TERASCALE SPECTRAL ELEMENT DYNAMICAL CORE FOR ATMOSPHERIC GENERAL CIRCULATION MODELS [1]

Richard D. Loft, Stephen J. Thomas, John M. Dennis

National Center for Atmospheric Research
1850 Table Mesa Drive, Boulder CO 80303, USA

Abstract

Climate modeling is a grand challenge problem where scientific progress is measured not in terms of the largest problem that can be solved but by the highest achievable integration rate. These models have been notably absent in previous Gordon Bell competitions due to their inability to scale to large processor counts. A scalable and efficient spectral element atmospheric model is presented. A new semi-implicit time stepping scheme accelerates the integration rate relative to an explicit model by a factor of two, achieving 130 years per day at T63L30 equivalent resolution. Execution rates are reported for the standard shallow water and Held-Suarez climate benchmarks on IBM SP clusters. The explicit T170 equivalent multi-layer shallow water model sustains 343 Gflops at NERSC, 206 Gflops at NPACI (SDSC) and 127 Gflops at NCAR. An explicit Held-Suarez integration sustains 369 Gflops on 128 16-way IBM nodes at NERSC.

1. Introduction

Climate simulation is a grand challenge problem requiring multiple, century long integrations of the equations governing the earth's atmosphere. Consequently, grid resolutions in atmospheric climate models are coarser than in numerical weather models, where accurate predictions are limited to about ten days. Current climate models are typically run with a 300 km equatorial grid spacing (T42 spectral truncation), whereas global weather model resolutions are approaching 10 km. To date, achieving high integration rates for climate models on highly parallel clusters of microprocessors has been problematic. Operating at low resolutions effectively limits the scalability of climate models on MPP's. This limitation is further compounded by the widespread choice of dynamical

SC2001 November 2001, Denver (c) 2001 ACM 1-58113-293-X/01/0011 $5.00

algorithms based on the spectral transform method. The spectral transform method is mathematically attractive because the global spherical harmonic basis functions provide an isotropic representation on the sphere. In addition, it is trivial to implement semi-implicit time stepping schemes, because the spherical harmonics are eigenfunctions of the Laplacian on the sphere and the resulting Helmholtz problem is embarrassingly parallel in spectral space. Although spectral models have exhibited good performance on modestly parallel vector architectures, they require non-local operations, such as array transpositions, which are expensive on parallel RISC systems. In this paper we examine the performance of multi-layer shallow water and 3-D primitive equation models based on scalable spectral element numerics. The shallow water test suite of Williamson et al (1992) and the Held-Suarez (1994) idealized climate tests are used to evaluate these for climate modeling. Issues such as the load imbalance caused by a full physics package or the introduction of a semi-Lagrangian advection scheme have not been addressed. Nevertheless, demonstrating the scalability of a dynamical core for an atmospheric general circulation model represents a major advance.

Spectral elements maintain the accuracy and exponential convergence rate exhibited by the spectral transform method and have proven to be effective in computational fluid dynamics applications, Karniadakis and Sherwin (1999). Spectral elements have also been applied in geophysical fluid dynamics by Taylor et al (1997) and Iskandarani et al (1995). We note that the 1999 Gorden Bell prize in the special category was awarded to Tufo and Fischer (1999) for a spectral element incompressible Navier-Stokes model. Spectral elements also offer several computational advantages on parallel systems composed of RISC microprocessors. The computations are naturally cache-blocked and derivatives may be computed using nearest neighbor communication. An explicit version of a spectral element atmospheric model (SEAM) has demonstrated linear scaling on a variety of parallel machines, Taylor et al (1997). Unfortunately, the explicit model suffers from severe time step restrictions. An efficient semi-implicit formulation of this spectral element model was developed recently by Thomas and Loft (2001). Numerical innovations discussed in this paper include a weak formulation of the governing equations and incorporation of a block-Jacobi preconditioned conjugate gradient solver that is latency tolerant. Our parallel implementation of SEAM is a true hybrid MPI/OpenMP code and the entire model time-step is threaded over elements using an SPMD parallel region. The domain decomposition strategy is fully general and based on the METIS graph partitioning software. Performance of the shallow water version of SEAM exceeds 127 Gflops on the NCAR IBM SP, 206 GFlops on the NPACI IBM SP and 343 Gflops on the NERSC IBM SP. The primitive equations version sustains 369 Gflops at NERSC.

The model code is performance portable and runs at 93 Gflops on 256 processors of the ORNL Compaq Alphaserver.

2. Spectral Element Formulation

The shallow water equations have been used for many years by the atmospheric modeling community to test promising numerical methods. These equations contain the same horizontal wave propagation mechanisms found in more complete models. In particular, they admit the Rossby and gravity wave solutions found in 3-D primitive equations models. The governing equations for the inviscid flow of a thin layer of fluid in 2-D are the horizontal momentum and continuity equations for the velocity $\mathbf{v}$ and geopotential height ϕ. In curvilinear coordinates, the shallow water equations can be written as follows (see Sadourny 1972)

$$\begin{aligned}\frac{\partial u^i}{\partial t} &= -g^{ij}\left[\epsilon_{jk}\, u^k\, g\,(\,f+\zeta\,)+\frac{\partial}{\partial x^j}\left(\frac{1}{2}u_k\, u^k\right)+\frac{\partial \phi}{\partial x^j}\right]\\ \frac{\partial \phi'}{\partial t} &= -u^j\,\frac{\partial \phi}{\partial x^j}-\frac{\phi}{g}\,\frac{\partial}{\partial x^j}\,(\,g\,u^j\,)\end{aligned}$$

where $\phi=\phi'+\phi_0$, f is the Coriolis force and ζ is the relative vorticity. ϵ_{jk} is a permutation matrix. Covariant and contravariant vectors are related through the metric tensor g_{ij}, $u^i = g^{ij}\, u_j$, $g_{ij}^{-1} = g^{ij}$ and $g = \{\,\det(g_{ij})\,\}^{1/2}$. Divergence and vorticity are given by

$$g\,\nabla\cdot\mathbf{v} = \frac{\partial}{\partial x^j}\,(\,g\,u^j\,),\quad g\,\zeta = \epsilon_{ij}\,\frac{\partial u_j}{\partial x^i},$$

The sphere is tiled with rectangular elements by subdividing the six faces of the cube, which circumscribes the sphere, and then using a gnomonic projection to map these elements onto the surface of the sphere (see Figure 1). Rancic et al (1996) showed that an equal angular projection results in a more uniformly spaced grid. For equal angular coordinates $(\,x_1,\,x_2\,)$, $-\pi/4 \le x_1, x_2 \le \pi/4$, and the metric tensor for all six faces of the cube is

$$g_{ij} = \frac{1}{r^4\cos^2 x_1\cos^2 x_2}\begin{bmatrix} 1+\tan^2 x_1 & -\tan x_1\tan x_2\\ -\tan x_1\tan x_2 & 1+\tan^2 x_2\end{bmatrix},$$

$r = \{1+\tan^2 x_1+\tan^2 x_2\}^{1/2}$, and $g = 1/r^3\cos^2 x_1\cos^2 x_2$. A vector $\mathbf{v} = (\,v_1,\,v_2\,)$ in spherical coordinates is entirely defined by its covariant and contravariant components u_i and u^i on the cube. For the vector $(\,u_1,\,u_2\,)$ on the cube, define the mapping

$$A = \begin{bmatrix}\cos\theta\,\partial\lambda/\partial x_1 & \cos\theta\,\partial\lambda/\partial x_2\\ \partial\theta/\partial x_1 & \partial\theta/\partial x_2\end{bmatrix},\quad A^T\begin{bmatrix}v_1\\ v_2\end{bmatrix}=\begin{bmatrix}u_1\\ u_2\end{bmatrix},\quad A\begin{bmatrix}u^1\\ u^2\end{bmatrix}=\begin{bmatrix}v_1\\ v_2\end{bmatrix}.$$

where $A^T A = g_{ij}$. Requiring that the velocities in spherical coordinates match along the cube edge shared by face i and face j, the relationship is $A_i \, \mathbf{u}_i = A_j \, \mathbf{u}_j$, $\mathbf{u}_i = A_i^{-1} A_j \, \mathbf{u}_j$. Our semi-implicit scheme applied to the shallow water equations combines an explicit leapfrog scheme for the advection terms with a Crank-Nicholson scheme for the gradient and divergence terms. For the shallow water equations the scheme can be written in terms of the differences $\delta \, u^i = u^{i(n+1)} - u^{i(n-1)}$ and $\delta \, \phi = \phi^{n+1} - \phi^{n-1}$

$$\delta \, u^i + \Delta t \, g^{ij} \, \frac{\partial}{\partial x^j} \, (\, \delta \, \phi \,) \;\; = \;\; 2\Delta t \left[-g^{ij} \, \frac{\partial}{\partial x^j} \, (\, \phi \,)^{n-1} + f_u^{i(n)} \right] \qquad (2.1)$$

$$\delta \, \phi + \Delta t \, \frac{\phi_0}{g} \, \frac{\partial}{\partial x^j} \, (\, g \, \delta u^j \,) \;\; = \;\; 2\Delta t \left[-\frac{\phi_0}{g} \, \frac{\partial}{\partial x^j} \, (\, g \, u^j \,)^{n-1} + f_\phi^n \right] \qquad (2.2)$$

where the tendencies f_u and f_ϕ contain nonlinear terms.

