



An atmosphere-ocean model: Code optimization and application to El Niño

C.R. Mechoso, J.D. Farrara, L.A. Drummond, J. Spahr & J.-Y. Yu
*Department of Atmospheric Sciences,
University of California, Los Angeles, USA*

Abstract

The top of the hierarchy of numerical models used for research and prediction of climate variability (e.g., El Niño and anthropogenic climate change) is occupied by general circulation models of the coupled atmosphere-ocean system. These are large codes characterized by heterogeneous components and diverse algorithms. This paper presents methods to overcome two major obstacles to performance efficiency when the application runs in a multi-processor environment: balancing of the computing load among processors and data transfers among model components. The power of these methods is illustrated in the context of the model developed at the University of California, Los Angeles. An example of El Niño simulated by that model is presented.

1 Introduction

Interactions between the atmosphere and underlying oceans are responsible for El Niño and other important phenomena of the climate system. El Niño years, in which sea surface temperatures (SSTs) over a large part of the tropical Pacific can become 3-4°C warmer than average, bring flood to Peru and Ecuador as well as drought to northern Australia. Other rainfall anomalies develop in locations far away from the equatorial Pacific. The sign of those anomalies tends to reverse in La Niña years, when the tropical Pacific becomes colder than average. Phenomena with strong atmosphere-ocean components are not restricted to the Pacific. For example, SST anomalies in the tropical Atlantic have been associated with drought and flood in Northeast Brazil.

Intense research is being dedicated to finding the mechanisms at work for the generation of oceanic anomalies and their remote impacts, particularly over



continental areas. The principal tools for research on mechanisms and predictability, as well as for predictions, of the climate system are numerical models. Coupled atmosphere-ocean general circulation models (CGCMs) include interactions and feedbacks between different components of atmosphere and ocean circulations. CGCMs are currently used to study El Niño and the role of the oceans in moderating the greenhouse warming effect of carbon dioxide and other gases. Addition of global atmospheric and oceanic chemistry allows for consideration of a wide range of problems such as the formation of the Antarctic "ozone hole" and the impact of volcanic eruptions on the ozone layer and on climate. Further addition of model components down to the societal impacts scales will result in "Earth System Models". The extraordinary complexity of full Earth System Models will epitomize the challenge of Earth sciences to computer technology.

In this paper we examine the challenges posed by the efficient running of CGCMs and the procedures we have designed to confront those challenges. Special emphasis is given to methods aimed at overcoming problems in load balancing and data exchanges between model components. We also present an example of code performance in the context of the model developed at the University of California, Los Angeles (UCLA). Finally, we present a simulation by that model of El Niño.

2 Model and code description

CGCMs comprise atmospheric and oceanic general circulation models (AGCMs and OGCMs, respectively). An AGCM is a complex code representing many physical processes. One can generally recognize two components in the structure of AGCM codes: Physics and Dynamics. Physics produces the drivers for the atmospheric motions through "parameterizations," which are numerical algorithms that estimate the effect of physical processes such as the heating of the atmosphere by solar and infrared radiation, clouds, and energy inputs/losses through the interfaces with ocean and land. The Physics component of the AGCM is generally computed on a latitude-longitude-height grid. The major current parameterizations (i.e., radiation and convection) operate in vertical columns of the atmosphere. Very little communication is required between processors if a two-dimensional grid partition in the horizontal (longitude-latitude) is used and, therefore, the code is almost "embarrassingly" parallel.

The drivers of the flow estimated by Physics are transferred to Dynamics, which computes the evolution of the atmosphere based on the equations that govern fluid motion in a form suitable for the time and space scales of weather phenomena. The present paper focuses on models in which Dynamics is also computed on a latitude-longitude-height grid (i.e., 'grid-point models'). This requires the "smoothing" or "filtering" of certain fields in high latitudes. Generally, there are several timesteps of Dynamics for each timestep of Physics.

The OGCM is also a complex code that has two major components: Baroclinic, and Barotropic. The former determines the deviations from the vertical average of the velocity, temperature and salinity fields, while the latter



determines the vertically averaged distributions of those fields. Unlike AGCMs, which are global, the domain of the OGCM can be restricted to a single ocean basin or even to a part of it.

