A Moment Matching Particle Filter for Nonlinear Non-Gaussian Data Assimilation

Jing Lei * and Peter Bickel

Department of Statistics, University of California, Berkeley

* *Corresponding author address:* Jing Lei, Department of Statistics, UC Berkeley, 367 Evans Hall, Berkeley, CA 94720.

E-mail: jinglei@stat.berkeley.edu

ABSTRACT

The ensemble Kalman filter is now an important component of ensemble forecasting. While using the linear relationship between the observation and state variable makes it applicable for large systems, relying on linearity introduces non-negligible bias since the true distribution will never be Gaussian. We review the ensemble Kalman filter from a statistical perspective and analyze the sources of its bias. We then propose a de-biasing method called the nonlinear ensemble adjustment filter. This new filter transforms the forecast ensemble in a statistically principled manner so that the updated ensemble has the desired mean and variance which is calculated by importance sampling. We also show that the new filter is easily localizable and hence potentially useful for large systems. The new filter is tested through various experiments on Lorenz 63 system and Lorenz 96 system, showing promising performance when compared with other Kalman filter and particle filter variants. The results show that the new filter is stable and accurate for very challenging situations such as nonlinear, high dimensional system with sparse observations.

1. Introduction

The ensemble Kalman filter (EnKF, (Evensen 1994, 2003, 2007)) has become a popular tool for data assimilation because of its computational efficiency and flexibility (Houtekamer and Mitchell 1998; Anderson 2001; Whitaker and Hamill 2002; Ott et al. 2004; Evensen 2003; Mitchell and Houtekamer 2009).

Although many variants have been developed, the EnKF update approximates the probability distributions of the forecast state vector, the observation, and the updated state vector by Gaussian distributions. Such a Gaussian approximation allows a linear update which makes the EnKF applicable for many large-scale data assimilation problems. However, in reality such a Gaussian approximation and linear update will introduce systematic bias because the true distributions can be significantly non-Gaussian and the relationship between the observation and state might be nonlinear.

A filtering approach that is adaptive to nonlinearity and non-Gaussianity is the particle filter (Gordon et al. 1993; van Leeuwen 2003). However, it is known that the ordinary particle filter requires a prohibitively large ensemble size to avoid collapsing (Snyder et al. 2008). On the other hand, the particle filter suffers from sample impoverishment when the forecast model is deterministic. Both the efficiency problem and sample impoverishment problem can be alleviated by reducing the dimensionality. Therefore, a major challenge for particle filters in geophysical systems is localization. The particle filter updates the forecast ensemble directly from likelihood function and the covariance matrix is not used in the update, therefore traditional covariance tapering techniques for the EnKF is not applicable for the particle filter. The sliding window localization method used in Ott et al. (2004)

seems feasible but the re-sampling/re-weighting step of ordinary particle filter breaks the connection between overlapping local windows. van Leeuwen (2009) gives an introduction of particle filter for data assimilation.

In this article we propose a new particle filtering method which enables localization. It combines some of the advantages of both the EnKF and the particle filter. First, one can view the EnKF as a linear regression of the state vector on the observation vector (Anderson 2003). The updated state is obtained by using the true observation as predictor in the fitted linear model. Under nonlinear and non-Gaussian models, such a linear update is biased in both location (the posterior expectation) and shape (the posterior higher order moments). The bias in location is a direct consequence of using linear regression in a nonlinear model. Our method uses importance sampling to estimate the conditional expectation of the state given the observation which substantially reduces the bias in location. Unlike the particle filter, which uses importance sampling to estimate the whole posterior distribution, the new method uses importance sampling only to estimate the posterior mean. Although, in principle, this process is still subject to collapse because of using importance sampling, it lends itself more easily to localization because it does not involve any re-weighting or re-sampling steps.

Xiong et al. (2006) and Nakano et al. (2007) also use importance sampling to update the mean and covariance, and transform the forecast ensemble so that the updated ensemble has asymptotically the desired mean and covariance. In particular, Xiong et al. (2006) proposes a particle filter with Gaussian re-sampling (PFGR) which uses a deterministic transform on the forecast ensemble, which is very similar to the ensemble square root filter. This method depends on the particle filter to estimate the updated mean and covariance, which is hard if the dimensionality of state-space is moderately high. On the other hand, Nakano et al. (2007) proposes a merging particle filter (MPF) which generates a multiple sample points using importance sampling and re-sampling, and each updated particle are obtained by a fixed linear combination of a group of sample points. This method gives good simulation results but in order to avoid collapsing, the importance sampling and re-sampling step cannot collapse in the first place, which is usually not the case in high dimension state spaces if the dynamics is deterministic. We compare the performance of our method with both PFGR and MPF in the Lorenz 63 system. In higher dimension systems with deterministic dynamics, such as the Lorenz 96 system, both PFGR and MPF degenerates.

The rest of this article is organized as follows: In Section 2 we review the EnKF and the particle filter. In Section 3 we examine the sources of bias in the EnKF and introduce the NonLinear Ensemble Adjustment Filter (NLEAF). In Section 4 the NLEAF algorithm is tested and compared with other methods in Lorenz 63 system and Lorenz 96 system. Some final remarks are given in Section 5.

2. Ensemble filters

Filtering algorithms for data assimilation usually work sequentially. There are two major steps in each recursion. In the forecasting step, a forecast (prior) ensemble is obtained by applying the forecast model to each update (posterior) ensemble member that is produced at previous time. In the update step, the forecast ensemble is modified to incorporate the information provided by the new observation. We focus on the update step assuming that the forecast can be done in a standard way.

Formally, suppose the uncertainty of the forecast ensemble can be represented by a random variable $\mathbf{x}$ with probability density function $p_{\mathrm{f}}(\cdot)$, where the subindex "f" stands for "forecast". Assume that the observation $\mathbf{y}$ is given by

$$\mathbf{y} = h(\mathbf{x}) + \boldsymbol{\epsilon},$$

where the observation mechanism $h(\cdot)$ may be nonlinear and $\boldsymbol{\epsilon}$ is the observation noise with probability density function $g(\cdot)$ which could be non-Gaussian. The likelihood function of $\mathbf{x}$ given $\mathbf{y}$ is then $g(\mathbf{y} - h(\mathbf{x}))$. The optimal way to update the state distribution is Bayes rule, in which the optimal updated probability density function of the state variable is then:

$$p_{\mathrm{a}}(\mathbf{x}) = \frac{p_{\mathrm{f}}(\mathbf{x})g(\mathbf{y} - h(\mathbf{x}))}{\int p_{\mathrm{f}}(\mathbf{x}')g(\mathbf{y} - h(\mathbf{x}'))d\mathbf{x}'},$$

where the subindex "a" stands for "analysis" (update).

