JOURNAL OF COMPUTATIONAL PHYSICS **130,** 92–108 (1997)
ARTICLE NO. CP965554

The Spectral Element Method for the Shallow Water Equations on the Sphere

Mark Taylor,* Joseph Tribbia,* and Mohamed Iskandarani†

*National Center for Atmospheric Research, Boulder, Colorado 80303; and †Institute of Marine and Coastal Sciences, Rutgers University, P.O. Box 231, New Brunswick, New Jersey 08903-0231

Received December 15, 1995; revised July 15, 1996

The spectral element method is implemented for the shallow water equations in spherical geometry and its performance is compared with other models. This is the first step in evaluating the suitability of spectral elements for climate modeling. The potential advantages and disadvantages of spectral elements over more conventional models used for climate studies are discussed. The method requires that the sphere be tiled with rectangles, for which we make use of the gnomonic projection to map the sphere onto the cube. To measure the performance of the method relative to other models, results are presented from a standard suite of shallow water test cases for the sphere. These results confirm the spectral accuracy of the method. © 1997 Academic Press

1. INTRODUCTION

The spectral element method [18, 17], has several potential advantages over conventional spherical harmonic spectral models used in atmospheric modeling. This fact, along with the recent success of the spectral element method in ocean modeling [16, 12, 9] has prompted us to consider its use in global climate modeling. Spectral elements combine the accuracy of conventional spectral methods and the geometric flexibility of finite element methods. In the spectral element discretization, the computational domain is broken up into rectangular regions called elements, and within each of these elements all variables are approximated by polynomial expansions. The discrete equations are derived by using a Galerkin or integral form of the equations to be solved in conjunction with a suitable set of test functions and quadrature formula.

The first step in evaluating the suitability of spectral elements for climate modeling is to implement the method in spherical geometry and measure its performance relative to other methods with a full range of numerical simulations. We have modified the oceanic spectral element model in [12, 9] for this purpose, and we adopt the now standard suite of shallow water test cases proposed in [21]. The shallow water equations have many of the difficulties associated with the dynamical aspects of climate modeling and any potential climate model should perform well on these tests. In this paper we will present the results from these test cases after first comparing and contrasting the spectral element method with global spectral methods and describing the details of the model specific to spherical geometry.

2. SPHERICAL HARMONICS AND SPECTRAL ELEMENTS

Global atmospheric modeling is in one sense easier than ocean modeling. The geometrically complex ocean basins are replaced by the surface of a sphere. The sphere has no boundary, and for boundary-less or periodic domains spectral methods have unsurpassed accuracy. For the sphere, the natural basis functions are spherical harmonics, and indeed, the majority of all major climate models in use today are spherical harmonic-based spectral methods [3, 8]. We thus first describe the advantages and disadvantages of this approach and show that spectral elements are potentially very competitive.

In addition to high accuracy, spherical harmonic spectral methods can provide a completely isotropic representation of a scalar function on the sphere. Any finite difference model cannot have this property, since the only isotropic grids would be those with grid points chosen to be the vertices of a platonic solid, and the largest such grid comes from the dodecahedron which has only 20 vertices. The common latitude-longitude grid is extremely nonisotropic, clustering points near the poles resulting in a severe CFL condition. This ''pole problem'' is naturally avoided by the use of spherical harmonics. Another advantage of spherical harmonics is that they are eigenfunctions of the Laplace operator on the sphere. This property makes it very easy to invert the linear elliptic operators required for semi-implicit time stepping schemes.

Spherical harmonics also have some disadvantages, the most notable being their high cost. Unlike the fast fourier transform (FFT), the discrete spherical harmonic trans-

0021-9991/97 $25.00

form (DST) is computed by direct methods since it is currently unknown if a practical fast DST exists. Letting N^2 be the number of grid points, the computational cost of the DST is $O(N^3)$. This is a factor of N larger than the cost of all other aspects of climate models. Consequently, at high resolutions the transforms will completely dominate the computations. In addition to this drawback, the DST is also difficult to implement on distributed memory machines. Every spectral coefficient computed requires data from every grid point, resulting in a large amount of interprocess communication. Last, it is difficult to incorporate local mesh refinement in global spectral models.

On the sphere, spectral elements have most of the advantages of spherical harmonics. Due to the element boundaries and lower spectral degree, they are not quite as accurate for a given resolution, but they do converge exponentially fast under mesh refinement. Spectral elements do not have any pole problems since the sphere can be tiled with squares of approximately the same size and thus avoid clustering points at the poles. By using a local coordinate system within each element, any singularities in the coordinate system can be avoided. Spectral elements do have the disadvantage that it is no longer trivial to invert the linear differential operators required for semi-implicit time stepping schemes. As of yet we have not found an efficient way to do this.

The spectral element method also has an $O(N^3)$ complexity due to the many Legendre transforms that must be performed within each element. However, in this case, N refers to the spectral degree used in each element. Since the ratio of elements to spectral degree N can be controlled, N can be kept small, thus keeping the $O(N^3)$ cost of the transforms in line with the rest of the model. Spectral elements are ideal for parallel machines. The communication within each element is global, but elements only require boundary information from their neighboring elements. Thus when each processor of a parallel machine can handle one or more elements, communications overhead is found to be very low [9].

Perhaps the biggest advantage of the spectral element method is that its use of unstructured grids provides a very natural setting for local mesh refinement, allowing tasks such as regional climate modeling or resolving local topographical features to be carried out within a global model. Higher resolution can be obtained in a given region by using a larger number of smaller elements in that region. This can be accomplished by using variable-sized elements which can be made to fit conformally. Examples of such grids for the ocean can be seen in [9, 12]. There are also automated methods for generating such grids, like ''paving'' [2]. Yet another possibility is to switch to a nonconforming spectral element method (such as in [1, 11]).

3. THE SPECTRAL ELEMENTS DISCRETIZATION

As mentioned in the Introduction, the spectral element model used for this work is the one described in [12], the only differences coming from the complications of spherical geometry and in the treatment of the diffusion term. In this section we give a brief summary of the spectral element discretization. For a more thorough review of this material; see [17]. The details relevant to the implementation of the method in spherical geometry are described in the next section.

The spectral element discretization is particularly simple because of a clever combination of an integral form of the equations, the family of test functions used, and the quadrature formula chosen to approximate integrals. The procedure is as follows: First an integral form of the equations is chosen. To do this, both sides of the equations are multiplied by a test function and integrated over the entire computational domain (in our case, the surface of the sphere). The sphere is then tiled with rectangular elements and the integrals are written as the sum of integrals over each element. The element integrals are then approximated by Gauss–Lobatto quadrature, and in each element we choose for our set of test functions the associated Gauss–Lobatto Cardinal functions [12, 4]. Also within each element we take the same Cardinal functions as our basis functions for representing the dependent variables.

Except for the choice of a time-stepping scheme, the resulting integral relations completely specify how the equations are to be solved. Remarkably, this leads us to a simple Legendre spectral transform method within each element. Derivatives are calculated with Legendre transforms on the Gauss–Lobatto grid. Because of this grid and the associated Cardinal functions, the only communication between elements occurs at the element boundaries, where neighboring elements share common points. At these points the terms appearing in the equations are multiple-valued and the required area-weighted averaging is specified by the discretized form of the equations. This averaging is independent of the flow and thus does not result in any type of upwinding.

Following the lead of [16, 12], we use the third-order Adams–Bashforth time stepping scheme. This explicit scheme requires information at two previous time levels, so a fourth-order Runge–Kutta scheme is used to start the model.

4. SPECTRAL ELEMENTS ON THE SPHERE

The first step in applying the spectral element method to spherical geometry is to tile the sphere with rectangular elements, or regions that can be easily mapped to rectangles. This tiling is most easily accomplished by inscribing

FIG. 1. The cube projected onto the sphere. Each of the six faces of the cube has been divided into an 8×8 array of elements.

a polyhedron with rectangular faces inside the sphere and then using the gnomonic projection (projection from the center of the sphere) to map the surface of the polyhedron to the surface of the sphere. The most elementary such polyhedron is the cube, which creates six large elements. One can then further divide each of these elements into smaller elements, and the projection onto the sphere of such a configuration is shown in Fig. 1. Each of the six faces of the cube has been divided into an 8×8 array of elements.

The cube has the further advantage that, modulo rotations, there is only one mapping which, along with its derivatives, can be easily derived analytically (see the Appendix). Further subdividing each face of the cube into smaller elements is now a simple matter since we are working in a square domain with Cartesian coordinates. More complicated tilings of the sphere would allow greater flexibility for local mesh refinement, at the cost of complicating the mappings. The approach taken in [12], if modified to handle the polar singularities, could also be applied to an arbritrary tiling of the sphere. The method only requires that the sphere be subdivided into regions which are then conformally (as opposed to gnomonically) mapped to rectangles. This type of mapping and the associated metric terms must be approximated numerically.