Coupling of the AGCM and OGCM is achieved through data exchanges at the atmosphere-ocean interface. The AGCM passes the time-averaged surface wind stress, heat and water fluxes to the OGCM, which returns the SST.

3 Load balancing

Load imbalances generated by computations related to the parameterizations that depend on time and space and by filtering operations largely determine the performance of an AGCM code. For the most part, these imbalances can be dealt with via load redistribution techniques that target the reduction of memory latency (local or remote to a node) while maximizing the use of available resources in the system. We next review the load balancing schemes that we have designed for AGCMs. The impact on code performance of the computation in some of its kernels (i.e., "single-node" optimization) will not be discussed.

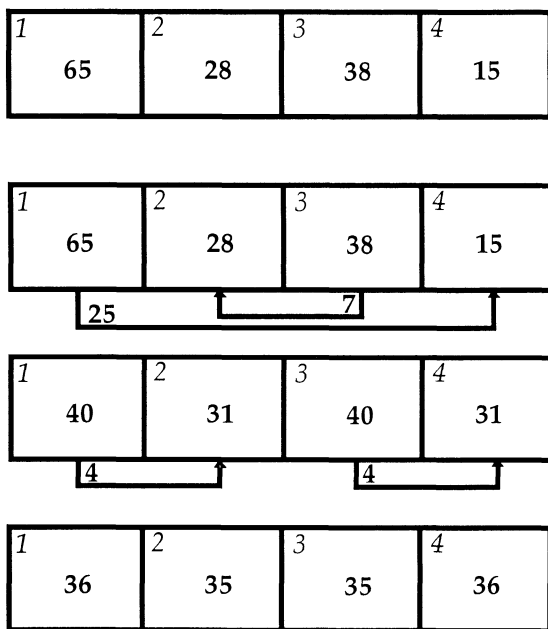


Figure 1: An example of AGCM/Physics load balancing involving four processors. Boldface numbers represent AGCM/Physics loads in seconds per simulated day of climate, italics numbers in the upper left corner of boxes represent processor number. Arrows represent data exchanges between processors.

3.1 AGCM/Physics

Variations in time and space of the intensity of atmospheric convection and differences in extent between sunlit regions and those in darkness are primarily responsible for the load imbalances in the Physics. In view of the variability of this load imbalance, we have addressed it with the dynamic scheme of Lou and Farrara [1]. This scheme is based on pairwise exchanges of data between processors with very different predicted AGCM/Physics loads (see Fig. 1). The load predictions are obtained by periodically measuring the unbalanced AGCM/Physics load in each processor. As shown in the example in Fig. 1, it may be necessary to perform the pairwise data exchange more than once to obtain a satisfactory final load distribution.

