An Ensemble Analysis of Forecast Errors Related to Floating Point Performance

STEPHEN J. THOMAS

Scientific Computing Division, National Center for Atmospheric Research, Boulder, Colorado*

JOSHUA P. HACKER

University of British Columbia, Vancouver, British Columbia, Canada

MICHEL DESGAGNÉ

Recherche en Prévision Numérique, Environment Canada, Dorval, Quebec, Canada

ROLAND B. STULL

University of British Columbia, Vancouver, British Columbia, Canada

23 March 2001 and 25 October 2001

ABSTRACT

The dynamical core of the Mesoscale Compressible Community (MC2) model is described. Ensemble forecast techniques for high-resolution mesoscale simulations are applied to assess the impact of floating point optimization, mathematics libraries, and processor configuration on forecast accuracy. It is shown that the iterative solver in the dynamical core is most sensitive to processor configuration, but it also shows weak sensitivity to the usage of fast mathematics libraries and floating point instruction reordering. Semi-implicit pressure solver errors are amplified in the physical parameterization package, which is sensitive to small pressure differences and feeds back to the dynamical solution. In this case, local rms spreads are around 1°C in temperature by the end of a 42-h forecast. It is concluded that careful validation is required when changing computing platforms or introducing fast mathematics libraries.

1. Introduction

In recent years there has been a rapid shift in the weather forecasting and research communities away from the use of centralized forecast products toward running mesoscale models on a regional or even local basis. For example, both the University of Washington (UW) and the University of British Columbia (UBC) have set up real-time forecasting systems. The Pennsylvania State University–National Center for Atmospheric Research (Penn State–NCAR) fifth-generation Mesoscale Model (MM5) has been run in production mode at many sites (Mass and Kuo 1998), including UW, whereas the Mesoscale Compressible Community model (MC2) and Wisconsin Nonhydrostatic Modeling System are run at UBC. In both cases the model production runs are carried out on microprocessor-based shared or distributed-memory clusters (Baillie et al. 1997). To sustain a significant percentage of peak performance, code restructuring in combination with compiler optimizations and fast mathematics (''math'') libraries are required. The most aggressive optimizations and libraries must be applied with care because they may seriously degrade the accuracy of a fluid flow simulation if the model code or the compiler is not robust. It is for this reason that we believe ensemble analysis techniques can be useful in assessing the impact of optimization strategies on atmospheric models with complex physical parameterization packages.

In this note we apply ensemble forecast techniques to assess the effect of floating point optimizations, fast math libraries, and processor configuration on model error, which have been shown to dominate in the first few hours of forecast time (Stensrud et al. 2000; Orrell et al. 2002). Our study is based on the nonhydrostatic fully compressible MC2 limited-area atmospheric model. The MC2 is widely used in Canadian universities and at Environment Canada for mesoscale and micro-

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

Corresponding author address: Dr. Stephen J. Thomas, Scientific Computing Division, National Center for Atmospheric Research, 1850 Table Mesa Dr., Boulder, CO 80303.
E-mail: thomas@ucar.edu

scale atmospheric research. A detailed description of the adiabatic kernel and numerical formulation of the MC2 model with open boundaries is given in Thomas et al. (1998). In particular, the model employs a semi-implicit semi-Lagrangian time discretization scheme. The use of a terrain-following height coordinate (Gal-Chen and Sommerville 1975) results in a nonsymmetric linear system to solve every time step for the pressure. An iterative minimal residual Krylov solver has thus been implemented, and the computational efficiency of the model using line relaxation preconditioners designed for both hydrostatic and nonhydrostatic flow regimes is presented. The use of an iterative solver on a parallel computer leads to spread in the ensemble errors, and this spread is amplified by the physical parameterization package in combination with fast math libraries and changing processor configurations.

Atmospheric models based on explicit time stepping schemes such as the Penn State–NCAR MM5 would not be subject to the same forecast error spreads because the pressure is bit-reproducible and does not depend on the processor configuration. Indeed, physical parameterizations are strictly column-based and do not require interprocessor communication for distributed-memory computation. However, several operational centers including Canada (Côté et al. 1998) use semi-implicit models based on iterative solvers, and our results indicate that model validation should also account for the variation in the pressure with processor configuration. To establish that the error spread in the physics is in fact due to the iterative pressure solver, a sensitivity study was performed using one processor by varying the iterative solver convergence tolerance.

2. MC2 model formulation

The MC2 is a nonhydrostatic fully compressible limited-area model formulated with respect to a horizontally homogeneous, time-invariant, and hydrostatically balanced base state reference atmosphere (subscript 0). Thermodynamic variables, temperature T, and normalized log pressure $q = \ln(p/p_0)$ are decomposed into a sum of the base state and perturbations:

$$T = T_0 + T', \qquad q = q_0(z) + q',$$

where

$$\frac{\partial q_0}{\partial z} = -\frac{g}{RT_0}$$

and the equation of state is $p = \rho RT$. The gas constant for dry air is R, $\gamma = c_p/c_v$ and $c_p - c_v = R$, where c_p and c_v are specific heats at constant pressure and volume, ρ is density, z is geometric height, and g is gravitational acceleration. After introduction of the base state, the governing equations are given by

$$\frac{D\mathbf{v}}{Dt} = -RT\boldsymbol{\nabla}q' + B\mathbf{k} - f\hat{\mathbf{k}} \times \mathbf{v}$$

$$\frac{DT'}{Dt} - \frac{RT_0}{c_p}\frac{Dq'}{Dt} = -T_0\frac{N_0^2}{g}w - \frac{RT'}{c_v}\boldsymbol{\nabla} \cdot \mathbf{v}$$

$$\frac{Dq'}{Dt} = -\gamma\boldsymbol{\nabla} \cdot \mathbf{v} + \gamma\frac{g}{c_0^2}w,$$

where $\mathbf{v}$ is the velocity, f is the Coriolis force, $B = gT'/T_0$ is the buoyancy, and c_0 is approximately the speed of sound.

