2004

A scalable spectral element eulerian atmospheric model (SEE-AM) for NWP: Dynamical core tests

Giraldo, F.X.; Rosmond, T. E.

Monthly Weather Review / Volume 132, Issue 1, 133-153
http://hdl.handle.net/10945/25520

A Scalable Spectral Element Eulerian Atmospheric Model (SEE-AM) for NWP: Dynamical Core Tests

FRANCIS X. GIRALDO AND THOMAS E. ROSMOND

Naval Research Laboratory, Monterey, California

(Manuscript received 26 December 2002, in final form 28 April 2003)

ABSTRACT

A new dynamical core for numerical weather prediction (NWP) based on the spectral element method is presented. This paper represents a departure from previously published work on solving the atmospheric primitive equations in that the horizontal operators are all written, discretized, and solved in 3D Cartesian space. The advantages of using Cartesian space are that the pole singularity that plagues the equations in spherical coordinates disappears; any grid can be used, including latitude–longitude, icosahedral, hexahedral, and adaptive unstructured grids; and the conversion to a semi-Lagrangian formulation is easily achieved. The main advantage of using the spectral element method is that the horizontal operators can be approximated by local high-order elements while scaling efficiently on distributed-memory computers. In order to validate the 3D global atmospheric spectral element model, results are presented for seven test cases: three barotropic tests that confirm the exponential accuracy of the horizontal operators and four baroclinic test cases that validate the full 3D primitive hydrostatic equations. These four baroclinic test cases are the Rossby–Haurwitz wavenumber 4, the Held–Suarez test, and the Jablonowski–Williamson balanced initial state and baroclinic instability tests. Comparisons with four operational NWP and climate models demonstrate that the spectral element model is at least as accurate as spectral transform models while scaling linearly on distributed-memory computers.

1. Introduction

Because of the changing trends in high-performance computers, from large vector machines to distributed-memory architectures, numerical methods that decompose the physical domain into smaller pieces have been receiving significant attention. This new focus on local methods is especially true in the atmospheric sciences, where very large models covering the entire globe are run in timescales ranging from days (in numerical weather prediction) to thousands of years (in climate simulations). Finite-difference and finite-element methods are two such methods that decompose the domain locally, thereby facilitating their implementation on distributed-memory computers. However, one of the biggest disadvantages of these methods is that traditionally they have not been able to compete, in terms of accuracy, with spectral transform methods, which are typically used operationally in numerical weather prediction (NWP) and climate modeling. For example, spectral transform models are used by the National Centers for Environmental Prediction (NCEP; Sela 1980), the European Centre for Medium-Range Weather Forecasts (ECMWF; Simmons et al. 1989), the National Center for Atmospheric Research (NCAR; Hack et al. 1992), and the U.S. Navy (Hogan and Rosmond 1991).

Spectral element methods combine the local domain decomposition property of finite-element methods with the high-order accuracy of spectral transform methods. In other words, spectral elements are as local as finite-element methods and thereby can be used as efficiently on distributed-memory computers while sustaining the same level of accuracy obtained with spectral transform methods. Spectral element methods have been used successfully for the shallow water equations on the sphere (Giraldo 2001; Giraldo et al. 2002; Taylor et al. 1997) and have shown to be promising for ocean and climate modeling (Iskandarani et al. 2002; Loft et al. 2001; Thomas et al. 2002). These methods are essentially high-order finite-element methods where the grid points are chosen to be the Legendre–Gauss–Lobatto (LGL) points. This choice of grid points allows for stable high-order interpolations and results in efficient numerical integration strategies because the LGL points are also used as the quadrature points in the numerical integration required by the weak integral formulation common to all Galerkin methods.

In this paper we extend the 3D Cartesian spectral element method for the spherical shallow water equations introduced in Giraldo (2001) to the full 3D primitive hydrostatic equations governing the motion of the atmosphere. This method represents a radical departure

Corresponding author address: Dr. Francis X. Giraldo, Naval Research Laboratory, 7 Grace Hopper Avenue, Monterey, CA 93943.
E-mail: giraldo@nrlmry.navy.mil

from all previous numerical methods for flow on spherical geometry in that the horizontal operators are written, discretized, and solved completely in 3D Cartesian space. By doing so, we avoid the pole singularity problem associated with the governing equations in spherical coordinates. For a spherical shell, described by the coordinates (λ, φ), of radius a, the divergence of a vector field, $\mathbf{F} = f\hat{\boldsymbol{\lambda}} + g\hat{\boldsymbol{\varphi}}$, is given as

$$\nabla \cdot \mathbf{F} = \frac{1}{a\cos\varphi}\left(\frac{\partial f}{\partial \lambda} + \frac{\partial g\cos\varphi}{\partial \varphi}\right).$$

At the poles (i.e., $\varphi = \pm\pi/2$), this is a source of numerical problems, caused by the specific coordinate formulation rather than the nature of the primitive equations and its solutions. While the use of a local Cartesian coordinate system has been used to overcome these problems in the past (Taylor et al. 1997), we have, guided by the results of previous work (Giraldo 2001; Giraldo et al. 2002), chosen to maintain the Cartesian formulation everywhere.

Therefore, in our formulation the poles are treated as any other point in Cartesian space. Because the numerical method is constructed independently of the grid, this then implies that any grid can be used within this framework including icosahedral, hexahedral, lat–lon, and adaptive unstructured grids. The option of using adaptive unstructured grids will facilitate the coupling of this dynamical core with the Naval Research Laboratory's (NRL) mesoscale model (Hodur 1997). The independence of our numerical methodology from the grid also means that we can change the basis functions from continuous to discontinuous, as we showed in Giraldo et al. (2002), or the elements on which these functions are constructed from quadrilaterals to triangles (Warburton et al. 2000), which then simplifies the construction of adaptive solutions. This independence from the grid is not shared by any of the existing and newly proposed global atmospheric models, including the spectral element model in Taylor et al. (1997), Loft et al. (2001), and Thomas et al. (2002) and the icosahedral model in Randall et al. (2002). In fact, the formulations of all these models are bounded on a specific class of grids. Furthermore, the Cartesian formulation simplifies the addition of semi-Lagrangian schemes. In this paper, we refer to our current model as *Eulerian* in order to distinguish it from the semi-Lagrangian version we are currently testing in other work. In brief, the objective of this paper is to show the feasibility of the Cartesian spectral element formulation for constructing hydrostatic primitive equation models that are as accurate as current spectral transform models and more efficient on distributed-memory computers.

The remainder of the paper is organized as follows. Section 2 contains a description of the governing equations of motion used in numerical weather prediction models, along with a detailed definition of the prognostic and diagnostic variables used in our model. Section 3 contains the description of the numerical approximation of the equations, including, the horizontal, vertical, and temporal discretization methods. In section 4 we describe the tessellation of the sphere into the quadrilateral elements used by the spectral element method to construct the local element matrix operations. This leads directly into section 5, which contains a discussion on the domain decomposition of the sphere and how it translates into the implementation of the model on distributed-memory computers using the Message-Passing Interface. In section 6 we present the results for the seven test cases used to validate our model. Finally, in section 7 we summarize the key findings of this research and discuss the direction of future work.

2. Atmospheric equations

The dynamics of a hydrostatic atmosphere (i.e., dynamical core) are governed by

$$\frac{\partial \pi}{\partial t} + \nabla \cdot (\pi \mathbf{u}) + \frac{\partial}{\partial \sigma}(\pi \dot{\sigma}) = 0, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \dot{\sigma}\frac{\partial \mathbf{u}}{\partial \sigma} = -\frac{2\omega z}{a^2}(\mathbf{x}\times\mathbf{u}) - \nabla\phi - c_p\theta\frac{\partial P}{\partial \pi}\nabla\pi - \mu\mathbf{x}, \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u}\cdot\nabla\theta + \dot{\sigma}\frac{\partial \theta}{\partial \sigma} = 0, \quad \text{and} \tag{3}$$

$$\frac{\partial \phi}{\partial P} = -c_p\theta, \tag{4}$$

where the prognostic variables are the surface pressure, π, the three Cartesian velocity components, $\mathbf{u} = (u, v, w)$, and the potential temperature, θ. The diagnostic variables are the vertical velocity $\dot{\sigma}$, pressure p, and geopotential height ϕ.

In Eq. (2), a and ω are the earth's radius and angular velocity, and μ is a Lagrange multiplier used to constrain the fluid particles to remain on each spherical shell defined by the vertical coordinate σ [we shall describe the role of the Lagrange multiplier in detail in section 3d(1)]. The independent variables in this coordinate system are (x, y, z, σ, t), where the triple (x, y, z) represents the grid point on the sphere defined by the spherical coordinates (λ, φ) and are related by

$$x = a\cos\lambda\cos\varphi,$$
$$y = a\sin\lambda\cos\varphi,$$
$$z = a\sin\varphi.$$

Thus in Eqs. (1), (2), and (3), ∇ is defined as

$$\nabla = \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} \end{pmatrix}$$

at constant σ.

The surface pressure variable, π, in the governing equations is defined as

$$\pi = p_s - p_t,$$

where p_s is the true surface pressure, and p_t is the pressure at the top of the atmosphere. The potential temperature θ is defined as

$$\theta = \frac{T}{P},$$

where T is the temperature, and P is the Exner function. The Exner function is defined as

$$P = \left(\frac{p}{p_o}\right)^{\kappa},$$

where κ is the air constant R_d/c_p, $R_d = 287$ J/kg K is the gas constant of dry air, $c_p = 1004$ J/kg K is the specific heat of dry air at constant pressure, and $p_o = 1000$ hPa is the standard surface pressure. The vertical velocity is defined as $\dot{\sigma} = d\sigma/dt$, where σ is given by

$$\sigma = \frac{p - p_t}{\pi},$$

which yields a value of 0 at the top of the atmosphere and 1 at the surface, and d/dt is the Lagrangian derivative.

3. The numerical scheme

To solve Eqs. (1), (2), and (3), we split the spatial operators into their horizontal and vertical components. Therefore for a given σ value, we discretize the horizontal operators defined on a constant σ spherical shell, as was done in Giraldo (2001) using the spectral element method. The vertical operators are discretized by a mass- and energy-conserving flux-form finite-difference method. We begin with the horizontal discretization of the equations by the spectral element method.

a. Approximating the solution in the horizontal direction

1) BASIS FUNCTIONS AND INTEGRATION

To define the local operators that shall be used to construct the global approximation of the solution, we begin by decomposing the spherical domain $\mathbf{\Omega}$ into N_e nonoverlapping quadrilateral elements such that

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

To perform differentiation and integration operations, we introduce the nonsingular mapping $\mathbf{x} = \Upsilon(\boldsymbol{\xi})$, which defines a transformation from the physical Cartesian coordinate system $\mathbf{x} = (x, y, z)$ defined in $\mathbf{\Omega}_e$ to the reference coordinate system $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ defined in each element, where $(\xi, \eta) \in [-1, +1]^2$ in each element, and $\zeta = 1$ on the surface of the sphere.

Associated with the local mapping, Υ, is the transformation Jacobian, $J = \partial\mathbf{x}/\partial\boldsymbol{\xi}$, and the determinant

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

where $\mathbf{G}$ represents the surface-conforming component of the mapping [see Giraldo (2001) for further details].

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

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

$$\mathbf{q}_N(\mathbf{x}) = \sum_{k=1}^{(N+1)^2} \psi_k(\mathbf{x})\mathbf{q}_N(\mathbf{x}_k), \tag{5}$$

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

$$\psi_k(\mathbf{x}) = h_i[\xi(\mathbf{x})]\, h_j[\eta(\mathbf{x})], \tag{6}$$

where $i, j = 0, \ldots, N$, and $k = 1, \ldots, (N + 1)^2$. In Eq. (6), h are the one-dimensional Lagrange polynomials

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

where $P_N(\xi)$ is the Nth-order Legendre polynomial. For the grid points (ξ_i, η_j), we choose the LGL points, given as the tensor product of the roots of

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

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 that we shall describe shortly.

The choice of the LGL points enables the straightforward approximation of local element integrals, that is,

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

where $|\mathbf{J}|$ represents the local Jacobian for the transformation between $\boldsymbol{\Omega}_e$ and $\mathbf{I}$, 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.

Let us represent the governing equations by the simplified form

$$\frac{\partial \mathbf{q}}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{F} = \mathbf{S}(\mathbf{q}), \tag{7}$$

where $\mathbf{F}$ represents the flux tensor, and $\mathbf{S}$ represents the source terms that we define explicitly in the appendix. Taking the weak form of Eq. (7) with respect to global basis functions Ψ gives

$$\int_{\boldsymbol{\Omega}} \Psi\left[\frac{\partial \mathbf{q}}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{F} - \mathbf{S}(\mathbf{q})\right] d\mathbf{x} = 0, \tag{8}$$

and substituting for $\mathbf{q}$ and $\mathbf{F}$ by the global polynomial approximation similar to Eq. (5) yields the global Galerkin projection of the governing equations

$$\int_{\boldsymbol{\Omega}} \Psi_I\Psi_J\frac{\partial \mathbf{q}_J}{\partial t} + \Psi_I(\boldsymbol{\nabla}\Psi_j\cdot\mathbf{F}_J) - \Psi_I\mathbf{S}(\mathbf{q})\, d\mathbf{x} = 0, \tag{9}$$

where $I, \mathbf{J} = 1, \ldots, N_p$, with N_p representing the number of grid points in the horizontal. We shall return to the discussion on the construction of the global solution in section 3a(2). Because the global operators given in Eq. (9) are never explicitly defined, we begin by defining the local operators that are in fact constructed and then used to construct the action of these global operators on the state vector.