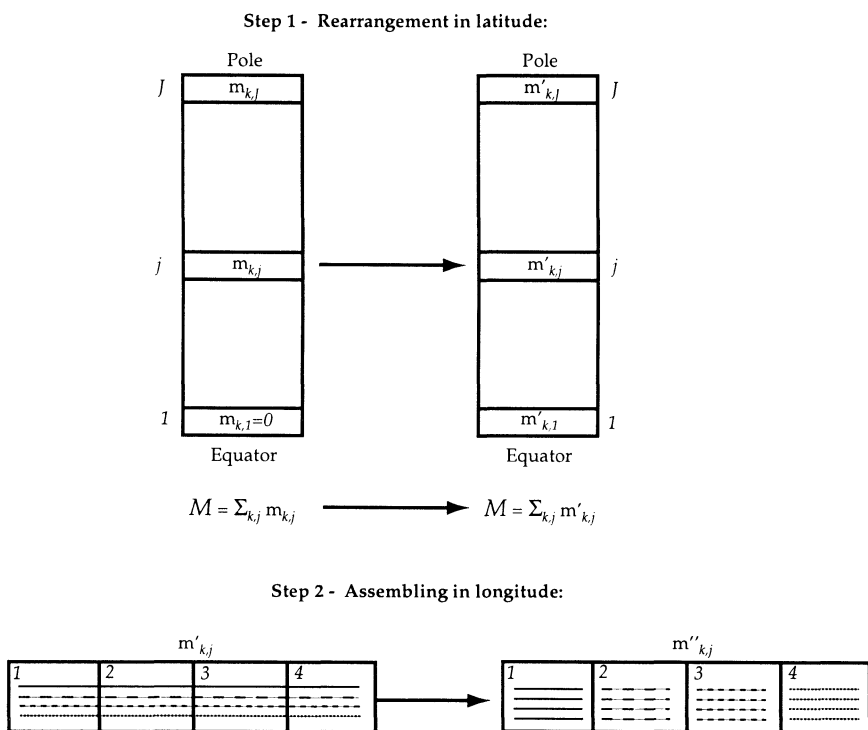


Figure 2: Schematic of the first two steps in the load balancing for AGCM/Dynamics of Lou and Farrara [1]. $m_{j,k}$ is the number of latitudes to be filtered of variable k in processor j . The solid and dashed lines in Step 2 represent complete longitude-height slices of the data. Prior to Step 2 these slices are distributed among all four processors; upon completion of Step 2, each resides in a single processor. Steps 3 and 4 (not shown) are inversions of Steps 2 and 1, respectively.



3.2 AGCM/Dynamics

The smoothing of fields in high-latitudes is performed via a Fourier filter, resulting in the primary load imbalance. The characteristics of this particular load imbalance do not change with time and are addressed by the static scheme of Lou and Farrara [1]. This is based on evenly distributing latitude bands of fields (covering all heights and a subset of longitudes) to be filtered among all processors. Such a distribution is made in four steps, of which the first two are schematically shown in Figure 2. The first step consists of a rearrangement in the latitudinal direction, and results in a more even distribution of filtering load in latitude. The second step consists of a rearrangement in the longitudinal direction, and results in each processor containing fields that are complete in longitude and ready to be filtered. The third step is an inversion of the second, and the fourth step an inversion of the first. Since the number of such latitude-height cross sections is necessarily limited whatever the model's resolution, the load redistribution using this scheme becomes poor when large numbers of processors are used. Therefore, we have modified this scheme to allow the breakup of latitude-height cross sections into individual latitudes of data. This results in a larger number of units of work to be redistributed, giving a better load balance on large numbers of processors.

3.3 OGCM

The primary source of load imbalance in the OGCM is the differing number of land points included in processor when the simplest longitude-latitude domain decomposition is used. This static load imbalance can be nearly eliminated by application of the re-partitioning scheme of Wang et al. [2], which removes most of the land points from the computational domain. For a North Atlantic version of POP use of this scheme resulted in a 33% savings in total execution time.

4 Coupling and the data broker

When run on a single node, the AGCM and the OGCM codes execute sequentially and exchange information corresponding to the air-sea interface. The special role of each model component allows for task parallelization. AGCM/Dynamics can run in parallel with OGCM/Barotropic, since these components do not exchange data. Further, AGCM/Physics can start as soon as OGCM/Baroclinic completes its calculation, since this includes the SST, and can run in parallel with this component of the OGCM. Figure 3 is a schematic of this running strategy, under the assumption that overheads due to distribution can be minimized and that model components running in parallel can be approximately balanced.

For running in multi-processor environments we have designed a system, referred to as the Data Broker, to move data between model components, the data archive, and visualization. A number of such systems exist but most are

specialized for particular applications and do not meet all of our requirements (see http://www.atmos.ucla.edu/~drummond/AGCM_infra/csw.html).

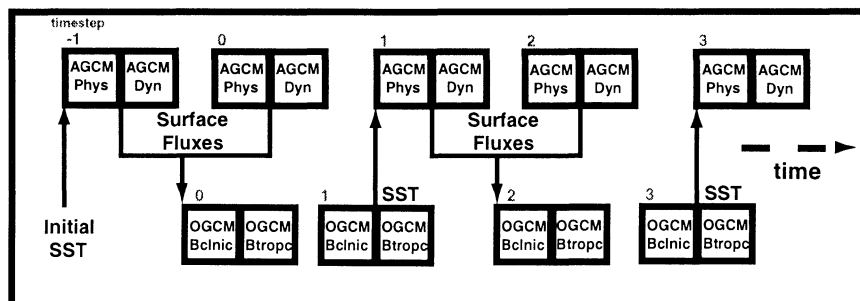


Figure 3: An example of task parallelization with the CGCM in which coupling performed every two steps of the AGCM/Physics.

