*Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.4, 883–897*

# SIMULATIONS OF GRAVITY WAVE INDUCED TURBULENCE USING 512 PE CRAY T3E[†]

Joseph M. PRUSA[*], Piotr K. SMOLARKIEWICZ[**],
Andrzej A. WYSZOGRODZKI[***]

A 3D nonhydrostatic, Navier-Stokes solver has been employed to simulate gravity wave induced turbulence at mesopause altitudes. This paper extends our earlier 2D study reported in the literature to three spatial dimensions while maintaining fine resolution required to capture essential physics of the wave breaking. The calculations were performed on the 512 processor Cray T3E machine at the National Energy Research Scientific Computing Center (NERSC) in Berkeley. The physical results of this study clearly demonstrate advantages of highly parallel technologies. We briefly outline the physical outcome of the study, as well as compare the relative model performance across several machines using both MPI and Shmem communication software.

**Keywords:** nonhydrostatic gravity wave turbulence, parallel Navier-Stokes solver

## 1. Introduction

In recent years, a number of new machines based on massively parallel processing (MPP) technology have become available for large-scale computations in science and engineering. Among the existing MPP computers, those consisting of hundreds or thousands of processors communicating via explicit message passing implementation of application programs appear particularly competitive with conventional vector supercomputers. On the other hand, there are a number of important yet sufficiently small problems that can be addressed successfully using vector supercomputers, single processor scalar workstations, or even modern PCs.

In order to best utilize the wide range of computing resources now available for science and engineering, application codes require a high degree of portability between different systems. To appreciate the significance of portability, consider that in the area of computational fluid dynamics, numerical research models usually solve

[†] The National Center for Atmospheric Research is sponsored by the National Science Foundation.

[*] Department of Mechanical Engineering, Iowa State University, Ames, IA 50010, U.S.A., e-mail: prusa@iastate.edu

[**] Division of Microscale and Mesoscale Mereorology, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80303, U.S.A., e-mail: smolar@ncar.ucar.edu

[***] Faculty of Physics, University of Warsaw, ul. Hoża 69, 06–681 Warsaw, Poland, e-mail: wyszog@lanl.gov

systems of nonlinear partial differential equations on discrete meshes consisting of millions of points over $\mathcal{O}(10^2) - \mathcal{O}(10^4)$ time steps (iterations). The associated computer programs consist of $\mathcal{O}(10^3) - \mathcal{O}(10^5)$ lines of code, and often evolve on a daily basis. Clearly, this makes supporting several versions of the same model cumbersome, expensive, and overall impractical. In this paper we emphasize the portability issue and report on our MPP model performance across several machines representative of the modern computing environment.

The MPP Fortran code adopted for the purpose of this study has already been described in the literature (Anderson and Smolarkiewicz 1997; Anderson *et al.*, 1997). The underlying solver is an incompressible-type fluid model cast in a curvilinear rotating framework, with a subgrid-scale turbulence parameterization and water substance phase-change processes included. The distinctive aspect of our model (Smolarkiewicz and Margolin, 1997) is its numerical design which incorporates a two-time-level; either semi-Lagrangian (Smolarkiewicz and Pudykiewicz, 1992) or Eulerian (Smolarkiewicz and Margolin, 1993), nonoscillatory forward-in-time (NFT) algorithm. The finite-difference approximations to the resulting trajectory-wise or point-wise integrals of the governing fluid equations are at least second-order-accurate. The Eulerian algorithm requires the traditional CFL stability condition, limiting thereby local communications to nearest neighboring points on the mesh; the semi-Lagrangian algorithm admits Courant numbers well exceeding unity and results in irregular communications patterns extending over a number of grid points. In order to take full advantage of MPP systems, the solver has been implemented using a single program multiple data (SPMD) message passing approach. In (Anderson and Smolarkiewicz, 1997), the authors evaluated the performance of the prototype dynamic core of the model (ideal Boussinesq fluid) for the two optional formulations of the model algorithm and two distinct parallelization approaches (High Performance Fortran, HPF, vs. message passing). In (Anderson *et al.*, 1997), this earlier study was extended to a more complete model (i.e., one including planetary rotation, phase change processes, and subgrid-scale turbulence schemes) suitable for simulating natural atmospheric flows. There, the authors quantified the overall performance of the complete model, as well as the relative performance of its distinct components (transport, elliptic pressure solver, phase-change modeling, subgrid-scale parameterization, etc.), on a distributed memory Cray T3D.

In this paper, we demonstrate a satisfactory performance of the model on a large scientific application, using one of the most complicated options of the model algorithm. As the application addressed is much too large to be executed straightforwardly on other machines available to us, the accompanying studies of the model performance exploit either an abbreviated version of this same experiment, or a less extreme physical scenario of large eddy simulation (LES) of convective planetary boundary layer using the default Eulerian variant of the model algorithm (cf. Smolarkiewicz and Margolin, 1998).

## 2. Model Description

The numerical model used in this study was described in (Anderson and Smolarkiewicz, 1997; Anderson *et al.*, 1997; Grabowski and Smolarkiewicz, 1996; Prusa

*et al.*, 1996; Smolarkiewicz and Margolin, 1997; 1998). It is a generalization of non-hydrostatic atmospheric models that solve the anelastic equations of motion in the standard, nonorthogonal terrain-following coordinates (Gal-Chen and Sommerville, 1975). Below we comment briefly on the essential aspects of the model design while referring the reader to earlier works for further details. A key analytical feature of this model is that it uses a time variable coordinate transformation to effect stretching of the domain of analysis. This allows the numerical algorithm to concentrate grid points in areas of interest and to change the shape of the domain as the solution develops. In the present study, this adaptive capability is used to force the gravity wave field with a deflection of the lower boundary streamline (see Section 4).