In the spectral element discretization, 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. The velocity is expanded in terms of the N–th degree Lagrangian interpolants h_i defined in Ronquist (1988),

$$\mathbf{u}_h^k(r_1, r_2) = \sum_{i=0}^{N} \sum_{j=0}^{N} \mathbf{u}_{ij} \, h_i(r_1) \, h_j(r_2)$$

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

$$\phi_h^k(r_1, r_2) = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \phi_{ij} \, \tilde{h}_i(r_1) \, \tilde{h}_j(r_2)$$

A weak variational problem is obtained 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.

$$(\, f, \, g \,)_{GL} \;\; = \;\; \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} f^k(\xi_i, \xi_j) \, g^k(\xi_i, \xi_j) \, \rho_i \, \rho_j$$

$$(\, f, \, g \,)_{G} \;\; = \;\; \sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} f^k(\zeta_i, \zeta_j) \, g^k(\zeta_i, \zeta_j) \, \sigma_i \, \sigma_j$$

where $(\, \xi_i, \, \rho_i \,)$, $i = 0, \ldots N$ are the Gauss-Lobatto nodes and weights and $(\, \zeta_i, \, \sigma_i \,)$, $i = 1, \ldots N-1$ are the Gauss nodes and weights on $\Lambda = [\, -1, \, 1 \,]$. Physical coordinates are mapped according to $\mathbf{x} \in \Omega_k \Rightarrow \mathbf{r} \in \Lambda \times \Lambda$. C^0 continuity of the velocity is enforced at inter-element boundaries which share Gauss-Lobatto points and direct stiffness summation is applied to assemble the global matrices. Derivative matrices $\tilde{\mathbf{D}} = (\tilde{D}_1, \tilde{D}_2)$ are rectangular. $\mathbf{B} = (B_1, B_2)$ and $\tilde{B}$ are diagonal velocity and geopotential mass matrices, with $\tilde{D}_j = \tilde{B}\, D_j$. The assembled discrete shallow water equations are then

$$B_i\, \delta\, u^i - g^{ij}\, \Delta t\, \tilde{D}_j^T\, \delta\, \phi \;=\; R_u^i \tag{2.3}$$

$$\tilde{B}\, \delta\, \phi + \Delta t\, \frac{\phi_0}{g}\, \tilde{D}_i\, g\, \delta\, u^i \;=\; R_\phi \tag{2.4}$$

The pressure is defined on the interior of an element and is 'communicated' between elements through the divergence in the continuity equation. An averaging procedure is required at element boundaries to enforce continuity, where velocity mass matrix elements in equation (2.3) are summed. This averaging procedure is related to the fact that basis functions interpolating boundary nodes are not local to a specific element.

A Helmholtz problem for the geopotential perturbation is obtained by solving for the velocity difference $\delta\, u^i$

$$\delta\, u^i = B_i^{-1}\, \left(\, R_u^i + \Delta t\, g^{ij}\, \tilde{D}_j^T\, \delta\, \phi\, \right) \tag{2.5}$$

and then applying back-substitution to obtain

$$g\, \tilde{B}\, \delta\, \phi + \Delta t^2\, \phi_0\, \tilde{D}_i\, g\, B_i^{-1}\, g^{ij}\, \tilde{D}_j^T\, \delta\, \phi = R'_\phi \tag{2.6}$$

where

$$R'_\phi \equiv g\, R_\phi - \Delta t\, \phi_0 \tilde{D}_i\, g\, B_i^{-1}\, R_u^i$$

Once the change in the geopotential $\delta\phi$ is computed, the velocity difference δu^i is obtained from (2.5). Given that the metric tensor g_{ij} is symmetric, g, $\tilde{B}$ and B_i are diagonal, it can be easily shown that the Helmholtz operator

$$H = g\, \tilde{B} + \Delta t^2\, \phi_0\, \tilde{D}_i\, g\, B_i^{-1}\, g^{ij}\, \tilde{D}_j^T$$

is symmetric positive definite. Thus preconditioned conjugate gradient iterative solvers can be applied. An effective preconditioner can be constructed by using local element direct solvers for the Helmholtz problem with zero Neumann pressure gradient boundary conditions. The inverse of these local matrices is computed once and applied as a matrix-vector product during time-stepping. This block-Jacobi preconditioner is strictly local to an element and requires no communication. Application of the Helmholtz operator as a matrix-vector multiply requires element edge communication for each CG iteration. At C56N6 resolution using 6 × 6 pressure points per element, the CG solver converges in two iterations at the bottom of the atmosphere and four or five iterations at the top where the gravity wave phase speed is on the order of 300 ms^{-1}.

3. Primitive Equations

The 3-D primitive equations neglect vertical acceleration and are derived from the Navier-Stokes equations by invoking scaling arguments for the atmospheric general circulation. We follow the formulations described in Simmons and Burridge (1981) and Ritchie et al (1995) and introduce a general, terrain-following, vertical coordinate η. This hybrid coordinate $\eta(\,p,\,p_s)$ is a monotonic function of pressure p and is dependent 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 momentum, thermodynamic and continuity equations for frictionless adiabatic motion are given by

$$\begin{aligned} \frac{D\mathbf{v}}{Dt} + f\,\mathbf{k}\times\mathbf{v} + \nabla\phi + R\,T\,\nabla\ln p &= 0 && (3.1)\\ \frac{D\,T}{D\,t} - \frac{\kappa\,T\,\omega}{p} &= 0 && (3.2)\\ \frac{\partial}{\partial\eta}\left(\frac{\partial p}{\partial t}\right) + \nabla\cdot\left(\mathbf{v}\,\frac{\partial p}{\partial\eta}\right) + \frac{\partial}{\partial\eta}\left(\dot\eta\,\frac{\partial p}{\partial\eta}\right) &= 0 && (3.3) \end{aligned}$$

where the material derivative is given by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} + \dot\eta\,\frac{\partial}{\partial\eta}$$

$\mathbf{v}=(\,u,\,v)$ is the horizontal velocity vector, T is the temperature R is the gas constant, c_p is the specific heat at constant pressure and $\kappa = R/c_p$. The geopotential ϕ is defined by the hydrostatic equation

$$\frac{\partial\phi}{\partial\eta} = -\frac{R\,T}{p}\,\frac{\partial p}{\partial\eta}, \qquad (3.4)$$

and the vertical velocity $\omega \equiv Dp/Dt$ is given by

$$\omega = -\int_0^\eta \nabla\cdot\left(\mathbf{v}\,\frac{\partial p}{\partial\eta}\right)\,d\eta + \mathbf{v}\cdot\nabla p. \qquad (3.5)$$

Expressions for log surface pressure $\ln p_s$ and vertical advection of pressure are obtained by integrating the continuity equation (3.3) using the boundary conditions $\dot\eta = 0$ at $\eta = 0$ and $\eta = 1$.

$$\begin{aligned} \frac{\partial}{\partial t}(\ln p_s) &= -\frac{1}{p_s}\int_0^1 \nabla\cdot\left(\mathbf{v}\,\frac{\partial p}{\partial\eta}\right)\,d\eta && (3.6)\\ \dot\eta\,\frac{\partial p}{\partial\eta} &= -\frac{\partial p}{\partial t} - \int_0^\eta \nabla\cdot\left(\mathbf{v}\,\frac{\partial p}{\partial\eta}\right)\,d\eta && (3.7) \end{aligned}$$

The prognostic equations (3.1), (3.2), (3.6) for $\mathbf{v}$, T, and $\ln p_s$ are integrated using either an explicit or semi-implicit time-stepping scheme. The horizontal discretization follows the staggered spectral element formulation of the shallow-water equations described in the previous section. The scalar variables T and $\ln p_s$ reside on the Gauss pressure grid, whereas $\mathbf{v}$ is located at Gauss-Lobatto velocity points. The momentum and thermodynamic equations in curvilinear coordinates are given by

$$\frac{\partial u^i}{\partial t} = -g^{ij}\left[\epsilon_{jk}u^k g\,(f+\zeta) + \frac{\partial}{\partial x^j}\left(\frac{1}{2}u_k\,u^k\right) + \frac{\partial \phi}{\partial x^j} + RT\frac{\partial}{\partial x^j}(\ln p)\right] + F_u^i \quad (3.8)$$

$$\frac{\partial T}{\partial t} = -u^j\,\frac{\partial T}{\partial x^j} + \frac{\kappa\,T\,\omega}{p} + F_T \quad (3.9)$$

where F_u and F_T contain vertical advection terms. The hydrostatic relation (3.4) is used to diagnose the geopotential and the pressure p is obtained from the surface pressure p_s using the definition of the hybrid vertical coordinate. The vertical discretization employs finite differences and is designed to conserve both energy and angular momentum in the absense of external forcing terms. NLEV layers are defined by the 'half-level' pressures