However, a closed form solution is available only for a few special cases such as when h is linear and $p_{\mathrm{f}}(\cdot)$, $g(\cdot)$ are Gaussian (the Kalman filter), or when $\mathbf{x}$ is discrete (hidden Markov models). In ensemble filtering, p_{f} and p_{a} are approximated by a discrete set (ensemble) of sample points (particles), and the ensemble is propagated by the forecast model and updated according to the observation at each time. We recall two typical ensemble filtering methods, the particle filter and the EnKF.

a. The particle filter

Gordon et al. (1993) proposes the particle filter, using the importance sampling technique followed by a re-sampling step. Given the forecast ensemble $\{\mathbf{x}_{\mathrm{f}}^1, \ldots, \mathbf{x}_{\mathrm{f}}^n\}$ as a random sample from p_{f}, and the observation $\mathbf{y}^{\mathrm{o}}$, a simple particle filter algorithm works as follows:

PF1. Evaluate the likelihood for each forecast ensemble member: $w^j = g(\mathbf{y}^{\mathrm{o}} - h(\mathbf{x}_{\mathrm{f}}^j))$ for $j = 1, \ldots, n$.

PF2. For $j = 1, \ldots, n$, sample the updated ensemble member $\mathbf{x}_{\mathrm{a}}^j$ independently from $\{\mathbf{x}_{\mathrm{f}}^j\}_{j=1}^n$ with weights proportional to $(w^1, \ldots, w^n)$.

Re-weighting the particles according to their likelihoods is called *importance sampling* which dates back at least to Hammersley and Handscomb (1965). The particle filter is statistically consistent in the sense that when the ensemble size goes to infinity, the updated ensemble will be exactly a random sample from p_{a} (Künsch 2005). However, in high dimensional situations, the ordinary particle filter requires a very large ensemble size to search the whole state space and the ensemble tends to collapse in a few steps. Snyder et al. (2008) give a quantitative estimation of the rate of collapse on a very simple Gaussian model.

Another potential problem of the particle filter is the issue of degeneracy or sample impoverishment. When re-sampling is used, there will be inevitably a loss of diversity of distinct ensemble members during the update. If the dynamics is deterministic, the updated sample will soon have few distinct ensemble members. A remedy to this problem is perturbing the updated ensemble members with small random noises, which is known as the regularized particle filter (Musso et al. 2001). However, the perturbation introduces another source of noise, which may impact the performance negatively even in low dimensional problems.

However, due to its natural advantage in dealing with nonlinear non-Gaussian problems, the particle filter is still a promising tool for data assimilation. Many re-sampling methods have been proposed to improve the efficiency of particle filters. Examples include, but not limited to, Xiong et al. (2006) and Nakano et al. (2007). Of course, finding a good proposal density in the re-sampling step is key. The extent to which this is possible in very high dimensional geophysical systems remains to be analyzed but see Chorin and Tu (2009); van Leeuwen (2010)

b. The ensemble Kalman filter

A widely used approach in data assimilation is the EnKF (Evensen 1994; Burgers et al. 1998). Instead of the particle filter which tries to capture the whole posterior distribution, the EnKF approximates the prior and posterior distributions by Gaussians and updates only the first two moments of the state variable. Assuming a linear observation:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon},$$

with Gaussian noise $\boldsymbol{\epsilon} \sim Norm(0, \mathbf{R})$. Given the forecast ensemble $\{\mathbf{x}_{\mathrm{f}}^1, \ldots, \mathbf{x}_{\mathrm{f}}^n\}$ and true observation $\mathbf{y}^{\mathrm{o}}$, the EnKF update with perturbed observation works as follows:

En1. Compute the forecast sample covariance matrix: $\mathbf{P}_{\mathrm{f}} = n^{-1}\sum_{j=1}^{n}(\mathbf{x}_{\mathrm{f}}^j - \bar{\mathbf{x}}_{\mathrm{f}})(\mathbf{x}_{\mathrm{f}}^j - \bar{\mathbf{x}}_{\mathrm{f}})^T$, where $\bar{\mathbf{x}}_{\mathrm{f}} = n^{-1}\sum_{j=1}^{n}\mathbf{x}_{\mathrm{f}}^j$ is the forecast ensemble mean, and the superscript T means matrix transpose.

En2. Estimate the linear regression coefficient of $\mathbf{x}$ on $\mathbf{y}$: $\mathbf{K} = \mathbf{P}_{\mathrm{f}}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{\mathrm{f}}\mathbf{H}^T + \mathbf{R})^{-1}$. This is usually done by multiplying the sample covariance between the forecast ensemble and background observations and the inverse of the sample variance of background observations.

En3. Generate background observations: $\mathbf{y}^j = \mathbf{H}\mathbf{x}_{\mathrm{f}}^j + \boldsymbol{\epsilon}^j$, with $\boldsymbol{\epsilon}^j \sim Norm(0, \mathbf{R})$.

En4. Update sample points: $\mathbf{x}_{\mathrm{a}}^j = \mathbf{x}_{\mathrm{f}}^j + \mathbf{K}(\mathbf{y}^{\mathrm{o}} - \mathbf{y}^j)$.

It is also known that using perturbed observations will introduce some sampling bias especially for small ensembles. A remedy is use some systematic sampling scheme to make sure the $\boldsymbol{\epsilon}^j$'s have zero mean, covariance $\mathbf{R}$, and zero correlation with $\mathbf{x}^j$'s. Covariance inflation is also used to mitigate the bias in the forecast ensemble which is generated by a biased analysis ensemble. There has been a vast literature on different variants and applications of the EnKF. Please see Evensen (2003) for a nice review and Mitchell and Houtekamer (2009) for more recent developments.

Another closely related ensemble filter also updates the first two moments of the forecast ensemble but in a deterministic manner, known as the Kalman square root filter (Anderson 2001; Bishop et al. 2001; Whitaker and Hamill 2002). There have been many works on the comparison between the perturbed observation filter and the square root filter (Lawson and Hansen 2004; Lei et al. 2010). Although deterministic filters have no sampling error and some are equivalent to the Kalman filter under Gaussian linear models, the bias caused by the nonlinearity and non-Gaussianity still remains even when the ensemble size is large.

One key feature of the EnKF is that it updates each particle directly, without re-weighting or re-sampling, which makes it applicable to large-scale problems. In the next section we introduce another filter with the same property but with reduced bias under nonlinear, non-Gaussian models.

3. A New Nonlinear Non-Gaussian Ensemble Filter

a. Why does the ensemble Kalman filter work?

