

PARALLEL IMPLEMENTATION ISSUES: GLOBAL VERSUS LOCAL METHODS

Spectral elements are a potential alternative to the spectral transform method widely used in atmospheric general circulation models. A semi-implicit formulation permits larger time steps than an explicit formulation. Thus, a semi-implicit 3D spectral-element dynamical core should achieve high performance levels on microprocessor-based parallel clusters and substantially accelerate the climate simulation rate.

Climate modeling differs from weather forecasting primarily in terms of the spatial resolution and timescales involved. For example, weather models run at high resolution for short periods (from two to 10 days), whereas climate models run for decades or centuries of simulated time at much lower resolution to make the computations tractable. There are on the order of 100,000 to 1 million grid points in a climate model as opposed to 10 million or more in a global weather model. This implies that there is far less inherent parallelism available in climate models and hiding the latency inherent in vendor memory systems and communication networks is considerably more difficult.

The atmospheric 3D primitive equations are employed as the dynamical core of most existing climate models and are derived from the incompressible Navier-Stokes equations by scaling arguments applied to planetary flow on a rotating

sphere. Numerical methods for solving these partial differential equations can be broadly characterized as having either global or local data dependencies, both of which have specific drawbacks on highly parallel distributed-memory microprocessor-based architectures. Network bisection bandwidth usually limits highly efficient but less scalable global methods, whereas network latency tends to hinder less efficient but more scalable local methods. Despite the lack of exploitable parallelism at relatively low climate resolutions, global spectral models have exhibited high performance on modestly parallel vector architectures. This is due to the higher efficiency and greater per-processor performance of vector systems compared to microprocessors.

As a result, the spectral transform method is widely used in current climate models. It is a global method based on a representation of atmospheric flow variables by spherical harmonics. The spectral element method, in contrast, is a relatively new approach that combines the local finite-element method with high-order orthogonal polynomial expansions. Global communication might still be required for such local methods when the time-stepping scheme involves an implicit solver, such as the iterative conjugate gradient (CG) algorithm.

1521-9615/02/\$17.00 US Government Work Not Protected by US Copyright

STEPHEN J. THOMAS, RICHARD D. LOFT, AND JOHN M. DENNIS
National Center for Atmospheric Research

The associated global summations for computing inner products are sensitive to network latency. In this article, we examine a scalable 3D climate dynamical core based on a spectral element formulation of the primitive equations and compare the associated parallel communication requirements with the spectral transform method. Physical parameterizations and semi-Lagrangian advection schemes could introduce load imbalances, which further limit the parallel scalability of climate and weather models. Strategies for dealing with these load imbalances are not addressed here.

Spectral transform

Traditionally, researchers have implemented climate model dynamical cores based on the *spectral transform method*, where the dynamical flow variables are decomposed into spherical harmonics. These basis functions provide an isotropic representation on the sphere, and the associated time-stepping schemes are not adversely affected by clustering grid points near the poles. Solving the Helmholtz elliptic problem for the pressure or divergence, arising from a semi-implicit time discretization scheme, is trivial because these same spherical harmonics are eigenfunctions of the Laplacian on the sphere. These basis functions consist of a 2D tensor-product of 1D complex exponentials and the associated Legendre polynomials. The Helmholtz problem for the 3D primitive equations then decouples into independent tridiagonal linear systems in the vertical direction, which a direct solver can easily handle.

The spectral transform itself requires all latitude points (in the zonal direction) to compute Fourier transforms (via the fast Fourier transform algorithm) followed by the Legendre transform (matrix-matrix multiplies) in the meridional direction. The now standard approach for performing these computations on distributed-memory parallel computers is to use either 1D or 2D horizontal domain decomposition (along with all-to-all array transposition), which permits parallel computation of multiple FFTs—for example, on each processor for a block of latitudes. Typically, a transposition must ensue to set up a Legendre transform data layout. On distributed-memory machines, this transposition involves a global all-to-all communication step. Thus, as the number of processors P increases, the total message volume stays essentially constant, while the size of individual messages decreases as $1/P$. Bisection bandwidth limits the transposition for low P , and latency dominates for high values of P . Therefore,

the advantage that the semi-implicit Helmholtz problem is trivial to solve in spectral space is counterbalanced on many systems by the communication overhead of transpositions.

