Tellus (2008), 60A, 113–130
Printed in Singapore. All rights reserved

TELLUS

A local ensemble transform Kalman filter data assimilation system for the NCEP global model

By ISTVAN SZUNYOGH[1]*, ERIC J. KOSTELICH[2], GYORGYI GYARMATI[3], EUGENIA KALNAY[1], BRIAN R. HUNT[4], EDWARD OTT[5], ELIZABETH SATTERFIELD[1] and JAMES A. YORKE[6], [1]*Department of Atmospheric and Oceanic Science and Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA;* [2]*Department of Mathematics and Statistics, Arizona State University, AZ, USA;* [3]*Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA;* [4]*Institute for Physical Science and Technology and Department of Mathematics, University of Maryland, College Park, MD, USA;* [5]*Institute for Research in Electronics and Applied Physics, Department of Electrical and Computer Engineering and Department of Physics, University of Maryland, College Park, MD, USA;* [6]*Institute for Physical Science and Technology, Department of Mathematics and Department of Physics, University of Maryland, College Park, MD, USA*

(Manuscript received 14 December 2006; in final form 21 August 2007)

ABSTRACT

The accuracy and computational efficiency of a parallel computer implementation of the Local Ensemble Transform Kalman Filter (LETKF) data assimilation scheme on the model component of the 2004 version of the Global Forecast System (GFS) of the National Centers for Environmental Prediction (NCEP) is investigated.

Numerical experiments are carried out at model resolution T62L28. All atmospheric observations that were operationally assimilated by NCEP in 2004, except for satellite radiances, are assimilated with the LETKF. The accuracy of the LETKF analyses is evaluated by comparing it to that of the Spectral Statistical Interpolation (SSI), which was the operational global data assimilation scheme of NCEP in 2004. For the selected set of observations, the LETKF analyses are more accurate than the SSI analyses in the Southern Hemisphere extratropics and are comparably accurate in the Northern Hemisphere extratropics and in the Tropics.

The computational wall-clock times achieved on a Beowulf cluster of 3.6 GHz Xeon processors make our implementation of the LETKF on the NCEP GFS a widely applicable analysis-forecast system, especially for research purposes. For instance, the generation of four daily analyses at the resolution of the NCAR-NCEP reanalysis (T62L28) for a full season (90 d), using 40 processors, takes less than 4 d of wall-clock time.

1. Introduction

In Ott et al. (2004) we proposed a Local Ensemble Kalman Filter (LEKF) for atmospheric data assimilation. The most important features of this scheme are that (i) it assimilates all observations that may affect the analysis at a given grid point simultaneously and (ii) it obtains the analysis independently for each model grid point. An implementation of the LEKF on the National Centers for Environmental Prediction Global Forecast System Model (NCEP GFS) has previously been successfully tested using simulated grid point observations (Szunyogh et al., 2005). The LEKF was also used to generate initial conditions for the assessment of predictability in the NCEP GFS model (Kuhl et al., 2007).

Hunt et al. (2007) proposed many algorithmic changes to the LEKF and suggested that the new scheme should be called the Local Ensemble Transform Kalman Filter (LETKF). While the LETKF retains the two aforementioned distinguishing features of the LEKF, it also introduces changes that improve the computational efficiency of the algorithm and adds flexibilities that are beneficial when non-local observations, such as satellite radiances, are assimilated (Fertig et al., 2007b). The change in the name of the scheme reflects that, similar to the Ensemble Transform Kalman Filter (ETKF) of Bishop et al. (2001), matrix computations are done without orthogonalizing the background ensemble perturbations. When only local observations are assimilated, as done in the present paper, the LETKF and the LEKF schemes produce identical results. The main goal of this paper

*Corresponding author.
e-mail: szunyogh@ipst.umd.edu
DOI: 10.1111/j.1600-0870.2007.00274.x

is to assess the accuracy and computational efficiency of the LETKF in a realistic setting.

Section 2 describes our particular formulation of the LETKF for the NCEP GFS, while Section 3 explains our parallel computer implementation of the scheme. Section 4 provides a detailed description of the experimental design, and Section 5 presents the results of the numerical experiments. Section 6 summarizes our main findings. A reader more interested in the accuracy of the scheme than in the details of the computer implementation can skip Section 3, as the material presented in that section is not essential for understanding the rest of the paper. (More details on the computer implementation are given in Appendices A and B.)

2. Implementation of the LETKF on the NCEP GFS

In this section, we explain how the LETKF algorithm of Hunt et al. (2007) is implemented on the NCEP GFS. Hunt et al. (2007) provides not only a detailed justification of the LETKF algorithm, but also explains how different aspects of the scheme relate to ideas published by others in the literature. The reader interested in a comprehensive review of ensemble-based Kalman filters is referred to the recently published book by Evensen (2007) or to one of the review articles by Evensen (2003), Houtekamer and Mitchell (2005) and Hamill (2006).

As in Szunyogh et al. (2005), the analysis calculations are carried out in model grid space. Since the NCEP GFS model is a spectral-transform model, performing the grid space calculations requires that the members of the background ensemble first be transformed from spectral to grid space. Once the analysis step is completed, the analysis ensemble is transformed back from grid to spectral space. The components of the state vector $\mathbf{x}$ in our description of the LETKF algorithm are the model grid point variables.

Following the convention at NCEP, we prepare analyses with 6-h frequency at 0000UTC, 0600UTC, 1200UTC and 1800UTC, using observations from a 6-h window centred at the analysis time. However, unlike the current NCEP data assimilation scheme, where a single analysis is prepared at each time, we generate a k-member ensemble of global analyses, $\mathbf{x}^{a(i)}$, $i = 1, 2, \ldots, k$.

The background is an ensemble of global forecast trajectories $\mathbf{x}^{b(i)}(t)$, $i = 1, 2, \ldots, k$, for the 6-h time window from which observations are considered. These trajectories are obtained by integrating each ensemble member for 9 h starting from the global analysis ensemble of the previous analysis time. That is, the members of the global background ensembles are all 3–9-h lead time forecast trajectories. The role of the forecast trajectories is to provide the information needed to compute the ensemble mean background state and the ensemble-based estimate of the background error covariance at the times and locations of the observations (Hunt et al., 2004, 2007; Fertig et al., 2007a). To suppress gravity wave oscillations, the background trajectories and all other forecasts described in this paper are initialized with a digital filter algorithm (Lynch and Huang, 1992) employing a 3-h half-window.

The data assimilation procedure we follow is summarized by the nine steps given below.

(i) The observation operator H is applied to each ensemble background trajectory, $\mathbf{x}^{b(i)}(t)$, $i = 1, 2, \ldots, k$, to obtain the global background observation ensemble $\{\mathbf{y}^{b(i)}\}$, $i = 1, \ldots, k$. The ensemble average of the background ensemble is calculated in observational space by calculating the ensemble average $\bar{\mathbf{y}}^b$ of $\{\mathbf{y}^{b(i)}\}$. Then a global background observation ensemble perturbation matrix $\mathbf{Y}^b$ is constructed by taking its rows to be the vectors obtained by subtracting $\bar{\mathbf{y}}^b$ from each ensemble member $\mathbf{y}^{b(i)}$. Here, H is a four-dimensional observational operator that interpolates the background ensemble members to the time and location of the observations. The time interpolation is carried out by outputting the background ensemble trajectories with a 1-h frequency and applying a linear interpolation to obtain the background state at the observation time. In the two horizontal spatial dimensions, H is a simple bilinear interpolation. Since the vertical coordinate in the NCEP GFS model is sigma (defined as the ratio of the pressure to the surface pressure), while the vertical coordinate of the observations is pressure, the vertical interpolation is somewhat more complicated than the temporal and the horizontal interpolations: for each ensemble member, we first calculate the pressure at the sigma levels, multiplying sigma by the background surface pressure of the given ensemble member; the 28 sigma levels define 28 sigma layers, where the lowest layer is defined by the surface (where sigma is 1) and the lowest sigma level; we find the sigma layer that contains the observation; finally we linearly interpolate using the logarithm of the pressure values at the bottom and top of the sigma layer. Since the logarithm of the surface pressure is part of the state vector, H is a non-linear function.

(ii) The global background ensemble perturbation matrix $\mathbf{X}^b$ is constructed by first taking the ensemble mean $\bar{\mathbf{x}}^b$ of $\{\mathbf{x}^{b(i)}\}$ at the analysis time and then taking the ith column of the matrix to be $\mathbf{x}^{b(i)} - \bar{\mathbf{x}}^b$.

(iii) All data needed to obtain the analysis ensemble at a given grid point are selected. In this study, we assimilate only local observations and follow the strategy outlined in Ott et al. (2004) and Szunyogh et al. (2005): observations are used if they lie in the local volume centred at the grid point where the state is to be estimated. Below, $\bar{\mathbf{y}}^b_{[\ell]}$ and $\mathbf{Y}^b_{[\ell]}$ refer to the rows of $\bar{\mathbf{y}}^b$ and $\mathbf{Y}^b$ that correspond to the *local* background information at the location of the observations that were chosen for the calculation of the local analysis ensemble, $\{\mathbf{x}^{a(i)}_{[\ell]}\}$, at the given grid point. The notation $\mathbf{y}^o_{[\ell]}$ refers to the local vector of observations, and $\mathbf{R}_{[\ell]}$ is the local observation error covariance matrix for the chosen set of observations. (Hereafter, the subscript $[\ell]$ refers to a local region associated with an arbitrary grid point.)

(iv) The matrix $\mathbf{C}_{[\ell]} = (\mathbf{Y}^b_{[\ell]})^T \mathbf{R}^{-1}_{[\ell]}$ is computed. We define the entries of $\mathbf{R}_{[\ell]}$ by the values provided by NCEP in the operational observational data files. In addition, we assume that the observational errors are uncorrelated. This assumption makes $\mathbf{R}_{[\ell]}$ diagonal, so the calculation of $\mathbf{C}_{[\ell]}$ is inexpensive. [Hunt et al. (2007) suggest solving $\mathbf{R}_{[\ell]}\mathbf{C}^T_{[\ell]} = \mathbf{Y}^b_{[\ell]}$ for $\mathbf{C}_{[\ell]}$ when $\mathbf{R}_{[\ell]}$ is non-diagonal.] In some of the experiments, we reduce the weight of observations that are located further than a given distance (e.g. 500 km in this paper) from the analysis grid point. We achieve this by multiplying the entries of $\mathbf{R}^{-1}_{[\ell]}$ by a factor $\mu(r) \leq 1$, which is a monotonically decreasing function of the distance, r, between the location of the state estimation and the location of the observation.

(v) The eigensolution of $(k-1)\mathbf{I}/\rho + \mathbf{C}_{[\ell]}\mathbf{Y}^b_{[\ell]}$ is determined. The eigenvectors and eigenvalues are then used to calculate the matrix $\tilde{\mathbf{P}}^a_{[\ell]} = \left[(k-1)\mathbf{I}/\rho + \mathbf{C}_{[\ell]}\mathbf{Y}^b_{[\ell]}\right]^{-1}$. Here $\rho \geq 1$ is a multiplicative covariance inflation factor. In our implementation, ρ is a smoothly varying, three-dimensional scalar field. (See Section 4 for more details.)

