Development of a Scalable Global Discontinuous Galerkin Atmospheric Model

J.M. Dennis, R.D. Nair and H.M. Tufo[*†]
Scientific Computing Division, National Center for Atmospheric Research
1850 Table Mesa Dr., Boulder, CO 80305, USA.
E-mail: {dennis, rnair, tufo}@ucar.edu
[†] Corresponding author

M. Levy
Department of Applied Mathematics, University of Colorado
Engineering Center - 526UCB, Boulder, CO 80309, USA.
E-mail: levym@colorado.edu

T. Voran
Department of Computer Science, University of Colorado
Engineering Center - 430UCB, Boulder, CO 80309, USA.
E-mail: theron.voran@colorado.edu

Abstract: An efficient and scalable global discontinuous Galerkin atmospheric model (DGAM) on the sphere is developed. The continuous flux form of the nonlinear shallow water equations on the cubed-sphere (in curvilinear coordinates) are developed. Spatial discretization is a nodal basis set of Legendre polynomials. Fluxes along internal element interfaces are approximated by a Lax-Friedrichs scheme. A third-order strong stability preserving Runge-Kutta scheme is applied for time integration. The standard shallow water test suite of Williamson et al. (1992) is used to validate the model. It is observed that the numerical solutions are accurate, the model conserves mass to machine precision, and there are no spurious oscillations in a test case where zonal flow impinges a mountain. The serial execution time of the high-order nodal DG scheme presented here is half that of the modal version DG scheme. Development time was substantially reduced by building the model in the High Order Method Modeling Environment (HOMME) developed at the National Center for Atmospheric Research (NCAR). Performance and scaling data for the steady state geostrophic flow problem Williamson et al. (1992) is presented. Sustained performance of 8% of peak is observed on 2048 processor of a IBM Blue Gene/L supercomputer.

Keywords: discontinuous Galerkin method; high-order methods; cubed-sphere; climate and atmospheric modeling, parallel computing; high-performance computing.

Biographical notes: [*]H.M. Tufo is also an Associate Professor of Computer Science at the University of Colorado, Boulder

1 INTRODUCTION

The spectral transform method has played a central role in global climate modeling for the past two decades (Washington and Parkinson (2005)). However, the scalability of the spectral transform method is inherently limited because it requires non-local communication. As a result, the spectral transform method is poorly suited to take advantage of a new generation of massively-parallel distributed-memory computers with $\mathcal{O}(10,000)$ processors. In order to address this issue, several groups have adopted scalable high-order methods such as spectral element (SE) method to build the next generation of climate models (Loft et al. (2001); Giraldo et al. (2004); Fournier et al. (2004)). Spectral elements have numerous attractive features such as exponential convergence, computational efficiency, scalability, and the ability to handle complex geometries. However, a major disadvantage of SE atmospheric models is a lack of

conservation. For climate and atmospheric chemistry applications, conservation of integral invariants such as mass and energy as well as monotonicity of the solutions are crucial. To date there have been several efforts to develop conservative atmospheric models but they are all based on classic low-order finite-volume methods (Lin and Rood (1997)).

The high-order discontinuous Galerkin (DG) method is ideally suited for atmospheric numerical modeling because it is inherently conservative and can incorporate monotonic limiters. Moreover, it retains all the advantages of the SE method. The DG method is a hybrid approach combining the finite-volume and the finite-element methods, exploiting the merits of both. DG methods became popular following the work of Cockburn and Shu (1989, 2001).

The High-Order Method Modeling Environment (HOMME (2004)) is a global atmospheric modeling framework developed at the National Center for Atmospheric Research (NCAR), and is based on highly scalable SE method employing the cubed-sphere geometry (Loft et al. (2001); Thomas and Loft (2002)). In this paper we present an overview of the DG shallow water model based on *nodal* discretization, as implemented in HOMME. We compare the results with the *modal* version of the DG shallow water model on the cubed-sphere developed by Nair et al. (2005a,b). Moreover, we present numerical results on the Williamson et al. (1992) shallow water test suite, and show performance results for a selected test case on 2048 processors of the IBM Blue Gene supercomputer.

2 GLOBAL SHALLOW WATER MODEL

As described in Nair et al. (2005a), the sphere is decomposed into six identical regions (Figure 1), obtained by central (gnomonic) projection of the faces of the inscribed cube onto the spherical surface Sadourny (1972); Ronchi et al. (1996). Each of the six local coordinate systems is free of singularities and employ identical metric terms, creating a non-orthogonal curvilinear coordinate system on the sphere. Figure 1 shows a cubed-sphere with a total of 150 elements, where each element consists of a Gauss-Lobatto-Legendre (GLL) grid. In Figure 1, cube-face edges are marked by thick lines, while the intersection of thin lines mark each of the 8×8 GLL points.

Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be the covariant base vectors of the transformation between inscribed cube and spherical surface. Let $\mathbf{v} = \mathbf{v}(\lambda, \theta)$ be the horizontal velocity vector specified on the sphere with longitude λ and latitude θ. The components of the covariant vectors are given by $u_1 = \mathbf{v} \cdot \mathbf{a}_1$, $u_2 = \mathbf{v} \cdot \mathbf{a}_2$ and the corresponding contravariant components are expressed as $\mathbf{v} = u^1 \mathbf{a}_1 + u^2 \mathbf{a}_2$. The metric tensor of the transformation is defined as $G_{ij} \equiv \mathbf{a}_i \cdot \mathbf{a}_j$. Covariant and contravariant vectors are related through

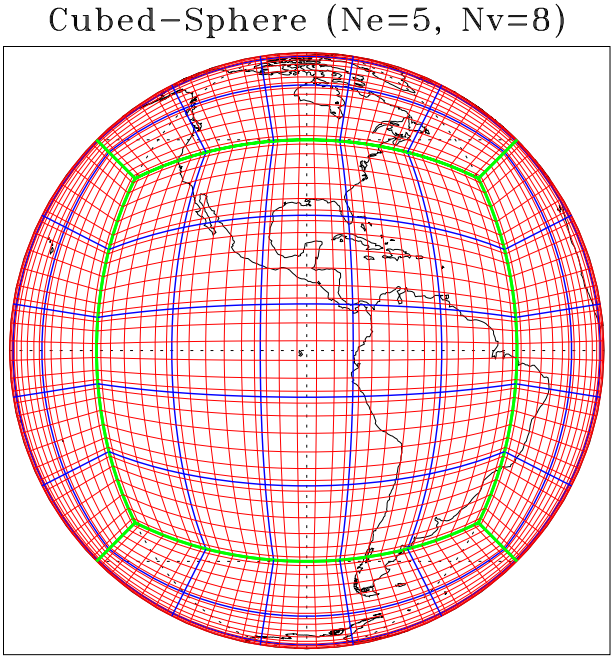

Figure 1 *A cubed-sphere with 150 elements ($N_e = 5$) that span the surface of the sphere. Each element contains 8×8 Gauss-Lobatto-Legendre (GLL) quadrature points ($N_v = 8$).*