An explanation of the EnKF (with perturbed observation) is based on the simple fact: if $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian, then the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$ depends on $\mathbf{y}$ only through the mean. That is, for any $\mathbf{y}$, let $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$ be the posterior mean of $\mathbf{x}$ given $\mathbf{y}$, then the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$ is $Norm(\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}), \mathbf{P}_{\mathrm{a}})$, where $\mathbf{P}_{\mathrm{a}}$ does not depend on $\mathbf{y}$. By the construction of background observation $\mathbf{y}^j$, $(\mathbf{x}_{\mathrm{f}}^j, \mathbf{y}^j)$ is jointly Gaussian. Therefore, recall that $\boldsymbol{\mu}_{\mathrm{f}}$ is the forecast ensemble mean, $\mathbf{x}_{\mathrm{f}}^j - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^j) = (\mathbf{I} - \mathbf{KH})(\mathbf{x}_{\mathrm{f}}^j - \boldsymbol{\mu}_{\mathrm{f}}) + \mathbf{K}\boldsymbol{\epsilon}^j$ is a random draw from $Norm(0, \mathbf{P}_{\mathrm{a}})$ and $\mathbf{x}_{\mathrm{a}}^j = \mathbf{x}_{\mathrm{f}}^j - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^j) + \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}})$ is a random draw from $Norm(\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}}), \mathbf{P}_{\mathrm{a}})$, the true posterior distribution. On the other hand, the joint Gaussianity also implies that $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$ is a linear function of $\mathbf{y}$ with linear coefficient $\mathbf{K}$ as given in ($\mathbf{En}2$). Thus the update can be written as $\mathbf{x}_{\mathrm{a}}^j = \mathbf{x}_{\mathrm{f}}^j + \mathbf{K}(\mathbf{y}^{\mathrm{o}} - \mathbf{y}^j)$. Note that by definition we have $\mathbf{y}^j = \mathbf{H}\mathbf{x}^j + \boldsymbol{\epsilon}^j$. As a result, the update formula can be written as $\mathbf{x}_{\mathrm{a}}^j = \mathbf{x}_{\mathrm{f}}^j + \mathbf{K}\left((\mathbf{y}^{\mathrm{o}} - \boldsymbol{\epsilon}^j) - \mathbf{H}\mathbf{x}^j\right)$, which is exactly the same as the "observation perturbation" used in equation (13) of Burgers et al. (1998). In the present paper we use the background observation $\mathbf{y}^j$ for the convenience of a unified argument for both Gaussian linear and non-Gaussian, nonlinear models.

In a non-Gaussian nonlinear model, the EnKF will be biased for two reasons. First, $\mathbf{K}(\mathbf{y}^{\mathrm{o}} - \mathbf{y}^j) \neq \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}}) - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^j)$ since $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$, the conditional expectation of $\mathbf{x}$ given $\mathbf{y}$, is no longer linear in $\mathbf{y}$. Second, the re-centered conditional random variables $\mathbf{x}_{\mathrm{f}}^j - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^j) + \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}})$ will not have the correct variance and higher moments. The first source of bias will result in a larger distance between the true state and the updated ensemble mean whereas the second source of bias will affect the shape of the updated ensemble, which might be problematic as the bias is propagated by the dynamics.

b. The NLEAF algorithm with first order correction

We now introduce the nonlinear ensemble adjustment filter (NLEAF) as a de-biasing alternative to the EnKF. It requires no assumptions on the prior distribution of $\mathbf{x}$ and works for a general observation function $h(\cdot)$ and observation noise distribution $g(\cdot)$. The basic idea is to estimate $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$ using importance sampling, instead of linear fitting as in the EnKF. Let $p_{\mathrm{f}}(\cdot)$ and $p_{\mathrm{a}}(\cdot)$ be the forecast (prior) and updated (posterior given $\mathbf{y}$) densities, respectively. Then we have, for any $\mathbf{y}$,

$$\begin{aligned} p_{\mathrm{a}}(\mathbf{x}) &= \frac{p_{\mathrm{f}}(\mathbf{x})g(\mathbf{y}-h(\mathbf{x}))}{\int p_{\mathrm{f}}(\mathbf{x}')g(\mathbf{y}-h(\mathbf{x}'))d\mathbf{x}'} \\ \Rightarrow \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}) &= \frac{\int \mathbf{x}p_{\mathrm{f}}(\mathbf{x})g(\mathbf{y}-h(\mathbf{x}))d\mathbf{x}}{\int p_{\mathrm{f}}(\mathbf{x})g(\mathbf{y}-h(\mathbf{x}))d\mathbf{x}}. \end{aligned} \tag{1}$$

The importance sampling estimator of the conditional expectation is given by

$$\hat{\boldsymbol{\mu}}_{\mathrm{a}}(\mathbf{y}) = \frac{\sum_{j=1}^{n} \mathbf{x}_{\mathrm{f}}^{j} g(\mathbf{y}-h(\mathbf{x}_{\mathrm{f}}^{j}))}{\sum_{j=1}^{n} g(\mathbf{y}-h(\mathbf{x}_{\mathrm{f}}^{j}))}. \tag{2}$$

Given the forecast ensemble $\{\mathbf{x}_{\mathrm{f}}^{1}, \ldots, \mathbf{x}_{\mathrm{f}}^{n}\}$ and observation $\mathbf{y}^{\mathrm{o}}$, the NLEAF update works as follows

NL1. Generate background observations $\mathbf{y}^{j} = h(\mathbf{x}_{\mathrm{f}}^{j}) + \boldsymbol{\epsilon}^{j}$, with $\boldsymbol{\epsilon}^{j}$ independently sampled from probability density function $g(\cdot)$.

NL2. Estimate the conditional expectation $\hat{\boldsymbol{\mu}}_{\mathrm{a}}(\mathbf{y})$ using (2) for $\mathbf{y}$ equals the true observation $\mathbf{y}^{\mathrm{o}}$ and all $\mathbf{y}^{j}$, $j = 1, \ldots, n$.

NL3. Update sample: $\mathbf{x}_{\mathrm{a}}^{j} = \mathbf{x}_{\mathrm{f}}^{j} + \hat{\boldsymbol{\mu}}_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}}) - \hat{\boldsymbol{\mu}}_{\mathrm{a}}(\mathbf{y}^{j})$.

This algorithm improves the EnKF by using importance sampling in estimating the conditional expectations. It is known that under mild conditions the importance sampling estimator $\hat{\boldsymbol{\mu}}_{\mathrm{a}}(\mathbf{y})$ converges to $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$ as the ensemble size tends to infinity (Künsch 2005). Therefore $\mathbf{x}_{\mathrm{a}}^{j}$ is centered approximately at $\mu_{\mathrm{a}}(\mathbf{y}^{\mathrm{o}})$, since $\mathbf{x}_{\mathrm{f}}^{j}$ is centered approximately at $\mu_{\mathrm{a}}(\mathbf{y}^{j})$ conditioning on $\mathbf{y}^{j}$. As a result, the first source of bias in the EnKF is reduced. On the other hand, it also keeps the simplicity of the EnKF, avoiding the re-weighting and re-sampling steps used by the particle filter. This is particularly useful for localization techniques that depend on local state vectors, for example, the sliding-window localization used by Ott et al. (2004). When applying to overlapping local state vectors, the absence of re-sampling and re-weighting can keep the spatial smoothness across overlapping local vectors. As a result, one would expect the NLEAF algorithm to be applicable for high-dimensional problems with a better accuracy than the EnKF. Before demonstrating its performance in Section 4, we state two important extensions of the NLEAF.

