitious force introduced to constrain the fluid particles to remain on the surface of the sphere. By switching from spherical (2D) to Cartesian (3D) coordinates we have allowed the fluid particles an additional degree of freedom which will manifest itself in the fluid particles flying off the sphere. In order to prevent this from happening we introduce the Lagrange multiplier μ (see [4]); we address the calculation of this coefficient in Section 3.4.

The shallow water equations in Cartesian form have received significant attention recently (see [10–12,16,31]). It should be mentioned that using these equations in Cartesian form on the sphere introduces no approximations whatsoever; the equations are completely equivalent to the equations in spherical coordinates as shown by Swarztrauber et al. [31]. Although the formulation we use in our paper may appear slightly different it can be shown to be identical to that of [16,31].

3. Discretization

In this section we describe the discretization of the shallow water equations. In Section 3.1, we describe the spatial discretization by the spectral element method including: the choice of basis functions and the mapping from physical 3D Cartesian space to the local 2D computational space. In Section 3.2 we discuss the implicit time discretization by the semi-Lagrangian method including the discretization of the trajectory equation. In addition, we describe the interpolation operators required by the semi-Lagrangian method as well as the construction of monotone interpolation schemes. Finally, in Section 3.4 we discuss the solution of the coupled implicit equations.

3.1. Spatial discretization by the spectral element method

3.1.1. Basis functions and integration
To define the local operators which shall be used to construct the global approximation of the solution we begin by decomposing the spherical domain Ω into N_e non-overlapping quadrilateral elements such that

$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e.$$

To perform differentiation and integration operations, we introduce the non-singular mapping $x = \Psi(\xi)$ which defines a transformation from the physical Cartesian coordinate system $x = (x, y, z)$ defined in Ω_e to the reference coordinate system $\xi = (\xi, \eta, \zeta)$ defined in each element such that (ξ, η) lies on the spherical surface; where $(\xi, \eta) \in [-1, +1]^2$ in each element.

Associated with the local mapping, Ψ, is the transformation Jacobian, $J = \partial x / \partial \xi$, and the determinant

$$|J| = \frac{\partial x}{\partial \zeta} \cdot \boldsymbol{G}, \qquad \boldsymbol{G} = \frac{\partial x}{\partial \xi} \times \frac{\partial x}{\partial \eta},$$

where $\boldsymbol{G}$ represents the surface conforming component of the mapping (see [11] for further details).

We can now use this mapping to define the local representation of the solution, $q = (\varphi, \boldsymbol{u})$, and the approximation of operations such as differentiation and integration. For simplicity, we assume ζ to be unity in what remains and denote $\xi = (\xi, \eta)$.

The simple structure of the reference element, I, spanned by $\xi \in [-1, 1]^2$, makes it natural to represent the local element-wise solution q by an Nth order polynomial in ξ as

$$\boldsymbol{q}_N(x) = \sum_{k=1}^{(N+1)^2} \psi_k(x) \boldsymbol{q}_N(x_k),$$

where x_k represents $(N + 1)^2$ grid points and $\psi_k(x)$ reflects the associated multivariate Lagrange polynomial. The logical square structure of I simplifies matters in that we can express the Lagrange polynomial by a tensor-product as

$$\psi_k(x) = h_i(\xi(x)) h_j(\eta(x)),$$

where $i, j = 0, \dots, N$.

For the grid points (ξ_i, η_j) we choose the Legendre–Gauss–Lobatto (LGL) points, given as the tensor-product of the roots of

$$(1 - \xi^2) P'_N(\xi) = 0,$$

where $P_N(\xi)$ is the Nth order Legendre polynomial. This choice simplifies the construction of the algorithm because the LGL points are also used as the sampling points in the Gaussian quadrature rule required by the numerical integration which we shall describe shortly. The one-dimensional Lagrange polynomials, $h_i(\xi)$, are

$$h_i(\xi) = -\frac{1}{N(N+1)} \frac{(1 - \xi^2) P'_N(\xi)}{(\xi - \xi_i) P_N(\xi_i)},$$

and likewise for $h_j(\eta)$.

The choice of the LGL points enables the straightforward approximation of local element integrals, i.e.,

$$\int_{\Omega_e} q(x) \, \mathrm{d}x = \int_I q(\xi) J(\xi) \, \mathrm{d}\xi \simeq \sum_{i,j=0}^{N} \omega(\xi_i) \omega(\eta_j) q(\xi_i, \eta_j) J(\xi_i, \eta_j),$$

where J represents the local Jacobian for the transformation between Ω_e and I, and $\omega(\xi_i)$ and $\omega(\eta_j)$ are the Gaussian quadrature weights,

$$\omega(\xi_i) = \frac{2}{N(N+1)} \left(\frac{1}{P_N(\xi_i)} \right)^2,$$

associated with the one-dimensional LGL quadrature.

To simplify the description of the numerical algorithm, let us define the following local element operators: let

$$M^e_{ij} = \int_{\Omega_e} \psi_i(\boldsymbol{x})\psi_j(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

represent the mass matrix and

$$\boldsymbol{D}^e_{ij} = \int_{\Omega_e} \psi_i(\boldsymbol{x})\nabla\psi_j(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

the differentiation matrix where $i, j = 1, \ldots, (N+1)^2$. Note that $\boldsymbol{D} = (D^x, D^y, D^z)$ is a vector of matrices corresponding to the three spatial directions for $\boldsymbol{D}$.

3.1.2. Satisfying the equations globally

To satisfy the equations globally requires assembling the global solution by virtue of an element-wise construction. This element-wise construction is based on the summation of the local element matrices to form their global representation. This summation procedure is known as the global assembly or direct stiffness summation. Let us represent this global assembly procedure by the summation operator

$$\bigwedge_{e=1}^{N_e}$$

with the mapping $(i, e) \rightarrow (I)$ where $i = 1, \ldots, (N+1)^2$ are the local element grid points, $e = 1, \ldots, N_e$ are the spectral elements covering the spherical shell, and $I = 1, \ldots, N_p$ are the global grid points. Applying the global assembly operator to the local element matrices results in the following global matrices:

$$M = \bigwedge_{e=1}^{N_e} M^e$$

for the mass matrix and

$$\boldsymbol{D} = \bigwedge_{e=1}^{N_e} \boldsymbol{D}^e$$

for the differentiation matrix.

With these operators defined and by denoting the global grid vector for the grid points as $\boldsymbol{x}_{\mathrm{G}}$, the geopotential as φ_{G}, and the wind velocity as $\boldsymbol{u}_{\mathrm{G}}$ we can now write the semi-discrete approximation to Eqs. (1) and (2) as follows:

$$M\frac{\partial\varphi_{\mathrm{G}}}{\partial t} + \boldsymbol{u}_{\mathrm{G}}^{\mathrm{T}}\boldsymbol{D}\varphi_{\mathrm{G}} = -\varphi_{\mathrm{G}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{u}_{\mathrm{G}}, \tag{3}$$

$$M\frac{\partial\boldsymbol{u}_{\mathrm{G}}}{\partial t} + \boldsymbol{u}_{\mathrm{G}}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{u}_{\mathrm{G}} = -M\left(\frac{2\omega z_{\mathrm{G}}}{a^2}(\boldsymbol{x}_{\mathrm{G}} \times \boldsymbol{u}_{\mathrm{G}})\right) - \boldsymbol{D}(\varphi_{\mathrm{G}} + \varphi_{\mathrm{G}}^{\mathrm{s}}) - M(\mu\boldsymbol{x}_{\mathrm{G}}). \tag{4}$$

It should be noted that the mass matrix, M, is diagonal and thereby trivial to invert. The diagonal property of this matrix is due to the dual role of the LGL points which are used both as grid points and quadrature points.