the metric tensor G_{ij} such that $u_i = G_{ij} u^j$, $u^i = G^{ij} u_j$, where $G^{ij} = (G_{ij})^{-1}$ and $G = \det(G_{ij})$. For equiangular coordinates (x^1, x^2), the metric tensor for all six faces of the cube is

$$G_{ij} = \frac{1}{r^4 \cos^2 x^1 \cos^2 x^2} \times \begin{bmatrix} 1 + \tan^2 x^1 & -\tan x^1 \tan x^2 \\ -\tan x^1 \tan x^2 & 1 + \tan^2 x^2 \end{bmatrix} = A^T A, \qquad (1)$$

where $r = (1 + \tan^2 x^1 + \tan^2 x^2)^{1/2}$ and $\sqrt{G} = 1/r^3 \cos^2 x^1 \cos^2 x^2$. The matrix A in (1) can be used for transforming $\mathbf{v}$ with spherical velocity components (u, v) to the local cube-face components (u^1, u^2) and vice versa, as follows (Nair et al. (2005a)).

$$A \begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix},$$
$$A = \begin{bmatrix} \cos\theta \, \partial\lambda/\partial x^1 & \cos\theta \, \partial\lambda/\partial x^2 \\ \partial\theta/\partial x^1 & \partial\theta/\partial x^2 \end{bmatrix}. \qquad (2)$$

The six local Cartesian coordinate systems (x^1, x^2) that span the surface of the sphere in Figure 1 are based on equiangular central projection such that $x^1 = x^1(\lambda, \theta)$, $x^2 = x^2(\lambda, \theta)$, and $-\pi/4 \le x^1, x^2 \le \pi/4$.

2.1 Shallow water equations

We consider the flux form shallow water equations in curvilinear coordinates as described in Sadourny (1972). The governing equations for an inviscid flow of a thin layer of fluid in 2D are the horizontal momentum and continuity

equations for the height h. Here, h is the depth of the fluid and it is related to the free surface geopotential height (above sea level) $\Phi = g\,(h_s + h)$, where h_s denotes the height of the underlying mountains and g is the gravitational acceleration.

In curvilinear coordinates, the continuity and momentum equations for the shallow water system may be written as follows (Sadourny (1972); Nair et al. (2005b)),

$$\frac{\partial}{\partial t}(\sqrt{G}\,h) + \frac{\partial}{\partial x^1}(\sqrt{G}\,u^1 h) + \frac{\partial}{\partial x^2}(\sqrt{G}\,u^2 h) = 0, \qquad (3)$$

$$\frac{\partial u_1}{\partial t} + \frac{\partial}{\partial x^1}E = \sqrt{G}\,u^2(f+\zeta), \qquad (4)$$

$$\frac{\partial u_2}{\partial t} + \frac{\partial}{\partial x^2}E = -\sqrt{G}\,u^1(f+\zeta), \qquad (5)$$

where

$$E = \Phi + \frac{1}{2}\,(u_1\,u^1 + u_2\,u^2), \quad \zeta = \frac{1}{\sqrt{G}}\left[\frac{\partial u_2}{\partial x^1} - \frac{\partial u_1}{\partial x^2}\right],$$

$f = 2\omega\,\sin\theta$ is the Coriolis parameter and ω is the rotation rate of the earth.

The system (3-5) may be expressed in the following flux form,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x^1}\mathbf{F}_1(\mathbf{U}) + \frac{\partial}{\partial x^2}\mathbf{F}_2(\mathbf{U}) = \mathbf{S}(\mathbf{U}), \qquad (6)$$

where

$$\begin{aligned}
\mathbf{U} &= \left[\sqrt{G}\,h, u_1, u_2\right]^T \\
\mathbf{F}_1 &= \left[\sqrt{G}hu^1, E, 0\right]^T \\
\mathbf{F}_2 &= \left[\sqrt{G}hu^2, 0, E\right]^T,
\end{aligned}$$

with the source term

$$\mathbf{S} = \left[0, \sqrt{G}u^2(f+\zeta), -\sqrt{G}\,u^1(f+\zeta)\right]^T.$$

3 DG FORMULATION

For simplicity, we proceed with a scalar component of (6) to describe the DG discretization.

$$\frac{\partial U}{\partial t} + \nabla\cdot\mathbf{F}(U) = S(U), \quad \text{in} \quad \mathcal{D}\times(0,T), \qquad (7)$$

for all $(x^1,x^2) \in \mathcal{D}$ with initial condition $U_0(x^1,x^2) = U(x^1,x^2,t=0)$. In (7), $\mathbf{F} = (F_1,F_2)$ is the flux function, $U = U(x^1,x^2,t)$, and $\nabla \equiv (\partial/\partial x^1, \partial/\partial x^2)$ is the gradient operator.

The computational domain $\mathcal{D}$ is the surface of the cubed-sphere, spanning six identical non-overlapping subdomains (faces) such that $\mathcal{D} = \bigcup_{\nu=1}^{6}\Omega^\nu$. Therefore, it is only necessary to consider the discretization for a single subdomain Ω as the procedure can be analogously extended to the remaining subdomains. Consider a subdomain Ω which is partitioned into $N_e \times N_e$ rectangular non-overlapping elements Ω_{ij}; $i,j = 1,2,\ldots,N_e$, such that

$$\begin{aligned}
\Omega_{ij} = \{(x^1,x^2) \quad | \quad & x^1 \in [x^1_{i-1/2}, x^1_{i+1/2}], \\
& x^2 \in [x^2_{j-1/2}, x^2_{j+1/2}]\}.
\end{aligned} \qquad (8)$$

Thus, the total number of elements on the cubed-sphere is $M = 6\times N_e^2$.

The size of an element Ω_{ij} is determined by $\Delta x^1_i = (x^1_{i+1/2} - x^1_{i-1/2})$ and $\Delta x^2_j = (x^2_{j+1/2} - x^2_{j-1/2})$ in the x^1 and x^2-directions, respectively. For $t > 0$, consider an element Ω_{ij} in the partition of Ω and an approximate solution $U_h = U_h(x^1,x^2,t)$ belongs to the finite dimensional space $\mathcal{V}_h(\Omega)$. Multiplication of (7) by a test function $\varphi_h = \varphi_h(x^1,x^2) \in \mathcal{V}_h$ and integration over the element Ω_{ij} results in a weak Galerkin formulation of the problem.

$$\begin{aligned}
\frac{\partial}{\partial t}\int_{\Omega_{ij}} U_h\,\varphi_h\,d\Omega \quad &- \quad \int_{\Omega_{ij}} \mathbf{F}(U_h)\cdot\nabla\varphi_h\,d\Omega + \\
\int_{\partial\Omega_{ij}} \mathbf{F}(U_h)\cdot\mathbf{n}\,\varphi_h\,ds \quad &= \quad \int_{\Omega_{ij}} S(U_h)\,\varphi_h\,d\Omega,
\end{aligned} \qquad (9)$$

where $\mathbf{n}$ is the outward-facing unit normal vector on the element boundary $\partial\Omega_{ij}$.