Spectral elements

Spectral elements maintain both the accuracy and the exponential convergence rate that the spectral transform method exhibits and have proven to be effective in geophysical fluid dynamics.^{1,2} They also permit local grid refinement and offer several computational advantages on microprocessors; for example, the computations are naturally cache-blocked, and derivatives are computed using nearest neighbor communication.

Combined with an explicit time-stepping scheme, parallel communication is strictly local, involving nearest neighbor exchanges between elements to enforce continuity. Finite-difference methods have analogous communication patterns involving many short messages that are sensitive to network latency. However, the explicit time step is limited by the Courant condition for gravity waves. For spectral elements, stability is determined by the eigenvalues of the discrete spatial operators. With quadratic clustering of Gauss points at element boundaries, these eigenvalues scale as $1/KN^2$, where K is the number of elements and N is the polynomial degree. For the spectral transform, the explicit time step decreases linearly with increasing resolution. Consequently, the explicit time step is more restrictive in the case of spectral elements. A semi-implicit scheme removes this stability constraint,³ introducing a semi-Lagrangian scheme for spectral elements can help avoid the stability constraint associated with the wind speed.

The longer semi-implicit time step must overcome the added computational cost of solving a linear system of equations for the pressure and result in an acceleration of the model integration rate (measured in simulated days per day). In a spectral transform dynamical core, the Helmholtz linear solver typically requires 5 percent of the serial computation time, making large acceleration factors possible. However, communication can dominate the dynamics computations on parallel machines due to the large network bisection bandwidth required. The Helmholtz problem in the spectral-element formulation lacks the global tensor-product structure that is key to the spectral transform.⁴ We have adopted the iterative CG algorithm, which is sensitive to network latency due to the global summations required for the CG algorithm's inner products. A suitable preconditioner can significantly

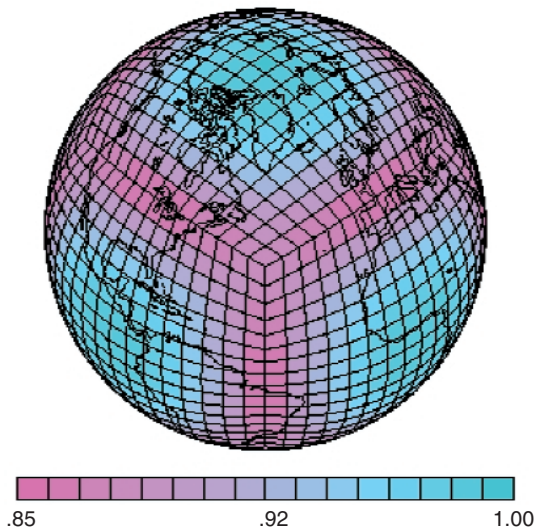


Figure 1. A cubed-sphere grid.

accelerate the CG solver convergence rate, thereby reducing the overall computational expense of the semi-implicit relative to the explicit time step. An overlapping Schwarz preconditioner leads to acceleration factors on the order of two in the case of the 2D shallow-water equations.^{5,6}

Atmospheric modelers often use the shallow-water equations—a prototype for the full 3D primitive equations—to evaluate promising numerical techniques. If successful, studies using these equations are followed up with more realistic 3D primitive equation tests with idealized physical forcings, such as the Held-Suarez model. This test also serves as a realistic parallel performance benchmark.⁷

The parallel implementation of our spectral element dynamical core is hybrid MPI–OpenMP, and the entire model time step is coded using shared-memory thread directives within a single-program multiple-data (SPMD) parallel region. Although it seems reasonable to expect that the hybrid code would be the most efficient on symmetric multiprocessor (SMP) cluster architectures, we found that the all-MPI variant consistently outperforms the hybrid code. Multiple MPI processes seem to do a better job of saturating the network interface cards in SMP clusters than the single MPI process that spawns threads in the hybrid paradigm. Additionally, for both all-MPI and hybrid codes, it is better to undersubscribe the SMP systems. This observation suggests that a small number of processors must be reserved for operating system activities occurring behind the scenes.

