



PERGAMON

Atmospheric Environment 33 (1999) 675–681

**ATMOSPHERIC
ENVIRONMENT**

Technical Note

On the parallelization of a global climate-chemistry modeling system

Pius C.S. Lee^{a,*}, Rahul A. Zaveri^b, Richard C. Easter^b, Leonard K. Peters^c

^a Center for Atmospheric Physics, Science Applications International Corporation, 1710 Goodridge Drive, MS 2-3-1, McLean, VA 22102, USA

^b Atmospheric Sciences Group, MSIN K9-30, Pacific Northwest National Laboratory, Richland, WA 99352, USA

^c Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0211 USA

Received 24 June 1997; accepted 29 July 1998

Abstract

Coupled climate-chemistry simulations are computationally intensive owing to the spatial and temporal scope of the problem. In global chemistry models, the time integrations encountered in the chemistry and aerosol modules usually comprise the major CPU consumption. Parallelization of these segments of the code can contribute to multifold CPU speed-ups with minimal modification of the original serial code. This technical note presents a single program-multiple data (SPMD) strategy applied to the time-split chemistry modules of a coupled climate – global tropospheric chemistry model. Latitudinal domain decomposition is adopted along with a dynamic load-balancing technique that uses the previous time-step's load/latitude estimates for distributing the latitude bands amongst the processors. The coupled model is manually parallelized using the Message Passing Interface standard (MPI) on a distributed memory platform (IBM-SP2). Load-balancing efficiencies and the associated MPI overheads are discussed. Overall speed-ups and efficiencies are also calculated for a series of runs employing up to eight processors. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Atmospheric chemistry modeling; Domain-decomposition; Load-balancing; Message Passing Interface

1. Introduction

The complexity and nonlinearity of the physical and chemical processes occurring over a wide range of spatial and temporal scales make global-scale numerical modeling of the troposphere a challenging venture. Climate forcing by anthropogenic aerosols has become an important area of scientific investigation in the recent

years (Charlson et al., 1992). Computational speed of the coupled climate-chemistry models becomes a critical factor as long simulation periods, from several model months to a few years, are necessary to obtain reliable model statistics. Depending on the complexity of the model and the spatial resolution of the grid, impractically long turnaround times may result, even on the leading high-performance computers available today. The use of parallel computing is one way to decrease the turnaround times.

This technical note discusses parallelization of a coupled climate-chemistry model on a distributed

*Corresponding author. Tel.: 001 703 749 8778; fax: 001703 821 1134; e-mail: pius.lee@apo.saic.com.

memory platform of IBM Scalable Parallel Processor (SP2). Since the majority of the computational time in a climate-chemistry simulation is spent on chemistry integrations and aerosol dynamics calculations, we have focused our efforts on parallelizing only these modules. This approach is relatively less complicated and requires fewer modifications to the existing serial code than parallelizing the entire coupled climate-chemistry model.

The parallelization technique presented here is portable to most distributed memory platforms. The de facto internodal communication standard, Message Passing Interface Standard (MPI) (Message Passing Forum, 1994), is used to assure machine portability. The focus of this note is to evaluate the speed-up achieved by this partial modular parallelization scheme, as well as the message passing overhead on the distributed memory platform of the SP2, the latter being machine specific.

Section 2 describes the coupled global climate-chemistry modeling system. Section 3 discusses the philosophy and the steps for parallelizing the coupled CCM2-GChM code on a distributed memory platform. In Section 4, we evaluate the speed-ups and efficiencies achieved by the technique against the asymptotic optimal values according to Amdahl's Law.

2. Climate-chemistry model description

2.1. The coupled model

Atmospheric aerosols are formed as a result of sulfur-containing fossil fuel combustion, biomass burning, and degradation of a variety of naturally emitted trace gases, or are directly emitted as dust or sea-salt particles. These aerosols influence the radiation balance of our planet, and thus, impact its climate with a net cooling effect. The radiative forcing by aerosols includes a direct forcing involving scattering and absorption of sunlight by aerosol particles, and an indirect forcing in which aerosols acting as cloud condensation nuclei affect the properties and occurrence of clouds thereby changing the cloud radiative forcing. These forcing effects can be evaluated in a global climate model that uses aerosol concentration and composition information provided by a global chemistry model, which in turn is driven by the meteorological fields (winds, clouds, precipitation, etc.) predicted by the climate model.