(vi) The matrix $\mathbf{W}^a_{[\ell]} = [(k-1)\tilde{\mathbf{P}}^a_{[\ell]}]^{1/2}$ is computed using the eigenvalues and eigenvectors calculated in the previous step.

(vii) The vector $\bar{\mathbf{w}}^a_{[\ell]} = \tilde{\mathbf{P}}^a_{[\ell]}\mathbf{C}_{[\ell]}(\mathbf{y}^o_{[\ell]} - \bar{\mathbf{y}}^b_{[\ell]})$ is computed and then added to each row of $\mathbf{W}^a_{[\ell]}$. The columns of the resulting matrix are the weight vectors $\{\mathbf{w}^{a(i)}_{[\ell]}\}$.

(viii) The analysis ensemble members $\{\mathbf{x}^{a(i)}_{[\ell]}\}$ at the analysis grid point are obtained from $\mathbf{x}^{a(i)}_{[\ell]} = \mathbf{X}^b_{[\ell]}\mathbf{w}^{a(i)}_{[\ell]} + \bar{\mathbf{x}}^b_{[\ell]}$; here $\mathbf{X}^b_{[\ell]}$ and $\bar{\mathbf{x}}^b_{[\ell]}$ represent the rows of $\mathbf{X}^b$ and $\bar{\mathbf{x}}^b$ corresponding to the analysis grid point.

(ix) After completing steps (iii)–(viii) for each grid point, the results of step (viii) are collected to form the global analysis ensemble $\{\mathbf{x}^{a(i)}\}$.

3. Parallel code implementation

In this section, we outline our computer implementation of the LETKF, including program design, preliminary timing results, and parallelization issues. Appendix A explains the computational procedure for the simple case when only two processors are used, while Appendix B provides further details on the technical aspects of the program design.

The code consists of three main components: pre-analysis, analysis, and post-analysis. Pre-analysis transforms the background ensemble from spectral to grid-point space, executes steps (i)–(iii) of the LETKF algorithm and distributes the model grid points and the data from the associated local regions between the processors. Analysis carries out steps (iv)–(viii), while post-analysis carries out step (ix) and transforms the analysis ensemble from grid to spectral space.

Of the three main components of the code, the analysis is by far the most computationally expensive: it takes about 90% of the total CPU time. The wall-clock time is primarily limited by the slowest processor in the analysis component, but an efficient load-balancing scheme must also account for the time needed to distribute the data among the processors.

3.1. The pre- and post-analysis components

Given p processors and k ensemble solutions, the simplest implementation is to transform k/p solutions from spectral to physical space on each processor if p divides k (and any leftover solutions distributed in the obvious way otherwise). The transforms between spectral and physical space, which are adapted from the GFS model code, are fast: they take less than 10% of the total wall-clock time.

However, the data transport costs are significant: they comprise approximately 1/3 to 1/2 of the total wall-clock time in our current implementation on a Beowulf cluster. The present version of the GFS writes out one disk file for every background ensemble solution at every forecast time. Since we need the background trajectory with 1-h resolution, there are seven forecast files that must be read from disk for each member of the ensemble background. In the present implementation, all these files are written to a central disk farm; significant time savings would be possible if a distributed file system were used instead. (Appendix B contains additional discussion on this topic.)

3.1.1. Observation lookup. Step (iii) of the LETKF requires finding all the observations that belong to a given local region. Without an efficient lookup algorithm, this step can dominate the total computation time. The data structure of choice for this problem is the K-D tree, which is a binary search tree whose comparison key is cycled between K components. Here $K = 3$ because each observation has a three-dimensional location vector. At the Lth level of the K-D tree (where $L = 0$ is the root), the comparison key is the $[(L \bmod 3) + 1]$st component of the location vectors.

The time required to locate all the observations within a given distance of a three-dimensional reference point is proportional to $\log s$, where s is the number of observations on a given processor. The time required to build the K-D tree is proportional to $s \log s$. In our implementation of the LETKF, observation lookups take less than 5% of the total CPU time. The construction of the K-D tree, which needs to be done only once, takes only a few seconds. (On a 3.6-GHz Intel Xeon processor, for example, the required time is about 2 s for 45 000 observations.) More information about K-D trees and associated algorithms can be found on the Web or in any textbook on data structures (e.g. Gonnet and Baeza-Yates, 1990). Unlike conventional binary search trees, there is no simple procedure to remove an element from a K-D tree. Thus the quality control is applied to the observations *before* the K-D tree is constructed.

3.1.2. Load balancing algorithm. Once the observation lookup is completed, the grid points and the data from the associated local regions ($\bar{\mathbf{y}}^b_{[\ell]}$, $\mathbf{Y}^b_{[\ell]}$, $\bar{\mathbf{x}}^b_{[\ell]}$, $\mathbf{X}^b_{[\ell]}$, $\mathbf{y}^0_{[\ell]}$) are distributed between the processors. In principle, the LETKF is trivially parallelizable: one can assign every grid point to a different processor on a sufficiently large computer. However, the wall-clock time is limited by the time required (i) to distribute the data to the processors, (ii) to obtain the analysis ensemble at the centre grid point of the

Table 1. Processing time (in seconds) for the two slowest processors and for the fastest processor when the LETKF is run at 0Z on 40 processors with an ensemble of size k. The last row shows the total number of assimilated observations

k	20 Jan 2004	21 Jan 2004	22 Jan 2004
40	397/390/58	397/383/30	421/393/48
60	697/696/58	904/858/124	929/897/83
80	1570/1546/201	1518/1498/284	1581/1523/168
Number of	316 633	334 455	332 907
observations	316 633	334 455	332 907

Table 2. Approximate median time and time range (in milliseconds) required to compute $\mathbf{C}\mathbf{Y}^b$ as a function of the number of observations, s, for one local region for an ensemble of size $k = 60$. The timing resolution is 1 ms

s	Median time	Time range
30	1	<1–2
100	2	1–3
1000	14	13–22
5000	330	220–490
10 000	4000	1600–6200
15 000	7500	2900–7800

Table 3. Approximate median time and time range (in milliseconds) of the eigensolver DSYEVR for one local region as a function of the ensemble size k

k	Median time	Time range
40	2.8	1–10
80	11.2	3–30
100	15.1	5–40

most densely observed local region and (iii) to collect the analysis ensemble information from the different grid points. In our current implementation, the load-balancing strategy is simply to minimize the maximum time spent by any CPU to process its set of local regions (data transport times will be considered in a future version).

First, we estimate how many observations belong to every local region. Timing data collected from previous runs are used to estimate the corresponding processing time for each local region. Then, we find the division (into northern and southern slices) of the model grid that most nearly equalizes the total estimated running time on two processors. At every subsequent step, a similar bisection (either longitudinally or latitudinally) is performed on the remaining region with the largest estimated running time. The process continues until the number of regions equals the number of available processors. Table 1 shows the two largest and the slowest processing times for the LETKF on 40 processors, using observation datasets at 0000 UTC, for various ensemble sizes on three typical dates.

One advantage of the bisection strategy is that it assigns a geographically contiguous region to every processor, which minimizes the amount of observation data that must be replicated when adjoining local regions are assigned to different CPUs. Another advantage is that the observation and model grid data can be distributed in large chunks, which optimizes the efficiency of message-passing systems. The disadvantage is that the bisection strategy ignores the time required to distribute the data, which may be significant on a Beowulf architecture.

In the current implementation, the bisection strategy is applied only in the horizontal direction; each processor handles all the vertical levels in its assigned region. Depending on the number of available processors and the distribution of observations, the bisection algorithm may produce a horizontal region consisting of a single point. If a one-point subregion is the most expensive, then the algorithm halts.

In operational practice, it may be desirable to run the LETKF on a number of processors that is between two consecutive powers of 2 (e.g. on 96 or 384 processors). In this case, the globe can be divided initially into three latitudinal regions of approximately equal computational cost, and the bisection algorithm can be applied to each of the three subregions.

3.2. *The analysis component*

The implementation of the analysis step is straightforward. Given a subgrid assigned to a processor and the set of relevant observations for all of its grid points, one loops over every point of the subgrid and performs steps (iv)–(viii). Particular attention must be paid to the implementation of step (v). This step is by far the most expensive: it takes about 85% of the total CPU time in our implementation. Unless the number of observations in a local region is small, the calculation of $\mathbf{C}_{[\ell]}\mathbf{Y}^b_{[\ell]}$ takes most of the time. For good performance, a highly-tuned matrix multiplication routine is essential.[1] Table 2 shows some typical results for a 3.6-GHz Intel Xeon processor with a 2 MB secondary memory cache using the matrix multiplication routines in Intel's Math Kernel Library (version 8.0.1). The Intel library spawns threads to divide the computation among available processor cores, and this processing, along with cache misses, makes the running time a non-linear function of the local observation count, s.

To perform the eigensolution, we use the routine DSYEVR from version 3 of the LAPACK library (Anderson et al., 1999). This routine uses an iterative algorithm whose running time varies but is generally much less than that required to form $\mathbf{C}_{[\ell]}\mathbf{Y}^b_{[\ell]}$; see Table 3.

[1] Version 9.1 of the Intel Fortran compiler calls the appropriate BLAS routine in the Intel Math Library where available. The latter makes effi-

Table 4. Wall-clock time (in seconds) required to complete an analysis step at January 20, 2004 0000UTC with an ensemble of size $k = 60$ using varying numbers of processors. The total number of assimilated observations at this time is 334 455

Number of CPUs	Pre-analysis	Analysis	Post-analysis	Total
16	347	1383	43	1773
32	245	768	45	1060
64	248	500	45	793

3.3. Wall-clock time

Typical wall-clock times of a complete data assimilation cyle, including pre- and post-analysis and analysis, for a typical 0000UTC observational data set are 9–10, 14–16 and 24–27 m for a 40-, 60- and 80-member ensemble, respectively, using 40 processors on a 3.6 GHz Xeon cluster.[2] That is, doubling the ensemble size increases the total wall-clock time by about a factor of three. In a practical application, the burden of this increase of the computational time should always be weighed against the increase of accuracy that can be achieved by increasing the ensemble size (Section 5.4).

Table 4 shows the reduction in wall-clock time as the number of processors is increased for a 60-member ensemble and a fixed set of observations. The wall-clock time decreases with the number of processors due to the faster execution of the analysis component. In pre- and post-analysis, the faster execution of the spectral transforms and interpolation of the background ensemble to the time and location of the observations is offset by the increase of time spent on I/O operations and data transport. We expect to see a gain from using more processors in pre-analysis once satellite radiances are also assimilated. In that case, in addition to the interpolations, the observation operator also involves the calculation of the radiative transfer model.

The very reasonable wall-clock times achieved on our Beowulf cluster suggest that our data assimilation system should be widely applicable for research purposes. For instance, on our cluster, the GFS takes about 2.5 mins to generate one forecast on two CPU cores at T62L28 resolution. That is, a 40-member background ensemble can be generated in 5 mins using 40 processors. Adding this time to that needed to obtain the analysis, the estimated wall-clock time to complete a 0000UTC analysis cycle is about 15 mins. Since the number of observations is smaller at the other three analysis times, the four daily analyses can be calculated in less than 1 h. Thus, the generation of the analyses for a full season (90 d) takes less than 4 d. This computational speed is more than sufficient for the investigation of most research problems concerning global-scale atmospheric dynamics. Furthermore, since the computational cost scales linearly with the number of observations, the number of assimilated observations can be easily increased.