The semi-implicit scheme results in an elliptic problem to solve every time step for a log pressure perturbation q' about a stationary isothermal hydrostatic basic state:

$$(1 - c_0^2\Delta t^2\tilde{\nabla}_h^2)q' - c_0^2\Delta t^2 D_z^{(2)} N^{-1} D_z^{(1)} q' = Q^*_{q'}.$$

This nonsymmetric system of equations resulting from a terrain-following height coordinate system is solved using the generalized minimal residual (GMRES) algorithm of Saad and Schultz (1986). Skamarock et al. (1997) use the mathematically equivalent generalized conjugate residuals (GCR) algorithm of Eisenstat et al. (1983) in a semi-implicit formulation of a compressible model. The solver convergence criteria are based on the rms divergence. Computational efficiency (overall wall-clock time) is improved by finding a suitable preconditioner, which is often problem dependent. The discretized elliptic operator in a nonhydrostatic pressure solver will be dominated by the vertical terms when the aspect ratio $\Delta X/\Delta Z$ is large. Therefore, an effective preconditioning strategy is to invert the vertical components of the elliptic operator, and Skamarock et al. (1997) apply a vertical alternating direction implicit (ADI) line relaxation preconditioner.

A vertical ADI line relaxation preconditioner for the $n \times n$ linear system $\mathbf{A}x = b$ is based on the splitting $\mathbf{A} = \mathbf{H} + \mathbf{V}$, where the $\mathbf{H}$ and $\mathbf{V}$ represent the horizontal and vertical components of the discrete elliptic operator based on centered second-order finite differences. The ADI iteration is derived from the pseudo–time integration of the heat equation $u_t = \Delta u + r$ to steady state, where the matrix $\mathbf{A}$ represents the discrete Laplacian and

$$(I - \beta\mathbf{V})x^{k+1} = (I + \beta\mathbf{H})x^k - \beta b.$$

The largest possible pseudo–time step β is chosen so that the above integration scheme remains stable. A slightly more implicit scheme can be constructed using a line Jacobi relaxation scheme. For nonhydrostatic problems on isotropic grids, a fully 3D ADI preconditioner is implemented as in Skamarock et al. (1997). The solution of tridiagonal linear systems of equations implies global data dependencies; thus, a parallel data transposition strategy has been adopted in a distributed-memory implementation of the 3D ADI preconditioner.

For a distributed-memory model of parallel compu-

TABLE 1. MC2 execution time on an SGI Origin 2000 (h:min).

$P_X \times P_Y$	1 × 1	2 × 2	2 × 4	3 × 4	4 × 4
1D Jacobi	29:09	7:18	3:34	2:37	2:00
3D ADI	32:51	8:13	3:38	2:42	2:19

tation, the $N_i \times N_j \times N_k$ computational grid is partitioned across a $P_X \times P_Y$ logical processor mesh. A domain decomposition in the horizontal direction is employed because of the strong vertical coupling in physical parameterization packages and since the number of grid points in the vertical direction is typically one order of magnitude less than in the horizontal. Each processor therefore contains $N_i/P_X \times N_j/P_Y \times N_k$ points. For both algorithms the communication overhead associated with boundary data exchanges between subdomains is minimal when compared with computations. The 1D vertical ADI and Jacobi line relaxation preconditioners are also well suited to a horizontal decomposition, since the only global data dependency is in the vertical direction within tridiagonal solvers. However, the 3D ADI preconditioner requires global data in each of the three coordinate directions in order to solve tridiagonal linear systems of equations during each ADI sweep. Thus, the right-hand side b and solution x^k must be remapped to perform line relaxations in each coordinate direction in turn. Such a remapping takes the form of a data transposition algorithm requiring collective all-to-all communication of $O(N^3)$ grid points. Vertical sweeps in the Z direction are performed using the domain decomposition described above. Each processor contains $N_i \times N_j/P_Y \times N_k/P_X$ grid points for sweeps in the X direction and $N_i/P_Y \times N_j \times N_k/P_X$ points in the Y direction. ADI sweeps progress from left to right and then right to left as indicated below with arrows representing communication steps:

$$\frac{N_i}{P_X} \times \frac{N_j}{P_Y} \times N_k \Leftrightarrow N_i \times \frac{N_j}{P_Y} \times \frac{N_k}{P_X} \Leftrightarrow \frac{N_i}{P_Y} \times N_j \times \frac{N_k}{P_X}.$$

To compare the computational efficiency of the model using the parallel 1D Jacobi and 3D ADI line relaxation schemes, a quasi-hydrostatic test case was run on a Silicon Graphics, Inc., (SGI) Origin computer with 16 processors. A 120 × 120 × 35 grid at 2.5-km horizontal resolution with model lid set at 23 km was employed in a 30-h mesoscale forecast over the British Columbia lower mainland. The integration consisted of 1800 time steps of length $\Delta t = 60$ s using both 1D and 3D preconditioners. The results are summarized in Table 1 using optimization level O3 (described in section 3). Despite the fact that the 3D ADI preconditioner results in a faster solver convergence rate, model execution times are close.

