

Local Ensemble Transform Kalman Filter: An Efficient Scheme for Assimilating Atmospheric Data

John Harlim* and **Brian R. Hunt**

Department of Mathematics

and

Institute for Physical Science and Technology

University of Maryland, College Park, MD 20742

Abstract

We present an efficient variation of the Local Ensemble Kalman Filter (Ott et al. 2002, 2004) and the results of perfect model tests with the Lorenz-96 model. This scheme is locally analogous to performing the Ensemble Transform Kalman Filter (Bishop et al. 2001). We also include a four-dimensional extension of the scheme to allow for asynchronous observations.

1. Introduction

This paper describes an efficient method of implementing an Ensemble Kalman Filter (EnKF), which we call a Local Ensemble Transform Kalman Filter (LETKF). Unlike variational based data assimilation schemes, LETKF (or any EnKF scheme) represents the forecast uncertainty with an ensemble of forecasts. Ensemble based data assimilation is a natural approach since numerical weather prediction centers, such as NCEP and ECMWF, already employ ensemble forecasting operationally to assess the uncertainty in their forecasts. Using this information in the data assimilation procedure has the potential to provide better initial conditions, both for the main forecast and for the ensemble forecast.

*E-mail: jharlim@math.umd.edu

The goal of an EnKF is to generate at regular time intervals an *analysis* ensemble, that is, an ensemble of model states that reflects both an estimate of the true atmospheric state (through its mean) and the uncertainty of this estimate (through its spread). If successful, then applying the forecast model to the analysis ensemble at one time yields a background ensemble at the next time, representing a probabilistic estimate of the atmospheric state before new observations are taken into account. The analysis cycle is then completed by adjusting the background ensemble to better fit the new observations. In particular, the analysis ensemble mean is formed, in a sense, as a weighted average of the background ensemble mean and the observations, with the weights determined from the background and observation uncertainties. More precisely, the analysis ensemble mean is the model state that best fits the given background and observation probability distributions.

In a Kalman filter, these distributions are treated as Gaussian and the uncertainties are thus characterized by covariance matrices. The background and observation covariances determine the analysis covariance. The background covariance is computed as the sample covariance of the background ensemble, and thus to be consistent one must choose an analysis ensemble whose sample covariance matches the analysis covariance determined by the Kalman filter. One approach is to add to the analysis ensemble mean the columns of a square root of the analysis error covariance matrix. This type of analysis scheme is known as an Ensemble Square Root Filter (EnSRF) (Tippett et al. 2003) or a deterministic EnKF (Whitaker and Hamill 2002). In contrast, the early versions of EnKF (Evensen 1994; Burgers et al. 1996; Houtekamer and Mitchell 1998) are stochastic in the sense that they perturb the observations randomly in generating each ensemble member. Several variations of implementing EnSRF were introduced by exploiting the non-uniqueness of matrix square root, including the Ensemble Adjustment Kalman Filter (EAKF) of Anderson (2001), Ensemble Transform Kalman Filter (ETKF) of Bishop et al. (2001), and Local Ensemble Kalman Filter (LEKF) of Ott et al. (2002, 2004).

In LEKF, the analysis state is obtained by performing “local analyses” at each model grid point. Each local analysis accounts for only the observations within a local region surrounding the grid point, and therefore the choice of the size of the local regions should reflect the distance over which dynamical correlations represented by the ensemble are meaningful. The localization improves the efficiency of the scheme because each local analysis involves much less data than a global analysis, and the local analyses can be computed independently in parallel. More importantly, as pointed out in Anderson (2001), Hamill et al. (2001), and Houtekamer and Mitchell (2001), the localization suppresses spurious long-range correlations produced by a limited ensemble size.

For ensemble of size k , LEKF performs the analysis in a $(k - 1)$ -dimensional space E , using an orthonormal basis consisting of eigenvectors of the background covariance matrix. In this paper, we show that by performing the analysis in a k -dimensional space S with the background ensemble perturbations as the “basis”, LEKF computationally becomes more efficient. Formally,

each local analysis with this choice of basis is analogous to the ETKF and hence we call our scheme a Local ETKF (or LETKF). In contrast to the global ETKF, which requires the ensemble encompass the global uncertainty, in LETKF the ensemble need only encompass the uncertainty within each local region. We believe that this is feasible with an ensemble of moderate size due to the local low dimensionality of the atmospheric uncertainty observed in (Patil et al. 2001) and the results in (Szunyogh et al. 2005) for LEKF on the NCEP GFS model using fewer than 100 ensemble members. In our implementation of LETKF, we form the analysis ensemble perturbations from the symmetric square root of the analysis error covariance matrix, as opposed to the non-symmetric square root used by Bishop et al. (2001). In this respect, our approach is analogous to the Spherical Simplex ETKF of Wang et al. (2004).

In an operational setting, the initial conditions are generated every six hours, though many observations (mainly from the satellite and commercial flight observations) are available more frequently. Very limited computational time is allowed for each analysis (less than 10 minutes at NCEP). Given such constraints, an efficient algorithm becomes very important. One approach is the 4D-EnKF of Hunt et al. (2004). This four-dimensional extension of EnKF finds the analysis ensemble mean by fitting the linear combinations of the trajectories of the background ensemble to the asynchronous observations. This scheme may be thought of as an approximation to 4D-VAR (Talagrand 1981), the four-dimensional data assimilation technique used operationally by ECMWF; its main advantages is that it does not require computing the linear adjoint model for the (nonlinear) forecast model. In this paper, we introduce a 4D-LETKF that is analogous to 4D-EnKF but mathematically simpler.

The remainder of this paper is organized as follows. In Section 2 we present our global filter, this is an alternate formulation of ETKF, together with a step-by-step guide on how to efficiently implement this formulation. We discuss localization, our four-dimensional extension, and variance inflation in Section 3. We conclude this paper by showing some results obtained for the Lorenz-96 model (Lorenz and Emmanuel 1998) in Section 4, and give a brief summary in Section 5.

2. Global Filter Formulation

In this section, we first formulate the governing equations of LETKF in a global setting, then we give step-by-step instructions for applying them in an efficient manner. For a more detail derivation, see (Hunt 2005).

2a. Filter Derivation

The goal of an ensemble square root filter is to evolve an ensemble of model states that at any given time reflects, in the manner described below, both an estimate of the true system state and an uncertainty of that estimate. The ensemble members are evolved independently according to the model, except when new observations of the system state become available, at which point the entire ensemble is adjusted in tandem to reflect the new state estimate and (reduced) uncertainty dictated by the observations. This adjustment is called the “analysis”.