Cache blocking, combined with looping over model layers between synchronizations for MPI calls, results in per-processor execution rates exceeding 25 percent of peak for explicit 3D simulations. For low

processor counts (with many degrees of freedom per processor), the execution rate is about 25 percent of the peak or theoretical maximum. For large processor counts, the flop rate drops off to around 16 percent of peak. This is still quite efficient for microprocessor-based systems—most climate codes achieve around 4 percent or less of peak at large processor counts. For example, our explicit Held-Suarez run sustains 370 Gflops on the 128 16-way nodes of the National Energy Research Supercomputer Center (NERSC) Power 3-based IBM SP. In contrast, NCAR’s CCM3 climate code benchmark runs at roughly 3.6 Gflops on 64 Power-3 processors.

Spectral-element formulation of the primitive equations

The 3D primitive equations neglect vertical acceleration and are derived from the incompressible Navier-Stokes equations by invoking scaling arguments for the atmospheric general circulation model. We follow the NCAR CCM3 formulation by introducing a general, terrain-following, vertical coordinate η .⁸ This hybrid coordinate $\eta(p, p_s)$ is a monotonic function of pressure p and depends on the surface pressure p_s , where boundary conditions are given by $\eta(0, p_s) = 0$ and $\eta(p_s, p_s) = 1$.

The hydrostatic relation is used to diagnose the geopotential height field $\phi = gb$; we obtain the pressure p from the surface pressure p_s using the definition of the hybrid vertical coordinate. The vertical discretization uses finite differences and is designed to conserve both energy and angular momentum in the absence of external forcing terms. NLEV vertical layers are defined by the “half-level” pressures $p_{l+1/2} = A_{l+1/2} + B_{l+1/2} p_s$. The constants $A_{l+1/2}$ and $B_{l+1/2}$ effectively define the vertical coordinate, where $B = \partial p / \partial p_s$. Prognostic variables are represented by “full-level” values at pressure levels l . A hybrid terrain-following coordinate is a combination of sigma $\sigma = p/p_s$ and pure pressure levels. Using η vertical coordinates can significantly reduce pressure gradient truncation errors over steep topography.⁷

In the horizontal discretization via spectral elements, the computational domain Ω is partitioned into K elements Ω_k in which the dependent and independent variables are approximated by N th-order tensor-product polynomial expansions. For the P_N/P_{N-2} formulation discussed elsewhere,¹⁰ the velocity is expanded in terms of the N th-degree Lagrangian interpolants b_j ,

$$\mathbf{u}_k^k(r_1, r_2) = \sum_{i=0}^N \sum_{j=0}^N \mathbf{u}_{ij} b_i(r_1) b_j(r_2) \quad , \quad (1)$$

and the geopotential is expanded using the $(N-2)$ th degree interpolants \tilde{b}_i

$$\phi_b^k(r_1, r_2) = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \phi_{ij} \tilde{b}_i(r_1) \tilde{b}_j(r_2), \quad (2)$$

where r_1 and r_2 are coordinates in the box $[-1, 1] \times [-1, 1]$. We obtain a weak variational form of the problem by integrating the equations with respect to test functions and directly evaluating inner products using Gaussian quadrature. Two integration rules are defined for a staggered mesh by taking the tensor product of Gauss and Gauss-Lobatto quadrature rules on each element:

$$\begin{aligned} (f, g)_{GL} &= \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N f^k(\xi_i, \xi_j) g^k(\xi_i, \xi_j) \rho_i \rho_j \\ (f, g)_G &= \sum_{k=1}^K \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} f^k(\zeta_i, \zeta_j) g^k(\zeta_i, \zeta_j) \sigma_i \sigma_j, \end{aligned} \quad (3)$$