3.2. Time discretization by the semi-Lagrangian method

The time-step restriction of the shallow water equations is dominated by the advection and gravity wave terms. In fact, the characteristic wave speed of this system is $C = U + \sqrt{\varphi}$. However, the gravity wave term although traveling at far greater speeds than the wind velocity only carries a fraction of the total energy. Thus, explicit time integration schemes must use very small time-steps in order to maintain stability while much larger values could be used without diminishing accuracy. Typically, in order to enlarge the time-step, semi-implicit schemes [19] have been used whereby the advection and Coriolis terms are discretized explicitly in time while the remaining terms are discretized implicitly. The generalized leapfrog method

$$\frac{\varphi^{n+1} - \varphi^{n-1}}{2\Delta t} = -(\boldsymbol{u} \cdot \boldsymbol{\nabla}\varphi)^n - \Phi\big(\theta\boldsymbol{\nabla} \cdot \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{\nabla} \cdot \boldsymbol{u}^{n-1}\big), \tag{5}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n-1}}{2\Delta t} = -(\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u})^n - \frac{2\omega z}{a^2}(\boldsymbol{x} \times \boldsymbol{u})^n - \Big(\theta\boldsymbol{\nabla}(\varphi + \varphi^s)^{n+1} + (1-\theta)\boldsymbol{\nabla}(\varphi + \varphi^s)^{n-1}\Big) \tag{6}$$

has been the most popular method for constructing semi-implicit algorithms for the shallow water and atmospheric equations (see for example the operational models in [14,17,29,30] and the shallow water model in [34]). Note that φ has been linearized about some mean state Φ in order to avoid having to solve a nonlinear problem and we have omitted the Lagrange multiplier for the time being. In Eqs. (5) and (6) the parameter θ determines the type of scheme used in conjunction with the leapfrog method for the advection terms. For example, $\theta = 0, \frac{1}{2}, 1$ yields the Euler method (explicit), the Crank–Nicholson method (implicit), and the backward Euler method (implicit), respectively. For $\theta = \frac{1}{2}$ the scheme is second order accurate in time and for all other values it is only first order accurate.

To further enlarge the time-step we can incorporate the advection terms into the Lagrangian derivative; this is the semi-Lagrangian method [26]. Integrating Eqs. (5) and (6) along characteristics results in

$$\frac{\varphi^{n+1} - \widetilde{\varphi}^{n-1}}{2\Delta t} = -\Phi\big(\theta\boldsymbol{\nabla} \cdot \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{\nabla} \cdot \widetilde{\boldsymbol{u}}^{n-1}\big),$$

$$\frac{\boldsymbol{u}^{n+1} - \widetilde{\boldsymbol{u}}^{n-1}}{2\Delta t} = -\frac{2\omega\widetilde{z}}{a^2}(\widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{u}})^n - \Big(\theta\boldsymbol{\nabla}(\varphi + \varphi^s)^{n+1} + (1-\theta)\boldsymbol{\nabla}(\widetilde{\varphi} + \widetilde{\varphi}^s)^{n-1}\Big),$$

where it should be understood that the values at times n and $n-1$ are not the values at the Eulerian grid points but rather those at the Lagrangian departure points found by integrating the trajectory equation

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{u}, \tag{7}$$

where the Lagrangian derivative is

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla}.$$

We have marked the variables that need to be interpolated at the departure points with a tilde (˜). The statement of the semi-Lagrangian method is: find $\widetilde{\boldsymbol{q}}^{(n,n-1)}$ using Eq. (7) such that at time $n+1$ these characteristics arrive at the Eulerian grid points (arrival points).

Although the Crank–Nicholson method has been the most popular method used with the semi-Lagrangian method it requires the computation of not only the primitive variables at the departure points but their derivatives as well. This step in the semi-Lagrangian procedure requires a very careful treatment; solving these terms indiscriminately results in severe loss of conservation. For this reason, many operational

NWP models use off-centering, $\theta > 1/2$, to combat the inaccuracies of computing derivatives at the departure points. The problem with off-centering is that the scheme is only first order accurate. Models which perform off-centering in either an Eulerian or semi-Lagrangian context include the operational NWP and climate models in [14,17,29,30].

Galerkin methods pose some difficulties for semi-Lagrangian methods. The problem stems from the C^0 basis functions typically used in Galerkin methods which translates into the derivatives being discontinuous across elements. For this reason many semi-Lagrangian implementations construct the right-hand side (i.e., the terms that must be evaluated at the departure points) at the arrival points and then interpolate the entire right-hand side vector to the departure points. This, however, will conserve neither mass nor energy. This leads us to the following conjecture.

Conjecture 1. Semi-Lagrangian methods with C^0 basis functions should only be used for homogeneous Lagrangian equations of the type

$$\frac{\mathrm{d}\varphi}{\mathrm{d}t} = 0.$$

However, if there are source terms with derivatives on the right-hand side then they should be either evaluated at arrival points only or C^k basis functions should be used, where k represents the maximum order of the derivatives in the system ($k = 1$ for the shallow water equations). We reserve the proof of this conjecture for a future study.

The problem with using C^k functions is that the number of unknowns increases from 4 to 16 per grid point due to the first order derivatives that need to be carried. Xiu and Karniadakis [37] also noted some problems with evaluating derivatives at the departure points and thus decided to use a backward difference formula (BDF) which gives rise to a fully implicit scheme.

Before discretizing the equations along the characteristics let us write the shallow water equations in the Lagrangian form which we shall use to solve the equations. Using the idea of Rochas [27] we include the Coriolis term inside the Lagrangian derivative and write the equations as follows:

$$\frac{\mathrm{d}\varphi}{\mathrm{d}t} = -\Phi \mathbf{\nabla} \cdot \boldsymbol{u}, \tag{8}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{u} + 2\omega(\boldsymbol{k} \times \boldsymbol{x})) = -\mathbf{\nabla}(\varphi + \varphi^{\mathrm{s}}) - \mu\boldsymbol{x}, \tag{9}$$

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{u},$$

where $\boldsymbol{k} = (0, 0, 1)$ is the unit vector in the z-direction. We can now discretize these equations using a second order BDF to yield

$$\frac{\varphi^{n+1} - \frac{4}{3}\widetilde{\varphi}^n + \frac{1}{3}\widetilde{\varphi}^{n-1}}{\Delta t} = -\frac{2}{3}\Phi\mathbf{\nabla} \cdot \boldsymbol{u}^{n+1}, \tag{10}$$

$$\frac{\widehat{\boldsymbol{u}}^{n+1} - \frac{4}{3}\widehat{\boldsymbol{u}}^n + \frac{1}{3}\widehat{\boldsymbol{u}}^{n-1}}{\Delta t} = -\frac{2}{3}\mathbf{\nabla}(\varphi + \varphi^s)^{n+1}, \tag{11}$$

where

$$\widehat{\boldsymbol{u}} = \boldsymbol{u} + 2\omega(\boldsymbol{k} \times \boldsymbol{x})$$

represents the absolute velocity in an inertial reference frame. Note that Eqs. (10) and (11) do not require the interpolation of derivatives at the departure points; for this reason we have elected to use a BDF.

Moving the implicit terms to the left-hand side and the explicit terms to the right-hand side results in

$$\varphi^{n+1} + \frac{2}{3}\Delta t \Phi \mathbf{V} \cdot \boldsymbol{u}^{n+1} = b^\varphi, \tag{12}$$

$$\boldsymbol{u}^{n+1} + \frac{2}{3}\Delta t \mathbf{V}\varphi^{n+1} = \boldsymbol{b}^u, \tag{13}$$

where the right-hand side vectors are defined as follows:

$$b^\varphi = \frac{4}{3}\widetilde{\varphi}^n - \frac{1}{3}\widetilde{\varphi}^{n-1}, \tag{14}$$

$$\boldsymbol{b}^u = \frac{4}{3}(\widetilde{\boldsymbol{u}} + 2\omega(\boldsymbol{k} \times \widetilde{\boldsymbol{x}}))^n - \frac{1}{3}(\widetilde{\boldsymbol{u}} + 2\omega(\boldsymbol{k} \times \widetilde{\boldsymbol{x}}))^{n-1} - \frac{2}{3}\Delta t(\mathbf{V}\varphi^s)^{n+1} - 2\omega(\boldsymbol{k} \times \boldsymbol{x})^{n+1}. \tag{15}$$

We now describe the computation of the departure point values.