Along the boundaries of an element (internal interfaces) $\partial\Omega_{ij}$, the function U_h is discontinuous and the boundary integral (third term in (9)) is not uniquely defined. Therefore, the analytic flux $\mathbf{F}(U_h)\cdot\mathbf{n}$ in (9) must be replaced by a numerical flux $\widehat{F}(U_h^-, U_h^+)$. The numerical flux resolves the discontinuity along the element edges and provides the only mechanism by which adjacent elements interact. For simplicity, the Lax-Friedrichs numerical flux as considered in Nair et al. (2005a,b) is chosen for the present study, given by

$$\begin{aligned}
\widehat{F}(U_h^-, U_h^+) \quad = \quad & \frac{1}{2}\left[(\mathbf{F}(U_h^-) + \mathbf{F}(U_h^+))\cdot\mathbf{n} - \right. \\
& \left. \alpha(U_h^+ - U_h^-)\right],
\end{aligned} \qquad (10)$$

where U_h^- and U_h^+ are the left and right limits of the discontinuous function U_h evaluated at the element interface, α is the upper bound for the absolute value of eigenvalues of the flux Jacobian $\mathbf{F}'(U)$ in the direction $\mathbf{n}$. For the shallow water system (6), the local maximum values of α in x^1 and x^2-directions for each element Ω_{ij} are defined as (Nair et al. (2005b)) $\alpha^1 = \max\left(|u^1| + \sqrt{\Phi\,G^{11}}\right)$, $\alpha^2 = \max\left(|u^2| + \sqrt{\Phi\,G^{22}}\right)$. Treatment of flux terms and vector quantities at the cube-face edges needs special attention, and it is discussed in Nair et al. (2005a).

3.1 Discretization

For each element Ω_{ij}, define the local variables $\xi^k = 2(x^k - x^k_i)/\Delta x^k_i$, with $x^k_i = (x^k_{i+1/2} + x^k_{i-1/2})/2$ and $k = 1,2$, denote the x^1, x^2-directions, respectively. By using these

relations, an element Ω_{ij} is mapped onto the reference element $\widetilde{\Omega}_{ij} \equiv [-1,1] \otimes [-1,1]$.

An important aspect of the DG discretization is the choice of an appropriate set of basis functions (polynomials) that span $\mathcal{V}_h$. While Nair et al. (2005a,b) used a modal expansion basis, in this work we use a high-order nodal basis set due to its computational efficiency. Nodal basis sets are popularly used in SE methods (Deville et al. (2002)). Recently Giraldo et al. (2002) developed a nodal shallow water model on an icosahedral grid. The choice of a particular type of basis is problem dependent and their relative merits are discussed in Karnoadakis and Sherwin (1999).

The nodal basis set is constructed using Lagrange-Legendre polynomials $h_\ell(\xi)$, $\xi \in [-1,1]$, with roots at Gauss-Lobatto quadrature points. The basis functions are defined by

$$h_\ell(\xi) = \frac{(\xi-1)(\xi+1)\,L'_N(\xi)}{N(N+1)\,L_N(\xi_\ell)\,(\xi-\xi_\ell)} \tag{11}$$

where $L_N(\xi)$ is the Legendre polynomial of degree N. Because the basis function is a Lagrange polynomial it has the property $h_p(\xi_q) = \delta_{pq}$; where $\delta_{pq} = 1$, if $p = q$ and $\delta_{pq} = 0$, if $p \neq q$. Discrete orthogonality of $h_\ell(\xi)$ can be established through the following quadrature

$$\int_{-1}^{1} f(\xi) d\xi \approx \sum_{i=0}^{N} f(\xi_i) w_i, \tag{12}$$

where w_i is the weight associated with GLL quadrature rule, and it is defined to be

$$w_i = \frac{2}{N(N+1)\,[L_N(\xi_i)]^2}.$$

Note that the quadrature (12) is *exact* for polynomials of degree up to $2N-1$. Consequently, discrete orthogonality of the basis function $h_\ell(\xi)$ can be derived using the properties of the Lagrange polynomial as follows (Deville et al. (2002))

$$\int_{-1}^{1} h_i(\xi)\,h_j(\xi) d\xi = \sum_{l=0}^{N} h_i(\xi_l)\,h_j(\xi_l)\,w_l = w_i\,\delta_{ij}. \tag{13}$$

In the two-dimensional (2D) (ξ^1, ξ^2) coordinate system, the test function (φ_h) as well as the the approximate solution U_h are expanded in terms tensor-product functions from the basis set. Thus,

$$U_h(\xi^1,\xi^2) = \sum_{\ell=0}^{N}\sum_{m=0}^{N} U_{\ell m}\,h_\ell(\xi^1)\,h_m(\xi^2), \qquad \text{for} \quad -1 \le \xi^1, \xi^2 \le 1 \tag{14}$$

where

$$h_\ell(\xi^k) = \frac{(\xi^k-1)(\xi^k+1)\,L'_N(\xi^k)}{N(N+1)\,L_N(\xi^k_\ell)\,(\xi^k-\xi^k_\ell)}, \quad k \in \{1,2\}. \tag{15}$$

The weak formulation (9) is simplified by mapping the elements onto the reference element $\widetilde{\Omega}_{ij}$, and utilizing (13,14). The final approximation of (7) for an element Ω_{ij} takes the form,

$$\frac{d}{dt}U_{\ell m} = \frac{4}{\Delta x^1_i\,\Delta x^2_j\,w_\ell w_m}\left[I_G + I_F + I_S\right], \tag{16}$$

where I_G, I_F, and I_S are the integrals corresponding to the gradient, flux, and source terms, respectively, in (9). The coefficient terms in the right hand side of (16) are the elements of the inverted mass-matrix that are block diagonal for the 2D problems by virtue of the orthogonality relation (13). Integrals appearing in (16) are computed with an accurate GLL quadrature rule.

A remarkable difference between (16) and corresponding modal version (Nair et al. (2005b)) is the absence of spectral coefficients. The nodal DG scheme is potentially more computationally efficient because it relies on solutions in physical space, obviating the need to transform between spectral and physical space, which are required by the modal DG scheme (Nair et al. (2005a)).

3.2 Time Integration

The semi-discretized equation (16) for the shallow water system takes the following form

$$\frac{d}{dt}\,\mathbf{U} = \mathbf{L}(\mathbf{U}) \quad \text{in} \quad (0,T). \tag{17}$$

Time integration of the shallow water (SW) equations is performed by solving a system of ODEs (17). We employ a strong stability preserving Runge-Kutta scheme (also known as total variation diminishing Runge-Kutta) that does not introduce spurious oscillations for smooth problems (Gottlieb et al. (2001)). These schemes are widely used for solving (17) in the DG discretization, and are given as follows

$$\begin{aligned}
\mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\,\mathbf{L}(\mathbf{U}^n) \\
\mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}[\mathbf{U}^{(1)} + \Delta t\,\mathbf{L}(\mathbf{U}^{(1)})] \\
\mathbf{U}^{n+1} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}[\mathbf{U}^{(2)} + \Delta t\,\mathbf{L}(\mathbf{U}^{(2)})],
\end{aligned} \tag{18}$$

where the superscripts n and $n+1$ denote time levels t and $t+\Delta t$, respectively.