c. *Higher order corrections*

The NLEAF update described above only adjusts the posterior mean. In fact the same idea can be applied to get higher order corrections. For example, an NLEAF with second order correction uses importance sampling to estimate the posterior covariance of $\mathbf{x}$ given $\mathbf{y}$, denoted as $\mathbf{P}_{\mathrm{a}}(\mathbf{y})$. Theoretically we have

$$\mathbf{P}_{\mathrm{a}}(\mathbf{y}) = \int (\mathbf{x} - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}))(\mathbf{x} - \boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y}))^{T} p_{\mathrm{a}}(\mathbf{x}) d\mathbf{x},$$

where $p_{\rm a}(\mathbf{x})$ is defined as in Eq. (1). Then the importance sampling estimator for $\mathbf{P}_{\rm a}(\mathbf{y})$ is

$$\hat{\mathbf{P}}_{\rm a}(\mathbf{y}) = \frac{\sum_{j=1}^{n}(\mathbf{x}_{\rm f}^{j} - \hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y}))(\mathbf{x}_{\rm f}^{j} - \hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y}))^{T} g(\mathbf{y} - h(\mathbf{x}_{\rm f}^{j}))}{\sum_{j=1}^{n} g(\mathbf{y} - h(\mathbf{x}_{\rm f}^{j}))}, \tag{3}$$

where $\hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y})$ is obtained using (2). The update with second order correction is

$$\mathbf{x}_{\rm a}^{j} = \hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y}^{\rm o}) + \hat{\mathbf{P}}_{\rm a}(\mathbf{y}^{\rm o})^{1/2}\hat{\mathbf{P}}_{\rm a}(\mathbf{y}^{j})^{-1/2}\left(\mathbf{x}_{\rm f}^{j} - \hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y}^{j})\right).$$

To understand this algorithm, note that when the ensemble size is large, $\hat{\boldsymbol{\mu}}_{\rm a}(\mathbf{y}) \approx \boldsymbol{\mu}_{\rm a}(\mathbf{y})$ and $\hat{\mathbf{P}}_{\rm a}(\mathbf{y}) \approx \mathbf{P}_{\rm a}(\mathbf{y})$ for all $\mathbf{y}$, then $\mathbf{x}_{\rm f}^{j} - \boldsymbol{\mu}_{\rm a}(\mathbf{y}^{j})$ has mean zero and covariance $\mathbf{P}_{\rm a}(\mathbf{y}^{j})$. Therefore the updated ensemble member $\mathbf{x}_{\rm a}^{j}$ has approximately mean $\boldsymbol{\mu}_{\rm a}(\mathbf{y}^{\rm o})$ and covariance $\mathbf{P}_{\rm a}(\mathbf{y}^{\rm o})$. As shown in numerical experiments, this higher order correction does improve the accuracy. It even outperforms the particle filter in some settings with deterministic dynamics. As a trade-off, it requires a larger ensemble size and is computationally more intensive than the first order NLEAF because of the covariance matrix estimation and inversion. We also note this is actually a stochastic version of the Gaussian re-sampling particle filter proposed by Xiong et al. (2006).

d. *NLEAF with unknown likelihood function*

Another useful extension of the NLEAF is to use nonlinear regression methods to estimate $\boldsymbol{\mu}_{\rm a}(\mathbf{y})$. In some practical problems, the relationship between the observation $\mathbf{y}$ given the state vector $\mathbf{x}$ might be complicated so that the density function of $\mathbf{y}$ given $\mathbf{x}$ is not available in analytic form, for example, when $\mathbf{y}$ is generated by a black-box function. One practical example is satellite radiance. Satellite radiance is related to temperature, humidity and other trace variables depending on the wavelength. But no simple direct analytical relationship is known between the state and the observation. In this case, all we have is the forecast ensemble paired with the background observations: $\{(\mathbf{x}_{\mathrm{f}}^j, \mathbf{y}^j)\}_{j=1}^n$. One may use regression methods to estimate the conditional moments of $\mathbf{x}$ given $\mathbf{y}$. For example, the EnKF uses a linear regression to estimate $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$. One can also use more general methods, such as polynomial regression, to handle nonlinearity. More specifically, one may estimate $\boldsymbol{\mu}_{\mathrm{a}}(\mathbf{y})$ as a function of $\mathbf{y}$ by minimizing over all quadratic functions $m(\cdot)$ under the square loss $\sum_{j=1}^n ||m(\mathbf{y}^j) - \mathbf{x}_{\mathrm{f}}^j||_2^2$. This is could be a promising method for nonlinear data assimilation problems that lacks knowledge of observation generating mechanism.

4. Numerical experiments

a. The Lorenz 63 system

The Lorenz 63 system is a three dimensional model determined by an ordinary differential equation system

$$d\mathbf{z}_\tau(1)/d\tau = -\sigma \mathbf{z}_\tau(1) + \sigma \mathbf{z}_\tau(2), \tag{4}$$

$$d\mathbf{z}_\tau(2)/d\tau = -\mathbf{z}_\tau(1)\mathbf{z}_\tau(3) + \rho \mathbf{z}_\tau(1) - \mathbf{z}_\tau(2), \tag{5}$$

$$d\mathbf{z}_\tau(3)/d\tau = \mathbf{z}_\tau(1)\mathbf{z}_\tau(2) - \beta \mathbf{z}_\tau(3), \tag{6}$$

where $\mathbf{z}_\tau$ is the three-dimensional state vector describing a simplified flow of heated fluid with τ being the continuous time index and parameters are set as $\beta = 8/3$, $\rho = 28$ and $\sigma = 10$.

In simulation the system is discretized using the fourth order Runge-Kutta method. Let

$\mathbf{x}_t = \mathbf{z}_{\Delta t}$, for all integers t, with Δ being the step size. A larger value of Δ indicates a more non-linear relationship between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. A hidden true orbit $\{\mathbf{x}_t^*, t \geq 0\}$ and observation $\mathbf{y}_t^{\mathrm{o}} = \mathbf{x}_t^* + \boldsymbol{\epsilon}_t$ are generated with starting point $\mathbf{x}_0^*$ randomly chosen from the attractor and independent observation noise $\boldsymbol{\epsilon}_t$ with mean 0 and covariance $\theta^2\mathbf{I}$. At the starting time, the initial ensemble $\{\mathbf{x}_0^j\}_{j=1}^n$ is obtained by perturbing $\mathbf{x}_0^*$ with random noise. At each time $t \geq 1$, filtering methods are applied and the updated ensemble average is used as the best single estimate of $\mathbf{x}_t^*$. The major evaluation criterion is the root mean squared error (RMSE): $||\bar{\mathbf{x}}_{t,\mathrm{a}} - \mathbf{x}_t^*||_2/\sqrt{d}$, where $||\cdot||_2$ is the Euclidean norm, $d = 3$ is the dimensionality of the state space, and $\bar{\mathbf{x}}_{t,\mathrm{a}}$ is the updated ensemble mean at time t. For a measurement of the sample spread (sharpness) we look at $[\mathrm{trace}(\hat{\mathbf{P}}_{t,\mathrm{a}})/d]^{1/2}$ in comparison with the RMSE, where $\hat{\mathbf{P}}_{t,\mathrm{a}}$ is the sample covariance matrix of the updated ensemble at time t. In addition, we use the percentage of $\mathbf{x}_t^*(3)$ being covered by the range between 0.025 and 0.975 quantiles (the sample 95% confidence interval) of the updated ensemble as a measurement of how well the updated ensemble covers the true state.

