ELSEVIER

Parallel Computing 21 (1995) 1677–1694

PARALLEL COMPUTING

Practical considerations in development of a parallel SKYHI general circulation model

Philip W. Jones [a,*], Christopher L. Kerr [b], Richard S. Hemler [b]

[a] *Theoretical Division, T-3 MS B216, Los Alamos National Laboratory, Los Alamos, NM 87544, USA*

[b] *Geophysical Fluid Dynamics Laboratory / NOAA, Princeton University, Princeton, NJ, USA*

Received 1 April 1995; revised 18 June 1995

Abstract

We have developed a parallel version of the SKYHI atmospheric general circulation model. The new parallel model has been designed for shared and distributed memory machines that support data parallel, message passing or worksharing programming paradigms. The newly developed model has a framework that makes the code easier to understand, maintain, and modify, increasing the model's flexibility and scientific productivity. Numerous model changes are described (code design, programming models, language choice, data decomposition, communications, table lookups, memory management, and i/o) that were necessary to develop the model. The performance and verification of the model is described on several systems including a shared-memory machine with high-level worksharing and a distributed-memory system with a data parallel programming paradigm.

Keywords: Atmospheric general circulation model; Climate modeling; Shared-memory parallel computing; Message passing; Spectral transform

1. Introduction

SKYHI is an atmospheric general circulation model developed at the National Oceanic and Atmospheric Administration's (NOAA's) Geophysical Fluid Dynamics Laboratory (GFDL) to study the dynamics, climate and chemistry of the Earth's atmosphere [3]. The SKYHI model simulates the Earth's atmosphere from the surface to an altitude of approximately 80 km and is well-suited for addressing problems in upper troposphere and middle atmosphere dynamics, including polar

* Corresponding author. Email: pwjones@lanl.gov

SSDI 0167-8191(95)001021-9

stratospheric warmings and wave-mean flow interactions in the stratosphere and mesosphere. In addition, SKYHI is used to address chemical and tracer-transport problems including stratospheric ozone depletion and climatological impacts of greenhouse gas changes.

We describe here an effort to develop an improved version of the SKYHI model which is able to take advantage of parallel computers. The new version of SKYHI supports both shared and distributed memory systems with an arbitrary number of processors by providing support for data parallel, message passing and work sharing programming models. While developing this new version of SKYHI, we have also taken the opportunity to redesign the code to make it more modular and flexible. The new version is easier to understand, maintain and modify and thus increases scientific productivity. To minimize the impact of our changes to current production users, we have been careful not to adversely affect the performance of the original code. We have also avoided changes that could seriously disturb the ongoing development of new model packages.

In the next section, we will first describe briefly the SKYHI model. In subsequent sections, we will detail the major changes we have made to SKYHI, discuss the performance and verification of the new version on various machines, and present future directions for the new model.

2. The SKYHI model

Most atmospheric general circulation models utilize the primitive equations under the hydrostatic approximation. The vertical coordinate is generally either pressure or a terrain-following sigma ($\sigma = p/p_{surface}$) or a hybrid of the two. In σ coordinates (pressure coordinates are similar in form), the equations for eastward and northward horizontal winds (u, v), temperature (T), and water vapor mixing ratio (r) are

$$\frac{\partial(p_s u)}{\partial t} = -D_3(u) + \left(f + \frac{\tan\theta}{a}u\right)p_s v - \left[RT\frac{\partial p_s}{a\cos\theta\partial\lambda} + p_s\frac{\partial\phi}{a\cos\theta\partial\lambda}\right] + {}_H F_\lambda + {}_V F_\lambda, \tag{1}$$

$$\frac{\partial(p_s v)}{\partial t} = -D_3(v) - \left(f + \frac{\tan\theta}{a}u\right)p_s u - \left[RT\frac{\partial p_s}{a\partial\theta} + p_s\frac{\partial\phi}{a\partial\theta}\right] + {}_H F_\theta + {}_V F_\theta, \tag{2}$$

$$\frac{\partial(p_s T)}{\partial t} = -D_3(T) + \frac{R}{c_p}\frac{T\omega}{\sigma} + \left[\frac{\partial(p_s T)}{\partial t}\right]_C + \left[\frac{\partial(p_s T)}{\partial t}\right]_{RAD} + {}_H F_T + {}_V F_T, \tag{3}$$

$$\frac{\partial(p_s r)}{\partial t} = -D_3(r) + \left[\frac{\partial(p_s r)}{\partial t}\right]_C + {}_H F_r + {}_V F_r, \tag{4}$$

where θ is latitude, λ is longitude, a is the radius of the earth, f is the Coriolis parameter, R is the gas constant for air, c_p is the specific heat of air at constant pressure, ϕ is the geopotential height, p_s is the surface pressure, ω is the vertical p velocity, the subscripts C and RAD denote convective adjustment and radiation terms and ${}_{H,V}F$ are the horizontal and vertical sub-grid scale diffusion terms to be described later. D_3 is a three-dimensional divergence operator

$$D_3() = \frac{\partial[()p_s u]}{a\cos\theta\partial\lambda} + \frac{\partial[()p_s v\cos\theta]}{a\cos\theta\partial\theta} + \frac{\partial[()p_s\dot{\sigma}]}{\partial\sigma}. \tag{5}$$

The surface pressure p_s and vertical σ and p velocities are found from the mass continuity equation [2].