$$p_{k+1/2} = A_{k+1/2} + B_{k+1/2}\,p_s \quad (3.10)$$

for $0 \le k \le$ NLEV. The constants $A_{k+1/2}$ and $B_{k+1/2}$ effectively define the vertical coordinate, where $B = \partial p/\partial p_s$. Prognostic variables are represented by 'full-level' values at pressure levels p_k. The discrete form of the surface pressure tendency equation (3.6) is derived using the product rule and is written as (neglecting horizontal indices)

$$\frac{\partial}{\partial t}\,(\,\ln p_s\,) = -\sum_{k=1}^{\mathrm{NLEV}}\left[\frac{1}{p_s}\frac{\Delta p_k}{g}\frac{\partial}{\partial x^j}(\,g\,u_k^j\,) + u_k^j\,\frac{\partial}{\partial x^j}(\,\ln p_s\,)\,\Delta B_k\right] \quad (3.11)$$

where $\Delta p_k = p_{k+1/2} - p_{k-1/2}$ and $\Delta B_k = B_{k+1/2} - B_{k-1/2}$. Simmons and Burridge (1981) and Ritchie et al (1995) discretize the pressure gradient term as follows

$$\frac{\partial}{\partial x^j}\,(\,\ln p\,)_k = \frac{1}{\Delta p_k}\left[\ln\left(\frac{p_{k+1/2}}{p_{k-1/2}}\right)\frac{\partial}{\partial x^j}\,(\,p_{k-1/2}\,) + \alpha_k\frac{\partial}{\partial x^j}\,(\,\Delta p_k\,)\right] \quad (3.12)$$

where $\alpha_1 = \ln 2$, and for $k > 1$,

$$\alpha_k = 1 - \frac{p_{k-1/2}}{\Delta p_k}\ln\left(\frac{p_{k+1/2}}{p_{k-1/2}}\right). \quad (3.13)$$

An alternative form for (3.12) can be derived using $\partial p/\partial t = (\partial p/\partial p_s)\,\partial p_s/\partial t$. For the hybrid coordinate (3.10), it follows that $\nabla p_{k+1/2} = p_s\,B_{k+1/2}\,\nabla \ln p_s$ and

$$\frac{\partial}{\partial x^j}\,(\,\ln p\,)_k = \frac{p_s}{\Delta p_k}\left[\Delta B_k + \frac{C_k}{\Delta p_k}\ln\left(\frac{p_{k+1/2}}{p_{k-1/2}}\right)\right]\frac{\partial}{\partial x^j}\,(\,\ln p_s\,) \quad (3.14)$$

where $C_k = A_{k+1/2}B_{k-1/2} - A_{k-1/2}B_{k+1/2}$. Thus, the 3-D gradient of $\ln p$ can be computed from the 2-D gradient of $\ln p_s$.

The discrete formulation of the thermodynamic (3.9) and hydrostatic (3.4) equations follows equations (2.21), (2.22) and (2.25) of Ritchie et al (1995). Vertical advection for a variable X is given by

$$\left(\dot\eta\frac{\partial X}{\partial\eta}\right)_k = \frac{1}{2\Delta p_k}\left[\left(\dot\eta\frac{\partial p}{\partial\eta}\right)_{k+1/2}(X_{k+1}-X_k) + \left(\dot\eta\frac{\partial p}{\partial\eta}\right)_{k-1/2}(X_k - X_{k-1})\right] \tag{3.15}$$

If $\mathbf{v}$ is located on the velocity grid (Gauss-Lobatto) and p, p_s, ϕ, T are located on the pressure (Gauss) grid, then it is clear that two terms require interpolation from the pressure grid to the velocity grid. These are the velocity vertical advection $\dot\eta\partial u^i/\partial\eta$ and $RT\partial/\partial x^j(\ln p)$ appearing in the momentum equation. Inspection of (3.15) reveals that the velocity vertical advection requires, in general, two interpolations from the pressure grid to the velocity grid and (3.14) requires one interpolation per layer.

The communication required for the explicit time discretisation of the primitive equations reduces to exchanging the velocity tendency $u^j\partial/\partial x_j$ and $\partial/\partial x_j(\ln p_s)$. Because C^0 continuity for the latter term is required to compute the former, this quantity must be computed in advance during a time step. This can be accomplished without changing the numerics and provides the advantage of a single communication per time step. The size of this communication is $3\times\mathrm{NLEV}+2$, compared to $3\times\mathrm{NLEV}$ for the shallow water equations. In terms of computational complexity, the primitive equations have, in addition to the operations required by the shallow water equations, three 3-D interpolations from the pressure grid to the velocity grid, one 3-D gradient and one interpolation from the velocity to the pressure grid. Of course there is also one additional horizontal 2-D gradient $\partial/\partial x^j(\ln p_s)$.

Pressure on the full levels must be computed for input to the physical forcing terms and this will be discussed in the section on numerical results for the Held-Suarez test. Full level pressures are computed according to equation (3.18) of Simmons and Burridge (1981),

$$p_k = \exp\left[\frac{1}{\Delta p_k}\left(p_{k+1/2}\ln p_{k+1/2} - p_{k-1/2}\ln p_{k-1/2}\right) - C\right] \tag{3.16}$$

where $C = 1$ for $k > 1$ and $C = \ln 2$ for $k = 1$.

4. SEAM Software Implementation

The Spectral Element Atmospheric Model (SEAM) is written in modular Fortran 90. This design permits the rapid development of new model components from proven subcomponents and also provides a sound basis for the performance intercomparison of numerical schemes. SEAM has been carefully designed to achieve high performance from this modular design. In SEAM, Fortran 90 `SEQUENCE` derived types are used not only for data encapsulation but also to collect related data together for cache efficiency. For example, repeatedly accessed data objects such as spectral element derivative and interpolation matrices are stored together in a derived type. This type declaration appears in a Fortran 90 `MODULE` that also contains the interpolation and differentiation functions that apply these matrices. This hides hand optimizations such as loop unrolling in a single module that can be readily customized for different architectures should the need arise. The resultant derivative module performs well, achieving 2.1 GFlops on 8×6 matrix multiplies, or 35% of peak on a four processor IBM SP node.

In order to accommodate different architectures, parallelism in SEAM is achieved using a hybrid MPI/OpenMP programming model. This flexible parallel design permits SEAM to execute in purely threaded, purely message passing, or true 'hybrid' mode with multiple threads per node executing and one thread per node handling the message passing tasks. Running in hybrid mode on SMP clusters has several potential advantages relative to pure message passing. First, hybrid consolidates nodal message traffic into a single large message, reducing network latency overhead. Second, a single MPI process eliminates contention for the network interface, typically a shared resource. Third, shared memory message passing requires an additional copy between communication buffers. Finally, threads can be easily dynamically load balanced.

Hybrid mode also has potential disadvantages relative to pure message passing. Threads face possible cache conflicts while accessing shared memory. The performance of the single message passing thread can be impacted by spin/sleep policies enforced on the remaining threads, which are typically blocked at a barrier synchronization. Threads introduce additional overheads related to creation and synchronization. To address these issues, SEAM employs an SPMD style of shared memory programming. A main driver routine initializes MPI and determines the element graph partitioning. The model is called from within a single OpenMP parallel region of the driver. This approach reduces thread overhead to the minimum required by data dependencies. OpenMP scoping rules for subroutines called from within parallel regions ensure that most of the working set is thread private. The shared data space consists of just communication buffers, related structures and state variables. Only the message passing calls are serialized in hybrid mode: in particular all communication buffer packing and unpacking routines are multi-threaded to reduce overhead.

The communications design of SEAM achieves flexibility and high performance. Messages in SEAM are aggregated over multiple layers and fields to reduce latency. The number of elements, the number of MPI processes and threads per SMP node are all set at runtime via namelist. The cubed-sphere is partitioned at runtime using the METIS 4.0 graph partitioning package of Karypis et al (1995). METIS provides SEAM the capability to run with any number of message passing processes and produces partitionings that minimize total communication cost. On multiprocessor clusters, inter-node message passing via shared memory is typically much faster than intra-node. Exploiting this fact to minimize communication cost is critical for SEAM's pure message passing mode. To address this issue, SEAM employs a hierarchical graph partitioning strategy. METIS is first used to decompose the spectral element grid into nodal subgraphs. Then each nodal subgraph is fed back into METIS to produce subdomains for each process on the node. METIS provides both recursive bisection and K-way graph partitioning algorithms. Recursive bisection partitions a graph by attempting to minimize the number of edge cuts. In addition to minimizing edge cuts, the K-way algorithm also has the ability to minimize total communication volume.

The preconditioned conjugate gradient solver in SEAM is designed to solve multiple 2-D Helmholtz problems for the semi-implicit time stepping scheme. The CG solver core operates via reverse communication and thus is independent of the implementation details of the preconditioner or matrix being inverted. Since solver convergence rates vary across vertical layers, the solver masks out layers as they converge and only operates on those remaining. Finally, inner products in the classical CG algorithm require two global reductions `MPI_allreduce` which scale as $\mathcal{O}(t_l \log_2 P^2)$. To control this cost, we have implemented a latency tolerant variant of the CG algorithm, developed by D'Azevedo et al (1992), requiring one reduction. This saves approximately $\mathcal{O}(t_l \log_2 P)$, per iteration where t_l is the message-passing latency, $t_l \approx 30\mu$sec on the IBM SP with TBMX switch.