The inputs to the analysis are the forecast (or “background”) ensemble $\{\mathbf{x}^{b(i)} : i = 1, 2, \dots, k\}$ of size k , the observation operator $H : \mathbb{R}^m \rightarrow \mathbb{R}^s$ that maps the model space to the observation space, the observations $\mathbf{y}^o \in \mathbb{R}^s$, and the observation error covariance matrix $\mathbf{R} \in \mathbb{R}^{s \times s}$. The analysis assumes that the best available estimate to the system state, before the observations are taken into account, is the background mean

$$\bar{\mathbf{x}}^b = k^{-1} \sum_{i=1}^k \mathbf{x}^{b(i)}.$$

Define the $m \times k$ matrix of background ensemble perturbations \mathbf{X}^b , whose i th column is $\mathbf{X}^{b(i)} = \mathbf{x}^{b(i)} - \bar{\mathbf{x}}^b$. Then the background uncertainty in this state estimate is described by the background error covariance matrix

$$\mathbf{P}^b = (k - 1)^{-1} \mathbf{X}^b (\mathbf{X}^b)^T. \quad (1)$$

Thus \mathbf{X}^b is a matrix square root of $(k - 1)\mathbf{P}^b$. (Since \mathbf{P}^b can have rank at most $k - 1$, it accounts for uncertainty only in $k - 1$ directions in model space, which can be problematic if k is too small.)

The output of the analysis is the analysis ensemble $\{\mathbf{x}^{a(i)} : i = 1, 2, \dots, k\}$. The analysis mean

$$\bar{\mathbf{x}}^a = k^{-1} \sum_{i=1}^k \mathbf{x}^{a(i)}$$

and error covariance matrix

$$\mathbf{P}^a = (k - 1)^{-1} \mathbf{X}^a (\mathbf{X}^a)^T, \quad (2)$$

should represent respectively the estimate of the system state after the observations are assimilated, and uncertainty in this estimate. Here, the matrix \mathbf{X}^a of analysis ensemble perturbations, whose i th column is $\mathbf{X}^{a(i)} = \mathbf{x}^{a(i)} - \bar{\mathbf{x}}^a$, is an $m \times k$ matrix square root of $(k - 1)\mathbf{P}^a$.

The transformation from background ensemble $\{\mathbf{x}^{b(i)}\}$ to analysis ensemble $\{\mathbf{x}^{a(i)}\}$ is based on the transformation from the background mean $\bar{\mathbf{x}}^b$ and error covariance matrix \mathbf{P}^b to the analysis mean $\bar{\mathbf{x}}^a$ and error covariance matrix \mathbf{P}^a in the standard Kalman Filter, which assumes Gaussian observation errors, a linear model, and a linear observation operator. Ensemble Kalman Filters handle nonlinear model dynamics by evolving the analysis ensemble of model states from

one analysis time to the next and using the sample mean and covariance of the evolved ensemble as the background mean and covariance for the next analysis. If the time interval and analysis error covariance are sufficiently small, a Gaussian distribution whose mean and covariance match the ensemble's at the beginning of the time interval evolves approximately to a Gaussian distribution at the end of the time interval. So, we approximate the background distribution as a Gaussian. Later, we will describe how we handle nonlinear operator H ; for now, we assume a linear operator \mathbf{H} . Then, with a Gaussian background distribution and Gaussian observation errors, the analysis distribution will be Gaussian too.

Based on the standard Kalman Filter, the analysis ensemble mean is

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{K}(\mathbf{y}^o - \mathbf{H}\bar{\mathbf{x}}^b), \quad (3)$$

where

$$\mathbf{K} = \mathbf{P}^b \mathbf{H}^T (\mathbf{H} \mathbf{P}^b \mathbf{H}^T + \mathbf{R})^{-1} \quad (4)$$

is the Kalman gain matrix. The analysis error covariance is given by:

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^b. \quad (5)$$

To construct the analysis ensemble, one must then find an $m \times k$ matrix \mathbf{X}^a for which (2) is satisfied, and add $\bar{\mathbf{x}}^a$ to each column of \mathbf{X}^a .

The matrix $\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R}$, which is invertible as long as \mathbf{R} is, is an $s \times s$ matrix that changes from one analysis time to the next. Thus as written, computing \mathbf{K} takes at least of order s^3 floating-point operations. Computing \mathbf{P}^a and \mathbf{X}^a involves operations on $m \times m$ matrices. However, since \mathbf{P}^b has rank less than k , so do \mathbf{K} and \mathbf{P}^a , and they can be computed much more efficiently if k is small compared to m and s .

Since our ultimate goal is to transform the background ensemble $\{\mathbf{x}^{b(i)}\} = \{\bar{\mathbf{x}}^b + \mathbf{X}^b\}$ into an analysis ensemble $\{\mathbf{x}^{a(i)}\} = \{\bar{\mathbf{x}}^a + \mathbf{X}^a\}$, the most efficient method for doing so should avoid computing \mathbf{P}^b altogether. Ott et al. (2002, 2004) performed a reduced-rank analysis in the space E spanned by the background ensemble of perturbations $\{\mathbf{X}^{b(i)}\}$, using an orthonormal basis of E formed by the eigenvectors of \mathbf{P}^b . Then $\mathbf{P}^b = \mathbf{Q}\hat{\mathbf{P}}^b\mathbf{Q}^T$, where $\hat{\mathbf{P}}^b$ is a $(k-1) \times (k-1)$ diagonal matrix containing the nonzero eigenvalues of \mathbf{P}^b , and \mathbf{Q} is an $m \times (k-1)$ orthogonal matrix of the corresponding eigenvectors. The matrix $\hat{\mathbf{P}}^b$ represents the background error covariance in the chosen orthogonal coordinate system on E .