Atmospheric models often divide the terms in the above equations into two types, dynamics and physics. Dynamics refers to the momentum equations and the advection and diffusion terms in the remaining equations. With the exception of subgrid-scale diffusion, dynamical processes are not parameterized and are computed as accurately as possible. Differences between atmospheric GCM dynamics are primarily in the different numerical algorithms employed to compute derivatives accurately while conserving quantities like momentum or energy.

The physics terms in the above equations are the radiative heating, convection and condensation terms. These terms represent processes which are either too complex to treat realistically or occur at much smaller spatial scales than can be resolved by global models. They are highly parameterized and take place within vertical columns. The physics in each column can thus be done in parallel, but parameterizations are often computationally intensive and can take a significant amount of time even when computed in parallel. Choices of physical parameterizations can differ greatly between atmospheric GCM's. The implementation of model dynamics and physics used by SKYHI will now be described.

2.1 Grid

SKYHI is a grid-point model on an Arakawa "A" grid [1], meaning all prognostic variables are co-located rather than staggered. The horizontal grid is a latitude-longitude grid with constant spacing in both longitude and latitude. The vertical coordinate is a hybrid grid with modified sigma coordinates in the lower layers and pressure coordinates above the 321 mb level [3]. Typical recently-used resolutions for the SKYHI model are shown in Table 1.

Table 1
Typical recently-used SKYHI resolutions

Resolution	Domain size $N_{lon}\times N_{lat}\times N_{vert}$	Time step (seconds)
n30	100× 60×40	225
n90	300×180×40	60

2.2 Time-stepping

Prognostic variables are advanced in time using an explicit leap-frog scheme with periodic backward Euler steps to damp computational instabilities. Fully explicit methods are ideally suited for parallel computing as they do not require solving linear systems. The disadvantage to explicit methods is that they must satisfy time-step constraints like the Courant condition. This is particularly problematic with a latitude-longitude grid as the grid lines converge near each pole. In order to avoid excessively small time-steps due to small grid spacing near the pole, a Fourier filter is used with a latitude-dependent wave-number cutoff. Filtering is performed in the azimuthal direction poleward of 50° and the wave-number cutoff is chosen to maintain a relatively uniform azimuthal resolution [13]. The Fourier filter requires information from all points on a latitude circle and requires volume communication on distributed memory machines as discussed in Section 3.5.

2.3 Dynamics

The two most common methods used in atmospheric GCMs for computing horizontal dynamical terms are the spectral transform method [2] and finite-difference based methods [1]. Spectral transform methods compute linear terms in spectral space using spherical harmonic basis functions. Such methods are more accurate than finite-difference methods at the same resolution and eliminate the small time-step constraints for grids which converge near the poles. However, the spectral transforms required are computationally intensive and require information from the entire domain, making an implementation on distributed-memory machines more difficult than finite-difference methods [6,8]. Both methods use finite-differences for vertical dynamics terms.

SKYHI dynamics utilizes a finite-difference scheme. Vertical derivatives are computed using second-order centered differences for the wind and temperature and fourth-order differences for moisture and tracers. Horizontal derivatives for all fields are computed using second-order differences based on a "box" method [14]. Using this method, the finite-difference form of each derivative for an arbitrary variable U is

$$\frac{1}{a\cos\theta}\frac{\partial U}{\partial\lambda} = (U_{i+\frac{1}{2},j} - U_{i-\frac{1}{2},j})\frac{\Delta\theta}{2a\,\Delta\lambda\cos\theta_{i,j}\,\sin(\frac{\Delta\theta}{2})} \tag{6}$$

$$\frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}(U\cos\theta) = U_{i,j+\frac{1}{2}}\frac{\cos\theta_{j+\frac{1}{2}}}{2a\cos\theta_j\sin(\frac{\Delta\theta}{2})} - U_{i,j-\frac{1}{2}}\frac{\cos\theta_{j-\frac{1}{2}}}{2a\cos\theta_j\sin(\frac{\Delta\theta}{2})} \tag{7}$$

where the half points denote values at the cell wall in a particular direction (e.g. $u_{i,j+1/2}$ refers to the u velocity on the north cell wall). Because SKYHI uses an "A" grid with all values defined at cell centers, the values at cell walls are

determined using a simple two-point average of the two neighboring cell centres. The above schemes are second-order accurate and require information only from nearest neighbors, simplifying the implementation on distributed-memory machines.

Subgrid-scale diffusion in SKYHI utilizes a second-order diffusion with a Smagorinsky eddy diffusivity based on the local wind shear. The vertical diffusion is also a second-order scheme with a diffusivity based on the Richardson number [12].

As shown in Eqs. (1) and (2), the pressure gradient force consists of two terms, namely the horizontal gradient of geopotential height and the horizontal gradient of surface pressure. Near steep topography, this may result in two large gradients similar in size but of opposite sign. In pressure coordinates, the pressure gradient force is a single term and is not subject to this error. To avoid roundoff errors, a vertical interpolation to pressure coordinates is performed so the geopotential gradient can be computed on pressure levels. During such an interpolation, the result must be checked to insure the pressure coordinate is not below the level of the topography.