Our coupled model consists of a global climate model — the Pacific Northwest National Laboratory (PNNL) version of the NCAR Community Climate Model (CCM2) (Hack, 1994; Kiehl et al., 1994; Ghan et al., 1997a,b), and a Global Chemistry Model (GChM) (Ghan et al., 1997). Both are elaborate physically based models. The CCM2 and GChM have been coupled in-core, allowing two-way transfer of information between

the two models. CCM2 supplies GChM with wind vectors, cloud fields, relative humidity, temperature and other meteorological data, while GChM carries out transport and chemical transformation of the trace chemical species and aerosols, and supplies CCM2 with the updated aerosol mass and number concentration data.

GChM's aerosol module currently treats four aerosol size modes via a Modal Aerosol Dynamics approach that keeps track of the number and mass distributions of each mode. The aerosol module is driven by primary emissions and aerosol precursors predicted by GChM's tropospheric chemistry module. Gas-phase chemistry treats anthropogenic hydrocarbons, biogenic species such as isoprene and monoterpenes, and dimethylsulfide of marine origin. The degradation mechanism used in this study is based on an earlier version of the more comprehensive set given in Zaveri (1997), consisting of 125 chemical reactions and 60 species. The aqueous-phase chemistry was limited to the reactions of SO₂ with H₂O₂ and O₃.

The chemistry module must integrate a large system of stiff, coupled ordinary differential equations describing the evolution of species concentrations due to chemical reactions. Such calculations are very expensive and tend to dominate the total CPU time consumption. The chemistry and aerosol dynamics calculations performed in GChM require almost 90% of the total CPU time for the coupled CCM2-GChM simulation.

2.2. Numerics of the GChM time integration

The GChM code solves the following transport, transformation and deposition differential equation for each model trace gas and aerosol species

$$\frac{\partial C}{\partial t} = -u \frac{\partial C}{\partial x} - v \frac{\partial C}{\partial y} - w \frac{\partial C}{\partial z} + \frac{\partial}{\partial z} \left(K_z \frac{\partial C}{\partial z} \right) + R + S + G \quad (1)$$

where C is the mixing ratio of a species; u , v , and w are the wind vectors; K_z is the vertical eddy diffusivity; R is a source and sink term describing chemical reactions; S is a source and sink term for emissions and removal; and G is a source and sink term describing interphase exchanges. The coupling between gas, aqueous, and aerosol species occurs via R and G terms.

The Eulerian model is constructed with the longitude, latitude and normalized pressure coordinate system, but x , y , and z are used here for simplicity. The R15 resolution (7.5° longitude × 4.5° latitude) used in this study has a system of 48 × 40 × 24 = 46080 grid points. At this resolution, the physical horizontal diffusion is generally smaller than the numerical diffusion associated with the advection algorithm. Horizontal diffusion terms are, therefore, neglected in Eq. (1).

A locally one-dimensional, time-splitting solution approach is used. It is a methodology generally used by atmospheric chemistry modelers (Carmichael et al., 1986; Kitada and Lee, 1993). After the time-split operator is applied to Eq. (1), one obtains an equivalent equation system (Eqs. (2)–(5))

$$\frac{\partial C}{\partial t} = -u \frac{\partial C}{\partial x} \quad (2)$$

$$\frac{\partial C}{\partial t} = -v \frac{\partial C}{\partial y} \quad (3)$$

$$\frac{\partial C}{\partial t} = -w \frac{\partial C}{\partial z} + \frac{\partial}{\partial z} \left[K_z \frac{\partial C}{\partial z} \right] + S \quad (4)$$

$$\frac{\partial C}{\partial t} = R + G. \quad (5)$$

Scalar transport Eqs. (2)–(4) are solved using the Modified Bott's Advection Form algorithm, based on finite-difference approximation (Easter, 1993). The Livermore Solver for Ordinary Differential Equations with general Sparse Jacobian matrices (LSODES) is chosen to integrate the system of non-linear ordinary differential equations describing the dynamic chemistry (Hindmarsh, 1983; Saylor and Ford, 1995). The above set of time-split integrations are performed sequentially using a transport time step of 20 min.