In this paper, we replace \mathbf{Q} by \mathbf{X}^b and avoid solving the eigenvalue problem. Now \mathbf{X}^b represents a transformation from an abstract k -dimensional space S onto the space E within the m -dimensional model space. This transformation maps the orthonormal basis vectors in S to the background ensemble perturbations. Thus S and \mathbf{X}^b represent a choice of coordinates on E that is non-orthogonal and over determined (each point in E has multiple coordinate representations

in S). The same coordinate system is used in the ETKF (Bishop et al. 2001), and the analysis described below is equivalent to ETKF aside from the choice of matrix square-root to form \mathbf{X}^a . The main convenience of doing the analysis in the space S is that the effective background error covariance matrix is simply

$$\tilde{\mathbf{P}}^b = (k - 1)^{-1} \mathbf{I}. \quad (6)$$

This matrix satisfies the relationship $\mathbf{P}^b = \mathbf{X}^b \tilde{\mathbf{P}}^b (\mathbf{X}^b)^T$, and thus transforms under \mathbf{X}^b into the correct covariance matrix. Furthermore, $\tilde{\mathbf{P}}^b$ is invertible, unlike \mathbf{P}^b , and this allow us to use a more convenient form of the Kalman filter equations involving the inverse of the background error covariance matrix.

To derive the LETKF equations, we begin by substituting (1) to (4), yielding

$$\mathbf{K} = (k - 1)^{-1} \mathbf{X}^b (\mathbf{X}^b)^T \mathbf{H}^T [(k - 1)^{-1} \mathbf{H} \mathbf{X}^b (\mathbf{X}^b)^T \mathbf{H}^T + \mathbf{R}]^{-1}. \quad (7)$$

At this point we can deal effectively with a nonlinear observation operator H . In (7), we see that every time the linearized operator \mathbf{H} appears, it is next to the matrix \mathbf{X}^b . The i th column of $\mathbf{H} \mathbf{X}^b$ is $\mathbf{H}(\mathbf{x}^{b(i)} - \bar{\mathbf{x}}^b)$, a first order Taylor approximation of $H(\mathbf{x}^{b(i)}) - H(\bar{\mathbf{x}}^b)$. Instead of linearizing H on the entire model space, we linearly approximate $\mathbf{H} \mathbf{X}^b$ by matrix \mathbf{Y}^b of background ensemble observation perturbations, whose i th column is $H(\mathbf{x}^{b(i)}) - \bar{y}^b$, where \bar{y}^b is the average over i of $H(\mathbf{x}^{b(i)})$. Notice that the sum of the columns of \mathbf{Y}^b is zero.

Next, using matrix identity¹

$$\mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \mathbf{R})^{-1} = (\mathbf{I} + \mathbf{A}^T \mathbf{R}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{R}^{-1}$$

with $\mathbf{A} = (k - 1)^{-\frac{1}{2}} \mathbf{Y}^b$, we have

$$\mathbf{K} = (k - 1)^{-1} \mathbf{X}^b [\mathbf{I} + (k - 1)^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1}.$$

Then from (5),

$$\begin{aligned} \mathbf{P}^a &= (k - 1)^{-1} (\mathbf{I} - \mathbf{K} \mathbf{H}) \mathbf{X}^b (\mathbf{X}^b)^T \\ &= (k - 1)^{-1} \mathbf{X}^b (\mathbf{I} - (k - 1)^{-1} [\mathbf{I} + (k - 1)^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b) (\mathbf{X}^b)^T. \end{aligned}$$

Using the identity $\mathbf{I} - (\mathbf{I} + \mathbf{B})^{-1} \mathbf{B} = (\mathbf{I} + \mathbf{B})^{-1}$ with $\mathbf{B} = (k - 1)^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b$ yields

$$\begin{aligned} \mathbf{P}^a &= (k - 1)^{-1} \mathbf{X}^b [\mathbf{I} + (k - 1)^{-1} (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1} (\mathbf{X}^b)^T \\ &= \mathbf{X}^b [(k - 1) \mathbf{I} + (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1} (\mathbf{X}^b)^T. \end{aligned}$$

¹To verify this identity, multiply both sides on the right by $\mathbf{A} \mathbf{A}^T + \mathbf{R}$ and observe that $\mathbf{A}^T \mathbf{R}^{-1} (\mathbf{A} \mathbf{A}^T + \mathbf{R}) = (\mathbf{A}^T \mathbf{R}^{-1} \mathbf{A} + \mathbf{I}) \mathbf{A}^T$.

Then

$$\mathbf{P}^a = \mathbf{X}^b \tilde{\mathbf{P}}^a (\mathbf{X}^b)^T, \quad (8)$$

where

$$\tilde{\mathbf{P}}^a = [(k-1)\mathbf{I} + (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1} \quad (9)$$

is the $k \times k$ matrix that represents the analysis error covariance in the space S . Then, (3) becomes

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b \tilde{\mathbf{P}}^a (\mathbf{Y}^b)^T \mathbf{R}^{-1} (\mathbf{y}^o - \bar{\mathbf{y}}^b), \quad (10)$$

where we have replaced $\mathbf{H}\bar{\mathbf{x}}^b$ with $\bar{\mathbf{y}}^b$; one could use instead $H(\bar{\mathbf{x}}^b)$. Writing

$$\mathbf{w}^a = \tilde{\mathbf{P}}^a (\mathbf{Y}^b)^T \mathbf{R}^{-1} (\mathbf{y}^o - \bar{\mathbf{y}}^b), \quad (11)$$

we have

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b \mathbf{w}^a,$$

where \mathbf{w}^a represents the analysis increment in S .

To construct an the analysis ensemble, choose

$$\mathbf{X}^a = \mathbf{X}^b \mathbf{W}^a, \quad (12)$$

where \mathbf{W}^a is a $k \times k$ matrix square root of $(k-1)\tilde{\mathbf{P}}^a$; that is, $\mathbf{W}^a (\mathbf{W}^a)^T = (k-1)\tilde{\mathbf{P}}^a$. Here the columns of \mathbf{W}^a represent the analysis ensemble perturbations in S . Then from (8), we have the required relationship (2). The i th member of the analysis ensemble is created by adding the i th column of (12) to (10)

$$\mathbf{x}^{a(i)} = \bar{\mathbf{x}}^a + \mathbf{X}^{a(i)}.$$

In order that the mean of the analysis ensemble be $\bar{\mathbf{x}}^a$, we need that the sum of the columns $\mathbf{X}^{a(i)}$ of \mathbf{X}^a is zero; that is, that $\mathbf{X}^a \mathbf{v} = \mathbf{0}$ where $\mathbf{v} = (1, 1, \dots, 1)^T$. Since $\mathbf{X}^b \mathbf{v} = \mathbf{0}$, it suffices that \mathbf{v} be eigenvectors of \mathbf{W}^a . As shown in (Wang et al. 2004), this is true for the symmetric square root

$$\mathbf{W}^a = ((k-1)\tilde{\mathbf{P}}^a)^{\frac{1}{2}},$$

but is not true for the choice of the matrix square root described in (Bishop et al. 2001); we will discuss this further in Section 4b. Thus the filter we described in this section is equivalent to ETKF using the symmetric square root of the analysis covariance in the k -dimensional space S .