2.4 Radiation

The absorption of shortwave radiation by O_3, O_2, H_2O, and CO_2 is computed using a parameterization of Lacis and Hansen [15]. Longwave radiative transfer is computed using table lookups to determine emission and transmission coeffcients; the tables themselves are pre-computed using detailed line-by-line integrations in 15 spectral bands [4,5]. Because the radiation package is computationally intensive, radiative heating terms are not computed at every time step. In typical simulations, the radiation routines are called only every four simulated hours. Radiative processes require information from an entire column and it is important to keep such data local. Lookup tables must also be allocated efficiently. Shortwave radiation can potentially create load imbalances when a diurnal cycle is simulated as it depends on the location of the sun (primarily a longitudinal variation) and on the distribution of clouds (a zonal variation).

2.5 Condensation and convective adjustment

After all other forcing terms have been computed, condensation and convective adjustment are calculated. If the predicted temperatures result in an unstable lapse rate over two or more contiguous vertical levels in a column, convective adjustment must be performed. In sub-saturated layers, this is accomplished by adjusting the temperature so that a dry adiabatic lapse rate is achieved and total energy is conserved. If the layer is super-saturated (regardless of whether the lapse rate is stable or unstable), the moisture and temperature fields are adjusted to the moist adiabatic lapse rate, with the excess mositure assumed to condense and precipitate out [12]. The saturation vapor pressure necessary for moist convection is read from a lookup table. As with radiation, lookup tables and load imbalances (due to convection occurring more often in tropical regions for example) may create problems for parallel implementations.

2.6 Other physics

SKYHI includes prognostic time-tendency equations for ground temperature, soil moisture, snow cover and a variable number of tracers. The land surface temperature is computed using a prognostic equation for a single soil layer [7] and a simple bucket model is used for soil moisture [12]. Tracers follow an equation similar to Eq. (4) with the convective term replaced by a user-supplied source/sink function. Cloud cover, pack ice, and sea surface temperatures are currently prescribed. The land-air interactions are relatively simple parameterizations based on bulk aerodynamic formulae for the transfer of heat, momentum and moisture. With the exception of the tracers which add an additional three-dimensional prognostic field, these processes consume very little time in the current version of SKYHI.

2.7 SKYHI model structure

The structure of the new version of SKYHI is shown in Fig. 1. The program begins with initialization routines in which constants, tables and prognostic variables are defined either internally or through input files. After initialization is complete, time integration of the model equations begins with the computation of time-manager switches and flags. Certain physical quantities that depend only on time (e.g. solar zenith angle) are also computed at this time. Following these routines is a spatial loop over the number of horizontal partitions or "chunks" of the spatial domain. This spatial loop integrates the prognostic equations in time, archives, and computes partial sums of global and hemispheric diagnostic integrals. Note that radiation, diagnostic sums, and data archiving are not required at every time step; the frequency of these operations are supplied by the user at run time through an input namelist. After this loop is complete, a second spatial loop completes the global and hemispheric sums by combining the partial "chunk" sums computed in the first loop. The latter sum has been isolated into a second loop to assure a consistent summation order and provide reproducible integrals and checksums. After completion of any wrap-up procedures necessary for archive files, the time step ends and the next time step is started.

3. Code changes

3.1 Code design

Considerable effort has been made to restructure the model for greater flexibility and scientific productivity. The most important change was to adopt a structure which groups routines into separate packages for dynamics, radiative processes, adjustment physics, utilities and diagnostics. Prior to this change, components of some processes were spread across different subroutines, making modifications to the processes difficult to implement. By consolidating the elements of each process