3. Model parallelization

Depending on the computational platforms available, and on the desirable speed-ups, there exist several methodologies for parallelizing the coupled CCM2-GChM modeling system. Although parallelization of the entire coupled model is ideally preferred, it is relatively difficult to actually accomplish this feat. On the other hand, since within a transport time step the photochemical and aerosol dynamics calculations at each grid point are independent of the calculations at the other grid points, they are highly amenable to parallelization. Partial parallelization of these time-intensive computations can still give useful overall speed-ups for a modest amount of coding effort.

This section illustrates the above philosophy and the steps for parallelizing the coupled CCM2-GChM code on a distributed memory platform. The Virginia Tech Scalable Parallel Processor (SP2) used in this work is composed of fourteen nodes which are POWER2 architecture IBM RS/6000 processors connected by an ethernet and a low-latency, high bandwidth switching network. It has two wide nodes of 512 Mb RAM, four thin nodes of 256 Mb RAM, and eight thin nodes of 128 Mb RAM.

3.1. Domain decomposition and load balancing

One of the major concerns while parallelizing a code is uniform distribution of the computational load among the processors (e.g. Foster and Toonen, 1994; Elbern, 1997). This minimizes the processor idle time, and thereby assures optimal parallel performance. At the global scale, the CPU load for chemistry integrations tend to vary from grid to grid depending on the presence of clouds, air composition, and the time of the day. Although it is difficult to estimate the exact amount of CPU time required for chemistry calculations at any given grid point, it is generally known that chemistry integrations take much longer during the daytime hours than at nighttime. Latitudinal global decomposition is, therefore, a logical choice for parallel processing of the chemistry and aerosol calculations since each latitude band contains both daytime and nighttime grids which provides some inherent load balancing.

In this work we use a dynamic load balancing technique in which the 40 latitude bands are distributed amongst processors based on the CPU time/latitude band estimates from the previous time step. For the first time step the bands are evenly distributed. The underlying assumption of this technique is that the CPU times required by the latitude bands do not change drastically over consecutive time steps. Parallelizing an advection operator is considerably more complicated than parallelizing the chemistry calculations, and depends greatly on the numerical algorithm used. The time-split finite-difference algorithm employed in GChM can be readily parallelized in the east-west and vertical directions (*X*- and *Z*-transport) when latitudinal decomposition is adopted; however, the north-south advection (*Y*-transport) calculations are done in a serial fashion.

3.2. Parallelization using the MPI standard

For FORTRAN codes, one can choose the highly automated option of preprocessing the serial code with a high-performance FORTRAN (HPF) compiler to generate a parallelized code. For example, the recently released IBM XLHPF and the Portland Group pgHPF can be used to parallelize FORTRAN codes on distributed memory platforms. However, due to lack of physical insight of the problem, HPF's evaluations of code during parallelization tend to be conservative, and usually results in a less optimally parallelized code (Bergmark, 1996). On the other hand, manual technique allows better control and more opportunities for parallelization.

We adopted manual parallelization of our code using the Message Passing Interface Standard (MPI Forum, 1994) for communication of data between the SP2 nodes. MPI standard consists of a platform and language independent library of user-callable subroutines. Manually

parallelizing a serial code involves insertion of MPI library calls at various points in the serial code, and may even require some program redesigning.

A simplified schematic of the structure of the parallelized CCM2-GChM code for three processors is shown in Fig. 1. The CCM2 calculations are performed serially on the master node, and is followed by transfer of meteorological fields from CCM2 to GChM. GChM performs *Y*-transport serially, and broadcasts the meteorological and chemical fields to the slave nodes via MPI library calls. All three processors then perform *X*- and *Z*-transport, and the chemistry (including aerosol dynamics) calculations in parallel for the domain-decomposed latitude bands. Once all the nodes have completed their calculations, the slave nodes transfer the relevant chemical fields back to the master node in asynchronous fashion.