In fact, we have designed and implemented two Data Brokers (Drummond and Mechoso [3]) The first, which is centralized (CDB), assembles an entire data field from a producer on a single processor before sending it (or parts of it) to consumers. This simplifies the logic, especially when the data field is divided in different subsets for the producer and consumer processors. The memory and bandwidth of a single processor limit the performance of the CDB, which makes the design not scalable in time or space. The second Data Broker we designed is distributed (DDB). There is still a centralized 'registration' facility that figures out which parts of a local data field on each producer processor are to be sent to which consumer processors. This permits parallel communication of data fields to occur.

The functionality of the DDB is encapsulated in a modular design that contains three libraries of routines that are built to work together and called on demand from different places inside the codes to be coupled: Communication Library (CL), Model Communication Library (MCL), and Data Translation Library (DTL). CL is the core library of routines that perform the point-to-point communication between computational nodes in a distributed environment. CL makes use of the functionality of widely used message passing software such as MPI.

The routines in MCL are provided to fulfill the two steps necessary to couple two or more independent codes using the DDB: a registration step and a communication step. First, the registration step, or initial code handshake, is provided for the code to exchange information regarding their coupling roles, and the production and consumption of the data to be exchanged. The registration step begins with the identification of a process or task as the Registration Broker (RB) with the instruction `MCLlamRegistrationBroker`. There is only one Registration Broker per coupled run and this role is only played during the registration step. The registration broker collects all information (data to be produced or consumed, type of data, grid representation,



etc.) from codes participating in the coupling. The RB processes the registration data collected to match producers against consumers at global or subdomain scales for sequential or distributed codes, respectively. The communication step is characterized by patterns of communication between the coupled components.

Both registration and communication steps are implemented in MCL through a set of C or Fortran callable routines that handle the initial registration step and a high level communication syntax. Therefore, a producer code that wants to send data to its consumers, simply executes a call to MCLSendData, which gets translated into several CL commands that in turn call the MPI or other message-passing software libraries to complete the communication step. Thus, the MCL-CL interface provides a level of transparency and code portability because the communication syntax used inside a program remains invariant when porting the code from one message-passing system to another.

DTL handles the data transformations from a producer's grid to the consumer's grid. The DTL routines are invoked by calls to the MCL that deliver data at the consumer end (i.e., MCLGetData). The DTL can include several numerical interpolation and extrapolation routines and the user can decide the transformation algorithm to be used according to his/her application. In this case, the calls to the MCL library remain the same but each of the low-level transformation routines in the DTL are overloaded with different procedures depending on the context. In view of our current coupling scenarios and requirements for data transformations, we have implemented a set of linear interpolation routines in a Linear-Interpolation-Library (LIL).

5 A particular application: The UCLA CGCM

The CGCM we use at UCLA consists of the AGCM developed at that institution and the OGCM based on the NASA Jet Propulsion Laboratory's implementation of the Parallel Ocean Program developed at Los Alamos National Laboratory. A detailed description of the models can be found in Mechoso et al. [4] and <http://www.atmos.ucla.edu/esm>. The computer codes of those model components have several tens of thousands of lines of FORTRAN with pre-processor directives to generate optimized code for various different computer platforms. The AGCM and OGCM are independent codes and can run in a purely distributed environment as long as it can be guaranteed that each model will be running at the same time and is available to be synchronized for coupling. In some current architectures (e.g., IBM SP-2, SGI/CRAY T3E), this is only possible when all codes are combined in a single executable. The component models periodically exchange data via the DDB.

The AGCM resolution currently used for climate studies is 2.5° longitude by 2° latitude and 29 layers, which is $29 \times 144 \times 89$ grid points. Lou and Farrara [1] showed that for this model version, application of their load-balancing scheme for Physics reduced the time spent in that component by 43% resulting in a 21% reduction in total execution time. They also showed that their revised load-balanced filtering reduced by 38% the time spent in that operation resulting in a 7% reduction in total execution time.