We will refer to the natural Cartesian coordinates in each face of the cube as the cube coordinate system. To write the shallow water equations in this coordinate system, we first start with their vector formulation

$$\frac{\partial \mathbf{v}}{\partial t} = -(\zeta + f)\hat{k} \times \mathbf{v} - \nabla\left(\frac{1}{2}\mathbf{v} \cdot \mathbf{v} + gH\right),$$

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\mathbf{v}),$$

where $\mathbf{v}$ is the velocity vector, h is the thickness of the fluid, h_s is the height of the underlying mountains, and

$$H = h + h_s, \quad \zeta = \hat{k} \cdot \nabla \times \mathbf{v}, \quad f = 2\Omega \sin(\theta).$$

Let x and y be the Cartesian coordinates in a face of the cube. On the sphere, we denote longitude by λ, latitude by θ, and the associated unit vectors by $\hat{\lambda}$ and $\hat{\theta}$. We will denote the derivative of the mapping from $(x, y) \to (\lambda, \theta)$ by the matrix $\mathbf{D}$,

$$\mathbf{D} = \begin{pmatrix} \cos(\theta)\lambda_x & \cos(\theta)\lambda_y \\ \theta_x & \theta_y \end{pmatrix},$$

where a subscript of either x or y represents differentiation with respect to x or y. The Jacobian of the transformation is given by the determinant of $\mathbf{D}$. The formulas for $\mathbf{D}$, $\mathbf{D}^{-1}$, the mapping functions and the other metric terms which appear below are given in the Appendix, where it can be seen that these terms are completely free of any singularities.

To solve the shallow water equations for the velocity, we must pick a set of basis vectors and solve for the coefficients of the velocity when expressed in terms of these vectors. The choice of the spherical coordinate system

$$\mathbf{v} = u_1\hat{\lambda} + u_2\hat{\theta}$$

is particularly troublesome since the coordinates are discontinuous at the poles. One could consider using spherical coordinates in the elements which are near the equator and a rotated spherical coordinate system in those elements near the poles. However, the most natural choice is that associated with the coordinate system we are using for the independent variables x and y, and this choice also gives us the most compact form of the transformed equations. Defining

$$\mathbf{x} = \mathbf{D^{-1}}\hat{\lambda}, \quad \mathbf{y} = \mathbf{D}^{-1}\hat{\theta},$$

we can then write

$$\mathbf{v} = u_1\hat{\lambda} + u_2\hat{\theta} = v_1\mathbf{x} + v_2\mathbf{y}$$

and the components of the velocity vector $\mathbf{v}$ in the (λ, θ) and (x, y) coordinate systems, in matrix notation, are related by

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

We must now write the shallow water equations for the dependent variables h, v_1, and v_2. To do this, we take each vector and differential operator appearing in the shallow water equations and write them in matrix notation in the cube coordinate system:

$$\mathbf{v} \cdot \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}^{\mathrm{T}} \mathbf{D}^T \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

$$\hat{k} \times \mathbf{v} = \mathbf{D}^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

$$\nabla h = \mathbf{D}^{-1} \mathbf{D}^{-T} \begin{pmatrix} h_x \\ h_y \end{pmatrix},$$

$$\nabla \cdot \mathbf{v} = \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \mathbf{p}^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

$$\zeta = \left[\mathbf{D}^{\mathrm{T}} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{D}^{-\mathrm{T}} \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix} \right]^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \mathbf{q}^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

where $\mathbf{p}$ and $\mathbf{q}$ are 2-vectors containing additional metric terms. For efficiency, $\mathbf{p}$, $\mathbf{q}$, and all the 2×2 matrices which appear in the above formulas are analytically derived and their values at each grid point are precomputed and stored for use in the model. The method used to derive these expressions is given in the Appendix.

The final equations have the form

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}_t = -(\zeta + f) \mathbf{D}^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} - \mathbf{D}^{-1} \mathbf{D}^{-\mathrm{T}} \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix},$$

$$h_t = - \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} hv_1 \\ hv_2 \end{pmatrix} - \mathbf{p}^{\mathrm{T}} \begin{pmatrix} hv_1 \\ hv_2 \end{pmatrix},$$

where

$$\phi = \tfrac{1}{2} \mathbf{v} \cdot \mathbf{v} + gH.$$

Since each of the six elements is using a different coordinate system, the communication between an element on one face of the cube and an adjacent element on a different face is slightly more complex than the communication between elements contained within a single face of the cube. All vector quantities passed to an element from a neighbor on a different cube face must first be mapped to the correct coordinate system. Denoting the different coordinate systems of two adjacent elements by a subscript of A and B, if a vector $\mathbf{v}$ is written

$$\mathbf{v} = \begin{cases} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, & \text{A coordinate system} \\ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, & \text{B coordinate system,} \end{cases}$$

then the coordinates can be related by mapping from coordinate system A to spherical coordinates and then to coordinate system B:

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{D}_B^{-1} \mathbf{D}_A \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

5. MESH REFINEMENT AND OPERATION COUNTS

Spectral element methods, being a form of $h - p$ type finite element methods, allow for two types of mesh refinement. One can keep the number of elements fixed and increase the spectral degree used in each element, or, keeping the spectral degree fixed, one can subdivide the elements into smaller elements. Both types of mesh refinement have advantages, the former method providing exponential convergence but at a higher computational cost than the latter method, which only converges algebraically.

There are two problems with using high spectral degree within each element, both related to the cost of the method. These are most easily illustrated in a one-dimensional problem, where we have M elements, each using a degree $N - 1$ Legendre polynomial (requiring N grid points inside each element) for a total of MN grid points.

The first problem is the expense of the spectral transforms. The cost of performing one Legendre transform (or related operations like computing derivatives) on N points is $O(N^2)$, making the cost of performing one time step in a spectral element model $O(MN^2)$. Thus one can see increasing N while holding M fixed will become much more expensive than increasing M while holding N fixed. However, since modern computers are very efficient at performing these kinds of operations and the transforms represent only a percentage of the total calculations, this drawback is not significant except at very large values of N.

TABLE I

Grids Used for Various Models

Method	Spectral coefficients	Grid points	Δt in seconds
Spec. elem. $150 \times 8 \times 8$	9600	9600	120
Spec. elem. $486 \times 8 \times 8$	31104	31104	72
Spec. elem. $1734 \times 8 \times 8$	110976	110976	45
Spec. elem. $24 \times 16 \times 16$	6144	6144	90
Spec. elem. $96 \times 16 \times 16$	24576	24576	45
Spec. elem. $6 \times 24 \times 24$	3456	3456	50
Spec. elem. $6 \times 40 \times 40$	9600	9600	10
Spec. elem. $6 \times 56 \times 56$	18816	18816	5
NCAR T42 (64×128)	1849	8192	1200
NCAR T63 (96×192)	4096	18432	900
NCAR T106 (160×320)	11449	51200	600
NCAR T213 (320×640)	45796	204800	360

Note. Spec. elem. $M \times N \times N$ refers to the spectral element model with M elements, each containing an $N \times N$ grid. The time steps are given to show the more severe CFL restriction when the spectral element mesh is refined by keeping M fixed and increasing N, as opposed to keeping N fixed and increasing M. For comparison, specifications are also given for the NCAR spherical harmonic spectral model [13, 14] with the standard triangular truncation and the dimensions of the associated latitude–longitude grid given in parenthesis. The NCAR model uses a semi-implicit time stepping scheme, allowing for much larger time steps than in an explicit model.

The second, more severe problem is the CFL condition associated with increasing N over M. As with most spectral methods for nonperiodic domains, the quadrature points used for the Legendre transforms cluster grid points near the boundaries of the elements. When a high spectral degree is used in the elements, this clustering creates a more restrictive CFL condition for the size of the time step. For the shallow water equations, the time step Δt is proportional to the mesh size Δx. With equally spaced points, Δx, and thus Δt, would be inversely proportional to MN; but in a spectral element model, due to the clustering of the Legendre points, Δx and Δt are inversely proportional to MN^2 (as shown in Table I). Thus it is significantly more expensive to increase N rather than M.

In practice, one must strike a balance between accuracy and computational cost by adjusting M and N to their most efficient values. A thorough study of this issue for shallow water test case 7 (described in Section 7) is shown in Fig. 2. The l_2 error versus CPU time (IBM RS6000 workstation) required to run the test case are plotted for various values of M and N. The plot shows that, except at very low resolution, the configuration of the model which requires the least amount of CPU time to achieve a given l_2 error is N in the range of 8–16 and M chosen large enough to achieve the desired error. Using larger values of N results in a more accurate solution with fewer grid points, yet it is computationally more expensive because of the factors mentioned above. The choice of $N = 8$ is in the range typically used for most spectral element methods.

Figure 2 can also be used to establish the algebraic convergence of the spectral element method when N is held fixed and M is increased. The fact that the curves connecting results with various M at a fixed N are approximately straight lines when plotted on a log–log scale shows that the l_2 error is proportional to a power of CPU time. And CPU time is easily shown to be proportional to a power of M when N is held fixed; thus the l_2 error is proportional to a power of M and, thus, the method converges algebraically in M.

To establish the exponential convergence of the method when M is held fixed and N is increased, data from all of the seven test cases is plotted in Fig. 3. The lines drawn are the least squares fit to the data points which are plotted as circles. Except in test case 5, the lines show the almost perfect exponential decay of the error as a function of N, which is the number of grid points on a side per element. The lack of convergence in test case 5 is due solely to uncertainties in the approximation used for the exact solution described in Section 7.2.

For this study, we do not attempt to use the shallow water test case results to compare the relative efficiency of different numerical methods. A true comparison would involve running optimized versions of all the codes for a real model of the atmosphere. In these models, often the majority of computations are spent on physical processes

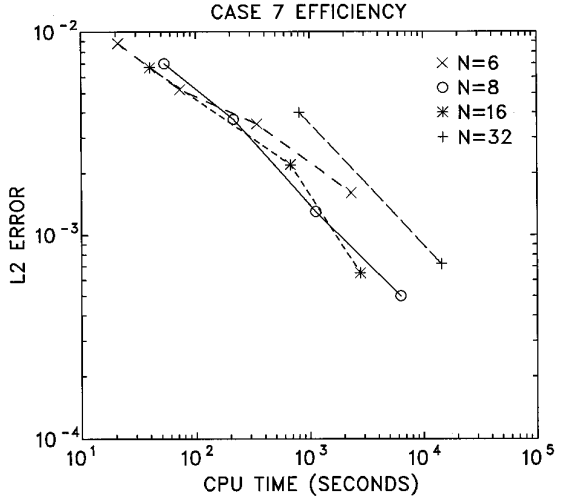

FIG. 2. The l_2 error and CPU time are plotted from 13 different runs of the spectral element model using various numbers of elements M and various values of spectral degree N within each element. Lines connect results using a fixed value N and steadily increasing M. For each N, these curves show how the errors decrease while CPU time increases as the mesh is refined by increasing M. The plot shows that the most efficient configuration of the model is N in the range of 8 to 16 and M chosen to obtain the desired resolution.