3.2.1. Trajectory equation

In order to evaluate the variables $\widetilde{\boldsymbol{q}}^{(n,n-1)}$ require the accurate computation of the departure points which are found from Eq. (7). The simplest second order scheme for discretizing this equation is the two-step Runge–Kutta scheme:

$$\widetilde{\boldsymbol{x}}^{n+1/2} = \boldsymbol{x}^{n+1} - \frac{\Delta t}{2}\boldsymbol{u}(\boldsymbol{x}^{n+1}, t^{n+1}),$$

$$\widetilde{\boldsymbol{x}}^n = \boldsymbol{x}^{n+1} - \Delta t\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{n+1/2}, t^{n+1/2}),$$

$$\widetilde{\boldsymbol{x}}^{n-1/2} = \boldsymbol{x}^n - \frac{\Delta t}{2}\boldsymbol{u}(\widetilde{\boldsymbol{x}}^n, t^n),$$

$$\widetilde{\boldsymbol{x}}^{n-1} = \boldsymbol{x}^n - \Delta t\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{n-1/2}, t^{n-1/2}),$$

where the velocity field is extrapolated as follows:

$$\boldsymbol{u}(\boldsymbol{x}, t^{n+1}) = 2\boldsymbol{u}(\boldsymbol{x}, t^n) - \boldsymbol{u}(\boldsymbol{x}, t^{n-1}),$$

$$\boldsymbol{u}(\boldsymbol{x}, t^{n+1/2}) = \frac{3}{2}\boldsymbol{u}(\boldsymbol{x}, t^n) - \frac{1}{2}\boldsymbol{u}(\boldsymbol{x}, t^{n-1}),$$

$$\boldsymbol{u}(\boldsymbol{x}, t^{n-1/2}) = \frac{1}{2}\boldsymbol{u}(\boldsymbol{x}, t^n) + \frac{1}{2}\boldsymbol{u}(\boldsymbol{x}, t^{n-1}).$$

However, we do not have to stop at second order but instead can construct higher order trajectory approximations. For example we can construct the fourth order Runge–Kutta approximation

$$\widetilde{\boldsymbol{x}}^n = \boldsymbol{x}^{n+1} - \frac{\Delta t}{6}\left[\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{(1)}, t^{n+1}) + 2\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{(2)}, t^{n+1/2}) + 2\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{(3)}, t^{n+1/2}) + \boldsymbol{u}(\widetilde{\boldsymbol{x}}^{(4)}, t^n)\right], \tag{16}$$

where the trial points are

$$\widetilde{\boldsymbol{x}}^{(1)} = \boldsymbol{x}^{n+1},$$

$$\widetilde{\boldsymbol{x}}^{(2)} = \boldsymbol{x}^{n+1} - \frac{\Delta t}{2} \boldsymbol{u}(\boldsymbol{x}^{n+1}, t^{n+1}),$$

$$\widetilde{\boldsymbol{x}}^{(3)} = \boldsymbol{x}^{n+1} - \frac{\Delta t}{2} \boldsymbol{u}(\widetilde{\boldsymbol{x}}^{n+1/2}, t^{n+1/2}),$$

$$\widetilde{\boldsymbol{x}}^{(4)} = \boldsymbol{x}^{n+1} - \Delta t \boldsymbol{u}(\widetilde{\boldsymbol{x}}^{n+1/2}, t^{n+1/2})$$

and the velocity field needs to be extrapolated to sufficiently high order. We have included a tilde in the trial points $\widetilde{\boldsymbol{x}}^{(k)}$ for $k = 1, \ldots, 4$ to remind the reader that the value of the velocity field $\boldsymbol{u}(\widetilde{\boldsymbol{x}}^{(k)})$ must be interpolated at these trial points since they will generally not lie on a grid point.

3.2.2. Interpolation and flux-corrected transport

Once we determine the departure point and the element in which it lies we must then construct an interpolation operator using its local information (the departure point search is described in Section 4.2). Let us assume that the element Ω_d claims the departure point $\boldsymbol{x}_d$. Then the interpolated value $\widetilde{\varphi}$ is simply given as

$$\widetilde{\varphi}_d = \sum_{i=0}^{N} \sum_{j=0}^{N} h_i(\xi_d) h_j(\eta_d) \varphi_{ij},$$

where φ_{ij} are the grid point values of the element Ω_d, and the departure point value $\boldsymbol{x}_d$ has been mapped to computational space (ξ_d, η_d). Thus for high order N we cannot expect to have a monotonic interpolation. Therefore, just as the spectral element method is not naturally monotonic than neither is the high order semi-Lagrangian method. The lack of monotonicity does not have serious repercussions except when very large gradients (such as shocks) are encountered. One of the test cases that we use in this paper contains a shock (Case 1C) and for this case we shall use a monotonic version of SESL.

The simplest way to make a high order interpolation monotonic is to use a linear interpolation operator. In other words, instead of using all of the $(N + 1)^2$ grid points in an element we can just use its vertices. However, using these values indiscriminately throughout the computation will result in a very diffused solution. Therefore, we apply the monotonicity constraint based on whether the full polynomial interpolation generates values that are larger or smaller than the values of the element Ω_d. This FCT-type method (flux-corrected transport, see [1]) can be written as follows:

$$\widetilde{\varphi}_d = \min[\varphi_{\max}, \max(\varphi_{\min}, \widetilde{\varphi}_N)],$$

where $\widetilde{\varphi}_N$ is the interpolation using the full polynomial

$$\widetilde{\varphi}_N = \sum_{i=0}^{N} \sum_{j=0}^{N} h_i(\xi_d) h_j(\eta_d) \varphi_{ij},$$

and $\varphi_{\max}$ and $\varphi_{\min}$ are the maximum and minimum values in the element Ω_d, respectively. In other words, this method now keeps the interpolated value from exceeding the extrema of the element Ω_d and thus resulting in a monotonic interpolation operator.

3.3. Space–time discretization

In all previous fluid dynamics applications whereby Lagrangian schemes are combined with Galerkin methods (either spectral or finite elements) the two methods are applied virtually independently. As an example, when spectral methods are used only ad-hoc cubic Lagrange polynomials are constructed to perform the interpolation at the departure points [25]. In the finite element method, only low order finite element methods (linear) are used in conjunction with cubic interpolation [5]. In these approaches, the interpolation operator is constructed independently of the spatial discretization methods; note that the two previously mentioned works both use cubic interpolation even though they use different order accuracy spatial discretization methods. In our approach, we use the high order basis functions of the spectral element method as the interpolation functions and so the interpolation operators are naturally and inherently contained within the spectral element method.

Since the appearance of [9] in which we introduced SESL for the advection–diffusion equation, two papers have appeared which have contributed to the further development of SESL-type methods. In the paper by Falcone and Ferretti [7], they show a rigorous proof for the approximation error of the advection equation with Lipschitz continuous solutions. In that paper, the approximation error for Lagrangian methods using high order interpolation was shown to be

$$\mathcal{O}\left(\Delta t^K + \frac{\Delta x^{N+1}}{\Delta t}\right),\tag{17}$$

where K is the order of accuracy used to solve the trajectory equation

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{u}$$

and N is the order of the interpolation functions. Therefore, if either high order K is used with low order N or vice-versa, then the overall accuracy of the scheme is compromised. In all previous semi-Lagrangian implementations researchers used both low order K and N. They then noticed that when they increased one or the other they did not obtain increased accuracy. This led to the incorrect conclusion that it is not beneficial to increase these parameters beyond a certain value; however, no one tried to increase both values.

In the paper by Xiu and Karniadakis [37], they show numerically that the error of the semi-Lagrangian method combined with the spectral element method is indeed controlled by Eq. (17); they show this for the 2D advection–diffusion equation and argue that the same behavior holds for the Navier–Stokes equations at high Reynolds number. In this paper we show that the error in SESL is bounded by Eq. (17).

Besides giving a bound on the error for high order Lagrangian methods, the result of Falcone and Ferretti can also be used to obtain the optimal time-step to use with a given order of accuracy spatial discretization. The optimal time-step is obtained when the first and second terms in Eq. (17) are balanced which occurs for the following time-step:

$$\Delta t^K = \frac{\Delta x^{N+1}}{\Delta t}.\tag{18}$$

Following Malevsky and Thomas [21] we can write the time-step as a function of the grid spacing as follows:

$$\Delta t = \Delta x^{\alpha}.\tag{19}$$

Substituting Eq. (19) into Eq. (18) results in the optimal time-step

$$\alpha = \frac{N+1}{K+1}.$$

3.4. Implicit formulation of the discrete system

In this section we describe the implicit formulation for the discrete equations. Discretizing Eqs. (12) and (13) by the spectral element method results in the following discrete system:

$$M\varphi_G^{n+1} + \lambda\Phi D^T u_G^{n+1} = B^\varphi, \tag{20}$$

$$M u_G^{n+1} + \lambda D\varphi_G^{n+1} = B^u - M(\mu x_G^{n+1}), \tag{21}$$

where $\lambda = (2/3)\Delta t$, $B^\varphi = Mb_G^\varphi$ and $B^u = Mb_G^u$. We next need to compute the Lagrange multiplier which will constrain the fluid particles to the sphere. Using Eq. (21) we obtain the grid point values of the momentum equation as follows:

$$u_G^{n+1} = M^{-1}B^u - \lambda M^{-1}D\varphi_G^{n+1} - \mu x_G^{n+1}. \tag{22}$$

Since we need to ensure that the velocity field remains tangential to the sphere, we require

$$x_G \cdot u_G = 0.$$

Taking the scalar product of Eq. (22) with x_G^{n+1} and rearranging yields

$$\mu = \frac{1}{a^2}M^{-1}(x_G^{n+1} \cdot B_u) - \frac{\lambda}{a^2}M^{-1}(x_G^{n+1} \cdot D\varphi^{n+1}). \tag{23}$$

Writing

$$Pq = q - \frac{1}{a^2}(x_G^{n+1} \cdot q), \tag{24}$$

where the projection matrix P is given by

$$P = \frac{1}{a^2}\begin{pmatrix} a^2 - x^2 & -xy & -xz \\ -xy & a^2 - y^2 & -yz \\ -xz & -yz & a^2 - z^2 \end{pmatrix} \tag{25}$$

we can now write the constrained equations as

$$M\varphi_G^{n+1} + \lambda\Phi D^T u_G^{n+1} = B^\varphi, \tag{26}$$

$$M u_G^{n+1} + \lambda P D\varphi_G^{n+1} = PB^u. \tag{27}$$

The projection matrix P constrains any vector quantity q to be tangential to the sphere.

