ABSTRACT

Title of dissertation: ERRORS IN THE INITIAL CONDITIONS FOR NUMERICAL WEATHER PREDICTION: A STUDY OF ERROR GROWTH PATTERNS AND ERROR REDUCTION WITH ENSEMBLE FILTERING

John Harlim, Doctor of Philosophy, 2006

Dissertation directed by: Professor Brian R. Hunt
Department of Mathematics

In this dissertation, we study the errors of a numerical weather prediction due to the errors in initial conditions and we present efficient nonlinear ensemble filters for reducing these errors.

First, we investigate the error growth, that is, the growth in time of the distance E between two solutions of a global weather model with similar initial conditions. Typically E grows until it reaches a saturation value E_s. We find two distinct broad *log-linear regimes*, one for E below 2% of E_s and the other for E above. In each, $\log(E/E_s)$ grows as if satisfying a linear differential equation. When plotting $d\log(E)/dt$ vs $\log(E)$, the graph is convex. We argue this behavior is quite different from error growth in other simpler dynamical systems, which yield concave graphs.

Secondly, we present an efficient variation of the Local Ensemble Kalman Filter [32, 33] and the results of perfect model tests with the Lorenz-96 model. This scheme is locally similar to performing the Ensemble Transform Kalman Filter [5]. We also include a “four-dimensional” extension of the scheme to allow for asynchronous observations.

Finally, we present a modified ensemble Kalman filter that allows a non-Gaussian background error distribution. Using a distribution that decays more slowly than a Gaussian is an alternative to using a high amount of variance inflation. We demonstrate the effectiveness of this approach for the three-dimensional Lorenz-63 model and the 40-dimensional Lorenz-96 model in cases when the observations are infrequent, for which the non-Gaussian filter reduces the average analysis error by about 10% compared to the analogous Gaussian filter. The mathematical formulation of this non-Gaussian filter is designed to preserve the computational efficiency of the local filter described in the previous paragraph for high-dimensional systems.

ERRORS IN THE INITIAL CONDITIONS FOR NUMERICAL WEATHER PREDICTION: A STUDY OF ERROR GROWTH PATTERNS AND ERROR REDUCTION WITH ENSEMBLE FILTERING

by

John Harlim

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Professor Brian R. Hunt, Chair/Advisor
Professor James A. Yorke
Professor Eugenia Kalnay
Professor Eric J. Kostelich
Professor Denny Gulick

ACKNOWLEDGMENTS

The making of this dissertation is indebted to several people. First, I thank my parents Gemiati Wirjawan and Zain Harlim for their support on my education and career. The fact that they sent me to North America for graduate study, which is very expensive in Rupiahs (Indonesian currency), has given me my present opportunity and future that direct me on achieving my career goal.

I would like to thank my advisor Brian R. Hunt for letting me flourish, especially when he listened to all of my ideas and counter-examples when I was on the wrong track. It has been a pleasure to work with and learn from him. He has shown many and many times his best quality through his brilliant ideas and MOSTLY with his patience and trust of me.

I would like to thank James A. Yorke for many hours he spent with me when he taught me how to write a paper, and helped me prepare my CV and research statement. I will always remember his criticisms that have improved my writing and talks. On one occasion that I always remember, he took the time to help me replace some phrases that I kept repeating in my talk such as “wanna” with “want to” and “gonna” with “going to”.

I would also like to thank Eugenia Kalnay for her assistance in publishing my first scientific paper. She, with her always positive attitude, brought back my confidence in wrapping up that paper.

I would like to thank Eric Kostelich and Denny Gulick for serving on my dissertation committee and reading my dissertation, Frances Gulick for writing me a good teaching letter of recommendation and for her critiques and assistance on my teaching statement, Bill Dorland, Istvan Szunyogh and David Levermore for showing me what scientific computation is through the AMSC 663-664 course, Gyorgy Gyarmati for her assistance with my big struggles with UNIX scripting during my second year, Alverda McCoy and Sandy Allen for their assistance on my job applications, and Bill Goldman for helping me several times getting travel funding. I thank all of the professors that I had during my studies here, they are: Jian-Guo Liu, Da-Lin Zhang, Jim Carton, Ricardo Nochetto, Mark Freidlin, Pete Stewart, Paul Smith, Rebecca Herb, Tobias Von Petersdorff, and Joachim Ballabrera.

I thank D.J. Patil, Michael Oczkowski, Aaron Lott, J.T. Halbert, Elana J. Fertig, Amy Finkbiner, Chris Danforth, Suzanne Sindi, Aleksey Zimin, Junjie Liu, Hong Li, Shu-Chih Yang, Takemasa Miyoshi, and Alfredo Nava-Tudela for assisting me on talk preparations and computing guidance.

This research was partially supported by the NSF grants DMS0104087, ATM-0328402, and ATM0434225, NOAA grant NA040AR4310103 and NA040AR4310-104, NASA grant NNG04GK78A, the Army Research Office and the James S. McDonnell Foundation 21st Century Research Award.

TABLE OF CONTENTS

List of Tables v

List of Figures vi

1 Introduction 1

2 Convex Error Growth Patterns in a Global Weather Model 4
- 2.1 Overview 4
- 2.2 Error Growth of a Global Weather Model 6
- 2.3 Error growth of the Lorenz-96 and QG model 9
- 2.4 A simple error growth model 13
- 2.5 Summary 15

3 Local Ensemble Transform Kalman Filter 17
- 3.1 Motivation 17
- 3.2 Global Filter Formulation 20
  - 3.2.1 Filter Derivation 21
  - 3.2.2 Efficient Computation of the Analysis 27
- 3.3 Localization, 4D Extension, and Variance Inflation 29
  - 3.3.1 Localization 29
  - 3.3.2 4D-LETKF 30
  - 3.3.3 Variance Inflation 34
- 3.4 Simulations on the Lorenz-96 Model 35
  - 3.4.1 Experimental Design 35
  - 3.4.2 Results 37
- 3.5 Summary 43

4 A Non-Gaussian Local Ensemble Filter 46
- 4.1 Motivation 46
- 4.2 Variational Formulation of Ensemble Transform Kalman Filter 49
- 4.3 Non-Gaussian Filter 54
- 4.4 Results 57
  - 4.4.1 Numerical Simulations on the Lorenz-63 Model 57
  - 4.4.2 Numerical Simulations on the Lorenz-96 Model 61
- 4.5 Summary 65

5 Conclusions and Work in Progress 68

Bibliography 71

LIST OF TABLES

4.1 RMS analysis error of Gaussian and non-Gaussian filters on the Lorenz-63 model. The filters are run with ensemble size k=10 and observation error 2. The analysis is performed every 50 steps with time step $\Delta t = 0.01$/step. In the no model error case, the forecast model parameter set is similar to the true model parameter set, $\mu_f = \mu = (10, 8/3, 28)$. Small and large model errors are introduced by setting $\mu_f = (10, 8/3, 30)$ and $\mu_f = (10, 8/3, 35)$, respectively. . . . 59

4.2 Variability of the analysis error for the experiments of of Table 4.1. For direct insertion this quantity is about 1.83. For each coordinate, x, y, z, we computed the square root of the variance of $|\bar{x}^a - x^t|$, $|\bar{y}^a - y^t|$, and $|\bar{z}^a - z^t|$, respectively. 60

LIST OF FIGURES

2.1 Convex exponential error growth rate of the NCEP GFS model as a function of relative error E/E_s. The graph reveals two linear regimes. For $E < 0.02E_s$, E'/E lies close to line $y_f = \lambda_f \log(E/E_f)$ with $\lambda_f \approx -2.5$ and intercept at $E_f \approx 2.1\%$ of the saturation level E_s. For $E > .02E_S$, E'/E lies close to the line $y_s = \lambda_s \log(E/E_s)$ with $\lambda_s \approx -0.17$. The doubling time of errors are 1/8, 1/4, 1, 2, and 3 days when E is 0.23%, 0.7%, 1.6%, 13% and 26% of saturation level, respectively. Small errors grow in amplitude, moving from left to right, first along the left line and then along the right until they saturate at E_s. Each point plotted here is for an individual pair of initial conditions. The dashes represent the "superfast" regime for $E < 2 \times 10^{-3}E_s$ (see text). 8

2.2 Concave error growth rates, estimated by finite-differences in (2.1), as functions of relative error E/E_s. (a) The error growth rate of the Lorenz-96 model fluctuates about 0.35 for $E < 10^{-3}E_s$ and decays in accordance to (2.2) where $\lambda = -0.22$ (see the line tangent to the growth rate) and $C = E_s$. (b) In the QG model, the growth rate fluctuates about 0.044 when $E < 10^{-3}E_s$ and decays asymptotically to (2.2) where $C = E_s$, with $\lambda = -0.035$ as $E > 0.4E_s$. As time increases, errors move from left to right, asymptoting to E_s. Each point plotted here is averaged over $L = 1000$ pair of trajectories for the Lorenz-96 model and $L = 100$ for the QG model. 12

2.3 Concave exponential growth rate, E'/E for (2.7) as a function of E in logarithmic scale. We set $a = 0.35$, $\lambda = -0.22$, $E(0) = 10^{-6}$, and $\Delta t = 1$. When $E < 10^{-3}$, we get $E'/E \approx a = 0.35$. As the error grows, the instantaneous doubling time T_d increases. We show $T_d = 2, 3$ and 4 corresponding to exponential growth rate of 0.34, 0.23 and 0.17, respectively. The line $y = \lambda \log(E)$ with $\lambda = -0.22$ is tangent to the E'/E curve at the asymptotic value $E = 1$. 14

3.1 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. 38

3.2 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 $m = 80$. 40

3.3 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) $\mathbf{W}^a = \mathbf{U}\mathbf{\Sigma}^{1/2}$. 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. 42

3.4 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. 44

4.1 Density of a one-dimensional Gaussian (solid) error distribution with mean 0 and variance 1 and non-Gaussian (dashes) error distribution with $\alpha = 1$. The standard Gaussian density is proportional to $\exp\left\{-\frac{1}{2}\mathbf{w}^T\mathbf{w}\right\}$, while the non-Gaussian pdf is proportional to $\exp\left\{-\frac{1}{2}\frac{\mathbf{w}^T\mathbf{w}}{1+\alpha\sqrt{\mathbf{w}^T\mathbf{w}}}\right\}$. 56

4.2 Forecast errors as functions of time for the experiments in Table 4.1: no model error (top), small model error (middle) and large model error (bottom). We use a logarithmic scale on the vertical axis so that the distance between two curves represents the ratio between their errors. The forecast errors of the initial conditions from direct insertion (dotted) are significantly worse than both the Gaussian (solid) and non-Gaussian (dashed) filters. The observation RMS error is 2 (thin dashed horizontal line). 62

4.3 Forecast error (again on a logarithmic scale) as a function of time: with no model error (top) and with model error (bottom). The forecast errors of the Gaussian filter (solid) are larger than those produced by the non-Gaussian filter (dashed) by about 10% in the case of no model error and 5% in the presence of model error. The observation RMS error is 1 (dotted horizontal line). The Gaussian filter uses variance inflation coefficient $r = 1$ and 1.6 without and with model error, respectively. The non-Gaussian filter uses $\alpha = 0.6$ and 0.8 without and with model errors, respectively. 65

4.4 Forecast variability (again on a logarithmic scale) as a function of time: with no model error (top) and with model error (bottom). The forecast errors variability of the Gaussian filter (solid) are about 30% larger than those produced by the non-Gaussian filter (dashed). . . . 66

Chapter 1

Introduction

In numerical weather prediction, forecasts are generated by solving initial value problems. Forecast errors are caused by model error (deviation of the model from the true atmospheric dynamics) and errors in the initial conditions. Errors in the initial conditions are unavoidable since our collected measurements (from rawinsondes, commercial aircraft, and satellites) are either prone to measurement error or are incomplete and infrequent. The focus of this dissertation is to study the forecast error due to errors in the initial conditions and to develop an efficient scheme that reduces this error.

In Chapter 2, we study the behavior of the forecast error due to errors in the initial conditions. Specifically, we would like to answer the following question: Is there a significant benefit in reducing this particular error? If the answer is yes, for how much longer can a reliable forecast be made if one is able to reduce this error? If the answer is no, then in order to improve forecasts one has to improve the model, which requires better understanding of the physics, better spatial resolution, and requiring greater computing power.

We investigate the error growth over time of initial errors for a global weather model and compare it to that of simpler chaotic models (Lorenz [27], Marshall and Molteni [29]). In these models, a small initial error grows exponentially at a constant rate until it nears saturation. We find that the relation between the exponential error growth rate and the error amplitude is convex in a global weather model (The National Centers for Environmental Predictions Global Forecast System) with multiple time scales, a phenomenon not found in simpler models with a single time scale. After an initial rapid growth phase that occurs in the first 18 hours, the exponential error growth rate slows down, approximately following a linear differential equation in two separate regimes: fast and slow. The current state-of-art of initial conditions in weather prediction is in the slow regime, where errors double roughly in 2 days (see Bengtsson and Hodges [4]). This suggests that reducing the errors in the initial conditions by half will enable accurate forecasts about 2 days further in the future. This result answers our question and it serves as a motivation for the rest of this dissertation.

In Chapter 3, we develop a data assimilation scheme to reduce the errors in the initial conditions. That is, we approximate the current state of the atmosphere by correcting a “background” forecast from a prior initial condition to be more consistent with the current observations. Meteorologists call this correction *analysis*. The revised (also called *analysis*) state is then fed into the model as an initial condition that provides the background forecast at the next time analysis is performed. In an ensemble Kalman filter, one generates forecasts from an ensemble of initial conditions to assess the uncertainty in the background state. We present an efficient scheme to generate the analysis state with a variation of the Local Ensemble Kalman Filter [32, 33]. We test this scheme on a perfect-model scenario with the 40-dimensional Lorenz-96 model [27]. We call this scheme the Local Ensemble

Transform Kalman Filter (LETKF) because it is locally similar to performing the Ensemble Transform Kalman Filter [5]. We also include a four-dimensional extension of the scheme to allow for asynchronous observations.

Finally, we propose an extension of the LETKF to non-Gaussian error distributions in Chapter 4. Our motivation is that with ensemble Kalman filters, one often needs to artificially inflate the uncertainty suggested by the ensemble forecast from one analysis time to the next, sometimes by a significant factor. This uncertainty is typically underestimated due to the nonlinearity of the model and to the small ensemble size. In this non-Gaussian filter, we use a symmetric forecast error distribution with longer tails than a Gaussian. Hence, the closed-form formulas of Kalman Filter are not usable since they assume a Gaussian error distribution. Instead, a minimization scheme such as a conjugate gradient method is used. Formally, this scheme is similar to the Maximum Likelihood Ensemble Filter [44] with a non-quadratic likelihood function. We test this scheme on the 3-dimensional Lorenz-63 [24] model and the 40-dimensional Lorenz-96 model and compare it to our LETKF and see a substantial improvement for the case of infrequent observations. We end this dissertation with a conclusion and discussion of work in progress in Chapter 5.