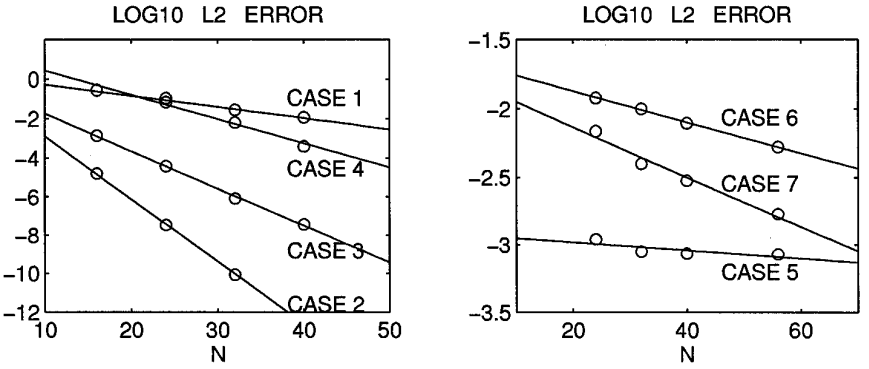

FIG. 3. Exponential convergence under mesh refinement for the spectral element method. The circles represent the $\log_{10}$ of the l_2 norms for various N. The lines drawn are the least-squares fit to the data. Each of the six elements uses a grid of size N × N.

not present in a shallow water model, and thus the shallow water test cases are not ideally suited for this kind of comparison. Also the results would depend greatly on the computer of interest. For example, on vector supercomputers at moderate resolution, it is doubtful that any method will be able to do as well as a spherical harmonic model such as the NCAR spectral model [14]. But as the resolution is increased, the NCAR model will start to become more expensive than other methods. For parallel computers the situation may be different since spectral/finite element or finite difference methods require much less interprocessor communication and, thus, could conceivably run faster even if they require a greater number of computations to achieve a given accuracy.

We can get a crude estimate of relative efficiency by comparing operation counts per grid point for one time step. The NCAR spectral model uses a grid of $N_s \times 2N_s$ with a cost of

$$8.5N_s^3 + 214N_s^2 \log N_s + O(N_s^2)$$

floating point operations [7]. Neglecting the lower order terms, the cost per grid point is $4.2N_s + 107 \log N_s$. Our implementation of the spectral element model, with M elements each containing an $N \times N$ grid has an operation count of

$$24MN^3 + O(MN^2).$$

Again neglecting the lower order terms and using the most efficient value of $N = 8$, the cost per grid point is 192. Thus at almost all resolutions the spectral element model will be faster per grid point per time step than a spherical harmonic-based spectral method.

This advantage is quickly lost by the ease at which a global spectral model can use a semi-implicit time stepping scheme, allowing for a time step 6 to 10 times larger for the shallow water test cases. Taking this factor to be 10, then, we can conclude that the spectral element method is faster per grid point than the NCAR spherical harmonic method at resolutions above T169 ($N_s = 254$). In Section 7.2, we show that for the realistic test cases, the NCAR model and the spectral element model have about the same accuracy per grid point (verified up to T106). If this holds at higher resolutions, we can conclude that spectral elements will be more efficient than spherical harmonic methods at resolutions above T169.

A summary of the specifications of the grids used in this work is given in Table I, along with similar statistics associated with conventional spherical harmonic spectral models. We point out a common misconception about the resolution of spherical harmonic models: Their resolution is better approximated by the degrees of freedom in the model, which is the number of spectral coefficients retained, not the number of grid points. The number of grid points is misleadingly large due to two factors: First, ''efficient'' quadrature formulas do not exist for the sphere [19] and the formulas currently in use require twice as many points as spectral coefficients. Second, the calculations are de-aliased which inflates the number of points required by another factor of 9/4. Thus at a typical resolution of T42, in which spherical harmonics up to degree 42 are retained, there are exactly $43^2 = 1849$ real-valued spectral coefficients, but the associated grid has $64 \times 128 = 8192$ points.

6. THE BOYD–VANDEVEN FILTER

As mentioned in [21], the addition of a diffusion term to damp small scales may improve the solutions of the test problems. Furthermore, spectral element methods are not usually de-aliased, and thus the aliasing errors generated by the nonlinear terms must be controlled with a diffusion term or the method will be unstable. There may also be other instabilities associated with the nonlinear terms. As

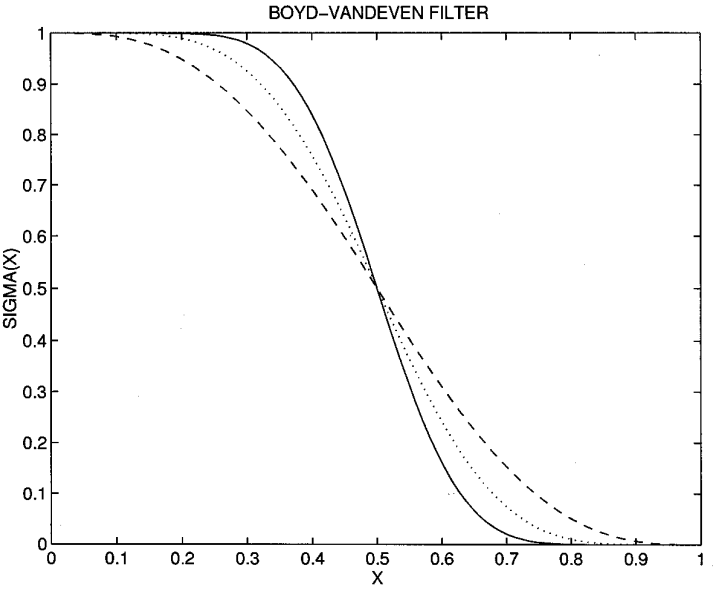

FIG. 4. The Boyd–Vandeven filter of order $p = 3$ (dotted line), $p = 6$ (dashed line), and $p = 12$ (solid line).

is often the case, we have found experimentally that a diffusion term successfully controls all these instabilities. We note the model will successfully integrate the shallow water equations without any type of diffusion if the nonlinear terms are removed.

Instead of a more traditional diffusion operator, we have chosen to use the Boyd–Vandeven filter (BV filter), a form of the Vandeven filter [20] developed by Boyd [5]. This filter is simpler to apply than a diffusion operator since it does not require the computation of high order derivatives. The BV filter of order p has the form

$$\sigma(x) = \frac{1}{2}\operatorname{erfc}\left(2\sqrt{p}\Omega\sqrt{-\frac{\log(1-4\Omega^2)}{4\Omega^2}}\right), \quad \Omega = |x| - \frac{1}{2},$$

where $\operatorname{erfc}(x)$ is the complementary error function, $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$. The response of this filter for $p = 3$, 6, and 12 is shown in Fig. 4.

For a function $f(x)$ of one variable, the filter is applied as follows: Assuming f can be written as a sum of the first N Legendre polynomials $P_k(x)$,

$$f(x) = \sum_{k=0}^{N-1} f_k P_k(x),$$

then the filtered function f' is given by

$$f'(x) = (1-\mu)f(x) + \mu\sum_{k=0}^{N-1} w_k f_k P_k(x),$$

where

$$w_k = \begin{cases} 1, & k < s \\ \sigma\left(\dfrac{k-s}{N-s}\right), & s \le k \le N. \end{cases}$$

In addition to the filter order p, we have introduced two more parameters, a filter "viscosity" μ and a filter lag coefficient s. The value of μ can range from 0 (no filtering) to 1 (full BV filter). The value of s represents how much the filter is lagged. For example, setting $s = 2N/3$ can be thought of as applying the filter only to the remaining $\frac{1}{3}$ of the spectrum of f. The values we use for these parameters are given at the end of this section.

The adaptation of the BV filter to the spectral element model is straightforward but somewhat involved. The basis functions used within each element in our formulation of the spectral element method are the Legendre–Cardinal functions. Since the N Legendre–Cardinal functions are linear combinations of the first N Legendre polynomials, the dependent variables u, v, and h can be expanded equally well in terms of Legendre polynomials. This can be done on our Gauss–Lobatto grid with the use of the Gauss–Lobatto transform [6, p. 59]. So within each element, we first expand u, v, and h in terms of Legendre polynomials, apply the filter, and then transform back. For two-dimensional elements, this entire operation is applied twice, once in each direction. Each application can be expressed as a matrix vector product, and the cost is the same as that of computing a derivative. Finally, as with computing derivatives, the filtered function will be multiple valued at element boundary points. The filtered function is made continuous again by taking area-weighted averages. We use the same type of averaging that is specified by the spectral element discretization for handling multiple-valued derivatives. Recently a more sophisticated version of this filter (especially with regards to smoothing across element boundaries) was developed for the spectral element method in [15].

For the spectral element model we apply the filter every 20 min of model time, and use $p = 12$, $\mu = 0.2$, and $s = 2N/3$. These parameters were chosen mostly through experimentation. Minor improvement in the errors for certain test cases can be obtained with further tuning of the filter, but we use these values since they work for all of the shallow water test cases at all resolutions that we have tried.