3. Ensemble analysis

The effect of compiler optimizations along with the use of fast math libraries on the accuracy of high-resolution mesoscale forecasts is an unexplored subject. One possible way of studying these effects is to generate an ensemble of forecast runs, regarding optimization and processor (PE) configuration as sources of model error. Given three optimization levels and four processors, two simple ensembles are generated. The same meteorological test case is run 12 times, without including physical parameterization schemes, but varying optimization levels and processor configurations. The 12 runs compose an ensemble, and standard analysis techniques can reveal error growth and forecast spread as a function of optimization, library usage, and floating point instruction reordering. A second ensemble is generated by running the MC2 with the external physics package to determine the relative contribution of model error from the dynamical core as compared with the physics. The use of different PE configurations and optimization levels are considered perturbations to the model, although they may be systematic rather than stochastic.

Three optimization levels were identified for the 195-MHz R10000 processor on an SGI Origin 2000 (in this paragraph, compiler option codes are given in parentheses). The first level of optimization (O1) allows some minimal code changes by the compiler. Interprocedural analysis (-IPA) is applied at the second level of optimization (O2) to improve both instruction and data cache usage through code movement. The most aggressive level of optimization (O3) permits floating point instruction reorderings that could adversely affect the precision of certain computations such as division. Fast math libraries (-lfastm) are also employed and precision may be reduced in exchange for speed. To evaluate these optimizations in the context of a real-time mesoscale forecasting system, the computational domain and related run-time parameters were taken from daily runs of the UBC ensemble forecasting system (http://spirit.geog.ubc.ca/wxfcst). Horizontal resolution is 10 km at 60°N with the model lid at 19 km. The grid contains 120 × 70 points on 35 terrain-following levels, implying the run is quasi-hydrostatic with $\Delta X/\Delta Z = 20$. Vertical line Jacobi preconditioning is therefore applied in all our tests.

Initial and boundary conditions are obtained from a coarse grid ($\Delta X = 30$ km) MC2 run starting at 0000 UTC 27 November 1997. A 42-h integration requires 1260 time steps of length $\Delta t = 120$ s. For the ensemble that includes physics, the full Recherche en Prévision Numérique (RPN) physics package described in Mailhot et al. (1997) is used. A forecast initialized at 0600 UTC 27 November 1997 is chosen as a benchmark since it is a representative mesoscale case for the British Columbia lower mainland. In particular, it exhibits typical autumn and winter characteristics, including moist, southwesterly flow at low levels;

TABLE 2. Optimization level O1, on a 195-MHz R10000.

$P_X \times P_Y$	1 × 1	1 × 2	1 × 3	2 × 2
Wall clock	31 395	16 666	11 169	8034
User	30 975	16 339	10 828	7879
Mflops	53.6	101.5	153.2	210.6
Mflops/CPU	53.6	50.8	51.1	52.6

TABLE 3. Optimization level O2, on a 195-MHz R10000.

$P_X \times P_X$	1 × 1	1 × 2	1 × 3	2 × 2
Wall clock	28 597	14 593	9707	7420
User	28 223	14 341	9449	7252
Mflops	58.8	115.7	175.6	228.8
Mflops/CPU	58.8	57.8	58.5	57.2

a persistent upper-level trough offshore; and dynamically forced precipitation enhanced by steep topography. The dynamic forcing came from a weak surface cold frontal passage supported by a short-wave trough aloft. Upstream boundary wind speeds were on the order of 10 m s^{-1} at the surface. Wall-clock execution times (including input/output) are summarized in Tables 2 and 3. Note in particular that single processor performance does not exceed 65 million floating point operations per second (Mflops) or 15% of the rated peak performance of the microprocessor.

Forecast spread suggests model error resulting from choices in processor configuration and optimization level. Unlike many ensemble studies, a control run to characterize deterministic model error is not available. In ensemble forecasting, a control run is often the categorical forecast, initialized with a ''best guess'' analysis. A single processor run without any optimization could be taken as the control run since it may be what the model developers intended. Given codes of such complexity this is difficult to quantify. Alternately, the control can be any member of the ensemble, chosen randomly. However, one case study cannot robustly determine the statistical properties of the ensemble and a randomly chosen member has no significance. Ensemble forecasters often look for forecast spread and a reduced ensemble-averaged error. The performance of the ensemble mean per se is not of interest here and sensitivity to initial conditions is not an issue. Because this is a single case without a true control run, it is unwise to select a top performer in this experiment or to average the results looking for an improvement. However, significant deviation of one group of forecasts from the others would indicate the presence of model error.

Each ensemble member is initialized identically and common boundary conditions are applied throughout the 42-h integration period. The 0-h forecasts may deviate nominally from each other, because grid staggering and destaggering require calculations that may be handled differently. Boundary value determination is subject to the same errors and by examining the spatial distribution of forecast spread such effects can be quantified. Forecast spread is defined quadratically as

$$S(x, y) = \left\{ \frac{1}{N} \sum_{j=1}^{N} [\psi_j(x, y) - \overline{\psi}(x, y)]^2 \right\}^{1/2}, \qquad (1)$$

where ψ is any forecast variable, $\overline{\psi}$ is the ensemble mean, and N is the number of ensemble members.

This study evaluates temperature T and vertical velocity W, because those two parameters largely determine the state of the model. The mass distribution is mostly described by T, while W is unique for a given wind solution. The vertical structure of the spread and the impact of optimization and processor configuration on the RPN physics package are shown in Fig. 1. It is clear for both vertical velocity and temperature that the spread in the ensemble is greater when the model is run with the full-physics package (dashed line). The dynamics do not use the functions available in the fast math library, whereas the physics do in many places. When physics are included, the spread increases with height. Without physics the spread is greater in the middle of the troposphere. In all cases after 36 h of forecast time, the spread of the physics ensemble is at least 2 times as large as the spread of the no-physics ensemble, showing that the library and processor configuration differences affect the physics more than the dynamics. Spreads in Fig. 1 increase rapidly at the beginning of the forecasts and then grow slowly or even decrease in some cases. This behavior is consistent with observed model error that follows a square-root curve (Orrell et al. 2002). Later in the forecast, we would expect sensitivity to initial conditions to take over, giving exponential-like error growth (e.g., Errico and Baumhefner 1987). The small domain here will restrict such error growth, explaining the weak growth in Fig. 1.