where (ξ_i, ρ_i) , $i = 0, \dots, N$ are the Gauss-Lobatto nodes and weights, and (ζ_i, σ_i) , $i = 1, \dots, N-1$ are the Gauss nodes and weights on $\Lambda = [-1, 1]$. We map physical coordinates according to $\mathbf{x} \in \Omega_k \Rightarrow \mathbf{r} \in \Lambda \times \Lambda$. C^0 velocity continuity is specified at inter-element boundaries that share Gauss-Lobatto points, and direct stiffness summation is applied, as in the standard finite-element method, to assemble global matrices.

For the primitive equations, we use a cubed-sphere coordinate system whereby the sphere is tiled with quadrilateral elements by subdividing the six faces of the cube and then using a gnomonic projection to map these elements onto the sphere (see Figure 1).^{1,11} We can specify either a staggered or nonstaggered horizontal discretization of the primitive equations. In the staggered formulation, scalar variables T and $\ln p_s$ reside on the Gauss pressure grid, whereas \mathbf{v} is located at Gauss-Lobatto velocity points. In the nonstaggered case, all variables are located on the Gauss-Lobatto grid, and the expansions use polynomials of degree N .

The 2D shallow-water equations admit the same horizontal wave motion as the 3D primitive equations—namely, fast-moving gravity waves and planetary scale Rossby waves. A semi-implicit scheme for a spectral-element discretization of the shallow-water equations using a CG solver appears elsewhere.⁶ A suitable preconditioner can significantly accelerate the convergence rate of the CG iterative linear solver, thereby reducing the over-

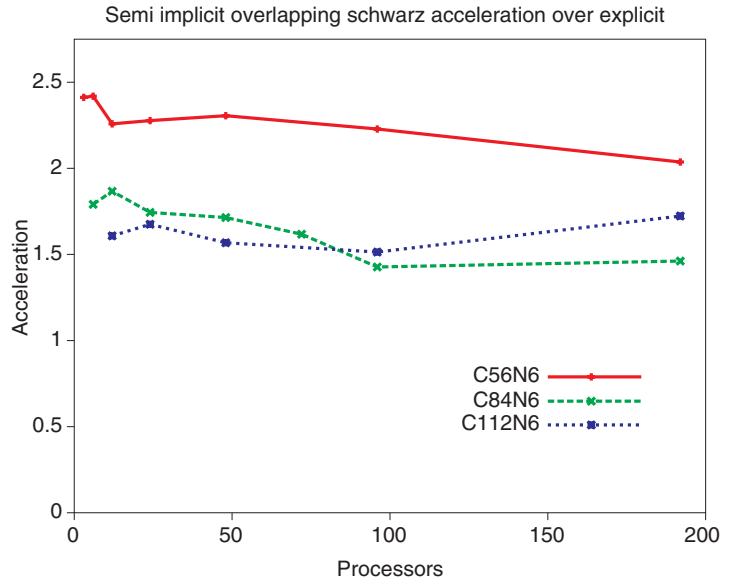


Figure 2. The semi-implicit integration rate acceleration factor over explicit with an overlapping Schwarz preconditioner ($np = 6$ Gauss pressure points).

all computational expense of the semi-implicit relative to the explicit time step. A block-Jacobi preconditioner performs well at low climate resolutions, but leads to a growth in the iteration count at higher resolutions as both the number of elements K and the polynomial order N increase. The 2D block-Jacobi preconditioner has a computational cost of $O(KN^4)$. We have recently investigated whether a generalization of the additive overlapping Schwarz approach leads to a more robust preconditioner.^{5,11} The subdomain problems in the Schwarz preconditioner are based on linear finite-element approximations of the local (separable) elliptic operators, and the computational cost drops to $O(KN^3)$.