5. Numerical Results

A standard test set for evaluating numerical approximations to the shallow water equations in spherical geometry has been proposed by Williamson et al (1992). Test case 1 is the advection of a cosine bell with compact support. This test case is designed to evaluate the advective component of the numerical scheme in isolation and is the only test that does not involve the full equation set. Because the first derivative of the initial field is discontinuous, the cosine bell is a challenge for spectral methods due to the resulting Gibbs phenomena or so-called spectral ringing. Several values for the angle α between the axis of solid body rotation and the spherical coordinate pole are specified and results are reported here for $\alpha = \pi/4$ which advects the cosine bell over the cubed-sphere corner points. The length of the integration is 12 days, corresponding to one complete rotation around the sphere. Figure 2 is a plot of the error metrics for a grid with $96 \times 16 \times 16$ pressure points, where each cube face contains 16 spectral elements. An explicit time step of length $\Delta t = 30$ secs was used for this run. A Boyd-Vandeven filter was applied every 20 mins in order to suppress the growth of aliasing errors. Our Figure 2 should be compared with Figure 5 of Taylor et al (1997), where it can be seen that the l_1 error is reduced by a factor of two. The oscillations visible in the error curves are similar to those observed in Jacob-Chien et al (1995) and are characteristic of spectral methods. They are due to sampling errors as the cosine bell moves through different parts of the grid.

The initial condition for test case 6 is a $R = 4$ wavenumber Rossby-Haurwitz wave. These waves are an ideal test because they represent exact analytic solutions to the nonlinear nondivergent barotropic vorticity equation. Rossby-Haurwitz waves are not closed-form solutions of the barotropic shallow water equations. However, a high-resolution integration of a spectral model can be used to generate a reference solution. A grid resolution of $1734 \times 8 \times 8$ pressure points was chosen in order to compare against the simulation results presented in Figure 11 of Taylor et al (1997). A stable time step for this simulation was $\Delta t = 120$ sec which is eight times the explicit step of $\Delta t = 15$ sec. Test case 6 is particularly difficult for a semi-implicit solver since the mean geopotential height is set at 8000 m and the maximum gravity wave phase velocity approaches 300 ms^{-1}. In this case, the CG solver requires 4 to 5 iterations to converge. The advective Courant number is also limited by wind speeds approaching 100 ms^{-1}. The l_1, l_2, and l_∞ errors are plotted in Figure 3, where it can be seen that all three error measures at 14 days are reduced by a factor of two when compared with the earlier results of Taylor et al (1997). These improvements can be attributed to the use of more velocity points in the staggered grid and exponential convergence. A contour plot of the Rossby-Haurwitz solution after 14 days of integration by the semi-implicit spectral element model is presented in Figure 4. The error field comparing a T213 reference and spectral element solutions at 14 days is plotted in Figure 5. The reference solution has been interpolated to a spherical latitude-longitude projection of the cubed-sphere spectral element grid using the spherical harmonic basis functions and T213 spectral coefficients, truncated to T106 and stored as 32-bit reals.

The shallow water equations encapsulate many of the dynamical aspects of climate modeling. Any numerical method being considered as a basis for a climate dynamical core should perform well on the above test cases. However, the main purpose of these tests is to compare the accuracy of numerical methods and their ability to handle spherical geometry. Because the longest integration is 15 days, these tests do not assess the ability of a dynamical core to generate accurate long term climate statistics. The Held-Suarez (1994) idealized climate forcing is designed to test the dry dynamical core of a GCM for longer integrations. It assumes an ideal gas atmosphere over a rotating sphere with no topography. The flow is not specified as hydrostatic, however, the hydrostatic primitive equations may be employed as described in Section 3. 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 initial state of the atmosphere is hydrostatic and isothermal $T = 300$ K. The model is integrated for 1200 days. Zonally averaged wind and temperature fields are then reported. The Held-Suarez forcings take the following form,

$$\frac{\partial \mathbf{v}}{\partial t} = \cdots - k_{\mathbf{v}}(\, \phi,\, \sigma \,)$$

$$\frac{\partial T}{\partial t} = \cdots - k_T(\, \phi,\, \sigma \,)\; [\, T - T_{\rm eq}(\, \phi,\, \sigma \,)\,]$$

where ϕ is the latitude and $\sigma = p/p_s$ is the vertical sigma coordinate level. The temperature is relaxed to the equilibrium temperature $T_{\rm eq}$ and relaxation rate k_T. The linear damping rate of the wind is given by $k_{\mathbf{v}}$.

6. Performance Results

All performance experiments reported here used 64-bit IEEE floating point arithmetic under the IBM mpxlf90_r compiler version 7.1.0.0. Manual flop counts for the shallow water equations were verified by instrumenting the code to access hardware counters. These agreed to within 1% for the explicit and 3% for the semi-implicit code. We report analytic flop counts which neglect additions by zero at the start of scalar products in matrix-matrix multiplies. These flop counts are more conservative and are about 6% lower than the hardware flop counts. Timings were collected from instrumented code on IBM systems using the recommended real-time clock function `rtc()`, which is thread safe and accurate to within 1 μsec (see IBM 1998). Unless otherwise stated, reported results are the best observed on a dedicated system. Williamson et al (1992) specify shallow water test case 2 (quasi-geostrophic flow) as a performance benchmark; all shallow-water performance results reported here refer to 1 day integrations of this problem. Performance data for the primitive equations were obtained for 1 day integrations using a $dt = 50$ sec time step. Each of the METIS algorithms: recursive bisection (RB), K-way with edge cut minimization (KWAY) and K-way with total volume minimization (TV) were run at NERSC. The experiments at NCAR and NPACI were performed using RB.

The purpose of the first set of experiments is to measure the integration rate of the semi-implicit scheme relative to the explicit for the multi-layer shallow water equations at realistic next-generation climate model resolutions. Three resolutions with 30 vertical layers are studied, C56N6, C84N6, and C112N6. A detailed description of these grids is given in Table 1. The experiments were run in 4-thread hybrid mode on the NCAR IBM SP with a TBMX switch and four 375 MHz Power-3 processors per node. The integration rate acceleration factor of the semi-implicit over the explicit scheme for each resolution is plotted in Figure 6 as a function of processor count. The acceleration factor at small processor counts and high resolutions is suppressed due to cache effects associated with the preconditioner. The average CG solver iteration count grows from 4.2 to 5.9 as the resolution increases, also causing the acceleration factor to decrease. Nevertheless, the semi-implicit integration rate is over two times faster than the explicit at C56N6. Figure 7 shows that 130 years per day is achieved at this resolution.

The second set of experiments compares the performance of SEAM with the parallel spectral transform shallow water model (PSTSWM) from ORNL (Worley and Foster 1994). PSTSWM is a well-known parallel performance benchmark for the spectral transform method. Performance data for PSTSWM were supplied to the authors by Worley (2001) for the most efficient communication algorithm choice (Legendre Transposition). Both codes use 16 layer shallow water equations. Two resolutions are studied, C84N6 and C154N6, which correspond roughly to T85 and T170, respectively. PSTSWM and SEAM data were both collected on similar IBM SP systems with a TBMX switch and four 375 MHz Power-3 processors per node. PSTSWM was run using 4 MPI processes per node. At C84N6 resolution the semi-implicit solver averages 5.1 CG iterations and, as illustrated in Figure 8, accelerates the integration rate by a factor of 1.5, achieving an overall integration rate of 95.3 years per day and a sustained performance of 48.2 Gflops on 432 processes. This is 3.6 times faster than the best integration rate achieved by T85L16 PSTSWM on 128 processes, primarily the result of the better scalability of the semi-implicit SEAM core. C154N6 results are plotted in Figure 9. Hybrid and pure MPI runs are included here for comparison. The hybrid semi-implicit integrates the most efficiently, achieving 19.2 years per day at 63.3 GFlops on 488 processors, a rate 1.33 times faster than the hybrid explicit. Pure MPI runs take advantage of SEAM's hierarchical graph partitioning algorithm to reduce communication costs by mapping subdomains which share edges onto the same node. This strategy improves the performance of the pure MPI mode by 20%. This enables the explicit pure MPI version to achieve 18.9 years/day and 127 GFlops on 536 processors of the NCAR IBM SP.

The C154N6 SEAM was also benchmarked on an IBM SP at NPACI with a 'Colony' switch and eight 375 Mhz Power-3 processors per node. Four parallel configurations of explicit SEAM are studied; MPI with 7 and 8 processes per node, and hybrid with 7 and 8 threads per node. Table 2 records the message passing and thread environment settings used in this experiment. The 16 layer C154N6 results are plotted in Figures 10 and 11. MPI with 7 processes per node outperforms all other configurations, achieving 206.7 Gflops on 134 nodes and an integration rate of 30.6 years per day. Slightly undersubscribing with either 7 threads or 7 message passing processes improves performance for large node counts. In both cases, the fully subscribed nodes appear to behave as an over-subscribed system would. The degradation of 8 process MPI results on large node counts is not fully understood. MPI generally outperforms hybrid. For example, 7 MPI processes per node ranges between 19% and 35% faster than 7-thread hybrid. This suggests that contention for the network interface by MPI processes is not an important factor. It is interesting to note that the computation rate of pure MPI is about 10-15% faster than hybrid code, possibly due to the absence of synchronization points.