Before discussing the construction of the global solution, let us first describe the local elementwise operators that are used to construct the global solution. Let

$$\mathbf{M}^e_{ij} = \int_{\boldsymbol{\Omega}_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})\, d\mathbf{x} \tag{10}$$

represent the mass matrix, and

$$\mathbf{D}^e_{ij} = \int_{\boldsymbol{\Omega}_e} \psi_i(\mathbf{x})\boldsymbol{\nabla}\psi_j(\mathbf{x})\, d\mathbf{x} \tag{11}$$

the differentiation matrix, where $\psi(\mathbf{x})$ are the local element basis functions given in Eq. (6), $i, j = 1, \ldots, (N+1)^2$ are the number of grid points within each element $\boldsymbol{\Omega}_e$, and $\mathbf{D} = (\mathbf{D}^x, \mathbf{D}^y, \mathbf{D}^z)$ is a vector of matrices corresponding to the three spatial directions. The role of these local element matrices are described below.

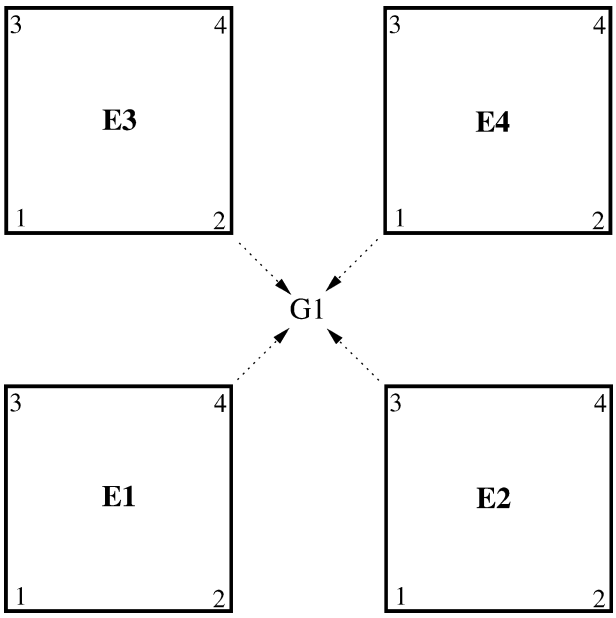

FIG. 1. The contributions of the local element matrices are summed across all elements in order to construct the corresponding global matrices; this is the global assembly procedure. The local element matrix values at the local grid points 4, 3, 2, 1 of elements E1, E2, E3, E4, respectively, are summed to obtain the value of the global matrix at the global grid point G1. These local grid points are in fact the exact same point that in the global indexing is referred to as G1.

2) SATISFYING THE EQUATIONS GLOBALLY

To satisfy the equations globally requires assembling the global solution by virtue of an elementwise construction. This elementwise 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 and is depicted graphically in Fig. 1. In this figure, the local element matrices given in Eqs. (10) and (11) are constructed inside each of the four elements ($E1, \ldots, E4$), and then each element contributes its local approximation to the global sum. For example, the element matrix contributions at the local grid points 4, 3, 2, and 1 of elements E1, E2, E3, and E4 are summed in order to construct the value of the global matrix at the global grid point G1. It should be understood that the local grid points (4, 3, 2, 1) are the same grid points that are claimed by different elements (and possibly processors), and which is G1 in the global indexing. Let us represent this global assembly procedure by the summation operator

$$\bigwedge_{e=1}^{N_e}, \tag{12}$$

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:

$$\mathbf{M} = \bigwedge_{e=1}^{N_e} \mathbf{M}^e \tag{13}$$

for the mass matrix and

$$\mathbf{D} = \bigwedge_{e=1}^{N_e} \mathbf{D}^e \tag{14}$$

for the differentiation matrix.

With these operators defined and by denoting the global grid vector for the surface pressure as $\pi_\mathbf{G}$, the wind velocity as $\mathbf{u}_\mathbf{G}$, the potential temperature as $\theta_\mathbf{G}$, and the geopotential height as $\phi_\mathbf{G}$, we can now write the semidiscrete approximation to Eqs. (1), (2), and (3) as follows:

$$\mathbf{M}\frac{\partial \pi_\mathbf{G}}{\partial t} + \mathbf{D}^\mathrm{T}(\pi_\mathbf{G}\mathbf{u}_\mathbf{G}) + \mathbf{M}\left\{\frac{\partial}{\partial \sigma}(\pi\dot{\sigma})\right\}_\mathbf{G} = 0, \tag{15}$$

$$\mathbf{M}\frac{\partial \mathbf{u}_\mathbf{G}}{\partial t} + \mathbf{u}_\mathbf{G}^\mathrm{T}\mathbf{D}\mathbf{u}_\mathbf{G} + \mathbf{M}\left\{\dot{\sigma}\frac{\partial \mathbf{u}}{\partial \sigma}\right\}_\mathbf{G} = -\mathbf{M}\left[\frac{2\omega z_\mathbf{G}}{a^2}(\mathbf{x}_\mathbf{G} \times \mathbf{u}_\mathbf{G})\right] - \mathbf{D}\phi_\mathbf{G} - c_p\theta_\mathbf{G}\left(\frac{\partial P}{\partial \pi}\right)_\mathbf{G}\mathbf{D}\pi_\mathbf{G}, \tag{16}$$

$$\mathbf{M}\frac{\partial \theta_\mathbf{G}}{\partial t} + \mathbf{u}_\mathbf{G}^\mathrm{T}\mathbf{D}\theta_\mathbf{G} + \mathbf{M}\left\{\dot{\sigma}\frac{\partial \theta}{\partial \sigma}\right\}_\mathbf{G} = 0, \tag{17}$$

where the superscript T denotes the transpose operation, and the terms $\{\}_\mathbf{G}$ denote the global grid vector of the quantities inside the brackets after they have been vertically discretized. 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. Furthermore, the global matrix $\mathbf{D}$ is never actually constructed, but rather only its action on the state vector $\mathbf{q}$ is computed by virtue of the local element matrix and the global assembly procedure. We now address the discretization of the vertical operators.

b. *Approximating the solution in the vertical direction*

The equations are discretized in the vertical direction using a conservative flux-form finite-difference method. This is the same vertical differencing method used in the Navy Operational Global Atmospheric Prediction System (NOGAPS), which is the U.S. Navy's current global atmospheric NWP model (Hogan and Rosmond 1991; Rosmond 2000). NOGAPS is used by the Navy for medium-range weather forecasts worldwide. This model is used to drive the Navy's mesoscale model (Hodur 1997) and is used as a coupled ocean–atmosphere system (Rosmond et al. 2002). NOGAPS uses the spectral transform method in the horizontal, a flux-form finite-difference in the vertical, and a semi-implicit leapfrog scheme in time. The horizontal resolution of NOGAPS recently increased from T159 with 24 vertical levels to T239 with 30 vertical levels (Hogan et al. 2002).

Although we could also discretize the vertical operators in the spectral element Eulerian atmospheric model (SEE-AM) with the spectral element method, we have chosen to use the finite-difference method in order to remain as similar as possible to NOGAPS. This will ensure that any differences in the results are due only to discrepancies in the discrete horizontal operators between the two models. We hope to report on a spectral element vertical discretization in future work.

To simplify the proceeding discussion, let us define the vertical integration of the global gridpoint solution vector $\mathbf{q}_\mathbf{G} = (\pi_\mathbf{G}, \mathbf{u}_\mathbf{G}, \theta_\mathbf{G})$ to be

$$\int_{\sigma_K} \mathbf{q}_\mathbf{G}\, d\sigma = \sum_{L=1}^{K} (\mathbf{q}_\mathbf{G})_L \Delta\sigma_L, \tag{18}$$

where K denotes the number of vertical levels to be integrated across, and

$$\Delta\sigma_L = \sigma_{L+1/2} - \sigma_{L-1/2}$$

represents the thickness of the vertical layer.

To discretize the equations in the vertical direction, we begin by integrating Eq. (15) across all the vertical levels of the atmosphere. Applying no-flux boundaries at the top and bottom levels of the atmosphere results in

$$\frac{\partial \pi_\mathbf{G}}{\partial t} = -\mathbf{M}^{-1}\sum_{K=1}^{N_\mathrm{lev}} \mathbf{D}^\mathrm{T}(\pi_\mathbf{G}\mathbf{u}_\mathbf{G})_K \Delta\sigma_K, \tag{19}$$

where N_lev denotes the total number of vertical levels. Once the surface pressure tendency $\partial\pi_\mathbf{G}/\partial t$ is computed, the vertical velocity $\dot{\sigma}$ at each vertical level is obtained by integrating Eq. (15) from the top of the atmosphere ($\sigma = 0$) to the desired vertical level K, thus giving

$$(\pi_\mathbf{G}\dot{\sigma}_\mathbf{G})_{K+1/2} = -\frac{\partial \pi_\mathbf{G}}{\partial t}\sigma_{K+1/2} - \mathbf{M}^{-1}\sum_{L=1}^{K} \mathbf{D}^\mathrm{T}(\pi_\mathbf{G}\mathbf{u}_\mathbf{G})_K \Delta\sigma_K, \tag{20}$$

where K denotes a full level and $K \pm 1/2$ denote the half levels of this staggered sigma coordinate system. The prognostic variables and ϕ all reside at the full levels, while the diagnostic variables reside at the half levels. The top and bottom of the atmosphere are at the half levels $K = 1/2$ and $K = N_\mathrm{lev} + 1/2$, respectively. Figure 2 illustrates the sigma coordinate system and the location of the prognostic and diagnostic variables.

The vertically differenced terms $\dot{\sigma}(\partial/\partial\sigma)$ of the global

solution vector $\mathbf{q}_\mathbf{G}$ are computed in the following manner:

$$\left(\dot{\sigma}\frac{\partial \mathbf{q}_\mathbf{G}}{\partial \sigma}\right)_K = \dot{\sigma}_{K+1/2}\left[\frac{(\mathbf{q}_\mathbf{G})_{K+1/2} - (\mathbf{q}_\mathbf{G})_K}{\sigma_{K+1/2} - \sigma_K}\right] + \dot{\sigma}_{K-1/2}\left[\frac{(\mathbf{q}_\mathbf{G})_K - (\mathbf{q}_\mathbf{G})_{K-1/2}}{\sigma_K - \sigma_{K-1/2}}\right], \quad (21)$$

where

$$(\mathbf{u}_\mathbf{G})_{K+1/2} = \frac{1}{2}[(\mathbf{u}_\mathbf{G})_{K+1} + (\mathbf{u}_\mathbf{G})_K], \quad \text{and}$$

$$(\theta_\mathbf{G})_{K+1/2} = (\theta_\mathbf{G})_K\left(\frac{P_{K+1/2} - P_K}{P_{K+1} - P_K}\right) + (\theta_\mathbf{G})_{K+1}\left(\frac{P_{K+1} - P_{K+1/2}}{P_{K+1} - P_K}\right).$$