Generally, symbols with *overbars* indicate variables in the transformed coordinates. Symbols without overbars indicate variables in Cartesian coordinates. Vector quantities are denoted by boldface. The conservation laws for the dependent variables of the model may then all be written in the compact form

$$\frac{\partial \rho^* \psi}{\partial \overline{t}} + \overline{\nabla} \cdot (\rho^* \overline{\mathbf{v}}^* \psi) = \rho^* F(\mathbf{\Psi}). \tag{1}$$

Here $\overline{\nabla} \equiv (\partial/\partial\overline{x}, \partial/\partial\overline{y}, \partial/\partial\overline{z})$; $\psi$ denotes any of the three Cartesian velocity components $(u, v, w)$: the potential temperature, water substance mixing ratios (vapor, cloud water, rain, etc.), as well as turbulent kinetic energy; and $\rho^* \equiv \rho_o \overline{G}$ is the reference (Boussinesq type, cf. Section 4) density profile premultiplied by the Jacobian of the coordinate transformation (from the Cartesian to the terrain following, time-dependent, curvilinear framework). Note that the advective velocity appearing in (1) is the contravariant velocity in transformed coordinates, e.g., $\overline{\mathbf{v}}^* \equiv (\overline{u^*}, \overline{v^*}, \overline{w^*}) \equiv \mathrm{d}\overline{\mathbf{x}}/\mathrm{d}\overline{t}$. The corresponding anelastic approximation of the mass conservation law is (Lipps and Hemler, 1982)

$$\frac{\partial \rho^*}{\partial \overline{t}} + \overline{\nabla} \cdot (\rho^* \overline{\mathbf{v}^*}) = 0. \tag{2a}$$

The time derivative must be retained in (2a) because of the time variation in the coordinate transformation (Prusa *et al.*, 1996). This form of continuity is used to develop the flux form conservation law given in (1). The continuity equation is also used to construct an elliptic pressure equation. However, the time derivative in (2a) results in a source term that reduces the efficiency of the elliptic solver. Instead, we use the alternate, divergence free form

$$\overline{\nabla} \cdot (\rho^* \overline{\mathbf{v}^s}) = 0. \tag{2b}$$

where $\overline{\mathbf{v}^s} \equiv (\overline{u^s}, \overline{v^s}, \overline{w^s}) \equiv \overline{\mathbf{v}^*} - \partial\overline{\mathbf{x}}/\partial t$ denotes what we term the *solenoidal* velocity in the transformed coordinates. The solenoidal and contravariant velocity differ by the coordinate rate of stretching term. Proofs for the forms (2a) and (2b) are outlined in the Appendix.

The associated $F(\mathbf{\Psi})$ terms on the right-hand side of (1) are, in general, functionals of the vector $\mathbf{\Psi}$ of all dependent variables $\psi$, and they represent the sum of the resolved and subgrid-scale parts of the total forcings. In the momentum equations, the resolved terms include pressure gradient forces, Coriolis accelerations, buoyancy

force, as well as wave absorbing devices in the vicinity of open boundaries. In the thermodynamic equations, the resolved terms include heat and moisture sink/sources due to the phase changes of water, and the wave absorbers near the boundaries. Turbulence subgrid-scale (SGS) forcing terms are complicated but standard. We employ a turbulence model based on the prognostic turbulent kinetic energy equation (Schumann, 1991) or, optionally, its abbreviated version—the celebrated Smagorinsky model (Smagorinsky, 1993).

The integration of the discrete equations over a time step uses a regular unstaggered mesh. We write the finite-difference approximations to (1) in the compact form

$$\psi_{\mathbf{i}}^{n+1} = LE(\widetilde{\psi}) + 0.5\Delta t F_{\mathbf{i}}^{n+1}. \tag{3}$$

Here, $LE$ denotes either the advective semi-Lagrangian or flux-form Eulerian NFT transport operator; $\widetilde{\psi} \equiv \psi^n + 0.5\Delta t F^n$; indices $\mathbf{i}$ and $n$ have the usual meaning of the spatial and temporal location on a (logically) rectangular Cartesian mesh.

Completion of the model time step requires $F^{n+1}$ values of forcings in (3). Gravity wave absorbers, Coriolis accelerations, condensation, and pressure gradient forces are treated implicitly, whereas subgrid-scale terms and slow phase-change tendencies (such as rain formation or evaporation, see (Grabowski and Smolarkiewicz, 1996)) are treated explicitly (i.e., $F^{n+1}$ is predicted from earlier values of dependent variables). The implicitness of the pressure gradient forces is essential as it enables projecting the preliminary values $LE(\widetilde{\psi})$ onto solutions of the continuity equation (2b), cf. (Chorin, 1968). Here, it requires a straightforward algebraic inversion of the linear system composed of equations (3), and the formulation of the boundary-value problem for pressure implied by the continuity constraint (2b). The resulting elliptic equation is solved (subject to appropriate boundary conditions) using the generalized conjugate-residual approach—a preconditioned nonsymmetric Krylov solver, see (Smolarkiewicz and Margolin, 1994; 1997; Smolarkiewicz *et al.*, 1997) for further details. The numerical stability of computations is controlled by properly limiting Courant and Lipschitz numbers $\mathrm{C} = \|\Delta t \mathbf{V}/\Delta \mathbf{X}\|$ and $\mathrm{L} = \|\Delta t(\partial \mathbf{V}/\partial \mathbf{x})\|$, respectively, for the Eulerian and semi-Lagrangian variants of the model.