2b. Efficient Computation of the Analysis

We now give a step-by-step description of how to implement the analysis described in the previous section with an eye toward minimizing the amount of computation. The inputs to the steps below are the m -dimensional vectors $\{\mathbf{x}^{b(i)} : i = 1, 2, \dots, k\}$, a nonlinear operator H from m variables to s variables, an s -dimensional vector \mathbf{y}^o , and an $s \times s$ matrix \mathbf{R} .

1. Form $\{\mathbf{x}^{b(i)}\}$ into an $m \times k$ matrix \mathbf{X} , and apply H to each column of \mathbf{X} to form an $s \times k$ matrix \mathbf{Y} . Average its columns to get the s -dimensional vector $\bar{\mathbf{y}}^b$, and subtract this vector from each column of \mathbf{Y} to get \mathbf{Y}^b . This requires k applications of H , plus $2ks$ floating-point operations. If H is an interpolation operator that requires only a few model variables to compute each observation, then the total number of floating-point operations for this step is proportional to ks , multiplied by the average number of model variables required to compute each scalar observation.
2. Average the columns of \mathbf{X} to get the $m \times 1$ vector $\bar{\mathbf{x}}^b$, and subtract this vector from each column of \mathbf{X} to get \mathbf{X}^b ; This step requires a total of $2km$ operations (additions and multiplications).
3. Compute the $k \times s$ matrix $\mathbf{C} = (\mathbf{Y}^b)^T \mathbf{R}^{-1}$. Since this is the only step in which \mathbf{R} is used, it may be most efficient to compute \mathbf{C} by solving the linear system $\mathbf{R}\mathbf{C}^T = \mathbf{Y}^b$ rather than inverting \mathbf{R} . In practice \mathbf{R} will be a block diagonal with each block representing a group of correlated observations. As long as the size of each block is relatively small, inverting \mathbf{R} or solving the linear system above will not be computationally expensive. Furthermore, many or all of the blocks that make up \mathbf{R} may be unchanged from one analysis time to the next, so that their inverses need not be computed each time. Based on these considerations, we estimate the number of operations required for this step in a typical application to be proportional to ks , multiplied by the average value of the cube of the block size of \mathbf{R} .
4. Compute the $k \times k$ matrix $\tilde{\mathbf{P}}^a = [(k-1)\mathbf{I} + \mathbf{C}\mathbf{Y}^b]^{-1}$. Multiplying \mathbf{C} and \mathbf{Y}^b requires $2k^2s$ operations, while the number of operations required is proportional to k^3 .
5. Compute the $k \times k$ matrix $\mathbf{W}^a = [(k-1)\tilde{\mathbf{P}}^a]^{1/2}$. Again the number of operations required is proportional to k^3 .
6. Compute the k -dimensional vector $\mathbf{w}^a = \tilde{\mathbf{P}}^a \mathbf{C}(\mathbf{y}^o - \bar{\mathbf{y}}^b)$ and add it to each column of \mathbf{W}^a , forming the $k \times k$ matrix \mathbf{W} . Computing the formula for \mathbf{w}^a from right-to-left, the total operations required are $s + 2sk + 2k^2$.
7. Compute $\mathbf{X}^b \mathbf{W}$ and add $\bar{\mathbf{x}}^b$ to each column. This requires a total of $2k^2m$ operations.

The output of the final step is an $m \times k$ matrix whose columns are the analysis ensemble members $\{\mathbf{x}^{a(i)}\}$. If k is reasonably large but still small compared to m and s , then the number of floating-point operations required is proportional to k^2s (multiplying \mathbf{C} and \mathbf{Y}^b in Step 4) or k^2m (multiplying \mathbf{X}^b and \mathbf{W} in Step 7), whichever is larger, If k is small enough, then more operations may be required for Step 3.

3. Localization, 4D Extension, and Variance Inflation

The fundamental difference between LETKF and ETKF is of course localization, and in this section we describe how to localize the approach of the previous section. We also describe a four-dimensional extension that assimilates asynchronous data, and a way to do variance inflation.

3a. Localization

To perform ensemble data assimilation for a global atmospheric model with an ensemble of moderate size, some form of localization is important. As we will see in Section 4b, even for relatively small models, localization can improve filter performance. See Hunt et al. (2004) for a discussion of different localization strategies.

Our localization is similar to that of (Keppenne 2000) and (Ott et al. 2002, 2004), in that the analysis is done separately and, if desired, in parallel for different local regions that cover the globe. In our formulation, the localization is relatively simple; for each grid point of the model, we choose a local subset of the global observations and apply the equations of Section 2 using only the local observations. To be more precise, we first perform Steps 1 and 2 of Section 2b globally (though in parallel implementation, it is possible to perform them locally if H is a local interpolation operator). Then for each model grid point, we truncate \mathbf{y}^o , $\bar{\mathbf{y}}^b$, and \mathbf{Y}^b to include only observations from a local region surrounding that point, and truncate $\bar{\mathbf{x}}^b$ and \mathbf{X}^b to include only the model variables for that grid point. Performing Steps 3 to 7 then yields an analysis ensemble $\{\mathbf{x}^{a(i)}\}$ of model states at the given grid point. After we do Steps 3 to 7 for each grid point, we have determined the global analysis ensemble.

In order to use the analysis ensemble members as initial conditions for the forecast model, it is essential that the results of the analysis be similar at nearby grid points. This can be ensured by choosing similar sets of observations for neighboring grid points. As long as the observation sets overlap heavily, the analyses will be similar. Our choice of the matrix square root is important here; the symmetric square root ensures that \mathbf{W}^a depends continuously on $\tilde{\mathbf{P}}^a$.

3b. 4D-LETKF

In this section, we describe four-dimensional extension to LETKF that handles asynchronous observations. The main idea of this method is to find the linear combination of the ensemble trajectories that best fits all of the observations collected between two analysis times. Our approach is analogous to that of (Hunt et al. 2004), but is simplified in the LETKF setting.