1) Simulation set-up

Here we look at two different observation error distributions. First we consider Gaussian observation noise which has been used widely in previous studies. In this case, each coordinate of $\boldsymbol{\epsilon}_t$ is independent Gaussian with mean 0 and variance θ^2 to the corresponding coordinate of state variable. On the other hand, in order to study a non-Gaussian observation noise we let $\boldsymbol{\epsilon}_t$ have independent coordinates with a common double exponential distribution. A double exponential density (also known as the Laplace distribution) with mean 0 and variance $2b^2$ is $p(z) = (2b)^{-1}\exp(-|z|/b)$. The EnKF variants use the noise variance $2b^2$ in the update and pretend the noise is still Gaussian, which would be biased because the true noise is non-Gaussian. The NLEAF update uses the true double exponential density as the likelihood.

The system is run for $T = 2000$ steps and at each time filtering methods are applied with an ensemble of size $n = 400$. The observation interval The number of steps is chosen to be 2000 throughout our presentation because the result is relatively easy and quick to reproduce. For both the Lorenz 63 system and the Lorenz 96 system (see below), we have done simulations with the number of time steps increased to 50000 and the results are the same. The system is propagated using a fourth order Runge-Kutta method with single step size 0.01. We studied two different lengths of observation intervals (denoted Δ), 0.02 and 0.05. The observations are obtained by perturbing the true state by a Gaussian (or double exponential) noise with variance θ^2. Three values of θ are tested: 0.5, 1, and 2.

The methods in comparison are: the EnKF with perturbed observation (EnKF), NLEAF with first order correction (NLEAF1), and NLEAF with second order correction (NLEAF2), particle filter (PF), particle filter with Gaussian re-sampling (Xiong et al. 2006, PFGR), and merging particle filter (Nakano et al. 2007, MPF). PFGR, and MPF uses inflation

$$\mathbf{x}_{\mathrm{a}}^{j} + (1+\delta)(\mathbf{x}_{\mathrm{a}}^{j} - \bar{\mathbf{x}}_{\mathrm{a}}) \tag{7}$$

with $\delta = 0.01$ to avoid sample degeneracy because of the deterministic nature of these algorithms. The PF uses a slightly different inflation method which replaces the jth updated ensemble member $\mathbf{x}_{\mathrm{a}}^{j}$ by $\mathbf{x}_{\mathrm{a}}^{j} + 2\delta \mathrm{Cov}(\mathbf{x}_{\mathrm{a}})^{1/2}\xi^{j}$, where $\mathrm{Cov}(\mathbf{x}_{\mathrm{a}}) =< \mathbf{x}_{\mathrm{a}}^{j}\mathbf{x}_{\mathrm{a}}^{j\mathrm{T}} > - < \mathbf{x}_{\mathrm{a}}^{j} >< \mathbf{x}_{\mathrm{a}}^{j} >^{\mathrm{T}}$ is the posterior covariance matrix of the state vector state covariance δ is a positive number that controls the amount of perturbation (it essentially plays the role of the covariance inflation coefficient in (7)). ξ^j is independent Gaussian noise with unit variance.

Remark 1. The EnKF used in all of the simulations is the standard single ensemble EnKF, which typically needs inflation in non-linear, high-dimensional applications. More sophisticated EnKF variants (Mitchell and Houtekamer (2009), for example) are also in use and can produce stable updated ensembles without using covariance inflation.

2) Results

The simulation results are summarized in Tables 1 and 2. The first number in each cell is the average RMSE over 2000 steps, the number in the parenthesis is the average sample spread measured as the square root of trace of the empirical covariance matrix scaled by $1/\sqrt{3}$ to match the RMSE. The number below them is the percentage of $x_t^*(3)$ being covered by the sample 95% confidence interval.

A good filtering method should produce 1) a small average RMSE and 2) an ensemble spread that well represents the ensemble mean error (Sacher and Bartello 2008). If the spread is too small, the updated ensemble is not likely to cover the true state; and if the spread is too large, the updated ensemble would be too non-informative. In other words, the true state should look like a random sample from the updated ensemble. Therefore, the sample confidence interval coverage of the true state would be a good indicator of the filter performance. Based on our experiment (see also Figures 1-6 in Sacher and Bartello (2008)), it is usually good to have the updated variance slightly larger than the RMSE.

As we see from the tables, the NLEAF gives unparalleled performance except for $\Delta = 0.05$

and $\theta = 2$, which is the hardest case with a very nonlinear dynamics and large observation noise. Even in this case, it loses only slightly to the particle filter in the RMSE but still provides tighter sample spreads with a reasonable confidence interval coverage.

The EnKF gives large RMSE's and negatively biased empirical confidence interval coverage due to its bias under nonlinearity and non-Gaussianity. As for the particle filter, in Section 2a we point out that it has the issue of sample impoverishment in deterministic systems. In our implementation, small random perturbations are added to the updated ensemble members to overcome this difficulty, increasing both the uncertainty of the update ensemble and the sample confidence interval coverage.

It is interesting to compare the performance between the PF and NLEAF1. When $\Delta = 0.05$, NLEAF1 gives slightly larger RMSE. Conversely, the NLEAF1 performs better than the PF when Δ is smaller. This is because when the step size is small there is less bias in the higher order moments so it does not lose much information by ignoring the higher order moments.

We observe that the NLEAF2 tends to produce a tight update ensemble which might fail to cover the true state. In practice one can mitigate this issue by inflating the updated ensemble covariance as in (7) for some small positive constant δ.

b. The Lorenz 96 system

The Lorenz 96 system (Lorenz 1996) is another common testbed for data assimilation algorithms (Bengtsson et al. 2003; Ott et al. 2004; Anderson 2007). The state vector has 40 coordinates evolving according to the following ordinary differential equations:

$$d\mathbf{z}_\tau(i)/d\tau = \big[\mathbf{z}_\tau(i+1) - \mathbf{z}_\tau(i-2)\big]\mathbf{z}_\tau(i-1) - \mathbf{z}_\tau(i) + 8, \quad \text{for } i = 1, \ldots, 40, \tag{8}$$

where $\mathbf{z}_\tau(0) = \mathbf{z}_\tau(40)$, $\mathbf{z}_\tau(-1) = \mathbf{z}_\tau(39)$ and $\mathbf{z}_\tau(41) = \mathbf{z}_\tau(1)$. This system mimics the evolution of some meteorological quantity at 40 equally spaced grid points along a latitude circle. The attractor has 13 positive local Lyapunov vectors. Similarly, the system is discretized with step size Δ: $\mathbf{x}_t = \mathbf{z}_{\Delta t}$, for all integers t.