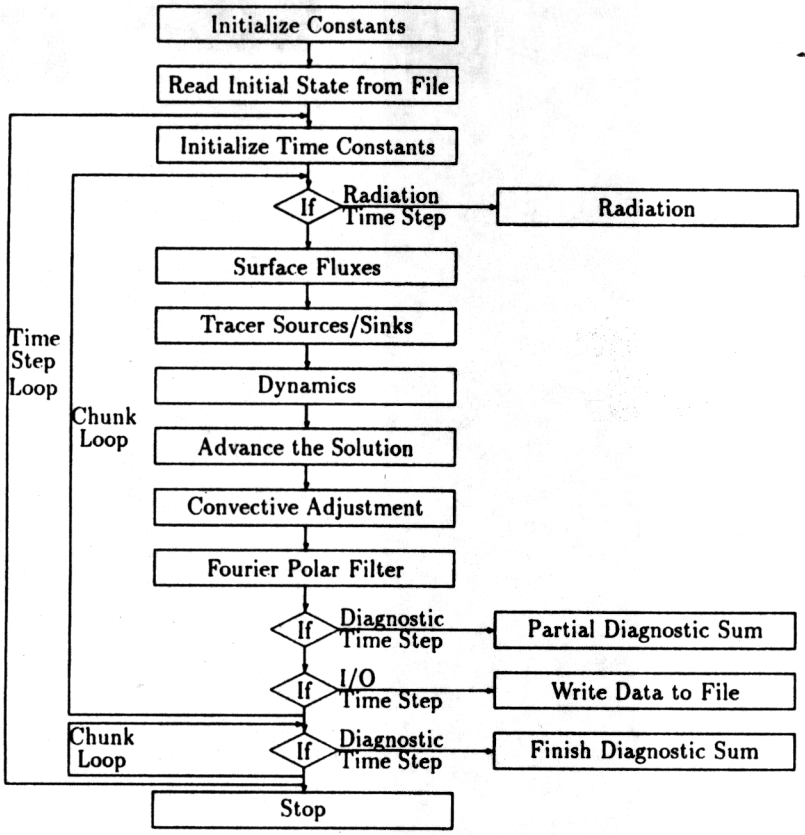

Fig. 1. SKYHI model structure.

into a definable package, it becomes much easier to modify the package, and to test and evaluate alternative parameterizations. Also, machine-dependent routines are now isolated in separate routines, allowing easy substitution of appropriate code for a particular machine.

The main program is now used simply to call the model subroutines, and does not perform any calculations. This format makes the large-scale model structure more obvious to the user, since the "big picture" is not cluttered by details. The details of the model packages are likewise pushed into lower level subroutines of the package, allowing the user to successively see more and more levels of detail, if desired, rather than having the various levels intermixed. This structure, along with increased in-line documentation, allows the user to better understand the model structure to the level desired, and to leave lower level details as "black box" code.

In the original code, much of the data was statically allocated in global common blocks. In the new version, much of the data is dynamically allocated within the packages where the data is used. As a result, the amount of memory required by the model has been reduced. Another consequence of this change is that data must

now be passed through an argument list and data needed by the package must be explicitly defined. The use of argument lists and explicit declarations provides an aid to understanding the purpose of each routine and how data flows through the model.

3.2 Programming models

SKYHI supports three different parallel programming models: data parallel, message passing, and worksharing. Support for these models is accomplished within a single source code with a few exceptions. These exceptions have been isolated to a small number of machine dependent routines and a description of each is given below as they pertain to the items under discussion.

In the data parallel programming model, prognostic and work arrays utilize a global address space and operations are carried out for all data simultaneously through the use of array syntax. Array operations require arrays to conform in shape and size, thus requiring additional memory in some cases. Because of the global address space, access to non-local data can be achieved using array indices, although optimal data access is generally achieved using Fortran 90 intrinsic functions like CSHIFT and SUM. On distributed-memory machines, arrays are partitioned across the machine with "distribution" directives. The model's eastern and western cyclic boundary conditions are enforced through horizontal communications routines (i.e. the CSHIFT intrinsic function). At the northern and southern boundaries, the computational domain is extended and boundary conditions are imposed by filling the extra rows with appropriate data.

The message passing programming model requires that prognostic arrays, work arrays and tables be allocated to individual processors with each processor using its own local address space. Access to non-local data is achieved using explicit passing of messages. To minimize non-local data requirements, ghost cells containing surrounding data are added to each processor's domain of computation; the number of ghost cells is determined by the order of the finite-difference scheme. The setting of ghost cells from neighboring processors is accomplished though a communications routine called at the beginning of each time-step. Global boundary conditions are also imposed during this update step.

The worksharing programming model on shared-memory systems utilizes a high-level approach to parallelism. With this approach, individual processors are assigned to work on separate "chunks" in the spatial domain of the prognostic spatial loop. Statically-allocated prognostic and work arrays must be explicitly declared as private to prevent processors from accessing the data used by other processors. Tables are shared, thus eliminating the need for multiple copies of table data. The implementation of boundary conditions exactly follows the data parallel programming model.

3.3 Language choice

SKYHI is written in Fortran 77 (F77) with some Fortran 90 (F90) extensions including array operations, dynamic array assignments, function assignments and

namelist input processing. For vendors who do not support the above F90 features, translation of these extensions (except namelists) to F77 can be accomplished using commercially-available translators.

Conditional array operations are written in both F77 "if" statements and in F90 "where" statements; the appropriate choice is enabled using the C language preprocessor. The support of both is necessary because of current limitations of the "where" statement. Compared with "if" statements, "where" statements can increase memory and computational requirements and introduce complicated logical structures. These differences occur because "where" statements in the current Fortran standard cannot support either nesting or indirect addressing. In some implementations, operations within a "where" block are performed over the entire array and the logical mask is not applied until the results are stored. Such an implementation performs unnecessary computations and prevents the "where" and "elsewhere" blocks from being processed in parallel. Compressed index implementation (conditional vector merges) is not standard across machines and has been avoided in the new version of SKYHI.

High Performance Fortran (HPF) statements are included to describe the layout of data required for data parallel programming on distributed-memory machines. Because only a few vendors currently implement HPF directly, HPF distribution directives are translated by the C language preprocessor into the appropriate vendor syntax. As more vendors support full F90 and HPF compilers and as these compilers become more mature, much of the preprocessing and machine-dependent routines can be eliminated.