Chapter 2

Convex Error Growth Patterns in a Global Weather Model

2.1 Overview

The defining characteristic of a temporal chaotic dynamical system is a positive leading Lyapunov exponent. This quantity measures the long time average exponential growth rate of two typical solutions that initially are separated by an infinitesimal distance. One simple example of a chaotic system is the logistic map: $x_{n+1} = 4x_n(1-x_n)$, whose Lyapunov exponent equals $\log(2)$. This quantity suggests that the difference between two nearby solutions doubles on each iteration until it reaches saturation. From the point of view of weather forecasting, the logistic map has an infinite limit of predictability: we can always extend the forecast for one more iterate by halving the uncertainty in the initial conditions. Lorenz [27] suggested that the Lyapunov exponents may be unbounded for the partial differential equation representing a global atmospheric model. He also argued that as saturation is approached, the error growth rate behaves differently from Lyapunov exponents.

The atmosphere includes multiple scales of motion, which suggests that different scales of motion grow with different rates. In [25], Lorenz observed that small scales tend to grow at a fast rate and the larger scales of motion grow with a slower rate. He concluded that as one refines the accuracy of the initial states, smaller increments of forecast skill are obtained, and there appears to be a finite limit of predictability. Numerous studies have been devoted to the predictability of different types of motion that mimic the atmosphere. For example, Aurell et al. [3] show that in the case of turbulence, the growth rate is determined by the cumulative effects of multiple characteristic times.

A hallmark of chaos is the exponential growth of errors, where by ***error*** we mean the distance $E(t)$ between two trajectories that are close to each other at time $t = 0$. When trajectories are bounded, the exponential growth of $E(t)$ cannot continue indefinitely; $E(t)$ saturates near a value E_s that is representative of the size of the chaotic attractor. We consider the dependence of the exponential growth rate of the error on the size of $E(t)/E_s$ for a global weather forecast model, and we contrast our results with those for some simpler models. The ***exponential growth rate*** is E'/E, where E' is either the time derivative of E or a finite-time approximation of the time derivative. Throughout this chapter, we approximate the exponential growth rate as:

$$\frac{E'}{E} = \frac{d \log E(t)}{dt} \approx \frac{\log E(t + \Delta t) - \log E(t - \Delta t)}{2\Delta t} \tag{2.1}$$

for a suitable Δt. This approximation is used to mask the rapid fluctuation of the derivative. In some cases, again to suppress fluctuations, we may take E to be an average distance, averaging over several pairs of solutions.

We call an interval J of values of E a ***log-linear regime*** if for some $\lambda < 0$ and $C \leq E_s$ and all $E \in J$, E'/E approximately satisfies the linear differential equation

[1]:

$$\frac{E'}{E} = \lambda \log(E/C). \tag{2.2}$$

The exponent λ describes how E approaches C. A typical dynamical system has one log-linear regime with $C = E_s$; then λ is called the ***saturation exponent***. Here, we investigate a realistic weather forecast model and find that this model has two distinct broad log-linear regimes (Figure 2.1).

2.2 Error Growth of a Global Weather Model

We report a striking behavior for a global weather model, the National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS). This operational global atmospheric model is a pseudo-spectral model, described in detail in [8, 35]. The resolution considered here is chosen such that it has a maximum zonal wavenumber of 62, and so is referred to as T62, where "T" stands for triangular truncation. There are 192×94 horizontal grid points at each of 28 vertical levels. In addition to the surface pressure, there are five variables defined at every grid point: 1) the vertical component of the vorticity of the horizontal wind, 2) the divergence of the horizontal wind, 3) a generalized temperature that reflects humidity, 4) relative humidity, and 5) ozone. Overall the state has dimension $N \approx 3 \times 10^6$.

We select a point $\mathbf{x}$ in the state space from a trajectory after transients have died away. We also choose at random an N-vector $\delta\mathbf{p}$ and renormalize it so that

[1] All logs are natural logarithms; generally we want E'/E to be within 10% of $\lambda \log(E/C)$

$\|\delta\mathbf{p}\| = E_s$. We choose perturbed initial points

$$\mathbf{p}^k(0) = \mathbf{x}(0) + 10^{-k}\delta\mathbf{p} \text{ for } k = 3, ..., 7. \tag{2.3}$$

We do not show the cases for $k = 1, 2$ because we are interested in the behavior of perturbations that are initially small. Both the reference state $\mathbf{x}$ and perturbations $\mathbf{p}^k$ are integrated from $t = 0$ for 14 days with a 20 minute time step. Now, define $E^k(t)$ to be the root-mean-square (rms) error of $\mathbf{x}$ and $\mathbf{p}^k$ for each k:

$$E^k(t) = \|\mathbf{p}^k(t) - \mathbf{x}(t)\|. \tag{2.4}$$

We focus our study on midlatitude tropospheric wind prediction. That is, calculations are restricted to the midlatitude bands in the Northern and Southern Hemispheres ($22.5° - 70°$ N/S) where the model is considered most accurate. We report the rms errors calculated for the atmospheric level where the pressure equals half of the surface pressure. We show only the rms errors of the vorticity, but the other variables behave in a similar way.

In Figure 2.1, for $E = E^3(t), \ldots, E^7(t)$, the exponential growth rate E'/E is plotted as a function of relative error size E/E_s in logarithmic coordinates. The error growth rate E'/E decays along two broad log-linear regimes (for time $t > 18$ hours). In what we call the "fast regime" ($0.002E_s < E < 0.02E_s$), the errors double in less than a day. Here the error growth rates move along a straight line (2.2) with $\lambda = -2.5$ ($= \lambda_f$ in Figure 2.1) as if to saturate at $C = E_f \approx 0.021E_s$, that is, 2.1% of the actual saturation size. After E reaches about $0.02E_s$, the growth rate enters the "slow regime." We refer to the point $E = 0.02E_s$ as the $\boldsymbol{KT}$

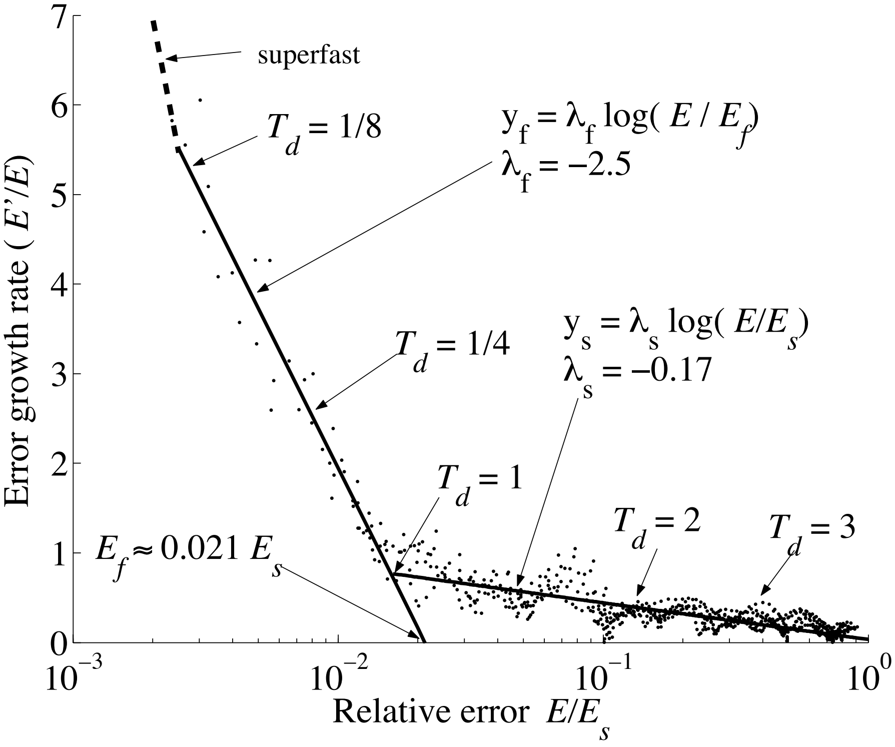

Figure 2.1: Convex exponential error growth rate of the NCEP GFS model as a function of relative error E/E_s. The graph reveals two linear regimes. For $E < 0.02E_s$, E'/E lies close to line $y_f = \lambda_f \log(E/E_f)$ with $\lambda_f \approx -2.5$ and intercept at $E_f \approx 2.1\%$ of the saturation level E_s. For $E > .02E_S$, E'/E lies close to the line $y_s = \lambda_s \log(E/E_s)$ with $\lambda_s \approx -0.17$. The doubling time of errors are 1/8, 1/4, 1, 2, and 3 days when E is 0.23%, 0.7%, 1.6%, 13% and 26% of saturation level, respectively. Small errors grow in amplitude, moving from left to right, first along the left line and then along the right until they saturate at E_s. Each point plotted here is for an individual pair of initial conditions. The dashes represent the "superfast" regime for $E < 2 \times 10^{-3}E_s$ (see text).

boundary [2]. After E passes the KT boundary, E'/E follows the straight line (2.2) with $\lambda = -0.17$ ($= \lambda_s$) and $C = E_s$. Here the growth rate E'/E slows to zero as E approaches its saturation level E_s. In creating Figure 2.1, we approximate E'/E using (2.1) with $\Delta t = 6$ hours, with t in steps of 3 hours. The finite differences show some oscillation about the lines. We plot E'/E starting from $t = 18$ hours because the NCEP model exhibits a third regime: superfast growth of extremely small perturbations. Errors of size less than $10^{-4}E_s$ climb rapidly to $10^{-4}E_s$, usually in about 1 hour or less, even when beginning with size $10^{-7}E_s$ (not shown here). Thus, the errors grow to approximately $0.002E_s$ after 18 hours independently of $E(0)$ provided $E(0) < 10^{-4}E_s$. Toth and Kalnay [39, 40] investigated weather prediction using a reference state plus multiple perturbations as initial conditions. They report "enormous" growth of errors, more than a factor of 5 per day, when perturbations have amplitude less than $0.001E_s$. They attribute such growth mostly to tropical convection, which they say saturates at less than $0.01E_s$. They see slow growth for amplitudes between $0.01E_s$ and $0.1E_s$ and attribute this behavior to baroclinic instabilities. They do not discuss the transition between these behaviors.

2.3 Error growth of the Lorenz-96 and QG model

In contrast with the striking results of Figure 2.1, simpler models often have a single log-linear regime. We illustrate the typical behavior of simpler models with the 40-dimensional Lorenz-96 model [27] of differential equations and the Quasi-

[2] Named after Toth and Kalnay [39, 40] and the famous paleontological demarcation point.

Geostrophic (QG) model of Marshall and Molteni [29] with resolution T21.

The Lorenz-96 model [27, 28] represents an "atmospheric variable" with values x_j at N equally spaced points around a circle of constant latitude:

$$5\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F \tag{2.5}$$

where $j = 1, ..., N$ represent the spatial coordinates ("longitude"). Periodic boundary conditions are imposed by identifying $x_{-1} \equiv x_{N-1}$, $x_0 \equiv x_N$, and $x_{N+1} \equiv x_1$. This model is designed to satisfy three basic properties: it has linear dissipation (the $-x_j$ term) that decreases the total energy (defined as $V = \frac{1}{2}\sum_{j=1}^{N} x_j^2$), an external forcing term F that can increase or decrease the total energy, and a quadratic advection-like term that conserves the total energy (i.e., it does not contribute to $\frac{d}{dt}V$). Following [27, 28], we choose the external forcing to be $F = 8$ and the number of spatial elements to be $N = 40$. The "5" in (2.5) scales the unit time to correspond to one day in real time. We also use a fourth-order Runge-Kutta scheme for time integration of (2.5) with time step $\Delta t = 1/4$ days. With these parameters, the solution to (2.5) has a behavior reminiscent of the midlatitude atmosphere. It has 13 positive Lyapunov exponents, with the leading Lyapunov exponent corresponding to a doubling time of 2.1 days, and a Kaplan-Yorke dimension of 27.1 [27].

The QG approximation describes weather-like slow atmospheric motion. Steady state flow in a rotating sphere satisfies the geostrophic balance, i.e., the horizontal pressure gradient balances the Coriolis force. Slow Rossby waves satisfy a quasi-geostrophic balance as represented by the conservation of quasi-geostrophic potential vorticity (QGPV), see Holton [15]. In our study, we use the three-level QG model developed by Marshall and Molteni [29] with T21 resolution.

In the NCEP model, we examined the rms difference between a pair of trajectories $\mathbf{x}$ and $\mathbf{p}^k$. For the Lorenz-96 and QG models, we examine L such pairs of trajectories and average the resulting errors in order to mask the rapid fluctuation of E. The rms of the i-th pair of $\mathbf{x}(t)$ and $\mathbf{p}^k(t)$ at time t, denoted as $E_i^k(t)$, is calculated from (2.4) for $i = 1, ..., L$. Then, we let $E^k(t)$ be the geometric mean

$$E^k(t) = \exp\langle \log E_i^k(t) \rangle, \tag{2.6}$$

where the average $\langle \cdot \rangle$ is computed over L pairs. The previous $E^k(t)$ in Figure 2.1 is an average of a single ($L = 1$) trajectory. We use $L = 1000$ for the Lorenz-96 model and $L = 100$ for the QG model. Both the reference state $\mathbf{x}$ and perturbations $\mathbf{p}^k$, with $k = 3, ..., 6$, are integrated with 6 hour time steps from time $t = 0$ for 60 days for the Lorenz-96 model and for 360 days for the QG model.