Eqs. (26) and (27) can now be written in the matrix form

$$\begin{pmatrix} M & \lambda PD \\ \lambda\Phi D^T & M \end{pmatrix}\begin{pmatrix} u_G \\ \varphi_G \end{pmatrix}^{n+1} = \begin{pmatrix} PB_u \\ B^\varphi \end{pmatrix}. \tag{28}$$

Applying a block LU decomposition yields the following matrix system:

$$\begin{pmatrix} M & \lambda PD \\ 0 & M - \lambda^2\Phi D^T M^{-1}PD \end{pmatrix}\begin{pmatrix} u_G \\ \varphi_G \end{pmatrix}^{n+1} = \begin{pmatrix} PB_u \\ B^\varphi - \lambda\Phi D^T PB_u \end{pmatrix}. \tag{29}$$

The two-step solution process is: solve the pseudo-Helmholtz equation

$$\left(M - \lambda^2 \Phi \boldsymbol{D}^{\mathrm{T}} M^{-1} P \boldsymbol{D}\right) \varphi_{\mathrm{G}}^{n+1} = B^{\varphi} - \lambda \Phi \boldsymbol{D}^{\mathrm{T}} P \boldsymbol{B}_u \tag{30}$$

for $\varphi_{\mathrm{G}}^{n+1}$ and then use this solution to solve

$$\boldsymbol{u}_{\mathrm{G}}^{n+1} = M^{-1} P \boldsymbol{B}_{\mathrm{G}}^{u} - \lambda M^{-1} P \boldsymbol{D} \varphi_{\mathrm{G}}^{n+1}. \tag{31}$$

This formulation only requires the solution of an $N_p \times N_p$ matrix system corresponding to Eq. (30). Furthermore, the associated matrix is symmetric positive-definite and thus solvable by conjugate gradient (CG) methods. We use GMRES with a fast projection method (see [6]) to solve this matrix only because this is what we happen to have developed at the moment. The solution of Eq. (31) is completely trivial due to the diagonal nature of M and thus only requires the computation of a $3N_p$ vector.

4. Grid generation on the sphere

The element-based interpolation operator used by SESL permits the use of any type of grid. Defining the order of the SE basis functions automatically defines the order of the interpolation stencil as well. This independence from the grid allows for the use of grids other than the typical latitude–longitude grids used by spectral models. We only use hexahedral grids in this paper but to show the method's independence on the grid we refer the reader to the papers [11,12] in which we used icosahedral grids for the explicit Eulerian method. The choice of using hexahedral grids in this paper is predicated on that fast searching algorithms can be constructed quite easily on this class of grids; we are currently investigating similarly fast search algorithms for the icosahedral grids.

4.1. Hexahedral grids

Hexahedral (a.k.a. cubic gnomonic) grids are constructed by mapping the six faces of a hexahedron onto the surface of a sphere. Each of the six faces of the hexahedron are then subdivided into the desired number of quadrilateral elements. We begin by constructing a spectral element grid on the gnomonic space G. G is defined by the square region $X = [-\pi/4, +\pi/4]^2$ in a 2D Cartesian space (see [28]). This region is divided into the elements and inside each element we construct the LGL grid points. Upon constructing this grid we then map the gnomonic coordinates X to the corresponding spherical coordinates, λ_{G}, via

$$\lambda_{\mathrm{G}} = X, \tag{32}$$

$$\varphi_{\mathrm{G}} = \arcsin\left(\frac{\tan Y}{\sqrt{1 + \tan^2 X + \tan^2 Y}}\right).$$

It should be noted that λ_{G} only gives the spherical coordinates of one of the six faces of the hexahedron. Therefore, we have to rotate this face to the six faces of the hexahedron by the rotation mapping R

$$\lambda = \lambda_{\mathrm{c}} + \arctan\left(\frac{\cos \varphi \sin \lambda_{\mathrm{G}}}{\cos \varphi_{\mathrm{G}} \cos \lambda_{\mathrm{G}} \cos \varphi_{\mathrm{c}} - \sin \varphi_{\mathrm{G}} \sin \varphi_{\mathrm{c}}}\right), \tag{33}$$

$$\varphi = \arcsin(\sin \varphi_{\mathrm{G}} \cos \varphi_{\mathrm{c}} + \cos \varphi_{\mathrm{G}} \cos \lambda_{\mathrm{G}} \sin \varphi_{\mathrm{c}}),$$

where the centroids $(\lambda_{\mathrm{c}}, \varphi_{\mathrm{c}})$ of the six faces are located at $(\lambda_{\mathrm{c}}, \varphi_{\mathrm{c}}) = ([c-1]\pi/2, 0)$ for $c = 1, \ldots, 4$ and $(\lambda_5, \varphi_5) = (0, \pi/2), (\lambda_6, \varphi_6) = (0, -\pi/2)$.

This approach results in the construction of the hexahedral grid with the following properties:

$$N_{\mathrm{p}} = 6(n_{\mathrm{H}} N)^2 + 2,$$

$$N_e = 6(n_H)^2,$$

where N_p and N_e are the number of points and elements comprising the grid. The parameter n_H refers to the number of quadrilateral elements in each direction contained in each of the six faces of the hexahedron and N is the order of the Legendre cardinal functions of each element.

Table 1 gives some grid configurations for the hexahedral grid. Fig. 1 shows sample hexahedral grids for $N = 4, 8$, and 16 with $n_H = 4$.

4.2. Search algorithms

The semi-Lagrangian method requires finding the location of the departure points $\widetilde{\boldsymbol{x}}^{(n,n-1)}$. Unfortunately for quadrilateral elements there is no general way of finding the element which claims a specific departure point; in triangles this is possible through the use of the natural coordinates such as in Giraldo [8,10], and Xiu and Karniadakis [37]. Therefore, the searching algorithms for quadrilateral elements must be specifically tailored to the grid. In the case of hexahedral grids we know that the grid is constructed by mapping each of the six faces of the hexahedron onto a local 2D planar Cartesian space.

Therefore, the first step in determining the location of a departure point is to determine which of the six faces of the hexahedron claims the departure point. We determine this face by checking which face centroid is closest to the departure point. Then we map this face to the rotated-gnomonic space via Eq. (33) and the inverse of Eq. (32). This gives the departure point and the elements contained in this face in terms of the local 2D Cartesian coordinates X. Because on this gnomonic projection each face is a perfect square then the element which claims the departure point can be obtained by using the following integer operations:

$$i_d = 1 + \text{int} \frac{(X_d + \frac{\pi}{4})}{\Delta X}, \tag{34}$$

Table 1
The number of grid points and elements for the hexahedral grid as a function of n_H and N

n_H	N	N_p	N_e
1	4	98	6
1	8	386	6
1	16	1538	6
1	32	6146	6
4	4	1538	96
4	8	6146	96
4	16	24,578	96
4	32	98,306	96

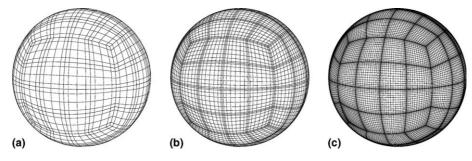Fig. 1. A hexahedral grid for $n_H = 4$ and (a) $N = 4$, (b) $N = 8$, and (c) $N = 16$.

$$j_d = 1 + \text{int} \frac{(Y_d + \frac{\pi}{4})}{\Delta Y},$$

where (i_d, j_d) are the indices of the quadrilateral element which claims the departure point, (X_d, Y_d) is the departure point in gnomonic space, and $\Delta X = \Delta Y = \pi/2n_H$. The next step is to compute the departure point in the local computational space. We require the local computational coordinate ξ_d because the Legendre cardinal functions are only defined in terms of these coordinates.