3.2.1. Latency and bandwidth costs

A unit latency cost is incurred every time a MPI communication is invoked, which is about 30 μs for MPI communication over the SP2's high performance switch. Thus, it is desirable to send or broadcast fewer messages, and is achieved via lumped message passing using a pointer. A bandwidth cost, which is proportional to the length of the message is also incurred. It is about 28 μs Kbyte⁻¹ on the user accessible SP2 nodes.

In our simulations, approximately 40 MPI calls are made every time step, which cost about 1200 μs . Band-

width cost, estimated for passing 60 (species) \times 46080 (grid points) \times 0.004 (Kbytes/species) \times 2 (passes) = 11059.2 Kbytes, is about $6.2 \times 10^5 \mu\text{s}$. Thus, the total MPI overhead per model time step is about 0.62 s.

4. Computational results

A maximum of eight 128 Mb SP2 nodes were available for this study on the Virginia Tech SP2. To test the performance of the partially parallelized CCM2-GChM model, we performed simulations with one to eight processors. For the purpose of this study, only 4 h simulations were performed; these short runs suffice to evaluate speed-up and efficiency of the partially parallelized model. In the single processor simulation, the CPU time required for the serial section of the code was recorded at $t_{\text{ser}} = 15.7$ min, while the parallelized section of the code required about $t_{\text{par}}(1) = 109.4$ min. Thus, the fractional amount of time spent on the parallelized portion of the code in is $p = 0.874$. As mentioned before, the serial portion of the code includes the CCM2 calculations, data transfer between CCM2 and GChM, *Y*-transport, and input/output processes, while the parallel portion consists of *X*- and *Z*-transport, and column-wise photochemistry and aerosol dynamic calculations.

Fig. 2a shows the cumulative CPU time spent on each latitude grid bands (*j*) by the parallelized segment of the

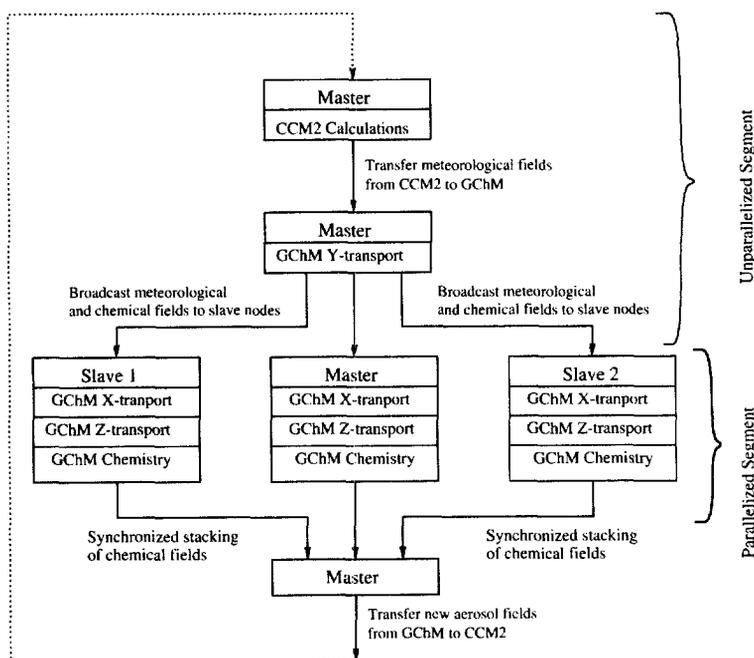


Fig. 1. Schematic flow chart of the parallelized CCM2-GChM code.