3.4. On the feasibility of an operational implementation

In an operational implementation, both the model resolution and the number of assimilated observations will be larger than what we have considered so far. In Appendix B, we provide rough estimates of the computational resources that would be needed for an operational implementation. Since numerical weather prediction centres already generate ensemble forecasts as part of their routine operational procedure, the need to generate background ensembles for the LEKTF does not necessarily increase the overall computational burden, especially if the resolution of the operational ensemble forecasts has been deemed sufficient for data assimilation purposes. (It has been a standard practice to generate operational 4D-VAR analyses and ensemble forecasts at resolutions somewhat lower than the resolution of the deterministic model forecasts.) In addition, when a variational data assimilation procedure is used, the operational centres generate the ensemble of analyses by a separate procedure (e.g. singular vectors, ensemble-based Kalman filter) at a significant additional computational cost. Finally, the LETKF is a much simpler algorithm than 4D-VAR, because it does not require the development of the adjoints of the model dynamics and the observation operator, so may require fewer human resources to maintain.

4. Experimental design

The goal of our numerical experiments is to compare the performance of the LETKF with a state-of-the-art operational data assimilation system. In our experiments, we assimilate all observations that were operationally assimilated at NCEP between January 1, 2004 0000UTC and February 29, 2004 1800UTC, with the exception of satellite radiances, but including all satellite-derived wind observations. Since our current implementation does not include a model or observational bias correction term, we also exclude all types of surface observations except for the surface pressure. The state-of-the-art reference data assimilation system used in this study is the Spectral Statistical Interpolation (SSI) of NCEP (Parrish and Derber, 1992; Environmental Modeling Center, 2003), one particular implementation of 3D-VAR. NCEP generously provided analyses that were generated with the SSI using the same version of the model (at T62L28 resolution) and the same observational data set as those we used in our LETKF experiments. In what follows, we refer to this data set as the *NCEP benchmark*. NCEP also made available their operational high-resolution (T254L64) analyses for the 2-month period used in this study. We use these operational analyses for verification purposes after truncating them to the T62L28

cient use of all local processing cores and provides excellent performance on Intel architectures.

[2]For comparison, the operational SSI takes approximately 3 m of wall-clock time using 56 processors in a 2.2 GHz Xeon cluster assimilating the same observations at the same model resolution. (Jeff Whitaker, 2007, pers. comm.).

resolution. The most important differences between the NCEP operational and benchmark systems are in their resolution and the use of satellite radiances in the operational system. [The same NCEP benchmark and operational data sets are also used in the study of Whitaker et al. (2007).]

4.1. Verification methods

The main challenge in verifying the performance of a data assimilation system for real observations is that the true value z^t of the verified meteorological parameter z is not known, which makes it difficult to obtain reliable estimates of the statistics for the analysis error $\bar{z}^a - z^t$, where $\bar{z}^a$ is the ensemble mean analysis of z. When an estimate z^v of the true state is used as a proxy for the true state, there are two sources of error in the estimate $E = \langle(\bar{z}^a - z^v)^2\rangle^{1/2}$ of the true root-mean-square error $T = \langle(\bar{z}^a - z^t)^2\rangle^{1/2}$, since

$$E = [T^2 + V^2 - 2C]^{1/2}, \tag{1}$$

where $V = \langle(z^t - z^v)^2\rangle^{1/2}$ is the root-mean-square error in the verifying data set and $C = \langle(\bar{z}^a - z^t)(z^v - z^t)\rangle$ is the covariance between the error in the verifying data set and the error in the state estimate. Here, the angled bracket $\langle\cdot\rangle$ stands for the average of the grid point values of the given meteorological field over space and/or time. Eq. (1) shows that errors in the verifying data lead to an overestimation of the true error, while a positive covariance between the errors of the verifying and verified data leads to an underestimation of the true error.

When the accuracy of two estimates of the same field is compared, it is usually wise to compare E^2 instead of E, since when the difference is taken between the two values of E^2, the term V^2 cancels out. Thus C is the more problematic component when comparing two analysis fields. In what follows, we use E^2 to compare fields of estimated errors for the LETKF and benchmark analysis-forecast cycles and to test the significance of the difference between the mean errors in the two cycles.

Let Δ^i be the difference between the values of E^2 for the two cycles at time i, where the mean in the calculation of E^2 is taken only over space. To test the statistical significance of the difference between the estimated errors of the two cycles, we apply a two-sample t- test for correlated data, as described in Wilks (2006), to the time series of Δ^i, $i = 1, \ldots, n$. In our case, the total sample size n is 96 (twice-daily forecasts for 48 d). The test involves calculating the *effective* sample size, $n' \leq n$, based on the assumption that the random variable $\Delta^i, n = 1, \ldots, n$, describes a first-order autoregressive process. Under this assumption, the effective sample size n' can be estimated by

$$n' \approx n(1 - r_1)(1 + r_1)^{-1},$$

where the auto-correlation coefficient r_1 is computed by

$$r_1 = \frac{\sum_{i=1}^{n-1}\left[\left(\Delta^i - \bar{\Delta}_-\right)\left(\Delta^{i+1} - \bar{\Delta}_+\right)\right]}{\left[\sum_{i=1}^{n-1}\left(\Delta^i - \bar{\Delta}_-\right)^2 \sum_{i=2}^{n}\left(\Delta^i - \bar{\Delta}_+\right)^2\right]^{1/2}}. \tag{2}$$

Here, $\bar{\Delta}_- = (n-1)^{-1}\sum_{i=1}^{n-1}\Delta^i$ and $\bar{\Delta}_+ = (n-1)^{-1}\sum_{i=2}^{n}\Delta^i$. If r_1 were zero, n' would equal n, but as the auto-correlation increases, n' decreases. We find that, depending on the atmospheric level and verification variable, the effective sample size n' is 25–50% smaller than the sample size n for the results presented in Section 5. In what follows, we regard the difference between the performance of the LETKF and benchmark as statistically significant when the null hypothesis, that the true value of the time mean $\bar{\Delta} = n^{-1}\sum_{i=1}^{n}\Delta^i$ is zero, can be rejected at the 99% confidence level.

We use three different methods to compare the analysis and forecast errors for the LETKF and the NCEP benchmark. First, error statistics are computed by comparing the LETKF and NCEP benchmark analyses to the operational NCEP analyses. In general, this verification technique favors the NCEP benchmark analysis, since the benchmark and the operational analyses are obtained with the same data assimilation scheme, which may lead to a strong covariance between the errors in the two NCEP analyses [that is, it is expected that $C > 0$ in eq. (1)]. On the other hand, a strong positive correlation is less likely in regions where the LETKF and the benchmark systems assimilate very few observations and the verifying analysis assimilates many observations. The most important such region is the Southern Hemisphere, where satellite radiances are known to have a strong positive impact on the quality of the operational analyses.

The second approach is to compare 48-h forecasts started from the LETKF and benchmark analyses to the verifying analyses. This approach reduces the effect of correlations between the benchmark and verifying analyses and is expected to provide more reliable comparison results in and downstream of regions where large numbers of observations have been assimilated into the verifying analyses during the previous 48-h period. While this approach reduces the effects of correlated errors, we can still expect it to be more favourable for the benchmark.

Our third approach is based on comparing short-term forecasts to radiosonde observations. More precisely, the root-mean-square error is estimated by comparing the z component of $H(\mathbf{x}^f)$ with observations z^v of the verified field at the observational locations, and averages are calculated over observational locations instead of grid points within the verification region. Here, $\mathbf{x}^f$ is a forecast started from the analysis $\bar{\mathbf{x}}^a$. While this approach has the advantage that the verified and the verification data sets are truly independent, thus providing a neutral verification approach for the comparison of the two systems, it has the significant disadvantage that the radiosonde network is strongly inhomogeneous, and the verification results reflect forecast accuracy over highly populated areas of the globe (Fig. 1).

4.2. LETKF parameters

In what follows, we provide results for the following choice of the LETKF parameters.

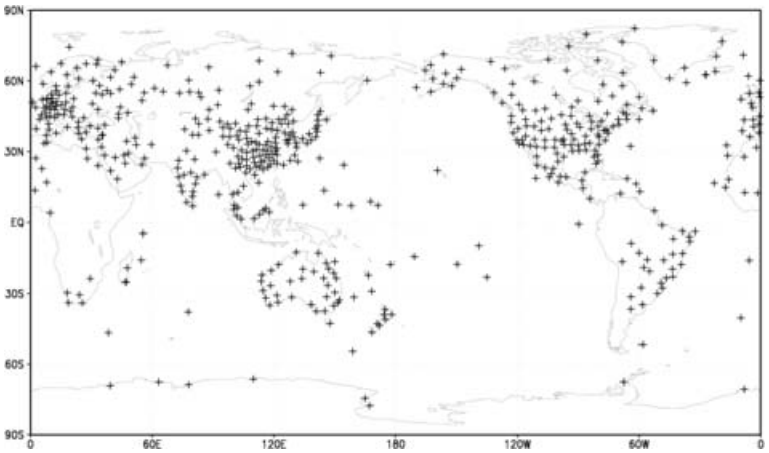

Fig. 1. Location of the radiosonde observations on a typical day at 0000UTC at the 300 hPa level. The number of radiosonde observations at the mandatory levels changes little between the surface and the 50 hPa pressure level.

(i) The ensemble has $k = 60$ members.

(ii) In the horizontal direction, observations are considered from an 800-km radius neighbourhood of the grid point at which the state is to be estimated.

(iii) The observations have equal weight [$\mu = 1$, see step (iv) of the LETKF algorithm] within a 500-km radius of the grid point, beyond which the weight of the observations, μ decreases linearly to zero at 800 km.

(iv) In the vertical direction, observations are considered from a layer around the grid point, and this layer has a depth of 0.35 scale height between model levels 1 and 15 (below sigma = 0.37 2), above which the depth gradually increases to 2 scale heights at the top of the atmosphere (defined by sigma = 0.00 3). Here, the scale height is defined as the vertical distance in which the surface pressure decreases by a factor of e, that is, the unit-scale-height deep vertical layer is defined by $\ln(\sigma_1/\sigma_2) = 1$, where σ_1 is sigma at the bottom of the layer and σ_2 is sigma at the top of the layer.

(v) The covariance inflation coefficient ρ tapers from 1.25 at the surface to 1.2 at the top of the model atmosphere in the SH extratropics and from 1.35 to 1.25 in the NH extratropics, while ρ changes smoothly throughout the tropics (between 25°S and 25°N) from the values of the SH extratropics to the values of the NH extratropics.