## 3. Parallelization versus Portability Strategy

In (Anderson and Smolarkiewicz, 1997), we evaluated the relative merits of message passing and HPF strategies of parallelization. Overall, we concluded that the message passing code runs 2.5 and 1.8 times faster (on Cray T3D) than the HPF code, respectively, for the Eulerian and semi-Lagrangian versions of the model. Consequently, we settled on a message passing approach. We used a two-dimensional horizontal decomposition of the grid, explicitly dimensioned each array to contain a subgrid of the total array plus extra space for a copy of the neighboring processors' boundary cells. This is a common technique (cf. Johnson *et al.*, 1994), where the extra boundary points are often referred to as "halo cells" or "halos". They are used to minimize communications needed when finite difference operations are performed. The number of halo cells depends on the local stencils used in the model algorithm and on the

maximum Courant number. In simulations reported here, $C \leq 3$. When necessary, the halo cell information is updated by having each processor exchange information with its neighbors. This communication process is further economized by admitting only a partial update of halos with their selected portions being exchanged between the processors as implied by the finite-difference algorithms employed. Reduction operations such as sums and extrema, unavoidable in fluid models, require exchanging information globally between all processors.

To exchange messages between processors, in general, we use the most portable and widely supported MPI (message-passing-interface) standard. However, on Crays T3D and T3E machines we optionally employ Shmem (shared-distributed memory data-passing support) library routines.

In order to facilitate the portability of the code, we use the same halo-update subprograms on the distributed or shared-memory parallel architectures, as well as on a single processor machines. On single-processor and shared-memory platforms, all updates are elementary. They employ one processor for the total domain dimension with halo used to set appropriate conditions at the domain boundaries. In this case, there is no need for an explicit message-passing protocol, and only selected parts of total arrays are rewritten to halo cells on the same processor.

In regards to the portability issue, input/output (I/O) operations raise some serious concerns. In general, outputed fields should be independent of the machine size and number of parallel processors used in simulations. Files written by programs running on $N$ processors should be readable to applications running on $M$ processors. This is convenient for debugging and is especially important for postprocessing (e.g., diagnostics, visualizations, etc.) of large computing projects. Furthermore, the output files must be also readable on different platforms with different binary file formats (Cray floating-point, Cray 64-bit IEEE, standard 32-bit IEEE, etc.). In our code, one processor performs all I/O communication operations by collecting and distributing arrays between other processors. Relative efficiencies of these I/O operations depend on the particular computer at hand and are important to the overall model performance. Keeping the total grid array on one processor does have the disadvantage of limiting the size of the application. But this is more than compensated by simplicity in coding, and seems to offer an optimal performance.

## 4. Physical Problem

The test problem for the Cray T3E simulated the evolution of an internal gravity wave packet generated by a narrow, 2D squall line at tropopause levels ($\sim 15$ km altitude) and its subsequent breaking near the mesopause ($\sim 85$ km altitude). We used a spatio-temporal Gaussian deflection of the lower domain boundary (streamline) as a proxy for the squall line disturbance. The transformed vertical coordinate was computed from $\overline{z} = H(z - z_s)/(H - z_s)$, where $z_s = A\ \exp[-((x - x_o + c_x t)/\sigma_x)^2]\ \exp[-((t - t_o)/\sigma_t)^2]$. The values of the parameters were: $A = 200$ m, $x_o = -30$ km, $c_x = 7\ ms^{-1}$, $\sigma_x = 2$ km, $t_o = 2$ h, and $\sigma_t = 1$ h. This vertical coordinate transformation clearly illustrates the time variable stretching capability that is built into the core of the model.

The basic state was one of uniform zonal wind ($u_o = -32\,\mathrm{ms}^{-1}$), stability (with Brunt-Väisällä frequency $N = 0.02\,\mathrm{s}^{-1}$), and density scale height (e.g., $\rho_o(z) = \exp(-z/H)$ with $H = 6.63\,\mathrm{km}$). This basic state and forcing initially favor the development of a 2D wavefield that is monochromatic, quasi-stationary, and has near unity aspect ratio. Each of these characteristics is due to linear wave dispersion. A few Brunt-Väisällä periods after maximum forcing, this wavefield undergoes a 2D primary (convective) instability. For a comprehensive description of the basic state, run set up, and results and analysis of the ensuing convective instability, see (Prusa *et al.*, 1996).

The problem is of interest for at least two reasons. First, the middle atmosphere is known to be far from radiative equilibrium at mesopause altitudes, and wave forcing is the main factor behind this phenomenon (Garcia and Solomon, 1985). Determination of the extent to which gravity wave breaking is responsible for this non-equilibrium has great relevance to a complete understanding of the process and its parameterization. Second, numerical simulation of turbulence is of considerable theoretical interest. The wavebreaking in this study generated a highly inhomogeneous, anisotropic turbulence. It was not initialized according to any *a priori* turbulence model nor constrained by the domain size (which can limit wave-wave interactions, see (Scinocca, 1995)). Instead, the turbulence developed from a very smooth linear wavefield in accordance with the physics of a wave packet propagating into a very deep model atmosphere (of over 16 density scale heights—corresponding to a density variation from the bottom to the top of over $10^7$ to one).

Some idea of the inhomogeneity of the wavefield can be gleaned from Fig. 1, which shows a contour density plot of the potential temperature ($\theta$) field. The vertical plane of this plot is perpendicular to the zonal flow. The wavefield is seen to be homogeneous in the spanwise direction (left to right) but inhomogeneous in the vertical (note that the complete altitude range is $15 \leq z \leq 125\,\mathrm{km}$; the regions above and below that shown in Fig. 1 are very smooth and characterized by constant stratification). Similar inhomogeneity occurs in the zonal direction (Prusa *et al.*, 1996; 1999).