In such a high-dimensional problem, the particle filter often collapses in a single step with all the weight concentrating on a single ensemble member. The EnKF also has difficulties when the sample size n is as small as a few tens if no dimension reduction techniques are used. Most dimension reduction techniques for the Lorenz 96 system are based on the fact that two coordinates of $\mathbf{x}_t$ have little dependence if they are far away in the physical space. For example, the correlation between $\mathbf{x}(1)$ and $\mathbf{x}(2)$ might be significant, but $\mathbf{x}(1)$ and $\mathbf{x}(20)$ would be nearly independent. As a result, the covariance tapering idea (Houtekamer and Mitchell 1998; Gaspari and Cohn 2001) is applicable. A more straightforward method is the sliding-window localization proposed by Ott et al. (2004) (see also Bengtsson et al. (2003)), which uses local observations to update local state coordinates, and the whole state vector is reconstructed by aggregating the local updates. Specifically, the state vector $\mathbf{x} = (\mathbf{x}(1), \ldots, \mathbf{x}(40))$ is broken down to overlapping local vectors $\mathbf{x}(N_j)$, $j = 1, \ldots, 40$, with $N_j = (j-l, \ldots, j, \ldots, j+l)$, for some positive integer l and all numbers being mod 40. Then each local vector $\mathbf{x}(N_j)$ is updated by the ensemble filter using local observations $\mathbf{y}(N_j)$ (assuming $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$). Therefore the each coordinate $\mathbf{x}(j)$ is updated simultaneously in $2l+1$ local vectors including $\mathbf{x}(N_{j-l}), \mathbf{x}(N_{j-l+1}), \ldots, \mathbf{x}(N_{j+l})$. The final update for $\mathbf{x}(j)$ is then obtained by averaging its updates in local vectors $\mathbf{x}(N_{j-k})$, $\mathbf{x}(N_{j-k+1})$, ..., $\mathbf{x}(N_{j+k})$ for some positive integer $k \leq l$.

Our study of the Lorenz 96 system consists of two different settings corresponding to two different levels of difficulty.

1) The hard case

(i) Simulation set-up

This setting has been considered by Bengtsson et al. (2003) as an early effort towards a high-dimensional nonlinear non-Gaussian filter. In this case, the system is run using fourth order Runge-Kutta method with single step size 0.05. We run the system and get 2000 observations with observation interval length $\Delta = 0.4$, which indicates a highly nonlinear non-Gaussian forecast ensemble. A sample of size $n = 400$ is used and observation $\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \boldsymbol{\epsilon}_t$, where $\mathbf{H} = (\mathbf{e}_1, \mathbf{e}_3, \ldots, \mathbf{e}_{39})^T$ with $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0)^T$ being the unit vector with all zero entries except the ith position, and $\boldsymbol{\epsilon}_t \sim Norm(0, \mathbf{I}_{20}/2)$. That is, only state coordinates with odd indices are observed with independent Gaussian noise of variance 0.5. Such a combination of incomplete observation, high nonlinearity, and high dimensionality poses great challenge to filtering algorithms.

Here we compare the performance of four methods: 1) NLEAF1, (NLEAF with first order correction); 2) NLEAF1q, (NLEAF using quadratic regression); 3) EnKF (the EnKF with perturbed observation); 4) XEnsF, a nonlinear filter using Gaussian mixture approximations to the prior and posterior distributions Bengtsson et al. (2003). Both NLEAF1 and NLEAF1q uses localization and averaging parameters $(l, k) = (3, 1)$. The inflation follows equation (7) with rate δ is 0.045 for NLEAF1 and NLEAF1q, and 0.005 for the EnKF.

(ii) Results

The results are summarized in Table 3.

We only look at the mean, median and standard deviation of the 2000 RMSE's since they are the only available measurements of performance reported for the XEnsF Bengtsson et al. (2003). Clearly the NLEAF1 gives the smallest and stablest RMSE on average. Note that in this setting the observation is still linear with Gaussian noise, the NLEAF outperforms other methods because of its adaptivity to nonlinearity. The performance of NLEAF1q is also surprisingly good. As introduced in Section 3d, the NLEAF1q algorithm pretends that the observation noise distribution is unknown and uses the forecast ensemble and background observations to estimate a quadratic regression function of $\mathbf{x}$ on $\mathbf{y}$. One might note that the difference in the RMSE seems not too large compared to the standard deviation. The reason is that the distribution of 2000 RMSE's are non-Gaussian with very heavy right tail, which can be seen from the difference between the mean and median.

It should be noted that in the result for XEnsF is quoted directly from Bengtsson et al. (2003), therefore the methods compared here might be using different observation sequences.

2) The easy case

(i) Simulation set-up

The easy case uses observation interval $\Delta = 0.05$, which makes the system much more linear than in the hard case. The choice of $\Delta = 0.05$ can be thought of as nominally equivalent to 6 hours in real-world time (Lorenz 1996). This setting allows a more comprehensive comparison with existing methods since it has been studied extensively in the literature, see Whitaker and Hamill (2002), Ott et al. (2004), Sakov and Oke (2008a,b) for examples. In particular, we can study the effect of the sample size n on filter performance, which is an important practical concern.

The system also runs to generate 2000 observations with a complete observation $\mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\epsilon}_t$ and double exponential observation error $\boldsymbol{\epsilon}$ since we are also interested in non-Gaussian situations.

We compare the NLEAF with first order correction (NLEAF1), the EnKF, and the local ensemble transform Kalan filter (Ott et al. 2004, LETKF). The LETKF is among the best performing ensemble filters in this setting with Gaussian observation noise. It uses the sliding-window localization and performs well especially when the sample size is very small. However, it is also reported that its performance can hardly be improved by a larger sample size. Other particle filter based methods either have difficualty in localization Xiong et al. (2006, PFGR) or diverge with limited ensemble size for deterministic dynamics Nakano et al. (2007, MPF).

The LETKF uses localization parameter $k = l = 3$ for ensemble size 10 and 20, $k = l = 4$ for ensemble size 50, 100, and no localization for ensemble size 400 (in this case it is just the Ensemble Transform Kalman filter, ETKF, Bishop et al. (2001)). It uses inflation with $\delta = 0.01$ for all ensemble sizes. The EnKF uses Cohn-Gaspari covariance tapering with $c = 5, 10$ for ensemble size 10 and 20, respectively, and no tapering is necessary for larger ensemble sizes. The inflation rate $\delta = 0.06, 0.05, 0.05, 0.03, 0.01$ for ensemble size 10, 20, 50, 100, 400, respectively. For NLEAF1 and NLEAF1q, the localization parameter is $k = l = 3$ for ensemble size 10, 20, and $k = l = 4$ for ensemble size 50, 100, 400; the inflation rate δ is 0.09 for $n = 11$, 0.05 for $n = 20$, and 0.01 for $n \geq 50$. All the inflation is as described as in (7).

(ii) Results

Results on the average RMSE is summarized in Figure 1 for a quick comparison. Some further comparison on the sample spread and sample confidence interval coverage is included in Table 4. In Table 4, the number in the parenthesis is the average sample spread measured as the square root of trace of the empirical covariance matrix scaled by $1/\sqrt{40}$ to match the RMSE. The number below them is the percentage of $\mathbf{x}_t^*(1)$ being covered by the sample 95% confidence interval. The methods in comparison are: the LETKF, the NLEAF with first order correction (NLEAF1), the first order NLEAF using quadratic regression (NLEAF1q), and the EnKF. An N/A entry indicates that the filter diverges (the whole sample is far away from the true orbit).