3.4 Data decomposition

Because shared-memory machines historically had limited memory, the SKYHI model was originally designed with latitude domain decomposition. This decomposition may not be suitable for distributed-memory systems because the number of processors can easily exceed the number of latitudes. Even if the number of processors is less than the number of latitudes, it would be better to utilize a general block distribution in the horizontal as this would minimize the surface-to-volume ratio and hence the ratio of communication to computation. We have restructured the model to support any block distribution in the horizontal . The original latitude decompostion and the data-parallel implementation (no decomposition) are subsets of this more general decomposition. No decomposition in the vertical is performed because many physics parameterizations (e.g. radiation, adjustment physics and vertical diffusion) require extensive communication in the vertical direction.

3.5 Communications

Different programming paradigms on distributed-memory machines use different methods to obtain non-local data. Within the data parallel model, communications can be performed using F90 intrinsic functions like CSHIFT and SUM. In the

message passing model, explicit SEND/RECEIVE calls are used. Because of these differences, all communications have been isolated into routines which are machine dependent. The appropriate routines for a given machine are then chosen at compile time.

The nearest-neighbor communications required for computing horizontal derivative terms in the dynamics reduce to a set of recognizable stencil patterns, including a five-point stencil, a three-point stencil in each horizontal direction and occasional two-point stencils. These patterns were originally encapsulated into a set of machine dependent stencil routines. However, the stencil formulation was found to adversely affect the performance of the original code on shared-memory machines. First, the stencil structure forced an intermediate storage step to store the stencil weights. Second, using stencils to perform simple sums or differences of neighbors introduced unnecessary multiplications of scalar weights. Last, the most general form of the stencils required some computation on boundary points when such computations are not always necessary. For some decompositions with a large ratio of boundary points to physical points (e.g. the latitude decomposition), additional computations on boundary points can increase the computational time by a factor of two. The stencil formulation was thus abandoned and horizontal nearest-neighbor communications are performed using shift functions (which simply call the F90 CSHIFT functions when available).

The Fourier polar filter requires knowledge of model data on a complete latitude circle. Such information is not available when longitude decomposition is allowed. For those decompositions, a transpose method [6,8] is utilized in which a matrix transpose is performed to ensure all necessary longitude data is locally available. A local Fourier transform and filter can then be performed, followed by a second transpose to return the filtered data to the original location. A machine-dependent matrix transpose algorithm is provided for distributed-memory machines.

Some routines use global communications corresponding to F90 intrinsics SUM, MAXVAL, MINVAL and COUNT. In SKYHI, subroutines are provided with a consistent argument list so that equivalent machine-dependent code can be supplied where the F90 intrinsics are not available.

3.6 Table lookups

There are a variety of tables used in SKYHI's model physics including saturation vapor pressure and radiative transmission functions. In the original code, table lookups were performed using indirect addressing into table arrays. In order to avoid communications on distributed-memory machines, we allocate a local copy of tables on each processor. The tables are relatively small so allocating multiple copies does not require a large amount of memory. Again, we have found it useful to provide machine-specific routines for this process.

3.7 Memory management

Due to machine memory limitations as described in Section 3.4, the original version of SKYHI operated on a single latitude at a time. Arrays in memory were

only as large as a single longitude-vertical slab, with a few arrays containing adjacent latitude slices for north-south boundary information. Using this strategy, each array incurs a relatively small memory cost and there is little incentive to reduce the number of arrays allocated, particularly if such a reduction reduces performance or requires significant changes in model routines. Distributed-memory systems require all latitudes simultaneously and the amount of memory required quickly became prohibitive.

Several steps were taken to reduce the amount of memory used by SKYHI. First, arrays statically allocated in named common were reorganized so that they more closely followed the model structure. Second, variables that did not need to be statically allocated were dynamically allocated within relevent subroutines. Third, algorithms that required large arrays were redesigned so only a fraction of the memory is required. Fourth, many quantities were re-computed rather than stored. Despite these changes we remain memory-limited and are continuing to restructure routines such as longwave radiation that currently require a large number of dynamic arrays.

3.8 I/O

While some vendors have provided efficient parallel I/O using standard Fortran, many vendors still do not have simple I/O mechanisms. We have therefore adopted the strategy of Section 3.5, and provided machine-dependent routines with standard interfaces for reading and writing arrays. The appropriate routines for a given machine are chosen at compile time.

SKYHI writes restart files and several different types of data files. The restart files contain the prognostic variables and surface fields for the two time levels necessary to restart the simulation from a previous run. Such information is typically read and written once per model run and typical file sizes are shown in Table 2. The frequency and quantity of data for other files depends on the experiment being run. Usually full snapshots of prognostic and diagnostic fields are written once or twice per simulated day. File sizes for this example are also shown in Table 2. For other experiments, more frequent (hourly) data is written, but such data is generally restricted to only a few latitudes to reduce the storage and I/O requirements.