The computational grid consisted of $544 \times 80 \times 291$ points with a resolution of 380 m. To save computer resources, the problem was executed in 2D on a $544 \times 1 \times 291$ grid until 120 minutes of the physical time. At 120 minutes, the 3D domain was created by repeating the solution in the spanwise direction $y$, and seeding the buoyancy field with a small amplitude (1% of the basic state) white noise. Further computations continued in five minute portions of physical time. The lateral zonal and spanwise boundaries were periodic with lateral zonal sponges. A specially tuned vertical sponge was also employed, such that it approximated the effects of atmospheric viscosity. The explicit sub-grid scale viscosity option was not employed in this simulation. Instead, energy removal at the grid scale was effected with the monotonicity option of the interpolator. This option invokes a topological constraint whereby no two streamtubes are allowed to intersect. Essentially energy is removed at the grid scale to the extent needed to avoid local negative entropy production. This corresponds well with the Kolmogorov microscale, which is of the same order of magnitude as the grid size at the initial altitude of breaking. The time chosen for 3D seeding was carefully selected based upon data generated with earlier 2D (Prusa *et al.*, 1996) and 3D (Prusa *et al.*,

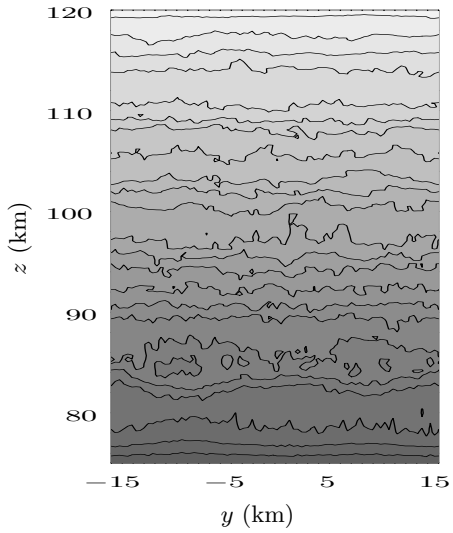


Fig. 1. Results from the wavebreaking experiment: contour density plot of $\ln(\theta)$ in vertical $yz$ (spanwise) plane at zonal location $x = -35\,\mathrm{km}$ at $t = 155$ minutes showing the region of vigorously breaking waves.

1997) experiments. The run was terminated at 180 minutes because at that point breaking had consumed the zonal extent of the domain.

The evolution of the turbulence was assessed by examining 1D and 2D energy spectra computed from $\theta$ fields (approximate equipartition occurs even during wave breaking, cf. (Prusa *et al.*, 1997)). The inhomogeneity of the wavefield caused severe windowing problems (Briggs and Henson, 1995) in the zonal (and vertical) spectra at intermediate to high wavenumbers. In particular, the localization of high wavenumber features (e.g., turbulence) to an approximately $\sim 50\,\mathrm{km}$ zonal ($\sim 30\,\mathrm{km}$ vertical) sub-domain caused (i) severe Gibb's oscillations, and (ii) excess power at the highest wavenumbers in spectra computed using the full extent of the computational domain. These high wavenumber problems did not occur, or were greatly diminished if the spectra were computed using sub-domains embedded in the region of turbulence. After considerable experimentation, we settled on fixed sub-domains that were approximately centered at 100 km altitude and $-45\,\mathrm{km}$ zonal location, and had zonal $\times$ vertical extents of $30 \times 20\,\mathrm{km}$. The vertical extent was smaller because the strong vertical stratification of the basic state limited the extent of the altitude spread of the turbulence. The spanwise extent of the sub-domains was $30\,\mathrm{km}$—the full spanwise extent of the computational domain. Although spectra based upon this sub-domain captured the correct high wavenumber behavior, they contained no information at scales larger than the sub-domain. To incorporate the lowest wavenumber power into the analysis, spectra were also computed using the full zonal and vertical extent of the computational domain.

All raw spectra were Hamming-Tukey smoothed, which acts to minimize the windowing effects of finite domain size (Bath, 1974). A differential correction algorithm

was developed to smoothly match the sub-domain $E_{\rm sd}(k)$ and full-domain $E_{\rm fd}(k)$ energy spectra into a single, corrected spectra $E_{\rm c}(k)$, according to

$$\frac{{\rm d}(\ln E_{\rm c})}{{\rm d}k} = \big(1-\beta(k')\big)\frac{{\rm d}(\ln E_{\rm fd})}{{\rm d}k} + \big(\beta(k')\big)\frac{{\rm d}(\ln E_{\rm sd})}{{\rm d}k} \equiv K(k,k'). \tag{4a}$$

Here $k'$ is a linear function of the wavenumber $k$ such that $k' \to 0$ and $k' \to 1$ as $k$ ranges from $k_{\rm min}$ to $k_{\rm max}$. The function $\beta(k')$ is a low order polynomial of $k'$ such that $\beta(0) = 0$; and $\beta = 1$ and ${\rm d}\beta/{\rm d}k' = 0$ at $k' = 1$. Integration of (4a) yields a finite difference equation for computing the corrected spectra

$$E_{\rm c}^{p+1} = E_{\rm c}^{p}\Big(1 + K\big(k^p, k'^p\big)\Delta k\Big). \tag{4b}$$

The initial condition for (4b) is ${E_{\rm c}}^0 = E_{\rm fd}(k_{\rm min})$. This algorithm corrects the full-domain spectra so that they show sub-domain spectral tendencies at the highest wavenumbers while simultaneously leaving the spectral tendencies at the lowest wavenumbers unaffected. The wavenumber domain for $k'$ was from $\ln(k_{\rm min}) = -1.0$ to $\ln(k_{\rm max}) = \ln(k_{\rm Nyq}) = 2.1$, where $k_{\rm Nyq} = 8.27\,{\rm km}^{-1}$ is the Nyquist wavenumber (and $k_{\rm min} = 0.37\,{\rm km}^{-1}$). The resulting zonal spectra shown in Fig. 2 depict both the highest (due to localized turbulence) and lowest (due to changes in the mean state of the atmosphere) wavenumber features with high fidelity.