To give an example of the computational resource requirements of the CGCM, we consider the configuration consisting of the AGCM with two atmospheric chemical tracers (CFC-11 and CFC-12) coupled to the OGCM covering the Atlantic Ocean north of 30°S with a resolution of 1/6° long., 1/6° lat. and 37 layers. A curve indicating timings of this version on different numbers of T3E-600 nodes is shown in Fig. 4. To simulate several decades of climate within a reasonable time frame (one month), the model must simulate one day in ~60s. According to Fig. 2, this would require using ~768 T3E-600 nodes corresponding to a performance of ~37 GFLOPS. The memory requirement for this configuration is ~33 Gbytes, or a minimum of 64 Mbytes per node. In this case, the AGCM and OGCM exchange information at the air-sea interface once each simulated day; the AGCM sends 0.5 Mbytes/simulated day to the OGCM and the OGCM sends back 0.1 Mbytes/simulated day. Data transfers to archival storage in this case amount to 330 Mbytes/simulated day. The peak performance achieved so far with this configuration is about 50 GFLOPS on 512 nodes of the faster SGI/CRAY T3E-900.

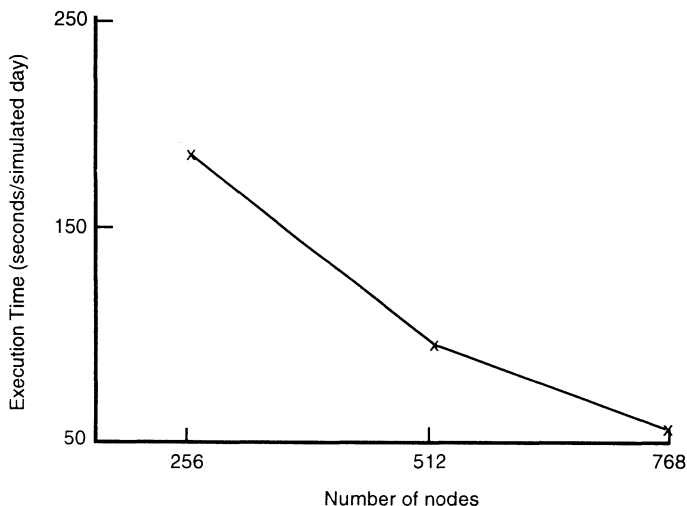


Figure 4: Timings (seconds/simulated day) of the CGCM using the DDB on the SGI/CRAY T3E-600.

6 El Niño simulation

Here, we present selected results from a simulation with the UCLA CGCM in which the OGCM domain covers the tropical Pacific and Indian Oceans (30°E – 60°W and 30°S – 50°N). For this simulation, the AGCM resolution is 5° longitude by 4° latitude with 15 layers and the OGCM resolution is 1° longitude by 1/3° latitude within 10° of the equator, increasing to 1° latitude in the extratropics, with 27 layers. The warm events obtained in the tropical Pacific