Figure 2 plots the integration rate acceleration factors for three different resolutions using $np = 6$ Gauss pressure points with $ne = 8, 12,$ and 16 elements along a cube-face edge (denoted C56N6, C84N6, and C112N6).¹² These results are promising, and we are currently extending the CG solver to the primitive equations by eigenmode decomposition of the 3D Helmholtz problem into NLEV decoupled 2D Helmholtz problems for the generalized pressure variable $P = \phi + RT_0 \ln p_s$. The semi-implicit scheme removes the stability constraint imposed by gravity waves, but the Courant number associated with the advecting wind is still limited by the quadratic clustering of grid points near element boundaries. A semi-Lagrangian advection scheme for spectral elements would remove this time-step restriction and allow for even larger acceleration factors over explicit Eulerian formulations.¹³

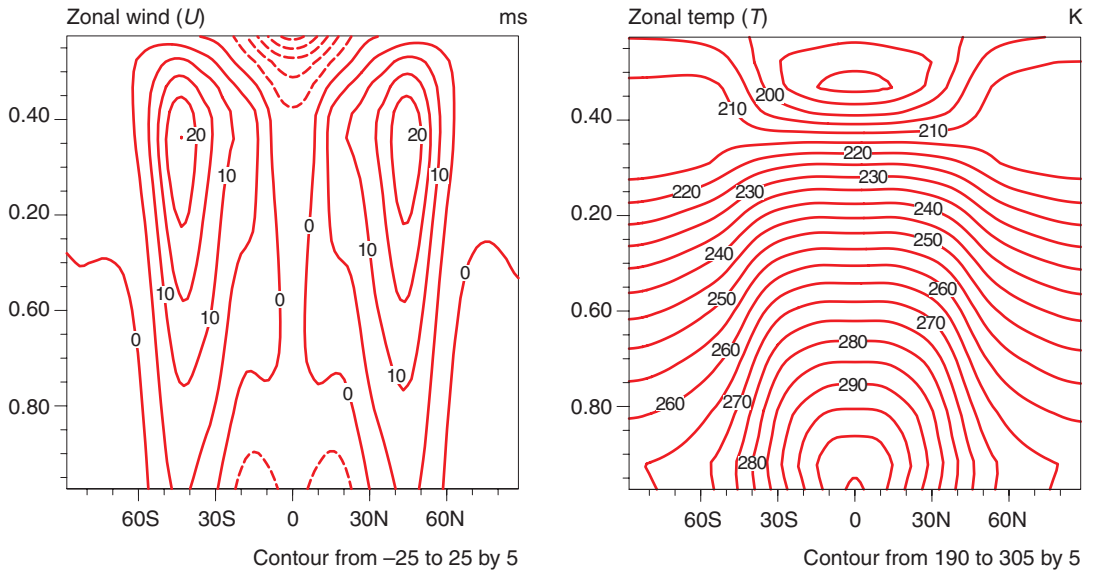


Figure 3. A Held-Suarez idealized climate with 1,200-day integration, 150 elements, 19 vertical sigma levels, and zonally averaged (a) wind [U] and (b) temperature [T].

Held-Suarez results

The Held-Suarez idealized climate forcings are designed to test the dry dynamical core of a general circulation model for multiyear integrations. It assumes an ideal gas atmosphere over a rotating sphere with no topography. The flow is not specified as hydrostatic, but we can use hydrostatic primitive equations. The prescribed forcing consists of a simple Newtonian relaxation of the temperature field to a zonally symmetric state and Rayleigh damping of the lower level wind field to approximate friction or drag caused by the atmospheric boundary layer near the surface. The atmosphere's initial state is hydrostatic and isothermal $T = 300$ K. The model is integrated for 1,200 days, and zonally averaged wind and temperature fields are then reported. The Held-Suarez forcings take the form

$$\frac{\partial \mathbf{v}}{\partial t} = \dots - k_v(\phi, \sigma) \mathbf{v} \quad (4)$$

$$\frac{\partial T}{\partial t} = \dots - k_T(\phi, \sigma) [T - T_{eq}(\phi, \sigma)], \quad (5)$$

where ϕ is the latitude, and $\sigma = p/p_s$ is the vertical sigma coordinate level. The temperature is relaxed to the equilibrium temperature T_{eq} and relaxation rate k_T ; the wind's linear damping rate is given by k_v . We performed a simulation using 150 spectral elements and 19 equally spaced sigma levels in the vertical direction (see Figure 3). These plots exhibit the characteristic formation of jets in the upper atmosphere.⁷