Using the indices of the element which claims the departure point given by Eq. (34) we can obtain the departure points in computational space as follows:

$$\xi_d = \frac{2(X_d - X_1)}{\Delta X} - 1,$$
$$\eta_d = \frac{2(Y_d - Y_1)}{\Delta Y} - 1, \tag{35}$$

where

$$X_1 = -\frac{\pi}{4} + (i_d - 1)\Delta X,$$
$$Y_1 = -\frac{\pi}{4} + (j_d - 1)\Delta Y. \tag{36}$$

This search algorithm requires a maximum of eight floating point operations and two integer operations regardless of the number of elements in the grid or the order of the polynomial basis N.

5. Results

For the numerical experiments, we use the normalized L_1, L_2, and L_∞ error norms

$$\|\boldsymbol{\varphi}\|_{L_1} = \frac{\int_\Omega |\boldsymbol{\varphi}_{\text{exact}} - \boldsymbol{\varphi}| \, d\Omega}{\int_\Omega |\boldsymbol{\varphi}_{\text{exact}}| \, d\Omega},$$

$$\|\boldsymbol{\varphi}\|_{L_2} = \sqrt{\frac{\int_\Omega (\boldsymbol{\varphi}_{\text{exact}} - \boldsymbol{\varphi})^2 \, d\Omega}{\int_\Omega \boldsymbol{\varphi}_{\text{exact}}^2 \, d\Omega}},$$

$$\|\boldsymbol{\varphi}\|_{L_\infty} = \frac{\max |\boldsymbol{\varphi}_{\text{exact}} - \boldsymbol{\varphi}|}{\max |\boldsymbol{\varphi}_{\text{exact}}|}$$

to judge the accuracy of the numerical scheme. The following two conservation metrics are also used

$$M = \frac{\int_\Omega \varphi d\Omega}{\int_\Omega \varphi_{\text{exact}} \, d\Omega}, \qquad E = \frac{\int_\Omega \varphi(u^2 + v^2 + w^2) + \varphi^2 \, d\Omega}{\int_\Omega \varphi_{\text{exact}}(u_{\text{exact}}^2 + v_{\text{exact}}^2 + w_{\text{exact}}^2) + \varphi_{\text{exact}}^2 \, d\Omega},$$

where M measures the conservation property of the mass and E measures the conservation of the total energy. In addition, we show plots as a function of Courant number for the time-step studies. The Courant number we use is the maximum Courant number of all the LGL cells. To compute the Courant number the elements are decomposed into their LGL grid points and these grid points form a fine grid which we refer to as the LGL cells. The velocities and grid spacings are then defined at the centers of these cells. Using these definitions the Courant number is then defined as

$$\text{Courant number} = \max\left(\frac{C\Delta t}{\Delta s}\right)^e_{\text{LGL}} \quad \forall e \in [1, \ldots, N_e].$$

where

$$C = \begin{cases} U & \text{for Case 1,} \\ U + \sqrt{\varphi} & \text{for Cases } 2, 3, 4, 5, \text{ and } 6, \end{cases}$$

where C is the characteristic speed, $U = \sqrt{\mathbf{u} \cdot \mathbf{u}}$, and $\Delta s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ is the grid spacing.

Eight test cases are used to judge the performance of SESL. Cases 1A, 2, 3, 5 and 6 correspond to the Williamson et al. [36] standard test case suite. Case 4 has been used as a test case for the shallow water equations in [10–12,23,24]. Cases 1A, 1B, and 1C involve the geopotential equation only whereas the remainder of the test cases concern the full shallow water equations. In addition, Cases 1–3 have analytic solutions and are used to determine the accuracy of SESL quantitatively. Cases 4–6, on the other hand, do not have analytic solutions and are thus only used to determine the accuracy of the scheme qualitatively. For many of the test cases in this section we compare SESL with the spectral element semi-implicit method defined in Eqs. (5) and (6) which we refer to as SESI; the value $\theta = 1/2$ is used unless otherwise specified. We also use a fully explicit spectral element leapfrog model for comparisons which we refer to as SEM-LF.

5.1. Case 1A: Advection of a cosine wave with steady-state velocity

Case 1A concerns the solid body rotation of a cosine wave. The velocity field remains unchanged throughout the computation. For this case the velocity field is defined as

$$u = -u_s \sin \lambda - v_s \sin \theta \cos \lambda,$$

$$v = +u_s \cos \lambda - v_s \sin \theta \sin \lambda,$$

$$w = +v_s \cos \theta,$$

where u_s and v_s are the zonal and meridional velocity components in spherical coordinates given in [36]. Results are obtained after one full revolution which corresponds to an integration of 12 days.

Fig. 2 shows that SESL converges algebraically for this test case. We cannot expect exponential convergence because the derivative at the base of the cosine hill is non-smooth.
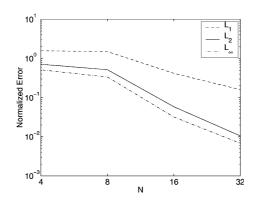

Fig. 2. Case 1A. The normalized φ error norms for SESL as a function of N after 12 days using $n_H = 1$.

Recall that Falcone and Ferretti [7] proved that the error of Lagrangian high order methods is governed by Eq. (17). In this section we show that the error in SESL is governed by Eq. (17); we hope to show this for the atmospheric equations in future work.

Fig. 3 shows the normalized φ L error norm for the $K = 2$ and $K = 4$ trajectory approximations using the $n_H = 1$ and $N = 16$ grid for time-steps yielding Courant numbers from 0.1 to about 50. The $K = 2$ scheme is the second order Runge–Kutta method given by Eq. (15) while the $K = 4$ scheme is the fourth order Runge–Kutta method given by Eq. (16). The error of the $K = 2$ scheme begins to increase at a Courant number of 2.5. This is due to the error being dominated by Δt^K. On the other hand, the error for the $K = 4$ scheme remains level throughout for Courant numbers below 25. Thus by simply increasing the order of K we have been able to sustain the same level of high accuracy for an order of magnitude greater time-step.

5.2. Case 1B: Advection of a cosine wave with time-dependent velocity

This case is similar to Case 1A except that the velocity field varies with time in the following manner:

$$u = (-u_s \sin \lambda - v_s \sin \theta \cos \lambda)\left(1 + \cos \frac{2\pi t}{12 \text{ days}}\right),$$

$$v = (+u_s \cos \lambda - v_s \sin \theta \sin \lambda)\left(1 + \cos \frac{2\pi t}{12 \text{ days}}\right),$$

$$w = (+v_s \cos \theta)\left(1 + \cos \frac{2\pi t}{12 \text{ days}}\right).$$

This case was introduced in [3] and the exact solution can be found in [10]. Fig. 4 shows that SESL converges rapidly even for flow with non-constant velocity.

To show that the Falcone and Ferretti result also holds for flows with time-dependent velocity fields we plot in Fig. 5 the normalized φ L_2 error norm as a function of time-step for $n_H = 1$ and $N = 16$. Once again by increasing the order of K we are able to maintain the same level of high accuracy for a much larger time-step; from a Courant number of 4 for $K = 2$ to a Courant number of 10 for $K = 4$.
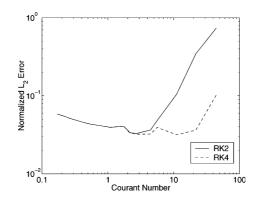

Fig. 3. Case 1A. The normalized φ L_2 error norm for SESL using RK2 (solid) and RK4 (dashed) for the trajectory computation as a function of Courant number after 12 days using $n_H = 1$ and $N = 16$.
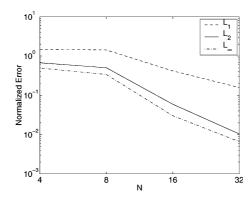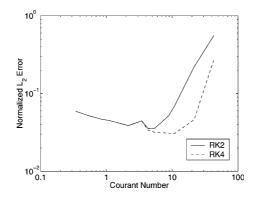
Fig. 4. Case 1B. The normalized φ error norms for SESL as a function of N after 12 days using $n_H = 1$.

Fig. 5. Case 1B. The normalized φ L_2 error norm for SESL using RK2 (solid) and RK4 (dashed) for the trajectory computation as a function of Courant number after 12 days using $n_H = 1$ and $N = 16$.

5.3. Case 1C: Advection of a circular column

In this case we use the steady-state velocity field from Case 1A but the geopotential is now a circular column. Unlike a cosine wave, the circular column represents a discontinuous function and is the 2D analog of a 1D square wave. For flows containing large gradients and/or discontinuities it is well known that high order methods produce oscillations near the discontinuities which contaminate the entire flow field (i.e., Gibbs phenomena). We illustrate this test case to show the ability of the semi-Lagrangian method to handle non-smooth flows. In the atmosphere, severe fronts can form and any method that can handle this test case should be adequate for handling the strongest atmospheric fronts. Fig. 6 shows the algebraic convergence rate of SESL for flows with sharp discontinuities. In Fig. 7 we plot the normalized φ L_2 error norm as a function of time-step for $n_H = 1$ and $N = 16$. The $K = 2$ scheme begins to lose accuracy at a Courant number of 2.5. Meanwhile, the $K = 4$ scheme retains the same accuracy up to Courant numbers of 50.