These interpolation stencils are chosen in order to enforce the numerical scheme to conserve energy. Finally, the hydrostatic equation, Eq. (4), is discretized as follows:

$$\phi_K - \phi_{K+1} = c_p(\theta_\mathbf{G})_K(P_{K+1/2} - P_K) + c_p(\theta_\mathbf{G})_{K+1}(P_{K+1} - P_{K+1/2}), \quad (22)$$

where the Exner functions used are

$$P_{K+1/2} = \left(\frac{p_{K+1/2}}{p_o}\right)^\kappa \quad \text{and}$$

$$P_K = \left(\frac{1}{\kappa + 1}\right)\frac{1}{p_o^\kappa}\left(\frac{p_{K+1/2}^{\kappa+1} - p_{K-1/2}^{\kappa+1}}{p_{K+1/2} - p_{K-1/2}}\right). \quad (23)$$

c. Temporal discretization

Discretizing the semidiscrete system, Eqs. (19), (20), (16), and (17), in time by an explicit Eulerian leapfrog method yields

$$\mathbf{M}\frac{\pi_\mathbf{G}^{n+1} - \pi_\mathbf{G}^n}{2\Delta t} = -\sum_{K=1}^{N_{\text{lev}}} \mathbf{D}^\mathrm{T}(\pi_\mathbf{G}\mathbf{u}_\mathbf{G})_K^n \Delta\sigma_K, \quad (24)$$

$$(\pi_\mathbf{G}\dot{\sigma}_\mathbf{G})_{K+1/2}^n = -\frac{\pi_\mathbf{G}^{n+1} - \pi_\mathbf{G}^n}{2\Delta t}\sigma_{K+1/2} - \mathbf{M}^{-1}\sum_{L=1}^{K} \mathbf{D}^\mathrm{T}(\pi_\mathbf{G}\mathbf{u}_\mathbf{G})_L^n \Delta\sigma_L, \quad (25)$$

$$\mathbf{M}\frac{\mathbf{u}_\mathbf{G}^{n+1} - \mathbf{u}_\mathbf{G}^n}{2\Delta t} = -\left\langle \mathbf{u}_\mathbf{G}^\mathrm{T}\mathbf{D}\mathbf{u}_\mathbf{G} + \mathbf{M}\left\{\dot{\sigma}\frac{\partial \mathbf{u}}{\partial \sigma}\right\}_\mathbf{G} + \mathbf{M}\left[\frac{2\omega z_\mathbf{G}}{a^2}(\mathbf{x}_\mathbf{G} \times \mathbf{u}_\mathbf{G})\right]\right\rangle^n - \left[\mathbf{D}\phi_\mathbf{G} + c_p\theta_\mathbf{G}\left(\frac{\partial P}{\partial \pi}\right)_\mathbf{G}\mathbf{D}\pi_\mathbf{G}\right]^n, \quad (26)$$

$$\mathbf{M}\frac{\theta_\mathbf{G}^{n+1} - \theta_\mathbf{G}^n}{2\Delta t} = -\left(\mathbf{u}_\mathbf{G}^\mathrm{T}\mathbf{D}\theta_\mathbf{G} + \mathbf{M}\left\{\dot{\sigma}\frac{\partial \theta}{\partial \sigma}\right\}_\mathbf{G}\right)^n. \quad (27)$$

Because the leapfrog method is an explicit time-differencing scheme, it does require a stringent time step restriction. In order to maintain stability throughout long time integrations (up to 1200 days for the Held–Suarez test case), we use a Courant number of 1/2. We base this time step on the following definition of Courant number:

$$C = \frac{\Lambda_{\max}\Delta t}{\Delta s},$$

where

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$$

is the physical spacing of the grid, and $\Lambda_{\max}$ is the maximum wave speed of the atmospheric equations (see the appendix for a derivation of this characteristic velocity). The physical spacing of the grid Δs scales as

$$\Delta s \propto \frac{1}{N_e N^2},$$

where N_e is the number of spectral elements comprising the grid, and N is the order of the polynomial approximation inside each element.

Because this temporal discretization method produces a computational mode, we apply the time-averaged Asselin filter (Asselin 1972),

$$\overline{\mathbf{q}}_\mathbf{G}^n = \mathbf{q}_\mathbf{G}^n + 0.02(\mathbf{q}_\mathbf{G}^{n+1} - 2\mathbf{q}_\mathbf{G}^n + \overline{\mathbf{q}}_\mathbf{G}^{n-1}),$$

to the global solution vector $\mathbf{q}_\mathbf{G}$ at the end of each time step. Although this is not the most sophisticated time-discretization method available, we have used it in order to keep our model as similar to NOGAPS as possible. NOGAPS uses a semi-implicit time discretization, but this is applied as a correction to the explicit leapfrog scheme. Future research involves the addition of semi-implicit Eulerian and fully implicit semi-Lagrangian methods to the current formulation, which will hopefully allow an increase in the time step by a factor of 10.

d. Lagrange multiplier and the high-pass filter

1) CONSTRAINING THE MOMENTUM

Because we are using Cartesian rather than spherical coordinates, we must carry three momentum equations (in addition to the equation of vertical motion); however, because flow on a spherical shell is really only two-dimensional (at each σ level), the fluid particles are allowed an extra degree of freedom. This degree of freedom will manifest itself in fluid particles flying off the spherical shell. Mollifying this undesirable situation requires constraining the velocity field to be tangential to the sphere. At each grid point we apply the following constraint

$$\mathbf{u}_c^{n+1} = \mathbf{u}_u^{n+1} + \mu\mathbf{x}, \tag{28}$$

where the subscripts c and u denote the constrained and unconstrained horizontal wind velocities, respectively [see Côté (1988) for further details]. For a fluid particle to remain on the spherical shell, the wind velocity must be orthogonal to the position vector of its grid point; that is,

$$\mathbf{u} \cdot \mathbf{x} = 0,$$

which results in the Lagrange multiplier

$$\mu = \frac{\mathbf{x} \cdot \mathbf{u}_u^{n+1}}{a^2}.$$

Equation (28) can now be written as

$$\mathbf{u}_c^{n+1} = \mathcal{P}\mathbf{u}_u^{n+1},$$

where

$$\mathcal{P} = \frac{1}{a^2}\begin{pmatrix} a^2 - x^2 & -xy & -xz \\ -xy & a^2 - y^2 & -yz \\ -xz & -yz & a^2 - z^2 \end{pmatrix} \tag{29}$$

is the projection matrix that constrains vector quantities to be tangential to the sphere. It should be pointed out that using this Cartesian formulation introduces no approximation from the original governing equations in spherical coordinates. Swarztrauber et al. (1997) have shown that the equations in Cartesian coordinates with this type of projection are in fact identical term by term to the equations in spherical coordinates.

2) HIGH-PASS FILTER

Like any high-order method, the spectral element method is susceptible to aliasing errors. In order to prevent these high-frequency waves from contaminating the solution through the introduction of nonphysical oscillations, a high-pass filter is used. We use the filter outlined in Boyd (1998), which is applied as follows. In one dimension, we expand the state vector $\mathbf{q}$ in the ξ direction as follows:

$$\mathbf{q} = L\tilde{\mathbf{q}}(\xi), \tag{30}$$

where

$$\mathbf{L}_{ij} = \begin{bmatrix} P_0(\xi_0) & P_1(\xi_0) & P_2(\xi_0) - P_0(\xi_0) & \cdots & P_i(\xi_0) - P_{i-2}(\xi_0) & \cdots & P_N(\xi_0) - P_{N-2}(\xi_0) \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ P_0(\xi_j) & P_1(\xi_j) & P_2(\xi_j) - P_0(\xi_j) & \cdots & P_i(\xi_j) - P_{i-2}(\xi_j) & \cdots & P_N(\xi_j) - P_{N-2}(\xi_j) \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ P_0(\xi_N) & P_1(\xi_N) & P_2(\xi_N) - P_0(\xi_N) & \cdots & P_i(\xi_N) - P_{i-2}(\xi_N) & \cdots & P_N(\xi_N) - P_{N-2}(\xi_N) \end{bmatrix} \tag{31}$$

is the the Legendre transform matrix, P_i are the Nth-order Legendre polynomials, and $\tilde{\mathbf{q}}$ are the Legendre modal coefficients. To filter the local solution $\mathbf{q}$, we transform them to modal space via the inverse of Eq. (31), apply the filter-weighting diagonal matrix $\mathbf{\Lambda}$ and then transform back to nodal (gridpoint) space. This can be written as

$$\mathbf{q}_\mathbf{F}^1 = \mathbf{F}\mathbf{q}, \tag{32}$$

where

$$\mathbf{F} = \mathbf{L}\mathbf{\Lambda}\mathbf{L}^{-1} \tag{33}$$

is the filter operator that is an $(N + 1) \times (N + 1)$ matrix. The success of the filter hinges on the weighting matrix $\mathbf{\Lambda}$. Following the idea of P. F. Fischer (2002, personal communication), we write

$$\mathbf{\Lambda}_i = \begin{cases} 1 & \text{for } i < i_\mathbf{F} \\ \mu\left(\dfrac{i - i_\mathbf{F}}{N + 1 - i_\mathbf{F}}\right)^2 & \text{for } i \geq i_\mathbf{F} \end{cases} \quad \forall i \in [0, N], \tag{34}$$

where $i = i_\mathbf{F}, \ldots, N$ denotes the modes curtailed by the filter. In this paper we use $i_F = N$, with $\mu = 0.05$, which represents only filtering the highest mode by 5%.

The reason for constructing the Legendre transform as $P_i - P_{i-2}$ for $i = 2, \ldots, N$ and the weighting matrix $\mathbf{\Lambda} = 1$ for $i = 0, 1$ is to avoid affecting the local element boundary values. By using this Legendre polynomial construction, only the first two modes (P_0 and P_1) affect the element boundary values; the remainder of the modes only affect the element interior values. This way we can apply the filter in an element-by-element sense without violating the C^0 continuity condition at the element interfaces (boundaries). This eliminates the need for global assembly of the local element filter matrix. The global assembly operation incurs communication costs on distributed-memory computers, and these costs must be minimized to achieve good performance.

In 2D, the filter is applied as follows:

$$\mathbf{q}_\mathbf{F}^e = \mathbf{F}\mathbf{q}^e\mathbf{F}^\mathrm{T}, \tag{35}$$

where $\mathbf{F}^\mathrm{T}$ is the transpose of $\mathbf{F}$, and $\mathbf{q}^e$ is an $(N + 1) \times (N + 1)$ matrix containing the solution vector of the element $\mathbf{\Omega}_e$.

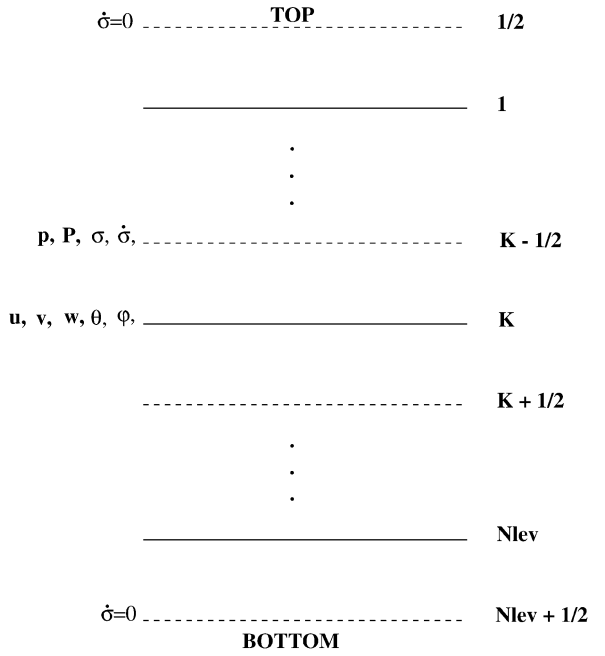

FIG. 2. The equally spaced sigma coordinate system used in the vertical discretization. The prognostic variables reside at the full levels (solid lines) and the diagnostic variables are at the half levels (dashed lines). No-flux boundary conditions are used at the top and bottom of the atmosphere.

4. Grid generation on the sphere

One of the advantages of using Cartesian coordinates is that any grid can be used with our spectral element atmospheric model. Although we can use any grid whatsoever, at the moment the grids must be conforming and quadrilaterally shaped. Using the discontinuous basis functions as in Giraldo et al. (2002) will permit using nonconforming grids, and the spectral element basis in Warburton et al. (2000) will allow the use of triangles, which we reserve for future work. In order to show the grid independence of our model, in this paper we show results on icosahedral and hexahedral grids.

a. Icosahedral grids

Icosahedral grids are constructed by subdividing the 20 triangular faces of the icosahedron by a Lagrange polynomial of order n_I, as described in Giraldo et al. (2002). Prior to mapping these elements onto the sphere, it is convenient to map the triangles onto a gnomonic space. The most unbiased mapping is obtained by mapping about the centroid of the triangles.