Finally, the C154N6 SEAM was run on the NERSC IBM SP with a 'Colony' switch and sixteen 375 MHz Power-3 processors per node. On this machine each node contains two network interface cards in a double-rail configuration. For the shallow water version of SEAM, a sustained performance of 343 Gflops is achieved using 128 nodes, representing 12.5 % of the total peak floating point execution rate of the 1920 processors. The corresponding integration rate for this run is 50.9 years per day. The total sustained Gflops rates are plotted in Figure 12 for both the C154N6 16 level shallow water and primitive equations versions of SEAM. The Held-Suarez test of the primitive equations was run with 15 and 12 MPI processes per node. Due to the higher number of flops to perform, load imbalance is a potential problem for the primitive equations. For 15 MPI processes per node, the optimal METIS algorithm was different for small, medium and large node counts. From 2 (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 RB algorithm performs best on small node counts since it minimizes both total edge cut as well as the communication volume. When the number of nodes exceeds 12, the K-way algorithm offers the best performance, since it provides the best load balance of all the METIS algorithms. At higher node counts, the algorithm which provides acceptable load balance and minimizes total communication volume performs best. This is typically the TV algorithm, although in the case of 121 nodes (1815 processors), RB provides a sufficiently balanced partition and minimal communication volume. The best 12 MPI process results are obtained using only the K-way algorithm and in this case load imbalance has less of an impact on the performance. Our best result of 369 Gflops is achieved on 128 nodes using 12 MPI processes for a total of 1536 processors.

7. Conclusions

Scientific progress in climate modeling depends more on accelerating the integration rate than the resolution. A major goal of our work is to demonstrate that a climate simulation rate of over 100 years per wall clock day is possible on microprocessor based clusters. This simulation rate is an order of magnitude faster than existing climate models and would represent a major advance in geophysical fluid dynamics. We have shown that a scalable semi-implicit spectral element multi-layer shallow water model can achieve an integration rate of 130 years per day. The performance results presented here represent realistic climate resolutions that would be used in production runs. We note that the choice of $N_p = 6$ pressure points is a trade-off between a high climate simulation rate and a much higher sustained flop rate per node. Indeed, the spectral element method begins to resemble the more traditional spectral transform method when the number of Gauss points per element is increased. The correct balance between scalability and numerical accuracy must be achieved in order to extract the full benefits of spectral element numerics. A C154N6 16 level shallow water model achieves 127 Gflops at NCAR, 206 Gflops at NPACI and 343 Gflops at NERSC on IBM SP clusters, using only $\mathcal{O}(10^5)$ horizontal degrees of freedom. An explicit 3-D primitive equations dynamical core for an atmospheric general circulation model at the same resolution achieves 369 Gflops on 128 nodes of the NERSC IBM SP. The shallow water and primitive equation models exhibit very similar scaling characteristics. Their sustained execution rates are also close despite the fact that the primitive equations involve more computations. These observations tend to validate our assumption that the multi-layer shallow water equations are a reasonable performance prototype for the full 3-D primitive equations.

Acknowledgments

The authors would like to thank NCAR, ORNL, NPACI/SDSC and NERSC for providing access to computing resources. Special thanks to George Fuentes at NCAR, Stuart Johnson at NPACI and Jonathan Carter at NERSC for technical assistance and Mary Haley for help with the NCAR Command Language (NCL).

References

Boer, G. J., and B. Denis, 1997: Numerical convergence of the dynamics of a GCM. *Climate Dynamics*, **13**, 359–374.

D'Azevedo, E., V. Eijkhout, and C. Romine, 1992: Reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessors. LAPACK working note 56, University of Tennessee.

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

IBM SC-09-2718-00, 1998: XL Fortran for AIX. Language Reference. First edition. Version 6, Release 1, p. 746.

Iskandarani, M., D. B. Haidvogel, and J. P. Boyd, 1995: A staggered spectral element model with application to the oceanic shallow water equations. *Int. J. Numer. Meth. Fluids*, **20**, 394–414.

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

Karniadakis, G. M., and S. J. Sherwin, 1999: *Spectral/hp Element Methods for CFD.* Oxford University Press, Oxford, England, 390p.

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

Rancic, M., R. J. Purser, and F. Mesinger, 1996: A global shallow-water model using an expanded spherical cube: Gnomic versus conformal coordinates. *Q. J. R. Meteorol. Soc.*, **122**, 959–982.

Ritchie, H., C. Temperton, A. Simmons, M. Hortal, T. Davies, D. Dent, and M. Hamrud. Implementation of the semi-Lagrangian method in a high-resolution version of the ECMWF forecast model. *Mon. Wea. Rev.*, **123**, 489–514.

Ronquist, E. M., 1988: *Optimal Spectral Element Methods for the Unsteady Three Dimensional Navier Stokes Equations*, Ph.D Thesis, Massachusetts Institute of Technology, 176p.

Sadourny, R., 1972: Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.*, **100**, 136–144.

Simmons, A. J., and D. M. Burridge, 1981: An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates. *Mon. Wea. Rev.*, **109**, 758–766.

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

Thomas, S. J. and R. D. Loft, 2001: Parallel semi-implicit spectral element methods for atmospheric general circulation models. *J. Sci. Comput.*, **15**, 499–518.

Tufo, H. M. and P. F. Fischer, 1999: Terascale spectral element algorithms and implementations *Proceedings of Supercomputing 99.*

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

Worley P. H., and I. T. Foster, 1994: Parallel Spectral Transform Shallow Water Model: A Runtime-tunable parallel benchmark code. *Proceedings of the Scalable High Performance Computing Conference*, eds. J. J. Dongarra and D. W. Walker, IEEE Computer Society Press, 207–214.

Worley P. H., 2001: *Personal communication.*

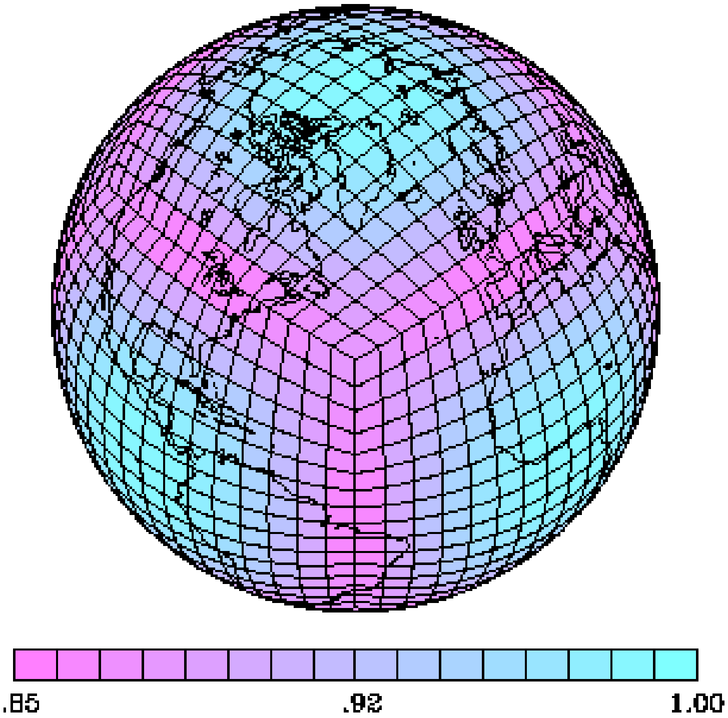

Figure 1: Cubed-sphere spectral element grid.

Table 1: A comparison of spectral transform versus spectral element grids. Given N_v velocity points, $N_p = N_v - 2$ pressure points and K elements on an edge, a cubed-sphere grid is defined according to $K(N_v - 1)$ degrees of freedom along a cube face edge. Equivalent grid spacing is given at the equator.

model	spectral coefs	grid points	semi-imp dt (sec)	equ. grid ΔX (km)
T63	4160	18432	900	208
T85	7482	32768	600	153
T127	16256	73728	450	104
T170	29412	131072	300	78

model	p-grid points	v-grid points	semi-imp dt (sec)	explicit dt (sec)	equ. grid ΔX (km)
C56N6	13824	18816	900	100	179
C84N6	42336	31104	600	72	134
C112N6	55296	75264	450	50	90
C154N6	104544	142296	300	36	65

Table 2: Message passing and thread IBM environment variables.