Table 2
Typical SKYHI I/O file characteristics

File type	Resolution	Amount of data (Mbytes)	Frequency
restart	n30	31	end of run
restart	n90	279	end of run
analysis	n30	56	every 24 hrs
analysis	n90	505	every 12 hrs

In the original version of SKYHI, information to be read or written was stored in a buffer array. Following the latitude-decomposition strategy, this buffer contained only information from a single latitude. In the new version, I/O buffers follow the general decomposition strategy and contain information for a particular "chunk". A short header is written (or read) with each chunk containing information identifying which piece of the decomposition is contained in that chunk. Such a mechanism allows for better I/O performance in the data-parallel model where the entire domain is written at once. It also enables separate processors in a message-passing implementation to write local data to a file in any order. The file format allows for easy conversion to any decomposition.

4. Performance

The new version of SKYHI is currently running on several Cray Research, Inc. (CRI) parallel vector processors and a Thinking Machines Corporation (TMC) CM-5 massively parallel computer. Performance numbers for a CRI C90 and the TMC CM-5 are given in Table 3. All simulations were performed in 64-bit precision.

When the parallel version of SKYHI is run on a single processor of the C90, the run time is between ten and twenty percent slower than the original version due largely to the new general decomposition strategy. In the original version, the code stepped logically through the latitudes and fluxes computed for the northern boundary of one latitude could be saved for use on the southern boundary for the next latitude's computations. In the parallel version, each portion of the domain may be computed in any order by any processor with no knowledge of previous computations. In this case, some fluxes must be computed redundantly by two

Table 3
SKYHI performance in both CPU time per simulated day and Mflops for different machines and configurations

Resolution	Machine	No. of nodes	CPU time (sec/model day)	Mflops
n30	CRI C90	1	318	351
n30	CRI C90	4	94	1300
n30	CRI C90	8	52	2350
n30	CRI C90	16	32	3830
n30	TMC CM5	64	436	294
n30	TMC CM5	128	276	465
n30	TMC CM5	256	195	658
n90	CRI C90	1	9705	405
n90	CRI C90	4	3014	1540
n90	CRI C90	8	1591	2920
n90	CRI C90	16	888	5220
n90	TMC CM5	256	3400	1329
n90	TMC CM5	512	2388	1892

processors. If the parallel code is run on a single processor, this adds additional computational overhead as the single processor performs the redundant computations. For maximal single-processor performance, a C pre-processor directive has been added to preserve the order and save appropriate boundary information. Use of this pre-processor option eliminates most of the overhead due to the general data decomposition and results in the performance shown in Table 3.

Performance on the C90 reaches 5.2 gigaflops on sixteen processors, about 40 percent of machine peak performance. Table 5 shows the speedups achieved running SKYHI with a latitude decomposition on multiple processors of the C90. Speedup is here defined as the ratio of performance for the lowest processor configuration possible. Speedups of 10 can be achieved on 16 processors of a C90. If the parallel code is run without the single-processor optimization (thus running the identical code across processor configurations), the speedups reach 13 for 16 processors. For long runs (where initialization costs are amortized), SKYHI is approximately 99.9% parallel and Amdahl's law would predict a nearly linear speedup. Deviations from Amdahl's law here are due to load imbalances associated with the Fourier filter. Because the filtering is not done at all latitudes, some processors are doing significantly more work than others. As the number of latitudes per processor decreases (i.e. the number of processors increase or the problem size decreases), the load imbalance becomes more of a problem.

SKYHI performance on the CM-5 as shown in Table 3 reaches 1.9 Gflops on a 512-node partition, only four percent of machine peak performance. In addition, the performance scales poorly with processor number as shown in Table 5. Processor ratio for the CM – 5 refers to the number of processors used for a simulation divided by the minimum number of processors required to run the model (e.g. the N90 simulation only fits on 256 nodes or greater so only two numbers are available to show speedup). This poor performance is primarily due to the large memory required for the model (see Section 3.7). For the simulations shown here, the number of horizontal grid points (or vertical columns) allocated to each vector unit is only between 24 and 52. Such a small number of points results in a high surface-to-volume ratio and therefore a high ratio of communications to computation. In addition, operations on horizontal slices (where there is no vertical dimension to increase the operations count) are unable to amortize costs associated with vector-unit setup or the broadcast of instructions from the control processor. Experiments we have performed with test codes on the CM-5 indicate that the number of horizontal grid points per vector unit should be at least 128 to achieve linear speedup or a reasonable fraction of machine peak performance.

By reducing the number of three-dimensional arrays dynamically allocated by SKYHI, the model can be run on fewer nodes with more horizontal grid points per node. For horizontal dynamics calculations, such a reduction can be achieved using an explicit loop over the vertical dimension thus requiring only two-dimensional dynamic arrays. Unfortunately, this results in short vector lengths and poor performance when the code is run using latitude decomposition on CRI parallel vector processors. A better option of reducing the number arrays is to restructure SKYHI routines that currently require a large number of three-dimensional arrays.

Efforts at restructuring the worst offender (longwave radiation) have so far only reduced the number of arrays by ten percent and more drastic restructuring will be required.

The fraction of time spent in various sections of SKYHI is shown in Table 4 for runs at n30 resolution on a single processor C90 and a 64 node CM-5. The C90 results show that the most time is spent for routines with the highest number of floating point operations. The lone exception is the calculation of the geopotential gradient where an interpolation from sigma surfaces to pressure surfaces is performed.