Let (λ_c, φ_c) be the centroid of the triangle we wish to map. The gnomonic mapping is then given by

$$x = \frac{a\cos\varphi\sin(\lambda - \lambda_c)}{\sin\varphi_c\sin\varphi + \cos\varphi_c\cos\varphi\cos(\lambda - \lambda_c)},$$

$$y = \frac{a[\cos\varphi_c\sin\varphi - \sin\varphi_c\cos\varphi\cos(\lambda - \lambda_c)]}{\sin\varphi_c\sin\varphi + \cos\varphi_c\cos\varphi\cos(\lambda - \lambda_c)}. \quad (36)$$

To simplify matters a bit, we first apply the rotation mapping R, whereby Eq. (36) becomes

$$x = a\tan\lambda_R, \qquad y = a\tan\varphi_R\sec\lambda_R \quad (37)$$

in the new coordinate system, with the coordinates (λ, φ) located at (0, 0). The rotation mapping, R, is defined as follows:

$$\lambda_R = \arctan\left[\frac{\cos\varphi\sin(\lambda - \lambda_c)}{\sin\varphi_c\sin\varphi + \cos\varphi_c\cos\varphi\cos(\lambda - \lambda_c)}\right],$$

$$\varphi_R = \arcsin[\cos\varphi_c\sin\varphi - \sin\varphi_c\cos\varphi\cos(\lambda - \lambda_c)].$$

Once the triangular icosahedral grid is constructed, we subdivide each triangular element into three quadrilateral elements. Upon dividing the triangles into quadrilaterals, one can construct the higher-order LGL grid points inside each element, resulting in a quadrilateral grid with the following properties:

$$N_p = 60(n_I N)^2 + 2, \quad (38)$$

$$N_e = 60(n_I)^2, \quad (39)$$

where N_p and N_e denote the total number of grid points and elements composing the icosahedral grid, and N is the polynomial order of the elements. Examples of corresponding grids for $n_I = 2$, $N = 8$ and $n_I = 4$, $N = 8$ are illustrated in Fig. 3.

b. Hexahedral grids

Hexahedral grids are constructed by subdividing the six faces of a hexahedron into the desired number of quadrilateral elements and then mapping these onto the sphere. We begin by constructing a spectral element grid on the gnomonic space **G**, which is defined by the square region $\boldsymbol{\xi}_\mathbf{G} = [-\pi/4, +\pi/4]^2$ in a 2D Cartesian space (Komatitsch and Tromp 2002; Ronchi et al. 1996). 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 to the corresponding spherical coordinates, $\boldsymbol{\lambda}_\mathbf{G}$, via

$$\lambda_\mathbf{G} = \xi_\mathbf{G},$$

$$\varphi_\mathbf{G} = \arcsin\left(\frac{\tan\eta_\mathbf{G}}{\sqrt{1 + \tan^2\xi_\mathbf{G} + \tan^2\eta_\mathbf{G}}}\right).$$

It should be noted that $\boldsymbol{\lambda}_\mathbf{G}$ 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

$$\lambda = \lambda_c + \arctan\left(\frac{\cos\varphi_\mathbf{G}\sin\lambda_\mathbf{G}}{\cos\varphi_\mathbf{G}\cos\lambda_\mathbf{G}\cos\varphi_c - \sin\varphi_\mathbf{G}\sin\varphi_c}\right),$$

$$\varphi = \arcsin(\sin\varphi_\mathbf{G}\cos\varphi_c + \cos\varphi_\mathbf{G}\cos\lambda_\mathbf{G}\sin\varphi_c), \quad (40)$$

where the centroids, (λ_c, φ_c), of the six faces are located at $(\lambda_c, \varphi_c) = ([c - 1]\pi/2, 0)$ for $c = 1, \ldots, 4$, and

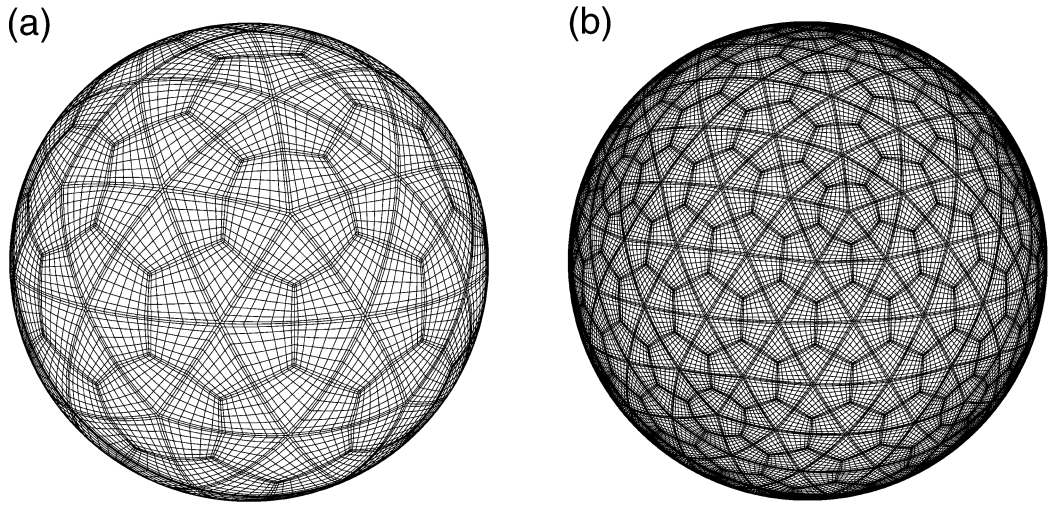

FIG. 3. The icosahedral grid for (a) $n_I = 2$, $N = 8$ and (b) $n_I = 4$, $N = 8$.

$(\lambda_5, \varphi_5) = (0, \pi/2)$, $(\lambda_6, \varphi_6) = (0, -\pi/2)$. This approach results in the construction of the hexahedral grid with the following properties:

$$N_p = 6(n_H N)^2 + 2, \tag{41}$$

$$N_e = 6(n_H)^2, \tag{42}$$

where N_p and N_e denote the number of grid points and elements composing the grid. The parameter n_H refers to the number of quadrilateral elements in each direction (ξ_G and η_G) contained in each of the six faces of the hexahedron, and N is the polynomial order of the elements. Figure 4 shows the grids $n_H = 4$, $N = 8$ and $n_H = 8$, $N = 8$. The hexahedral resolution H, where

$$H = n_H N, \tag{43}$$

has approximately the same number of grid points as the spectral triangular truncation T on a Gaussian latlon grid; these two different grid resolutions are related by the expression

$$H \sim T + 1. \tag{44}$$

To derive this relation requires a few definitions. Let the number of grid points in a spectral model be given by

$$N_p = N_{\text{lon}} \cdot N_{\text{lat}} \equiv 2N_{\text{lat}}^2,$$

where N_{lon} and N_{lat} denote the number of points in the longitudinal and meridional directions in a Gaussian grid. In addition, let the spectral triangular truncation be given by $T = (2/3)N_{\text{lat}} - 1$, which results in the number of grid points being

$$N_p = \frac{9}{2}(T + 1)^2.$$

Equating this expression to Eq. (41) yields $H \sim (\sqrt{3}/2)$

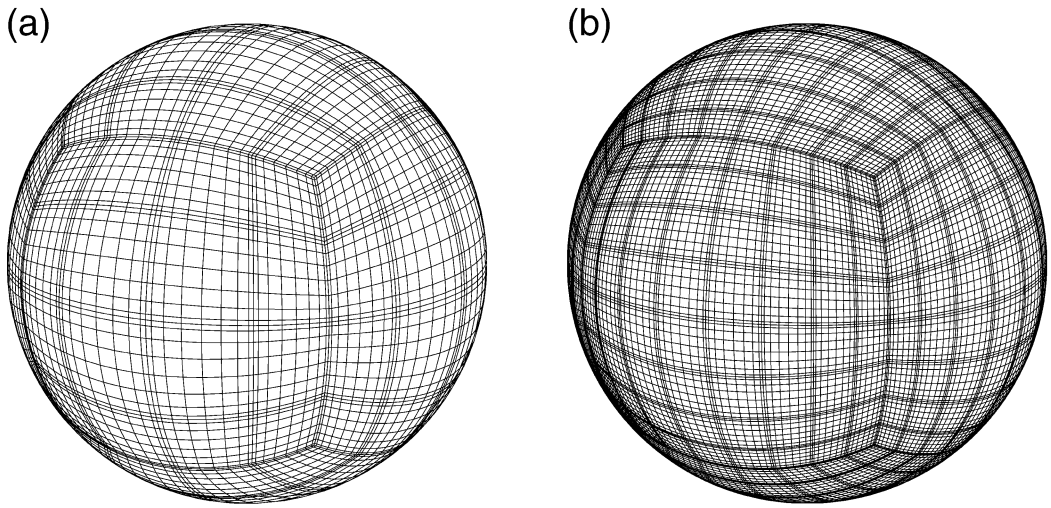

FIG. 4. The hexahedral grid for (a) $n_H = 4$, $N = 8$ and (b) $n_H = 8$, $N = 8$.

($T + 1$), which we approximate by Eq. (44). Of course, there are other ways of relating the resolution of different grids but certainly using the number of grid points is the most logical approach. With this relationship between spectral triangular truncation, *T*, and hexahedral resolution, *H*, we can obtain the equivalent hexahedral resolution for the NOGAPS operational resolution of *T*239 with $n_H = 30$ and $N = 8$ (yielding *H*240).

5. Parallel implementation

In this section we discuss the issues concerning the implementation of SEE-AM on distributed-memory computing platforms. Let us begin by describing the domain decomposition strategy.

a. Domain decomposition

Because our implementation of the spectral element method is completely independent from the grid, we are free to choose any grid; however, in order to simplify the discussion of our model we describe the domain decomposition as it pertains to hexahedral grids only. To construct a hexahedral grid, we map the six faces of a hexahedron onto a sphere. Therefore, the logical partitioning of the domain is the decomposition of the spherical domain into the six faces of the hexahedron. In keeping with this simple decomposition strategy we then further subdivide the domain into perfectly square regions. In other words, the following partitions are possible

$$N_{\text{proc}} = 6(n_P)^2, \tag{45}$$

where N_{proc} is the total number of processors and n_P represents the partitioning of processors in each of the two Cartesian directions on each of the six faces of the hexahedron. In addition, we require the following constraint on n_P:

$$n_P \leq n_H,$$

which states that the number of processors cannot exceed the number of spectral elements.

This is by no means the only possible partitioning strategy. A more sophisticated approach is to use the Metis graph partitioning software (Karypis and Kumar 1998) or the space-filling-curves strategy presented in Dennis (2003). We merely present this ad hoc hexahedral grid domain decomposition strategy as a proof of concept that SEE-AM performs efficiently on distributed-memory computers. In the future, we plan on implementing the space-filling-curves strategy, which will simplify the domain partitioning and allow us to use any grid whatsoever.

b. Communication

From the description of the global assembly procedure given in section 3a(2), it should be apparent that the communication in the spectral element method results from the summation of the local element matrices to construct the global matrices. In order to better understand the communication that takes place across neighboring processors, let us look at only one face of the hexahedral grid. For the sake of argument, let us assume that $n_P = 3$, meaning that there are nine processors per face. This situation is illustrated in Fig. 5. Furthermore, let us assume that $n_H = 9$; that is, there are 81 elements per face. From the figure it is evident that there will then be nine elements inside each processor denoted by *N*1, . . . , *N*9 for the neighbors, and *P*1, . . . , *P*9 for the on-processor elements. The processors are denoted by the thick lines, while the thin lines represent the spectral elements. However, in order to keep the discussion as general as possible, we shall not define the order of the polynomial, *N*, inside each element.

N7	N8	N9	N7	N8	N9	N7	N8	N9
N4	**NBR7**	N6	N4	**NBR6**	N6	N4	**NBR5**	N6
N1	N2	N3	N1	N2	N3	N1	N2	N3
N7	N8	N9	P7	P8	P9	N7	N8	N9
N4	**NBR8**	N6	P4	**PROC**	P6	N4	**NBR4**	N6
N1	N2	N3	P1	P2	P3	N1	N2	N3
N7	N8	N9	N7	N8	N9	N7	N8	N9
N4	**NBR1**	N6	N4	**NBR2**	N6	N4	**NBR3**	N6
N1	N2	N3	N1	N2	N3	N1	N2	N3

FIG. 5. The communication stencil required by the elements *P*1, . . . , *P*9 in processor PROC. The dashed box represents the perimeter values that each processor sends to its neighbors, and *N*1, . . . , *N*9 represent the elements of the eight neighboring processors (NBR).

Because of the C^0 continuity condition required by the spectral element method, the four corner points of processor PROC in Fig. 5 are each shared by four processors: PROC plus three neighboring processors. Similarly, edge points are shared by two processors. This continuity condition is satisfied by the global assembly procedure that is executed across processors as follows. In processor PROC the on-processor right-hand-side (rhs) vectors of Eqs. (24), (26), and (27) are constructed. For the interior grid points of PROC, these rhs vectors represent the globally assembled rhs vectors of the equations; however, for the boundary grid points they are only a portion of the global rhs vectors. Thus, processor PROC requires the boundary gridpoint rhs vectors from

processors NBH1, . . . , NBH8 to complete the global assembly of its rhs vectors. Once these global rhs vectors are constructed, each processor can then solve independently for the global solution via Eqs. (24), (26), and (27). Note that Eq. (25) is solved on-processor without communication after Eq. (24) is solved.

As an example of how the global solutions of the processor corner points are obtained, let us describe the procedure for the bottom-left grid point of PROC in Fig. 5. To construct the global solution at this grid point requires knowing the contribution of the on-processor–assembled rhs vectors from the neighbors NBR1, NBR2, and NBR8. Therefore, each processor computes its on-processor–assembled rhs vectors locally and then sends its perimeter values to its eight neighboring processors. This results in a message approximately of the size

$$20(N + 1)N_{\text{lev}}\alpha, \tag{46}$$

where the ratio

$$\alpha = \frac{n_H}{n_P} \tag{47}$$

represents the number of elements per processor in each of the two Cartesian directions on each of the six faces of the hexahedron. Using Eqs. (42) and (45), we can rewrite Eq. (47) as

$$\alpha = \sqrt{\frac{N_e}{N_{\text{proc}}}}. \tag{48}$$

Equation (46) illustrates that the message size scales linearly with N, N_{lev}, and α; the constant in Eq. (46) arises from each processor having four edges and the primitive equations, in Cartesian coordinates, having five prognostic variables. Thus at every time step the perimeter values of the full 3D rhs vectors of each processor are sent to its neighbors.