4.3. Data selection strategy

To analyse the surface pressure, we use all surface pressure observations from the local region and all temperature and wind observations between model levels 2 (sigma = 0.98 2) and 5 (sigma = 0.91 6). The virtual temperature and the two components of the wind vector are analysed at all model levels. To obtain the analysis of these variables, we assimilate temperature and wind observations, and when the state is estimated for the lower 15 model levels, the surface pressure observations are also added to the assimilated data set.

We do not analyse the humidity, liquid water content, or ozone mixing ratio model variables; the values of these variables are simply copied from the background ensemble members to the respective analysis ensemble members. We do not prepare a surface analysis either; all background ensemble members are obtained by using the operational NCEP surface analysis to initialize surface variables like soil temperature, sea surface temperature, snow cover, etc. This approach is suboptimal insofar as it underestimates the background error variance near the surface, which may explain the need for a stronger inflation of the variance closer to the surface. In the future, we hope to eliminate this potential problem by incorporating the surface data assimilation process into the LETKF system.

4.4. Adjustment of the surface pressure observations to the model orography

Due to the differences between the true and the model orographies, the surface pressure observations must be adjusted to the model surface. To make this adjustment, we follow the procedure described by Whitaker et al. (2004): the difference between the real and the model orographies at the observational location is calculated, then the hydrostatic balance equation is applied to calculate the pressure adjustment. The vertical mean temperature, which is needed for the evaluation of the hydrostatic balance equation for the layer between the real and model orography, is obtained by using the surface temperature observation at the location of the surface pressure observation and assuming that the lapse rate is 0.65 K/100 m in that layer. The observational error is also modified accordingly, assuming that the uncertainty in the estimation of the lapse rate is 0.3 K/100 m. When the difference between the two orographies is more than 600 m, the observation is discarded.

4.5. Quality control of the observations

We perform a simple quality control of each observation: an observation is rejected when the difference between the observed value and the background mean is at least five times larger than

both the ensemble spread (standard deviation of the ensemble) and the assumed standard error of the observation.

4.6. Generation of the initial background ensemble

We randomly select operational NCEP analyses from the period between January 1, 2004 and February 29, 2004 to define the initial background ensemble members. This choice has the disadvantage that, due to the relative closeness of the analysis times, the initial ensemble members are not as independent as they would be if chosen from a climatological archive. On the other hand, this choice has the advantage that the necessary data are easily available to us and would also be easily available for an operational implementation. For this choice of the initial ensemble members, the analysed values of the different meteorological fields settle after a 4–5-d initial spin-up period. To be conservative the verification statistics, we exclude all data associated with the analysis from the first 10 d of cycling.

5. Results

5.1. Verification against analyses

First, the LETKF and benchmark analyses are compared to the operational NCEP analyses (Fig. 2). Results are shown only for the SH extratropics, since (i) in the tropics, the difference between the accuracy of the two analyses is negligible except for the stratosphere, where the LETKF analyses are more similar than the benchmark to the operational analyses and (ii) in the NH extratropics, we cannot expect to obtain fair comparison results due to the strong similarity between the observational data sets used in the two systems that are compared and the operational system. (In the SH extratropics, the NCEP benchmark and the LETKF assimilates only a small portion of the the operationally assimilated observations.)

As seen from Fig. 2, the LETKF analyses are generally more similar than the benchmark analyses to the operational analyses. The only exceptions are the geopotential height and wind below the 700 hPa level. Based on the available information, it is impossible to decide with certainty whether in these regions the benchmark analyses are truly more accurate than the LETKF analyses or whether the correlation between the errors in the benchmark analysis and the operational analysis are simply stronger than the correlations between the errors in the LETKF analysis and the operational analyses.

We suspect that the latter explanation is more likely based on Fig. 3, which compares 48-h forecasts with the operational analyses: in the SH extratropics the LETKF forecasts are more similar to the operational analyses throughout the entire depth of the model atmosphere. Of course, it is also possible that this increased similarity near the surface is due to the influence of the stronger similarity in the upper layers at the analysis time. Nevertheless, we can conclude with high confidence that the

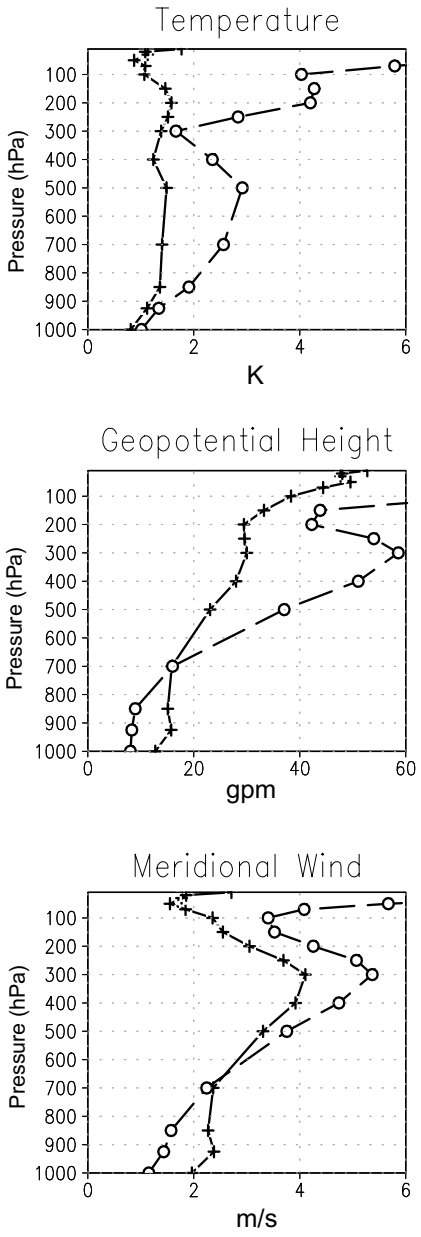

Fig. 2. Root-mean-square error, E, of the LETKF analyses (crosses connected by solid lines) and the NCEP benchmark analyses (open circles connected by dashes) in the SH extratropics. E is computed using the operational NCEP analyses as proxy for truth, z^v. The averages are taken over all model grid points south of 20°N and over all analyses times between January 11, 2004 0000UTC and February 29, 2004 1800UTC.

LETKF analyses lead to more accurate 48-h forecasts than the benchmark analyses in the SH extratropics. In the NH extratropics, the advantages of the LETKF system are less obvious: above 100 hPa, the LETKF provides more accurate predictions of all variables, while below that level, the benchmark shows

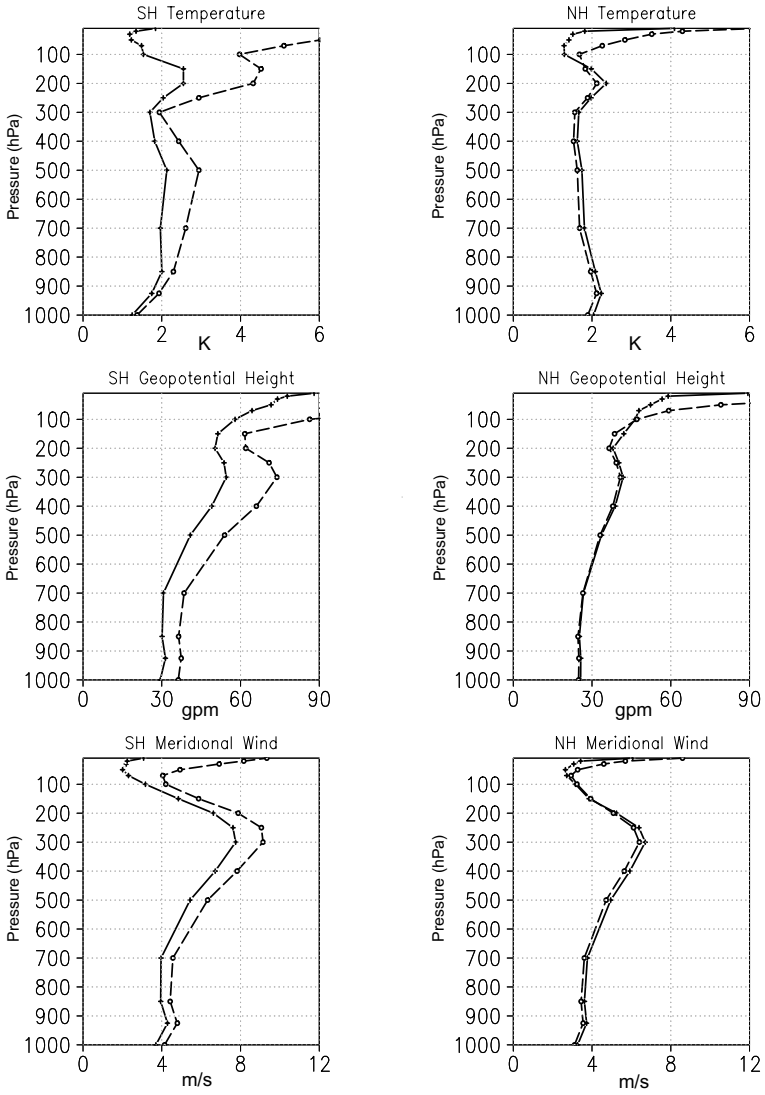

Fig. 3. Root-mean-square error, E, of the 48-h LETKF forecasts (crosses connected by solid lines) and the 48-h NCEP benchmark forecasts (open circles connected by dashes). E is computed using the operational NCEP analyses as proxy for truth, z^y. Left panels show the results for the SH extratropics, while right panels show the results for the NH extratropics. The averages are taken over all model grid points south of 20°N in the SH and over all model grid points north of 20°N in the NH, and over all forecasts started between January 11, 2004 0000UTC and February 27, 2004 1800UTC. (The last 2 d of February are not included in the verification, because the verifying data set was not available beyond the end of February.)

slightly better agreement with the operational model. Although the differences between the forecast errors are small at most levels, these differences are statistically significant except for the geopotential height below 300 hPa and the wind between 200 and 100 hPa.

5.2. *Geographical distribution of the forecast errors*

First we compare the 48-h LETKF forecasts with the verifying analyses (Fig. 4). The errors are the largest in the intersections of the regions of lowest observation density and the regions of the extratropical storm tracks. In particular, the errors are the largest in the oceanic region between, and east of, Cape Horn and the Antarctic Peninsula. In apparent contrast, in Szunyogh et al. (2005) we found that the LEKF analyses and short-term forecasts were the most accurate in the extratropical storm track regions. There are two important differences between the design of the two experiments, which may explain this important change in the quality of the short-term forecasts in the extratropical storm track regions. While the previous paper studied the perfect-model scenario, assimilating simulated observations that were homogeneously distributed in space, the present study is based on highly inhomogeneous observations of the real atmosphere.