7. SHALLOW WATER TEST CASE RESULTS

As a first test of the suitability of the spectral element model for spherical geometry and ultimately climate modeling, we use a standard set of shallow water test cases

[21]. The existence of these test cases allows us to compare the spectral element model with several other methods without having to obtain and run the other codes. The test cases prescribe several forms of error and how they are to be computed so that any method can be compared objectively to several other methods by simply consulting the literature.

Here we will briefly describe each test case, but we rely on [21] for the detailed mathematical description of the cases and performance measures. The first four test cases primarily determine how well the method handles spherical geometry. They have analytic solutions making the numerical errors easy to calculate, but they have no small scale structure and are thus not very realistic. More realistic cases with complicated flows are represented in the later test cases. These cases lack analytic solutions and the results must be compared with high resolution runs of other models.

7.1. *Test Case Results*

Test case 1 is the advection of a cosine bell with compact support around the sphere. This is the only case which does not involve the full set of shallow water equations. The velocity is fixed (rigid rotation about an axis at an angle α from the earth's axis) and we solve only the advection equation for the height field for 12 days (one rotation). Several orientations of the velocity are specified by the parameter α: around the equator, over the poles, and minor shifts from these two orientations. We also added an orientation which sends the cosine bell through the corners of the cube in our computational grid.

Contour plots of the height field and error from case 1 are shown in Fig. 5. These results are for a $96 \times 16 \times 16$ grid (see Section 5). The analytic and computed height fields are shown after one rotation of the cosine bell around the sphere. As specified in the test case, these are plotted on an orthographic projection centered over the cosine bell. The difference between these two are contoured in the figure, where we observe the global, low-level noise typical of spectral methods. To quantify this spectral "ringing," the test case also calls for plotting the normalized minimum and maximum of the height field, shown in Fig. 6. Although this ringing shows up rather dramatically in the contour plots, it is actually quite small, representing fluctuations of only 0.2%.

The normalized l_∞, l_1, and l_2 errors are also plotted as a function of time. To compute the integrals required for the l_1 and l_2 errors, we use the natural Gauss–Lobatto quadrature associated with the spectral element discretization. The oscillations visible in the curves are similar to those observed in [14]. They are due to sampling errors as the cosine bell moves through different parts of the computational grid.

To show the insensitivity of the model to α and, thus, the lack of any type of pole or corner problem, we plot the normalized l_∞, l_1, and l_2 errors for different values of α in Fig. 7. The error levels are almost identical as the cosine bell is sent around the equator, over the pole, through the element corners, and near the element corners.

Test case 2 is a steady state solution of the full nonlinear shallow water equations. It again consists of solid body rotation for the velocity, but this time we use the corresponding balanced height field. Again several orientations are specified by the parameter α. The height field consists of lows over each pole. Test case 3 is similar to case 2 except that the wind field has compact support. It is a mid-latitude jet and the associated height field has a single low over the region encircled by the jet.

The errors for these two test cases are summarized in Table II. Plots from cases 2 and 3 are not included, primarily because the errors are extremely small for all values of α and the height fields are uninteresting. Plots for these cases can be seen in [13]. Both these test cases again show that the spectral element model is insensitive to rotations and thus has no coordinate-related problems.

Case 4 tests the performance of a scheme on the full nonlinear unsteady equations. Forcing terms are added to the shallow water equations to generate a flow with similar structure to flows observed in the atmosphere. The flow is a translating low pressure center superimposed on a jet stream which is symmetrical about the equator. Two cases are specified, $u_0 = 20$ m/s and $u_0 = 40$ m/s, where u_0 is the speed of the jet stream. We only present results for $u_0 = 20$ m/s since both cases are very similar, the only difference being that the errors for $u_0 = 40$ m/s are systematically larger. Contour plots of the solution and error after 5 days on an orthographic projection centered on the center of the low pressure cell are shown in Fig. 8. The normalized l_∞, l_1, and l_2 errors are also plotted as a function of time. These results are from the $96 \times 16 \times 16$ grid. We note that for this case, the mean flow is removed from normalization used in calculating the errors; thus the normalized errors shown in the graph are larger than could be inferred from the contour plots.

Test case 5 is used to study the effectiveness of a scheme in conserving several integral invariants of the flow. It consists of zonal flow impinging on a mountain. No analytic solution is known for this case, so the results from the NCAR T213 spectral model are taken to be the "exact" solution. Contour plots of the solution and error after 15 days on a rectangular latitude/longitude projection are shown in Fig. 9. The normalized l_∞, l_1, and l_2 errors are also plotted as a function of time. These results are from the $1734 \times 8 \times 8$ grid.

In Fig. 10, we show the normalized mass, energy, and potential enstrophy integrals as a function of time. These integrals are defined as in [21] and are evaluated using

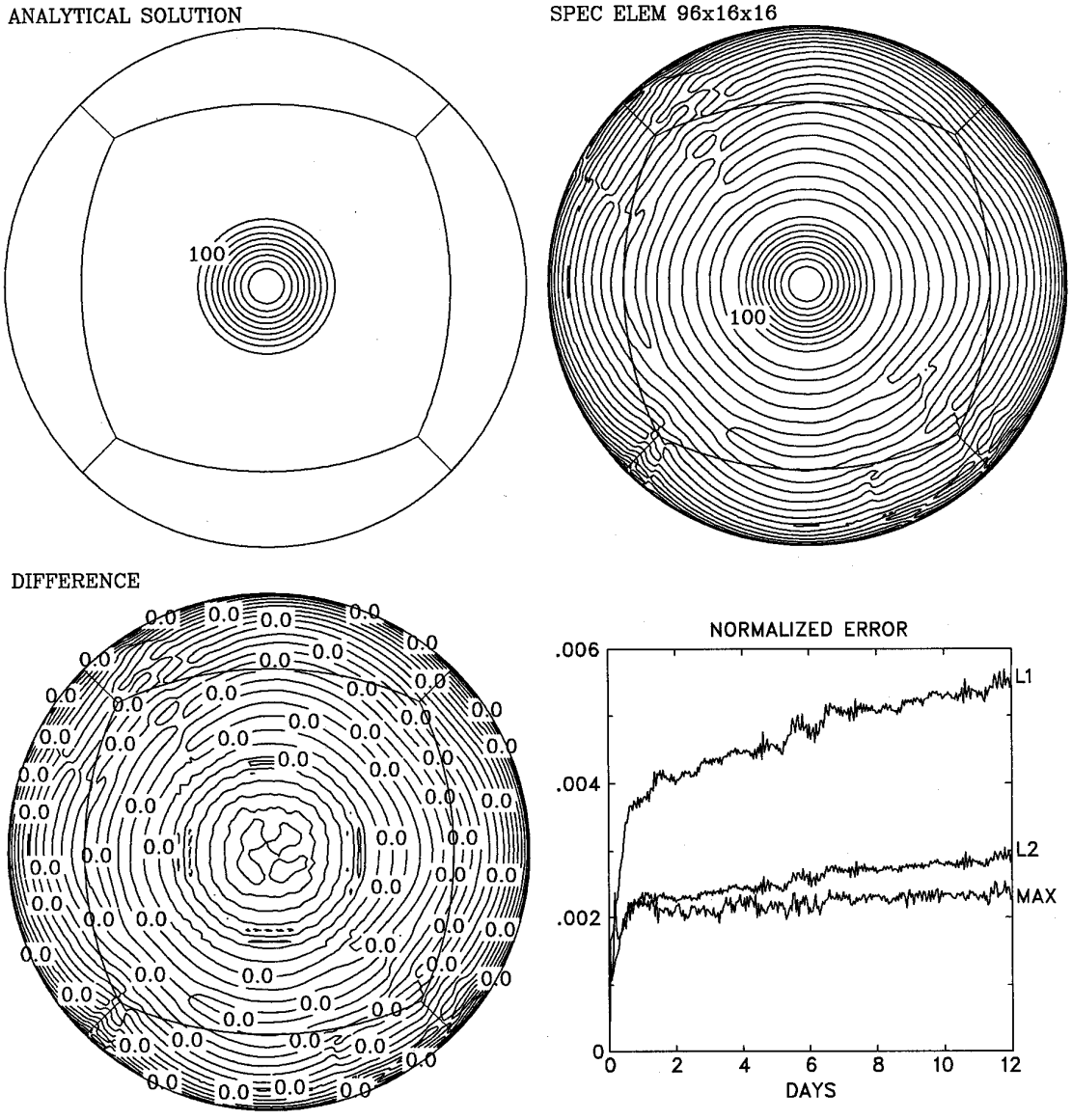

FIG. 5. Case 1. Contour plots of the height field after 12 days, for $\alpha = \pi/4$. The contour interval is 100 m for the analytic and spectral element solution, and 2 m for their difference. The cube projection has been superimposed on the contour plots. The graph shows the l_1, l_2, and l_∞ errors plotted as a function of time, sampled hourly.

the natural Gauss–Lobatto quadrature associated with the spectral element method. The energy and mass are conserved to around eight digits, the potential enstrophy six digits, and the vorticity and divergence integrals in units of seconds^{-1} remain less than 10^{-10} (not shown.)

Test case 6 is the standard $R = 4$ Rossby–Haurwitz wave. This wave pattern moves from west to east without change of shape in the nondivergent barotropic equations. This motion is only approximated in the shallow water equations, so again results from the NCAR T213 spectral model are taken to be the exact solution. Contour plots of the solution and error after 14 days on a rectangular latitude/longitude projection are shown in Fig. 11. The normalized l_∞, l_1, and l_2 errors are also plotted as a function of time. These results are from the $1734 \times 8 \times 8$ grid. The amplitude and phase of the $R = 4$ mode of the meridionally averaged height field for this run is 178.08 m and 176.60°, respectively. These compare well with the values of 177.11 m and 176.73° from the T213 reference solution.