In that paper, the observations at different times are expressed as a function of the model state at the analysis time, using the background ensemble at both the analysis time and the

observations times in conjunction with the observation operator H . In the LETKF framework, we are able to simplify this approach by simply mapping the background ensemble into observation space at the observation times without referring to the background state at the analysis time.

Recall that in Section 2 we wrote the analysis mean as $\bar{x}^a = \bar{x}^b + \mathbf{X}^b \mathbf{w}^a$, where \mathbf{w}^a is determined from \mathbf{R} , \mathbf{y}^o , $\bar{\mathbf{y}}^b$, and \mathbf{Y}^b by (9) and (11). In essence, the coordinates of \mathbf{w}^a specify the linear combination of background ensemble states that best fit the data \mathbf{y}^o ; recall that $\bar{\mathbf{y}}^b$, and \mathbf{Y}^b are formed by mapping the background ensemble into observation space. So if the observations are not made at the analysis time, then they must redefine $\bar{\mathbf{y}}^b$, and \mathbf{Y}^b accordingly.

Specifically, suppose that we have data (t_l, \mathbf{y}_l^o) from various times t_l since the last assimilation. Let H_l be the observation operator for time t_l and let \mathbf{R}_l be the error covariance matrix for these observations. Let $\bar{\mathbf{x}}^b(t_l)$ and $\mathbf{X}^b(t_l)$ be the ensemble background mean and matrix of background ensemble perturbations at time t_l . We now form a combined observation vector \mathbf{y}^o by concatenating (vertically) the (column) vectors \mathbf{y}_l^o . The corresponding error covariance matrix \mathbf{R} is a block diagonal matrix with blocks \mathbf{R}_l (thus, we assume that observations taken at different times have uncorrelated errors). Form $\bar{\mathbf{y}}_l^b$ and \mathbf{Y}_l^b as in Section 2: apply H_l to each background ensemble state $\mathbf{x}^{b(i)}(t_l)$ to get vectors $\mathbf{y}_l^{b(i)}$, average those vectors to get $\bar{\mathbf{y}}_l^b$, and subtract $\bar{\mathbf{y}}_l^b$ from $\mathbf{y}_l^{b(i)}$ to get the columns of \mathbf{Y}_l^b . Let $\bar{\mathbf{y}}^b$ be the vertical concatenation of the column vectors of $\bar{\mathbf{y}}_l^b$, and let \mathbf{Y}^b be the matrix formed by stacking the matrices \mathbf{Y}_l^b vertically. Then \mathbf{Y}^b maps the k -dimensional analysis space S to the observation space containing \mathbf{y}^o .

Here then is how to modify the steps in Section 2 for this scenario of asynchronous data. First, perform Step 1 for each observation time t_l and combine the results as described in the previous paragraph to form $\bar{\mathbf{y}}^b$ and \mathbf{Y}^b . But perform Step 2 only at the analysis time and save the resulting \mathbf{X}^b and $\bar{\mathbf{x}}^b$ to use in Step 7.

In an efficient implementation of the steps above, it is probably best to store the blocks \mathbf{R}_l of \mathbf{R} separately from each other and sum over l at appropriate places. For example, the matrix \mathbf{C} defined in Step 3 has blocks $\mathbf{C}_l = (\mathbf{Y}_l^b)^T \mathbf{R}_l^{-1}$, and the matrix $\mathbf{C}\mathbf{Y}^b$ in Step 4 is then the sum over l of $\mathbf{C}_l \mathbf{Y}_l^b$. Likewise, the vector \mathbf{w}^a defined in Step 6 is the sum over l of vectors $\mathbf{w}_l^a = \tilde{\mathbf{P}}^a \mathbf{C}_l (\mathbf{y}_l^o - \bar{\mathbf{y}}_l^b)$.

3c. Variance Inflation

In order to compensate for the tendency of a small ensemble to underestimate uncertainty, it may be desirable to artificially inflate the background error covariance matrix \mathbf{P}^b before each analysis. (Or, one could instead inflate the analysis error covariance matrix \mathbf{P}^a after each analysis.)

From the formulation in Section 2a,

$$\tilde{\mathbf{P}}^a = \left[(\tilde{\mathbf{P}}^b)^{-1} + (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b \right]^{-1},$$

where $\tilde{\mathbf{P}}^a$ and $\tilde{\mathbf{P}}^b$ is defined by (9) and (6), respectively. The standard variance inflation approach

is to multiply the background ensemble perturbations \mathbf{X}^b by a constant factor $\sqrt{\rho} > 1$, which effectively multiplies \mathbf{P}^b by ρ . A similar result can be achieved more efficiently by leaving \mathbf{X}^b alone and multiplying (6) by $1/\rho$. Therefore, (9) is replaced by

$$\tilde{\mathbf{P}}^a = [(k-1)\mathbf{I}/\rho + (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1},$$

In other words, replace $(k-1)\mathbf{I}$ by $(k-1)\mathbf{I}/\rho$ in Step 4 of Section 2b. One can check that this change yields the same $\bar{\mathbf{x}}^a$ and \mathbf{X}^a in (10) and (12) as leaving (9) unchanged and replacing \mathbf{X}^b and \mathbf{Y}^b by $\sqrt{\rho}\mathbf{X}^b$ and $\sqrt{\rho}\mathbf{Y}^b$ respectively. For linear H , this is the same as inflating the background ensemble by $\sqrt{\rho}$ before applying H to form \mathbf{Y}^b , and even for nonlinear H the result is similar if ρ is close to one.

4. Simulation on Lorenz-96

4a. Experimental Design

The Lorenz-96 model represents an “atmospheric variable” x at m -equally spaced points around a circle of constant latitude. The j th component is propagated in time following differential equation:

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F \quad (13)$$

where $j = 1, \dots, m$ represent the spatial coordinates (“longitude”). This model is designed to satisfy three basic properties: it has linear dissipation (the $-x_j$ term) that decreases the total energy defined as $V = \frac{1}{2} \sum_{j=1}^m x_j^2$, an external forcing term F that can increase or decrease the total energy, and a quadratic advection term that conserves the total energy (i.e. it does not contribute to $\frac{d}{dt}V$). Following Lorenz (1996) and Lorenz and Emmanuel (1998), we choose the external forcing to be $F = 8$ and the number of spatial elements to be $m = 40$. We also use a fourth-order Runge-Kutta scheme for time integration of (13) with time step $\Delta t = 0.05$. With these parameters, the attractor has 13 positive Lyapunov exponents, with the leading Lyapunov exponent corresponding to a doubling time of 0.42 time units, and a Kaplan-Yorke dimension of 27.1 (Lorenz 1996). On the basis of doubling time, Lorenz suggested that 1 time unit of the model is roughly equivalent to 5 days in a global weather model. Thus, performing data assimilation every time step of our model integration corresponds roughly to performing it every 6 hours in a global weather model.