Keeping track of how each ensemble member deviates from the ensemble mean gives us an idea of specifically how the fast math library and processor configuration affect the forecast. Figure 2 is a plot of domain-averaged absolute deviations for each member in the ensembles. Lower values indicate less contribution to the spread. Most of the error growth is contained within the first few hours of the forecast. Because of the large spread near the top of the troposphere, that level was chosen for demonstration, and only temperature deviations are shown. The ensemble with physics is on the left (Fig. 2a) and the dynamics-only ensemble is on the right (Fig. 2b). Note the different vertical axis scales and that the deviations are an order of magnitude greater when physics are included. Each configuration

TABLE 4. Optimization level O3, on a 195-MHz R10000.

$P_X \times P_Y$	1 × 1	1 × 2	1 × 3	2 × 2
Wall clock	26 862	13 919	8769	7099
User	26 828	13 679	8578	6935
Mflops	61.8	121.3	193.4	239.2
Mflops/CPU	61.8	60.6	64.5	59.8

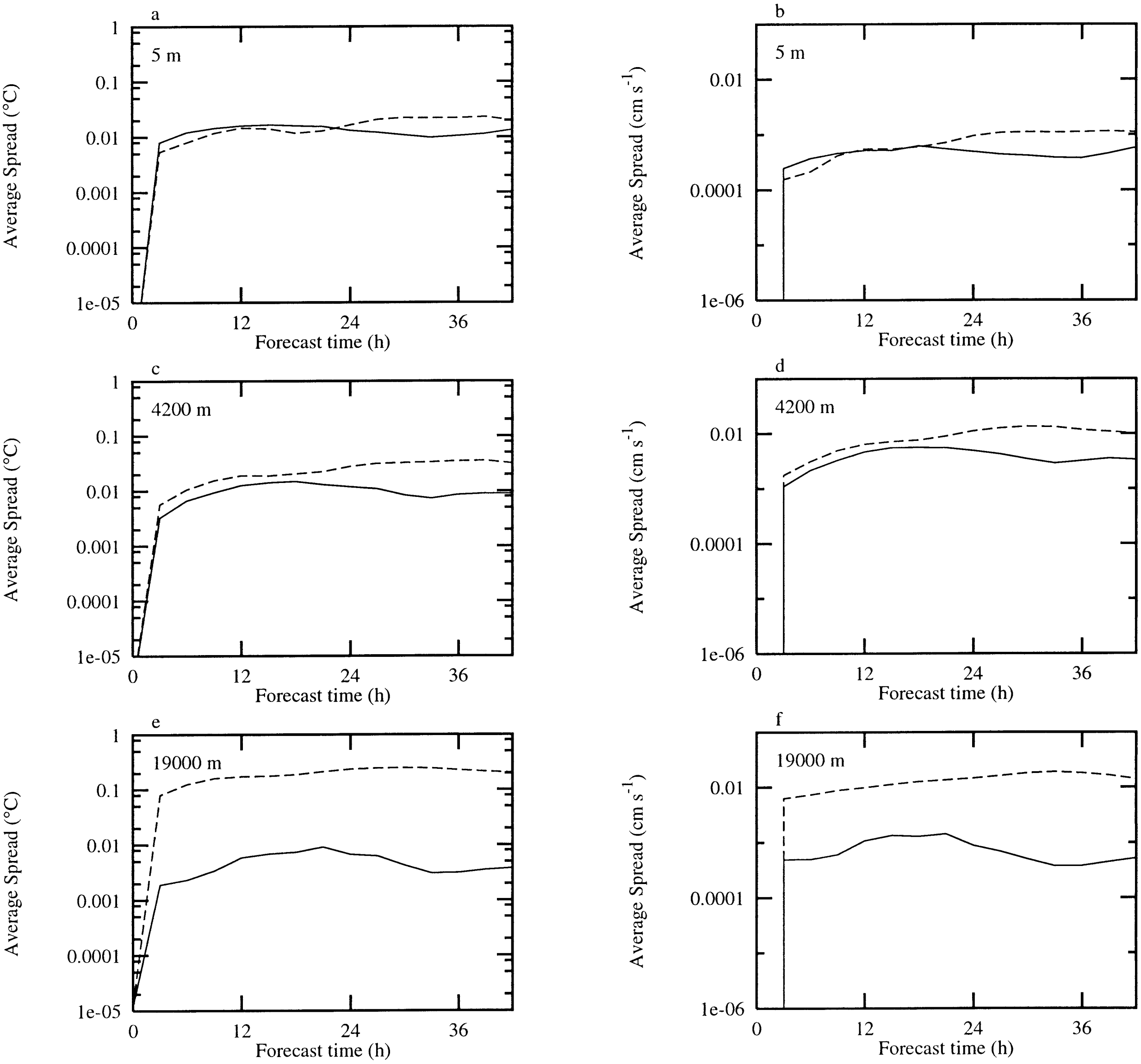

FIG. 1. Ensemble spread of (left) temperature (*T*) and (right) vertical velocity (*W*) averaged over the model grid points for computational levels at approximately (a), (b) 5, (c), (d) 4200, and (e), (f) 19 000 m above the surface. The dashed line is calculated from the ensemble with physics, and the solid line is from the ensemble without physics.

is plotted with a different line style, and all 12 members are plotted. Different optimization levels and libraries have nearly identical deviations from the mean, since the curves appear superimposed in Fig. 2. Thus, the processor configuration is almost completely responsible for the ensemble spread. In a global model, error growth is bounded by the climatological variance (Errico and Baumhefner 1987). Common boundary conditions in this ensemble run on a small, limited-area domain further restrict error growth. This results in an error saturation that is well below where it might occur on a global domain. Because the error growth is faster for the ensemble with physics included, saturation appears to be reached near where the curves are flat and the spread may have reached its limit as early as 24 h into the forecast (Fig. 2a). Conversely, many of the curves in Fig. 2b tend upward at the end of the forecast period, suggesting that the slower-growing spread in the dynamics-only ensemble has not yet reached its limit.