The communication described above is exact for all processors that do not contain one of the eight corner points of the hexahedron. In Fig. 5, processor NBH5 only has seven neighboring processors to communicate with because of the topology of the hexahedron. In addition, for the special case $n_H = 1$, each processor has only four neighbors.

c. Performance

1) Model scalability

One of the main advantages of using spectral element methods over spectral transform methods is that for an equivalent resolution the spectral element method allows the use of far more processors. As an example let us compare SEE-AM with NOGAPS, which uses the spectral transform method in the horizontal. The most efficient decomposition for NOGAPS is through a 1D decomposition along latitude rings. It should be noted, however, that in general 2D decompositions are more efficient for spectral transform models, as shown in Foster et al. (1992). We shall only be comparing the operational version of NOGAPS with SEE-AM, and it should be understood that the discussion in this section does not necessarily extend to all spectral transform models, but it should provide some reasonable estimates.

Using a 1D domain decomposition, the maximum number of processors that NOGAPS can use is

$$N^T_{\text{proc}} = N_{\text{lat}} \approx \frac{3}{2}T, \tag{49}$$

where N_{lat} denotes the number of latitude rings, and T the resolution of the spectral triangular truncation. In contrast, the maximum number of processors that SEE-AM can use is

$$N^H_{\text{proc}} = N_e \equiv \frac{6}{N^2}H^2, \tag{50}$$

where we have used Eqs. (42) and (43) in order to simplify the expression and write it as a function of hexahedral resolution, H. In other words, SEE-AM can use as many processors as there are elements. Thus for fixed N the number of processors allowed by SEE-AM increases quadratically with resolution, H, while only linearly for NOGAPS. At the operational T239 resolution NOGAPS can use 360 processors, whereas SEE-AM (assuming $n_H = 30$ and $N = 8$) can use 5400 processors; a 15-fold increase in the number of processors. Equation (50) shows that if we wish to further increase the number of processors with SEE-AM, we simply increase n_H while decreasing N accordingly in order to maintain the horizontal resolution. Therefore we could use $n_H = 60$ and $N = 4$ for a total of 21 600 processors; a 60-fold increase in the number of processors. However, decreasing N will impact the solution accuracy and the issue of efficiency versus accuracy must be carefully weighed. The point here is that the spectral element method offers this flexibility to increase either the accuracy or efficiency—a luxury not shared by the spectral transform method. We leave the detailed discussion of the issue of efficiency versus accuracy to future work, but at this point we anticipate using $N = 8$; even with $N = 8$, SEE-AM accommodates many more processors than NOGAPS.

2) Comparison with NOGAPS

The rate of floating point operations for SEE-AM increases as

$$O(n_H^2 N^3 N_{\text{lev}}), \tag{51}$$

which corresponds to the construction of derivatives and the application of the filter in the spectral element method. In contrast, the rate of floating point operations for NOGAPS increases as

$$O(N^3_{\text{lat}} N_{\text{lev}}), \tag{52}$$

which corresponds to the computation of the Legendre transform involving the meridional direction. The Legendre transform has been the bottleneck of the spectral transform method because of the lack of a fast transform, such as the fast Fourier transform (FFT) of Cooley and Tukey (1965), for the zonal direction.

In order to compare the rates of operations between SEE-AM and NOGAPS, we rewrite Eqs. (51) and (52) as functions of resolution, H and T, as follows:

$$O(H^2 N_{\text{lev}} N), \tag{53}$$

$$O(T^3 N_{\text{lev}}), \tag{54}$$

for SEE-AM and NOGAPS, respectively. Thus the cost increases cubically with resolution, T, for NOGAPS, while it only increases quadratically for SEE-AM, H, provided that N remains fixed.

To understand why the spectral transform method becomes increasingly more expensive than the spectral element method requires revisiting Eqs. (51) and (52). For the spectral element method, the horizontal resolution is governed by n_H and N. Recall that n_H governs the number of spectral elements, while N is the order of the polynomials. This last term is analogous to N_{lat} in Eq. (54) for the spectral transform method. To increase the horizontal resolution of a spectral transform model requires increasing N_{lat}, which increases the cost by its cube; there is no way around this. In contrast, with the spectral element method one has the choice of increasing either the number of elements or the order of the polynomial. Since the cost increases cubically with N and only quadratically with n_H, then it makes sense to keep the polynomial order fixed and increase the number of elements to obtain higher resolutions. This flexibility is due to the h–p nature of the spectral element method. By keeping n_H fixed (say $n_H = 1$) and increasing N, we reach the spectral transform limit of the spectral element method (known as the p-type method). On the other hand, by keeping N fixed (say $N = 1$) and increasing n_H, we obtain the linear finite-element limit of the spectral element method (known as the h-type method). On serial computers the optimal strategy for selecting n_H and N is usually to pick N in the range [8, 16] and increase n_H to yield the desired resolution.

The rates reported in Eqs. (53) and (54) are based on a per time step basis; however, in practice the spectral transform method admits a much larger time step than the spectral element method. This is due to the time step stability limits being different for the two models. For NOGAPS the time step must be decreased linearly with horizontal resolution T,

$$\Delta t^T \propto \frac{1}{T},$$

while for SEE-AM the time step scales as

$$\Delta t^H \propto \frac{1}{n_H N^2} \equiv \frac{1}{N}\frac{1}{H}.$$

Therefore, if we use the p version of the spectral element method, then the time step must be decreased quadratically with resolution; however, if we keep N fixed then we can achieve a linear decrease of the time step with resolution that will allow spectral element models to compete with spectral transform models. Let us now compare how the time steps differ for NOGAPS and SEE-AM.

The explicit leapfrog version of NOGAPS admits a time step 3.5 times larger than the explicit leapfrog version of SEE-AM, and the semi-implicit version of NOGAPS admits a time step 15 times larger than SEE-AM. Thus for SEE-AM to be competitive with NOGAPS, it must be far more efficient on a per grid point and per time step basis. In Taylor et al. (1997), they show that the cost per grid point of the spectral transform method is

$$4.25N_{\text{lat}} + 107 \log N_{\text{lat}},$$

and that of the spectral element is 192 for $N = 8$. These cost estimates are computed on a per processor basis. Based on these cost estimates, the explicit SEE-AM model will outperform the explicit NOGAPS beyond resolutions of T108, and the semi-implicit NOGAPS beyond T406. Currently, some spectral transform models are running beyond T406, such as the ECMWF model, which uses T511, and NOGAPS is expected to be at or beyond this resolution in the near future. A semi-implicit implementation of SEE-AM will outperform the semi-implicit NOGAPS beyond a resolution of T185, which is well below the current operational resolution of T239. This resolution is obtained by comparing the serial versions of our semi-implicit implementation of SEE-AM, in which we use a conservative estimate of a factor of 2 increase in performance over the explicit SEE-AM; however, this does not necessarily guarantee that we will achieve this gain in the parallel version, but it does provide a good estimate. Nonetheless, much work has been done regarding this issue, and we hope to benefit from the volume of work in the literature on this topic, most notably, the parallel elliptic solvers of Tufo and Fischer (1999) for the Navier–Stokes equations and the work by Loft et al. (2001) and Thomas et al. (2002) for the multilevel shallow water equations.

In Fig. 6 we show a performance comparison on an IBM SP3 for the explicit NOGAPS and SEE-AM models; both models use a resolution of T159 with $N_{\text{lev}} = 24$ vertical levels and the explicit SEE-AM time step, $\Delta t = 35$ s. The results of this figure are summarized as follows. First, SEE-AM (spectral element) is much faster than NOGAPS (spectral transform) on a per time step and per processor basis. For an equal number of processors, say 150, SEE-AM is more than 2 times faster than NOGAPS. Second, SEE-AM can use many more processors than NOGAPS. At a resolution of T159, NOGAPS can only use 150 processors effectively; note that beyond 150 processors the performance of NOGAPS decreases. In contrast, SEE-AM is able to use 600 pro-

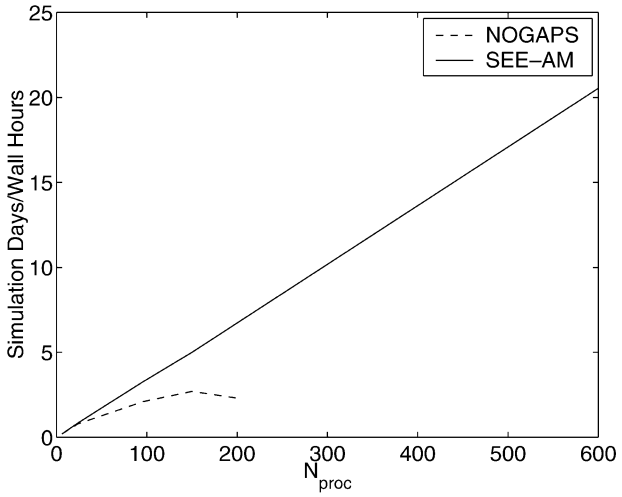

FIG. 6. The simulation days per wall clock time as a function of processors, N_{proc}, for the explicit NOGAPS and SEE-AM, for T159 and $N_{\text{lev}} = 24$. Both models use the maximum allowable time step for SEE-AM.

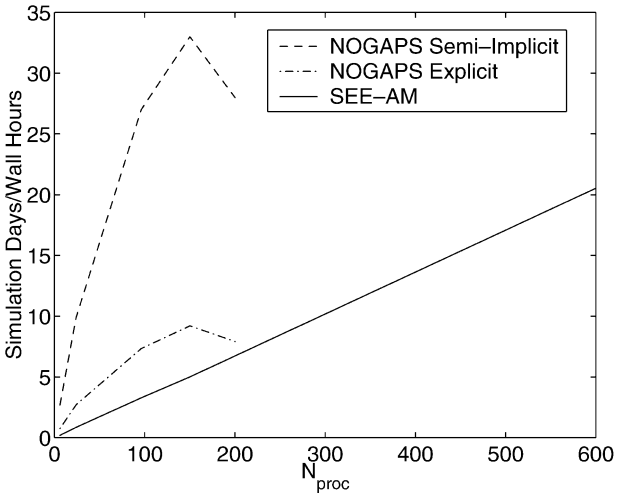

FIG. 7. The simulation days per wall clock time as a function of processors, N_{proc}, for the semi-implicit NOGAPS, explicit NOGAPS, and SEE-AM, for T159 and $N_{\text{lev}} = 24$. Each model uses its maximum allowable time step.

cessors effectively, and it can accommodate up to 2400 processors. Finally, the results presented in Fig. 6 are very reassuring because they show that SEE-AM scales linearly for increasing processor number. At 600 processors, both the model and the communication network did not suffer any severe penalties. Although we should not expect to get the same type of performance on 2400 processors, it is exciting to anticipate that it may be possible.

The results in Fig. 6 clearly show that SEE-AM is far more efficient than NOGAPS on a per time step basis and per processor basis. However, NOGAPS can use a time step much larger than SEE-AM. In Fig. 7 we plot the results for the semi-implicit NOGAPS, explicit NOGAPS, and explicit SEE-AM for a T159 horizontal resolution with 24 vertical levels. The results shown here are plotted using the maximum time step that each model allows, and they are $\Delta t = 540$, 120, and 35 s for the semi-implicit NOGAPS, explicit NOGAPS, and SEE-AM, respectively.

The results of this study are summarized as follows. For small processor numbers NOGAPS outperforms SEE-AM. However, for processor numbers greater than 250 SEE-AM outperforms the explicit NOGAPS. If the linear scalability of SEE-AM were to hold for increasing processor numbers, we would expect SEE-AM to outperform the semi-implicit NOGAPS at around 900 processors. However, if we could double the time step by introducing a semi-implicit implementation, then SEE-AM would outperform the semi-implicit NOGAPS beyond 500 processors. It is possible to further increase the efficiency of SEE-AM because this model has not yet been fully optimized. The results presented in this section should not be taken as the optimal performance of spectral element models but merely as a first attempt at constructing fast and efficient NWP models. We hope to benefit from the work of Thomas et al. (2002) and Loft et al. (2001) on the optimization of spectral element models.