We perform all simulations in the perfect model scenario; that is, a long integration of an arbitrary initial condition is assumed to be the “true” state. Throughout this paper, we assume that the observational variables are the same as to the model’s and can be obtained at each model grid. In other words, the measurement function H is identity. We create the observation vector \mathbf{y}^o

by adding to the true state a random vector, where each coordinate is chosen independently with standard normal distribution. Hence, the observation error covariance matrix \mathbf{R} is the identity matrix. The initial background ensemble ($\mathbf{x}^{b(i)} : i = 1, 2, \dots, k$) is created by adding uncorrelated perturbation vectors to the true state. In fact, one may start with an arbitrary ensemble where each member is uncorrelated to the the true state.

In all of the results below, we measure the quality of the analysis at time t by calculating the Root-Mean-Square (RMS) difference between the true state and the analysis ensemble mean at time t , where $t = \Delta t, 2\Delta t, \dots, N\Delta t$. We then take the root-mean-square average of these differences over 10 different runs of $N=20,000$ analysis each. For the rest of this paper, we refer to this averaged quantity as RMSE. An RMSE value of greater than one implies that the distance from the true state to the analysis ensemble mean is no smaller on average than from the true state to the observation state with unit variance, and hence, one may trust the observations rather than performing analysis.

4b. Results

In this section, we verify and assess the sensitivity of LETKF and 4D-LETKF toward different parameter quantities. The parameters are number of model variables m , localization distance d , ensemble size k , and variance inflation coefficient ρ . By localization distance d , we mean that the analysis at a given grid point uses the observations from the $2d+1$ grid points centered at the analysis point. In 4D-LETKF, additional parameter will be introduced later. In the experiments below, we refer the *default parameter set* as fixing $m = 40$, $d = 6$, and $k = 10$. This value of d was found to be optimal for the given value of m and k (Ott et al. 2002, 2004).

In our first experiment, we are interested with how well LETKF performs compared to LEKF using default parameter set. Particularly, we plot the accuracy (or RMSE) of both schemes as functions of variance inflation coefficient. Figure 1 indicates that both schemes perform about equally well for $1.04 \leq \rho \leq 1.06$ with $\text{RMSE} \approx 0.21$, and diverge when $\rho < 1.04$.

In the second experiment, we examine the sensitivity of the local and global analyses under the variations of the ensemble member k and the model dimension m . In particular we fix ρ , d , and m ; and plot the RMSE as a function of ensemble size k . For the global analysis, we use all observations for the analysis at each grid point (essentially, $d = m/2$). We use $\rho = 1.04$ in all cases; as in Figure 1, slightly larger values did not significantly change the results. In the local analysis, we fix $d = 6$. Figure 2(a) indicates that if one doubles the model variables m from 40 (dashes) to 80 (solid), similar accuracy (shown by $\text{RMSE} \approx 0.21$) can be obtained by performing the local analysis with ensemble of size $k = 10$. In Figure 2(b), results for the global analysis is plotted in similar fashion as the local's in Figure 2(a) for ensemble of size 10 to 70 with interval 5. The results here show that to obtain an accuracy of $\text{RMSE} \approx 0.19$, an ensemble of size 20 sufficed when $m = 40$, but an ensemble of size at least 40 is needed for $m = 80$. Thus while

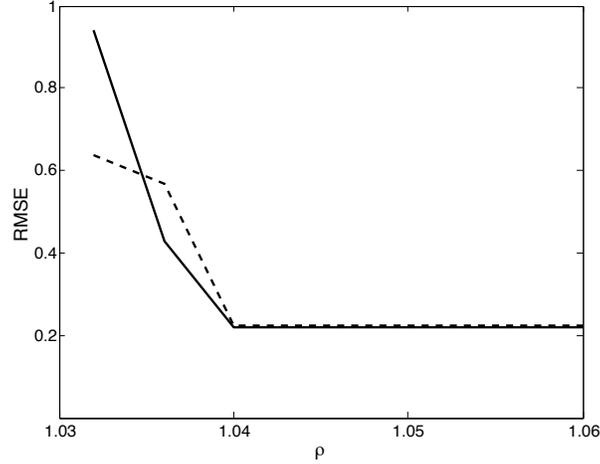


Figure 1: RMSE of the LEKF (solid) scheme and the LETKF (dashes) scheme as functions of variance inflation coefficient ρ . Differences for $\rho < 1.04$ are not significant; in these cases, both methods have RMS errors for some of the 10 runs, indicating that the inflation amount is insufficient.

the number of ensemble members required grows is proportional to system size for the global analysis, for the local analysis this number remains small as the system size grows. A similar result with LEKF was found in (Ott et al. 2004).

As we mentioned before, ensemble square-root filters generate the analysis ensemble by adding an analysis ensemble of perturbation to the analysis ensemble mean. The ensemble of perturbations is a square root \mathbf{X}^a of the scaled analysis covariance matrix $(k-1)\mathbf{P}^a$. In Section 2, we wrote $\mathbf{X}^a = \mathbf{X}^b \mathbf{W}^a$ where \mathbf{W}^a is a square root of $(k-1)\tilde{\mathbf{P}}^a$; that is, $\mathbf{W}^a (\mathbf{W}^a)^T = (k-1)\tilde{\mathbf{P}}^a$. We can then write

$$\mathbf{W}^a = \mathbf{U} \Sigma^{1/2} \mathbf{V} \quad (14)$$

where the i th column of $\mathbf{U} \in \mathbb{R}^{k \times k}$ is the eigenvector of $(k-1)\tilde{\mathbf{P}}^a$ corresponding to eigenvalue stored in the i th diagonal element of Σ , for $i = 1, \dots, k$; and \mathbf{V} can be any $k \times k$ orthogonal matrix ($\mathbf{V} \mathbf{V}^T = \mathbf{I}$). In all of the previous results, we used $\mathbf{V} = \mathbf{U}^T$ so that \mathbf{W}^a is symmetric. Wang et al. (2004) called this choice the Spherical Simplex ETKF. Now, we compare this choice of square root with the non-symmetric square root $\mathbf{W}^a = \mathbf{U} \Sigma^{1/2}$ ($\mathbf{V} = \mathbf{I}$), as was originally proposed by Bishop et al. (2001). In Figure 3, see that the symmetric square root (dashes) with $k = 20$ ensemble members converges with $RMSE \approx 0.19$ and the non-symmetric square root (solid) converges with $k = 40$ ensemble members and $RMSE \approx 0.25$. Thus, even for a global filter, the choice of the square root can have a significant impact on the results. In the case of local analysis, the choice of the symmetric square root is even more crucial since it ensures