To see whether the spatial distribution of the observations or the model error plays a more important role in changing the distribution of the short-term forecast errors, we have carried out experiments in which we assimilate simulated observations at the real observational locations. In this experiment, the time series of 'true' atmospheric states is generated by integrating the NCEP GFS starting from the NCEP operational analysis at January 1, 2004, 0000UTC. The random observational noise that is added to the 'true' states to create the simulated observations is generated by using the standard deviation values provided with each observation by NCEP. To simplify the generation of the

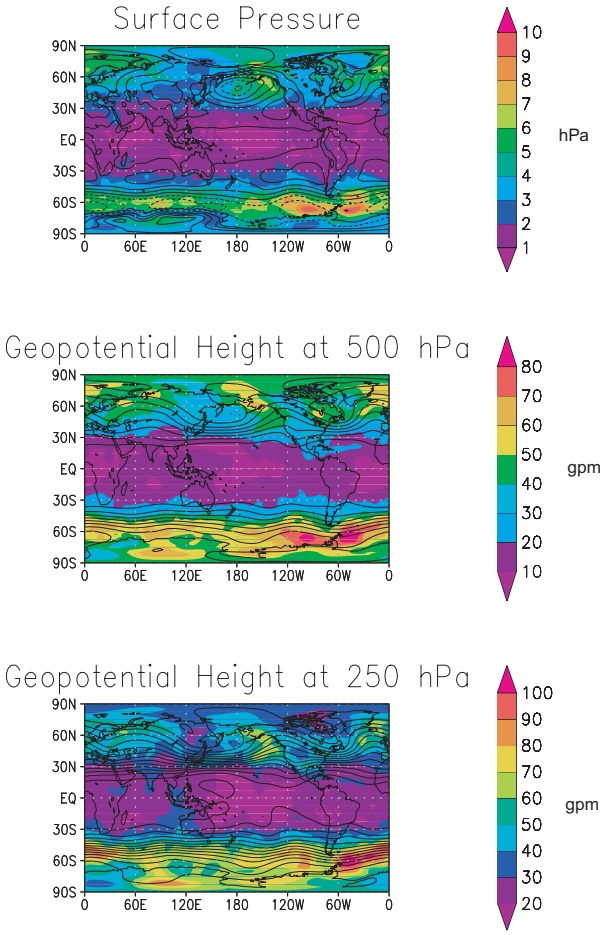

Fig. 4. Root-mean-square error, E, of the 48-h LETKF forecasts (color shades) for three different forecast variables. E is computed using the operational NCEP analyses as proxy for truth, z^v, and the average is taken over all forecasts started between January 11, 2004 0000UTC and February 27, 2004 1800UTC. Also shown are the time means of the related fields in the operational NCEP analyses (contours).

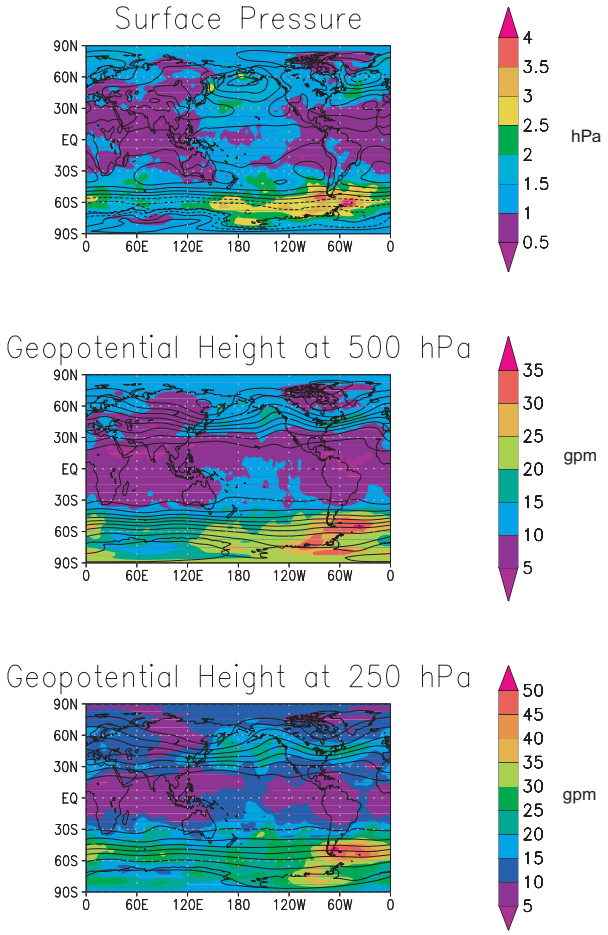

Fig. 5. Same as Fig. 4, except for simulated observations and using the known true states in place of z^v.

simulated observations, we assume that all observations are taken at the analysis time. The difference between the 48-h forecasts obtained when assimilating simulated observations at real observational locations and the true state is shown in Fig. 5. Based on the strong similarity between Figs. 4 and 5, we can conclude that the spatial error distribution shown in Fig. 4 mainly reflects the distribution of the observational density.

The ratio between the errors shown in Figs. 4 and 5 is about two, which indicates that imperfections of the model approximately double the forecast errors, but they do not dramatically change the spatial distribution of the forecast errors. However, the errors in the perfect-model experiment could likely be further reduced by reducing the variance inflation factor, which was tuned for the imperfect model. A more detailed comparison of Figs. 4 and 5 is not appropriate, as the state trajectory is different for the two experiments. In particular, although the cycling of the two data assimilation cycles is started from the same initial condition, the contours in Figs. 4 and 5 show that there are important differences in the temporal means along the trajectories (e.g. the stationary cyclone in the Bay of Alaska is much deeper in the real atmosphere than in the 'nature run').

We now turn our attention towards the comparison of the LETKF and the benchmark forecasts (Fig. 6). It is clear that on average, the better performance of the LETKF in the SH extratropics (depicted in Figs. 2 and 3) is predominantly due to better performance in the region between 150°W and 60°W, centred at around 75°S. This is the general area where the observational density is the lowest on Earth for the assimilated data set. Interestingly, the area where the LETKF has the largest advantage over the benchmark is to the west of the area where the LETKF forecast errors are the largest (Figs. 4 and 5). In the NH extratropics, the LETKF produces better forecasts in some areas, while in other regions, the NCEP benchmark is better. (This result is consistent with the spatially averaged results shown in Fig. 3.) In particular, the LETKF forecasts are more accurate in the southeast quadrant of North America. Considering the

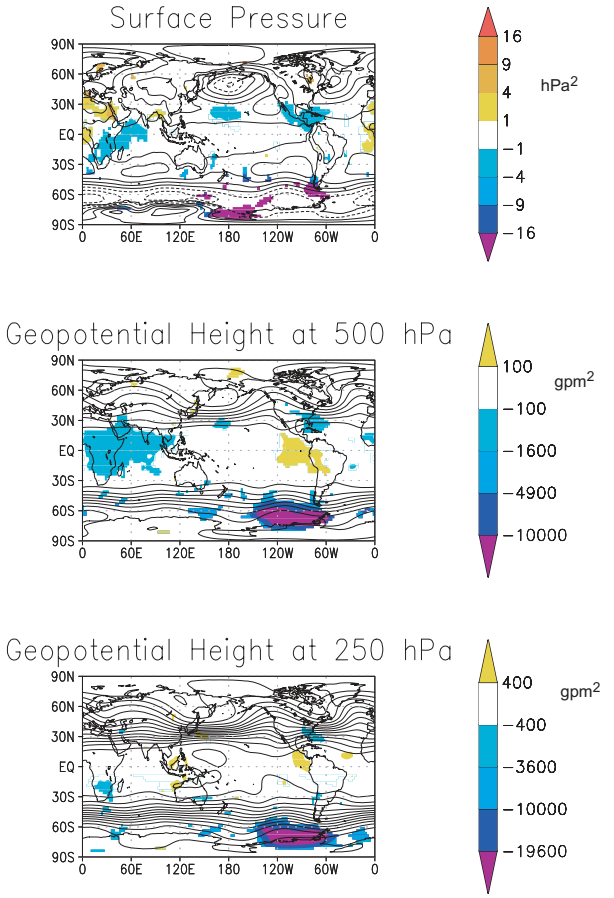

Fig. 6. Difference between the mean-square errors, E^2, of the 48-h LETKF and benchmark forecasts (color shades) for three different forecast variables. E^2 is computed using the operational NCEP analyses as proxy for truth, z^v, and the average is taken over all forecasts started between January 11, 2004 0000UTC and February 27, 2004 1800UTC. Negative values indicate regions where the estimated error is smaller for the LETKF forecasts. Results are shown only at locations where the difference between the two cycles is significant at the 99% confidence level. By lowering the significance level to 95 and 90%, we have found that the regions where the LETKF is superior become larger, but no important additional regions of significant differences between the two forecast cycles were revealed. Also shown are the time means of the related fields in the operational NCEP analyses (contours).

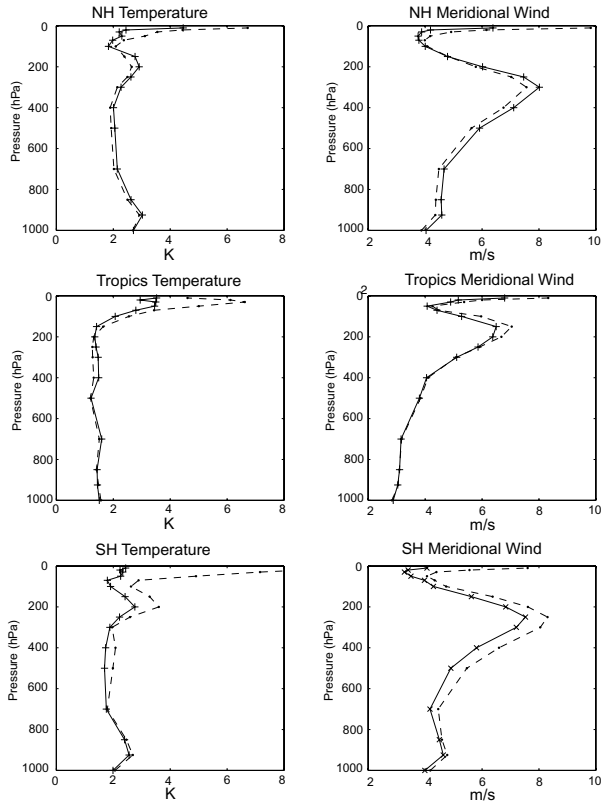

Fig. 7. Root-mean-square error, E, of the 48-h LETKF forecasts (crosses connected by solid lines) and the 48-h NCEP benchmark forecasts (open circles connected by dashes). E is computed using the radiosonde observations as proxy for truth, z^v, and the the averages are taken over all radiosonde observations and over all forecasts started between January 11, 2004 0000UTC and February 27, 2004 1800UTC. Shown are the results for the temperature (left panels) and for the meridional component of the wind (right panels). The differences between the two analysis cycles are statistically significant at the 99% confidence level except for the SH extratropics wind below 850 hPa, for the SH extratropics temperature below 700 hPa and at 300 hPa, for the tropical wind below 50 hPa, for the tropical temperature at 850, 500 and 200 hPa, and for the NH wind at 150 and 100 hPa.

westward (on average) propagation of the errors in the region, this result implies that the LETKF analyses are more accurate over the northeast Pacific and the western US.