Test case 7 consists of initial conditions taken from 500 mb heights and winds for several atmospheric states. These are 0000 GMT December 21, 1978; 0000 GMT January 16, 1979; and 0000 GMT January 9, 1979, which we refer to as cases 7a, 7b, and 7c, respectively. Contour plots of the

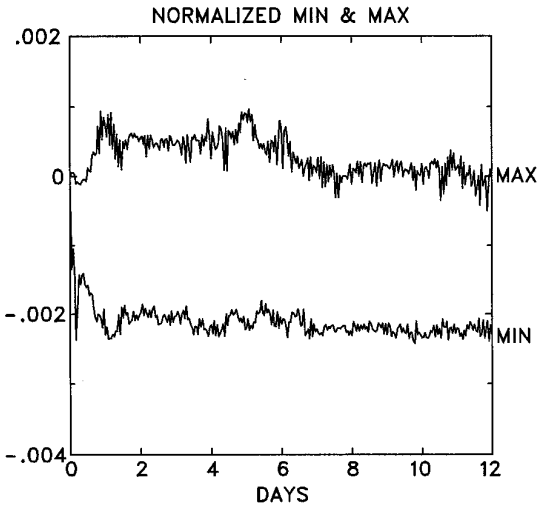

FIG. 6. Case 1. Normalized minimum and maximum of the height field, plotted as a function of time, sampled hourly for $\alpha = \pi/4$. This run used the $96 \times 16 \times 16$ grid.

solution and error after 5 days on a north polar stereographic projection are shown in Fig. 12. Along with the NCAR T213 reference solution, results are shown for three different grids: $150 \times 8 \times 8$, $486 \times 8 \times 8$, and $1734 \times 8 \times 8$. The l_2 errors for these three cases (see Table II) are 0.0037, 0.0013, and 0.00047, respectively. As one can see in the figure, as the mesh is refined and the errors reduced, noticeable improvements in the solution are obtained. Many of the smaller scale features in the reference solution are only well resolved in the $1734 \times 8 \times 8$ result.

The difference between these three solutions and the reference solution, along with the normalized l_∞, l_1, and l_2 errors are plotted in Fig. 13. The convergence of the method to the reference solution is even more visible in these plots than in the previous figure.

7.2. *Summary of Results*

Our results along with results published in [13, 14, 10] are summarized in Table II. For cases 1 through 4, the error listed in the table is a normalized l_2 error between the com-

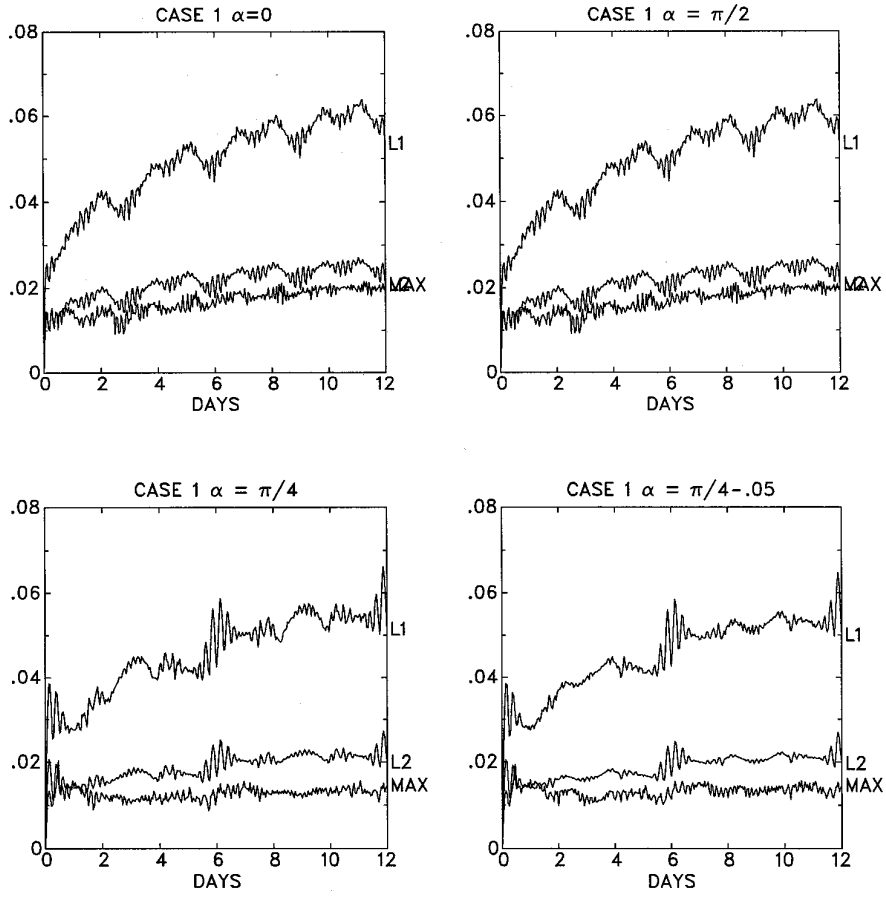

FIG. 7. Case 1. The l_1, l_2, and l_∞ errors plotted as a function of time, sampled hourly, for different values of the parameter α. The cosine bell is advected around the equator for $\alpha = 0$, over the poles for $\alpha = \pi/2$ and through the corners of the elements for $\alpha = \pi/4$. These runs used the $24 \times 16 \times 16$ grid.

TABLE II

Results from the Spectral Element Model, the NCAR Spectral Model [13, 14], and Two Finite Difference Models, the Twisted Icosahedral Grid and the Arakawa–Lamb Model [10]

Method	Grid points	Normalized l_2 Error Case 1	Case 2	Case 3	Case 4
Spec. elem. 24 × 16 × 16	6144	.02	3×10^{-9}	8×10^{-7}	.01
Spec. elem. 96 × 16 × 16	24576	.003	5×10^{-13}	1×10^{-10}	.00004
Spec. elem. 6 × 16 × 16	1536	.3	2×10^{-5}	1×10^{-3}	.28
Spec. elem. 6 × 32 × 32	6144	.03	9×10^{-11}	8×10^{-7}	.0065
NCAR T42	8192	.01	7×10^{-10}	7×10^{-10}	.002
Twisted icosahedral grid	2562	.9	1×10^{-4}	2×10^{-4}	
Twisted icosahedral grid	10242	.5	4×10^{-5}	1×10^{-4}	
Arakawa–Lamb 77 × 44	3168	.9	2×10^{-4}	6×10^{-4}	
Method	Grid points	Case 5	Case 6	Case 7a	
Spec. elem. 150 × 8 × 8	9600	.0019	.0091	.0037	
Spec. elem. 486 × 8 × 8	31104	.00085	.0041	.0013	
Spec. elem. 1734 × 8 × 8	110976	.00086	.0011	.00047	
Spec. elem. 6 × 24 × 24	3456	.0015	.012	.0069	
Spec. elem. 6 × 40 × 40	9600	.00086	.0079	.0030	
Spec. elem. 6 × 56 × 56	18816	.00085	.0053	.0017	
NCAR T42	8192	.0013	.0052	.0037	
NCAR T63	18432	.0012	.0029	.0029	
NCAR T106	51200	.0010	.0013	.0013	
Twisted icosahedral grid	2562	.003	.02		
Twisted icosahedral grid	10242	.001	.006	.005	
Twisted icosahedral grid	40962			.002	
Arakawa–Lamb 77 × 44	3168	.002	.01	.005	

puted and analytic height field. For cases 5 through 7, analytic solutions are not available, and results from the NCAR T213 spectral model are treated as the exact solution. The errors listed are those computed at the final time specified for the test cases. For test cases 1, 2, and 3, the spectral element model was relatively insensitive to α, and all results are given for the worst case α. The results for test case 4 are given for $u_0 = 20$ m/s, and for test case 7 we use initial data from 0000 GMT December 21, 1978 observations.

The use of the NCAR T213 reference solutions as exact solutions presents a minor problem. These solutions do of course contain errors, and this uncertainty has been estimated in [13, 14]. They give the uncertainty in the l_2 norm for test case 6 to be 0.0008 and for case 7 to be 0.00015. Using the same techniques, we estimate the uncertainty in the l_2 error for case 5 to be 0.00072. Changes in the diffusion coefficient (or using no diffusion at all with the T213 run), as well as changes in the time step, result in differences equal to or less than these uncertainties. Thus any model, even if its solution is in fact more accurate than the T213 solution, will not be able to achieve l_2 errors below these uncertainty levels. This can be observed in test case 5, where the l_2 errors for the spectral element model have reached the uncertainty level and no further improvement is seen as the mesh is refined. For test cases 6 and 7, the measured errors are all substantially larger than their reference solution uncertainty.

For the simple test cases with analytic solutions, Table II shows that the two spectral models are significantly more accurate than the two finite difference models. The spectral element model with comparable resolution is not quite as accurate as the NCAR model, but the table shows it is possible to achieve competitive accuracy with a modest increase in resolution. It appears doubtful that the finite difference based models could achieve such error levels at any practical resolution. We should note that test cases 2 and 3 happen to be trivial for the NCAR model since the exact solutions are expressible with only a few spherical harmonics. Thus the NCAR model has errors close to the round-off limit.