Environment variable	Setting	Purpose
MP_SHARED_MEMORY	yes	sets on node message passing via shared memory
MP_CSS_INTERRUPT	no	sets whether message packets cause interrupts
MP_INTRDELAY	100	signal handler/service thread wait time
MP_SINGLE_THREAD	no	sets single threaded message passing.
MP_EAGER_LIMIT	64K	limit above which rendezvous protocol must be used.
XLSMPOPTS	delay=500	number of delays before a spin wait
XLSMPOPTS	spins=0	number of spins before a yield
XLSMPOPTS	yields=0	number of yields before a sleep

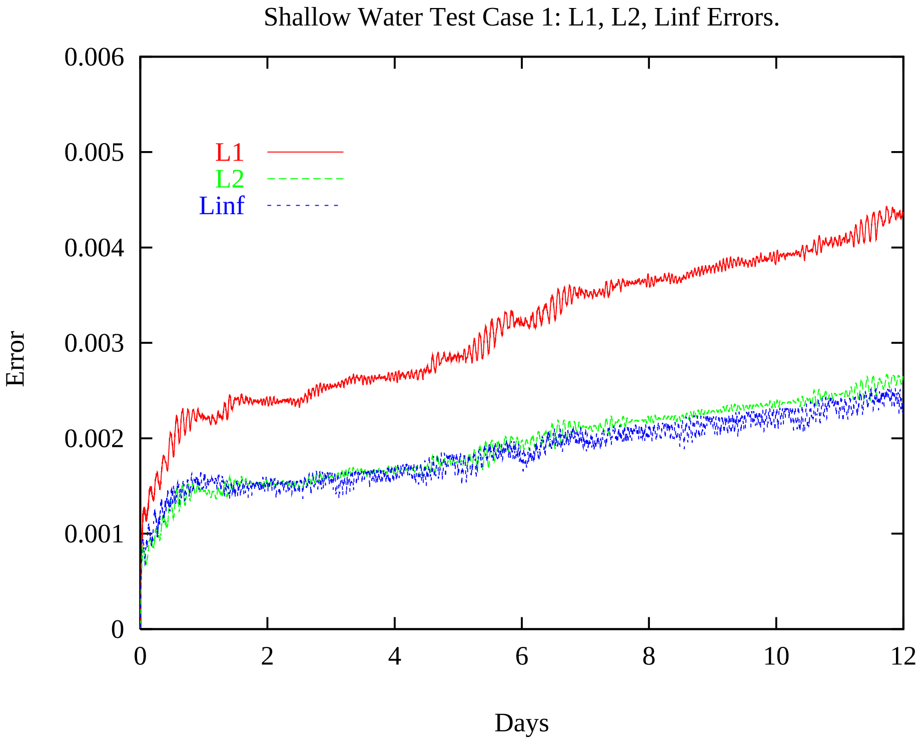

Figure 2: SWE Test Case 1. Cosine Bell. Explicit model. l_1, l_2 and l_∞ errors for 12 day integration. Grid size: 96 × 16 × 16 pressure points.

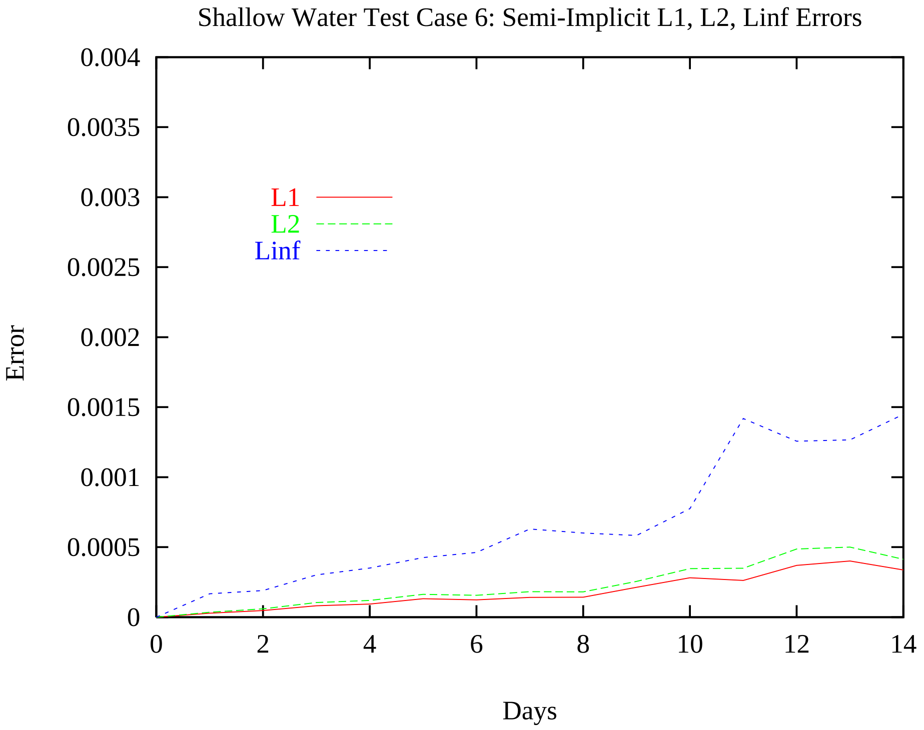

Figure 3: SWE Test Case 6. Rossby-Haurwitz Wave. Semi-implicit model. l_1, l_2 and l_∞ errors for 14 day integration.

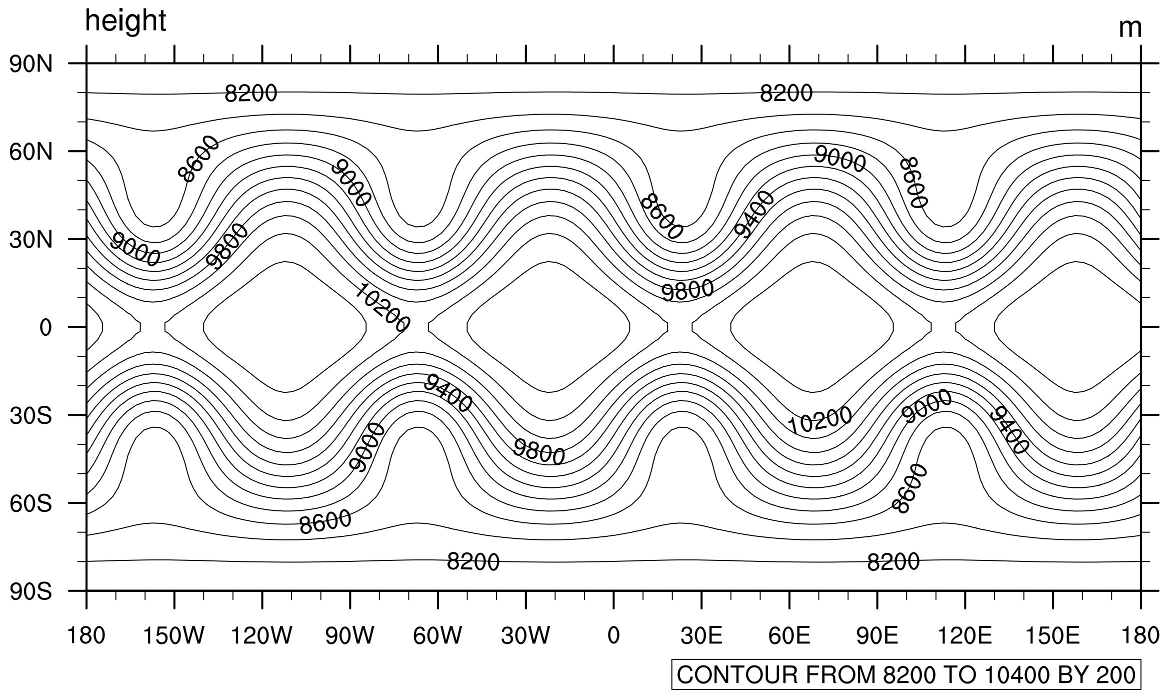

Figure 4: SWE Test Case 6. Rossby-Haurwitz Wave. Semi-implicit model. Geopotential height field after 14 day integration. Grid size: 1734 × 8 × 8 pressure points. 200 m contours.

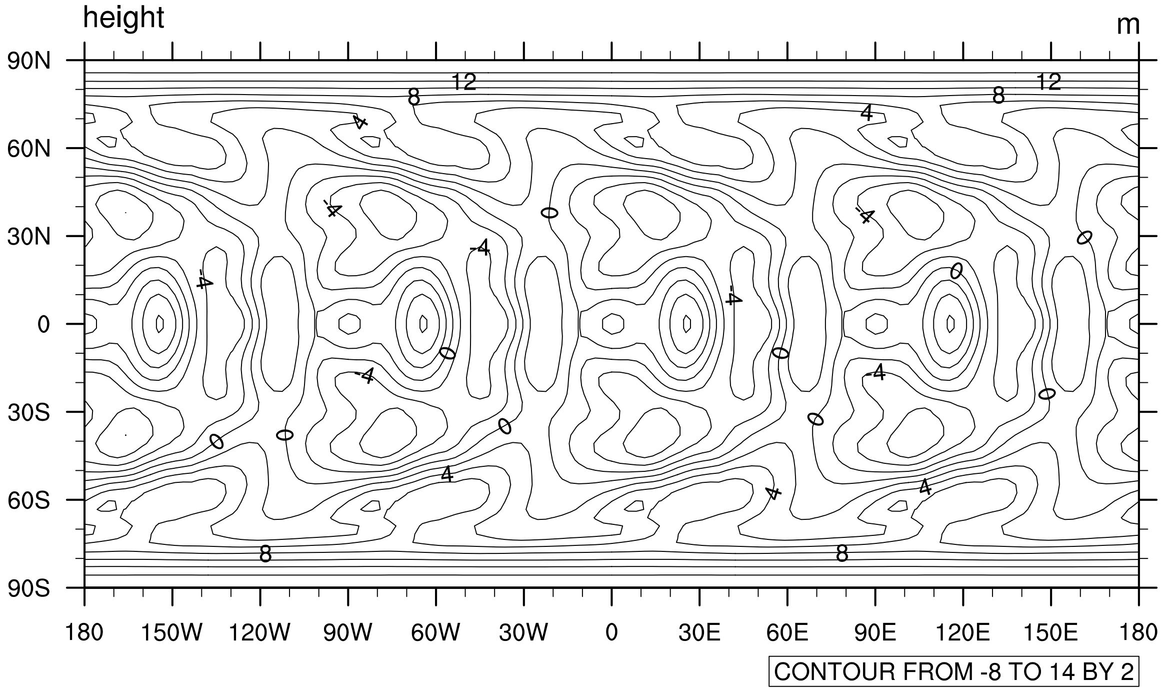

Figure 5: SWE Test Case 6. Rossby-Haurwitz Wave. Semi-implicit model. Geopotential height error field (vs. T213 reference) after 14 day integration. Grid size: 1734 × 8 × 8 pressure points. 2 m contours.