The shallow water test cases considered here are relatively "smooth" and numerical integration performed is on the order of days, therefore, we did not use any limiter or filter with time integration scheme. For non-smooth problems (i.e. solution containing shocks and discontinuities) a monotonic limiter is required.

4 NUMERICAL EXPERIMENTS

Our DG scheme has been extensively tested using various initial conditions. Williamson et al. (1992) proposed

a suite of standard tests for the shallow water equations on the sphere. These idealized tests of varying complexity include experiments with north-south symmetry, balanced steady-state flows, and extreme gradients.

We have employed a variety of grid resolutions with total number of elements $M = 6 \times N_e^2$ on the cubed-sphere and each element consisting of $N_v \times N_v$ GLL points. To visually compare our numerical results with existing solutions we use bilinear interpolations to map onto a 128×65 longitude-latitude grid (approximately equal to the T42 resolution).

4.1 Advection on the cubed-sphere

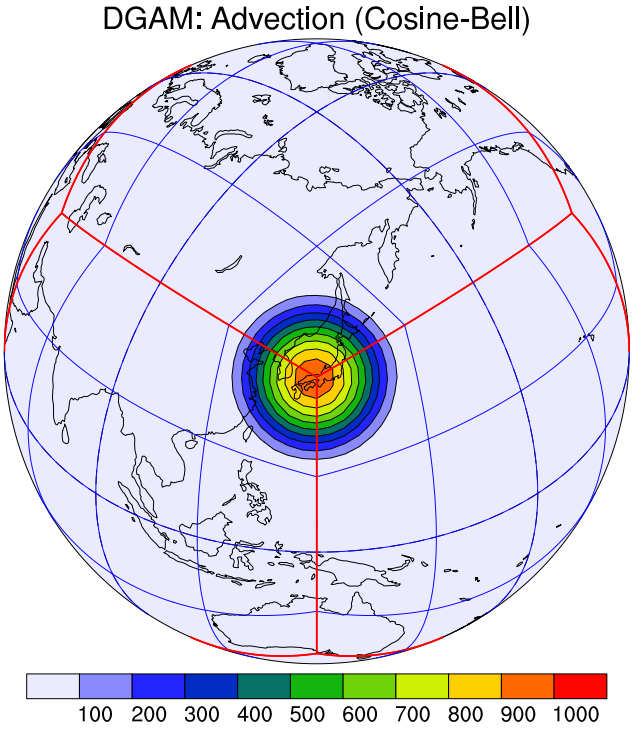

Figure 2 *Results on an orthographic projection for cosine-bell advection along the north-east direction when the bell center reaches a corner (vertex) of the cubed-sphere. Each element ($N_e = 4$) consists of 10×10 GLL points.*

Solid-body rotation is commonly used to test an advection scheme over the sphere. The first shallow water equation test case (SW1) suggested by Williamson et al. (1992) advects a "cosine-bell" along a great-circle trajectory on the sphere for 12 days. Note that here only the continuity equation (3) of the shallow water system is solved. The initial conditions are specified as follows,

$$h(\lambda,\theta) = \begin{cases} (h_0/2)\,[1+\cos(\pi r_d/r_0)] & \text{if} \quad r_d < r_0 \\ 0 & \text{if} \quad r_d \geq r_0, \end{cases} \tag{19}$$

where r_d is the great-circle distance between (λ,θ) and the bell center. The cosine bell of radius $r_0 = R/3$, where $R = 6.37122 \times 10^6$ m is the Earth's radius, is initially centered at $(3\pi/2, 0)$ and corresponds to the central point on a lateral face of the cubed-sphere (see Figure 1). The maximum height of the cosine bell is $h_0 = 1000$ m. The velocity components of the advecting wind field are

$$\begin{aligned} u &= u_0\,(\cos\alpha_0\,\cos\theta + \sin\alpha_0\,\cos\lambda\sin\theta), \\ v &= -u_0\,\sin\alpha_0\sin\lambda, \end{aligned}$$

where $u_0 = 2\pi R/(12 \text{ days})$, α_0 is the angle between the axis of solid-body rotation and the polar axis of the spherical coordinate system. When α_0, is either zero or $\pi/2$ the flow direction is along the equator in north-south (meridional) direction. For the cubed-sphere, flow in the north-east direction ($\alpha_0 = \pi/4$) is more challenging, because the cosine-bell passes over four vertices and two edges of the cube during a complete revolution.

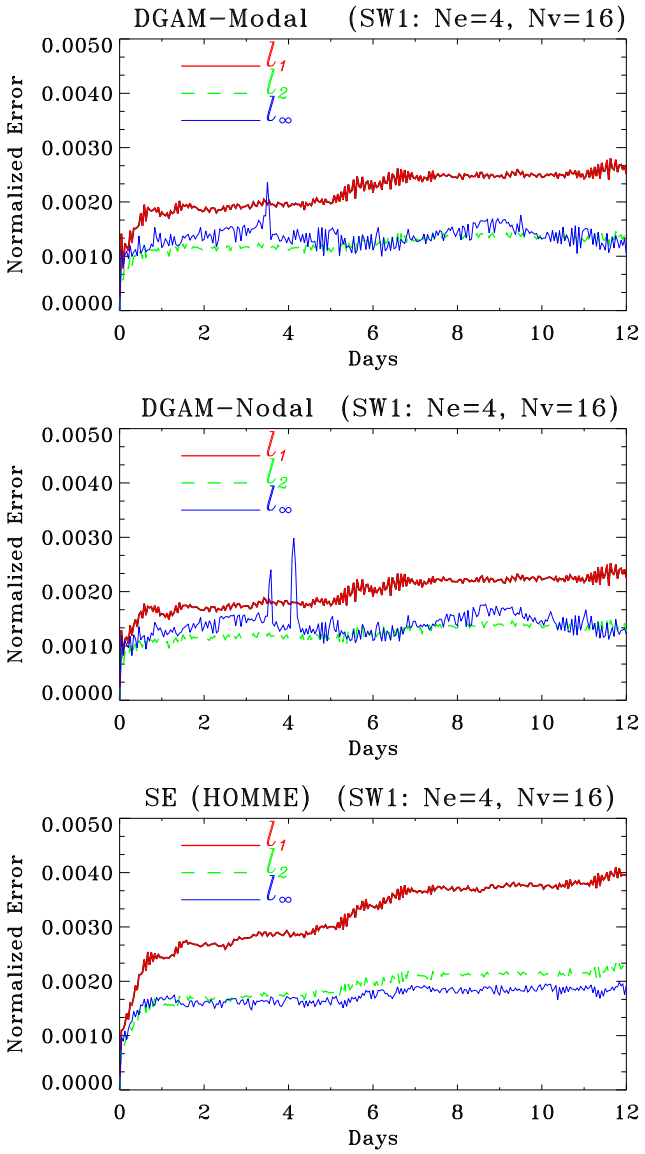

Figure 3 *Time traces of normalized ℓ_1, ℓ_2 and ℓ_∞ errors for the cosine-bell advection test. The top two panels show results with modal and nodal versions of the DG scheme, respectively. The bottom panel shows corresponding reference results with the spectral-element model (HOMME).*