The finding that the LETKF has the largest advantage over the SSI in regions where the observational density is the lowest is not unexpected. Earlier studies with simulated observations obtained similar results [e.g. Fig. 4 in Ott et al. (2004) and Fig. 4 in Hamill and Snyder (2000)]. Ensemble-based Kalman filters are expected to be more efficient than a 3D-VAR in propagating information into data voids and spreading information from sparse observations due to the use of a background error covariance matrix that adapts to the spatiotemporal changes of the flow and the observing network. Our results demonstrate that this important advantage of the ensemble-based Kalman filters is preserved when the simulated observations are replaced with observations of the real atmosphere. This finding suggests that the NCEP GFS model provides a representation of the atmospheric dynamics that is sufficiently accurate to provide useful information about the evolution of uncertainties in the short-term forecasts.

5.3. *Verification against radiosonde observations*

The results of the verification against radiosonde observations (Fig. 7) are in generally good agreement with those against the high-resolution operational analyses (Fig. 3). We believe that

the smaller advantage of the LETKF in the SH extratropics and the slight disadvantage of the LETKF in the NH extratropics are artefacts of the verification and are associated with underrepresentation of errors in data-sparse regions (Fig. 1): while in the verification against analyses, errors at different locations contribute with equal weight to the overall root-mean-square value, in the verification against observations, the errors in regions of high observation density contribute more to the overall root-mean square error than the errors in regions of low observation density. An interesting common property of the verification results against the analyses and observations is the big advantage of the LETKF in the stratosphere and above. Figure 8 suggests that this advantage of the LETKF over the SSI is due to the much smaller bias of the LETKF analyses in the upper layers.

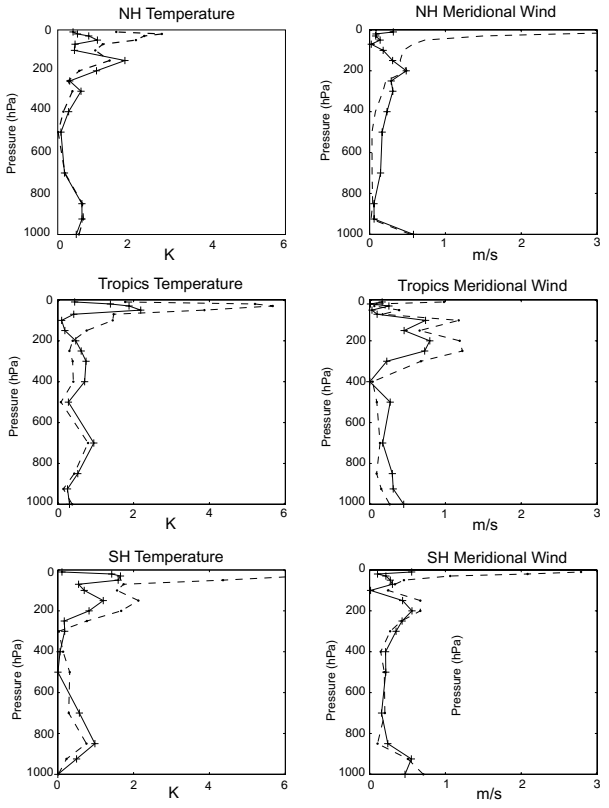

Fig. 8. Absolute value of the bias of the 48-h LETKF forecasts (crosses connected by solid lines) and the 48-h NCEP benchmark forecasts (open circles connected by dashes). E is computed using the radiosonde observations as proxy for truth, z^v, and the averages are taken over all radiosonde observations and over all forecasts started between January 11, 2004 0000UTC and February 27, 2004 1800UTC. Shown are the results for the temperature (left-hand panels) and for the meridional component of the wind (right-hand panels). The differences between the two analysis cycles are statistically significant at the 99% confidence level except for the SH extratropics wind below 850 hPa, for the SH extratropics temperature below 700 hPa and at 300 hPa, for the tropical wind below 50 hPa, for the tropical temperature at 850, 500 and 200 hPa, and for the NH wind at 150 and 100 hPa.

5.4. Sensitivity to the ensemble size and the definition of the local region

One key to optimizing the performance of an implementation of the LETKF on a particular model, in terms of accuracy of the state estimates, is finding a proper balance between the number of ensemble members and the size of the local regions. Although the LETKF algorithm does not require that all observations be selected from closed local volumes centred at the grid points, physical intuition suggests that whenever the observation operator is local, that is, whenever it depends on the model state only at the grid points that surround the observation, observations should be selected from a local volume.

A conceptual explanation for the importance of the balance between the number of ensemble members and the size of the local regions can be given along the following lines. The set of k background ensemble perturbations can be thought of as k basis vectors that represent the tangent space at the model state defined by the background mean. Our goal is to make an adjustment, through the analysis process, to the background mean in the tangent space to obtain the analysis. Thus, the number of ensemble members has to be sufficiently large to provide an accurate representation of all dynamically active degrees of freedom in the tangent space. The larger the local volume, the larger the number of ensemble members needed to properly represent the model dynamics in the local volume. This suggests that, on one hand, one should use the smallest possible local volume to minimize the size of the ensemble. On the other hand, to ensure the smoothness of the global analysis fields, the local volumes must be large enough to ensure that they overlap significantly with those at the neighbouring grid points and contain most of the same observations. The small grid point by grid point changes in the observational data sets guarantee the smooth spatial variation of the weight vectors $\{\mathbf{w}^{a(i)}\}$ (Ott et al. 2004). For instance, in the idealized experiments of Szunyogh et al. (2005), we found that when about 10% of the grid points were observed and the number of grid points in the local volumes was fixed in the vertical direction, a 40-member ensemble with a 5×5 grid point horizontal local region provided about the same accuracy as an 80-member ensemble with a 7×7 grid point horizontal local region. That is, preserving the accuracy required doubling the number of ensemble members when the number of grid points in the local volume was doubled. Using 3×3 horizontal grid points to define the local volume, however, always led to inferior results, regardless of the ensemble size.

The non-uniformity of the real observing network in space and time makes it harder to find the proper balance between ensemble size and local volume size. The parameters used in the experiments described so far have been determined by numerical experimentation, but we have found that the accuracy of the analysis is only weakly sensitive to the ensemble size and the selection of the local regions. For instance, we started our search for a reasonable choice of the parameters by first setting their

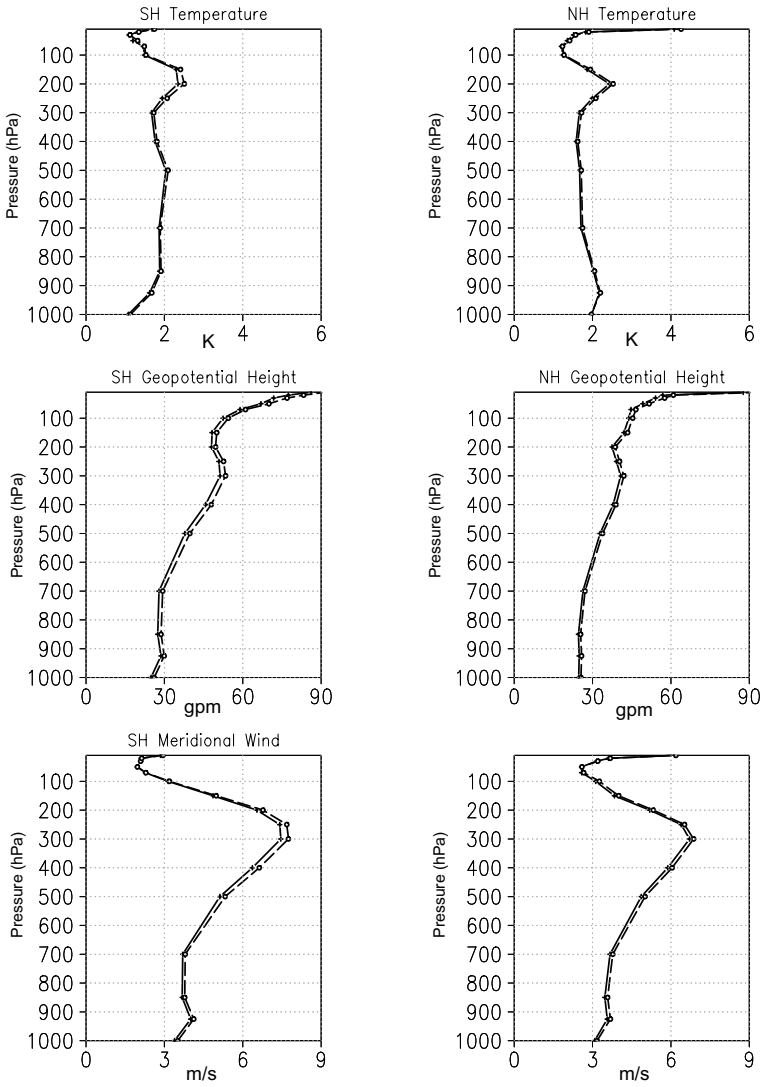

Fig. 9. Root-mean-square error, E, of the 48-h LETKF forecasts using an 80-member ensemble and 800 km radius local regions with tapering from 620 km (solid line and crosses) and the 48-h LETKF forecasts using a 40-member ensemble and 620 km radius local regions without tapering near the boundaries (dashes and open circles). E is computed using the operational NCEP analyses as proxy for truth, z^v. Left-hand side panels show the results for the SH extratropics, while right-hand side panels show the results for the NH extratropics. The averages are taken over all model grid points south of 20°N in the SH and over all model grid points north of 20°N in the NH, and over all forecasts started between January 11, 2004 0000UTC and January 27, 2004 1800UTC. (The last 2 d of February are not included in the verification, since the verifying data set was not available beyond the end of February.)

values to those we found best in Szunyogh et al. (2005): we set the ensemble size to $k = 40$ and the horizontal radius of the local volume to 620 km. This radius roughly spans 5 model grid points in the mid-latitudes, which is the value that we found optimal for a 40-member ensemble. This parameter set provides analyses (results not shown) that are almost as accurate, in the root-mean-square sense, as those we showed before for our standard choice of the LETKF parameters. We also carried out another experiment in which we increased the ensemble size to $k = 80$ members, but still used the 800-km regions and an observation localization coefficient μ that tapered from 1 to 0 between 500 and 800 km. The improvement that was achieved by increasing the ensemble size from 60 to 80 was hardly notable.

To illustrate the weak sensitivity of the accuracy to the localization parameters, we plot the vertical profile of the errors for the aforementioned 40- and 80-member ensembles (Fig. 9). The advantage of the 80-member ensemble is small but statistically significant. The price we pay for this small improvement is a threefold increase of the computational wall-clock time. Finally, we note that decreasing the ensemble size to 40, but keeping the 800 km radius local regions, led to a small, but more noticeable degradation, indicating that a 40-member ensemble is slightly smaller than what is needed to efficiently capture the degrees of freedom of the local model dynamics in an 800-km local region.

6. Summary

We have described an implementation of the LETKF data assimilation algorithm on the NCEP GFS model. Ours is one of the first successful attempts to assimilate a large number of observations of the real atmosphere for an extended period with an ensemble-based Kalman filter in an operational global weather prediction model. To our knowledge, the only other comparably successful attempts are those described in Houtekamer and Mitchell (2005) and Whitaker et al. (2007).