We tested the nonstaggered explicit primitive equations on the NERSC IBM SP with a Colony switch and 16 375-MHz Power-3 processors per

node. On this machine, each node contains two network interface cards in a double-rail configuration, effectively doubling the point-to-point bandwidth. The horizontal mesh consisted of $K = 2,904$ spectral elements, corresponding to $ne = 22$ elements on a cube-face edge and $nw = 8$ Gauss-Lobatto points (referred to as C154N6 resolution). This resolution is comparable to a spectral transform model with a T170 triangular truncation.

We also used a pure sigma pressure coordinate with 16 vertical levels. The explicit model time step was $\Delta t = 36$ s, whereas a semi-implicit time step would be on the order of $\Delta t = 300$ s. To assess the impact of asynchronous MPI communication on individual SMP nodes, we ran the model with 15 and 12 MPI processes per node. We partitioned the cubed-sphere spectral-element mesh into roughly equal-size subdomains using the METIS package.¹⁴ For 15 MPI processes per node, the optimal METIS mesh-partitioning algorithm differed for small, medium, and large node counts. From two (30 processors) to 12 (180 processors) nodes, the impact of load imbalance is not as significant as the number of neighbors and total communication volume sent. Hence, the recursive bisection (RB) algorithm performs best on small node counts because it minimizes both total edge cut as well as communication volume. When the number of nodes exceeds 12, the K -way algorithm offers the best performance, because it provides the best load balance of all the METIS algorithms. At higher node counts, the algorithm that provides acceptable load balance and minimizes total communication volume performs best. This is typically the total volume algorithm, although in the case of

121 nodes (1,815 processors), RB provides a sufficiently balanced partition and minimal communication volume. We obtained the best 12 MPI process results by using only the K -way algorithm; in this case, load imbalance has less of an impact on performance. We achieved our best result of 370 Gflops on 128 nodes using 12 MPI processes per node for a total of 1,536 processors.

Although sustained execution rates approaching a teraflop are desirable,¹⁵ the most important metric for climate modeling is the simulation throughput or integration rate measured in simulated years per wall-clock day of CPU time. Our initial tests of 2D shallow-water and 3D primitive equations models, based on spectral elements, suggest that century-per-day integration rates are now possible. These results are encouraging for parallel climate simulations on microprocessor-based SMP clusters. The scalability of the semi-implicit and explicit models seems limited only by the underlying system's interconnect. Future progress in climate simulation on highly parallel computers will be tied to the development of low-latency, high-bandwidth communication networks. In the short term, we hope to demonstrate that a semi-implicit 3D spectral-element dynamical core will scale to the same performance levels and substantially accelerate the simulation rate. ■

References

1. M. Taylor, J. Tribbia, and M. Iskandarani, "The Spectral Element Method for the Shallow Water Equations on the Sphere," *J. Computational Physics*, vol. 130, 1997, pp. 92–108.
2. M. Iskandarani, D.B. Haidvogel, and J.P. Boyd, "A Staggered Spectral Element Model with Application to the Oceanic Shallow Water Equations," *Int'l J. Numerical Methods Fluids*, vol. 20, 1995, pp. 394–414.
3. S.J. Thomas and R.D. Loft, "Semi-Implicit Spectral Element Atmospheric Model," *J. Scientific Computing*, vol. 17, nos. 1-4, Dec. 2002, pp. 339–350.
4. S.A. Orszag, "Spectral Methods for Problems in Complex Geometry," *J. Computational Physics*, vol. 37, 1980, pp. 70–92.
5. P.F. Fischer, N.I. Miller, and H.M. Tufo, "An Overlapping Schwarz Method for Spectral Element Simulation of Three-Dimensional Incompressible Flows," *Parallel Solution of Partial Differential Equations*, P. Bjorstad and M. Luskin, eds., Springer-Verlag, New York, 2000, pp. 159–180.
6. S.J. Thomas et al., "An Overlapping Schwarz Preconditioner for the Cubed-Sphere," submitted to *SIAM J. Scientific Computing*, 2002.
7. I.H. Held and M.J. Suarez, "A Proposal for the Intercomparison of the Dynamical Cores of Atmospheric General Circulation Models," *Bulletin Am. Meteorological Soc.*, vol. 75, no. 10, Oct. 1994, pp. 1825–1830.
8. J.T. Kiehl et al., *Description of the NCAR Community Climate Model*, tech. report NCAR/TN-420+STR, Nat'l Ctr. for Atmospheric