The situation is different for the more realistic test cases 5, 6, and 7. Here, all the models listed have very similar

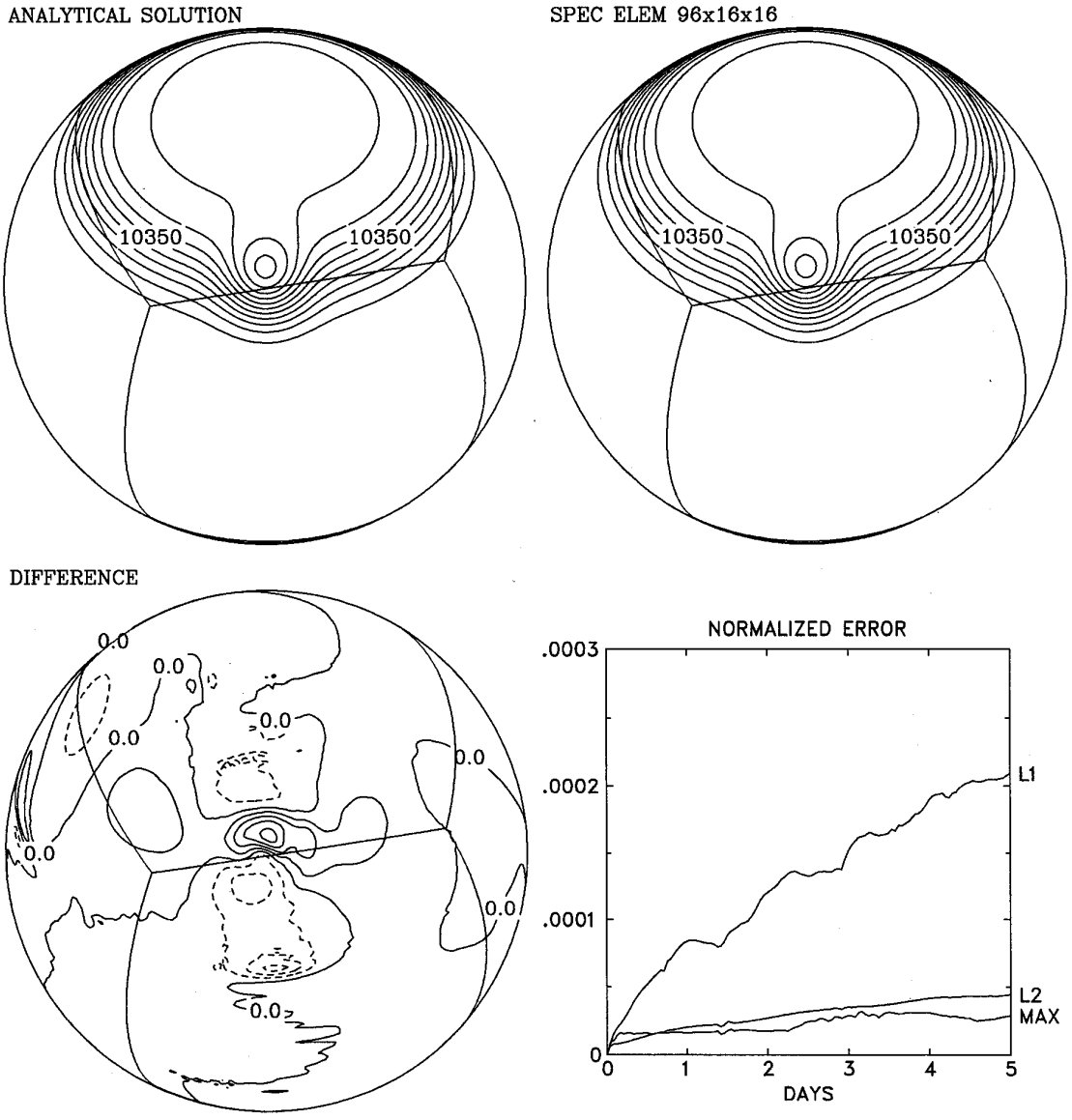

FIG. 8. Case 4. Contour plots of the height field after 5 days, for $u_0 = 20$ m/s. The contour interval is 50 m for the analytic and spectral element solution, and 0.002 m for their difference. The cube projection has been superimposed on the contour plots. The graph shows the l_1, l_2, and l_∞ errors plotted as a function of time, sampled hourly.

error levels, and the convergence rates are much slower. For example, in test case 5 (flow past a mountain), the l_2 errors in Table II from four different models at several resolutions only vary between 0.0008 and 0.003. The spread in the errors for test cases 6 and 7 is only slightly larger. However, this narrow spread does represent a significant difference in the solutions. For example, as discussed above for case 7 and shown in Fig. 13, results with an l_2 error of 0.00047 are visibly superior to those with an l_2 error of 0.0013. Due to the slow convergence rates of the spectral models in these test cases, the finite difference models can achieve accuracies competitive with that of the spectral models, if one is willing to use moderately higher resolutions. A careful look at the table also reveals that the advantage of the NCAR model over spectral elements seen in the analytical test cases is gone. On a per grid-point basis, spectral elements perform slightly better in several cases.

8. CONCLUSIONS

We have implemented the spectral element method for the shallow water equations on the sphere and studied its performance using a now standard set of test cases. The method has two types of mesh refinement: One is expensive

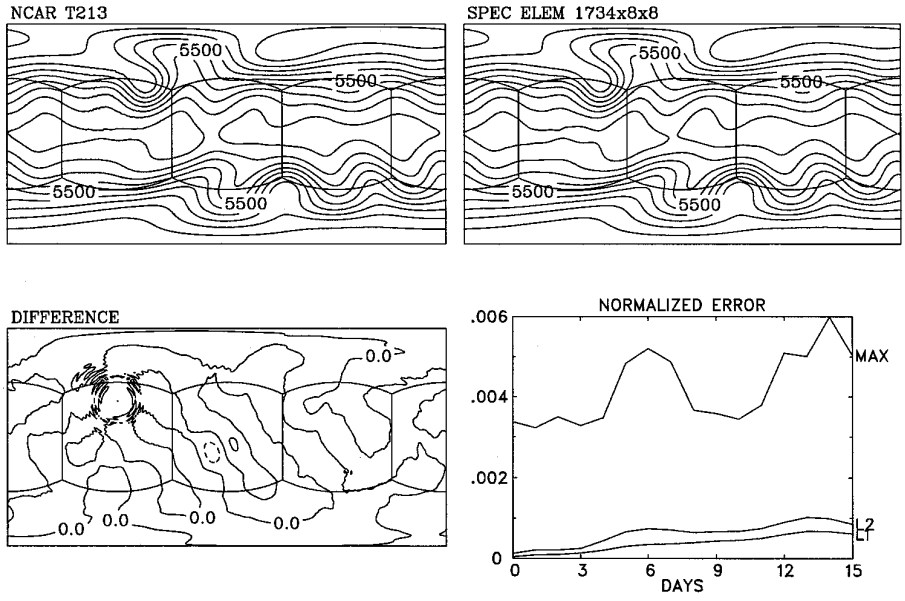

FIG. 9. Case 5. Contour plots of the height field after 15 days. The contour interval is 100 m for the T213 and spectral element solution, and 20 m for their difference. The cube projection has been superimposed on the contour plots. The graph shows the l_1, l_2, and l_∞ errors plotted as a function of time, sampled daily.

but with exponential convergence, while the other is more efficient but has only algebraic convergence. The existence of the test cases along with the reference solutions allowed us to show that these convergence rates hold up even in difficult nonlinear problems and to establish the most efficient configuration of the model.

For the analytic test cases, the spectral element method is extremely accurate. For the more realistic test cases, spectral elements again prove to be quite accurate. However, the spread in the performance of the various models is much smaller than the spread seen in the analytic test cases. For the realistic cases, it appears all the models are capable of achieving similar accuracies at resolutions not prohibitively larger than that used by the NCAR spectral model. Thus for atmospheric problems it might be more productive to choose a model based on features other than accuracy. Two such features where the spectral element model may excel are in local mesh refinement and suitability for parallel computers.

APPENDIX

In this appendix, we give the formulas for the terms used in Section 4. We start with a sphere of radius 1 inscribed

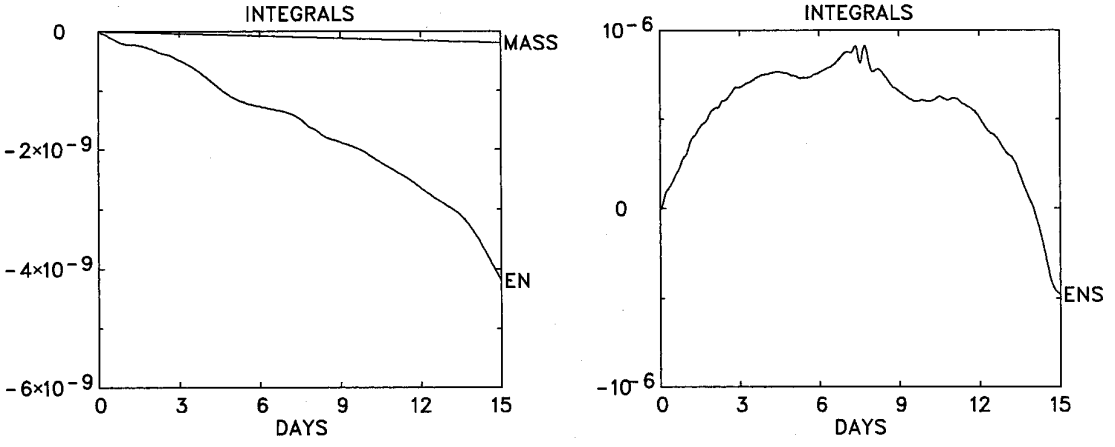

FIG. 10. Case 5. Mass, energy (EN), and potential enstrophy (ENS) plotted as a function of time, sampled hourly.