6. Results

In this section we validate SEE-AM using barotropic and baroclinic test cases. The barotropic cases are used to confirm the exponential accuracy of the discrete spectral element horizontal operators. The baroclinic cases are used to validate the full 3D primitive hydrostatic equation model. In order to judge the accuracy of the model, we plot normalized L_2 error norms defined as follows:

$$\|q_{\mathbf{G}}\|_{L_2} = \sqrt{\frac{\int_{\Omega} (q_{\text{exact}} - q_{\mathbf{G}})^2 \, d\mathbf{x}}{\int_{\Omega} q_{\text{exact}}^2 \, d\mathbf{x}}}, \tag{55}$$

where $q_{\mathbf{G}}$ is the computed solution vector, q_{exact} is the exact solution, and the norm is computed as a broken norm. All the results are computed using 64-bit arithmetic precision.

a. Barotropic tests

To validate the spectral element discrete operators, we run the model using the shallow water tests 1, 2, and 3 in Williamson et al. (1992). Because these tests admit exact solutions, we are able to plot normalized geopotential, ϕ, L_2 error norms. Figure 8 shows that SEE-AM achieves the expected exponential convergence regardless of whether the icosahedral or hexahedral grid is used. Case 1 will not yield exponential

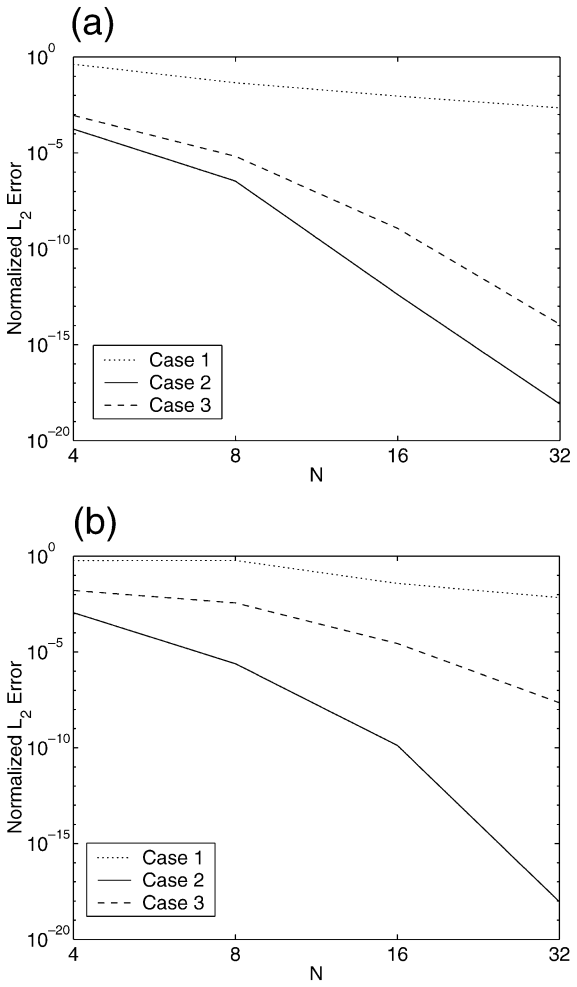

FIG. 8. Barotropic cases 1, 2, and 3: The geopotential, ϕ, normalized L_2 error as a function of polynomial order, N, for the Williamson et al. (1992) shallow water tests 1, 2, and 3 using (a) icosahedral and (b) hexahedral grids. The $n_I = 1$ icosahedral grid and $n_H = 1$ hexahedral grid are used, and the results are reported for 12, 5, and 5 days for cases 1, 2, and 3, respectively.

convergence because of the nonsmooth nature of the derivatives at the base of the cosine bell.

In Figs. 9, 10, and 11, we compare SEE-AM on hexahedral grids with various other models. These models are the Jakob-Chien et al. (1995) (spectral transform), Heikes and Randall (1995) (finite difference), Taylor et al. (1997) (spectral element), and the Tomita et al. (2001) (finite difference) models. These models were chosen because they are representative of the current methods being explored for future NWP and climate models. The Jakob-Chien et al. model uses the same horizontal operators used in the NOGAPS (Hogan and Rosmond 1991), ECMWF (Simmons et al. 1989), NCEP (Sela 1980), and NCAR (Hack et al. 1992) spectral transform models. The Heikes and Randall model uses the same horizontal operators as those in the Colorado State University geodesic grid climate model (Randall

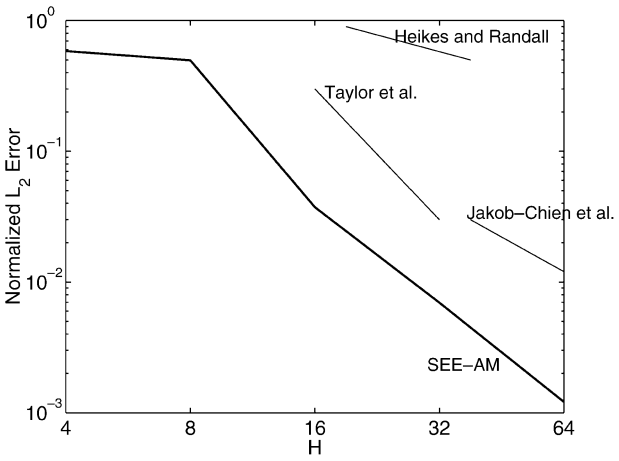

FIG. 9. Barotropic case 1: The geopotential, ϕ, normalized L_2 error as a function of horizontal resolution, H, for the Williamson et al. (1992) shallow water case 1 after 12 days for SEE-AM (thick line), the Jakob-Chien et al. (1995) (spectral transform), the Heikes and Randall (1995) (finite difference), and the Taylor et al. (1997) (spectral element) models. The SEE-AM model uses the $n_H = 1$ ($N_e = 6$) grid.

et al. 2002; Ringler et al. 2000) and is similar to the German Weather Service model (GME; Majewski et al. 2002). The Tomita et al. model uses the horizontal operators expected to be used in the Japanese Earth Simulator project. Finally, the Taylor et al. model uses similar horizontal operators to the NCAR spectral element dynamical core (Loft et al. 2001; Thomas et al. 2002) and the Rutgers University spectral element ocean model (Iskandarani et al. 2002). Although it is very difficult to compare different models, we have chosen to use equivalent horizontal resolutions based on the number of grid points. Thus for gridpoint models we compute

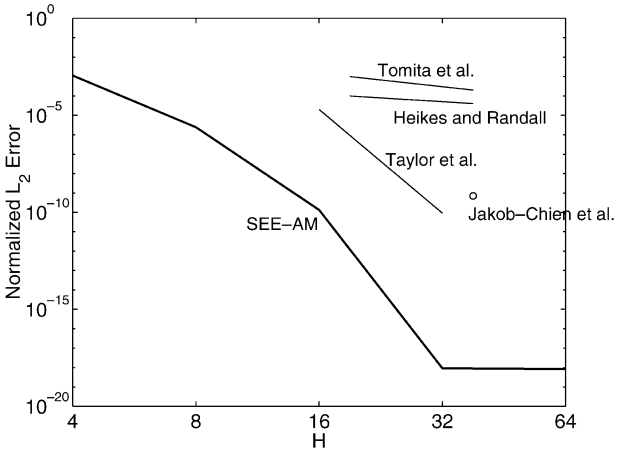

FIG. 10. Barotropic case 2: The geopotential, ϕ, normalized L_2 error as a function of horizontal resolution, H, for the Williamson et al. (1992) shallow water case 2 after 5 days for SEE-AM (thick line), the Jakob-Chien et al. (1995) (spectral transform), the Heikes and Randall (1995) (finite difference), the Taylor et al. (1997) (spectral element), and the Tomita et al. (2001) (finite difference) models. The SEE-AM model uses the $n_H = 1$ ($N_e = 6$) grid.

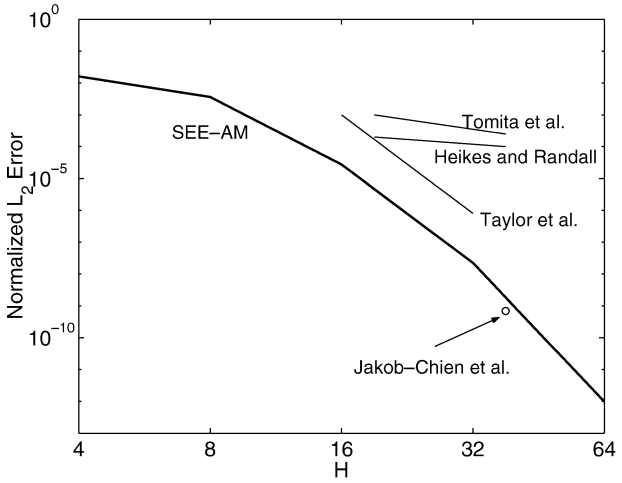

FIG. 11. Barotropic case 3: The geopotential, ϕ, normalized L_2 error as a function of horizontal resolution, H, for the Williamson et al. (1992) shallow water case 3 after 5 days for SEE-AM (thick line), the Jakob-Chien et al. (1995) (spectral transform), the Heikes and Randall (1995) (finite difference), the Taylor et al. (1997) (spectral element), and the Tomita et al. (2001) (finite difference) models. The SEE-AM model uses the $n_H = 1$ ($N_e = 6$) grid.

the equivalent hexahedral resolution by using Eqs. (41) and (43) such that for a given number of grid points N_p we get

$$H = \sqrt{\frac{N_p - 2}{6}}.$$

The results for the horizontal operators are summarized as follows. Figures 9 and 10 show that for cases 1 and 2 SEE-AM yields the best accuracy of all the models including the high-accuracy Jakob-Chien et al. (1995) and Taylor et al. (1997) models. For case 1, SEE-AM is almost an order of magnitude more accurate than the Taylor et al. model and is twice as accurate as the Jakob-Chien et al. model. For case 2, SEE-AM is approximately eight orders of magnitude more accurate than these two high-accuracy models. The plateauing of the error for SEE-AM is due to the accuracy-reaching machine precision. Finally, Fig. 11 shows that for case 3 SEE-AM gives better accuracy than all the other models except for the spectral transform model of Jakob-Chien et al., which yields a slightly more accurate result (0.7×10^{-9} compared to 1×10^{-9}).

b. Baroclinic tests

Because there are no analytic solutions to the full atmospheric equations, we cannot run test cases as in the barotropic case and compare to exact solutions. Instead, we need to either use test cases in which the outcome is a simple enough pattern that might be easily discerned beforehand or we need to run benchmark test cases run by a vast community. We have chosen to use both types of test cases: the Rossby–Haurwitz wave and the balanced initial state representing the former and the Held–Suarez test and the baroclinic instability testing the latter.

No diffusion operators are included in any of our results for both NOGAPS and SEE-AM. At every time step, the ⅔ triangular truncation is applied to NOGAPS and the elementwise filter is applied to SEE-AM.

1) ROSSBY–HAURWITZ WAVENUMBER 4

In order to judge the accuracy of SEE-AM, we compare it to NOGAPS for the Rossby–Haurwitz wavenumber 4. This test does not have an analytic solution, and so we use it for qualitative comparisons. From Mon-

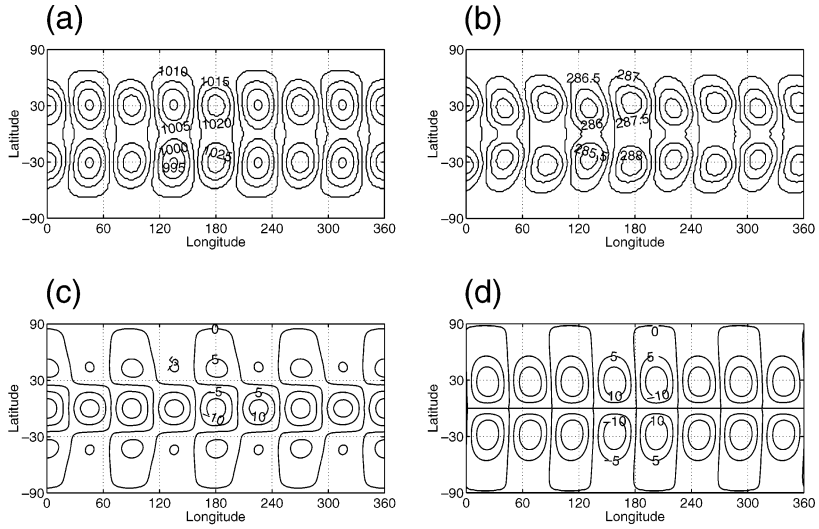

FIG. 12. Rossby–Haurwitz wavenumber 4: The surface (a) pressure (hPa), (b) temperature (K), (c) zonal velocity (m s^{-1}); and (d) meridional velocity (m s^{-1}) for SEE-AM, with $H160$ ($n_H = 20$, $N = 8$) and $N_{\text{lev}} = 24$, for a 5-day integration.

aco and Williams (1975), we initialize the model as follows: the wind velocity is

$$u(\lambda, \varphi, \sigma) = -\frac{1}{a}[A \sin(n\lambda) \cos^{n+1}\varphi - nA \sin(n\lambda) \times \cos^{n-1}\varphi \sin^2\varphi - Ba^2 \cos\varphi],$$

$$v(\lambda, \varphi, \sigma) = \frac{1}{a}[An \sin\varphi \cos^{n-1}\varphi - nA \sin(n\lambda) \cos(n\lambda)],$$

$\forall \sigma \in [0, 1]$, where

$$A = \frac{50a}{n}, \qquad B = \frac{20}{a},$$

where a is the earth's radius, and n is the wave number. The temperature field is based on the NACA standard atmosphere and is defined as

$$T = 288 - 0.0065z \text{ (K)},$$

where

$$z = 44\,308\left[1 - \left(\frac{p}{p_o}\right)^{0.19023}\right] \text{ (m)}.$$

The terrain pressure is given as

$$\pi = p_o\left(1 + \frac{\phi}{434\,505.6}\right)^{5.2568} - p_t,$$

where the geopotential is

$$\phi = a^2A_\phi + a^2B_\phi \sin(n\lambda) + a^2C_\phi[2 \sin^2(n\lambda) - 1] \quad \text{and}$$

$$A_\phi = \frac{B}{2}(2\Omega + B) \cos^2\varphi + \frac{1}{4}\left(\frac{A}{a^2}\right)^2 \cos^{2n}\varphi\left[(n + 1) \cos^2\varphi + (2n^2 - n - 2) - \frac{2n^2}{\cos^2\varphi}\right],$$

$$B_\phi = \frac{2(\Omega + B)\dfrac{A}{a^2}}{(n + 1)(n + 2)} \cos^n\varphi[(n^2 + 2n + 2) - (n + 1)^2 \cos^2\varphi],$$

$$C_\phi = \frac{1}{4}\left(\frac{A}{a^2}\right)^2 \cos^{2n}\varphi[(n + 1) \cos^2\varphi - (n + 2)].$$

Surface contours of the prognostic variables after a 5-day integration for T159, $N_{\text{lev}} = 24$ resolutions of SEE-AM and NOGAPS, are shown in Figs. 12 and 13, respectively. The results between the two models are virtually indistinguishable. This means that both models yield similar values for all of the prognostic variables as well as similar phase speeds—an important property for the successful tracking of tropical cyclones. It should not be surprising that SEE-AM gives identical results to NOGAPS. Both models use the same temporal and vertical discretization methods. The only difference is in the horizontal discretization methods. However, in section 6a, using barotropic test cases, we showed that the spectral element method gives almost identical accuracy to the spectral transform method.