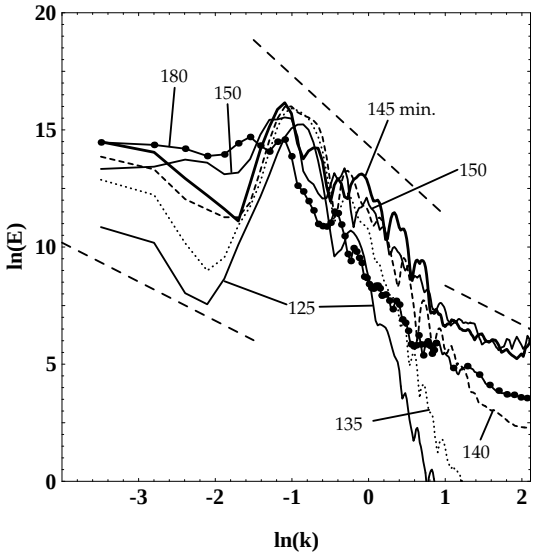


Fig. 2. Time evolution of zonal spectral energy from 125 to 180 minutes (straight dashed lines have slopes of $-5/3$ and $-3$).

Figure 2 shows the evolution of the zonal energy spectra. At 125 minutes, energy is concentrated at $k = 0.40\,{\rm km}^{-1}$ ($\ln(k) = -0.90$), corresponding to $\lambda_x = 15.5\,{\rm km}$. This fundamental is due to a linear growth of the gravity wave packet as it ascends.

At $k = 0.80\,\mathrm{km}^{-1}$ ($\ln(k) = -0.23$), a much weaker second harmonic of the fundamental can also be seen. This second harmonic is a harbinger of the primary convective instability that is about to occur. For the given basic state, linear waves have an evanescent limit at $k = 0.62\,\mathrm{km}^{-1}$, and this corresponds to the very sharp drop off between the fundamental and second harmonic. At later times, wave dispersion causes the fundamental to broaden and peak at lower wavenumbers (longer waves are slower and take more time to propagate upwards, see (Prusa *et al.*, 1996)). With the onset of vigorous wave overturning ($\sim$ 140 minutes), a buoyancy subrange (Weinstock, 1985) with a slope of $-3$ appears just upscale of the fundamental. Until this primary instability occurs, there is negligible energy at the highest wavenumbers. With the onset of a secondary (3D) instability, a tendency towards a $-5/3$ slope can be seen. The critical buoyancy wavenumber that separates the two regimes decreases from $k_b = 4.0$ to $1.8\,\mathrm{km}^{-1}$ as $t$ increases from 140 to 180 minutes, respectively. This compares favorably with earlier results on a Cray J90 (at 625 m resolution) which yielded $k_b = 2.1\,\mathrm{km}^{-1}$ at 150 minutes (Prusa *et al.*, 1997). The experimental value of $k_b$ may also be compared with the scaling result, $k_b \sim N^3/\epsilon_o \sim 1.6\,\mathrm{km}^{-1}$ (Weinstock, 1985), where $\epsilon_o$ is the turbulence dissipation rate. Finally, Fig. 2 shows another $-5/3$ power law regime at the lowest wavenumbers at earlier times (125 to 145 minutes). This is consistent with a 2D reversed energy cascade that is transferring energy into the zonal mean fields (Kraichnan, 1967). The Eliassen-Palm flux divergence has its maximum value precisely in this time interval, of order $0.02\,\mathrm{ms}^{-1}$, at breaking altitudes. After 150 minutes, the energy spectra flatten out at the lowest wavenumbers. At this point, wave breaking has disrupted the linear wave field sufficiently so that it lacks the large scale coherence needed to effectively modify the zonal average state. Vertical spectra (not presented) show very similar evolutionary tendencies, with the only significant difference being the lack of the $-5/3$ power law regime at the lowest wavenumbers. Spanwise energy spectra (not presented) show very different evolutionary tendencies, however. The spectrum at 125 minutes is quite flat and 15 orders of magnitude below the fundamental of the zonal spectra. Growth of spanwise energy is negligible for the first 10–15 minutes after 3D seeding. In the next 5–10 minutes, spanwise spectral energy explodes as the secondary instability undergoes a period of exponential growth. An inertial subrange, characterized by a $-5/3$ power law, appears at the highest wavenumbers. As $t$ continues to increase, this subrange expands to lower wavenumbers, until at 180 minutes, most of the spectrum lies within it.

## 5. Model Performance Results

The experiment described in the preceding section was performed on the 512 processor Cray T3E machine at NERSC. Table 1 outlines the history of its computational cost versus the overall model performance (measured by the elapsed wallclock time, or WCT) as functions of the simulated physical time, time step $\Delta t$, number of time steps $N_t$, and average number of iterations $N_{\mathrm{it}}$ in the elliptic Krylov solver (per timestep) per 5 minute portion of the experiment. In addition to summarizing elementary aspects of the model efficiency, this table illustrates the important point that the overall model performance (as well as the relative performance of various model

components such as advective transport, pressure solver, etc.; see (Anderson *et al.*, 1997) for a discussion) is a complicated function of the simulated flow. Consider, for instance, that at the onset of vigorous wave breaking at 145 minutes, accuracy arguments (Prusa *et al.*, 1996; Smolarkiewicz and Pudykiewicz, 1992) dictate halving the time step. Yet, as the flow becomes more quiescent following the onset of breaking, the elliptic solver converges (see (Smolarkiewicz *et al.*, 1997) for a discussion on the convergence criteria vs. the model algorithm) using only a third as many iterations per timestep.