Figure 2 shows results on an orthographic projection for the solid-body rotation for $\alpha_0 = \pi/4$. Nair et al. (2005a) extensively studied the transport process over the cubed-sphere with a modal version of DG advection scheme. They showed that the cosine-bell passes smoothly over the vertices and edges, while maintaining high-order accuracy. The nodal version presented here produces almost identical results. Figure 3 illustrates the standard ℓ_1, ℓ_2 and ℓ_∞ errors norms as a function of time for the modal and nodal DG schemes and the SE scheme for a 12 day simulation. The grid resolution for this test are $N_e = 4$ (i.e., 96 elements total), $N_v = 16$, and $\Delta t = 30$s.

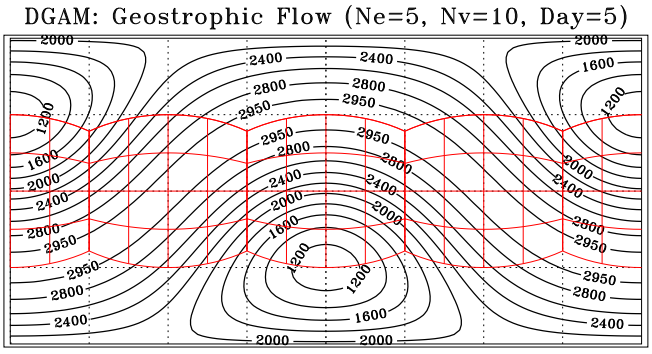

Figure 4 *Numerical solution (height fields in meter) for steady state geostrophic flow test (SW test case 2) after 5 days of integration. A cubed-sphere with parameters $N_e = 5$ and $N_v = 10$ were used for this test.*

The top and central panels in Figure 3 illustrate the error norms for the modal and nodal version DG, respectively. The ℓ_1 and ℓ_2 errors are virtually identical while ℓ_∞ error for the modal version is smaller that of the nodal version. For reference, we include the corresponding error norms for a spectral element model as implemented in HOMME from Thomas and Loft (2002) in the bottom panel of Figure 3. Unlike the SE model, no filter or limiter is used for DG model runs. The DG solution exhibits smaller error growth as compared to SE scheme. However, the ℓ_∞ plot for the SE scheme is less oscillatory because of the use of a spatial filter. While the nodal and modal versions of the DG discretization give almost identical numerical results, the serial execution time for the nodal version of the DG scheme is nearly half that of the modal version.

4.2 Steady state geostrophic flow

Now we consider test case 2 in Williamson et al. (1992), which is a steady-state solution of the full nonlinear shallow water equations (SW2). The wind field is uniform as in the solid-body rotation case and the equations are geostrophically balanced during the time evolution. The initial velocity and height fields are,

$$\begin{aligned} u &= u_0 \left(\cos\alpha_0 \cos\theta + \sin\alpha_0 \cos\lambda \sin\theta\right), \\ v &= -u_0 \sin\alpha_0 \sin\lambda, \\ g\,h &= g\,h_0 - \frac{u_0}{2}(2\,a\,\omega + u_0) \times \\ &\quad (\sin\theta \cos\alpha_0 - \cos\lambda \cos\theta \sin\alpha_0)^2 \end{aligned}$$

where a is the earth's radius, $u_0 = 2\pi a/(12 \text{ days})$, and $gh_0 = 2.94 \times 10^4 \text{ m}^2/\text{ s}^2$.

For SW2 we selected the most challenging flow orientation parameter $\alpha_0 = \pi/4$. Figure 4 shows the height fields after 5 days of integration. The experiment was performed on a grid with $N_e = 5$, $N_v = 10$, and $\Delta t = 36$ seconds.

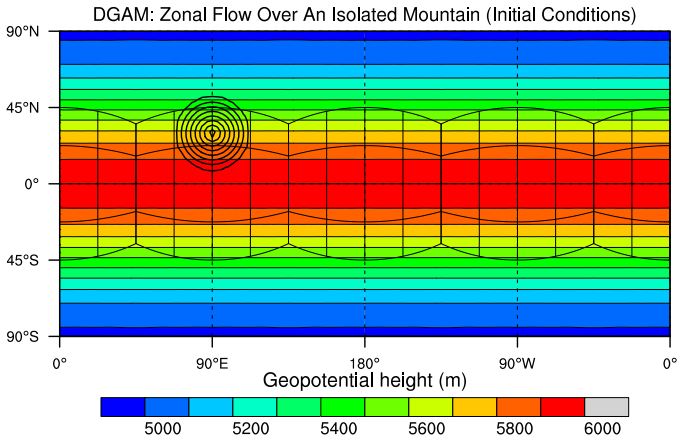

Figure 5 *Initial conditions for SW test case 5 (zonal flow over an isolated mountain). The closed contours indicate the location of the "mountain" with maximum height 2000 m.*

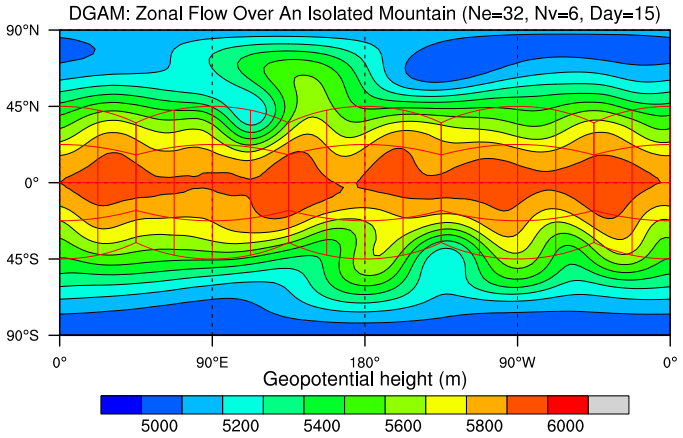

Figure 6 *Numerical solution for SW test case 5, zonal flow over an isolated mountain, after 15 days. A cubed-sphere with $N_e = 32$ and $N_v = 6$ was used for numerical integration.*

4.3 Zonal flow over an isolated mountain

The third experiment considered is test case 5 in Williamson et al. (1992), zonal flow over an isolated mountain (SW5). This test case, which has no analytical solution, is particularly useful for studying the effectiveness of the scheme in conserving integral invariants such as mass, total energy and potential enstrophy. The center of the mountain is located at $(3\pi/2, \pi/6)$ with height $h_s = 2000\,(1 - r/R)$ meters, where $R = \pi/9$ and $r^2 = \min[R^2, (\lambda - 3\pi/2)^2 + (\theta - \pi/6)^2]$. The wind velocity and height fields are the same as in the previous case with $\alpha_0 = 0$, $gh_0 = 5960$ m^2/s^2 and $u_0 = 20$ m/s. Figure 5 illustrates the initial conditions for this test.