The performance of our implementation is assessed by comparing the results to those obtained with the operational SSI system of NCEP. We find that our data assimilation system provides a computationally efficient and accurate estimation of the atmospheric state. The accuracy of the LETKF is competitive with that of operational SSI algorithm. In particular, the LETKF provides more accurate analyses in regions of sparse observations, such as the SH extratropics and the stratosphere, and comparably accurate results in regions of dense observations. The computational efficiency of our implementation of the LETKF on the NCEP GFS makes it a potentially widely applicable analysis-forecast system for research purposes. For instance, the generation of four daily analyses at the resolution of the NCEP-NCAR reanalysis (T62L28), using 40 ensemble members, takes less than 4 d on a Beowulf cluster of forty 3.6 GHz Xeon processors. Similar computers are readily available to many academic research groups. We speculate (see Appandix B for details) that an implementation of our data assimilation system in an operational environment would be feasible at model resolutions that are somewhat lower than the highest resolutions currently used at the operational centres.

The LETKF algorithm has also been tested and compared with a serial ensemble-based data assimilation scheme by Whitaker et al. (2007). They found that the performance of the two ensemble-based schemes was very similar. Although Whitaker et al. (2007) used larger local volumes than we do, we believe that the most important difference between our implementation and the implementation of Whitaker et al. (2007) of the LETKF are in the definition of H and $\mathbf{R}$. While Whitaker et al. (2007) used H from the operational NCEP data assimilation system, we use our own H, which in its current form is presumably less sophisticated than the one used by Whitaker et al. (2007). While Whitaker et al. (2007) used the $\mathbf{R}$ from the operational NCEP data assimilation system, we used $\mathbf{R}$ from the operational NCEP data files. The main difference between the two realizations of $\mathbf{R}$ is that the NCEP data assimilation system increases the value of the prescribed observational errors at locations where the density of the observations is high in the vertical direction. This is done to reduce the detrimental effects of the representativeness errors in the observations (Whitaker et al., 2007). These differences and the data-thinning procedure employed in the serial scheme may explain why, in contrast to our results, Whitaker et al. (2007) found a small but statistically significant advantage of both ensemble-based schemes over the benchmark system in the NH extratropics.

Our goal is to further develop our data assimilation system for both research and operational numerical weather forecasting purposes. (Centro de Previsão de Tempo e Estudos Climáticos of Brazil has already made the decision to replace its current data assimilation system with the one we have been developing.) In addition to further improving the implementation of the observational operator and tuning the entries of $\mathbf{R}$, we are planning to integrate the Community Radiative Transfer Model (CRTM) of the Joint Center for Satellite Data Assimilation (JCSDA) into the system to support the assimilation of satellite radiances. We are also planning to add capabilities to the system to correct for model and observational biases with the techniques described in Baek et al. (2006).

7. Acknowledgments

The LETKF data assimilation system is the result of five years of theoretical work and code development. Initial funding for this project was provided by the W. M. Keck Foundation and the James S. McDonnell Foundation through a 21st Century Research Award. Our efforts have also been funded by the Army Research Office, by the NASA AIRS program, by the Office of Naval Research (Physics), and by the National Science Foundation (Grants #0104087 and PHYS 0098632). The implementation of the LETKF on the NCEP GFS has been carried out as part of a project for the comparison of different ensemble-based Kalman filter algorithms. This project has been partially supported by a NOAA THORPEX grant and was organized by the program manager of that program, Zoltan Toth. The NCEP benchmark analysis and forecast was prepared by Yocheng Song of NCEP and the operational observational data files were made us available by Richard Wobus. These data sets were provided by NCEP in the framework of the ensemble-based data assimilation comparison project. We have greatly benefited from discussions with members of the other teams participating in the comparison project. We are especially thankful to Jeff Whitaker of the Earth System Research Laboratory of NOAA for sharing with us his experience with another implementation of an ensemble-based Kalman filter on the NCEP GFS and for his helpful comments on an earlier version of the manuscript. The critical comments of the two anonymous reviewers helped us significantly improve the presentation of our results. E. J. K. gratefully acknowledges partial support from the NSF Interdisciplinary Grants in the Mathematical Sciences Program under grant number DMS-0408102.

8. Appendix A: Parallel implementation on two processors

As a pedagogical example, here we illustrate our computer implementation of the LETKF algorithm on a distributed-memory computer with two processors. Assume an ensemble of 40 solutions and that each local region includes all observations within 600 km of the centre grid point for now. For simplicity, also assume that all observations occur at the analysis time.

8.1. Pre-analysis component

To begin, Processor 1 reads the 6-h forecast files containing the spectral coefficients for the first 20 ensemble members, then transforms each of them to physical space to obtain $\mathbf{x}^{b(i)}$. Processor 2 proceeds similarly for ensemble members 21–40. Next,

Processor 1 reads the observation file and sends a copy of all the observations to Processor 2. Processor 1 applies the observation operator H to obtain the global background observation ensemble $\mathbf{y}^{b(i)}$, $i = 1, \ldots, 20$. Processor 2 proceeds similarly to obtain $\mathbf{y}^{b(i)}$, $i = 21, \ldots, 40$.

Suppose that Processor 1 is assigned all the model grid points from 0° longitude west to the 180° longitude (the 'Western Hemisphere' subgrid). Processor 2 gets all model grid points from 0° longitude east to 180° longitude (the 'Eastern Hemisphere' subgrid). Processor 1 sends the components of $\mathbf{x}^{b(i)}$, $i = 1, \ldots, 20$ belonging to the Eastern Hemisphere subgrid to Processor 2. Processor 2 sends the components of $\mathbf{x}^{b(i)}$, $i = 21, \ldots, 40$ belonging to the Western Hemisphere subgrid to Processor 1. Each processor now has all 40 ensemble solutions for the background grid in its respective hemisphere.

A similar procedure is used for the background observation ensemble: Processor 1 gets the Western Hemisphere components of $\mathbf{y}^{b(i)}$, $i = 21, \ldots, 40$ from Processor 2. Processor 1 also needs the components of $\mathbf{y}^{b(i)}$, $i = 21, \ldots, 40$ up to 600 km to the east of the 0° line, because they belong to the local regions centred at model grid points nearest 0°.

Processor 2 gets the Eastern Hemisphere components of $\mathbf{y}^{b(i)}$, $i = 1, \ldots, 20$ from Processor 1, plus components up to 600 km west of the 0° line. Analogous overlaps occur at 180° longitude and at the poles.

8.2. *Analysis component*

Each processor then performs steps (iv)–(viii) of the LETKF algorithm using its respective data. This yields a Western Hemisphere analysis subgrid on Processor 1 and an Eastern Hemisphere analysis subgrid on Processor 2.

8.3. *Post-analysis component*

Processor 1 sends the components of the Western analysis subgrid corresponding to ensemble solutions 21–40 to Processor 2. Processor 2 sends the Eastern portion of ensemble solutions 1–20 to Processor 1. Each processor transforms the global analysis grid for its respective ensemble solutions to spectral space and writes out the results.

9. Appendix B: Technical aspects of the computer program design

This appendix provides additional information about the program design (including where to compute the H operator) and the load balancing algorithm. It also provides preliminary performance results of the implementation on a dataset containing approximately 8 million observations.

9.1. *Memory requirements*

The present implementation requires four physical grids: the surface pressure, virtual temperature, and the u- and v-components of the wind. Except for the surface pressure, each variable is defined on $L = 28$ vertical levels. At T62 resolution, a full physical grid consists of $192 \times 94 = 18\,048$ horizontal points at each level. With 28 levels (denoted T62L28), the full physical grid occupies $192 \times 94 \times (3L + 1) = 1\,534\,080$ single-precision (32-bit) memory locations (i.e. about 5.83 megabytes for each ensemble solution).

Table 5. Memory requirements for reduced physical grids consisting of G horizontal points at selected resolutions. The 3-D grid includes the virtual temperature, u- and v-components of wind, mixing ratio, and ozone concentration at each of L levels in addition to the surface pressure

Resolution	G	Grid size	Memory
T62L28	13 072	1 843 152	7.03 MB
T126L28	52 208	7 361 328	28.08 MB
T126L64	52 208	16 758 768	63.93 MB
T170L42	90 984	19 197 624	73.23 MB
T170L64	90 984	29 205 864	111.41 MB
T254L64	205 004	65 806 284	251.03 MB

Because the spacing between horizontal grid points on a full physical grid varies significantly from the equator to each pole, the GFS provides for a reduced-resolution grid with approximately constant horizontal spacing. At T62 resolution, the number of horizontal grid points ranges from 192 at the 28 latitudes nearest the equator to 30 at the three latitudes nearest each pole; there are $G = 13,072$ horizontal grid points altogether.

Eventually, it will be necessary to include the mixing ratio of the tracers (e.g. humidity, ozone, liquid water content) at each physical grid point to accommodate satellite radiances, and, of course, higher model resolutions will be used. The full three-dimensional grid will occupy $G \times (5L + 1)$ memory locations, where G is the number of horizontal grid points at a particular resolution. Table 5 summarizes the memory requirements to process one ensemble solution using a reduced-resolution grid at various resolutions, assuming that each grid value is stored in a 32-bit (4-byte) single-precision word.

The LETKF can be implemented on either shared- or distributed-memory computers. On a shared-memory machine (such as the SGI Altix), the entire background grid must be kept in main memory. On a distributed-memory machine (such as a Beowulf cluster), only the portion of the background grid corresponding to the geographical region assigned to each processor needs to be available.

If the observation operator H is computed in the pre-analysis step, as in our implementation, then the variables at each grid point of the background grid can be overwritten with the analysis after the corresponding local region is processed. Thus, the total memory required to store the physical grid is given by the product of the last column of Table 1 and the ensemble size, k. If $k = 100$, for instance, then the model grid in the analysis step occupies

about 25 gigabytes at T254L64 resolution. This memory requirement is quite manageable on a shared-memory supercomputer. On a Beowulf cluster with p processors, the required memory is about $25/p$ gigabytes per CPU at this resolution.

In typical applications of the LETKF, the computation of H involves temporal as well as spatial interpolation. If the analysis is done every 6 h, for example, then forecast grids at 3, 4, . . . , 9 forecast hours, plus the previous analysis grid, are required if the time interpolation interval is restricted to no more than 1 h. Therefore, the model grids needed to compute H for one ensemble member occupy up to 7 times as much memory as stated in Table 5.

In our implementation, H is computed one ensemble solution at a time. The memory required for one ensemble background grid even at T254L64 resolution is considerably less than 2 GB. Moreover, if observations are binned into 1-h intervals, then the time interpolation requires only that the two background grids bracketing each 1-h interval be present in memory.

Of course, storage is required for the observations themselves. The minimum required amount is linearly proportional to the number of observations, n. The present implementation uses 11 words per observation: its location, time, type, value, standard error, pressure level, reference surface pressure, and miscellaneous flags related to quality control. (Additional data would be needed to account for the cross-correlation between observations.) The H operator requires an $2nk$ words of data: one for each of the k members of the ensemble background grid interpolated to each observation location and another for the adjusted vertical level, relative to each ensemble background forecast, of each observation.

9.2. *Where to compute H?*

In our implementation, H is computed in the pre-analysis step using the global physical grid for one or more ensemble solutions. However, depending on the number of observations and the complexity of codes that compute satellite radiances and similar observations, which may need access to large geographical regions of the model grid, in certain cases it may be advantageous to apply H at the beginning of the analysis step (using a grid that contains all ensemble solutions for a portion of the globe).