In Figure 2.2 ((a) for Lorenz-96 and (b) for QG model), the exponential growth rates are plotted as functions of the relative rms difference between each $\mathbf{p}^k$ and $\mathbf{x}$. In creating these plots, the growth rate is approximated using (2.1) with $\Delta t = 1/2$ days for the Lorenz-96 and $\Delta t = 10$ days for the QG model. Our experiments with the Lorenz-96 and the QG models show small averaged errors growing exponentially. Close to saturation (i.e., as $E \to E_s$), the exponential growth rate slows down as with differential equation (2.2) with $C = E_s$ and enters the log-linear regime. Specifically, the log-linear regime of Lorenz-96 model is approximately $0.4E_s < E < E_s$ with saturation exponent $\lambda \approx -0.22$. We obtain a similar range of log-linear behavior with $\lambda \approx -0.035$ for the QG model. We believe that this single time scale behavior

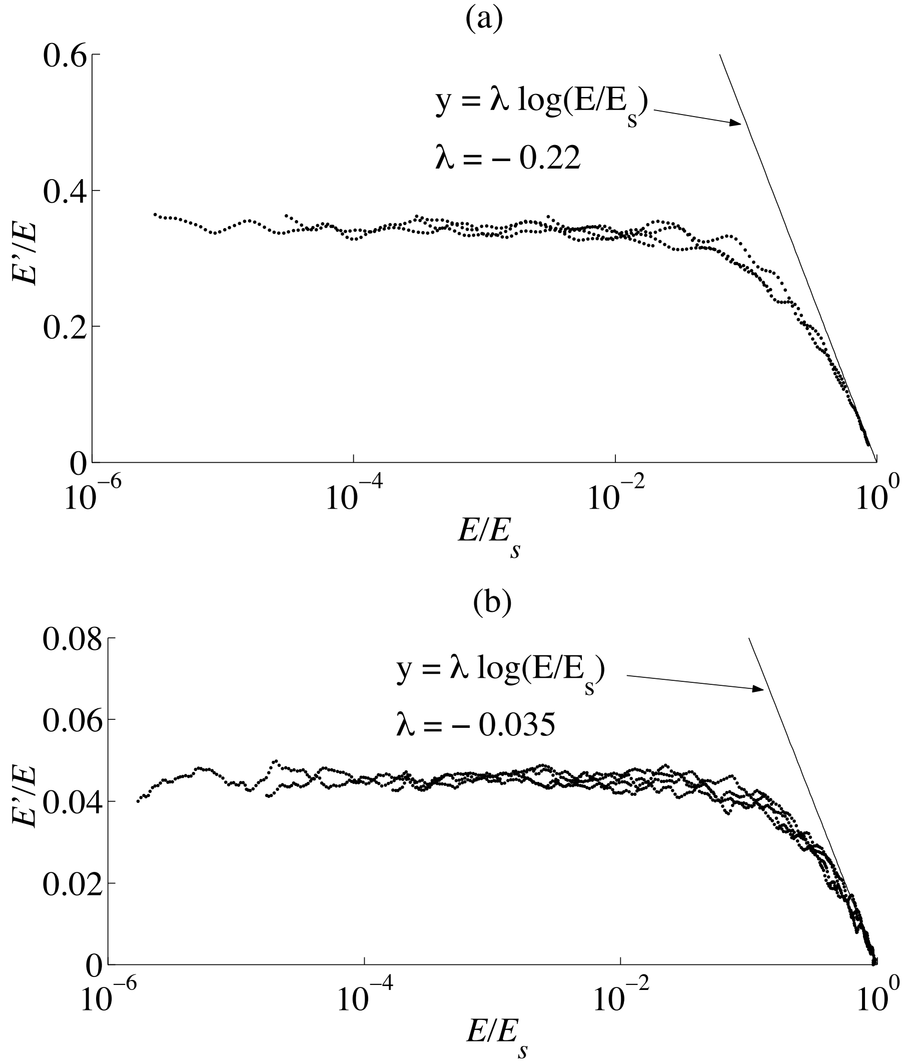

Figure 2.2: Concave error growth rates, estimated by finite-differences in (2.1), as functions of relative error E/E_s. (a) The error growth rate of the Lorenz-96 model fluctuates about 0.35 for $E < 10^{-3}E_s$ and decays in accordance to (2.2) where $\lambda = -0.22$ (see the line tangent to the growth rate) and $C = E_s$. (b) In the QG model, the growth rate fluctuates about 0.044 when $E < 10^{-3}E_s$ and decays asymptotically to (2.2) where $C = E_s$, with $\lambda = -0.035$ as $E > 0.4E_s$. As time increases, errors move from left to right, asymptoting to E_s. Each point plotted here is averaged over $L = 1000$ pair of trajectories for the Lorenz-96 model and $L = 100$ for the QG model.

is due to the presence of only baroclinic instabilities in the QG model, while other instabilities (discussed at Section 2.5) are present in the NCEP model. Next, we offer a scalar linear differential equation that justifies the typical error growth as observed in Lorenz-96 or QG model.

2.4 A simple error growth model

Here we modify the logistic differential equation [3] and model the growth of the error $E(t)$ by a scalar differential equation:

$$E' = \frac{dE}{dt} = aE(1 - E^{-\lambda/a}). \tag{2.7}$$

Near the steady state $E = 0$, we have $E'/E \approx a$. For $0 < E < 1$, we have $E \rightarrow 1$ as $t \rightarrow \infty$, and for $E \approx 1$, $E'/E \approx \lambda \log(E)$. With this variant of the logistic equation, two parameters a and λ can be selected independently. Figure 2.3 shows the exponential growth rate of the error, E'/E compared to E. We choose $a = 0.35$ and $\lambda = -0.22$ to fit the rate of Lorenz-96 model in Figure 2.2(a). For $E > 0.7$ [4], E'/E is close to the line (2.2) with $C = 1$. Similarly, the error growth rate of the QG model can be illustrated by (2.7) by choosing $a = 0.044$ and $\lambda = -0.035$.

[3] Lorenz [26] modeled the error growth by the one-parameter logistic equation $E' = aE(1 - E)$. This equation, however, has an initial exponential growth rate equal to its saturation rate; that is $a = -\lambda$ in (2.7). Such a constraint, as we see in Figure 2.2, is not necessarily satisfied.

[4] We say E is in the log-linear regime when $\frac{a(1-E^{-\lambda/a})}{\lambda \log E} > 0.9$, namely for $E \in (0.7, 1)$.

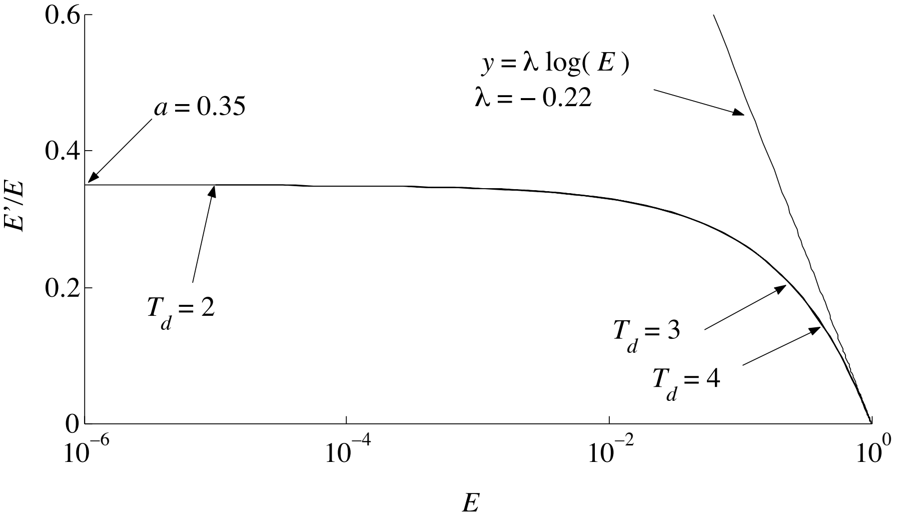

Figure 2.3: Concave exponential growth rate, E'/E for (2.7) as a function of E in logarithmic scale. We set $a = 0.35$, $\lambda = -0.22$, $E(0) = 10^{-6}$, and $\Delta t = 1$. When $E < 10^{-3}$, we get $E'/E \approx a = 0.35$. As the error grows, the instantaneous doubling time T_d increases. We show $T_d = 2, 3$ and 4 corresponding to exponential growth rate of 0.34, 0.23 and 0.17, respectively. The line $y = \lambda \log(E)$ with $\lambda = -0.22$ is tangent to the E'/E curve at the asymptotic value $E = 1$.

2.5 Summary

Convex error growth in the NCEP global atmospheric model contrasts with the concave error growth seen in simpler models that have a single type of instabilities. In the simple models we observe constant exponential growth that decreases once nonlinear effects become important. In the realistic weather model there is no exponential growth with constant E'/E. Instead, E'/E is strictly decreasing, indicating the presence of *multiple* types of instabilities that dominate at very different amplitude ranges. We see an initial super rapid growth, followed by two log-linear regimes. This is reminiscent of the difference between the 2-dimensional turbulence that dominates large scales (Charney [7]) and 3-dimensional turbulence that dominates smaller scales, with a transition taking place for wavelengths between 400 and 1000 km (Nastrom and Gage [31], 1985). The shortest waves present in our model (with triangular truncation T62) have a wavelength of 40000 km/62 $\approx$ 320 km so that there is only enough resolution to crudely resolve 3-dimensional turbulent scales.

Our results suggest that the growth of perturbations occurs in three phases and we examine the perturbation kinetic energy spectra (see Nastrom and Gage [31]) in each of these phases. In the first phase, tiny perturbations grow very fast through the most efficient mechanism that provides finite amplitude perturbation growth: the triggering of cumulus convection at slightly different times or locations. Because of the enormous latent heat energy released by convection, even the smallest perturbations that are computationally realizable become significant within an interval of a few convective time scales (within 1-3 hours). In the first few time steps the perturbation kinetic energy is dominated by wavenumbers larger than 32, corresponding to single grid point cumulus perturbations. After this initial supergrowth period, perturbations have acquired roughly the same finite size ($3 \times 10^{-3} E_s$) independent of the initial perturbations sizes (Fig. 2.1). The fact that during this period of superfast growth, "butterfly"-sized model perturbations become significant, supports the insight of Lorenz [25] and is consistent with the results of Zhang et al. [43] and Toth and Kalnay [39]. In the second phase, from 1-3 hours to about 12-36 hours, the perturbation spectrum shifts to waves between wavenumbers 20 and 50 (wavelengths between 800 km and 2000 km). This fast growth of small-scale perturbations is dominated by small-scale, turbulent dynamics. Beyond that time, the growth slows further as the perturbation energy shifts to lower wavenumbers in the range of 2-dimensional turbulence, until by the saturation time, the kinetic energy of the difference peaks at wavenumber 10 (about 4000 km, the characteristic wavelength of baroclinic instability).

The results of this chapter were published in [14]. While these results were obtained for the NCEP global model, we believe that they are relevant to all realistic models of the atmosphere that include convection. As mentioned before in Chapter 1, the current state-of-the-art of initial conditions in weather prediction is in the slow regime, where errors double roughly in 2 days. Therefore reducing the errors in the initial conditions by half will enable accurate forecasts about 2 days further in the future. Similar results were reported recently by Bengtsson and Hodges [4]. This motivates the next two chapters in this dissertation.

Chapter 3

Local Ensemble Transform Kalman Filter

3.1 Motivation

Here, we describe 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 because 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.

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 that at the next analysis time provides 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 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 least-squares sense.

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) [38] or a deterministic EnKF [42]. In contrast, the early versions of EnKF [9, 6, 16] are stochastic in the sense that they perturb the observations randomly when 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 [1], Ensemble Transform Kalman Filter (ETKF) of [5], and Local Ensemble Kalman Filter (LEKF) of [32, 33].

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 [1], [13], and [17], 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 chapter, 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 to 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 [34] and the results in [36] 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 [5]. In this respect, our approach is analogous to the Spherical Simplex ETKF of [41].

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 [20]. 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 [37], 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. Here, we introduce a 4D-LETKF that is analogous to 4D-EnKF but mathematically simpler.

The reminder of this chapter is organized as follows. In Section 3.2 we present our global filter, which 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 3.3. We conclude this chapter by showing some results obtained for the Lorenz-96 model [28] in Section 3.4, and give a brief summary in Section 3.5.

3.2 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 detailed derivation, see [19].

3.2.1 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 $\{\mathbf{x}^{b(i)} : i = 1, 2, \ldots, k\}$ of size k, the observation operator $H : \mathbb{R}^m \to \mathbb{R}^s$ 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 ith 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

$$\mathbf{P}^b = (k-1)^{-1} \mathbf{X}^b (\mathbf{X}^b)^T. \tag{3.1}$$

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 $\{\mathbf{x}^{a(i)} : i = 1, 2, \ldots, k\}$. The analysis mean

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

and error covariance matrix

$$\mathbf{P}^a = (k-1)^{-1} \mathbf{X}^a (\mathbf{X}^a)^T, \tag{3.2}$$

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 ith 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 (see e.g. [21]), 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 by 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 those of the ensemble 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 the 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 [21], the analysis ensemble mean is

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{K}(\mathbf{y}^o - \mathbf{H}\bar{\mathbf{x}}^b), \tag{3.3}$$

where

$$\mathbf{K} = \mathbf{P}^b\mathbf{H}^T(\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R})^{-1} \tag{3.4}$$

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

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^b. \tag{3.5}$$

To construct the analysis ensemble, one must then find an $m \times k$ matrix $\mathbf{X}^a$ for which (3.2) 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 on the order of 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. [32, 33] 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.

Our approach is to 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 overdetermined (each point in E has multiple coordinate representations in S). The main convenience of doing the analysis in the space S is that the effective background error covariance matrix is simply

$$\tilde{\mathbf{P}}^b = (k-1)^{-1}\mathbf{I}. \tag{3.6}$$

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 allows us to use a more convenient form of the Kalman filter equations involving the inverse of the background error covariance matrix.

Substituting (3.1) to (3.4) yields

$$\mathbf{K} = (k-1)^{-1}\mathbf{X}^b(\mathbf{X}^b)^T\mathbf{H}^T[(k-1)^{-1}\mathbf{H}\mathbf{X}^b(\mathbf{X}^b)^T\mathbf{H}^T + \mathbf{R}]^{-1}. \tag{3.7}$$

At this point we can deal effectively with a nonlinear observation operator H. In (3.7), we see that every time the linearized operator $\mathbf{H}$ appears, it is next to the matrix $\mathbf{X}^b$. The ith column of $\mathbf{HX}^b$ 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 $\mathbf{HX}^b$ by the matrix $\mathbf{Y}^b$ of background ensemble observation perturbations, whose ith column is $H(\mathbf{x}^{b(i)}) - \bar{\mathbf{y}}^b$, where $\bar{\mathbf{y}}^b$ 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 the matrix identity[1]

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

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 (3.5),

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

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

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

Then

$$\mathbf{P}^a = \mathbf{X}^b\tilde{\mathbf{P}}^a(\mathbf{X}^b)^T, \tag{3.8}$$

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

where

$$\tilde{\mathbf{P}}^a = \left[(k-1)\mathbf{I} + (\mathbf{Y}^b)^T\mathbf{R}^{-1}\mathbf{Y}^b)\right]^{-1} \tag{3.9}$$

is the $k \times k$ matrix that represents the analysis error covariance in the space S. Then, (3.3) becomes

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b\tilde{\mathbf{P}}^a(\mathbf{Y}^b)^T\mathbf{R}^{-1}(\mathbf{y}^o - \bar{\mathbf{y}}^b), \tag{3.10}$$

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

$$\mathbf{w}^a = \tilde{\mathbf{P}}^a(\mathbf{Y}^b)^T\mathbf{R}^{-1}(\mathbf{y}^o - \bar{\mathbf{y}}^b), \tag{3.11}$$

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. Despite the formal differences, one can show that the analysis increment given by (3.10) and (3.9) is equivalent to Ensemble Transform Kalman Filter (ETKF) of Bishop et al. [5].