The CM-5 results indicate a large fraction of time is spent in those routines where communications are required. All communications combined amount to 35–40% of the total CPU time. The general communication required for the Fourier polar filter requires 25% of the total CPU time at n30 resolution and 39% of the time at n90 resolution. The increased percentage at higher resolution is due to the relatively more efficient scaling of floating-point operations compared to the scaling of general communications. It is clear that eliminating the Fourier polar filter would greatly improve the performance on distributed-memory machines. This could be achieved either through the use of local filtering alternatives or through the use of alternative grids (e.g. reduced grids) which eliminate the need

Table 4
Fraction of time spent by package for an n30 resolution

Package	Percentage run time	
	CRI C90/1	TMC CM5/64
Dynamics		
Geopotenial gradient	13.6	10.5
Horizontal diffusion	12.9	10.0
Vertical diffusion	9.9	4.2
Horizontal advection	4.7	4.3
Vertical advection	2.5	0.9
Other	5.2	1.8
Radiative processes		
Longwave radiation	8.3	4.7
Shortwave radiation	7.0	3.8
Adjustment physics		
Moist convective adjustment	5.7	11.4
Dry convective adjustment	0.9	0.6
Other adjustments	3.6	2.2
Other		
Surface processes	2.7	1.4
Polar filter	8.3	25.3
I/O	6.1	7.1
Diagnostics	1.8	2.1
Miscellaneous	6.8	9.7
Total	100.0	100.0

Table 5
Speedup with processor number. For the C90, blanks indicate no simulations were run for that configuration. For the CM-5, only low processor ratios are possible due to memory limitations. Processor ratio is defined in Section 4

Machine	Resolution	Processor ratio			
		2	4	8	16
CRI C90	n30	—	3.4	6.1	9.9
CRI C90	n90	—	3.2	6.1	10.9
TMC CM5	n30	1.6	2.2	—	—
TMC CM5	n90	1.4	—	—	—

for filtering. Moist convection also requires significantly more time in the CM-5 version than is required for the C90. This routine contains many conditionals and the difference between the two versions illustrates the inefficient implementation of the F90 "where" construct compared to the F77 'if" as discussed in Section 3.3. Routines like vertical advection and vertical diffusion that contain only floating point operations and no communications are relatively faster on the CM-5 than on the C90. Note that I/O does not consume a significant percentage of time on either machine due to the high bandwidth on the C90 and transparent support for parallel I/O to a scalable disk array on the CM-5.

5. Testing and validation

Studies have shown that for some classes of systems, small differences in initial conditions and numerical accuracy of the computation can have a significant effect on the statistics of the final solution [11,16]. There are also systems that exhibit multi-decadal transient behavior [9,17]. It is not known whether the class of systems that encompass general circulation atmospheric models exhibit these behaviors. With constraints on the computational resources available, we can only validate and test the model for the periods of time SKYHI will be used to simulate. For SKYHI, these timescales are typically on the order of months or decades.

During testing, global diagnostic sums of a variety of quantities are used to assess the results. The purpose of these quantities is to adequately describe the state of the model atmosphere and the diagnostics should be sensitive to the state of the model atmosphere rather than being extremely sensitive to the numerical order of summation. Numerically sensitive diagnostics do provide information that is important for debugging purposes, but they can lead to misleading conclusions on the state of the atmosphere, particularly for longer simulations.

Validation of the new SKYHI model began with single-processor simulations on a CRI YMP located at the GFDL to test the reproducibility of results obtained with the original code. Using the same machine and software environment eliminated some potential differences during validation, but exact reproducibility remains impossible due to code modifications that change the order of operations.

Differences that result are initially of the order of round-off error; after a day's simulation, differences appear in the seventh or eighth significant figures of the most sensitive global integrals.

Verification of the correctness of the model on a different system is an important step that must be undertaken before the model can be used on that system. Differences in compilers, system libraries, internal number representations, and machine precision can all have a significant impact on the answers and make correctness extremely difficult to guarantee. This is particularly true of certain SKYHI diagnostic sums which involve differences of large numbers and are therefore an effective measure of round-off error. After a single time-step, differences between the YMP and CM-5 simulations in the more sensitive diagnostics were less than 10^{-12} and the largest of these were attributable to roundoff error. After one day, the differences grow to about one percent for the most sensitive diagnostics and for longer simulations only qualitative comparisons can be made.

A month-long simulation on both the YMP and CM-5 has been completed and the results are in excellent agreement with results obtained using the original version. The major difference between the control run and the new versions was attributable to a new saturation vapor pressure table and not to the changes we introduced to the model.

6. Conclusions

While work on SKYHI is still in progress, several conclusions can be drawn based on the work presented here. First, changes to the SKYHI model outlined above have demonstrated the ability to develop a parallel version of a general circulation atmospheric model within a unified source. The new model has been designed for shared and distributed memory machines that support different programming paradigms. The results suggest that while such an approach is possible, the best performance can only be achieved when the model has been optimized for that particular hardware and software.