From Figure 1 and Table 4 we see that the NLEAF1 performs better than the LETKF as long as the sample size exceeds 20, providing both accurate point estimate and confidence interval for the true trajectory. In this setting the dynamics is pretty linear and NLEAF1 outperforms the LETKF mainly because of the non-Gaussian observation noise. On the other hand, the NLEAF1q does provide useful updates when the sample size is fairly large. This could be practically helpful when the noise distribution is unknown or observation mechanism has no analytical form. Moreover, one might also notice that the LETKF performs worse for $n = 400$ than for smaller values of n. This could be due to its vulnerability to the presence of outliers Lawson and Hansen (2004); Lei et al. (2010). That is, increasing the sample size also increases the chance of having outliers, which has a significant impact on the performance of Kalman square-root filters. The simulation results on the EnKF also supports our earlier statement that it has difficulty when the sample size is only a few tens. For larger sample sizes, it works better but the bias never goes away, as can be seen from the sample spread and confidence interval coverage.

5. Further discussions

This article demonstrates how simple statistical ideas can help design better filtering algorithms. The NLEAF algorithm inherits some key features of the EnKF by keeping track of each particle, which makes it easily applicable in high-dimensional problems with spatial structures such as the Lorenz 96 system. It also reflects the bias-variance trade-off principle. The EnKF is computationally stable but biased for nonlinear non-Gaussian problems, whereas the particle filter is consistent but unstable for moderate-scale problems. The NLEAF is somewhere in the middle: it uses the particle filter only up to the first and/or second moment, while avoiding re-weighting/re-sampling to maintain the stability. We believe this would be the key philosophy to deal with complicated data with limited computational budget.

Although the NLEAF works nicely for the Lorenz models. It does not completely overcome the collapse problem of the particle filter. Currently the implementation of NLEAF depends on the localization of the state vector as well as the observation. One might expect difficulties if the state vector is only sparsely observed. On the other hand, in our experiments the largest local window is of size 9, which is still a small number for realistic models such as a GCM. For higher dimensionality, the importance sampling is likely to collapse even for the first moment. One potential remedy for this problem could be to assimilate each observation sequentially, with the impact on each state coordinate adjusted according to its distance or correlation between the observation (Anderson 2003, 2007).

The NLEAF algorithm originates from a regression perspective to the EnKF. Such a perspective can be used in other ways to design new filtering methods. For example, in some high-dimensional problems, the correlation between state coordinates might not depend only on their physical distance and the covariance matrix of the state vector is sparse but not banded. As a result, the sliding-window localization method described in Section 4b is no longer valid. Then one can employ sparse regression methods to identify the set of "effective" neighborhood for each state coordinate.

Another direction of future work is the development of a deterministic NLEAF. The NLEAF udpate introduced here uses the background observations which involve random perturbations. This resembles the EnKF with perturbed observation which is subject to sampling error. It is possible to design a deterministic NLEAF update which computes the posterior moments using importance sampling or nonlinear regression methods, then the ensemble can be updated deterministically such that the posterior ensemble has the updated moments.

The implementation of all filtering algorithms involves some tuning parameters. In our simulation some of the filtering methods use covariance inflation, including the particle filter, MPF and PFGR in Lorenz 63 system and all methods in Lorenz 96 system[1]. The updated sample spread is inflated slightly to adjust for underestimation of the forecast uncertainty. Such an inflation will put less weight on the forecast sample and more weight on the observation during the Bayes inference. As a result, the amount of inflation is a tuning parameter. In our experiments on the Lorenz 63 system, the EnKF and NLEAF require little inflation due to the large sample size and the use of synthesized observations. However, the particle filter always requires some substantial inflation/perturbation to avoid sample impoverishment, which results in a larger RMSE. In high-dimensional problems such as the Lorenz 96 system, inflation is always used for all methods discussed in this article. In our experiment, the inflation rate ranges from 0.5% ($n = 400$) to 6.5% ($n = 10$) for the NLEAF1 algorithm. Other tuning parameters include the localization window size l and the aggregation window size k. We tried different combinations of (k, l) and chose the one with best performance. Choosing optimal tuning parameters is not the major concern of this article since our goal is to illustrate the NLEAF algorithm. In practice these tuning parameters can be chosen by a simple grid search under the constraints of available computation resources.

[1]Some of versions of EnKF may produce reliable updated ensembles without inflation, see Mitchell and Houtekamer (2009) for example.

Acknowledgments.

The authors would like to thank Jeff Anderson, Thomas Bengtsson, Doug Nychka and Chris Snyder for helpful comments. Lei and Bickel are supported by NSF DMS-0605236 and NSF DMS-0906808.

REFERENCES

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

Anderson, J. L., 2003: A local least squares framework for ensemble filtering. *Monthly Weather Review*, **131**, 634–642.

Anderson, J. L., 2007: Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. *Physica D*, **230**, 99–111.

Bengtsson, T., C. Snyder, and D. Nychka, 2003: Toward a nonlinear ensemble filter for high-dimensional systems : Application of recent advances in space-time statistics to atmospheric data. *J. Geophys. Res.*, **108(D24)**, STS2.1–STS2.10.

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

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

Chorin, A. and X. Tu, 2009: Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, **106**, 17 249–17 254.

Evensen, G., 1994: Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99(C5)**, 10 143–10 162.

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

Evensen, G., 2007: *Data assimilation: the ensemble Kalman filter*. Springer.

Gaspari, G. and S. Cohn, 2001: Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, **125**, 723–757.

Gordon, N., D. Salmon, and A. Smith, 1993: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, **140**, 107–113.

Hammersley, J. M. and D. C. Handscomb, 1965: *Monte Carlo Methods*. Methuen & Co.

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

Künsch, H. R., 2005: Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, **33**, 1983–2021.

Lawson, W. G. and J. A. Hansen, 2004: Implications of stochastic and deterministic filters as ensemble-based data assimilation methods in varying regimes of error growth. *Monthly Weather Review*, **132**, 1966–1981.

Lei, J., P. Bickel, and C. Snyder, 2010: Comparison of ensemble Kalman filters under non-Gaussianity. *Monthly Weather Review*, **138**, 1293–1306.

Lorenz, E. N., 1996: Predictability: a problem partly solved. *Proc. Seminar on Predictability*, Shinfield Park, Reading, Berkshire, United Kingdom, European Centre for Medium-Range Weather Forecast, Vol. 1.

Mitchell, H. L. and P. L. Houtekamer, 2009: Ensemble Kalman filter configurations and their performance with the logistic map. *Monthly Weather Review*, **137**, 4325C4343.

Musso, C., N. Oudjane, and F. Le Gland, 2001: Improving regularized particle filters. *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., Springer-Verlag.

Nakano, S., G. Ueno, and T. Higuchi, 2007: Merging particle filter for sequential data assimilation. *Nonlinear Processes in Geophysics*, **14**, 395–408.