The results for Case 1 confirm that using large K permits increasing the time-step without increasing the error. However, this comes at a considerable cost especially for the full nonlinear equations. The number of departure point searches per grid point scales like $\mathcal{O}(K)$ and the number of iterations required to achieve convergence in the elliptic solver increases significantly with increasing time-step (due to the growth of the condition number). We found the most efficient Courant numbers for the shallow water equations to be in
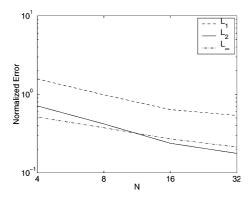
Fig. 6. Case 1C. The normalized φ error norms for SESL as a function of N after 12 days using $n_H = 1$.
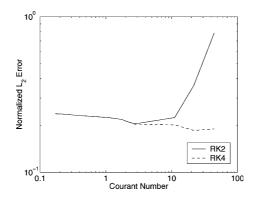

Fig. 7. Case 1C. The normalized φ L_2 error norm for SESL using RK2 (solid) and RK4 (dashed) for the trajectory computation as a function of Courant number after 12 days using $n_H = 1$ and $N = 16$.

the range of 10–20 which can be achieved most efficiently and with sufficient accuracy using $K = 2$. Thus for the remainder of the paper we use SESL with $K = 2$.

To see how the spectral element and the spectral element semi-Lagrangian method handle sharp discontinuities let us compare φ along the Equator for SESI and SESL using the grid $n_H = 4$ and $N = 16$. Figs. 8 and 9 show these slices after one complete revolution of the circular column. Figs. 8(a) and (b) show the

Fig. 8. Case 1C. The mass φ along the Equator after 12 days using $n_H = 4$ and $N = 16$ for: (a) SESI and (b) SESI with filtering.
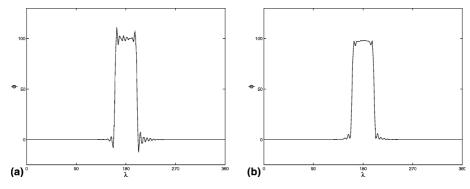
Fig. 9. Case 1C. The mass φ along the Equator after 12 days using $n_H = 4$ and $N = 16$ for: (a) SESL and (b) SESL with FCT.

results for SESI with and without filtering using a Courant number of 0.5. This is the maximum time-step allowed by SESI for this test case. In Fig. 8(a) it is quite evident that the large gradients at the front and lee of the discontinuous wave are causing severe oscillations. In fact, the entire domain is contaminated by Gibbs phenomena. In order to eliminate these aliasing errors high order methods typically employ some form of filtering. In Fig. 8(b) we plot the result for SESI using the Boyd–Vandeven filter (see [2,12]). The filter has eliminated a substantial amount of the numerical noise; however, there are still some undershoots in the front and lee of the wave and some disturbing oscillations at the top of the wave. The most disturbing aspect of this solution is that the correct extrema ($\varphi \in [0, 100]$) are violated. Thus if this equation represented moisture in the atmosphere we would overpredict the amount of rain expected in a certain region of the world and obtain negative rain rates in others.

Figs. 9(a) and (b) show the results for SESL and SESL-FCT using a Courant number of 5 with no filtering. SESL has eliminated almost all of the oscillations although small undershoots and overshoots still occur. Fig. 9(b) shows that SESL with the FCT-type monotone interpolation does not introduce any undershoots or overshoots.

5.4. Case 2: Steady-state nonlinear zonal geostrophic flow

This case is a steady-state solution to the nonlinear shallow water equations. The equations are geostrophically balanced and remain so for the duration of the integration. The velocity field thus remains constant throughout the computation. The geopotential height φ undergoes a solid body rotation but since the initial height field is given as a constant band in the zonal direction and the flow field is purely zonal, then the solution remains the same throughout the time integration. The velocity field is the same as that used in Case 1. Williamson et al. [36] recommend that the error be computed after five days of integration. Fig. 10 illustrates the normalized φ error norms as a function of polynomial order N for SESL. The errors are decreasing quite rapidly.

Fig. 11 compares the normalized φ L_2 error norm of SESI and SESL as a function of the time-step for the grid $n_H = 4$ and $N = 16$. This figure shows that SESL clearly retains its accuracy for significantly large Courant numbers while the accuracy of SESI diminishes quickly and eventually blows up for Courant numbers larger than 2.5. Note that SESL can use a Courant number 10 times larger than SESI (Courant number of 25) and yield the same accuracy as SESI for a Courant number of 2.5.

5.5. Case 3: Steady-state nonlinear zonal geostrophic flow with compact support

This case is another steady-state solution to the nonlinear shallow water equations. The equations remain geostrophically balanced for the duration of the integration. The initial velocity field is zero everywhere except
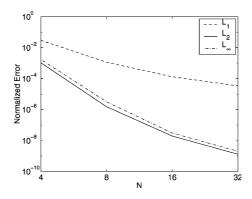
Fig. 10. Case 2. The normalized φ error norms for SESL as a function of N after five days using $n_H = 1$.
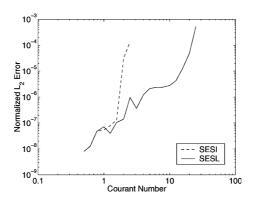

Fig. 11. Case 2. The normalized φ L_2 error norm for SESI (dashed) and SESL (solid) as a function of Courant number after five days using $n_H = 4$ and $N = 16$.

in a very small isolated region. This isolated region, or jet, encapsulates the flow and limits the geopotential height field to remain within a very confined circular region. The results are reported for a five day integration as suggested in [36].

Fig. 12 illustrates the normalized φ error norms as a function of polynomial order N for SESL; SESL converges exponentially for this test case. Fig. 13 compares the normalized φ L_2 error norm of SESI and
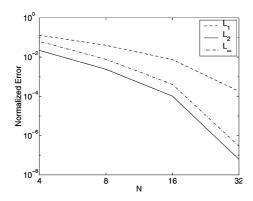

Fig. 12. Case 3. The normalized φ error norms for SESL as a function of N after five days using $n_H = 1$.
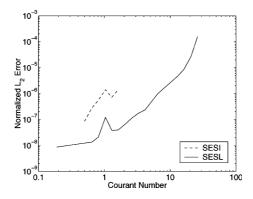
Fig. 13. Case 3. The normalized φ L_2 error norm for SESI (dashed) and SESL (solid) as a function of Courant number after five days using $n_H = 4$ and $N = 16$.

SESL as a function of the time-step for the grid $n_H = 4$ and $N = 16$. This figure shows that SESL clearly retains its accuracy for significantly large Courant numbers while the accuracy of SESI diminishes quickly and eventually blows up for Courant numbers larger than 2. SESL can achieve the same order of accuracy for Courant numbers of about 10.

5.6. Case 4: Dancing high–low waves

This case was introduced in McDonald and Bates [23] and is not an analytic solution to the shallow water equations. The initial geopotential height is comprised of two large waves with the left wave being the low wave and the right wave being the high wave, when viewed from the north pole. The waves rotate clockwise in a swirling dance-like fashion.

Fig. 14 shows the contours of φ, and the zonal and meridional velocities (u_s, v_s) for SEM-LF, SESI, and SESL after 10 days. The Eulerian models (SEM-LF and SESI) are run using their maximum allowable time-steps. The following Courant numbers are used for the three models:

$$\text{Courant number} = \begin{cases} 0.5 & \text{for SEM-LF,} \\ 1.4 & \text{for SESI,} \\ 14.0 & \text{for SESL.} \end{cases}$$

For SESI, the advection terms which are integrated explicitly limits the time-step to only three times that of the purely explicit leapfrog scheme (SEM-LF). SESL uses a time-step 10 times larger than SESI and almost 30 times larger than SEM-LF. The results show almost no differences between the three models. In addition, for 30 day integrations (not shown) : SEM-LF conserves mass exactly, and energy within 0.08%; SESI conserves mass within 0.005% and energy within 0.10%; and SESL conserves mass within 0.02% and energy within 0.75%. After 30 days, all three models yield almost identical results with φ only varying by less than 0.85% between the models.

5.7. Case 5: Zonal flow over an isolated mountain

This case uses the same initial conditions as Case 2 with the addition of a conical mountain at $(\lambda, \theta) = (180, 30)$, where λ is the zonal direction and θ the meridional direction. Due to the zonal flow impinging on the mountain, various wave structures form and propagate throughout the sphere.