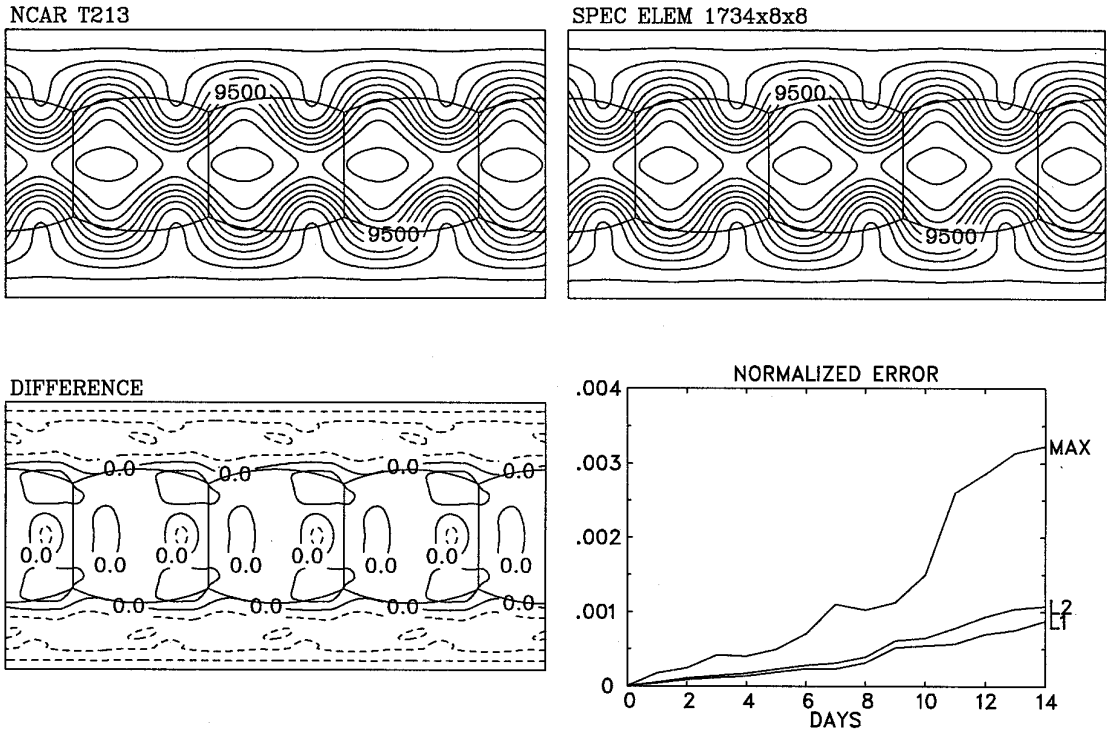

FIG. 11. Case 6. Contour plots of the height field after 14 days. The contour interval is 250 m for the T213 and spectral element solution, and 10 m for their difference. The cube projection has been superimposed on the contour plots. The graph shows the l_1, l_2, and l_∞ errors plotted as a function of time, sampled daily.

in a cube with sides of length 2. The coordinates are as described in Section 4: θ for latitude, λ for longitude, and x and y (each ranging from -1 to 1) are Cartesian coordinates in a given face of the cube. We number the six faces of the cube as follows: faces 1 through 4 have their centers tangent to the sphere at $\theta = 0$ and $\lambda = 0$, $\pi/2$, π, and $3\pi/2$, respectively. Face 5 is centered at the north pole and face 6 is centered at the south pole. We give formulas only for faces 1 and 5, since the others can be easily derived from these by rotating λ or changing the sign of θ. The gnomonic projection is now calculated, and these formulas are then differentiated to derive $\mathbf{D}$, which can be easily inverted to derive $\mathbf{D}^{-1}$:

Face 1:

$$x = \tan\lambda, \quad y = \frac{\tan\theta}{\cos\lambda}$$

$$\mathbf{D} = \cos\theta\cos\lambda \begin{pmatrix} \cos\lambda & 0 \\ -\sin\theta\sin\lambda & \cos\theta \end{pmatrix}$$

$$\mathbf{D}^{-1} = \sec\theta\sec\lambda \begin{pmatrix} \sec\lambda & 0 \\ \tan\theta\tan\lambda & \sec\theta \end{pmatrix};$$

Face 5:

$$x = \tan\theta\sin\lambda, \quad y = -\tan\theta\cos\lambda$$

$$\mathbf{D} = \sin\theta \begin{pmatrix} \cos\lambda & \sin\lambda \\ -\sin\theta\sin\lambda & \sin\theta\cos\lambda \end{pmatrix}$$

$$\mathbf{D}^{-1} = \csc\theta \begin{pmatrix} \cos\lambda & -\csc\theta\sin\lambda \\ \sin\lambda & \csc\theta\cos\lambda \end{pmatrix}.$$

One can easily check that there are no singularities in the above terms since in face 1,

$$-\frac{\pi}{4} \le \theta \le \frac{\pi}{4}, \quad -\pi/4 \le \lambda \le \pi/4;$$

and in face 5,

$$\frac{\pi}{4} \le \theta \le \frac{\pi}{2}.$$

We now give the derivation of the divergence operator $\nabla \cdot \mathbf{v}$ in the cube coordinate system. The formulas for

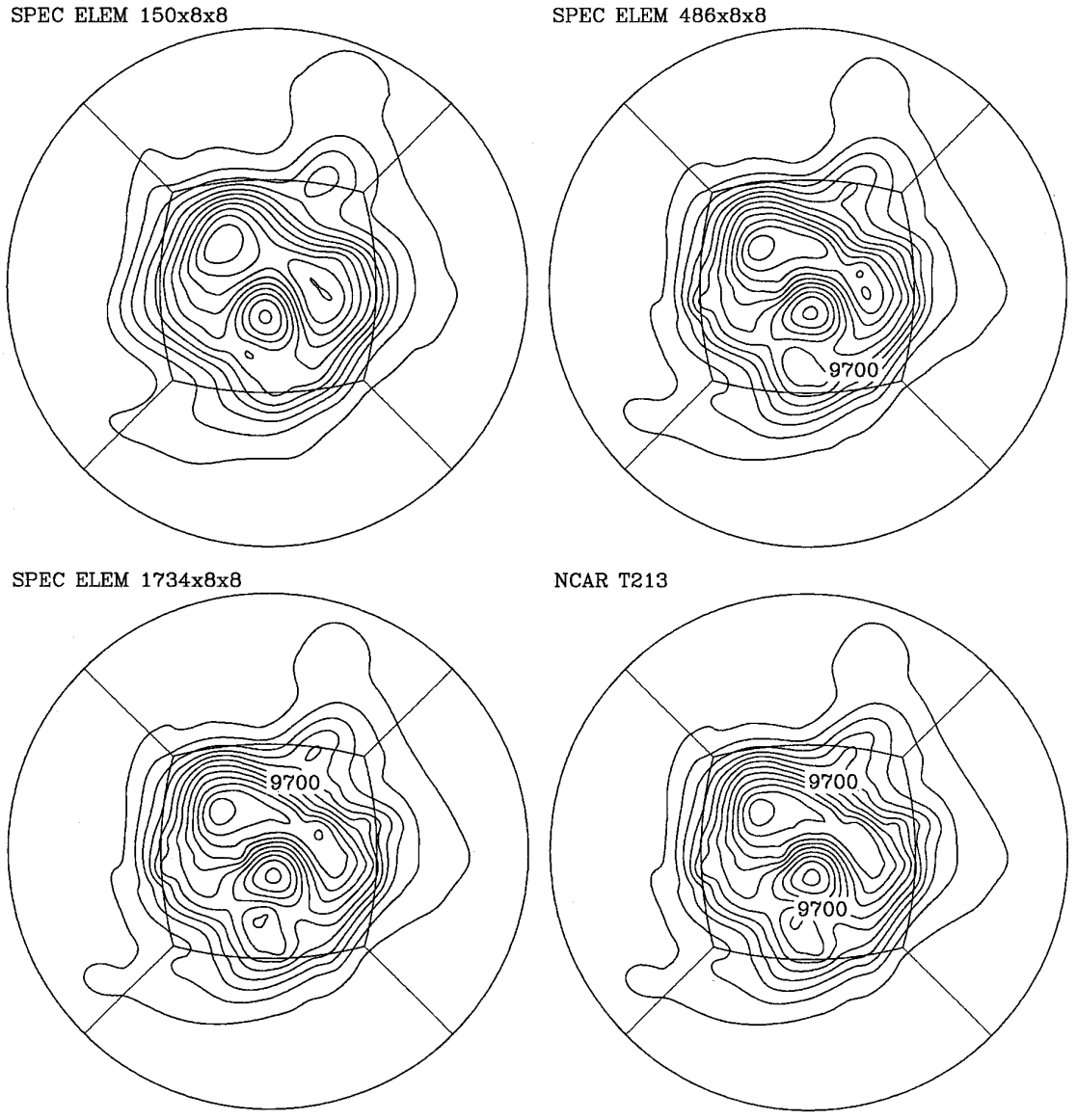

FIG. 12. Case 7a. Contour plots of the height field at various resolutions after 5 days, with a contour interval of 100 m. The cube projection has been superimposed on the contour plots.

the other differential operators given in Section 4 are all derived in a similar manner.