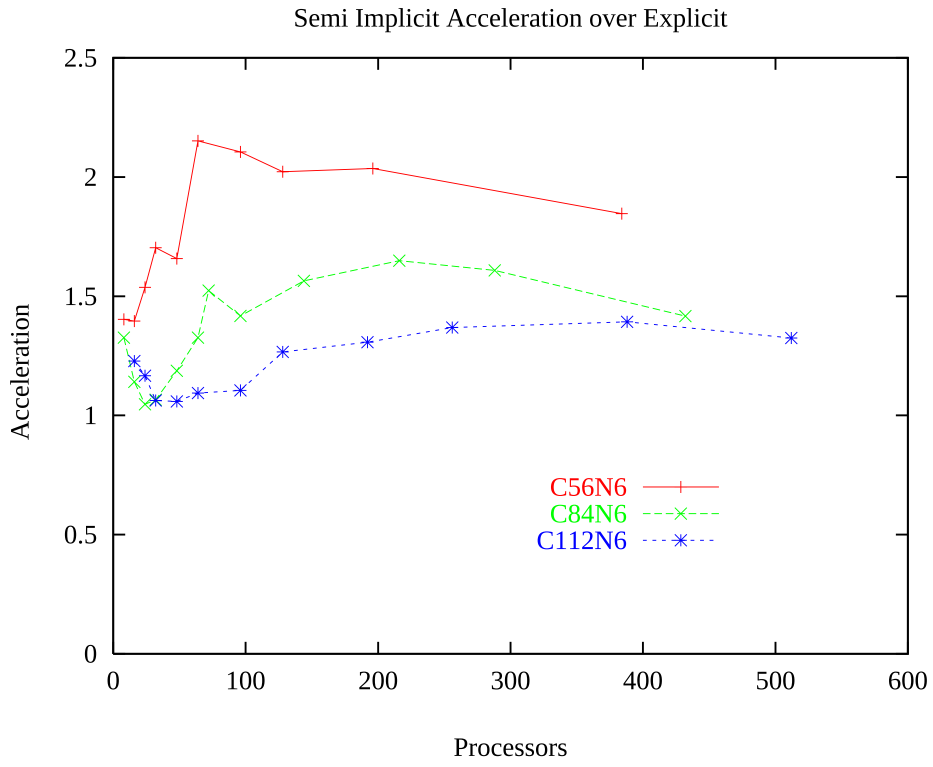

Figure 6: Semi-implicit integration rate acceleration factor over explicit.

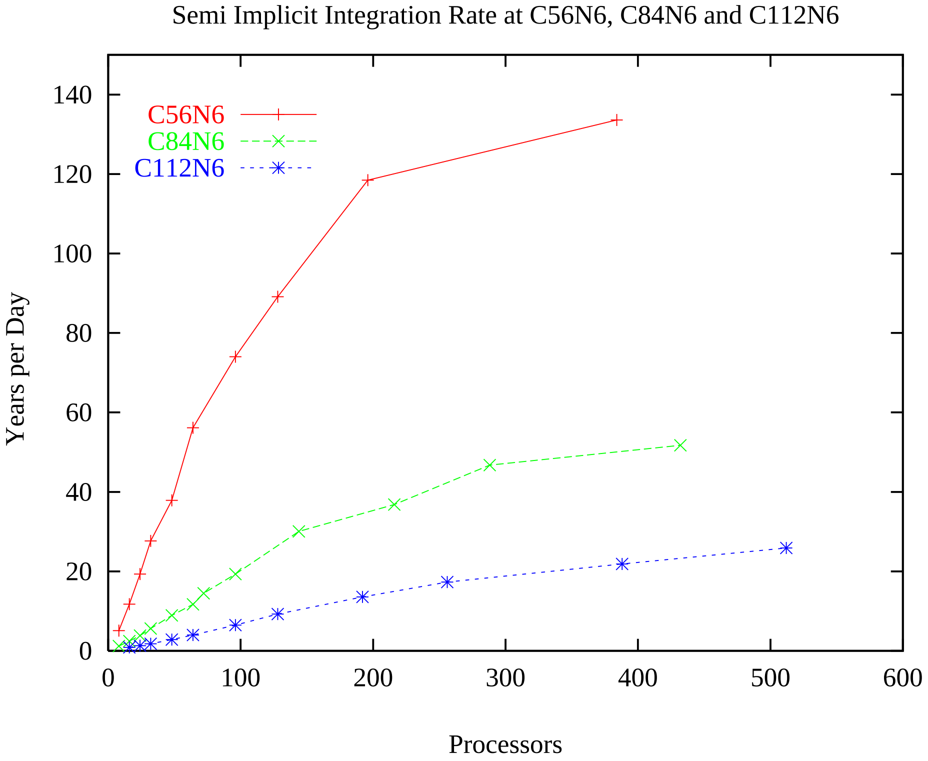

Figure 7: Integration rates for 30 layer spectral element models.

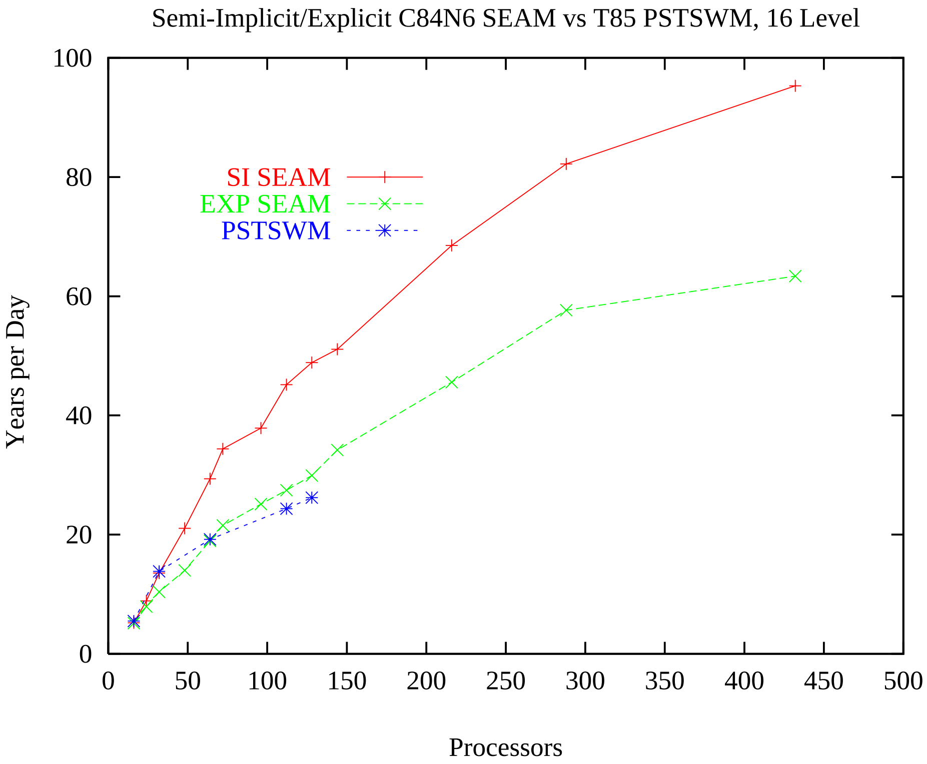

Figure 8: Integration rate inter-comparison. C84N6 SEAM vs T85L16 PSTSWM.