To construct the analysis ensemble, we choose

$$\mathbf{X}^a = \mathbf{X}^b\mathbf{W}^a, \tag{3.12}$$

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 (3.8), we have the required relationship (3.2). The ith member of the analysis ensemble is created by adding the ith column of (3.12) to (3.10)

$$\mathbf{x}^{a(i)} = \bar{\mathbf{x}}^a + \mathbf{X}^{a(i)}. \tag{3.13}$$

For the mean of the analysis ensemble to be $\bar{\mathbf{x}}^a$, we require the sum of the columns $\mathbf{X}^{a(i)}$ of $\mathbf{X}^a$ to be zero; that is, $\mathbf{X}^a\mathbf{v} = \mathbf{0}$ where $\mathbf{v} = (1, 1, \ldots, 1)^T$. Since $\mathbf{X}^b\mathbf{v} = \mathbf{0}$, it suffices that $\mathbf{v}$ be an eigenvector of $\mathbf{W}^a$. 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 [5]; see also Wang et al. [41]. We will discuss this further in Section 3.4.2. Thus the filter we describe in this section is equivalent to ETKF using the symmetric square root of the analysis covariance in the k-dimensional space S.

3.2.2 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 $\{\mathbf{x}^{b(i)} : i = 1, 2, \ldots, k\}$, 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 substract 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 $\tilde{\mathbf{P}}^a = \left[(k-1)\mathbf{I} + \mathbf{C}\mathbf{Y}^b\right]^{-1}$. Multiplying $\mathbf{C}$ and $\mathbf{Y}^b$ requires $2k^2 s$ 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.

3.3 Localization, 4D Extension, and Variance Inflation

The fundamental difference between LETKF and ETKF is the 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.

3.3.1 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 3.4.2, even for relatively small models, localization can improve filter performance. See [20] for a discussion of different localization strategies.

Our localization is similar to that of [22] and [32, 33], 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 3.2 using only the local observations. To be more precise, we first perform Steps 1 and 2 of Section 3.2.2 globally (though in a 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 analyses at nearby grid points be similar. 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 $\tilde{\mathbf{P}}^a$.

3.3.2 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 [20], 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 observation times in conjunction with the observation operator H. In the LETKF framework, we are able to simplify this approach by directly 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 3.2 we wrote the analysis mean as $\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b\mathbf{w}^a$, where $\mathbf{w}^a$ is determined from $\mathbf{R}, \mathbf{y}^o, \bar{\mathbf{y}}^b$, and $\mathbf{Y}^b$ by (3.9) and (3.11). 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 we must redefine $\bar{\mathbf{y}}^b$ and $\mathbf{Y}^b$ accordingly.

Let t_n be the assimilation time and $t_l \neq t_n$ be a time where the observations are collected. As in [20], we can use the background ensemble together with the observation operator H for time t_l to map a model state at time t_n to the observations at time t_l. Recall that in each analysis, we readjust the background ensemble state $\{\mathbf{x}^{b(i)}, i = 1, \ldots, k\}$ with equations (3.10), (3.12), and (3.13) to produce the analysis ensemble state $\{\mathbf{x}^{a(i)}, i = 1, \ldots, k\}$, for which each analysis ensemble member lies in the space spanned by the background ensemble state. Thus, we can write a prospective model state $\mathbf{x}_n$ at time t_n as

$$\mathbf{x}_n = \mathbf{X}(t_n)\mathbf{w}, \tag{3.14}$$

where column vectors of $\mathbf{X}(t_n)$ are the background ensemble states at time t_n (see Step 1 of Section 3.2.2) and $\mathbf{w}$ is a $k \times 1$ weight vector. Notice that unlike $\mathbf{w}^a$, which was multiplied by the ensemble perturbations from their mean, $\mathbf{w}$ is multiplied by ensemble states themselves. At the observation time t_l, we associate to $\mathbf{x}_n$ the model state

$$\mathbf{x}_l = \mathbf{X}(t_l)\mathbf{w},$$

where column vectors of $\mathbf{X}(t_l)$ denote the ensemble background states at time t_l.

The orthogonal projection of vector $\mathbf{v}$ onto the space spanned by the column vectors of $\mathbf{X}(t_n)$ is given by $\mathbf{X}(t_n)\left(\mathbf{X}(t_n)^T\mathbf{X}(t_n))^{-1}(\mathbf{X}(t_n)\right)^T\mathbf{v}$. A vector $\mathbf{x}_n$ in this space is equal to its projection, so (3.14) is satisfied with

$$\mathbf{w} = \left(\mathbf{X}(t_n)^T\mathbf{X}(t_n)\right)^{-1}(\mathbf{X}(t_n))^T\mathbf{x}_n.$$

Therefore, observations $H(\mathbf{x}_l)$ at time t_l can be described as a function of $\mathbf{x}_n$ by:

$$\begin{aligned} H'(\mathbf{x}_n) &= H(\mathbf{X}(t_l)(\mathbf{X}(t_n)^T\mathbf{X}(t_n))^{-1}(\mathbf{X}(t_n))^T\mathbf{x}_n) \\ &= H(\mathbf{X}(t_l)\mathbf{w}) \\ &= H(\mathbf{x}_l). \end{aligned} \tag{3.15}$$

In 4D-EnKF [20], H' replaces operator H during the analysis. Next we show that in our formulation, the coordinate transformation in the definition of H' becomes unnecessary.

In the analysis prescribed by (3.9) and (3.10), the terms $\mathbf{y}^b$ and $\mathbf{Y}^b$ depend on the images $H(\mathbf{x}^{b(i)})$ of the ensemble members in observation space. Let us replace every H by H' as in (3.15) and define $\mathbf{e}_i$ to be a column vector with one on the ith term and zero everywhere else. Thus, at time t_l we have

$$\begin{aligned} H'(\mathbf{x}^{b(i)}(t_n)) &= H\left(\mathbf{X}(t_l)\left(\mathbf{X}(t_n)^T\mathbf{X}(t_n)\right)^{-1}(\mathbf{X}(t_n))^T[\mathbf{X}(t_n)\mathbf{e}_i]\right) \\ &= H(\mathbf{X}(t_l)\mathbf{e}_i) \\ &= H(\mathbf{x}^{b(i)}(t_l)), \end{aligned}$$

so $\bar{\mathbf{y}}^b(t_l) = \overline{H(\mathbf{x}^{b(i)}(t_l))}$, and $\mathbf{Y}^b(t_l) = H(\mathbf{x}^{b(i)}(t_l)) - \bar{\mathbf{y}}^b(t_l)$. Hence, all of the analysis derived in Section 3.2.1 holds whenever observations at time t_l are incorporated during the analysis at time t_n.

Now suppose that we have data $(t_l, \mathbf{y}_l^o)$ from various times $t_l, l = 1, 2, \ldots, n$ 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 respective 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 3.2: 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 3.2 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 $\mathbf{w}_l^a = \tilde{\mathbf{P}}^a\mathbf{C}_l(\mathbf{y}_l^o - \mathbf{y}_l^b)$.

3.3.3 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 3.2.1,

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

where $\tilde{\mathbf{P}}^a$ and $\tilde{\mathbf{P}}^b$ is defined by (3.9) and (3.6), 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 (3.6) by ρ. Therefore, (3.9) is replaced by

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

In other words, replace $(k-1)\mathbf{I}$ by $(k-1)\mathbf{I}/\rho$ in Step 4 of Section 3.2.2. One can check that this change yields the same $\bar{\mathbf{x}}^a$ and $\mathbf{X}^a$ in (3.10) and (3.12) as leaving (3.9) 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.

3.4 Simulations on the Lorenz-96 Model

3.4.1 Experimental Design

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

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F \qquad (3.16)$$

where $j = 1, ..., m$ 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 (on average) increases 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 [27] and [28], 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 (3.16) 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 [27]. 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 the rest of this chapter, 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 the 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 $(\mathbf{x}^{b(i)} : i = 1, 2, ..., k)$ 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 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 $t = \Delta t, 2\Delta t, ..., N\Delta t$. We then take the root-mean-square average of these differences over 10 different runs of N=20,000 analyses each. For the rest of this chapter, 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 an analysis.

3.4.2 Results

In this section, we verify and assess the sensitivity of LETKF and 4D-LETKF toward different parameter quantities. The parameters are the 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, an 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 [32, 33].

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 the variance inflation coefficient. Figure 3.1 indicates that both schemes perform about equally well for $1.04 \leq \rho \leq 1.06$ for which we get the smallest RMSE of about 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

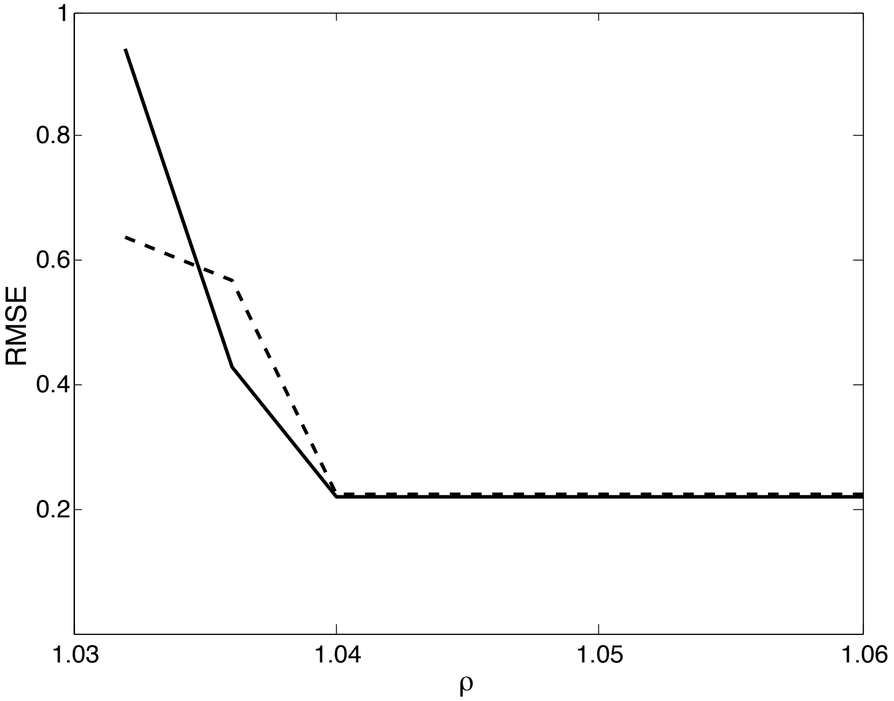

Figure 3.1: RMSE of the LEKF (solid) scheme and the LETKF (dashes) scheme as functions of variance inflation coefficient ρ. Differences for $\rho < 1.04$ are not significant; in these cases, both methods have RMS errors for some of the 10 runs, indicating that the inflation amount is insufficient.

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 3.1, slightly larger values did not significantly change the results. In the local analysis, we fix $d = 6$. Figure 3.2(a) indicates that if one doubles the model variables m from 40 (dashes) to 80 (solid), similar accuracy (shown by RMSE ≈ 0.21) can be obtained by performing the local analysis with an ensemble of size $k = 10$. In Figure 3.2(b), results for the global analysis are plotted in similar fashion as the local analysis in Figure 3.2(a) for ensembles of size 10 to 70 in increments of 5. The results here show that to obtain an accuracy of RMSE ≈ 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 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 [33].

As we mentioned before, ensemble square-root filters generate the analysis ensemble by adding an analysis ensemble of perturbations 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 3.2, 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

$$\mathbf{W}^a = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{V} \tag{3.17}$$

where the ith column of $\mathbf{U} \in \mathbb{R}^{k\times k}$ is the eigenvector of $(k-1)\tilde{\mathbf{P}}^a$ corresponding to the eigenvalue in the ith diagonal element of $\mathbf{\Sigma}$, for $i = 1, \ldots, k$; 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

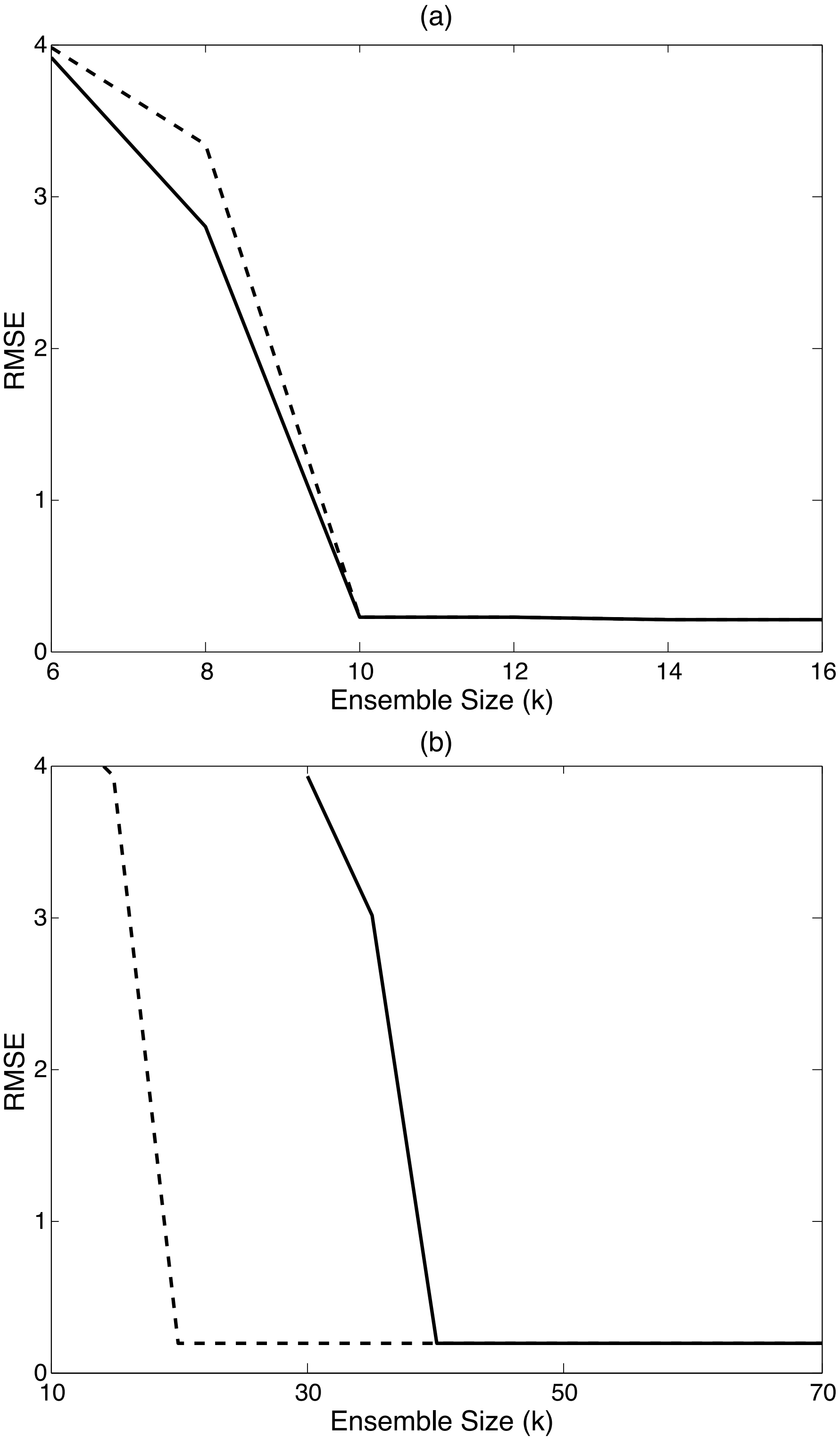

Figure 3.2: RMSE for LETKF as a function of ensemble size k: (a) for local analysis with $d = 6$ and (b) for global analysis. In both cases, the dashed curve is for $m = 40$ and the solid curve is for $m = 80$.