SW5 typically causes difficulties for high-order methods such as spectral element and spectral transform methods (Chien et al. (1995)) because of the generation of spurious oscillations (spectral ringing) at all scales. We have used a variety of grid resolutions using different combinations of N_e and N_v to study the numerical behavior of DGAM for this test. We observe no spurious oscillations using the DG scheme even with a relatively low resolution grid of $N_e = 32$, $N_v = 4$ $(864 \times 4 \times 4)$ (Nair et al. (2005b)). The DG solutions appear similar in quality to the high resolution spectral T213 solutions shown in Chien et al. (1995), but without the spurious oscillations.

Figure 6 shows the numerical DGAM solutions after 15 days of integration on a high resolution grid with $N_e = 32$, $N_v = 6$, and $\Delta t = 30$ seconds. For DGAM mass is conserved to machine precision and other integral invariants, such as total energy and potential enstrophy, are better conserved than comparable resolution finite-volume models (Lin and Rood (1997)). These results are shown in Nair et al. (2005b).

4.4 Rossby-Haurwitz Wave

The fourth and final experiment is the test case 6 in Williamson et al. (1992), a zonal wavenumber 4 Rossby-Haurwitz wave (SW6). The initial state is an exact steadily propagating solution of the nondivergent barotropic vorticity equation which is not an exact solution of the full shallow water system.

The top panel of Figure 7 illustrates the initial conditions, and the bottom panel the numerical solution after 14 days of integration using an $N_e = 10$, $N_v = 4$ $(600 \times 4 \times 4)$ resolution grid. Unlike the NCAR spectral model of Chien et al. (1995), the DG scheme does not employ any diffusion terms. Nair et al. (2005b) demonstrated that for this experiment the change in total energy is almost one order of magnitude lower than the finite-volume SW model of Lin and Rood (1997) and the potential enstrophy error (from initial value) is of the same magnitude.

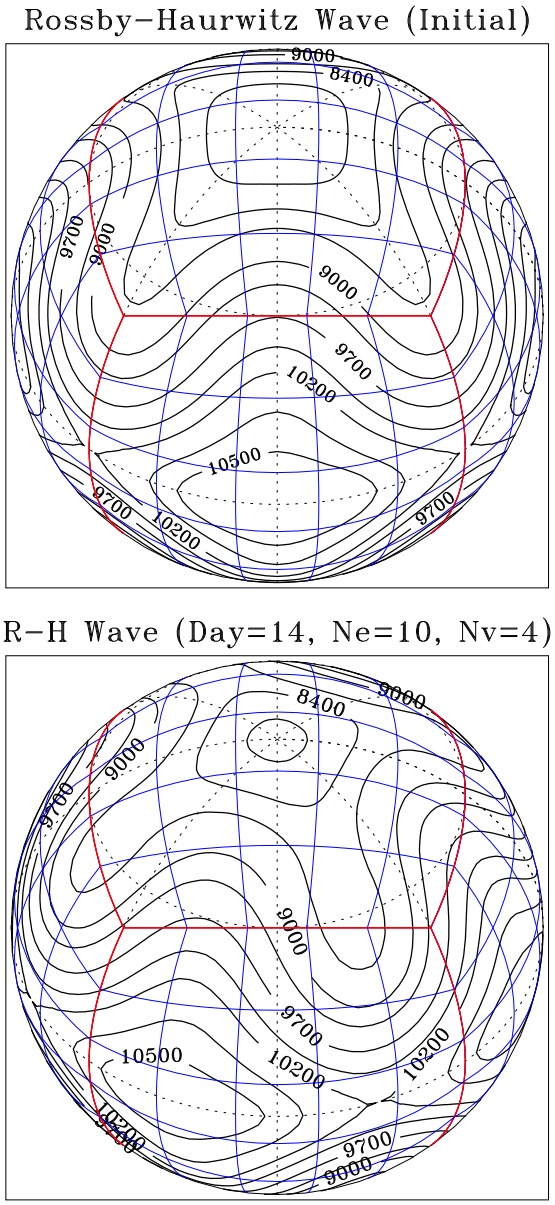

Figure 7 *Solution for SW test case 6 (Rossby-Haurwitz wave problem). Top panel shows the initial conditions and the bottom panel shows numerical solutions after 14 days of model integration, respectively. A cubed-sphere with $N_e = 10$ and $N_v = 4$ was used for numerical integration.*

5 Code Development and Performance

Development time was substantially reduced by building DGAM in the High Order Method Modeling Environment HOMME (2004) developed in the Computational Science Section (CSS) at NCAR. This environment was originally designed to support the spectral element method on the cubed-sphere and has been recently extended to provide the basic building blocks necessary for rapid development of parallel high-order atmospheric models. In adopting this environment we leverage previous and current work. For example, the ability to configure for shallow water and primitive equations (hydrostatic), support for various explicit and semi-implicit time-stepping schemes, efficient implementation of computational kernels, proven scaling to 1000's of processors (Loft et al. (2001)), interfaces to physics packages, and support for geometrically non-conforming elements and adaptive meshes.

5.1 Parallel Implementation

The parallel implementation of HOMME is based on a hybrid MPI/OpenMP approach. Partitioning of the computational grid is achieved using either METIS (Karypis and Kumar (2004)) or a locally developed space-filling curve algorithm (Dennis (2003)). Use of the space-filling curve approach eliminates load-imbalance caused by METIS partitioning (Loft et al. (2001)), and results in a 25-50% decrease in execution time on $\mathcal{O}(1000)$ processors. The space-filling curve approach generates the best partitions when $N_e = 2^n 3^m$, where n and m are integers. Because we provide scaling results for DG on the cubed-sphere grid where $N_e = 32$ we describe the space-filling curve partitioning technique based on a Hilbert space-filling curve (Sagan (1994)) in greater detail. The first step to partitioning the computational grid involves the mapping of elements from the 2D surface of the cubed-sphere into a linear array. Figure 8 illustrates the mapping of the 2D surface of the cubed-sphere into a linear array when $N_e = 2$. Figure 9 illustrates the Hilbert space-filling curve and elements, which are indicated by small circles, for one face of the cubed-sphere grid for $N_e = 16$.