Ott, E., et al., 2004: A local ensemble Kalman filter for atmospheric data assimilation. *Tellus*, **56A**, 415–428.

Sacher, W. and P. Bartello, 2008: Sampling errors in ensemble Kalman filtering. part i: theory. *Monthly Weather Review*, **136**, 3035–3049.

Sakov, P. and P. R. Oke, 2008a: A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters. *Tellus*, **60A**, 361–371.

Sakov, P. and P. R. Oke, 2008b: Implications of the form of the ensemble transformation in the ensemble square root filters. *Monthly Weather Review*, **136**, 1042–1053.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson, 2008: Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, **136**, 4629–4640.

van Leeuwen, P. J., 2003: A variance-minimizing filter for large-scale applications. *Monthly Weather Review*, **131**, 2071–2084.

van Leeuwen, P. J., 2009: Particle fltering in geophysical systems. *Monthly Weather Review*, **137**, 4089–4114.

van Leeuwen, P. J., 2010: Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, **to appear**.

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

Xiong, X., I. M. Navon, and B. Uzunoglu, 2006: A note on the particle filter with posterior Gaussian resampling. *Tellus*, **58A**, 456–460.

List of Tables

Table 1. Simulation results for Lorenz 63 system with Gaussian observation error.

	$\Delta = 0.02$			$\Delta = 0.05$		
	$\theta = 1/2$	1	2	1/2	1	2
EnKF	.038 (.047)	.075 (.103)	.179 (.232)	.059 (.074)	.131 (.167)	.330 (.412)
	96.2	94.6	97.5	95.9	94.7	93.9
NLEAF1	.038 (.047)	.074 (.101)	.169 (.227)	.056 (.071)	.122 (.160)	.295 (.364)
	96.8	94.6	98.4	96.1	95.9	94.8
NLEAF2	.037 (.042)	.073 (.085)	.141 (.174)	.049 (.059)	.090 (.121)	.220 (.262)
	95.2	88.3	96.6	96.5	96.6	95.4
PF	.053 (.076)	.090 (.141)	.185 (.272)	.062 (.087)	.109 (.164)	.245 (.323)
	98.8	98.4	99.2	97.9	99.1	96.7
PFGR	.048 (.062)	.080 (.124)	.180 (.261)	.056 (.073)	.117 (.160)	.279 (.341)
	96.5	97.7	99.3	96.4	96.7	94.5
MPF	.045 (.061)	.090 (.127)	.187 (.257)	.056 (.076)	.115 (.160)	.281 (.349)
	97.3	95.4	98.3	96.8	96.9	95.1

Table 2. Simulation results for Lorenz 63 system with double exponential observation error.

	$\Delta = 0.02$			$\Delta = 0.05$		
	$\theta = 1/2$	1	2	1/2	1	2
EnKF	.058 (.069)	.104 (.153)	.278 (.393)	.082 (.110)	.196 (.267)	.499 (.615)
	93.2	99.0	97.8	94.7	96.8	93.5
NLEAF1	.042 (.054)	.086 (.118)	.215 (.275)	.081 (.088)	.162 (.197)	.386 (.465)
	94.7	95.4	96.0	90.2	94.4	95.2
NLEAF2	.035 (.046)	.073 (.097)	.182 (.202)	.055 (.071)	.121 (.149)	.277 (.332)
	97.0	95.5	94.1	94.9	95.4	92.9
PF	.055 (.082)	.095 (.155)	.209 (.294)	.069 (.097)	.133 (.181)	.282 (.373)
	97.6	98.9	98.2	97.4	97.3	96.1
PFGR	.050 (.068)	.084 (.136)	.233 (.279)	.070 (.092)	.155 (.192)	.336 (.413)
	96.0	98.8	94.3	95.8	94.9	94.1
MPF	.047 (.066)	.088 (.134)	.216 (.287)	.066 (.090)	.144 (.185)	.329 (.424)
	96.4	97.7	96.2	96.8	95.7	96.1

Table 3. Simulation results for Lorenz 96 system, hard case

	NLEAF1	NLEAF1q	EnKF	XEnsF
mean	0.66	0.71	0.77	0.92
median	0.61	0.66	0.71	0.85
std	0.25	0.25	0.28	0.31

Table 4. Simulation results for Lorenz 96 system, easy case

n	10	20	50	100	400
LETKF	.30 (.23)	.30 (.23)	.29 (.23)	.29 (.23)	.30 (.33)
	77.1	85.7	86.8	87.6	95.8
NLEAF1	.33 (.33)	.28 (.30)	.24 (.28)	.23 (.24)	.24 (.25)
	85.2	92.7	93.9	94.2	95.0
NLEAF1q	N/A	N/A	N/A	N/A	.39 (.37)
					92.3
EnKF	.45 (.49)	.34(.48)	.30 (.24)	.27 (.22)	.26 (.19)
	92.2	98.3	81.6	85.6	80.5

List of Figures

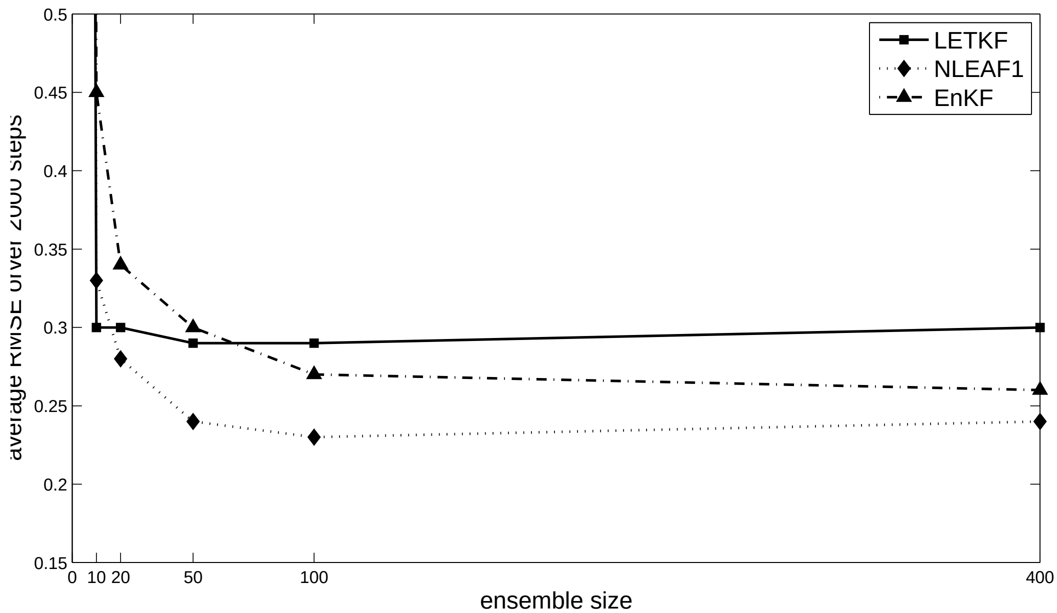

Fig. 1. Ensemble size against RMSE for Lorenz 96 system, easy case