Fig. 15 shows the contours of φ, u_s, and v_s for SEM-LF, SESI, and SESL after 15 days which is the maximum time-length suggested in [36]. The following Courant numbers are used for these results:
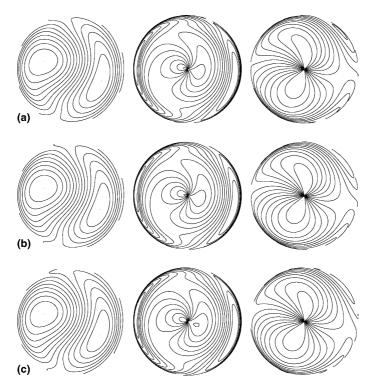
Fig. 14. Case 4. Contours of φ (left), u_s (center), and v_s (right) for: (a) SEM-LF, (b) SESI, and (c) SESL after 10 days using $n_H = 4$ and $N = 16$.

$$\text{Courant number} = \begin{cases} 0.5 & \text{for SEM-LF,} \\ 1.5 & \text{for SESI,} \\ 12.0 & \text{for SESL.} \end{cases}$$

SESI uses a time-step three times larger than SEM-LF, while SESL uses a time-step eight times larger than SESI and 24 times larger than SEM-LF. Once again, the results show almost no differences between the models, especially between the two implicit models (SESI and SESL). SEM-LF shows some differences but the underlying wave structures of the flow are clearly evident in all three models. For 30 day integrations the conservation metrics are as follows: SEM-LF conserves mass exactly, and energy within 0.70%; SESI conserves mass within 0.03% and energy within 0.57%; and SESL conserves mass within 0.03% and energy within 0.55%. After 30 days, the models appear slightly different although φ only varies by less than 0.3% between the models. SESI and SESL yield virtually identical solutions.

5.8. Case 6: Rossby–Haurwitz wave

Although Rossby–Haurwitz waves are not analytic solutions to the shallow water equations, they have become a de facto test case. In a non-divergent barotropic model, the initial geopotential height field undergoes a solid body rotation in a counterclockwise direction when viewed from the north pole.

Although this case does not have an analytic solution, it is well-known that the initial wave structure of the Rossby–Haurwitz wave should remain intact for the duration of the time-integration. Thus we know that for at least 30 days the waves should persist without any loss in shape or mass. For this reason this case represents the best test for judging the performance of the model for long time-integrations.
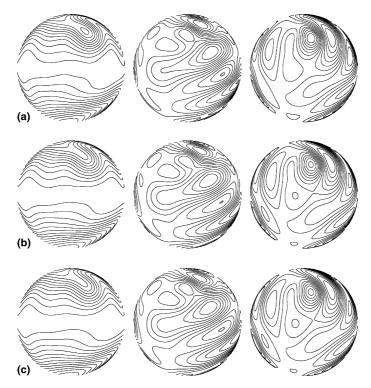
Fig. 15. Case 5. Contours of φ (left), u_s (center), and v_s (right) for: (a) SEM-LF, (b) SESI and (c) SESL after 15 days using $n_H = 4$ and $N = 16$.

Fig. 16 shows the contours of φ, u_s, and v_s for SEM-LF, SESI, and SESL after 14 days. The following Courant numbers are used for these results:

$$\text{Courant number} = \begin{cases} 0.32 & \text{for SEM-LF}, \\ 1.03 & \text{for SESI}, \\ 10.3 & \text{for SESL}. \end{cases}$$

SESI uses a time-step three times larger than SEM-LF, while SESL uses a time-step 10 times larger than SESI and 30 times larger than SEM-LF. Once again, the results show almost no differences between the models. For 30 day integrations the conservation metrics are as follows: SEM-LF conserves mass exactly, and energy within 0.37%; SESI conserves mass within 0.03% and energy within 0.57%; and SESL conserves mass within 0.03% and energy within 0.40%. After 30 days, the results of the three models are virtually indistinguishable with no variation in φ. However, for SESI to run for 30 days required off-centering to a value of $\theta = 0.8$.

The most important result is that all three models predict the same rate of rotation. This is an extremely important result because it confirms that the implicit models are not damping the Rossby waves. Rossby waves are the most important waves for large-scale meteorological processes and are often referred to as planetary waves; Rossby–Haurwitz waves are the Rossby waves on the sphere. Thus any numerical method being considered for constructing atmospheric models must handle these waves properly.
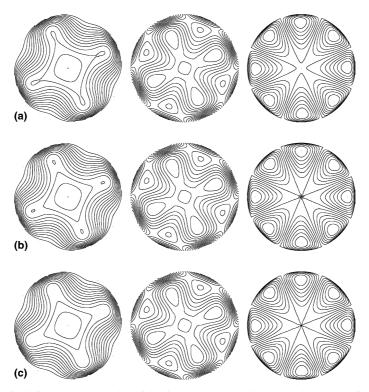
Fig. 16. Case 6. Contours of φ (left), u_s (center), and v_s (right) for: (a) SEM-LF, (b) SESI and (c) SESL after 14 days using $n_H = 4$ and $N = 16$.

5.9. Computational cost

In this section we compare the computational costs of the three models: SEM-LF, SESI, and SESL. We shall use the explicit model, SEM-LF, as the reference with which to compare the Eulerian implicit model, SESI, and the Lagrangian implicit model, SESL.

Fig. 17 illustrates the ratio of computer time required by SESI relative to SEM-LF (dashed line) and SESL relative to SEM-LF (solid line) as a function of N for $n_H = 1$. These time ratios are based on a 24 h integration for Case 5 with the models running the following Courant numbers:

$$\text{Courant number} = \begin{cases} 0.40 & \text{for SEM-LF,} \\ 1.50 & \text{for SESI,} \\ 12.00 & \text{for SESL.} \end{cases}$$

We chose Case 5 because it is the only test case with topography and thereby is the most realistic test case of the entire suite. SEM-LF is anywhere between 1.25 and 4 times more efficient than SESI. In contrast, SEM-LF is only more efficient than SESL for $N > 8$. In addition, SESL is more efficient than SESI for $N \leqslant 12$. This study seems to indicate that SESL should be used with $N \leqslant 12$, which is in agreement with our previous experience of cost versus accuracy (see [11,12]) where we found the range $8 \leqslant N \leqslant 16$ to be optimal for the construction of atmospheric models (see [13]).

In Fig. 18 we plot the time ratio comparing SESI and SESL to SEM-LF as a function of n_H with $N = 8$ and using the same Courant numbers given above. SESI uses a time-step almost four times larger than SEM-LF, while SESL uses a time-step eight times larger than SESI and 32 times larger than SEM-LF.
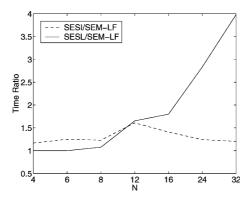
Fig. 17. The ratio of computational cost for SESL/SEM-LF (dashed) and SESL/SESI (solid) for various N for Case 5 using $n_H = 1$. The ratios reflect the amount of computer time required to complete a 24 h time-integration.
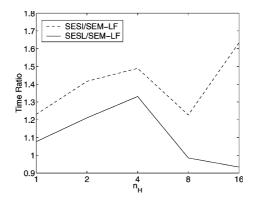

Fig. 18. The ratio of computational cost for SESI/SEM-LF (dashed) and SESL/SEM-LF (solid) for various n_H for Case 5 using $N = 8$. The ratios reflect the amount of computer time required to complete a 24 h time-integration.

Fig. 18 shows that SESL is as high as 70% more efficient than SESI (for $n_H = 16$). SESL is 30% less efficient than SEM-LF for $n_H \leqslant 4$. However, for $n_H > 4$ SESL becomes increasingly more efficient than SEM-LF with increasing n_H. For $n_H = 16$ SESL is 7% more efficient than SEM-LF. This is very good news because n_H is the parameter which controls the grid resolution (for constant N). Thus based on the results outlined in Fig. 18 we can expect SESL to continue to increase its performance with increasing grid resolution.

We certainly expect to be able to increase the performance of SESL beyond the current level. To further increase the performance of SESL requires increasing the performance of SESI. Increasing the performance of SESI will increase the performance of SESL because both models use the same solver and preconditioner. While we did not see a performance increase with SESI over SEM-LF, Thomas et al. [34] have shown that for the shallow water equations Eulerian semi-implicit methods can be constructed to be more efficient than explicit methods on parallel computers. We hope to learn from their experiences and use similar techniques in order to achieve the same kind of performance. In this paper, we present the results for SESL to show proof-of-concept only, and in future work we shall report on developments in the solvers, preconditioners, and search algorithms in the context of a 3D atmospheric model on parallel computers.