Plotting the spread for one forecast time without averaging shows that a variety of calculations performed in the MC2 are susceptible to processor configuration, but the physics are more sensitive. Figure 3 shows the temperature spread approximately 5 m above the surface (first model level) and 19 000 m above the surface for both ensembles, valid near the peak spreads in the phys-

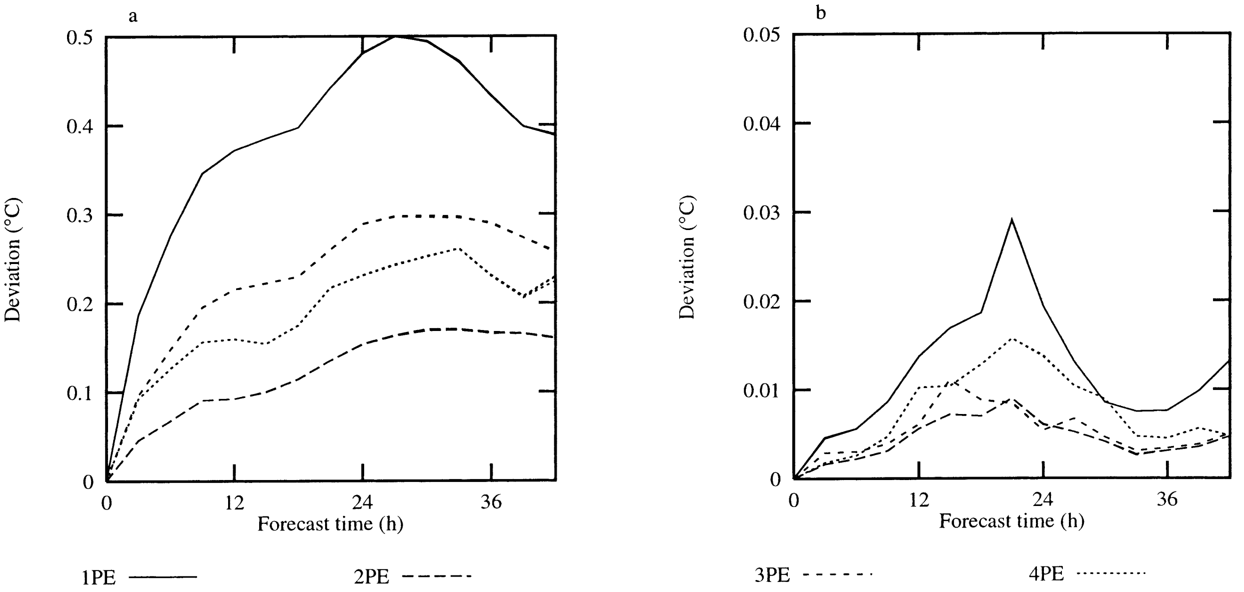

FIG. 2. Deviation from the ensemble mean for (a) the ensemble with physics included and (b) the ensemble with only the MC2 dynamical core. All 12 members of the ensemble are plotted, and each processor configuration uses a different line style (for three optimization levels each) as given in the legend. The differences between the three optimization levels are nearly undetectable. These results are for temperature at 19 000 m.

ics ensemble (Fig. 2) at forecast hour 30. It is again clear that the spread of the ensemble with physics (Figs. 3a and 3c) is an order of magnitude larger than the ensemble without physics (Figs. 3b and 3d). The 5-m spreads (Figs. 3a and 3b) are smaller, corresponding with Fig. 1, and do not appear to be correlated with topography as might be expected. The mountain ranges in this domain run parallel to the coastline, while the spread pattern for the no-physics ensemble is oriented perpendicular to the coastline, and the physics-included ensemble spread does not have an organized pattern at all. Figure 3c shows the importance of boundary condition contamination, but the large upper-tropospheric spread patterns are also oriented with the topography. Most of these results are for temperature because it is a sensible weather parameter that is easy to understand. The vertical velocity behaves similarly (Fig. 1). Because precipitation processes are closely (and nonlinearly) tied to vertical velocity, these results imply that a large spread in forecast precipitation events could result in some cases, especially when it is weakly forced.

To estimate the relationship between topography and the spread of the ensembles, correlation coefficients between the spread of *W* and topography are calculated (Fig. 4). Topographically forced vertical velocity is handled entirely in the dynamical core of the MC2, regardless of the external physical parameterizations. Vertical velocities themselves are well correlated with topography, especially with cross-barrier flow as in this case. A strong correlation between the spread of vertical velocity and the topography would indicate that the numerics associated with vertical advection may introduce errors with different processor configurations. Figure 4 shows a positive correlation through most of the forecast, although the correlation is never particularly strong. As might be expected from Fig. 3, the correlation in the upper troposphere for the no-physics ensemble is the greatest and the no-physics correlations are in general higher than the correlations for the physics ensemble. The overall weak correlations suggest that the topographically forced vertical velocities in the model dynamics are not prone to large errors from processor configuration. The correlation reduction in the physics ensemble near the end of the forecast time also indicates that the spread in *W* is more closely connected with errors resulting from the physical parameterizations, which are decoupled from the topography.