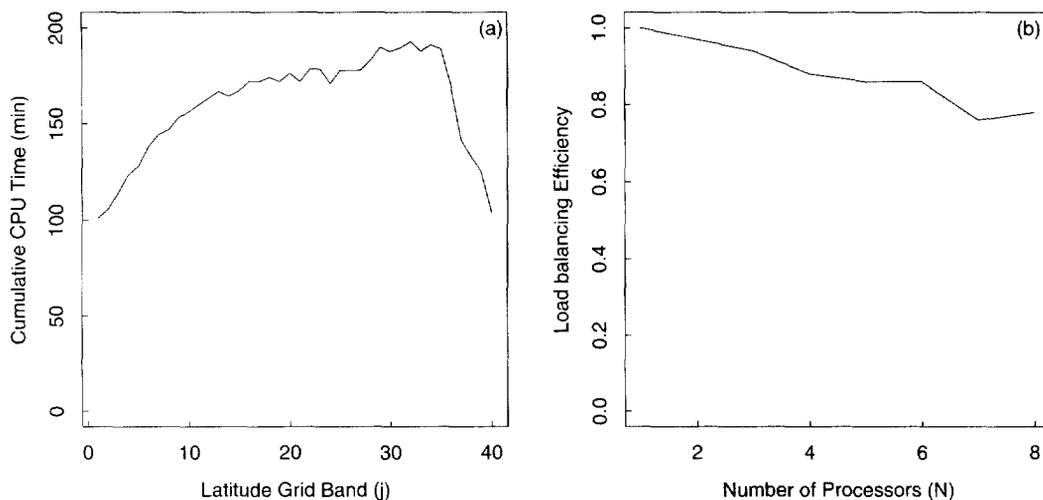


Fig. 2. (a) Cumulative latitudinal grid CPU load distribution for a 4 h simulation. (b) Dynamic load balancing efficiency as a function of number of parallel processors.

Table 1

Cumulative CPU times (min) for the parallelized segment (*X*-trans + *Z*-trans + Chem)

Total Nodes (<i>N</i>)	Individual node time								Time	Total $t_{\text{par}}(N)$	
	1	2	3	4	5	6	7	8			
1	109.4									109.4	109.4
2	53.7	56.2								109.9	56.2
3	35.5	36.1	38.6							110.2	38.6
4	25.3	26.6	27.9	31.1						110.9	31.1
5	20.1	20.9	21.5	22.7	25.4					110.6	25.4
6	16.8	16.6	18.2	18.4	19.5	21.3				110.8	21.3
7	14.2	13.5	14.6	15.2	15.3	16.5	20.6			109.9	20.6
8	11.9	13.2	12.2	12.3	14.4	14.3	14.9	17.5		110.7	17.5

code. It can be clearly seen that the chemistry calculations over the northern hemisphere ($j = 20\text{--}40$) require more CPU time than those over the southern hemisphere ($j = 1\text{--}20$). Also, note that the CPU loads near both the poles are significantly lower than the peak load in the northern hemisphere. Table 1 shows the cumulative CPU time spent by each node on the parallelized segment of the code employing dynamic load balancing technique described earlier, for all eight simulations. Also shown in Table 1 are the sum total of the CPU times required by each node and the actual CPU time required to execute the parallelized segment ($t_{\text{par}}(N)$), which is equal to the CPU time of the slowest node. The estimated cumulative MPI overheads for 12 model time steps is about 7.4 s, and is very small compared to the CPU time for integrations. Nearly constant total CPU times for executing the parallelized segment confirms that the MPI overheads

are negligible, and can be ignored in the performance calculations. However, it is clearly evident that computational load is not evenly distributed amongst various processors, and the load balancing efficiency can be calculated as

$$\eta_{\text{lb}} = \frac{1}{N} \times \frac{t_{\text{par}}(1)}{t_{\text{par}}(N)}. \quad (6)$$

Where $t_{\text{par}}(N)$ is the CPU time of the slowest node. The plot of load balancing efficiency against the number of processors in Fig. 2b indicates that load balancing gradually deteriorates as more processors are employed. This is due to the rather coarse “granularity” of the decomposition. For example, with eight processors, each processor gets an average of 5 latitude bands, and a “perfectly” balanced load is not possible.