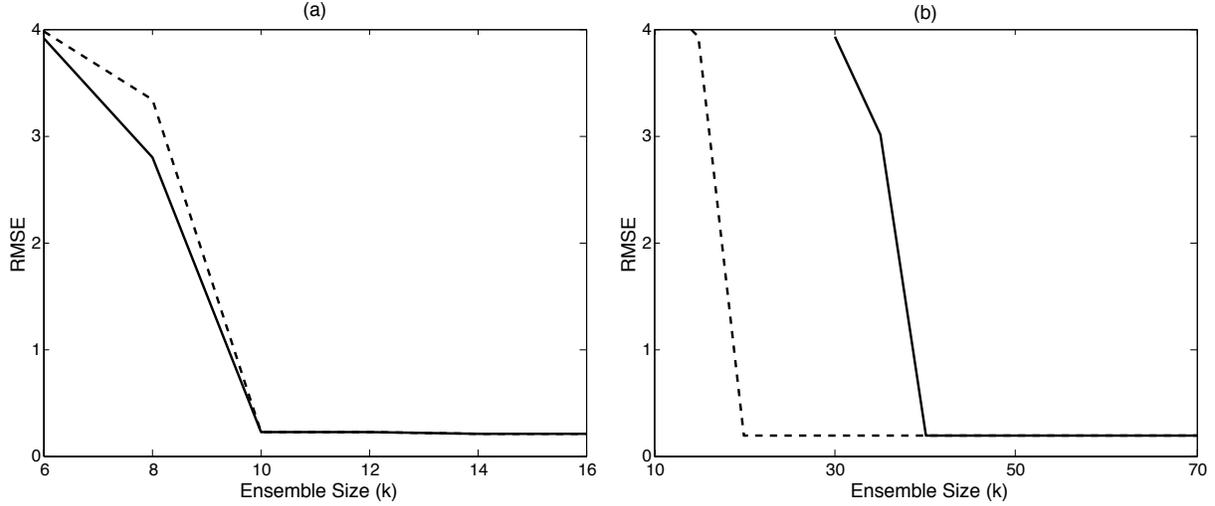


Figure 2: RMSE for LETKF as a function of ensemble size k : (a) for local analysis with $d = 6$ and (b) for global analysis. In both cases, the dashed curve is for $m = 40$ and the solid curve is for $m = 80$.

consistency between adjacent local analyses. As discussed in Section 3a, this is because it makes \mathbf{W}^a a continuous function of $\tilde{\mathbf{P}}^a$. By contrast, numerically computed eigenvectors can depend discontinuously on the input matrix, and thus $\mathbf{W}^a = \mathbf{U}\Sigma^{1/2}$ is not a continuous function of $\tilde{\mathbf{P}}^a$. Indeed, we found that the LETKF with this non-symmetric square root diverges even for $k = 12$ ensemble member and variance inflation $\rho = 1.30$.

In the last experiment, we validate the four-dimensional scheme described in Section 3b. Here, we compare the accuracy of the 4D-LETKF and LETKF as one varies the “steps per analysis” n . That is, we perform analysis every $n\Delta t$ time units, where $\Delta t = 0.05$ is the numerical integration time. For each analysis, our LETKF results ignore all observations at the non-analysis steps (times $l\Delta t$, where l is not a multiple of n) and use only the observations at the analysis time. On the other hand, 4D-LETKF uses all the observations collected since the previous analysis time. That is, for analysis at time $jn\Delta t$ it accounts observations at time $l\Delta t$ for $l = (j-1)n+1, (j-1)n+2, \dots, jn$. For $n = 1$, 4D-LETKF and LETKF are equivalent.

Figure 4 plots the RMSE of both the LETKF and 4D-LETKF as functions of number of steps per analysis n . In this simulation, we use our default values for model size ($m = 40$), localization distance ($d = 6$), and ensemble size ($k = 10$), and tune the variance inflation ρ so that the RMSE is minimized for each value of n . In LETKF, we obtained the lowest RMSE (solid) by setting $\rho = 1.04, 1.12, 1.24, 1.33$, and 1.65 for $n = 1, 2, \dots, 5$. In 4D-LETKF, we obtained the lowest RMSE (dashes) by setting $\rho = 1.04, 1.12, 1.24, 1.38$, and 1.75 for $n = 1, 2, \dots, 5$. Here ρ grows more

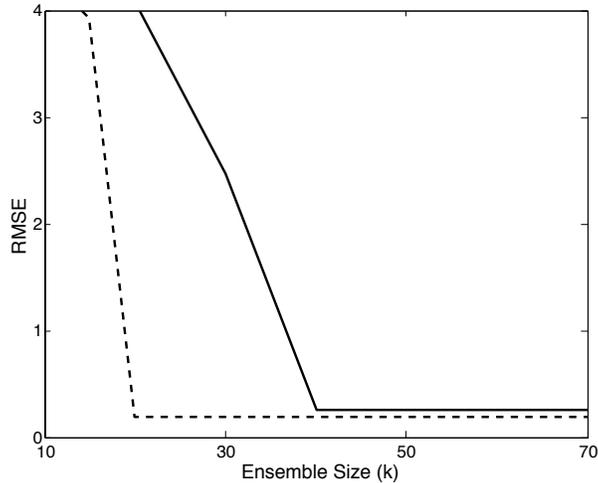


Figure 3: Plots of the RMSE for model size $m = 40$ as a function of ensemble size k for the symmetric square root (dashes) and non-symmetric square root (solid) $\mathbf{W}^a = \mathbf{U}\Sigma^{1/2}$. Here we used variance inflation $\rho = 1.04$ as before for the symmetric square root, but needed more inflation $\rho = 1.15$, to obtain convergence for the non-symmetric square root.

than linearly as a function of n , in contrast to the linear growth used in (Hunt et al. 2004). (In their experiment, they performed a global analysis (4D-EnKF) with a different type of “additive” variance inflation.) In Figure 4, we observe that the RMSE of LETKF increases almost linearly as the steps per analysis n increases up to five. The 4D-LETKF preserves its accuracy better as one increases n . Of course, this is because it uses more observations than LETKF for $n > 1$, but our main point is that 4D-LETKF uses them nearly as well as if the analysis were done every time step. The need for increasing variance inflation as n increases is primarily due to the increasing time between analyses for uncertainties not captured by the ensemble and the RMSE becomes worse as n gets large.