The results presented here suggest that PE configuration is the primary source of error in the iterative pressure solution. The errors by themselves may be small. When a full physical parameterization package is included, the pressure errors propagate through the physics and positively feed back to degrade the full model solution. The RPN physics package treats each atmospheric column separately, so processor configuration should not directly affect the physics output. To further test this hypothesis, runs were made with different elliptic solver convergence tolerances. The runs were made on one PE to eliminate processor configuration as a source of error, varying the iterative solver convergence tolerances from $\epsilon = 10^{-4}$ to $\epsilon = 10^{-5}$. Both of these result in converged solutions, but some small differences will result. Again, runs were made both with and without physics, giving a total of four

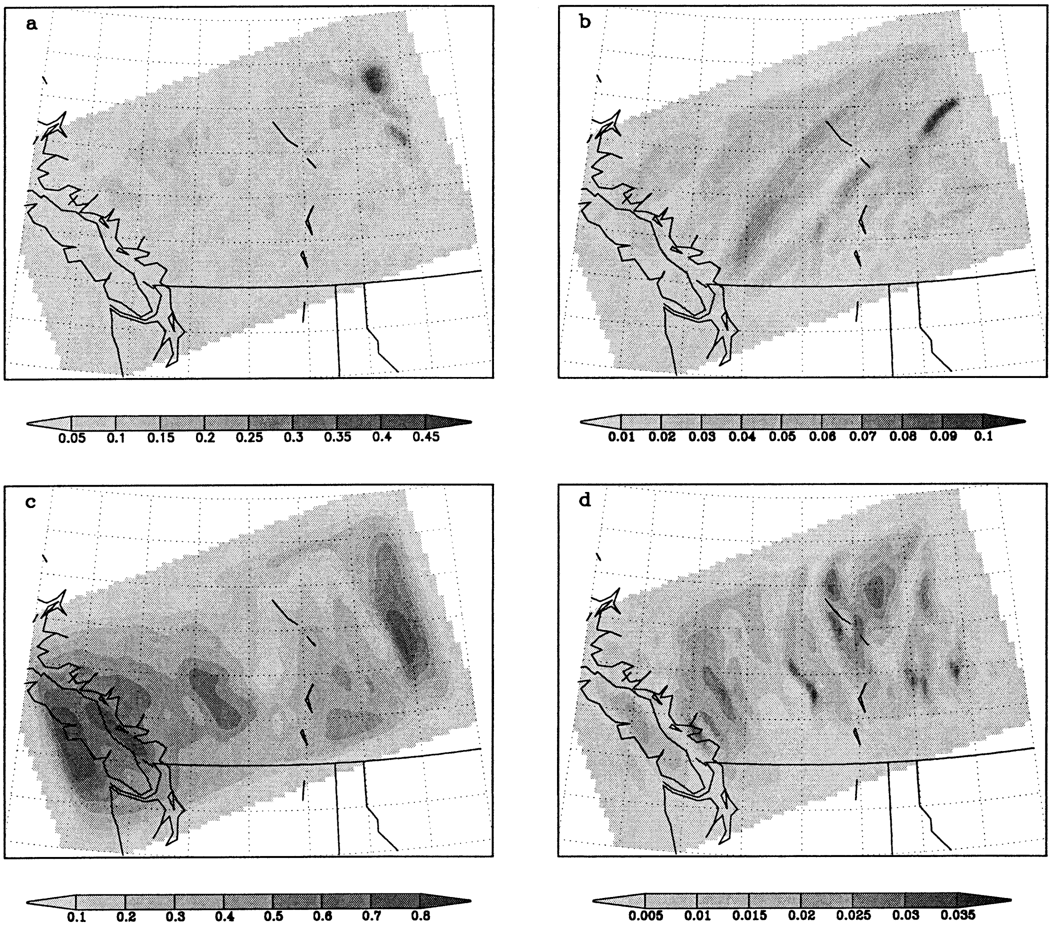

FIG. 3. Ensemble spread for temperature as defined in Eq. (1). (a), (b) Spreads at 5 m and (c), (d) spreads at 19 000 m. (a), (c) Results from the ensemble with full physics, and (b), (d) results from the ensemble with no physics. Note the difference scales for each map. Peak values are in the largest category defined by the grayscale bars.

runs. The results are shown in Fig. 5, which can be compared with Fig. 1. Although the overall magnitude of the spreads is smaller, the fact that the physics results in an order-of-magnitude increase in spread over the dynamics-only case is similar.

4. Conclusions

Perhaps the best way to evaluate the impact of code optimization strategies for microprocessors on atmospheric flow simulations is by using theoretical test cas-

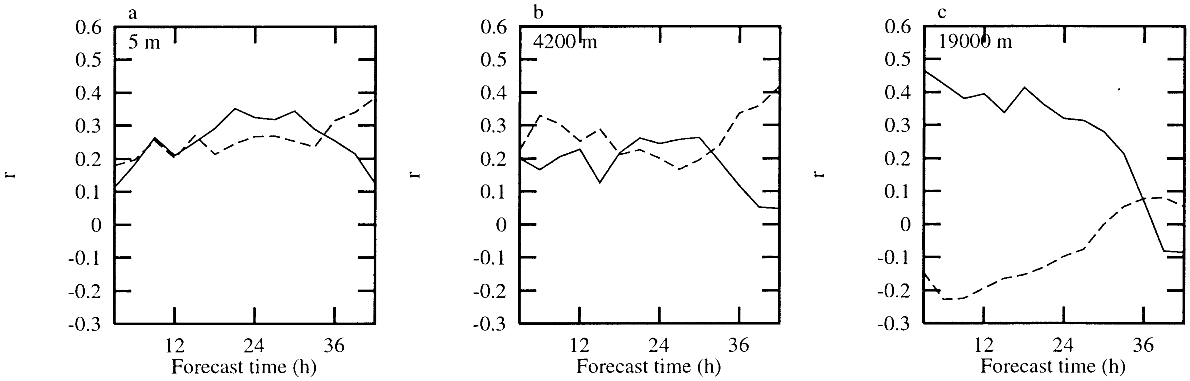

FIG. 4. Correlation coefficients between the spread of vertical velocity and the topography of the MC2 domain. The three levels are the same as are shown in Fig. 1. Results from the no-physics ensemble are shown with a solid line and results from the physics ensemble are shown dashed.