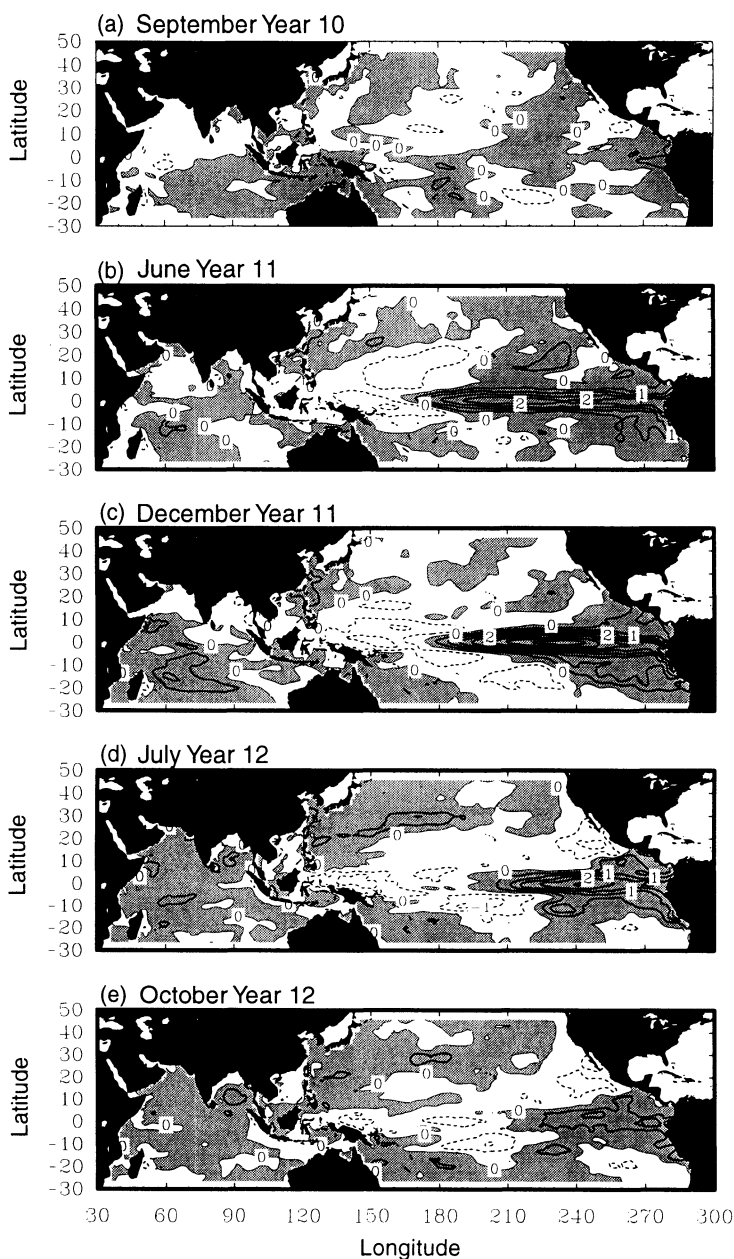


Figure 5: Evolution of the monthly-mean sea surface temperature anomalies ($^{\circ}\text{C}$) during an ENSO warm event simulated by tropical Pacific version of the UCLA CGCM. (a) September Year 10, (b) June Year 11, (c) December Year 11, (d) July Year 12 and (e) October Year 12. Positive values are shaded.



strongly resemble the El Niño events that unfold quasi-periodically in that ocean. As an example, we show in Figure 5, the spatial structure and evolution of SST anomalies during the simulated warm event beginning at the end of year 10. The warming of equatorial SST begins in the far eastern Pacific during the latter half of year 11. The warming continues and spreads westward until the entire central and eastern Pacific are dominated by strong positive SST anomalies. The maximum SST anomalies of 4°C occur at the end of year 11. Thereafter, positive SST anomalies weaken and retreat eastward during year 12. The strength and meridional extent of these simulated warm anomalies are reminiscent of those during strong El Niños, such as that of 1997-98. A more detailed analysis of this simulation is presented in Yu [5].

7 Conclusions

CGCMs are fundamental tools for climate studies. We believe that their role will become even more important as technological advances make them faster and friendlier to the users, and as societal demands for more accurate and detailed predictions increase even further. The models are making rapid progress towards a successful simulation of the coupled atmosphere-ocean system. Such progress requires the collaboration of meteorologists, oceanographers, and computer scientists in developing improved models, both scientifically and computationally. Their efforts are amply justified by the gains to be achieved by more successful climate predictions.

Acknowledgements. This research was funded by the NASA High Performance Computing and Communications for Earth and Space Sciences (HPCC-ESS) CAN 21425/041, by NOAA GOALS under Grant NA96GP0370, and DOE/NCPP under Grant DE-DG03-98ER62597.

8 References

- [1] Lou, J. Z. & J. D. Farrara, 1998: Performance analysis and optimization on the UCLA parallel atmospheric general circulation model code. *Concurrency: Practice and Experience*, **10(0)**, 1-17.
- [2] Wang, P., D. S. Katz & Y. Chao, 1997: Optimization of a parallel ocean general circulation model. *Proceedings, Supercomputing '97*, November 1997, San Jose, CA, IEEE Computer Society.
- [3] Drummond, L. A. & C. R. Mechoso, 2000: A Data Broker for distributed computing environments. In preparation.
- [4] Mechoso, C. R., J.-Y. Yu & A. Arakawa, 2000: A coupled GCM pilgrimage: From climate catastrophe to ENSO simulations. *General Circulation Model Development: Past, present and future. Proceedings of a Symposium in Honor of Prof. Akio Arakawa*. D. A. Randall. Ed., Academic Press. In press.
- [5] Yu, J.-Y., 2000: Climate variability over the Indo-Pacific region and the role of the Indian Ocean. *Proceedings of the Intl. Conf. on Environmental Modeling and Simulation*. Jan. 23-27, San Diego, CA, 17-22.