Research, Boulder, Colo., 1996.

9. A.J. Simmons and D.M. Burridge, "An Energy and Angular-Momentum Conserving Finite-Difference Scheme and Hybrid Vertical Coordinates," *Monthly Weather Rev.*, vol. 109, 1981, pp. 758–766.
10. E.M. Ronquist, *Optimal Spectral Element Methods for the Unsteady Three-Dimensional Navier Stokes Equations*, PhD thesis, Dept. of Mechanical Eng., Massachusetts Inst. of Technology, Cambridge, Mass., 1988, p. 176.
11. M. Rancic, R.J. Purser, and F. Mesinger, "A Global Shallow-Water Model Using an Expanded Spherical Cube: Gnomonic versus Conformal Coordinates," *Quarterly J. Royal Meteorological Soc.*, vol. 122, 1996, pp. 959–982.
12. D.L. Williamson et al., "A Standard Test Set for Numerical Approximations to the Shallow Water Equations in Spherical Geometry," *J. Computational Physics*, vol. 102, 1992, pp. 211–224.
13. F.X. Giraldo and J.B. Perot, "A Spectral Element Semi-Lagrangian Method for the Spherical Shallow Water Equations," submitted to *J. Computational Physics*, 2002.
14. G. Karypis and V. Kumar, "Multilevel K-Way Partitioning Scheme for Irregular Graphs," *J. Parallel and Distributed Computing*, vol. 48, 1998, pp. 96–129.
15. R.D. Loft, S.J. Thomas, and J.M. Dennis, "Terascale Spectral Element Dynamical Core for Atmospheric General Circulation Models," *Proc. ACM/IEEE Supercomputing01*, IEEE CS Press, Los Alamitos, Calif., 2001.

Stephen J. Thomas is a scientist in the Computational Science Section, Scientific Computing Division, at the National Center for Atmospheric Research. His technical interests include high-order methods, Krylov iterative solvers and preconditioners, and geophysical fluids. He has a B.Math in applied mathematics from the University of Waterloo, a M.Eng in electrical engineering from McGill University, and a PhD in computer science from the University of Montreal. Contact him at NCAR, 1850 Table Mesa Dr., Boulder, CO 80305; thomas@ucar.edu.

Richard D. Loft is head of the Computational Science Section, Scientific Computing Division, at the National Center for Atmospheric Research. His technical interests include optimization for microprocessor architectures applied to the spectral transform and high-order methods for geophysical fluids. He has a BS in chemistry from Harvey Mudd College and an MS and PhD, both in physics, from the University of Colorado. Contact him at NCAR, 1850 Table Mesa Dr., Boulder, CO 80305; loft@ucar.edu.

John M. Dennis is a software engineer in the Computational Science Section, Scientific Computing Division, at the National Center for Atmospheric Research. His technical interests include numerical linear algebra, mesh partitioning, and parallel communication algorithms. He has a BS and MS in computer science from the University of Colorado. Contact him at NCAR, 1850 Table Mesa Dr., Boulder, CO 80305; dennis@ucar.edu.

For more information on this or any other computing topic, please visit our digital library at <http://computer.org/publications/dlib>.