$\mathbf{V} = \mathbf{U}^T$ so that $\mathbf{W}^a$ is symmetric. Wang et al. [41] called this choice the Spherical Simplex ETKF. Now, we compare this choice of square root with the non-symmetric square root $\mathbf{W}^a = \mathbf{U}\mathbf{\Sigma}^{1/2}$ $(\mathbf{V} = \mathbf{I})$, as was originally proposed by Bishop et al. [5]. In Figure 3.3, see that the symmetric square root (dashes) with $k = 20$ ensemble members converges with RMSE ≈ 0.19 and the non-symmetric square root (solid) converges with $k = 40$ ensemble members and RMSE ≈ 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 consistency between adjacent local analyses. As discussed in Section 3.3.1, 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 $\mathbf{W}^a = \mathbf{U}\mathbf{\Sigma}^{1/2}$ 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 members and variance inflation $\rho = 1.30$.

In the last experiment, we validate the four-dimensional scheme described in Section 3.3.2. 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 the analysis at time $jn\Delta t$ it accounts observations at time $l\Delta t$ for $l = (j-1)n+1, (j-1)n+2, \ldots, jn$. For $n = 1$, 4D-LETKF and

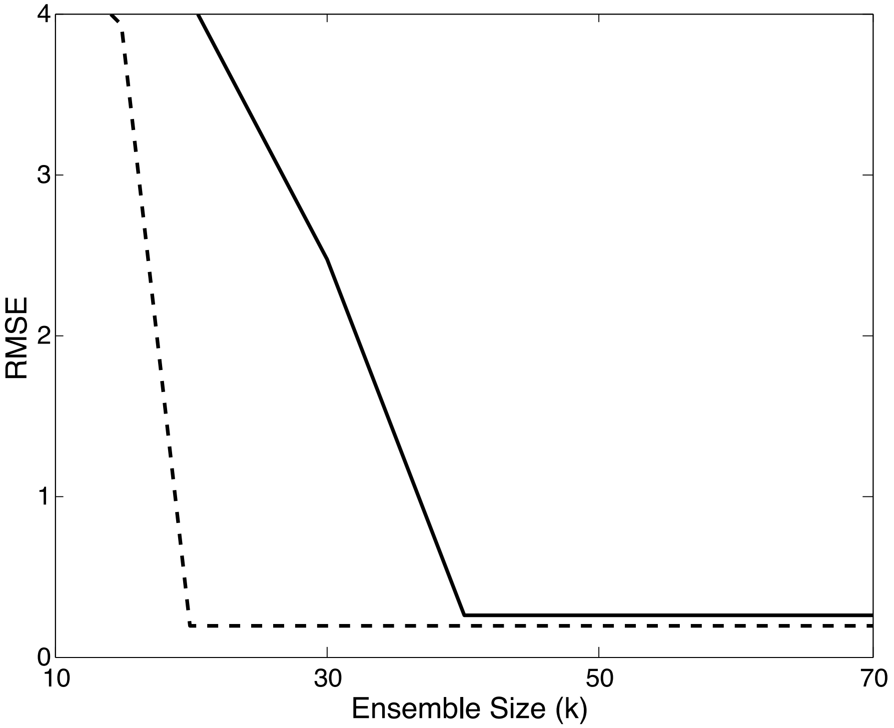

Figure 3.3: Plots of the RMSE for model size $m = 40$ as a function of ensemble size k for the symmetric square root (dashes) and non-symmetric square root (solid) $\mathbf{W}^a = \mathbf{U}\mathbf{\Sigma}^{1/2}$. Here we used variance inflation $\rho = 1.04$ as before for the symmetric square root, but needed more inflation $\rho = 1.15$, to obtain convergence for the non-symmetric square root.

LETKF are equivalent.

Figure 3.4 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 the 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 than linearly as a function of n, in contrast to the linear growth used in [20]. (In their experiment, they performed a global analysis (4D-EnKF) with a different type of "additive" variance inflation.) In Figure 3.4, 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.

3.5 Summary

We combine the local analysis suggested in [32, 33] and the simplified formulation of the global spherical simplex ETKF of [5] and [41]. In our experiments

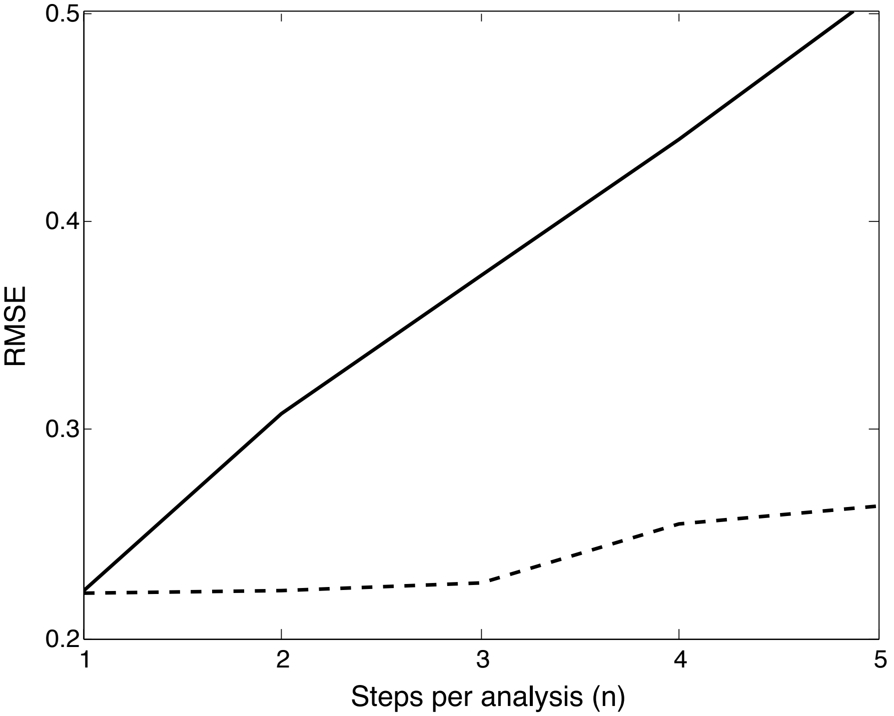

Figure 3.4: Plots of the RMSE as a function of the number of steps per analysis n for LETKF (solid) and 4D-LETKF (dashes) using our default parameters $m = 40, d = 6, k = 10$. See text for the amount of variance inflation used.

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 [32, 33] that localization allows one to maintain a constant ensemble size as the number of variables in the model increases. We also show 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 [20]. 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.

Chapter 4

A Non-Gaussian Local Ensemble Filter

4.1 Motivation

An Ensemble Kalman Filter (EnKF) approximates the state of a system from a time series of noisy observations by iteratively minimizing a quadratic cost function in the space spanned by an ensemble of forecast model states. Typically, this ensemble space is much lower dimensional than the model state space. Instead of using the Kalman filter equations, which analytically minimize the quadratic cost function, here we numerically minimize a non-quadratic cost function. Our approach is similar to the Maximum Likelihood Ensemble Filter (MLEF) of [44], which minimizes a cost function based on a non-Gaussian observation error distribution with a pre-conditioned conjugate gradient method [11]. In this chapter, we show that using a non-quadratic background error distribution can also improve results.

The goal of data assimilation is to approximate the true state $\mathbf{x}^t$ of a system, such as the atmosphere, given noisy observations

$$\mathbf{y}^o = H(\mathbf{x}^t) + \epsilon^o \tag{4.1}$$

at the current and past times, where H is the observation operator, and ϵ^o is the observation error. Typically, the true state $\mathbf{x}^t$ and its underlying dynamics are unknown. We assume that the evolution of $\mathbf{x}^t$ is modeled by a chaotic dynamical system governed by a deterministic differential equation

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mu), \tag{4.2}$$

where μ is a vector of parameters.

From a probabilistic point of view, the problem of data assimilation is that of characterizing the *analysis probability distribution* $p(\mathbf{x}^t|\mathbf{y}^o)$, the distribution of the system state given the observations $\mathbf{y}^o$. The analysis state $\mathbf{x}^a$ is typically chosen to be the mode of this distribution, i.e., the most likely state. We assume that prior observations have yielded a background distribution $p(\mathbf{x}^t)$ and that the distribution of observation errors is known. Then from (4.1), $p(\mathbf{y}^o|\mathbf{x}^t)$ is known as well. Applying Bayes's rule gives

$$p(\mathbf{x}^t|\mathbf{y}^o) \propto p(\mathbf{x}^t)p(\mathbf{y}^o|\mathbf{x}^t). \tag{4.3}$$

Kalman filters generally assume Gaussian background and observation error distributions: $p(\mathbf{x}^t) \sim N(\mathbf{x}^b, \mathbf{B})$ and $p(\mathbf{y}^o|\mathbf{x}^t) \sim N(H(\mathbf{x}^t), \mathbf{R})$, respectively. Here, $\mathbf{x}^b$ is the background state obtained by feeding the prior analysis state $\mathbf{x}^a$ into (4.2), $\mathbf{B}$ is the background error covariance matrix, and $\mathbf{R}$ is the observation error covariance matrix. Thus, maximizing (4.3) is equivalent to minimizing the cost function:

$$\begin{aligned} J(\mathbf{x}) &= J^b(\mathbf{x}) + J^o(\mathbf{x}) \\ &= \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^b\right)^T \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}^b\right) + \frac{1}{2}\left(\mathbf{y}^o - H(\mathbf{x})\right)^T \mathbf{R}^{-1}\left(\mathbf{y}^o - H(\mathbf{x})\right). \end{aligned} \tag{4.4}$$

Here we have replaced $\mathbf{x}^t$ by $\mathbf{x}$ to reflect that the truth is unknown when we minimize $J(\mathbf{x})$. Notice that the background term $J^b(\mathbf{x})$ is quadratic in $\mathbf{x}$ as a consequence of Gaussian distribution assumption. For linear H, the observation term $J^o(\mathbf{x})$ is quadratic in $\mathbf{x}$ as well. For a nonlinear observation operator H, Kalman filter typically make a linear approximation in order to analytically approximate the minimum of the cost function (see Section 4.2).

A classical approach that is employed by the National Centers for Environmental Prediction (NCEP) does not make (except in the course of numerical minimization) a linear approximation to H but uses the same $\mathbf{B}$ for each analysis; this method is known as 3D-VAR [23]. However, the background uncertainty can vary considerably from time to time, so it is desirable to allow $\mathbf{B}$ to vary from one analysis to the next. The extended Kalman filter [12] dynamically varies the background covariance matrix using the linear tangent model of (4.2). However, this background covariance matrix update is not practical for large models. Evensen [9] introduced an ensemble Kalman filter (EnKF), which samples the background distribution with an ensemble of background states $\{\mathbf{x}^{b(i)}, i = 1, \ldots, k\}$. The ensemble size k is typically much less than the model dimension m. In the last decade, many variations of EnKF have been introduced [1, 5, 16, 17, 22, 32, 33, 42].

In this chapter, we introduce a non-quadratic (hence non-Gaussian) convex and symmetric background cost term $J^b(\mathbf{x})$ into the computational framework of the Ensemble Transform Kalman Filter [5], and our local version (See Chapter 3). The local filter is designed to be computationally efficient for large systems, and this is not affected significantly by the non-Gaussian modification, which changes only the computations done within the relatively low-dimensional ensemble space. We choose the non-Gaussian background term $J^b(\mathbf{x})$ to grow linearly as $\mathbf{x} \to \infty$, corresponding to a distribution with exponential rather than Gaussian tail. This allows the filter to weight the observations more heavily than the Gaussian filter in cases when they disagree significantly from background. We will show that our non-Gaussian filter is superior for infrequent observations and that it handles model error better than ETKF.

In Section 4.2, we derive ETKF from the cost function (4.4). In Section 4.3, we introduce a non-quadratic $J^b(\mathbf{x})$ into the computational framework of Section 4.2. In Section 4.4, we present preliminary results obtained for the 3-variable [24] model and, using a local filter, for the 40-dimensional [27] model. We show results in both a perfect-model scenario, where the "truth" is generated from (4.2) and the filter uses (4.2) with the same parameter set μ as its "model", and in scenarios with deterministic model error, where the filter uses a different parameter set. Finally, we conclude with a short summary in Section 4.5.

4.2 Variational Formulation of Ensemble Transform Kalman Filter

Ensemble Kalman filters approximate the true state $\mathbf{x}^t$ by an ensemble whose mean and covariance represent respectively an estimate of $\mathbf{x}^t$ and the uncertainty in the estimate. In the cost function (4.4), we replace the background state $\mathbf{x}^b$ by the sample mean $\bar{\mathbf{x}}^b$ of the background ensemble, and the background error covariance matrix $\mathbf{B}$ by the sample covariance matrix

$$\mathbf{P}^b = (k-1)^{-1}\mathbf{X}^b(\mathbf{X}^b)^T, \tag{4.5}$$

where k is the number of ensemble members and

$$\mathbf{X}^b = [\mathbf{x}^{b(1)} - \bar{\mathbf{x}}^b | \mathbf{x}^{b(2)} - \bar{\mathbf{x}}^b | \dots | \mathbf{x}^{b(k)} - \bar{\mathbf{x}}^b] \tag{4.6}$$

is the $m \times k$ matrix of background ensemble perturbations. Notice that this approximation is problematic for $k < m$ since $\mathbf{P}^b$ is not a full rank matrix, and hence is not invertible. However, $(\mathbf{P}^b)^{-1}$ is well-defined on the "ensemble subspace" spanned by the columns of $\mathbf{X}^b$. Thus, ensemble Kalman filters minimize the cost function on this subspace where it is well-defined.