Table 1. Gravity-wave breaking experiment on 512 PE Cray T3E. The history of the average number of iterations $N_{\rm it}$ in the elliptic pressure-solver per 5 minute portion of the experiment, wallclock time (WCT), and CPU time are given as functions of the simulated physical time, time step $\Delta t$, and number of timesteps $N_t$ per 5 minutes portion of the experiment. Only every second portion is shown.

| physical time | $\Delta t$ | $N_t$ | $N_{\rm it}$ | WCT | User CPU |
|---|---|---|---|---|---|
| 125–130 (minutes) | 5 (s) | 60 | 32 | 1156 (s) | 583159 (s) |
| 135–140 | 5 | 60 | 31 | 1123 | 566583 |
| 145–150 | 2.5 | 120 | 19 | 1500 | 757776 |
| 155–160 | 2.5 | 120 | 15 | 1379 | 696690 |
| 165–170 | 2.5 | 120 | 12 | 1278 | 645268 |
| 175–180 | 2.5 | 120 | 11 | 1212 | 611210 |

The gravity-wave-breaking experiment was much too large to be used as a benchmark for any systematic performance studies across various machines. Also, our limited resources at NERSC precluded any 'lavishness' and left little room for additional tests beyond the production runs. As a result, relative performance issues (e.g., performance vs. machine architecture and/or the number of processors) were addressed using either a 2D variant of the experiment, or our earlier LES calculations of convective boundary layers, using the Eulerian variant of the model with the nonoscillatory option of the MPDATA algorithm for the $LE$ operator in (3) (Anderson *et al.*, 1997; Smolarkiewicz and Margolin, 1998).

The results of the model-performance analysis are gathered in Tables 2–4. They further demonstrate that the overall model performance is not only a function of the flow but it also depends upon the machine, communication software, compiler options, model algorithms, and the size of the problem. Table 2 describes the machines used in this study, whereas Table 3 addresses the scalability issue exploiting a 2D variant of the gravity-wave experiment ($544 \times 1 \times 291$ mesh with $\Delta t = 5\,\mathrm{s}$ and $N_t = 1800$). These simulations were performed using 16 and 32 processors only. On the HP and T3D, the resulting speedups were quite good regardless of the communication software employed (consistent with our earlier experiences in (Anderson and Smolarkiewicz, 1997; Anderson *et al.*, 1997), but were relatively poor on the T3E-900 using communication software. Table 4 shows a similar speedup factor analysis based

upon a larger 3D computational problem—an LES boundary-layer type experiment. In this case, the T3E-900 gave a more respectable performance (with speedup factors closer to the $\sim 1.8$ value concluded in our earlier studies) and outperformed the HP when using a larger number of processors.

Table 2. Machines employed in model performance studies. Columns 2–5 show the number of processors, location of the machine, nominal memory, and typical measured performance (Mflops/PE), respectively.

| machine | # PE | location | memory | performance |
|---|---|---|---|---|
| T3E-900 | 512 | NERSC | 256 MB/PE | 150-300 (Mflops) |
| T3E-600 | 32 | ICM[2] | 128 MB/PE | 100–200 |
| T3D | 128 | NCAR | 64 MB/PE | 15 |
| HP[1] | 64 | NCAR | 8 GB | 120 |
| Cray J932se | 24 | NCAR | 8 GB | 60 |
| Cray J916 | 16 | NCAR | 2 GB | 60 |

Table 3. Scalability results using 2D semi-Lagrangian simulation of the gravity wave. The "O*i*" symbol in the first column refers to the compiler optimization level. The second column specifies the communication software employed. Columns 3 and 5 show the wallclock time (s) of the entire experiment, whereas the fourth column shows the resulting speedup ($\nearrow$).

| machine | comm. soft. | 16 PE | $\nearrow$ | 32 PE |
|---|---|---|---|---|
| HP O1 | MPI | 22354 (s) | 1.93 | 11571 (s) |
| HP O2 | MPI | 7192 | 1.91 | 3758 |
| T3D | Shmem | 9911 | 1.88 | 5266 |
| T3E-900 | Shmem | 4365 | 1.58 | 2768 |
| T3E-900 | MPI | 5325 | 1.50 | 3542 |

The relatively smaller speedup factor for smaller jobs using smaller numbers of processors on the T3E-900 is not necessarily surprising. Since the simulated turbulent flows are highly chaotic and unpredictable, the model algorithm can react even to such minor changes in the code setup as the number of processors or the compiler employed[3]. This sensitivity is insignificant insofar as the physical issues are concerned, but it can quite substantially affect model performance.

[1] Experiments performed in the double 64-bit precision, to match the Cray standard.

[2] Interdisciplinary Center for Mathematical and Computational Modeling, University of Warsaw, Poland.

[3] The different execution of sums inherent in elliptic Krylov solvers can affect both the evaluated pressure field and the number of the iterations required.

Table 4. Scalability results using 3D Eulerian simulations of convective boundary layer.

| machine | comm. soft. | 8 PE | ↗ | 16 PE | ↗ | 32 PE | ↗ | 64 PE |
|---|---|---|---|---|---|---|---|---|
| HP O1 | MPI | 24308 (s) | 1.89 | 12836 (s) | 1.50 | 8542 (s) | 1.60 | 5325 (s) |
| HP O2 | MPI | 8912 | 1.65 | 5395 | 1.63 | 3307 | 1.44 | 299 |
| HP O3 | MPI | | | 6516 | 1.64 | 3969 | 1.43 | 2777 |
| hline HP O4 | MPI | | | 6463 | 1.54 | 4179 | 1.46 | 2859 |
| T3D | Shmem | 16622 | 1.87 | 8866 | 1.61 | 5492 | 1.71 | 3164 |
| T3D[4] | Shmem | | | 15569 | 1.78 | 8751 | 1.79 | 4876 |
| T3E-900 | Shmem | | | 3199 | 1.43 | 2236 | 1.77 | 1260 |
| T3E-900 | MPI | | | 3555 | 1.43 | 2476 | 1.63 | 1516 |
| T3E-600 | Shmem | 9127 | 1.92 | 4735 | | | | |
| T3E-600 | MPI | 7471 | 1.87 | 3995 | | | | |
| J932se[5] | none | 14087 | | 8545 | | 4987[6] | | |
| J916[5] | none | 7696 | | 7039 | | | | |

Table 4 contains a number of hints useful for the interested practitioner. We draw attention to a few points that are especially noteworthy. The results from autotasking runs on Crays J932se and J916 depend on the actual state of the machine, so they should be viewed only as examples of possible overall performances. However, the wallclock times attained with the maximal number of processors are representative of our experiences running numerous simulations and models. The semi-Lagrangian run is about 50% more expensive than the Eulerian run at the same $\Delta t$ (here $C \lesssim 1$). However, in our breaking gravity wave problem, much larger time steps are used (with $C \lesssim 3$) and, what is *more important*, the semi-Lagrangian algorithm is more accurate as it treats equally the incompressible and compressible numerical regimes of flow, dictated by the specified time-dependency of the problem geometry.