Second, it is clear that extreme memory use is causing poor performance on the CM-5 because it results in fewer points allocated per processor and increases the ratio of communication to computation. This memory problem is due to a few routines which allocate a large number of three-dimensional arrays and is primarily an algorithmic design problem. However, the data-parallel paradigm exacerbates this problem with the inherent assumptions of conformable global-sized arrays.

Last, the global communications required for performing a Fourier polar filter inhibits the performance of SKYHI on distributed memory machines without very fast networks. The use of a Fourier filter gives the model the performance disadvantages of a spectral model (global communications), while giving the model none of the numerical advantages except for increasing the size of the time step. Either local filtering alternatives to Fourier polar filters or grid choices which require no filtering would be highly desirable.

The conclusions above stress the limitations we have encountered with the SKYHI model. We hope to address these limitations as SKYHI continues to

evolve. On the positive side, good progress has been made to produce a model which is much easier to use and runs adequately on parallel machines. We are currently running a simulation on the CM-5 with 40 vertical levels and an n150 (0.6°) horizontal resolution which can not currently be done with the machines available at the GFDL. This higher resolution simulation will capture more of the atmospheric gravity-wave spectrum and the results will be used to extend the work of Hayashi *et al.* [10] in examining the effects of gravity waves in deceleration of stratospheric and mesospheric zonal flows.

Acknowledgements

We wish to thank Bob Malone, Kah-Song Cho, Charles Goldberg, Jim Sanderson, Jim Schwarzmeier, Tony Meys and Koushik Ghosh for advice and assistance throughout this project. We gratefully acknowledge the Advanced Computing Laboratory of Los Alamos National Laboratory; much of this work was performed using computing resources located at this facility. This work was supported by the Department of Energy's Computer Hardware, Advanced Mathematics and Model Physics (CHAMMP) Program for Climate Studies and by NOAA's High Performance Computing and Communications Program.

References

[1] A. Arakawa and V.R. Lamb, Computational design of the basic dynamical processes of the UCLA general circulation model, *Methods Comput. Phys.* 17 (1977) 173–265.

[2] W. Bourke, B. McAvaney, K. Pur, and R. Thurling, Global modeling of atmospheric flow by spectral methods, *Methods Comput. Phys.* 17 (1977) 267–323.

[3] S.B. Fels, J.D. Mahlman, M.D. Schwarzkopf and R.W. Sinclair, Stratospheric sensitivity to perturbations in ozone and carbon dioxide: radiative and dynamical response, *J. Atmos. Sci.* 37 (1980) 2265–2297.

[4] S.B. Fels and M.D. Schwarzkopf, An efficient accurate algorithm for calculating CO_2 15μm band cooling rates, *J. Geophys. Res.* 86 (1981) 1205–1232.

[5] S.B. Fels and M.D. Schwarzkopf, The simplified exchange approximation: a method for radiative transfer calculations, *J. Atmos. Sci.* 32 (1975) 1475–1488.

[6] I.T. Foster and P.H. Worley, Parallel algorithms for the spectral transform method, Oak Ridge National Laboratory Technical Report TM-12507, 1994.

[7] K.P. Hamilton, R.J. Wilson, J.D. Mahlman, and L.J. Umscheid, Climatology of the SKYHI troposphere-stratosphere-mesosphere general circulation model, *J. Atmos. Sci* 52 (1995) 5–43.

[8] S.W. Hammond, R.D. Loft, J.M. Dennis and R.K. Sato, A data parallel implementation of the NCAR community climate model (CCM2), in *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, in press.

[9] A. Hastings and K. Higgins, Persistence of transients in spatially structured ecological models, *Science*, 263 (1994) 1133–1136.

[10] Y. Hayashi, D.G. Golder, J.D. Mahlman and S. Miyahara, The effect of horizontal resolution on gravity waves simulated by the GFDL SKYHI general circulation model, *Pure Appl. Geophys.* 130 (1989) 421–443.

[11] J.F. Heagy, T.L. Carroll, and L.M. Pecora, Experimental and numerical evidence for riddled basins in coupled chaotic systems, *Phys. Rev. Lett* 73 (1994) 3528–3531.

[12] J.L. Holloway and S. Manabe, Simulation of climate by a global general circulation model I. Hydrologic cycle and heat balance, *Mon. Weather Rev.* 99 (1971) 335–370.

[13] J.L. Holloway, M.J. Spelman and S. Manabe, Latitude-longitude grid suitable for numerical time integration of a global atmospheric model, *Mon. Weather Rev.* 101 (1973) 69–78.

[14] Y. Kurihara and J.L. Holloway, Numerical integration of a nine-level global primitive equations model formulated by the box method, *Mon. Weather Rev.* 95 (1967) 509–530.

[15] A.A. Lacis and J.E. Hansen, A parameterization for the absorption of solar radiation in the Earth's atmosphere, *J. Atmos. Sci* 31 (1974) 118–133.

[16] Y.C. Lai, C. Grebogi, and E.J. Kostelich, Extreme final-state sensitivity in inhomogenous spatiotemporal chaotic systems, *Physics Letters A* 196 (1994) 206–212.

[17] Y.C. Lai, Persistence of supertransients of spatiotemporal chaotic dynamical-systems in noisy environment, *Physics Letters A* 200 (1995) 418–422.