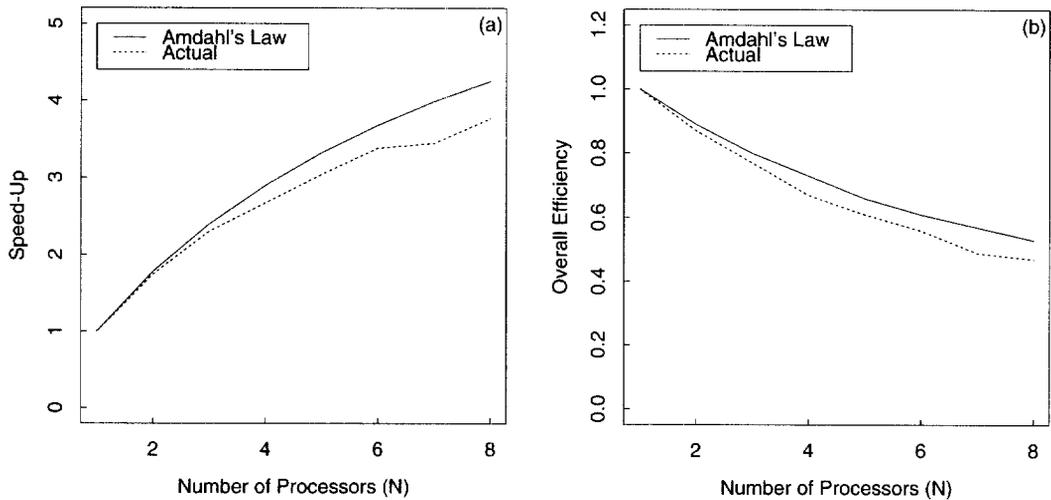


Fig. 3. (a) Overall speed-ups as a function of number of parallel processors. (b) Overall efficiencies as a function of number of parallel processors.

According to Amdahl's Law (Amdahl, 1967), the maximum speed-up, S_m , that can be achieved on a parallel processing machine with N processors is given by

$$S_m = \frac{1}{(1-p) + (p/N)} \quad (7)$$

According to the above equation, the maximum speed-up we can achieve with an infinite number of processors is about 7.9 for $p = 0.874$. However, the actual speed-up (S_a) realized on distributed memory platforms is lower than S_m due to data transfer (MPI) overheads and loss in efficiency due to non-uniform load distribution amongst processors. Since the MPI overheads are negligible, the actual overall speed-up, S_a , and overall efficiency, η_a , can be estimated by

$$S_a = \frac{t_{ser} + t_{par}(1)}{t_{ser} + t_{par}(N)} \quad (8)$$

$$= \frac{\eta_{lb}}{(1-p)\eta_{lb} + (p/N)} \quad (9)$$

$$\eta_a = \frac{S_a}{N} \quad (10)$$

Fig. 3a and b shows the plots of speed-ups and overall efficiencies as calculated by Amdahl's law and as actually estimated in each simulation against the number of processors employed. The disparity against the two is due to uneven load balancing. Nevertheless, quite good speed-ups and overall efficiencies are observed for our numerical experiments.

5. Conclusion

The coupled global chemistry model (GChM) and climate model (CCM2) was partially parallelized by applying latitudinal domain decomposition to the most computationally intensive sections of the chemistry model. The overall coupled model was about 87.4% parallelized in terms of CPU time consumption. The data communications were performed using the MPI standard according to a master-slave SPMD architecture, and the model was run on an IBM-SP2 distributed memory platform. Test runs were performed with one to eight processors.

A novel dynamic load-balancing technique was employed that distributes the load based on the previous time-step's CPU load per latitude estimates. This relatively straightforward approach yielded good load-balancing efficiencies as a function of number of processors, dropping to about 80% for eight processors due to the coarse granularity of the domain decomposition. An actual speed-up of 3.77 with an overall efficiency of 47% was observed for the same run. With four processors, an overall speed-up of 2.67 and efficiency of 67% were observed. The MPI overheads were found to be negligible (less than 0.1% of the total CPU time for the single processor run) in all simulations.

The parallelization techniques employed here could be easily extended to other air-quality models coupled with meteorological models. Although the performance of such partially parallelized codes will be limited by Amdahl's Law, useful speed-ups with reasonable efficiencies could be achieved for relatively modest code modification efforts.

Acknowledgements

This work was supported by the U.S. EPA Aerosol Program under contract CR822079-01-0 and by the NASA Aerosol Program under Grant NAGW 3367. Pacific Northwest National Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC06-76RLO 1830. Two of us (P.C.S.L and R.A.Z) are thankful for the helpful discussions with S.J. Ghan and R.D. Saylor throughout the progress of this work. The authors are also grateful to the two anonymous reviewers for their helpful comments and suggestions.