If H is computed as part of the pre-analysis step, then a copy of all the observations must be sent to each of the processors that perform the pre-analysis. Insofar as observations typically come from at most a few sources (one or more disk files or master processors), broadcasting all of them to many processors requires considerable bandwidth.

On the other hand, if H is computed at the start of the analysis step, then the appropriate portion of the background grid must be sent to each processor; when H involves time interpolation, this requires one grid at multiple intermediate forecast times, as noted above. Insofar as a given observation belongs to more than one local region in general, it must be copied to all processors that handle the relevant adjoining local regions.

The number of observations assimilated by the current data assimilation system is at least one order of magnitude less than the number of physical grid points in the forecast model. Unless the number of processors running the pre-analysis step is more than 100 or so, the cost of transporting model grid data is likely to exceed the cost of broadcasting all the observations to each processor. For this reason, it is preferable to incorporate the computation of H as part of the pre-analysis step.

The computation of H for satellite radiance observations requires a set of local regions that is not the same as the local regions used for the LETKF Fertig et al. (2007). If H is not computed in the pre-analysis step, then one must determine the appropriate background subgrid to send to each processor for the analysis step. This additional complexity is another argument in favour of putting H in the pre-analysis step.

9.3. *Scaling and performance on large data sets*

We now consider the performance of the LETKF on a data set that is more typical of those that would be used in the near future in an operational setting. The observations consist of the same data used elsewhere in the paper with the addition of high resolution temperature retrievals of The Atmospheric Infrared Sounder (AIRS) instrument soundings. (This data set was kindly provided by William Blackwell of the Massachusetts Institute of Technology). The dataset consists of 8.41 million observations, of which 7.88 million were retained after quality control checks. These data were assimilated into a T62L28 grid with 40 ensemble members and 32 processors using the localization criteria described in Section 5 (Fig. 10).

To assess the scalability of the algorithm, and to estimate the data transport requirements, we built a database consisting of the number of observations assimilated and the total processing

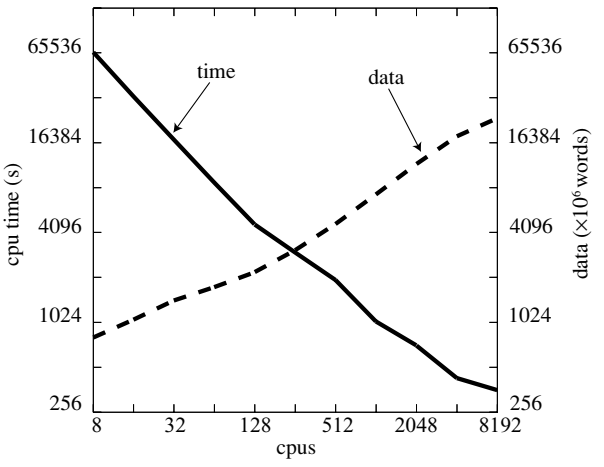

Fig. 10. Estimated maximum CPU times and data transport requirements for the LETKF on a dataset of 7.88 million observations assimilated into a 40-member ensemble of T62L28 grids.

time for each model grid point. The former is independent of how the model grid is partitioned among processors, and the latter is expected to be nearly so. We then ran the load balancing algorithm assuming that 2^k processors were available, $k = 3, 4, \ldots, 13$ (i.e. from 8 to 8192 CPUs). Using the database, we then estimated how long each processor would need to assimilate all the data in its local region as well as the amount of grid and observational data that would have to be transported from the other processors.

Figure 9 shows the estimated maximum time on any CPU and the total amount of data, in millions of 32-bit words, that need to be exchanged between the processors to complete the LETKF algorithm for this dataset. The actual CPU times were measured for 8, 16, and 32 processors and agree closely with the simulated results; the data transport requirements can be computed precisely.

The estimated maximum CPU time decreases inversely with the number of processors, p. The decrease goes almost exactly as $1/p$ for $p = 8, 16, \ldots 128$ and somewhat more slowly afterwards.

Each observation typically is assimilated in several local regions. The associated data must be copied to each processor that handles a local region to which the observation belongs. As the model grid is partitioned more finely, it becomes increasingly likely that a given observation must be copied to more than one processor. Thus, the total data transport requirements increase with the number of CPUs: from 2.23×10^6 32-bit words at $p = 128$ to 24×10^9 words at $p = 8192$.

The advent of dual- and quad-core commodity processors, however, can be expected to significantly reduce the data transport requirements. Our empirical measurements show that, on two cores, Intel's matrix multiplication routines almost exactly halve the time required to process each local region compared to a single-core implementation if the number of observations is more than a few hundred. This is because most of the CPU time is spent computing $\mathbf{C}_{[\ell]}\mathbf{Y}^b_{[\ell]}$, and the matrix multiplication routine in Intel's Cluster Math Library is threaded: each core multiplies two submatrices of approximately equal size. Therefore, one can obtain the wall-clock time performance of 64 CPUs but only need to transport data to 32 processors. If such performance gains can be extended to four cores, then 256 quad-core chips would be able to assimilate all the observations in about 1000 s (17 m) of wall-clock time, but only 3.1×10^9 words of data would need to be transported between them.

Of course, in operations, one would use higher resolution. A T254L64 grid contains nearly 36 times as many points as a T62L28 grid, and the total number of local regions would increase by the same amount.

9.4. Remarks on computing resources

It is clear that an operational implementation of the LETKF on a high-resolution grid will require substantial computing resources, particularly if the wall-clock time must be kept to a minimum, even though the algorithm can be expected to scale well among thousands of processors. Furthermore, one must generate the ensemble of background forecasts. On our cluster, the GFS takes about 2.5 m to generate one forecast on two CPU cores at T62L28 resolution. The required time for T254L64 probably would be 100 times greater. A 100-member background ensemble, therefore, would take roughly 50 000 CPU m to generate. However, it is not unreasonable to assume that, in a few years, an operational centre would be able to devote 1000 quad-core processors to the task and so be able to generate the background ensemble in less than 15 m of wall-clock time.

An extrapolation of the timing results above suggests that the LETKF, with 8 million observations and a T254L64 grid, would need about 9000 s of wall-clock time on 1024 quad-core chips. Most likely, some combination of data thinning and more processors will be needed to meet operational wall-clock time requirements.

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J. and co-authors 1999. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia. An online version is available at www.netlib.org/lapack/lug.

Baek, S.-J., Hunt, B. R., Kalnay, E., Ott, E. and Szunyogh, I. 2006. Local ensemble Kalman filtering in the presence of model bias. *Tellus* **58A**, 293–306.

Bishop, C. H., Etherton, B. J. and Majumdar, S. 2001. Adaptive sampling with the Ensemble Transform Kalman Filter. Part I: theoretical aspects. *Mon. Wea. Rev.* **129**, 420–436.

Evensen, G. 2007. *Data Assimilation. The Ensemble Kalman Filter*. Springer, Berlin.

Evensen, G. 2003. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dyn.* **53**, 343–367.

Fertig, E. J., Hunt, B. R., Ott, E. and Szunyogh, I., 2007. Assimilating nonlocal observations with a local ensemble Kalman filter. *Tellus* **59A** in press.

Fertig, E. J., Harlim, J. and Hunt, B. R. 2006. A comparative study of 4D-Var and 4D ensemble Kalman filter: perfect model simulations with Lorenz-96. *Tellus* **59A**, 96–100.

Global Climate and Weather Modeling Branch, Environmental Modeling Center 2003. The GFS atmospheric model. *Office Note 442*, http://www.emc.ncep.noaa.gov/officenotes/FullTOC.html. NCEP, NOAA/NWS, 5200 Auth Road, Camp Springs, MD 20742.

Gonnet, G. H. and Baeza-Yates, R. A. 1990. *Handbook of Algorithms and Data Structures* Chapter 6, 2nd Edition. Addison-Wesley, Reading, MA. 124–156.

Hamill, T. M. 2006. Ensemble-based data assimilation. In: *Predictability of Weather and Climate* (eds T. Palmer and R. Hagedorn) Cambridge University Press, Cambridge.

Hamill, T. M., Snyder, C., 2000. A hybrid ensemble Kalman Filter–3D Variational Analysis Scheme. *Mon. Wea. Rev.* **128**, 2905–2919.

Houtekamer, P. L. and Mitchell, H. L. 2005. Ensemble Kalman filtering. *Quart. J. Roy. Meteor. Soc.* **131**, 3269–3289.

Houtekamer, P. L., Mitchell, H. L., Pellerin, G., Buehner, M., Charron, M. and co-authors 2005. Atmospheric data assimilation with the ensemble Kalman filter: Results with real observations. *Mon. Wea. Rev.* **133**, 604–620.

Hunt, B. R., Kostelich, E. J. and Szunyogh, I. 2007. Efficient data assimilation for spatiotemporal chaos: a Local Ensemble Transform Kalman Filter. *Physica D*, **230**, 112–126.

Hunt, B. R., Kalnay, E., Kostelich, E. J., Ott, E., Patil, D. J. and co-authors. 2004. Four-dimensional ensemble Kalman filtering *Tellus* **56A**, 273–277.

Kuhl, D., Szunyogh, I., Kostelich, E. J., Gyarmati, G., Patil, D. J. and co-authors 2005. Assessing predictability with a local ensemble Kalman filter. *J. Atmos. Sci.* **64**, 1116–1140.

Lynch, P. and Huang, P. M. 1992. Initialization of the HIRLAM model using a digital filter. *Mon. Wea. Rev.* **120**, 1019–1034.

Oczkowski, M., Szunyogh, I. and Patil, D. J. 2005. Mechanisms for the development of locally low dimensional atmospheric dynamics. *J. Atmos. Sci.* **62**, 1135–1156.

Ott, E., Hunt, B. H., Szunyogh, I., Zimin, A. V., Kostelich, E. J. and co-authors 2004. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus* **56A**, 415–428.

Parrish, D. and Derber, J. 1992. The National Meteorological Center's spectral statistical interpolation analysis system. *Mon. Wea. Rev.* **120**, 1747–1763.

Szunyogh, I., Kostelich, E. J., Gyarmati, G., Patil, D. J., Hunt, B. R. and co-authors. 2005. Assessing a local ensemble Kalman filter: perfect model experiments with the National Center for Environmental Prediction global model. *Tellus* **57A**, 528–545

Whitaker, J. S. and Hamill, T. H. 2002. Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.* **130**, 1913–1924.

Whitaker, J. S., Hamill, T. M., Wei, X., Song, Y. and Toth, Z. 2006. Ensemble data assimilation with the NCEP Global Forecast System. *Mon. Wea. Rev.,* in press.

Whitaker, J. S., Compo, G. P., Wei, X. and Hamill, T. H. 2004. Reanalysis without radiosondes using ensemble data assimilation. *Mon. Wea. Rev.* **132**, 1190–1200.

Wilks, D. S., 2006. *Statistical Methods in the Atmospheric Sciences*, 2nd Edition. Academic Press, Burlington, MA, 143–146.