Let us employ a pre-condition (or a coordinate change) by expressing the deviation of a state $\mathbf{x}$ from the background mean state $\bar{\mathbf{x}}^b$ as a linear combination of the background ensemble of perturbations. Using the notation of Section 3.2.1, we write

$$\mathbf{x} = \bar{\mathbf{x}}^b + \mathbf{X}^b\mathbf{w} \tag{4.7}$$

where the weight $\mathbf{w} \in \mathbb{R}^k$ is to be determined, and we approximate the observation vector corresponding to the model state $\mathbf{x}$ by:

$$H(\mathbf{x}) = H(\bar{\mathbf{x}}^b + \mathbf{X}^b\mathbf{w}) \approx H(\bar{\mathbf{x}}^b) + \mathbf{Y}^b\mathbf{w}. \tag{4.8}$$

Here the ith column vector of the $s \times k$ matrix $\mathbf{Y}^b$ is the deviation of $H(\mathbf{x}^{b(i)})$ from its ensemble average; that is,

$$\mathbf{Y}^b = [H(\mathbf{x}^{b(1)}) - \bar{\mathbf{y}}^b | H(\mathbf{x}^{b(2)}) - \bar{\mathbf{y}}^b | \ldots | H(\mathbf{x}^{b(k)}) - \bar{\mathbf{y}}^b] \tag{4.9}$$

with $\bar{\mathbf{y}}^b = \frac{1}{k}\sum_{i=1}^{k} H(\mathbf{x}^{b(i)})$.[1] Replacing $\mathbf{x}^b$ with $\bar{\mathbf{x}}^b$ and $\mathbf{B}$ with $\mathbf{P}^b$, and using (4.5), (4.7), and (4.8), reduce the cost function (4.4) to:

$$J(\mathbf{w}) = \frac{1}{2}(k-1)\mathbf{w}^T\mathbf{w} + \frac{1}{2}\left(\mathbf{y}^o - H(\bar{\mathbf{x}}^b) - \mathbf{Y}^b\mathbf{w}\right)^T \mathbf{R}^{-1}\left(\mathbf{y}^o - H(\bar{\mathbf{x}}^b) - \mathbf{Y}^b\mathbf{w}\right). \tag{4.10}$$

[1] If H is linear, then $\bar{\mathbf{y}}^b = H(\bar{\mathbf{x}}^b)$, but for nonlinear H these quantities are different. One could use either $\bar{\mathbf{y}}^b$ or $H(\bar{\mathbf{x}}^b)$ in (4.8). We always use $\bar{\mathbf{y}}^b$ in forming $\mathbf{Y}^b$, so that the sum of the columns of $\mathbf{Y}^b$ is zero.

That is, we reduce the m-dimensional minimization problem to a k-dimensional minimization problem, assuming the ensemble size k is less than the number m of model variables. Notice that in the $\mathbf{w}$ coordinate system, the background error covariance matrix becomes the identity and hence we do not have to invert it.

The minimum of (4.10) is obtained by setting

$$\nabla J(\mathbf{w}) = (k-1)\mathbf{w} - (\mathbf{Y}^b)^T\mathbf{R}^{-1}\left(\mathbf{y}^o - H(\bar{\mathbf{x}}^b) - \mathbf{Y}^b\mathbf{w}\right) = \mathbf{0}.$$

The solution of this equation is the analysis weight vector

$$\mathbf{w}^a = \tilde{\mathbf{P}}^a(\mathbf{Y}^b)^T\mathbf{R}^{-1}(\mathbf{y}^o - H(\bar{\mathbf{x}}^b)), \tag{4.11}$$

where

$$\tilde{\mathbf{P}}^a = \left((k-1)\mathbf{I} + (\mathbf{Y}^b)^T\mathbf{R}^{-1}\mathbf{Y}^b\right)^{-1}. \tag{4.12}$$

The analysis error covariance matrix $\tilde{\mathbf{P}}^a$ in the ensemble space is the inverse of the Hessian of the cost function (4.10) [10, 44]. The analysis state is obtained by substituting (4.11) into (4.7):

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b\mathbf{w}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b\tilde{\mathbf{P}}^a(\mathbf{Y}^b)^T\mathbf{R}^{-1}(\mathbf{y}^o - H(\bar{\mathbf{x}}^b)). \tag{4.13}$$

In the case that H is linear, equations (4.12) and (4.13) are equivalent to the standard Kalman filter equations, which minimize J in closed form.

To complete the analysis, we generate an analysis ensemble of model states whose mean is $\bar{\mathbf{x}}^a$ and whose error covariance matrix in the model space is $\mathbf{P}^a = \mathbf{X}^b\tilde{\mathbf{P}}^a(\mathbf{X}^b)^T$. In this chapter, we update the ensemble using

$$\mathbf{x}^{a(i)} = \bar{\mathbf{x}}^a + \mathbf{X}^b\mathbf{W}^{a(i)} = \bar{\mathbf{x}}^b + \mathbf{X}^b(\mathbf{w}^a + \mathbf{W}^{a(i)}), \tag{4.14}$$

where $\mathbf{W}^{a(i)}$ is the ith column of the symmetric square root of $(k-1)\tilde{\mathbf{P}}^a$ (see Section 3.2.1 and 3.4.2).

Algorithm 1 ETKF pseudo-code with Kalman filter equations

```
1. Generate $\mathbf{x}^{b(i)}$ at the current analysis time by feeding $\mathbf{x}^{a(i)}$ from the previous analysis time into model (4.2).
2. Form the background ensemble average $\bar{\mathbf{x}}^b$ and matrix of perturbations $\mathbf{X}^b$ given by (4.6).
3. Form $\mathbf{Y}^b$ according to (4.9).
4. Evaluate $\tilde{\mathbf{P}}^a$ using (4.12) and $\mathbf{w}^a$ using (4.11).
5. Take the symmetric square root of $(k-1)\tilde{\mathbf{P}}^a$ and call its columns $\mathbf{W}^{a(i)}$.
6. Compute the analysis ensemble members $\mathbf{x}^{a(i)}$ at the current analysis time with (4.14).
```

Now we give step-by step pseudo-codes of ETKF: one uses Kalman filter formulas (see Algorithm 1) and the other uses a variational formulation (see Algorithm 2). Notice that the difference between both algorithms is only in Step 4. Regarding Step 4b, the Hessian of (4.10) is the same at every point, but when we generalize to a nonquadratic cost function, it becomes important where we evaluate the Hessian. Of course, in the quadratic case, Algorithm 1 is computationally faster. However, our main goal is to generalize the ETKF for non-Gaussian background error distributions where the Kalman filter equations do not apply. In the next section, we adopt the framework given in Algorithm 2 for a different background cost function term that is still convex and symmetric, with the same Hessian as the quadratic background term at $\mathbf{w} = 0$. Hereafter, by the Gaussian filter we mean ETKF, or

LETKF as described below, implemented by Algorithm 1.

Algorithm 2 ETKF pseudo-code with variational formulation

```
1. Generate $\mathbf{x}^{b(i)}$ at the current analysis time by feeding $\mathbf{x}^{a(i)}$ from the previous analysis time into model (4.2).
2. Form the background ensemble average $\bar{\mathbf{x}}^b$ and matrix of perturbations $\mathbf{X}^b$ given by (4.6).
3. Form $\mathbf{Y}^b$ according to (4.9).
4a. Minimize cost function (4.10) to obtain $\mathbf{w}^a$. One may use any unconstrained minimization scheme such as the conjugate gradient method.
4b. Evaluate $\tilde{\mathbf{P}}^a$ as the inverse of the Hessian of cost function (4.10) at $\mathbf{w}^a$.
5. Take the symmetric square root of $(k-1)\tilde{\mathbf{P}}^a$ and call its columns $\mathbf{W}^{a(i)}$.
6. Compute the analysis ensemble members $\mathbf{x}^{a(i)}$ at the current analysis time with (4.14).
```

For large systems, local analysis has been used for practical purposes and to prevent spurious correlations, caused by small ensemble size, of the model variables at grid points separated by a large distance [16, 17, 22, 33]. Here, we adopt the localization as in Section 3.3.1. In contrast with the global analysis, the local analysis performs a separate analysis at each model grid point, using observations only from a local region surrounding the grid point. The analysis ensembles computed at each grid point are combined to form a global analysis ensemble. Therefore, the analysis at each location reflects the observations within its neighborhood, which presumably are the most correlated observations. With this approach, each grid point is updated independently. Thus, analysis can be performed in parallel and so dramatically reduce the cost of its implementation.

4.3 Non-Gaussian Filter

In practice, even with a perfect model, ensemble Kalman filter are suboptimal due to model nonlinearity and finite ensemble size. Nonetheless, even for large systems one can obtain reasonable results with an ensemble of moderate size (less than 100) by spatially localizing the analysis [36, 16, 17, 22]. However, regardless of any localization employed, such as [33], [16], or [22], the nonlinearity of the model together with the small ensemble size still generally cause the analysis ensemble to underestimate its uncertainty. A common approach to overcome this problem is to inflate the variance of the covariance matrix [2].

As described earlier in Section 3.3.3, the variance inflation is applied to the background by multiplying the background covariance matrix $\tilde{\mathbf{P}}^b = (k-1)^{-1}\mathbf{I}$ in the ensemble space by a constant $\rho = 1 + r$ with $r > 0$. This changes the term $(k-1)\mathbf{I}$ of (4.12) to $(k-1)\mathbf{I}/(1+r)$. Thus in practice, we use Algorithm 1 with the following equation in place of (4.12):

$$\tilde{\mathbf{P}}^a = \left(\frac{(k-1)}{1+r}\mathbf{I} + (\mathbf{Y}^b)^T\mathbf{R}^{-1}\mathbf{Y}^b\right)^{-1}.$$

In the variational formulation (see Algorithm 2), the variance inflation changes the first term of the cost function $J(\mathbf{w})$ to

$$J^b(\mathbf{w}) = \frac{1}{2}\frac{(k-1)\mathbf{w}^T\mathbf{w}}{1+r}. \tag{4.15}$$

In a given scenario, we determine the value of r empirically, turning it to achieve the best results. If the best value of r is small, this suggests that the Gaussian assumption approximates the background statistics reasonably well. However, the best results are obtained with relatively large r when the observations are infrequent and when model errors are significant. In such cases, the Gaussian assumption probably does not fit the background statistics well. For such situations, we introduce a new background term by replacing (4.15) with

$$J^b(\mathbf{w}) = \frac{1}{2}\frac{(k-1)\mathbf{w}^T\mathbf{w}}{1+\alpha\sqrt{\mathbf{w}^T\mathbf{w}}}, \tag{4.16}$$

where α is a constant to be determined empirically. Notice that when $\mathbf{w}$ is small, (4.16) behaves like the quadratic function $\frac{1}{2}(k-1)\mathbf{w}^T\mathbf{w}$, and for large $\mathbf{w}$, (4.16) grows close to linearly. Function (4.16) corresponds to a Gaussian-like background error distribution with longer tails (see Figure 4.1 for an illustration). This non-Gaussian error distribution approaches a Gaussian error distribution when $\alpha \to 0$. On the other hand, the tail distribution gets thicker as α increases. Furthermore, this symmetric function remains convex with the same Hessian $(k-1)\mathbf{I}$ at $\mathbf{w} = 0$ for all α. This non-Gaussian filter can be easily implemented by applying Algorithm 2 to the following cost function instead of (4.10):

$$J(\mathbf{w}) = \frac{1}{2}\frac{(k-1)\mathbf{w}^T\mathbf{w}}{1+\alpha\sqrt{\mathbf{w}^T\mathbf{w}}} + \frac{1}{2}\left(\mathbf{y}^o - H(\bar{\mathbf{x}}^b) - \mathbf{Y}^b\mathbf{w}\right)^T\mathbf{R}^{-1}\left(\mathbf{y}^o - H(\bar{\mathbf{x}}^b) - \mathbf{Y}^b\mathbf{w}\right). \tag{4.17}$$

Notice that one when minimizing J, one can easily compute its gradient analytically.

Based on the discussion above, we see that (4.17) is very close to (4.10) when $\mathbf{w}$ is small, but the background term grows more slowly when $\mathbf{w}$ is large. Small

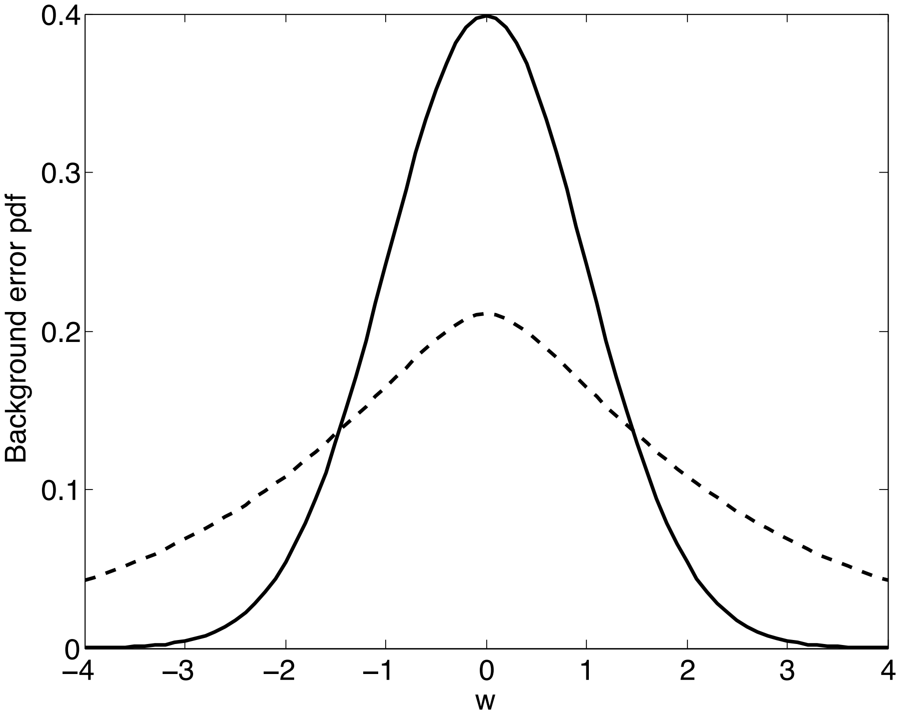

Figure 4.1: Density of a one-dimensional Gaussian (solid) error distribution with mean 0 and variance 1 and non-Gaussian (dashes) error distribution with $\alpha = 1$. The standard Gaussian density is proportional to $\exp\left\{-\frac{1}{2}\mathbf{w}^T\mathbf{w}\right\}$, while the non-Gaussian pdf is proportional to $\exp\left\{-\frac{1}{2}\frac{\mathbf{w}^T\mathbf{w}}{1+\alpha\sqrt{\mathbf{w}^T\mathbf{w}}}\right\}$.