## 6. Remarks

The horizontal grid decomposition employed for the message-passing MPP implementation of our model was dictated, in essence, by the physics of natural geophysical flows that makes the vertical (gravity) direction distinct. Coincidentally, it has a purely computational advantage of admitting efficient applications of the same model algorithm and code design on different types of machines including distributed- and memory-shared as well as single-processor architectures.

Although our MPP model has been designed to run efficiently on the distributed memory architectures, it appears to perform reasonably well on standard vector supercomputers. Consider that the original version of the same model, optimized for

[4] The equivalent semi-Lagrangian run included for comparison.

[5] J90 autotasking, shared (as opposed to dedicated) mode runs.

[6] 24 PE run.

the shared-memory Cray vector machines, achieves on average 65–90 Mflops per processor on J90s, depending on the number of processors and the application addressed, whereas the MPP code is only slightly slower on these machines with its speed falling in the range of 60–85 Mflops/PE.

Regardless of all the objective model-performance measures discussed in this paper, the single most important outcome of this study cannot be overstated: Our earlier, one order of magnitude smaller gravity-wave experiments performed in autotasking mode on the 24 processor Cray J90 at NCAR used to take several days (including wait time in economy queues) to accomplish a simulation of 5 minutes of physical time. Present experiments, on the 512 PE Cray T3E with an order of magnitude larger grid, were executed essentially overnight for the same 5 minutes period of simulated physical time!

## Acknowledgements

We thank William Anderson for his advice on parallelization issues and comments on an earlier version of the manuscript. The present work has been supported in part by the Department of Energy "Computer Hardware, Advanced Mathematics, Model Physics" (CHAMMP) research program, and by the National Science Foundation grant #ATM 9616811. The use of the 512 PE Cray T3E at NERSC and of the 32 PE Cray T3E at ICM is gratefully acknowledged.

## References

Anderson W.D. and Smolarkiewicz P.K. (1997): *A comparison of High Performance Fortran and message passing parallelization of a geophysical fluid model*, In: Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers (P. Schiano, A. Ecer, J. Periaux, N. Satofuka, Eds). — Amsterdam: Elsevier, pp.384–391.

Anderson W.D., Grubišić V. and Smolarkiewicz P.K. (1997): *Performance of a massively parallel 3D non-hydrostatic atmospheric fluid model.* — Proc. Int. Conf. *Parallel and Distributed Processing Techniques and Applications, PDPTA'97*, Las Vegas, pp.645–651.

Aris R. (1989): *Vectors, Tensors, and the Basic Equations of Fluid Mechanics.* — New York: Dover Publications.

Bath M. (1974): *Spectral Analysis in Geophysics, Developments in Solid Earth Geophysics.* — Amsterdam: Elsevier.

Briggs W.L. and Henson V.E. (1995): *The DFT.* — Philadelphia: Soc. Appl. Ind. Math.

Chorin A.J. (1968): *Numerical solution of the Navier-Stokes equations.* — Math. Comp., Vol.22, No.104, pp.742–762.

Gal-Chen T. and Somerville C.J. (1975): *On the use of a coordinate transformation for the solution of the Navier-Stokes equations.* — J. Comput. Phys., Vol.17, No.2, pp.209–228.

Garcia R.R. and Solomon S. (1985): *The effects of breaking gravity waves on the dynamics and chemical composition of the mesosphere and lower thermosphere.* — J. Geophys. Res., Vol.90, No.D2, pp.3850–3868.

Grabowski W.W. and Smolarkiewicz P.K. (1996): *On two-time level semi-Lagrangian modeling of precipitating clouds.* — Mon. Wea. Rev., Vol.124, No.3, pp.487–497.

Johnson K.W., Bauer J., Riccardi G.A., Droegemeier K.K. and Xue M. (1994): *Distributed processing of a regional prediction model.* — Mon. Wea. Rev., Vol.122, No.11, pp.2558–2572.

Kraichnan R.H. (1967): *Inertial ranges in two-dimensional turbulence.* — Phys. Fluids, Vol.10, No.7, pp.1417–1423.

Lipps F.B. and Hemler R.S. (1982): *A scale analysis of deep moist convection and some related numerical calculations.* — J. Atmos. Sci., Vol.39, No.10, pp.2192–2210.

Prusa J.M., Smolarkiewicz P.K. and Garcia R.R. (1996): *On the propagation and breaking at high altitudes of gravity waves excited by tropospheric forcing.* — J. Atmos. Sci., Vol.53, No.15, pp.2186–2216.

Prusa J.M., Garcia R.R. and Smolarkiewicz P.K. (1997): *Three-dimensional evolution of gravity wave breaking in the mesosphere.* — Proc. 11th Conf. *Atmos. Ocean. Fluid Dynamics*, Tacoma, WA, pp.J3–J4.

Prusa J.M., Smolarkiewicz P.K. and Wyszogrodzki A.A. (1999): *Parallel computation of gravity wave turbulence in the Earth's atmosphere.* — Apps. Adv. Archit. Comp., SIAM News, Vol.53, No.7, pp.1, 10–12.

Schumann U. (1991): *Subgrid length-scales for large-eddy simulation of stratified turbulence.* — Theoret. Comput. Fluid Dynamics, Vol.2, No.5, pp.279–290.

Scinocca J.F. (1995): *The mixing of mass and momentum by Kelvin-Helmholtz billows.* — J. Atmos. Sci., Vol.52, No.14, pp.2509–2530.