6. Conclusions

In this paper we present a high order Lagrangian method for solving advection-dominated fluid flows which may be well-suited for constructing future atmospheric models. The highlight of this method is that it uses the high order basis functions of the spectral element method as the interpolation functions required by the Lagrangian method to interpolate values at the Lagrangian departure points. This element-based compact interpolation renders the spectral element semi-Lagrangian (SESL) method quite general. In addition to high order accuracy and generality, locality is another property inherent to SESL. High order accuracy means that SESL can achieve exponential convergence (for smooth flows). Generality means that SESL can be used on adaptive unstructured grids. Locality means that SESL can be ported to distributed memory computers quite easily. In this paper we have shown the high order accuracy property. We have only shown results on hexahedral grids but it is immediately obvious that any type of quadrilateral grid can be used (generality property). The parallel implementation (locality property) of SESL is the topic of future work.

The results obtained for all eight cases show that SESL is not only high order accurate but possibly very efficient due to the large time-steps that it can use. The results show that SESL yields good convergence for increasing order of the basis functions. In addition, the results show that SESL with an FCT-type interpolation scheme is capable of resolving non-smooth flows such as those encountered in atmospheric fronts. The results obtained by SESL with FCT were shown to be better than those obtained by the spectral element method with filtering.

We compare SESL with an Eulerian semi-implicit method and show that these two methods yield results that are almost identical even when SESL uses time-steps 8–10 times larger. These time-steps are 20–30 times larger than that of the explicit Eulerian model (SEM-LF). The performance comparisons revealed that SESL is 70% more efficient than an Eulerian semi-implicit model but is only 7% more efficient than an Eulerian explicit model. We hope to develop faster solvers, preconditioners, and search algorithms in order to further increase the performance of SESL and we will report our progress in future work.

References

[1] R. Bermejo, A. Staniforth, The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes, Monthly Weather Review 120 (1992) 2622–2632.
[2] J. Boyd, The erfc-log filter and the asymptotics of the Euler and Vandeven sequence accelerations, Houston Journal of Mathematics (1996).
[3] G. Chukapalli, Weather and climate numerical algorithms: a unified approach to an efficient, parallel implementation, Ph.D. Dissertation, University of Toronto, 1997.
[4] J. Côté, A Lagrange multiplier approach for the metric terms of semi-Lagrangian models on the sphere, Quarterly Journal of the Royal Meteorological Society 114 (1988) 1347–1352.

[5] J. Côté, M. Roch, A. Staniforth, L. Fillion, A variable-resolution semi-Lagrangian finite-element global model of the shallow-water equations, Monthly Weather Review 121 (1993) 231–243.

[6] P.F. Fischer, Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides, Computer Methods in Applied Mechanics and Engineering 163 (1998) 193–204.

[7] M. Falcone, R. Ferretti, Convergence analysis for a class of high-order semi-Lagrangian advection schemes, SIAM Journal of Numerical Analysis 35 (1998) 909–940.

[8] F.X. Giraldo, Lagrange–Galerkin methods on spherical geodesic grids, Journal of Computational Physics 136 (1997) 197–213.

[9] F.X. Giraldo, The Lagrange–Galerkin spectral element method on unstructured quadrilateral grids, Journal of Computational Physics 147 (1998) 114–146.

[10] F.X. Giraldo, Lagrange–Galerkin methods on spherical geodesic grids: the shallow water equations, Journal of Computational Physics 160 (2000) 336–368.

[11] F.X. Giraldo, A spectral element shallow water on spherical geodesic grids, International Journal for Numerical Methods in Fluids 35 (2001) 869–901.

[12] F.X. Giraldo, J.S. Hesthaven, T. Warburton, Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations, Journal of Computational Physics 181 (2002) 499–525.

[13] F.X. Giraldo, T.E. Rosmond, A scalable spectral element Eulerian atmospheric model (SEE-AM) for fNWP: dynamical core test, Monthly Weather Review (2003), in press.

[14] J.J. Hack, B.A. Boville, B.P. Briegleb, J.T. Kiehl, P.J. Rasch, D.L. Williamson, Description of the NCAR community climate model (CCM2), NCAR Technical Note NCAR/TN-382+STR, National Center for Atmospheric Research, Climate Modeling Section, P.O. Box 3000, Boulder, CO 80307, 1992.

[15] R. Heikes, D.A. Randall, Numerical integration of the shallow water equations on a twisted icosahedral grid. Part I: Basic design and results of tests, Monthly Weather Review 123 (1995) 1862–1880.

[16] T. Heinze, A. Hense, The shallow water equations on the sphere and their Lagrange–Galerkin solution, Meteorology and Atmospheric Physics 81 (2002) 129–137.

[17] T.F. Hogan, T.E. Rosmond, The description of the navy global operational prediction system's spectral forecast model, Monthly Weather Review 119 (1991) 1786–1815.

[18] M. Iskandarani, D.B. Haidvogel, J.P. Boyd, Staggered spectral element model with application to the oceanic shallow water equations, International Journal for Numerical Methods in Fluids 20 (1995) 393–414.

[19] M. Kwizak, A.J. Robert, A semi-implicit scheme for grid point atmospheric models of the primitive equations, Monthly Weather Review 99 (1971) 32–36.

[20] S.J. Lin, R.B. Rood, An explicit flux-form semi-Lagrangian shallow-water model on the sphere, Quarterly Journal of the Royal Meteorological Society 123 (1997) 2477–2498.

[21] A.V. Malesky, S.J. Thomas, Parallel algorithms for semi-Lagrangian advection, International Journal for Numerical Methods in Fluids 25 (1997) 455–473.

[22] Y. Maday, A.T. Patera, E.M. Ronquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, Journal of Scientific Computing 5 (1990) 263–292.

[23] A. McDonald, J.R. Bates, Semi-Lagrangian integration of a gridpoint shallow water model on the sphere, Monthly Weather Review 117 (1989) 130–137.

[24] B. Neta, F.X. Giraldo, I.M. Navon, Analysis of the Turkel–Zwas scheme for the 2D shallow water equations in spherical coordinates, Journal of Computational Physics 133 (1997) 102–122.

[25] H. Ritchie, Semi-Lagrangian advection on a Gaussian grid, Monthly Weather Review 115 (1987) 608–619.

[26] A. Robert, A semi-Lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations, Journal of the Meteorological Society of Japan 60 (1982) 319–325.

[27] M. Rochas, ARPEGE Documentation, Part 2, Chapter 6 (available from Meteo-France), 1990.

[28] C. Ronchi, R. Iacono, P.S. Paolucci, The "cubed sphere": a new method for the solution of partial differential equations in spherical geometry, Journal of Computational Physics 124 (1996) 93–114.

[29] J.G. Sela, Spectral modeling at the National Meteorological Center, Monthly Weather Review 108 (1980) 1279–1292.

[30] A.J. Simmons, D.M. Burridge, M. Jarraud, C. Girard, W. Wergen, The ECMWF medium-range prediction models development of the numerical formulations and the impact of increased resolution, Meteorology and Atmospheric Physics 40 (1989) 28–60.

[31] P.N. Swarztrauber, D.L. Williamson, J.B. Drake, The Cartesian method for solving partial differential equations in spherical geometry, Dynamics of Atmospheres and Oceans 27 (1997) 679–706.

[32] M. Taylor, J. Tribbia, M. Iskandarani, The spectral element method for the shallow water equations on the sphere, Journal of Computational Physics 130 (1997) 92–108.

[33] C. Temperton, A. Staniforth, An efficient two time-level semi-Lagrangian semi-implicit integration scheme, Quarterly Journal of the Royal Meteorological Society 113 (1987) 1025–1039.

[34] S.J. Thomas, R.D. Loft, J.M. Dennis, Parallel implementation issues: global versus local methods, Computing in Science and Engineering 4 (2002) 26–31.

[35] E. Turkel, G. Zwas, Explicit large time-step schemes for the shallow water equations, in: R. Vichnevetsky, R.S. Stepleman (Eds.), Advances in Computer Methods for Partial Differential Equations, IMACS, Lehigh University, 1979, p. 65.

[36] D.L. Williamson, J.B. Drake, J.J. Hack, R. Jakob, P.N. Swarztrauber, A standard test set for numerical approximations to the shallow water equations in spherical geometry, Journal of Computational Physics 102 (1992) 211–224.

[37] D. Xiu, G.E. Karniadakis, A semi-Lagrangian high-order method for the Navier–Stokes equations, Journal of Computational Physics 172 (2001) 658–684.