$\mathbf{w}$ corresponds to model states close to the background mean $\bar{\mathbf{x}}^b$. Thus, if the background mean agrees well with the observations, both (4.10) and (4.17) will be minimized for small $\mathbf{w}$, and will produce very similar analyses. But when the background mean and observations differ significantly, (4.17) will be minimized for larger $\mathbf{w}$, and hence allows a larger analysis increment than (4.10).

Notice that we still use the linear approximation (4.8) to the observation operator in (4.17). While in principle there is no reason to force the observation term in (4.17) to be quadratic, in practice using $H(\bar{\mathbf{x}}^b + \mathbf{X}^b\mathbf{w})$ in the cost function could significantly complicate its minimization. With (4.17) we maintain a simple dependence on $\mathbf{w}$ that allows us to compute $J(\mathbf{w})$ and its partial derivatives efficiently during the minimization procedure. Indeed, we can compute the partial derivatives analytically.

4.4 Results

4.4.1 Numerical Simulations on the Lorenz-63 Model

In this section, we compare the performance of our proposed non-Gaussian filter with the Gaussian filter described in Section 4.2 for the three-dimensional [24] model:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz, \end{aligned} \tag{4.18}$$

where the parameter set μ (as in (4.2)) is the triplet (σ, b, r). We integrate the model using the fourth order Runge-Kutta method with time step 0.01.

In our numerical simulations, we generate the true state by running the model for 500 non-dimensional time units (i.e., 50,000 steps) with $\mu = (10, 8/3, 28)$, which results in a "butterfly"-like attractor [24]. We create infrequent "observations" every 0.5 time units[2] by adding Gaussian noise with mean 0 and variance 4 to each coordinate of the true state $\mathbf{x}^t$. Hence, the observation operator H is linear, equal to the identity, and the observation error covariance matrix R is a diagonal matrix with all diagonal components equal to 4.

We consider three cases: perfect model (no model error) by setting the forecast parameter set $\mu_f = \mu$, small model error by setting $\mu_f = (10, 8/3, 30)$, and large model error with $\mu_f = (10, 8/3, 35)$. Table 4.1 shows the Root-Mean-Square difference between analysis and true states for each coordinate, averaged over time and over 10 simulations; we refer to this RMS error as the analysis error hereafter. Each simulation was based on a different trajectory for the "true" state but we used the same trajectories and associated "observations" for each choice of the filter and model error.

All results use an ensemble of size $k = 10$. In the Gaussian filter experiments, we use $r = 4.5$ for the no model error case and $r = 5.5$ for experiment with small model error and r=10.5 for large model error. The non-Gaussian filter simulations with no model error and small model error use $\alpha = 2$, and with large model error uses $\alpha = 10$. In cases of small and no model errors, these values yielded the smallest

[2] A full oscillation around one of the "butterfly" wings corresponds to roughly 1 time unit.

	Gaussian No model error (r=4.5)	Non-Gaussian $\alpha = 2$
x	1.38	1.22
y	1.68	1.63
z	1.97	1.63
	Small model error (r=5.5)	$\alpha = 2$
x	1.55	1.33
y	1.84	1.70
z	2.00	1.77
	Large model error (r=10.5)	$\alpha = 10$
x	2.13	1.91
y	2.05	1.84
z	2.26	1.93

Table 4.1: RMS analysis error of Gaussian and non-Gaussian filters on the Lorenz-63 model. The filters are run with ensemble size k=10 and observation error 2. The analysis is performed every 50 steps with time step $\Delta t = 0.01$/step. In the no model error case, the forecast model parameter set is similar to the true model parameter set, $\mu_f = \mu = (10, 8/3, 28)$. Small and large model errors are introduced by setting $\mu_f = (10, 8/3, 30)$ and $\mu_f = (10, 8/3, 35)$, respectively.

analysis error among the valued we tried ($r = 1, 1.5, 2.5, 3.5, 4.5, 5.5, \ldots, 9.5$ and $\alpha = 1, 2, 4, 8, 10$). For the large model error, we stop tuning r since we can replace $r \to \infty$ and the Gaussian filter is equivalent to direct insertion, i.e., directly using the observations as analysis, for which the analysis error would equal the RMS observation error 2. In fact, Table 4.1 shows that the Gaussian filter with $r = 10.5$ is close to but no better than direct insertion. Table 4.1 also shows that the non-Gaussian filter performs better than the Gaussian filter by about 10% in all three cases of small, large and no model error experiments.

We also measured the variability of the analysis error in time by computing the standard deviation of $|\bar{x}^a - x^t|$, $|\bar{y}^a - y^t|$, and $|\bar{z}^a - z^t|$. In Table 4.2, we see that the analysis errors of the non-Gaussian filter are roughly 15% less varied than those

	Gaussian No model error (r=4.5)	Non-Gaussian $\alpha = 2$
x	0.99	0.81
y	1.08	1.03
z	1.32	1.03
	Small model error (r=5.5)	$\alpha = 2$
x	1.09	0.86
y	1.21	1.11
z	1.24	1.07
	Large model error (r=10.5)	$\alpha = 10$
x	1.36	1.17
y	1.32	1.13
z	1.42	1.15

Table 4.2: Variability of the analysis error for the experiments of of Table 4.1. For direct insertion this quantity is about 1.83. For each coordinate, x, y, z, we computed the square root of the variance of $|\bar{x}^a - x^t|$, $|\bar{y}^a - y^t|$, and $|\bar{z}^a - z^t|$, respectively.

of the Gaussian filter. For direct insertion, the analysis error variability is about 1.83 in each coordinate, which is significantly larger than both filters in all cases except the Gaussian filter with large model error. Thus the analysis results from the non-Gaussian filter are more consistent in time than those from the Gaussian filter, and both filters performed more consistently than direct insertion.

The filter analysis also shows a further advantage over direct insertion when we consider the forecasts they generate. An important feature of ensemble-based data assimilation is that it naturally yields initial conditions for an ensemble forecast, and we find that forecasting from each analysis ensemble member and averaging the forecasts yields better results than making a single forecast from the analysis ensemble mean. Thus for both filters, we measure their forecast error by calculating the RMS difference between the true state and the mean of the ensemble forecast, where each ensemble member is generated by feeding each analysis ensemble member to the model (4.2). In Figure 4.2, we show the RMS forecast errors as functions of time. We graph the errors on a logarithmic scale, so that the distance between two curves represents a relative (percentage) difference between their forecast errors. We observe that after the first few time steps, the forecast errors of direct insertion method (dotted) grow faster than those produced by two filters. In the small and no model error simulations, the forecast error of direct insertion saturates around 8.51, similar to the climatological error (see top and middle images of Figure 4.2). In these cases, notice also that the forecast errors of the non-Gaussian filter (dashes) remain lower than those of Gaussian filter (solid) at all times. As we mentioned before in Table 4.1, for large model error the Gaussian filter analysis error (or forecast error at time 0 in Figure 4.2) is significantly worse than that of direct insertion. On the other hand, the non-Gaussian filter analysis error is only slightly better than that of direct insertion. However, the forecast error of direct insertion again grows more quickly than that of the filters. Here, the model error is large enough so that the skill of the ensembles produced from both filters are indistinguishable after time 0.1 unit.

4.4.2 Numerical Simulations on the Lorenz-96 Model

In the previous simulations on the Lorenz-63 model, we see that our non-Gaussian filter yields better results than the Gaussian filter for a simple temporal chaotic dynamical system, in a case when a large amount of variance inflation is

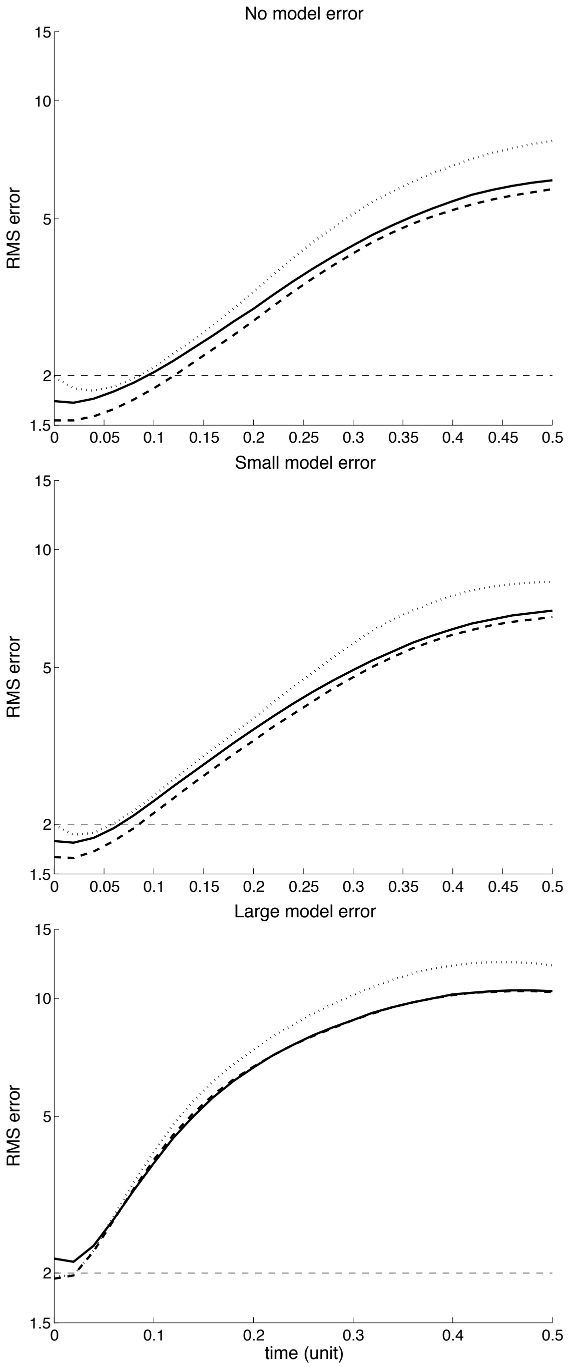

Figure 4.2: Forecast errors as functions of time for the experiments in Table 4.1: no model error (top), small model error (middle) and large model error (bottom). We use a logarithmic scale on the vertical axis so that the distance between two curves represents the ratio between their errors. The forecast errors of the initial conditions from direct insertion (dotted) are significantly worse than both the Gaussian (solid) and non-Gaussian (dashed) filters. The observation RMS error is 2 (thin dashed horizontal line).

needed for the Gaussian filter. Now, we want to show that this result also holds in a simple spatio-temporal chaotic dynamical system. For this purpose, we choose the 40-dimensional [27] model and perform a local analysis as described at the end of Section 4.2 and Section 3.3.1.

The Lorenz-96 model represents an "atmospheric variable" x at m equally spaced points around a circle of constant latitude. The jth component is propagated in time following the differential equation

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + \mu \tag{4.19}$$

where $j = 1, ..., m$ 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 μ 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 [27] and [28], we choose the external forcing to be $\mu = 8$ and the number of spatial elements to be $m = 40$. 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 [27]. We use a fourth-order Runge-Kutta scheme for time integration of (4.19) with time step $\Delta t = 0.05$, and we observe the system state and perform an analysis every 6 time steps. This is relatively infrequent in the following sense. 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 6 time step of our model integration corresponds roughly to performing it every 1.5 days in a global weather model.

In our numerical simulations, we compute the true state by running the model for 1,000 time units (i.e., 20,000 steps). We generate observations every 0.3 time units (6 steps) by adding Gaussian noise with mean 0 and variance 1 to each coordinate of the true state $\mathbf{x}^t$. Hence, the observation operator H is linear and the observation error covariance matrix R is the identity matrix. The analysis error is defined as in the previous section, except now the RMS error is not only temporally but also spatially averaged.

We consider two cases: perfect model (no model error) by setting the forecast parameter μ_f to be equal to the true state parameter $\mu = 8$ and with model error by setting $\mu_f = 8.5$. All results use an ensemble of size $k = 10$ and we perform the local analysis at each grid point with localization distance $d = 6$ as in Section 3.4.2; (that is, the local analysis at each grid point uses all observations from $2d + 1 = 13$ grid points centered at the analysis point). In the Gaussian filter experiments, we obtained the lowest analysis error with $r = 1.0$ for the no model error case and $r = 1.6$ for experiment with model error. For the non-Gaussian filter simulations without and with model error, we found the best results with $\alpha = 0.6$ and 0.8, respectively.

Figure 4.3 show the forecast errors as functions of time (as in Figure 4.2 but with linear vertical-axis). We observe that the forecast errors of the non-Gaussian filter (dashed) are about 5% and 10% lower than those produced by the Gaussian filter (solid) with and without model errors, respectively. We also calculate the time

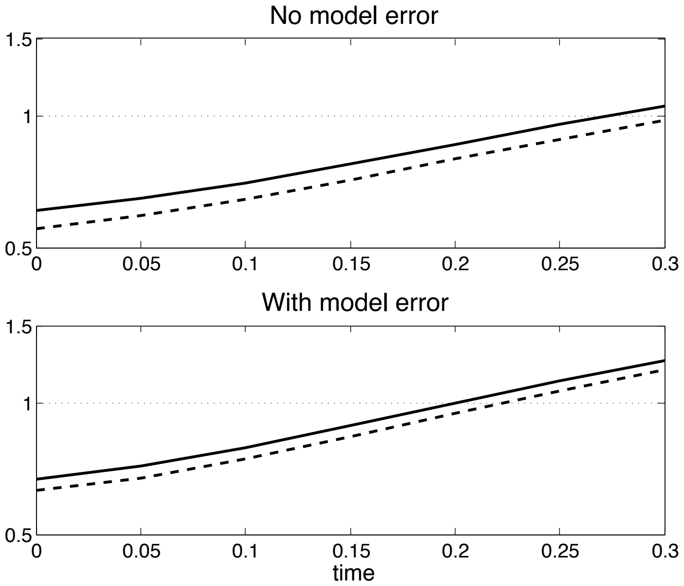

Figure 4.3: Forecast error (again on a logarithmic scale) as a function of time: with no model error (top) and with model error (bottom). The forecast errors of the Gaussian filter (solid) are larger than those produced by the non-Gaussian filter (dashed) by about 10% in the case of no model error and 5% in the presence of model error. The observation RMS error is 1 (dotted horizontal line). The Gaussian filter uses variance inflation coefficient $r = 1$ and 1.6 without and with model error, respectively. The non-Gaussian filter uses $\alpha = 0.6$ and 0.8 without and with model errors, respectively.

variability of the forecast error as a function of time. The forecast error variability is the standard deviation of $\|\bar{\mathbf{x}}^b - \mathbf{x}^t\|$, where $\bar{\mathbf{x}}^b$ denotes the ensemble average of the forecasts produced by propagating each analysis ensemble member as initial condition. We find that the non-Gaussian filter reduces the variation for about 30% (see Figure 4.4) both with and without model errors.