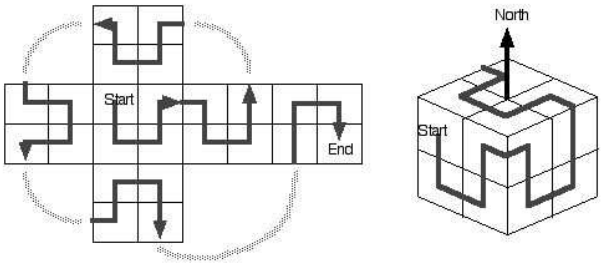

Figure 8 *A mapping of a Hilbert space-filling curve onto a flattened (left) and a perspective drawing (right) of the cubed-sphere*

The second steps involves partitioning the linear array into P contiguous groups, where P is the number of MPI tasks. The space-filling curve partitioning creates contiguous groups of elements and load-balances both the computational and communication costs.

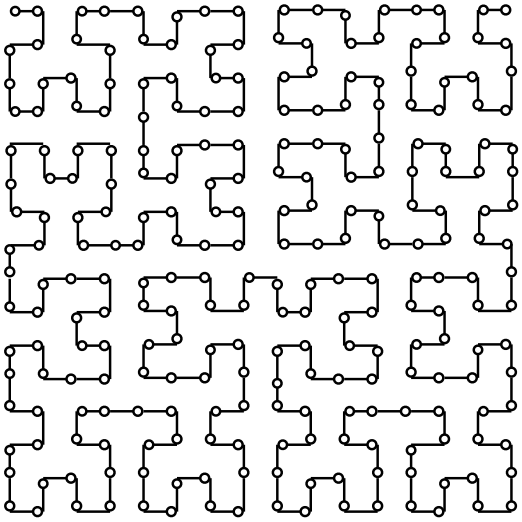

Figure 9 *A Hilbert curve with small circles, which correspond to the elements, for one face of the cubed-sphere for $N_e = 16$.*

The modifications required to build DGAM in HOMME were minor and did not impact parallel scalability. The existing HOMME communication framework consists of three different phases: packing, boundary exchange, and unpacking. We loosely synchronize the communication phase of the boundary exchange. We based DGAM on an existing scalable communication framework, though we do not support overlapping of flux calculations and communications, as in Baggag et al. (1999), because of time constraints. We first describe the existing HOMME communication framework followed by two additional routines added to support discontinuous Galerkin.

The pack routine edgeVpack() copies data from the element data space into a communication buffer. The routine bndry_exchange() exchanges data between neighboring MPI processes. The unpack routine edgeVunpack() performs a direct stiffness summation of the data from the communication buffer back to element space. Two additional routines, edgeDGVpack() and edgeDGVunpack(), were added to support DG within the existing framework. The packing routine edgeDGVpack() is identical to the existing edgeVpack() and is provided for naming consistency. The routine edgeDGVunpack() replaced the existing direct stiffness summation with a data copy into a padded element data structure. The padded data structure contains field values for the element as well as a halo of the surrounding elements. The padded data structure is then provided to the DG flux computation.

5.2 Blue Gene/L hardware

We next describe the Blue Gene/L (BG/L) hardware, a 1024 node system at NCAR, used for the scaling study. Each BG/L node has two 700 Mhz IBM PowerPC (PPC) 440 processors and 512 MB of memory. The PPC 440 processor is a low-power superscalar processor with 32 KB each of L1 data and instruction caches and a prefetch buffer referred to as the L2 cache. The two processors share a 4 MB L3 cache and an integrated memory controller. While the PowerPC 440 has two floating-point units (FPUs), the second FPU is only accessible through a set of special double-precision floating-point single instruction multiple data (SIMD) instructions. The SIMD instructions provide two multiply-add instructions per clock tick, for a theoretical CPU peak of 2.8 GFLOPS.

BG/L provides five networks, including a three-dimension torus for point-to-point communications and a broadcast tree for global reductions. A Message Passing Interface (MPI) BG/L job can be run in one of two modes. In coprocessor mode a single MPI process runs on each compute node: one of the processors of the compute node is used for computation and the other is used for communication. In virtual-node mode two MPI tasks are run on each compute node, one on each of the two processors. The advantage of coprocessor is that it affords each process access to the full 4 MB L3 cache and twice the main memory, whereas virtual-node mode affords computationally intensive codes access to the full peak speed of the system.

5.3 Performance on Blue Gene/L

To assess performance we examine the strong scaling for shallow water test case 2. We only present performance results for test case 2 because the floating-point costs for test cases 2, 5 and 6 are equivalent. We used a cubed-sphere with 6144 elements ($N_e = 32$). All tests were performed using 8×8 GLL points (i.e., 7^{th} degree Legendre polynomials).

To determine sustained MFLOPS per processor, the number of floating-point operations per time step was measured for the main time stepping loop using hardware performance counters on a IBM POWER4 system. Elapsed timers based on the rtc() function were employed in the main time stepping loop during performance tests to obtain the average execution time. An Allreduce() is invoked to determine the longest execution time per processor and is used to calculate the sustained execution rate. Model output is turned off during performance tests.

Figure 10 illustrates that the DGAM sustains between 221 to 306 MFLOPS (8-11%) of peak in virtual-node mode, and sustains between 266 to 340 MFLOPS (9-12%) of peak in coprocessor mode. Super-linear speedup is apparent for low-processor counts in coprocessor mode and is an indication that the $N_e = 32$ problem size is larger than the L3 cache for these processor counts. The super-linear speedup is even more pronounced using virtual-node mode, because the L3 cache is now split between the two processors.

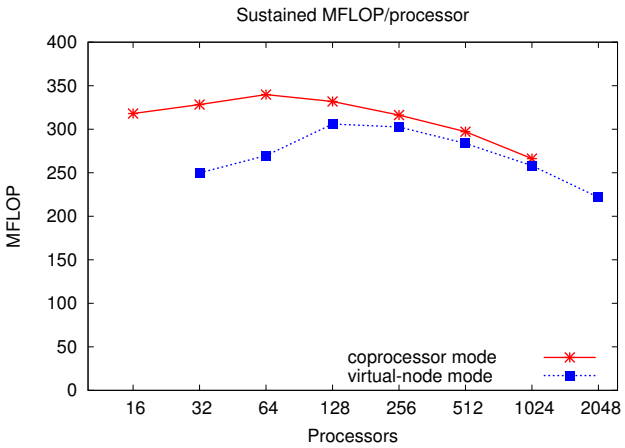

Figure 10 *Sustained MFLOPS per second per processor for cubed-sphere with $N_e = 32$ and $N_v = 8$ in coprocessor and virtual-node mode on BG/L.*

Though the strong scaling observed is quite good, sustained performance is lower than anticipated. A significant factor is due to the marginal single processor performance. While compilers for the PPC 440 are rapidly maturing they still have significant difficulty generating the SIMD instructions needed to engage the second FPU, when compiling with `-qarch=440d`. Consequently, we are only able to take advantage of one of the floating-point units (by compiling with `-qarch=440`.) We expect a significant increase in sustained performance with more mature compilers.

Another contributing factor to lower than anticipated sustained performance is that test case 2 is a 2D problem. The sister SE primitive equation implementation in HOMME (quasi-3D) achieves approximately 20% of peak on IBM POWER4 systems and experience with a 3D spectral element code indicates that 30% of peak is obtainable for slightly larger polynomial degrees (Tufo and Fisher (1992)).