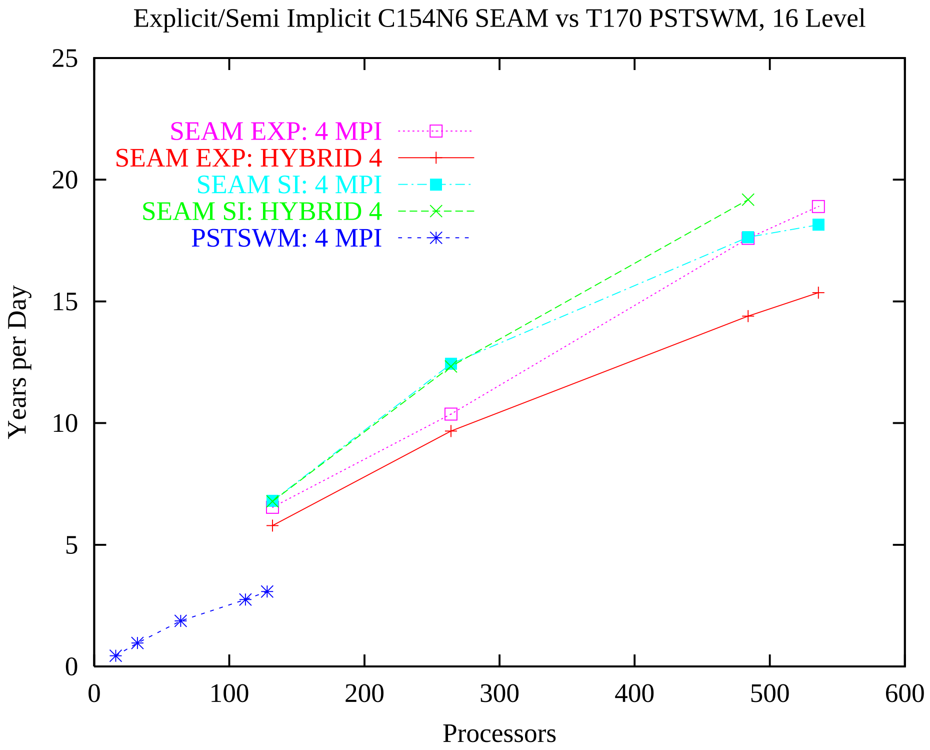

Figure 9: Integration rate inter-comparison. C154N6 SEAM vs T170L16 PSTSWM.

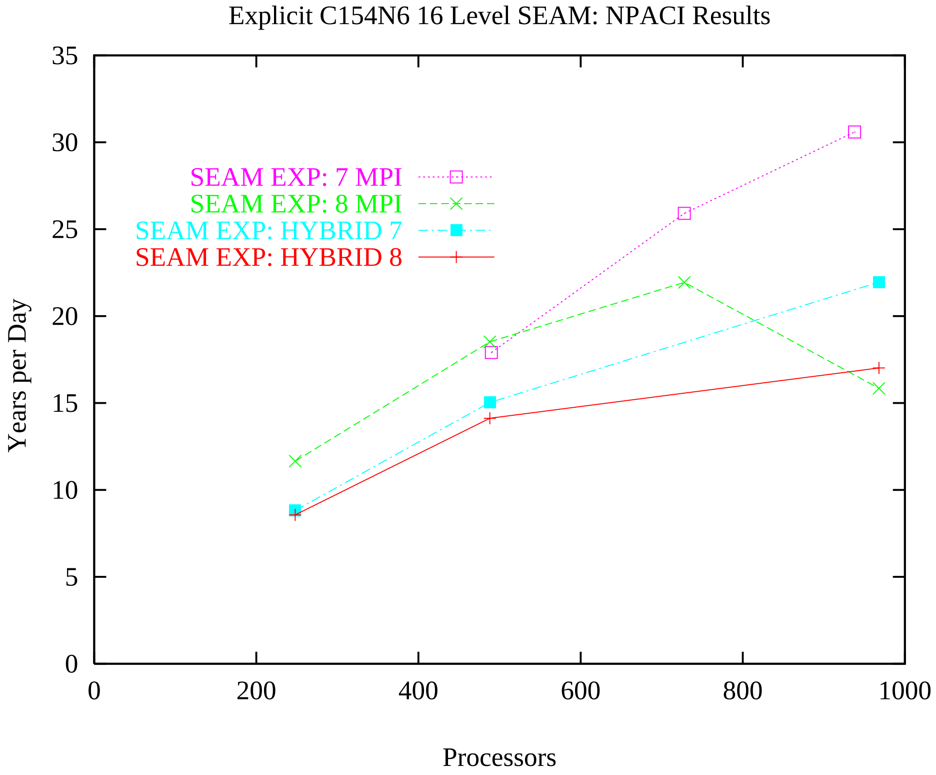

Figure 10: Integration rate. C154N6 SEAM.

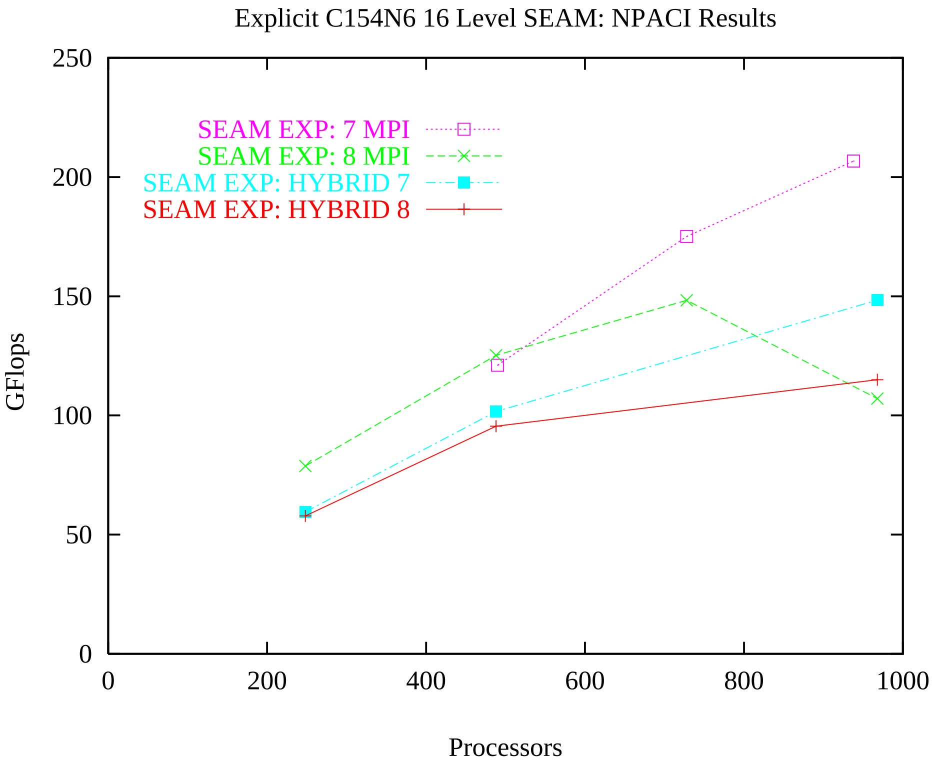

Figure 11: Total Gflops. C154N6 SEAM.

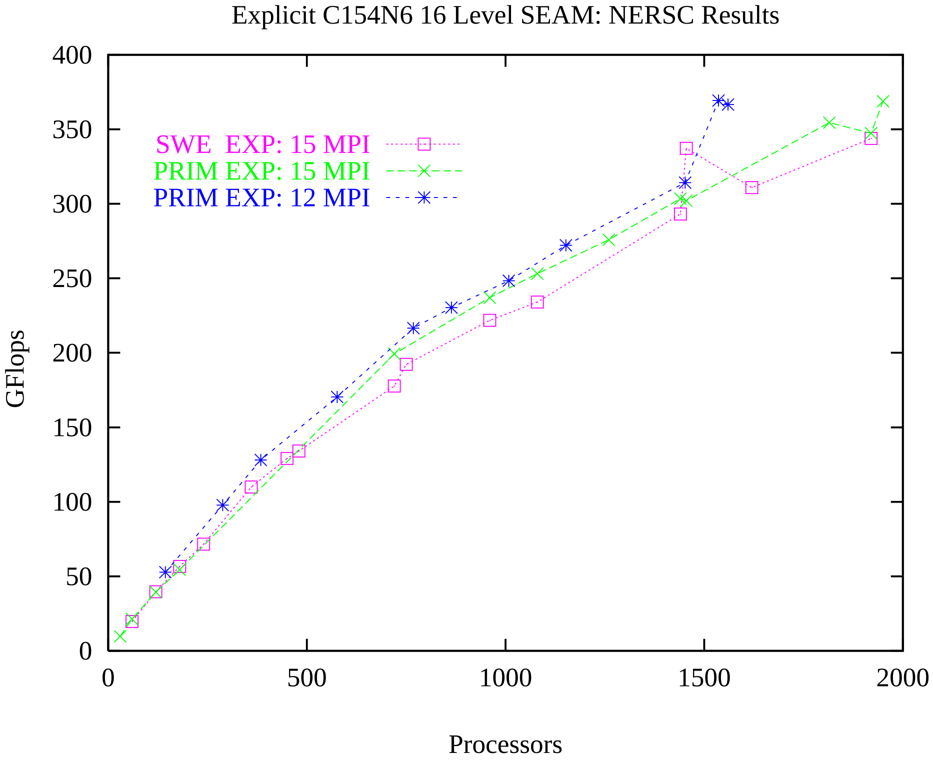

Figure 12: Total Gflops. C154N6 SEAM.