4.5 Summary

In this chapter, we review a Gaussian ensemble Kalman filter (LETKF) and extend it to a non-Gaussian filter by introducing a non-quadratic background term

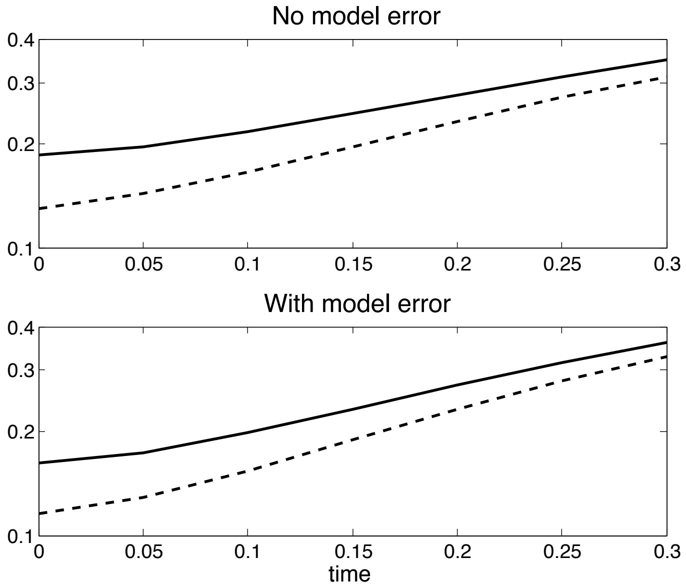

Figure 4.4: Forecast variability (again on a logarithmic scale) as a function of time: with no model error (top) and with model error (bottom). The forecast errors variability of the Gaussian filter (solid) are about 30% larger than those produced by the non-Gaussian filter (dashed).

in the cost function that is minimized by the analysis. This symmetric and convex non-quadratic term (4.16) is chosen to closely match the usual quadratic term near its minimum but grow more slowly away from the minimum. This cost term corresponds to a distribution with longer tails than the Gaussian distribution.

Our results from the Lorenz-63 model simulations show that when the observations are sufficiently infrequent, the non-Gaussian filter yields a significantly better analysis, reducing the analysis error by about 10%. The non-Gaussian filter also reduces the analysis error time variations and the forecast errors compared to both the Gaussian filter and direct insertion.

In our simulations on the 40-variable Lorenz-96 model, we performed the analysis locally as in LETKF. We find that with moderately infrequent observations, the local non-Gaussian filter yields analysis and forecast errors that are 5-10% lower than that of the local Gaussian filter. We also found that the local non-Gaussian filter analysis errors are about 30% less variable in time than those of the local Gaussian filter in cases with and without model error. For each model, the non-Gaussian and Gaussian filters produced analyses of similar quality when we tested them with more frequent observations. When the observations became more frequent, less variance inflation is needed, and as we discussed in Section 4.3, the two methods become more similar. For the low-dimensional Lorenz-63 model, we see significant differences between the two filters when the optimal variance inflation is larger than 500% for the Gaussian filter, whereas for the higher-dimensional Lorenz-96 model, we see a noticeable difference in a case where only 100% variance inflation is needed. This suggests to us that for high-dimensional systems, cases in which the non-Gaussian filter is advantageous to the Gaussian filter may be more common.

Finally, one can consider a more general class of non-quadratic terms than (4.16). The functions

$$\frac{(k-1)\mathbf{w}^T\mathbf{w}}{\gamma + \alpha(\mathbf{w}^T\mathbf{w})^{\beta}}, \tag{4.20}$$

include (4.16), with $\beta = 1/2$ and $\gamma = 1$, and have similar properties for other values of β and γ. In particular, the convexity of this function retained for $0 \leq \beta \leq 1/2$. We tested several different parameters and found that for $\beta = 1/4$ we did not get a better result than what we showed with $\beta = 1/2$. However, one may want to explore these parameters more thoroughly for other models.

Chapter 5

Conclusions and Work in Progress

In this dissertation, we show the importance of employing data assimilation in numerical weather prediction, introduce an efficient strategy for implementing ensemble Kalman filter, and generalize this efficient scheme to allow a the non-Gaussian forecast error distribution.

In Chapter 2, we see a distinct error growth pattern from a standard chaotic dynamical system. Here, the error growth pattern is dominated by two regimes (first with a fast regime for relatively small scales and then a slow regime for larger scales). Figure 2.1 suggests that in the fast regime, correcting errors in initial condition is not very helpful since the doubling time is less than 1 day. The results of Bengtsson and Hodges [4] indicate that the current state of art of weather forecasts have about 2 days of doubling time. In Figure 2.1, this is in the slow regimes. Therefore, reducing the errors in initial condition can lead to significant forecast improvements.

In Chapter 3, we derive a local filter, LETKF, based on the Local Ensemble Kalman Filter (LEKF) [32, 33] but whose local analysis is similar to the global analysis of the Ensemble Transform Kalman Filter of Bishop et al. [5]. We obtain results of similar quality to the LEKF but with a more efficient scheme. Specifically, we replace the coordinate system used in the analysis from using the singular vectors of the background error covariance matrix as a basis to using ensemble of perturbations as a basis. In addition to the results in this dissertation, we have seen a comparable results between LEKF and LETKF, in a simplified parametrized primitive equation (SPEEDY) model of Molteni [30] with 10^5 degrees of freedom with LETKF performing twice as fast as the LEKF. A second theme of this chapter is that with this choice of basis for analysis, we simplify the 4D-EnKF of Hunt et al. [20] mathematically. In a separate work, we compare this scheme with 4D-VAR (the current state of art of four dimensional data assimilation) and we find that while the average quality of the forecasts resulting from both schemes are comparable, the variability of the analysis produced by the 4D-LETKF is significantly lower than that of 4D-VAR. Recent results from the SPEEDY model show a significant advantage of this scheme over the simpler approach where all the asynchronous observations are assumed to occur at the analysis time and another approach where the differences between the observations and the model state at asynchronous times are assumed to occur at the analysis time. The latter approach is called FGAT (First-Guess at the Appropriate Time) by Huang et al. [18] who apply it to a High Resolution Limited Area Modeling (HIRLAM) within a 3D-VAR.

In Chapter 4, we introduce into the LETKF framework an exponential background error distribution with longer tails rather than the Gaussian distribution. The main motivation is that in practice we generally readjust the Gaussian background error distribution by inflating the entire distribution, but we think that the main improvement comes from inflating the tail. Especially when the observation time interval is sparse, the optimal inflation factor can be quite large. Our non-Gaussian filter is developed to retain the computational efficiency of the local ensemble filter in Chapter 3. The computational framework only replaces the Kalman filter equations with the conjugate-gradient minimization scheme, which is relatively inexpensive when the minimization is performed in the ensemble space. We speculate that this non-Gaussian filter is attractive for assimilation problems with low frequency of observations availability, such as in the ocean data assimilation, or for which high amounts of variance inflation are used due to model error.

BIBLIOGRAPHY

[1] J.L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129:2884–2903, 2001.

[2] J.L. Anderson and S.L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127:2741–2758, 1999.

[3] E. Aurell, G. Boffetta, A. Crisanti, G. Paladin, and A. Vulpiani. Predictability in systems with many degrees of freedom. *Phys. Rev. E*, 52(2337), 1996.

[4] L. Bengtsson and K.I. Hodges. A note on atmospheric predictability. *Tellus A*, 58:154–157, 2006.

[5] C.H. Bishop, B. Etherton, and S.J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter part I: the theoretical aspects. *Monthly Weather Review*, 129:420–436, 2001.

[6] G. Burgers, P.J. van Leeuwen, and G. Evensen. On the analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.

[7] J.G. Charney. Geostrophic turbulence. *Journal of Atmospheric Science*, 28:1087–1095, 1971.

[8] J. Derber and Coauthors. Changes to the 1998 NCEP operational MRF model analysis-forecast system. *NOAA/NWS Tech. Procedure Bull.*, 449(16pp), 1998.

Available from office of meteorology, NWS, 1325 East-West Hwy., Silverspring, MD 20910.

[9] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.

[10] M. Fisher and P. Courtier. Estimating the covariance matrix of analysis and forecast error in variational data assimilation. Tech. Memo. 220, ECMWF, 1995.

[11] S.J. Fletcher and M. Zupanski. A framework for data assimilation which allows for non-Gaussian errors. *ftp://ftp.cira.colostate.edu/milija/papers/Steven_nongauss.pdf*, 2005.

[12] M. Ghil, S. Cohn, J. Tavantzis, K. Bube, and E. Isaacson. Applications of estimation theory to numerical weather prediction. In A.L. Bengtsson, M. Ghil, and E. Kallen, editors, *Dynamics meteorology: data assimilation methods*, pages 139–224, New York, 1981. Springer Verlag.

[13] T.M. Hamill, J.S. Whitaker, and C. Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman Filter. *Monthly Weather Review*, 129:2776–2790, 2001.

[14] J. Harlim, M. Oczkowski, J.A. Yorke, E. Kalnay, and B.R. Hunt. Convex error growth patterns in a global weather model. *Physical Review Letters*, 94(228501), 2005.

[15] James R. Holton. *An introduction to dynamic meteorology.* Academic Press, 3rd edition, 1992.

[16] P.L. Houtekamer and H.L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126:796–811, 1998.

[17] P.L. Houtekamer and H.L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129:123–137, 2001.

[18] X.-Y. Huang, K.S. Morgensen, and X. Yang. First-guess at the appropriate time: the HIRLAM implementation and experiments. In *Proceedings, HIRLAM workshop on variational data assimilation and remote sensing, Helsinki, Finland*, pages 28–43, 2002.

[19] B.R. Hunt. Efficient data assimilation for spatiotemporal chaos: a local ensemble transform Kalman filter. *http://arxiv.org/abs/physics/0511236*, 2005.

[20] B.R. Hunt, E. Kalnay, E.J. Kostelich, E. Ott, D.J. Patil, T. Sauer, I. Szunyogh, J.A. Yorke, and A.V. Zimin. Four-dimensional ensemble Kalman filtering. *Tellus A*, 56:273–277, 2004.

[21] E. Kalnay. *Atmospheric modeling, data assimilation, and predictability.* Cambridge University Press, 2003.

[22] C.L. Keppenne. Data assimilation into a primitive-equation model with a parallel ensemble Kalman filter. *Monthly Weather Review*, 128:1971–1981, 2000.

[23] A.C. Lorenc. A global three-dimensional multivariate statistical interpolation scheme. *Monthly Weather Review*, 109:701–721, 1981.

[24] E.N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Science*, 20:130–141, 1963.

[25] E.N. Lorenz. The predictability of a flow which possesses many scales of motion. *Tellus*, 21:289–307, 1969.

[26] E.N. Lorenz. Atmospheric predictability experiments with a large numerical model. *Tellus*, 34:505–513, 1982.

[27] E.N. Lorenz. Predictability - A problem Partly Solved. In *Proceedings on predictability, held at ECMWF on 4-8 September 1995*, pages 1–18, 1996.

[28] E.N. Lorenz and K.A. Emanuel. Optimal sites for supplementary weather observations: simulation with a small model. *Journal of the Atmospheric Science*, 55:399–414, 1998.

[29] J. Marshall and F. Molteni. Toward a dynamical understanding of planetary-scale flow regimes. *Journal of the Atmospheric Sciences*, 50(12):1792–1818, 1993.

[30] F. Molteni. Atmospheric simulations using a GCM with simplified physical parametrizations, I: Model climatology and variability in multi-decadal experiments. *Climate Dynamics*, 20:175–191, 2003.

[31] G.D. Nastrom and K.S. Gage. A climatology of atmospheric wavenumber spectra of wind and temperature observed by commercial aircraft. *Journal of Atmospheric Science*, 42:950–960, 1985.

[32] E. Ott, B.R. Hunt, I. Szunyogh, A.V. Zimin, E.J. Kostelich, M. Corrazza, E. Kalnay, D.J. Patil, and J.A. Yorke. Exploiting local low dimensionality of the atmospheric dynamics for efficient ensemble Kalman filtering. *http://arxiv.org/abs/physics/0203058v3*, 2002.

[33] E. Ott, B.R. Hunt, I. Szunyogh, A.V. Zimin, E.J. Kostelich, M. Corrazza, E. Kalnay, and J.A. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56:415–428, 2004.

[34] D.J. Patil, Brian R. Hunt, Eugenia Kalnay, James A. Yorke, and Edward Ott. Local low dimensionality at atmospheric dynamics. *Physical Review Letters*, 86(26):5878–5881, 2001.

[35] J.G. Sela. Spectral modelling at the National Meteorological Center. *Monthly Weather Review*, 108:1279–1292, 1980.

[36] I. Szunyogh, E.J. Kostelich, G. Gyarmati, D.J. Patil, B.R. Hunt, E. Kalnay, E. Ott, and J.A. Yorke. Assessing a local ensemble Kalman filter: perfect model experiments with the NCEP global model. *Tellus A*, 57:528–545, 2005.

[37] O. Talagrand. A study if the dynamics of four-dimensional data assimilation. *Tellus*, 33:43–60, 1981.

[38] M.K. Tippett, J.L. Anderson, C.H. Bishop, T.M. Hamill, and J.S. Whitaker. Ensemble square-root filters. *Monthly Weather Review*, 131:1485–1490, 2003.

[39] Zoltan Toth and Eugenia Kalnay. Ensemble forecasting at NMC: the generations of perturbations. *Bull. Amer. Meteor. Soc.*, 74:2317–2330, 1993.

[40] Zoltan Toth and Eugenia Kalnay. Ensemble forecasting at NCEP and the breeding method. *Monthly Weather Review*, 125:3297–3319, 1997.

[41] X. Wang, C.H. Bishop, and S.J. Julier. Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble? *Monthly Weather Review*, 132(7):1590–1605, 2004.

[42] J.S. Whitaker and T.M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130:1913–1924, 2002.

[43] F. Zhang, C. Snyder, and R. Rotunno. Effects of moist convection on mesoscale predictability. *Journal of the Atmospheric Science*, 60:1173–1185, 2003.

[44] M. Zupanski. Maximum likelihood ensemble filter: theoretical aspects. *Monthly Weather Review*, 133:1710–1726, 2005.