6 Summary and Conclusions

A nodal high-order Discontinuous Galerkin atmospheric model (DGAM) has been developed. The model equation rely on the nonlinear flux form shallow water equations on the sphere. The computational domain is the cubed-sphere where the sphere is decomposed into six identical regions obtained by central (gnomonic) equiangular projections of the faces of the inscribed cube onto the spherical surface. The DG discretization employs a high-order nodal basis set consisting of Legendre polynomials; fluxes along the boundaries of the elements are approximated by a Lax-Friedrichs scheme. A third-order strong stability preserving Runge-Kutta scheme has been used for time integration, without any filter or limiter. DGAM has

been validated using the standard test suite proposed by Williamson et al. (1992).

The nodal version of the DG scheme is found to be nearly twice as fast as the the modal version (Nair et al. (2005b)). The nodal DG scheme exhibits exponential convergence for SW test case two (steady state geostrophic flow problem). The DG solutions to the SW test cases are much better than those of the spectral model of Chien et al. (1995) for a given spatial resolution. Even with high-order spatial discretization, the DG solutions do not exhibit spurious oscillations for the flow over a mountain test case. Conservation of integral invariants has also been compared with existing finite-volume models (e.g., Lin and Rood (1997)). Our model conserves mass to machine precision, and although the scheme does not formally conserve global invariants such as total energy and potential enstrophy, conservation of these quantities is better preserved than in lower order finite-volume models. The DGAM model implemented using the NCAR HOMME framework achieves between 8-11% of peak on processor counts of 16 - 2048 of the IBM Blue Gene supercomputer. Processor counts greater than 2048 requires careful attention to MPI process placement on BG/L to minimize communication cost. While the strong scaling is quite good for a 2D problem, the sustained performance is lower than anticipated; with more mature compilers, we expect somewhere between a 1.5 and 2x improvement. The development of a 3D dynamical core based on DG formulation presented here is in progress.

ACKNOWLEDGMENT

This work is supported by the DOE SciDAC program under award #DE-FG02-04ER63870. NCAR is sponsored by the National Science Foundation. This material is based upon work supported by the National Science Foundation under Grant No. 0325041. Computer time was provided by NSF MRI Grant #CNS-0421498, NSF MRI Grant #CNS-0420873, NSF MRI Grant #CNS-0420985, NSF sponsorship of the National Center for Atmospheric Research, the University of Colorado, and a grant from the IBM Shared University Research (SUR) program.

REFERENCES

Baggag, A. H. Atkins, C. Özturan, and D. Keyes (1999) 'Parallelization of an Object-Oriented Unstructured Aeroacoustics Solver', ICASE Report No. 99-11, Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX, March 22-24, 1999.

Cockburn, B., and Shu, C. W. (1989) 'TVB Runge-Kutta local projection discontinuous Galerkin method for conservation laws II: General framework', *Math. Comput.*, Vol. 52, pp.411–435.

Cockburn, B., and Shu, C. W. (2001) 'The Runge-Kutta discontinuous Galerkin method for convection-dominated problems' *J. Sci. Computing*, Vol. 16, pp.173–261.

Dennis, J. M. (2003) 'Partitioning with Space-Filling Curves on the Cubed-Sphere', IPDPS'03, Nice, France.

Deville, M. O., P. F. Fisher, and E. H. Mund (2002) 'High-order methods for incompressible fluid flow' *Cambridge monographs on applied and computational mathematics*, Cambridge University press, pp.499

Fournier, A., M. A. Taylor, and J. J. Tribbia (2004) 'A spectral element atmospheric model (SEAM): high-resolution parallel computation and localized resolution of regional dynamics' *Monthly Weather Review*, Vol. 132, pp.726–748.

Giraldo, F. X., Hesthaven, J. S. and Warburton, T. (2002) 'Nodal high-order discontinuous Galerkin methods for spherical shallow water equations' *Journal of Computational Physics*, Vol. 181, pp.499–525.

Giraldo, F. X., and T. E. Rosmond, 2004: 'A scalable spectral element Eulerian atmospheric model (SEE-AM) for NWP: dynamical core test.' *Mon. Wea. Rev.*, Vol. 132, pp.133–153.

Gottlieb, S., Shu, C. W. and Tadmor, E. (2001) 'Strong stability-preserving high-order time discretization methods' *SIAM Review*, Vol. 43, pp.89–112.

'High Order Method Modeling Environment (HOMME)' http://www.homme.ucar.edu/. (2005)

Jakob-Chien, R., Hack, J. J. and Williamson, D. L. (1995) 'Spectral transform solutions to the shallow water test set. *Journal Computational Physics*, Vol. 119, pp.164–187.

Karniadakis G. E., and Sherwin, S. J. (1999) *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.

Karypis, G., and Kumar, V. 2004: 'METIS Family of Multilevel Partitioning Algorithms' http://www-users.cs.unm.edu/ karypis/metis/.

Lin, S-J., and Rood, B. (1997) 'An explicit flux-form semi-Lagrangian shallow water model on the sphere' *Quarterly Journal of Royal Meteorological Society*, Vol. 123, pp.2531–2533.

Loft, R. D., Thomas, S. J. and Dennis, J. M. (2001) 'Terascale spectral element dynamical core for atmospheric general circulation models' Supercomputing 2001, ACM/IEEE conference, November 2001, Denver.

Nair, R. D., Thomas, S. J. and Loft, R. D. (2005) 'A discontinuous Galerkin transport scheme on the cubed-sphere' *Monthly Weather Review* Vol. 133, pp.814–828.

Nair, R. D., Thomas, S. J. and Loft, R. D. (2005) 'A discontinuous Galerkin global shallow water model' *Monthly Weather Review*, Vol. 133, pp.876–888.

Ronchi, C., Iacono, R. and Paolucci, P. S. (1996) 'The "Cubed Sphere" : A new method for the solution of partial differential equations in spherical geometry' *Journal Computational Physics*, Vol. 124, pp.93–114.

Sadourny, R. (1972) 'Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids' *Monthly Weather Review*, Vol. 100, pp.136–144.

Sagan, H. (1994) *Space-Filling Curves*, Springer-Verlag, 1994.

Thomas, S. J., and Loft, R.D. (2002) 'Semi-implicit spectral element model' *Journal Scientific Comput.*, Vol. 17, pp.339–350.

Tufo, H. M. and Fischer, P. F., (1999) 'Terascale spectral element algorithms and implementations' Supercomputing 1999, IEEE/ACM conference, November 1999, Portland.

Washington, W. M. and Parkinson, C. L. (2005) *An introduction to three-dimensional climate modeling*, University science books. 2^{nd} ed., 2005.

Williamson, D. L., Drake, J. B., Hack, J., Jakob, R. and Swartztrauber, P. N. (1992) 'A standard test set for numerical approximations to the shallow water equations in spherical geometry', *Journal of Computational Physics*, Vol. 102, pp.211–224.