5. Summary

In this paper, we combine the local analysis suggested in (Ott et al. 2002, 2004) and the simplified formulation of the global spherical simplex ETKF of (Bishop et al. 2001) and (Wang et al. 2004). In our experiments with the Lorenz-96 model, we compared LEKF and LETKF and conclude that both schemes produce a similar level of accuracy. We also tested the importance of the local analysis and found as in (Ott et al. 2002, 2004) that localization allows one to maintain a constant ensemble size as the number of variables in the model increases. We also show

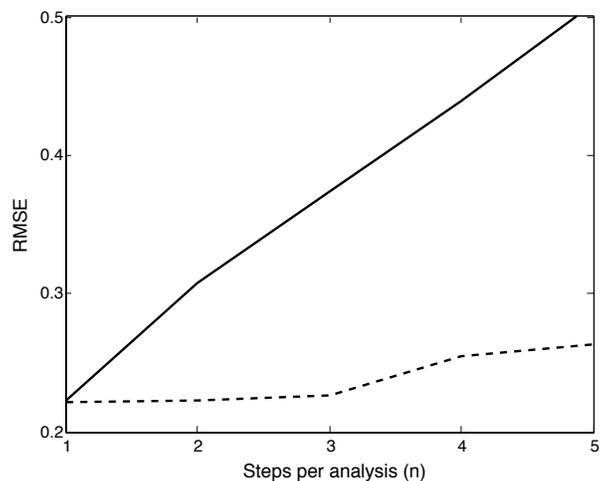


Figure 4: Plots of the RMSE as a function of the number of steps per analysis n for LETKF (solid) and 4D-LETKF (dashes) using our default parameters $m = 40, d = 6, k = 10$. See text for the amount of variance inflation used.

that using the symmetric square root in the analysis is significantly better than another possible choice of matrix square root.

In operational weather forecasting, one must assimilate data that are collected more frequently than the time between analysis. LETKF extends easily to this case, in a manner equivalent to but simpler than the 4D-EnKF of (Hunt et al. 2004). The only additional computational requirement is to make use of the background ensemble at intermediate times between analyses. In our numerical experiments, we found this approach to perform nearly as well as assimilating data more frequently.

Acknowledgement We thank the members of the University of Maryland Chaos/ Weather group for their input and feedback on the method we have presented. This research was supported by a James S. McDonnell Foundation 21st Century Research Award, NOAA THORPEX grant NA04OAR4310104, and NSF grants DMS 0104087 and ATM0434225.

References

- Anderson, J., 2001: An ensemble adjustment filter for data assimilation. *Monthly Weather Review*, **129**, 2884–2903.
- Bishop, C., B. Etherton, and S. Majumdar, 2001: Adaptive sampling with the ensemble transform kalman filter part i: the theoretical aspects. *Monthly Weather Review*, **129**, 420–436.
- Burgers, G., P. van Leeuwen, and G. Evensen, 1996: On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, **126**, 1719–1724.
- Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, **99**, 10143–10162.
- Hamill, T., J. Whitaker, and C. Snyder, 2001: Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter. *Monthly Weather Review*, **129**, 2776–2790.
- Houtekamer, P. and H. Mitchell, 1998: Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, **126**, 796–811.
- 2001: A sequential ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, **129**, 123–137.
- Hunt, B., 2005: Efficient data assimilation for spatiotemporal chaos: a local ensemble transform kalman filter. <http://arxiv.org/abs/physics/0511236>.
- Hunt, B., E. Kalnay, E. Kostelich, E. Ott, D. Patil, T. Sauer, I. Szunyogh, J. Yorke, and A. Zimin, 2004: Four-dimensional ensemble kalman filtering. *Tellus*, **56A**, 273–277.
- Keppenne, C., 2000: Data assimilation into a primitive-equation model with a parallel ensemble kalman filter. *Monthly Weather Review*, **128**, 1971–1981.
- Lorenz, E., 1996: Predictability - a problem partly solved. *Proceedings on predictability, held at ECMWF on 4-8 September 1995*, 1–18.
- Lorenz, E. and K. Emmanuel, 1998: Optimal sites for supplementary weather observations: simulation with a small model. *Journal of the Atmospheric Science*, **55**, 399–414.
- Ott, E., B. Hunt, I. Szunyogh, A. Zimin, E. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. Yorke, 2002: Exploiting local low dimensionality of the atmospheric dynamics for efficient ensemble Kalman filtering. <http://arxiv.org/abs/physics/0203058v3>.

- Ott, E., B. Hunt, I. Szunyogh, A. Zimin, E. Kostelich, M. Corazza, E. Kalnay, and J. Yorke, 2004: A local ensemble kalman filter for atmospheric data assimilation. *Tellus*, **56A**, 415–428.
- Patil, D., B. R. Hunt, E. Kalnay, J. A. Yorke, and E. Ott, 2001: Local low dimensionality at atmospheric dynamics. *Physical Review Letters*, **86**, 5878–5881.
- Szunyogh, I., E. Kostelich, G. Gyarmati, D. Patil, B. Hunt, E. Kalnay, E. Ott, and J. Yorke, 2005: Assessing a local ensemble kalman filter: perfect model experiments with the ncep global model. *Tellus*, **57A**, 528–545.
- Talagrand, O., 1981: A study if the dynamics of four-dimensional data assimilation. *Tellus*, **33**, 43–60.
- Tippett, M., J. Anderson, C. Bishop, T. Hamill, and J. Whitaker, 2003: Ensemble square-root filters. *Monthly Weather Review*, **131**, 1485–1490.
- Wang, X., C. Bishop, and S. Julier, 2004: Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble? *Monthly Weather Review*, **132**, 1590–1605.
- Whitaker, J. and T. Hamill, 2002: Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, **130**, 1913–1924.