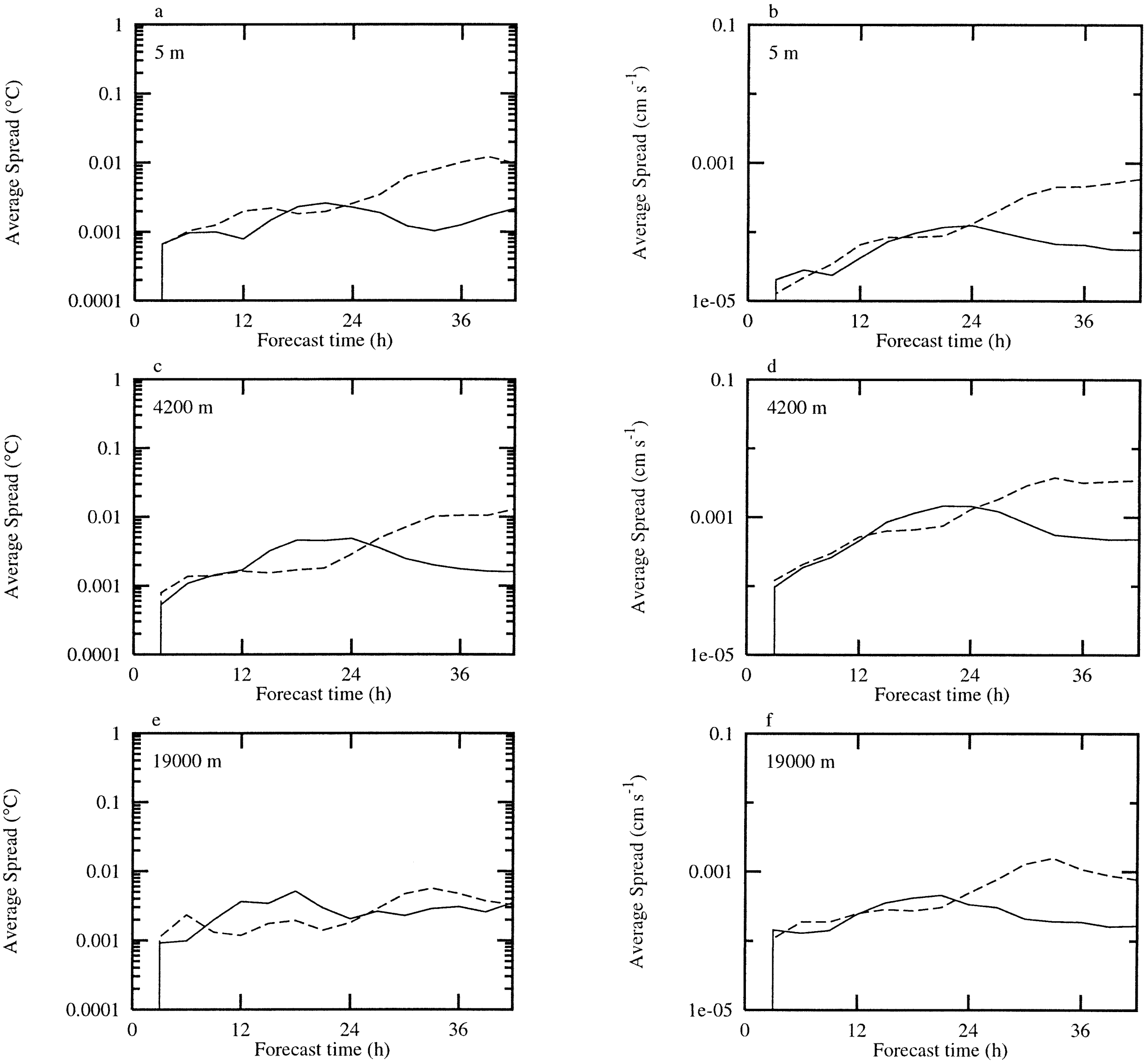

FIG. 5. Same as in Fig. 1. Runs were made with different convergence criteria for the iterative solver.

es with analytic or converged solutions. In such cases, a loss of accuracy can be isolated and easily corrected. With a fully configured atmospheric model interfaced to complex physical parameterization packages, the task is more difficult. One possible approach is to view aggressive compiler optimizations, fast math library usage, and parallel-processor topology as possible sources of floating point errors or perturbations of the model and boundary conditions. Ensemble forecast techniques can then be applied to determine the nature of the error growth that may result from these configuration options. If we interpret ensemble spread as the potential for model error from compiler optimization, fast math libraries, and processor configuration, the feedback between the iterative pressure solution and the physics can generate significant errors in the forecasts. Spread from an ensemble with only dynamics is an order of magnitude smaller than a run with the full RPN physics package. The deviations between ensemble members that use the fast math libraries and those that do not are nominal, showing that the spread is primarily a result of processor configuration. This is to be expected for the dynamics-only case.