Smagorinsky J. (1993): *Some historical remarks on the use of nonlinear viscosities*, In: Large Eddy Simulation of Complex Engineering and Geophysical Flows (B. Galperin and S.A. Orszag, Eds.). — Cambridge: Cambridge University Press, pp.3–36.

Smolarkiewicz P.K. and Pudykiewicz J.A. (1992): *A class of semi-Lagrangian approximations for fluids.* — J. Atmos. Sci., Vol.49, No.22, pp.2082–2096.

Smolarkiewicz P.K. and Margolin L.G. (1993): *On forward-in-time differencing for fluids: Extension to curvilinear coordinates.* — Mon. Wea. Rev., Vol.121, No.6, pp.1847–1859.

Smolarkiewicz P.K. and Margolin L.G. (1994): *Variational solver for elliptic problems in atmospheric flows.* — Appl. Math. Comp. Sci., Vol.4, No.4, pp.527–551.

Smolarkiewicz P.K. and Margolin L.G. (1997): *On forward-in-time differencing for fluids: An Eulerian/semi-Lagrangian nonhydrostatic model for stratified flows.* — Atmos.-Ocean Special, Vol.35, No.1, pp.127–152.

Smolarkiewicz P.K. and Margolin L.G. (1998): *MPDATA: A finite-difference solver for geophysical flows.* — J. Comput. Phys., Vol.140, No.2, pp.459–480.

Smolarkiewicz P.K., Grubišić V. and Margolin L.G. (1997): *On forward-in-time differencing for fluids: Stopping criteria for iterative solutions of anelastic pressure equations.* — Mon. Wea. Rev., Vol.125, No.4, pp.647–654.

Weinstock J. (1985): *Theoretical gravity wave spectrum in the atmosphere: Strong and weak wave interactions.* — Radio Sci., Vol.20, No.6, pp.1295–1300.

## Appendix: Continuity Equation in Transformed Coordinates

We begin by noting that the velocity vector $\mathbf{v} \equiv (u, v, w) \equiv \mathrm{d}\mathbf{x}/\mathrm{d}t$ in Cartesian coordinates is a contravariant vector. Using the contravariant velocity, the form of continuity given by (2a) is readily established for Cartesian coordinates as well as for selected transformed, non-Cartesian systems. Next we introduce the 4-form of velocity, $\overline{\mathbf{v}^*} \equiv (1, \overline{u^*}, \overline{v^*}, \overline{w^*})$, where the leading dimension refers to time. In index form, $\overline{u^*}^i \equiv \mathrm{d}\overline{x^i}/\mathrm{d}\overline{t}$, where $i = 0, 1, 2, 3$, and 4 ($i = 0$ refers to time, $\overline{t}$). Using index notation, the velocity 4-form, and dividing by the Jacobian $\overline{G}$, (2a) may be written as

$$\frac{1}{\overline{G}} \frac{\partial(\overline{G}\rho\overline{u^*}^i)}{\partial\overline{x}^i} = 0. \tag{A1}$$

The form (A1) is readily identified as being the covariant derivative with respect to $\overline{x}^i$ of the contravariant vector $\rho\overline{u^*}^i$ contracted over $i$ (Aris, 1962). This establishes that (A1), and hence (2a)—when divided by $\overline{G}$—are tensor equations and thus true in all coordinate systems.

To establish (2b), we introduce the solenoidal velocity into (A1) using $\overline{u^*}^i \equiv \overline{u^s}^i + \partial\overline{x}^i/\partial t$, multiply by $\overline{G}$, and reorganize as follows:

$$\frac{\partial(\rho^*\overline{u^s}^i)}{\partial\overline{x}^i} = -\frac{\partial(\rho^*\overline{x}^i{}_{,t})}{\partial\overline{x}^i}. \tag{A2}$$

$$\longrightarrow \frac{\partial(\rho^*\overline{u^s}^j)}{\partial\overline{x}^j} = -\frac{\partial(\rho^*\overline{u^s}^0)}{\partial\overline{x}^0} - \frac{\partial(\rho^*\overline{x}^j{}_{,t})}{\partial\overline{x}^j} - \frac{\partial(\rho^*\overline{x}^0{}_{,t})}{\partial\overline{x}^0} \equiv -R, \tag{A3}$$

where $j = 1, 2$ and 3 only. Since $\overline{u^*}^0 \equiv 1$ and $\overline{x}^0{}_{,t} \equiv \overline{t}_{,t} \equiv \partial\overline{t}/\partial t \equiv 1$, it follows that $\overline{u^s}^0 \equiv 0$. These identities lead to

$$R \equiv \frac{\partial(\rho^*\overline{x}^j{}_{,t})}{\partial\overline{x}^j} + \frac{\partial\rho^*}{\partial\overline{t}}. \tag{A4}$$

$$\longrightarrow R \equiv \overline{G}\left(\overline{x}^i{}_{,t}\frac{\partial\rho}{\partial\overline{x}^i}\right) + \rho\left(\frac{\partial(\overline{G}\overline{x}^i{}_{,t})}{\partial\overline{x}^i}\right). \tag{A5}$$

Note that (A5) is written using index $i$ (ranging from 0 to 3) rather than $j$. The first term in the parentheses on the right-hand side in (A5) is the material derivative of the reference state density following the motion of the transformed coordinates, and is identically zero (the reference state is always stationary by choice). The second term in the parentheses represents the dilatation of the Jacobian of the transformation multiplied by the rate of stretching of the transformed coordinates, and is also identically zero. The detailed proofs of these two identities are tedious but straigtforward. They indicate that the only required conditions are that (i) the reference density must be independent of time, $\rho = \rho(x, y, z)$ and (ii) that the transformed coordinates can be transformed into *any* stationary, curvilinear system. Now that $R = 0$, (A3) reduces exactly to (2b) and we have the required result.

Received: 8 September 2000