As before, we write a vector $\mathbf{v}$ in the two coordinate systems by

$$\mathbf{v} = u_1\hat{\lambda} + u_2\hat{\theta} = v_1\mathbf{x} + v_2\mathbf{y}.$$

We then write the divergence operator in matrix–vector form in spherical coordinates:

$$\nabla \cdot \mathbf{v} = \sec\theta \begin{pmatrix} \frac{\partial}{\partial\lambda} \\ \frac{\partial}{\partial\theta} \end{pmatrix}^{\mathrm{T}} \left[\begin{pmatrix} 1 & 0 \\ 0 & \cos\theta \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right].$$

Using that

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

and the chain rule

$$\begin{pmatrix} \frac{\partial}{\partial\lambda} \\ \frac{\partial}{\partial\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ 0 & 1 \end{pmatrix} \mathbf{D}^{-\mathrm{T}} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix},$$

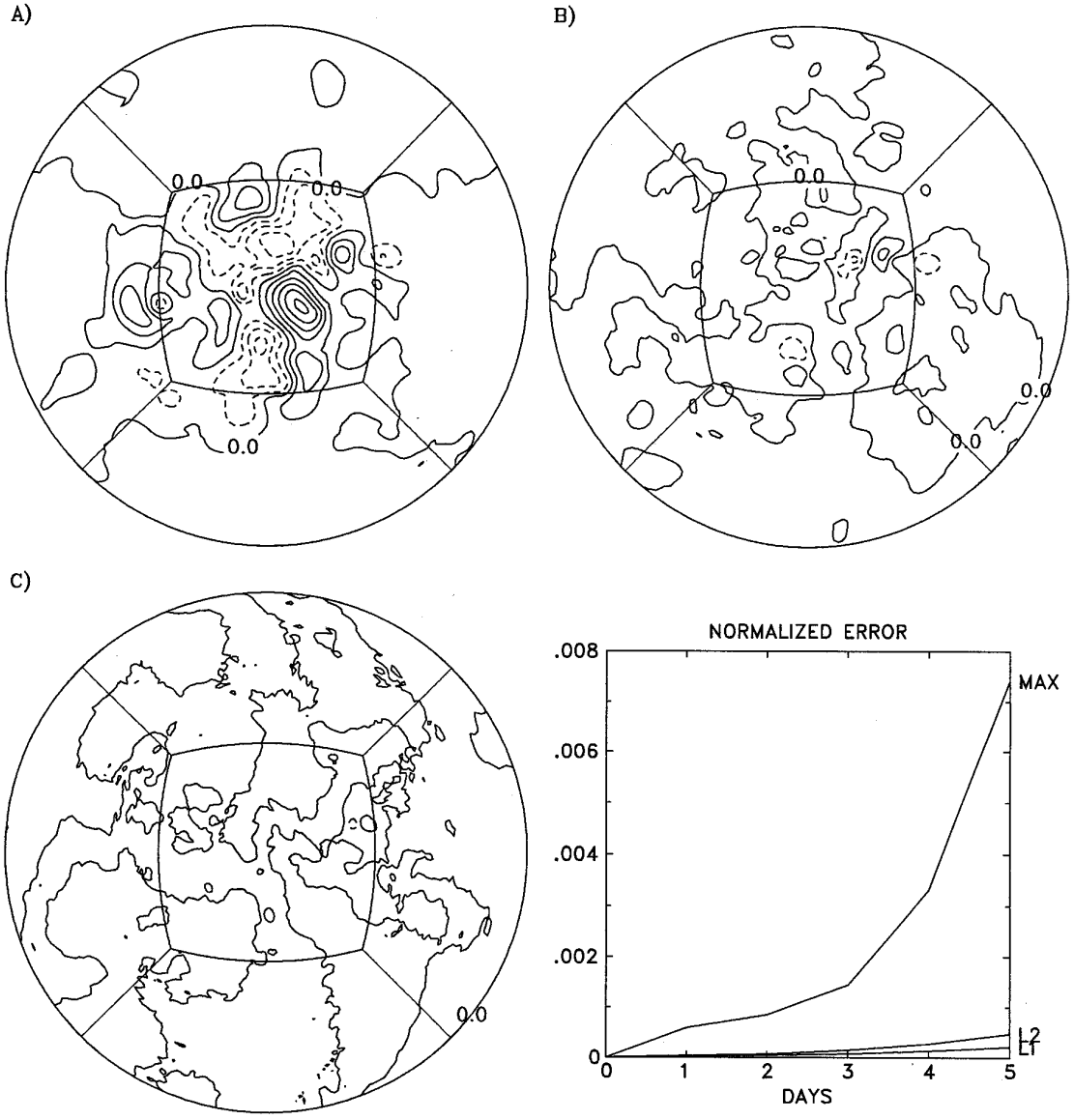

FIG. 13. Case 7a. Contour plots of the difference in the height field between the spectral element method and the NCAR T213 reference solution. Three resolutions are plotted: (A) $150 \times 8 \times 8$; (B) $486 \times 8 \times 8$; (C) $1734 \times 8 \times 8$. The contour interval is 50 m. The cube projection has been superimposed on the contour plots. The graph shows the l_1, l_2, and l_∞ errors for (C) plotted as a function of time, sampled daily.

we have

$$\nabla \cdot \mathbf{v} = \sec\theta \begin{pmatrix} \cos\theta & 0 \\ 0 & 1 \end{pmatrix} \mathbf{D}^{-\mathrm{T}} \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}^{\mathrm{T}} \left[\begin{pmatrix} 1 & 0 \\ 0 & \cos\theta \end{pmatrix} \mathbf{D} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \right].$$

Matrix manipulations lead to the formula used in Section 4,

$$\nabla \cdot \mathbf{v} = \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \mathbf{p}^{\mathrm{T}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

where

$$\mathbf{p} = \sec\theta \begin{pmatrix} \dfrac{\partial}{\partial \lambda} \\ \dfrac{\partial}{\partial \theta} \end{pmatrix}^{\mathrm{T}} \left[\begin{pmatrix} 1 & 0 \\ 0 & \cos\theta \end{pmatrix} \mathbf{D} \right].$$

The Mathematica computer program is then used to evaluate **p** in each face of the cube. The term **q** which appears in the expression for the vorticity is computed in a similar manner. The results for faces 1 and 5 are:

Face 1:

$$\mathbf{p} = -3\cos\theta\cos\lambda\begin{pmatrix}\cos\theta\sin\lambda\\ \sin\theta\end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix}\sin\theta\\ -\cos\theta\sin\lambda\end{pmatrix};$$

Face 5:

$$\mathbf{p} = \frac{3}{2}\sin 2\theta\begin{pmatrix}-\sin\lambda\\ \cos\lambda\end{pmatrix}, \quad \mathbf{q} = -\cos\theta\begin{pmatrix}\cos\lambda\\ \sin\lambda\end{pmatrix}.$$

ACKNOWLEDGMENTS

We would like to thank Ross Heikes and Ruediger Jakob-Chien for helpful discussions about the shallow water test cases. This work was supported by the U.S. Department of Energy, Office of Energy Research, Special Research Grant CHAMMP, Grant DE-FG05-91-ER61219.

REFERENCES

1. C. Bernardi, Y. Maday, and A. T. Patera, ''A New Nonconforming Approach to Domain Decomposition: The Mortar Element Method,'' in *Nonlinear Partial Differential Equations and Their Applications,* edited by H. Brezis and J. L. Lions (Pitman & Wiley, New York, 1992).
2. T. D. Blacker and M. B. Stephenson, *Int. J. Numer. Methods Eng.* **32,** 811 (1991).
3. G. J. Boer *et al*., *J. Geophys. Res.* **97,** 12771 (1992).
4. J. Boyd, *Chebyshev and Fourier Spectral Methods.* Lecture Notes in Engineering, edited by C. A. Brebbia and S. A. Orszag (Springer-Verlag, New York, 1989).
5. J. Boyd, *Houston J. Math.,* to appear.
6. C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, Berlin, 1988).
7. I. Foster, W. Gropp, and R. Stevens, *Mon. Weather. Rev.* **120,** 835 (1992).
8. W. L. Gates, *Bull. Am. Meteorol. Soc.* **73,** 1962 (1992).
9. D. L. Haidvogel, E. Curchitser, M. Iskandarani, R. Hughes, and M. Taylor, *Atmosphere-Ocean,* to appear.
10. r. Heikes and D. A. Randall, *Mon. Weather Rev.,* **123,** 1862 (1995).
11. R. D. Henderson and G. E. Karniadakis, *J. Comput. Phys.* **122,** 191 (1995).
12. M. Iskandarani, D. L. Haidvogel, and J. P. Boyd, *Int. J. Numer. Methods Fluids* **20,** 393 (1995).
13. R. Jakob, J. J. Hack, and D. L. Williamson, NCAR Technical Note NCAR/TN-388+STR (National Center for Atmospheric Research, Boulder, Colorado, 1993).
14. R. Jakob-Chien, J. J. Hack, and D. L. Williamson, *J. Comput. Phys.* **119,** 164 (1995).
15. J. Levin, M. Iskandarani, and D. Haidvogel, unpublished.
16. H. Ma, *J. Comput. Phys.* **109,** 133 (1993).
17. Y. Maday and A. T. Patera, ''Spectral Element Methods for the Incompressible Navier–Stokes Equations,'' in *State of the Art Surveys in Computational Mechanics,* edited by A. K. Noor (ASME, New York, 1988).
18. A. T. Patera, *J. Comput. Phys.* **54,** 468 (1984).
19. M. Taylor, *SIAM J. Numer. Anal.* **32,** 667 (1995).
20. H. Vandeven, *J. Sci. Comput.* **6,** 159 (1991).
21. D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, *J. Comput. Phys.* **102,** 211 (1992).