The iterative solver is the main source of differences, because the pressure is not bit-reproducible because of a global summation when computing inner products in the iterative solver. However, changing processor configurations while using the physics package has the potential to degrade the forecast further. For a realistic forecast, the physics must be included, so it is up to the forecaster to determine whether the errors are within acceptable bounds. In this case, the implied model errors averaged over the domain were relatively small and on the order of 0.5°C in temperature. Additional study is required to quantify the effects that the implied vertical velocity errors can have on sensible weather parameters such as precipitation. Errors of this magnitude may only be important in select cases.

Other published model verification studies can help to put these errors into context. Stensrud et al. (1999) evaluated Eta Model forecasts produced at the National Centers for Environmental Prediction (NCEP), run on a regional domain with $\Delta X = 29$ km. For 12-h forecasts, they found mean absolute errors (MAEs) in temperature that ranged from 0.93° to 1.66°C between 85.0 and 10.0 kPa. At 36 h, the range of errors was from 1.16° to 1.73°C. Hou et al. (2001) verified an ensemble designed to approximate forecast error resulting from initial-condition error and found mean 85.0-kPa temperature errors between 1.0° and 2.0°C at 36 h. Both of those studies used larger domains than this study. With more recognition that model error is large even at short ranges (Stensrud et al. 2000, Orrell et al. 2002), it is important to understand its sources, including those related to processor topology and/or floating point performance. The results here suggest that processor topology can produce errors that are 10%–25% of the magnitude of total initial-condition and model error. These results are conservative because of the small domain. For a larger domain, it is expected that the errors could grow to larger values and they are not limited by the boundary conditions as they are here. Also, the suggested feedback between the pressure solution and the physics solution means that other sources of model error, such as incorrect physical parameters, may result in greater model error than different PE configurations.

It must be noted that experiments performed with an earlier math library and operating system resulted in errors on the order of 5°C, which are large enough to be important most of the time. Those differences were caused primarily by the use of the fast math library. In other words, these results do not extend to all computing platforms. Our results also show that the influence of topography on spread is weak. This was examined by looking at the spread in vertical velocity, because vertical velocity itself is expected to be highly correlated with topography. The no-physics ensemble is generally better correlated with topography. The physics and no-physics ensembles behave differently, suggesting that other factors are at least as important.

These results have implications for investigators trying to verify a numerical weather prediction model over one or more seasons. It is recognized that a proper verification will cover a period during which the model is static. That is to say the physics and the dynamical core do not change over the verification period. If this is important, the number of processors must remain static as well. Within this context, there is no way to tell which forecast is the most accurate because we have not included any information about model biases, observations, or observation errors. Regardless, the results show that processor configuration, and possibly the use of fast math libraries, can lead to forecast error that should be considered when a new compiler, operating system, or fast math libraries are installed.

Acknowledgments. We thank SGI, NCAR, and the Canadian Meteorological Centre for providing dedicated computer time. The comments of three anonymous reviewers have greatly improved the manuscript.

REFERENCES

Baillie, C., J. Michalakes, and R. Skalin, 1997: Regional weather modeling on parallel computers. *Parallel Comput.* **23,** 2135–2142.

Côté, J., S. Gravel, A. Méthot, A. Patoine, M. Roch, and A. Staniforth, 1998: The operational CMC–MRB Global Environmental Multiscale (GEM) model. Part I: Design considerations and formulation. *Mon. Wea. Rev.,* **126,** 1373–1395.

Eisenstat, S. C., H. C. Elman, and M. H. Schultz, 1983: Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.,* **2,** 345–357.

Errico, R., and D. Baumhefner, 1987: Predictability experiments using a high-resolution limited-area model. *Mon. Wea. Rev.,* **115,** 488–504.

Gal-Chen, T., and R. C. Sommerville, 1975: On the use of a coordinate transformation for the solution of the Navier–Stokes equations. *J. Comput. Phys.,* **17,** 209–228.

Hou, D., E. Kalnay, and K. Droegemeier, 2001: Objective verification of the SAMEX 98 ensemble study. *Mon. Wea. Rev.,* **129,** 73–91.

Mailhot, J., R. Sarrazin, B. Bilodeau, N. Brunet, and G. Pellerin, 1997: Development of the 35-km version of the operational regional forecast system. *Atmos.–Ocean,* **35,** 1–28.

Mass, C. F., and Y.-H. Kuo, 1998: Regional real-time numerical weather prediction: Current status and future potential. *Bull. Amer. Meteor. Soc.,* **79,** 185–200.

Orrell, D., L. Smith, J. Barkmeijer, and T. Palmer, 2002: Model error in weather forecasting. *Nonlinear Processes Geophys.,* in press.

Saad, Y., and M. Schultz, 1986: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.,* **7,** 856–869.

Skamarock, W. C., P. K. Smolarkiewicz, and J. B. Klemp, 1997: Preconditioned conjugate-residual solvers for Helmholtz equations in nonhydrostatic models. *Mon. Wea. Rev.,* **125,** 587–599.

Stensrud, D., H. E. Brooks, J. Du, S. Tracton, and E. Rogers, 1999: Using ensembles for short-range forecasting. *Mon. Wea. Rev.,* **127,** 433–446.

——, J.-W. Bao, and T. T. Warner, 2000: Using initial condition and model physics perturbations in short-range ensemble simulations of mesoscale convective systems. *Mon. Wea. Rev.,* **128,** 2077–2107.

Thomas, S. J., C. Girard, R. Benoit, M. Desgagné, and P. Pellerin, 1998: A new adiabatic kernel for the MC2 model. *Atmos.–Ocean,* **36,** 241–270.