Having established the accuracy of the model, we now turn to the stability of the spectral element model for longer time integrations.

2) HELD–SUAREZ TEST CASE

This test case was introduced by Held and Suarez (1994) and has been the most widely used test for dynamical cores. In essence, this test case provides a good platform to assess the capabilities of the model in simulating a realistic climate circulation. Simple boundary conditions are used in order to parameterize the radiative forcing at the surface. The momentum and potential temperature equations are slightly altered in order to introduce an equilibrium temperature due to the subgrid-scale physical processes, and a Rayleigh damping of the low-level winds is included to represent boundary layer friction. The momentum equation is now defined as follows:

$$\frac{\partial \mathbf{u}}{\partial t} = \cdots - k_v\mathbf{u},$$

and the potential temperature is

$$\frac{\partial \theta}{\partial t} = \cdots - k_\theta(\theta - \theta_{\text{eq}}),$$

where the ellipses denote the usual terms in the momentum and potential temperature equations [see Held and Suarez (1994) for the values of k_v, k_θ, and θ_{eq}]. For this test we use an equivalent resolution to that used in Held and Suarez (1994), namely, $H64$ ($n_H = 8$ and $N = 8$), with 20 vertical levels (N_{lev}). The mean zonally averaged zonal velocity and temperature are shown as a function of the vertical coordinate σ in Fig. 14. These plots are obtained from a 1200-day integration, with the results sampled every 4 days beginning with day 200. These results compare quite well with those obtained

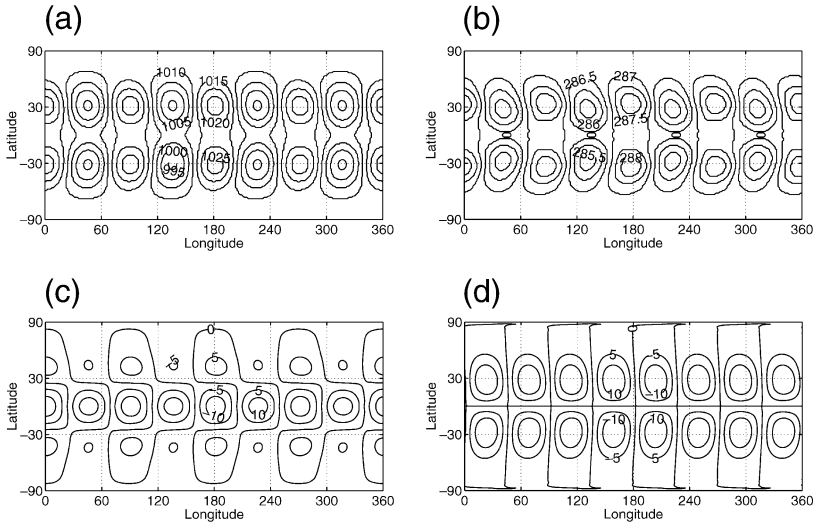

FIG. 13. Rossby–Haurwitz wavenumber 4: The surface (a) pressure (hPa), (b) temperature (K), (c) zonal velocity (m s^{-1}), and (d) meridional velocity (m s^{-1}) for NOGAPS, with $T159$ and $N_{lev} = 24$, for a 5-day integration.

with the spectral transform model in Held and Suarez (1994), where the midlatitude jets in the upper atmosphere are clearly visible (Fig. 14a) and a realistic temperature stratification is maintained (Fig. 14b).

3) JABLONOWSKI–WILLIAMSON TEST CASES

The following two cases represent a new set of tests for judging the accuracy and stability of dynamical cores. These tests are introduced in Jablonowski and Williamson (2002).

The surface pressure is initially given as $p_s(\lambda, \varphi) = 1000$ hPa, and the initial wind velocities are defined as

$$u(\lambda, \varphi, \sigma) = u_0 \cos^{3/2}\sigma_v \sin^2(2\varphi),$$
$$v(\lambda, \varphi, \sigma) = 0,$$

where $u_0 = 35$ m/s, $\sigma_v = (\sigma - \sigma_0)\pi/2$, $\sigma_0 = 0.252$, and σ is the vertical coordinate. The horizontally averaged temperature is

$$T(\sigma) = \begin{cases} T_0 \sigma^{R_d \Gamma/g} & \text{for } \sigma \geq \sigma_t \\ T_0 \sigma^{R_d \Gamma/g} + \Delta T(\sigma_t - \sigma)^5 & \text{for } \sigma < \sigma_t, \end{cases}$$

where $T_0 = 288$ K, $\Gamma = 0.005$ K m^{-1}, $g = 9.806$ m s^{-2}, $\Delta T = 4.8 \times 10^5$ K, and $\sigma_t = 0.2$ is the tropopause level. Defining the following functions

$$A_c = \left[-2 \sin^6\varphi\left(\cos^2\varphi + \frac{1}{3}\right) + \frac{10}{63}\right],$$
$$B_c = +a\omega\left[\frac{8}{5}\cos^3\varphi\left(\sin^2\varphi + \frac{2}{3}\right) - \frac{\pi}{4}\right],$$

we can now write the potential temperature as follows:

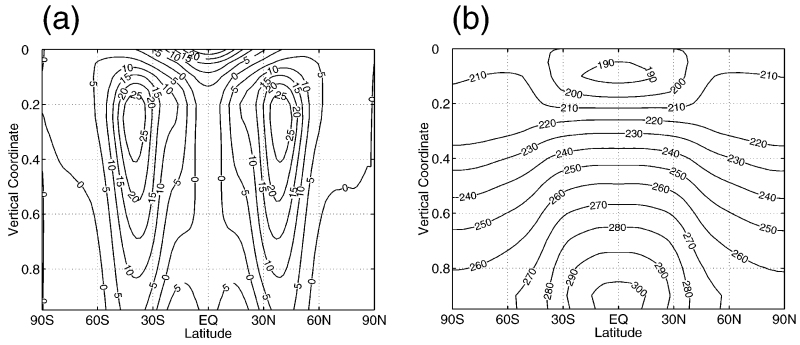

FIG. 14. Held–Suarez test: Plots of the (a) mean zonally averaged zonal velocity (m s^{-1}) and (b) mean zonally averaged temperature (K) for SEE-AM after 1200 days for $H64$ ($n_H = 8$, $N = 8$) and $N_{lev} = 20$.

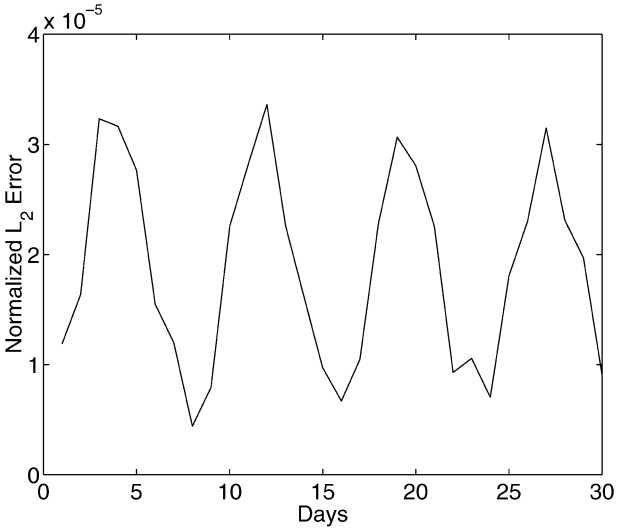

FIG. 15. Jablonowski–Williamson balanced initial state: The normalized surface pressure, π, L_2 error as a function of days for SEE-AM H160 ($n_H = 20$, $N = 8$), with 24 vertical levels.

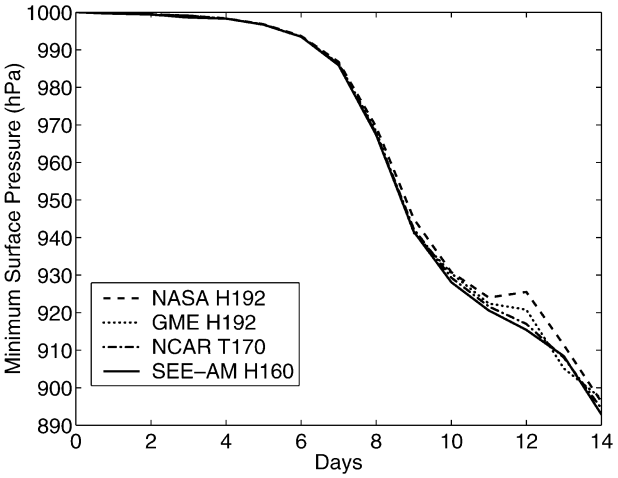

FIG. 16. Jablonowski–Williamson baroclinic instability: The minimum surface pressure (hPa) as a function of days for the NASA (finite volume), GME (finite difference), NCAR (spectral transform), and SEE-AM (spectral element, with $n_H = 20$ and $N = 8$) models using 26 vertical levels. (The data for the first three models are courtesy of C. Jablonowski.)

$$\theta(\lambda, \varphi, \sigma) = P(\lambda, \varphi, \sigma)T(\sigma) + \frac{3}{4}P(\lambda, \varphi, \sigma)\frac{\sigma\pi u_0}{R_d} \times \sin\sigma_v \cos^{1/2}\sigma_v[2u_0 \cos^{3/2}(\sigma_v)A_c + B_c],$$

where P is the Exner function and (a, ω) are the radius of the earth and its angular velocity, respectively. The surface geopotential is

$$\phi_s(\lambda, \varphi) = u_0 \cos^{3/2}\left[\frac{\pi}{2}(\sigma_s + \sigma_0)\right] \times \left\{u_0 \cos^{3/2}\left[\frac{\pi}{2}(\sigma_s - \sigma_0)\right]A_c + B_c\right\},$$

where $\sigma_s = 1$. We compare SEE-AM with three well-established models using these test cases:

- *Balanced initial state.* For this test case, the atmosphere is initially balanced by the above equations for surface pressure, p_s; wind velocities, (u, v); potential temperature, θ; and surface geopotential, ϕ_s. Using these initial conditions, the equations should remain balanced for an indefinite amount of time. Figure 15 shows the normalized surface pressure, π, L_2 error norm as a function of time for a 30-day period for SEE-AM with H160 horizontal resolution and 24 vertical levels. Note that while the error oscillates with time it remains bounded, which confirms that the initial balanced state is maintained.
- *Baroclinic instability.* This case is similar to the balanced initial state except that now a perturbation is added to the initial zonal velocity. This perturbation is given by

$$\hat{u}(\lambda, \phi) = \exp\left[-\left(\frac{r}{R}\right)^2\right],$$

where

$$r = a \arccos[\sin\varphi_c \sin\varphi + \cos\varphi_c \cos\varphi \cos(\lambda - \lambda_c)],$$

and $(\lambda_c, \varphi_c) = (\pi/9, 2\pi/9)$ and $R = a/10$ are the location of the perturbation and its radius.