References

- Amdahl, G.M., 1967. Validity of the single processor approach to achieving large scale computer capabilities. In: Proceedings of the AFIPS Spring Joint Computer Conference vol. 30. AFIPS Press, Reston, VA, pp. 483–485.
- Bergmark, D., 1996. Personal communication. Workshop on Parallel Programming on the IBM RS/6000 SP, 28 April–2 May. Cornell Theory Center, Ithaca, New York, USA.
- Bogdonoff, P., 1989. Parallel FORTRAN: an overview and description of how to use parallel FORTRAN at the CNSF. Cornell National Supercomputing Facility. Ithaca, New York, USA.
- Carmichael, G.R., Peters, L.K. and Kitada, T., 1986. A second generation model for regional-scale transport/chemistry/deposition. *Atmospheric Environment* 20, 173–188.
- Easter, R.C., 1993. Two modified versions of Bott's positive-definite numerical advection scheme. *Monthly Weather Review* 121, 297–304.
- Elbern, H., 1997. Parallelization and load balancing of a comprehensive atmospheric chemistry transport model. *Atmospheric Environment* 31, 3,561–3,574.
- Foster, I., and Toonen, B., 1994. Load balancing algorithms for climate models. In: Proceedings of Scalable High Performance Computing Conference. IEEE Computer Science Press, pp. 674–681.
- Ghan, S.J., Chuang, C.C., Easter, R.C., Penner, J.E., 1995. A parameterization of cloud droplet nucleation. Part II: multiple aerosol types. *Atmospheric Research* 36, 39–54.
- Ghan, S.J., Easter, R.C., Zhang, Y., Leung, L.R., Saylor, R.D., Peters, L.K., Zaveri, R., Lee, P.C.S., 1997. Radiative forcing of climate by aerosols: a physically based modeling approach. Proceedings of the Third AMS Conference on Atmospheric Chemistry. Long Beach CA, February 1997.
- Hack, J.J., 1994. Parameterization of moist convection in the National Center for Atmospheric Research community climate model (CCM2). *Journal of Geophysical Research* 99, 5,551–5,568.
- Hindmarsh, A.C., 1983. ODEPACK: a systematized collection of ODE solvers. In: Stepleman, R.S. (Ed.), *Scientific Computing*. North-Holland Press, Amsterdam, pp. 55–64.
- Kiehl, J.T., Hack, J.J., Briegleb, B.P., 1994. The simulated Earth radiation budget of the National Center for Atmospheric Research community climate model CCM2 and comparison with the Earth Radiation Budget Experiment (ERBE). *Journal of Geophysical Research* 99, 20,815–20,827.
- Kitada, T., Lee, P.C.S., 1993. Numerical modeling of long range transport of acidic species in association with meso- β -convective clouds across the Japan Sea resulting in acid snow over coastal Japan. Part I: Model description and verification; and Part II: results and discussion. *Atmospheric Environment* 27A, 1,061–1,090.
- Levin, E., 1989. Grand challenges to computational science communication *ACM* 32, 1,456–1,457.
- Message Passing Forum, 1994. MPI: A message-Passing Interface Standard. compiled by the University of Tennessee. Knoxville, and Oak Ridge National Laboratory, pp. 1–224.
- Peters, L.K., Berkowitz, C.M., Carmichael, G.R., Easter R.C., Fairweather, G., Ghan, S.J., Hales, J.M., Leung, L.R., Pennell, W.R., Potra, F.A., Saylor, R.D., Tsang, T.T., 1995. The current state and future direction of Eulerian models in simulating the tropospheric chemistry and transport of trace species: a review. *Atmospheric Environment* 29A, 189–222.
- Saylor, R.D., Ford, G.D., 1995. On the comparison of numerical methods for the integration of kinetic equations in atmospheric chemistry and transport models. *Atmospheric Environment* 29A, 2,585–2,593.
- Saylor, R.D., Fernandes, R.I., 1993. On the parallelization of a comprehensive regional-scale air quality model. *Atmospheric Environment* 27A, 625–631.
- Zaveri, R.A., 1997. Development and evaluation of a comprehensive tropospheric chemistry model for regional and global applications. Ph.D. Dissertation. Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.