This perturbation grows until a baroclinic instability develops and then breaks near day 9. Figure 16 shows the minimum surface pressure p_s as a function of time for SEE-AM against various models including the NCAR spectral transform model (Hack et al. 1992), the National Aeronautics and Space Administration (NASA) Goddard finite-volume model (Yeh et al. 2002), and the German Weather Service icosahedral finite-difference model (Majewski et al. 2002) known as GME; the results of the latter three models are courtesy of C. Jablonowski (2003, personal communication). Figure 17 shows a zoomed in version of Fig. 16. For the gridpoint models, we use the definition of equivalent hexahedral resolution, H, which we defined for the barotropic test cases. The results of this case are summarized as follows. Figure 16 shows that all four models are in complete agreement until day 8, at which point the two lower-order models, NASA and GME, diverge from the NCAR and SEE-AM models. NASA and GME compare well with each other throughout the 14-day integration. There are some slight deviations between days 12 and 13 (Fig. 17), but overall they both follow the same pattern. The two higher-order models, NCAR and SEE-AM, compare extremely well with each other through-

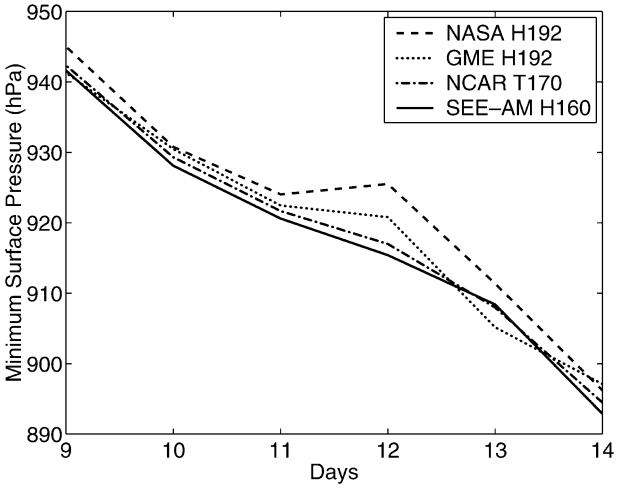

FIG. 17. Jablonowski–Williamson baroclinic instability: A zoomed-in view of Fig. 16.

out the 14-day integration. This can be seen more clearly in Fig. 17, where the pressure curves are directly on top of each other.

7. Conclusions

A new dynamical core constructed using the spectral element method based on 3D Cartesian coordinates is presented. The advantages of using Cartesian coordinates are the elimination of the polar singularity; the flexibility to use any grid, including adaptive unstructured grids; and the ease with which the Eulerian model can be converted to a semi-Lagrangian form. The advantage of using the spectral element method is that it achieves the same order of accuracy as the spectral transform method while taking better advantage of distributed-memory computers.

In this paper we show results for icosahedral and hexahedral grids, and in future work we expect to report on the use of adaptive unstructured grids. The exponential accuracy of the spectral element method is illustrated using analytic solutions to barotropic test cases, which confirms that the spectral element method yields at least the same level of accuracy obtained with the spectral transform method. Using baroclinic test cases, we demonstrate that our spectral element atmospheric model gives similar results to spectral transform models, including the U.S. Navy's NWP model and the NCAR climate model. Finally, the performance of the spectral element model is shown to scale linearly with increasing processors—a trait not shared by spectral transform models. Through our comparison of NOGAPS and SEE-AM, we show why the spectral element model will outperform spectral transform models for the types of horizontal resolutions required by future NWP applications.

The results confirm that SEE-AM offers an attractive alternative strategy for constructing future NWP and climate models on parallel computers. In order to make SEE-AM competitive with operational NWP spectral transform models, we are extending the explicit Eulerian model to semi-implicit Eulerian and fully implicit semi-Lagrangian, and we hope to report our findings in the future.

Acknowledgments. The authors gratefully acknowledge the support of their sponsor, the Office of Naval Research, through Program Element PE-0602435N. We gratefully acknowledge the computing resources from the Department of Defense NAVO Major Shared Resource Center. We would like to thank Christiane Jablonowski for allowing us to use a preprint of her paper and for sharing with us her datasets for the baroclinic instability test case. We would also like to thank Steve Lowder for his assistance in getting SEE-AM running on the NAVO IBM SP3, Jean Côté for fruitful discussions on the construction of atmospheric models using Cartesian coordinates, Todd Ringler for helpful discussions regarding the Held–Suarez test case, and João Teixeira for helpful comments regarding the manuscript.

APPENDIX

Characteristic Velocity of the Atmospheric Equations

The atmospheric equations can be written in the following conservation form:

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}(\mathbf{q}),$$

where

$$\mathbf{q} = \begin{pmatrix} \pi \\ \pi u \\ \pi v \\ \pi w \\ \pi \theta \end{pmatrix},$$

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} 0 \\ -\dfrac{2\omega z}{a^2}\pi(yw - zv) + \phi\dfrac{\partial \pi}{\partial x} - \mu x \\ -\dfrac{2\omega z}{a^2}\pi(zu - xw) + \phi\dfrac{\partial \pi}{\partial y} - \mu y \\ -\dfrac{2\omega z}{a^2}\pi(xv - yu) + \phi\dfrac{\partial \pi}{\partial z} - \mu z \\ 0 \end{bmatrix} - \frac{\partial}{\partial \sigma}(\mathbf{q}\dot{\sigma}),$$

$$\mathbf{F} = \mathbf{f}\mathbf{i} + \mathbf{g}\mathbf{j} + \mathbf{h}\mathbf{k},$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$ denote the Cartesian directional vectors, and the fluxes are

$$\mathbf{f} = \begin{pmatrix} \pi u \\ \pi u^2 + \pi\phi \\ \pi v u \\ \pi w u \\ \pi \theta u \end{pmatrix}, \qquad \mathbf{g} = \begin{pmatrix} \pi v \\ \pi v u \\ \pi v^2 + \pi\phi \\ \pi w v \\ \pi \theta v \end{pmatrix},$$

$$\mathbf{h} = \begin{pmatrix} \pi w \\ \pi u w \\ \pi v w \\ \pi w^2 + \pi\phi \\ \pi \theta w \end{pmatrix}. \tag{A1}$$

The eigenvalues of $\partial F/\partial q$ are

$$\mathbf{\Lambda} = \begin{pmatrix} U + \sqrt{\phi} \\ U - \sqrt{\phi} \\ U \\ U \\ U \end{pmatrix},$$

where $U = \mathbf{n} \cdot \mathbf{u}$, and the maximum wave speed of the atmospheric equations is $\mathbf{\Lambda}_{\max} = U + \sqrt{\phi}$.

REFERENCES

Asselin, R., 1972: Frequency filter for time integrations. *Mon. Wea. Rev.,* **100,** 297–301.

Boyd, J. P., 1998: Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: Preserving boundary conditions and interpretation of the filter as a diffusion. *J. Comput. Phys.,* **143,** 283–288.

Cooley, J. W., and J. W. Tukey, 1965: An algorithm for the machine calculation of complex Fourier series. *Math. Comput.,* **19,** 297–301.

Côté, J., 1988: A Lagrange multiplier approach for the metric terms of semi-Lagrangian models on the sphere. *Quart. J. Roy. Meteor. Soc.,* **114,** 1347–1352.

Dennis, J. M., 2003: Partitioning with space-filling curves on the cubed-sphere. *Proceedings of the International Parallel and Distributed Processing Symposium,* Nice, France, 6 pp.

Foster, I., W. Gropp, and R. Stevens, 1992: The parallel scalability of the spectral transform method. *Mon. Wea. Rev.,* **120,** 835–850.

Giraldo, F. X., 2001: A spectral element shallow water model on spherical geodesic grids. *Int. J. Numer. Methods Fluids,* **35,** 869–901.

——, J. S. Hesthaven, and T. Warburton, 2002: Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *J. Comput. Phys.,* **181,** 499–525.

Hack, J. J., B. A. Boville, B. P. Briegleb, J. T. Kiehl, P. J. Rasch, and D. L. Williamson, 1992: Description of the NCAR community climate model (CCM2). NCAR Tech. Note NCAR/TN-382+STR, Climate Modeling Section, National Center for Atmospheric Research, Boulder, CO, 108 pp.

Heikes, R., and D. A. Randall, 1995: Numerical integration of the shallow water equations on a twisted icosahedral grid. Part I: Basic design and results of tests. *Mon. Wea. Rev.,* **123,** 1862–1880.

Held, I. M., and M. J. Suarez, 1994: A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models. *Bull. Amer. Meteor. Soc.,* **75,** 1825–1830.

Hodur, R. M., 1997: The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS). *Mon. Wea. Rev.,* **125,** 1414–1430.

Hogan, T. F., and T. E. Rosmond, 1991: The description of the Navy Global Operational Prediction System's spectral forecast model. *Mon. Wea. Rev.,* **119,** 1786–1815.

——, M. S. Peng, J. A. Ridout, and W. M. Clune, 2002: A description of the impact of changes to NOGAPS convection parameterization and the increase in resolution to T239L30. NRL Memo. Rep. NRL/MR/7530-02-52, Global Modeling Section, Naval Research Laboratory, Monterey, CA, 16 pp.

Iskandarani, M., D. B. Haidvogel, J. C. Levin, E. Curchitser, and C. A. Edwards, 2002: Multiscale geophysical modeling using the spectral element method. *Comput. Sci. Eng.,* **4,** 42–48.

Jablonowski, C., and D. L. Williamson, 2002: Baroclinic instability test with two jets in the midlatitudes. *Extended Abstracts, 2002 Workshop on the Solution of Partial Differential Equations on the Sphere,* Toronto, ON, Canada, Fields Institute, p. 10.

Jakob-Chien, R., J. J. Hack, and D. L. Williamson, 1995: Spectral transform solutions to the shallow water test set. *J. Comput. Phys.,* **119,** 164–187.

Karypis, G., and V. Kumar, 1998: Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.,* **48,** 96–129.

Komatitsch, D., and J. Tromp, 2002: Spectral-element simulations of global seismic wave propagation—I. Validation. *Geophys. J. Int.,* **149,** 390–412.

Loft, R. D., S. J. Thomas, and J. M. Dennis, 2001: Terascale spectral element dynamical core for atmospheric general circulation models. *Proc. ACM/IEEE SC99 Conf. on High Performance Networking and Computing,* Denver, CO, IEEE Computer Society, CD-ROM, ACM 1-58113-213-X/01/0011.

Majewski, D., and Coauthors, 2002: The operational global icosahedral–hexahedral gridpoint model GME: Description and high-resolution tests. *Mon. Wea. Rev.,* **130,** 319–338.

Monaco, A. V., and R. T. Williams, 1975: An atmospheric global prediction model using modified Arakawa differencing scheme. NPS Tech. Rep. NPS-51WU75041, Dept. of Meteorology, Naval Postgraduate School, Monterey, CA, 86 pp.

Randall, D. A., T. D. Ringler, and R. P. Heikes, 2002: Climate modeling with spherical geodesic grids. *Comput. Sci. Eng.,* **4,** 32–41.

Ringler, T. D., R. P. Heikes, and D. A. Randall, 2000: Modeling the atmospheric general circulation using a spherical geodesic grid: A new class of dynamical cores. *Mon. Wea. Rev.,* **128,** 2471–2490.

Ronchi, C., R. Iacono, and P. S. Paolucci, 1996: The ''cubed sphere'': A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.,* **124,** 93–114.

Rosmond, T. E., 2000: A scalable version of the Navy Operational Global Atmospheric Prediction System spectral forecast model. *Sci. Programm.,* **8,** 31–38.

——, J. Teixeira, M. Peng, T. Hogan, and R. Pauley, 2002: The Navy Operational Global Atmospheric Prediction System (NOGAPS): Forcing for ocean models. *Oceanography,* **15,** 99–108.

Sela, J. G., 1980: Spectral modeling at the National Meteorological Center. *Mon. Wea. Rev.,* **108,** 1279–1292.

Simmons, A. J., D. M. Burridge, M. Jarraud, C. Girard, and W. Wergen, 1989: The ECMWF medium-range prediction models development of the numerical formulations and the impact of increased resolution. *Meteor. Atmos. Phys.,* **40,** 28–60.

Swarztrauber, P. N., D. L. Williamson, and J. B. Drake, 1997: The Cartesian method for solving partial differential equations in spherical geometry. *Dyn. Atmos. Oceans,* **27,** 679–706.

Taylor, M., J. Tribbia, and M. Iskandarani, 1997: The spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.,* **130,** 92–108.

Thomas, S. J., R. D. Loft, and J. M. Dennis, 2002: Parallel implementation issues: Global versus local methods. *Comput. Sci. Eng.,* **4,** 26–31.

Tomita, H., M. Tsugawa, M. Satoh, and K. Goto, 2001: Shallow water model on a modified icosahedral geodesic grid by using spring dynamics. *J. Comput. Phys.,* **174,** 579–613.

Tufo, H. M., and P. F. Fischer, 1999: Terascale spectral element algorithms and implementations. *Proc. ACM/IEEE SC99 Conf. on High Performance Networking and Computing,* Portland, OR, IEEE Computer Society, CD-ROM.

Warburton, T., L. F. Pavarino, and J. S. Hesthaven, 2000: A pseudo-spectral scheme for the incompressible Navier–Stokes equations using unstructured nodal elements. *J. Comput. Phys.,* **164,** 1–21.

Williamson, D. L., J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.,* **102,** 211–224.

Yeh, K. S., S. J. Lin, and R. B. Rood, 2002: Applying local discretization methods in the NASA finite-volume general circulation model. *Comput. Sci. Eng.,* **4,** 49–54.