A Local Particle Filter for High Dimensional Geophysical Systems

S. G. Penny[1,2,3] and T. Miyoshi[3,1]

[1]{University of Maryland, College Park, Maryland, United States}

[2]{National Centers for Environmental Prediction, College Park, MD, United States}

[3]{RIKEN Advanced Institute for Computational Science, Kobe, Japan}

Correspondence to: S. G. Penny (Steve.Penny@noaa.gov)

Abstract

A local particle filter (LPF) is introduced that outperforms traditional ensemble Kalman filters in highly nonlinear/non-Gaussian scenarios, both in accuracy and computational cost. The standard Sampling Importance Resampling (SIR) particle filter is augmented with an observation-space localization approach, for which an independent analysis is computed locally at each gridpoint. The deterministic resampling approach of Kitagawa is adapted for application locally and combined with interpolation of the analysis weights to smooth the transition between neighboring points. Gaussian noise is applied with magnitude equal to the local analysis spread to prevent particle degeneracy while maintaining the estimate of the growing dynamical instabilities. The approach is validated against the Local Ensemble Transform Kalman Filter (LETKF) using the 40-variable Lorenz-96 model. The results show that: (1) the accuracy of LPF surpasses LETKF as the forecast length increases (thus increasing the degree of nonlinearity), (2) the cost of LPF is significantly lower than LETKF as the ensemble size increases, and (3) LPF prevents filter divergence experienced by LETKF in cases with non-Gaussian observation error distributions.

1 Introduction

The Particle Filter (PF) has been explored in the data assimilation community since the introduction of its Gaussian linear variant, the Ensemble Kalman Filter (EnKF) in the mid-1990's *(Evensen, 1994)*. While general PFs have been intractable for high dimensional systems, the EnKF has experienced great success in numerical weather prediction (NWP) *(e.g. Kleist 2012; Hamrud et al., 2014)* and ocean data assimilation *(e.g. Penny et al. 2015)*. However, at least two limitations are on the horizon for EnKFs. Perhaps counter-intuitively, these limitations arise due to *increased* computational resources, and have already become challenges at the RIKEN Advanced Institute for Computational Science (AICS, e.g., *Miyamoto et al. 2013; Miyoshi et al. 2014; 2015*). First, global models will be pushed to higher resolutions in which they begin to resolve highly nonlinear processes. To maintain the Gaussian linear assumption required for the EnKF, much smaller timesteps are needed. For example, the standard 6-hour analysis cycles used for the atmosphere may need to be decreased to 5 minutes or even 30 seconds. Second, large ensembles (e.g. with ensemble size $k > 10{,}000$ members) will become feasible for lower-resolution models. While at first this may seem an advantage rather than a limitation, the computational cost of the local ensemble transform Kalman filter (LETKF) *(Hunt et al., 2007)*, for example, increases at a rate $O(k^3)$ with the ensemble size k. Thus as the ensemble size increases, the cost of computing the analysis increases at a much greater rate. Alternative EnKFs feasible for large geophysical systems scale in computational cost with the observation dimension, which is typically multiple orders of magnitude larger than the ensemble dimension.

The PF is generally applicable to nonlinear non-Gaussian systems, including cases with multi-modal distributions or nonlinear observation operators. With little difficulty, PFs can explicitly include representation of model error, nonlinear observation operators *(Nakano 2007; Lei and Bickel 2011)*, non-diagonal observation error covariance matrices, and non-Gaussian likelihood functions. For example, observed variables such as precipitation are inherently non-Gaussian and cannot be effectively assimilated by standard EnKF techniques *(e.g. Lien et al. 2013; 2015)*. In the expansion to sea-ice and land data assimilation applications, the non-Gaussian quantities such as ice concentration, ice thickness, snow cover, and soil moisture outnumber those that can be modeled with Gaussian error. *Bocquet et al. (2010)* further review the difficulties using observations with non-Gaussian error distributions. All of these problem-specific variations can create great difficulties for standard

methods, such as the EnKF or variational approaches (3D-Var/4D-Var), as used in current operational systems.

Sampling Importance Resampling (SIR) (also known as the bootstrap filter, *Gordon et al., 1993*) is a commonly used enhancement to the basic Sequential Importance Sampling (SIS) particle filter. However, even with resampling the number of ensemble members required by the SIR particle filter to capture the high probability region of the posterior in high-dimensional geophysical applications is too large to make SIR usable *(Ades and van Leeuwen 2013)*. *Snyder et al. (2008)* found that the number of required ensemble members scales exponentially with the size of the system, giving the example that a 200 dimensional system would require 10^{11} members. However, *Snyder et al.* note that clever choices of the proposal distribution could overcome the need for these exponentially large ensemble sizes in high-dimensional systems, which has been more recently explored by *Synder et al. (2015)*. Applying such an approach, *van Leeuwen (2003)* considers a model for the Agulhas Current with dimension of roughly $2\text{x}10^5$. Further, *Beskos et al. (2012)* discuss recursive methods for estimating the proposal densities, similar to the Running-in-Place algorithm *(Yang et al. 2012a/b, Penny et al. 2013)* that has been used with LETKF in meteorological and oceanographic data assimilation. *Xiong et al. (2006)* presented techniques related to the ETKF that may provide an alternative approach to the common SIR method for generating particle estimates for the posterior distribution.

Techniques such as localization and inflation are typically applied as modifications to make the EnKF operationally feasible. Inspired by this practice, we introduce a local particle filter (LPF) designed for geophysical systems that is scalable to high dimensions and has computational cost $O(k)$. Spatial localization is typically justified by the fact that long distance correlations are either spurious or weak in comparison to nearby correlations, particularly when the ensemble is under-sampled. We use this same approach to reduce the required ensemble size for the LPF. The findings of Jardak et al. (2000) indicate that the EnKF with localization works well in the case of a linear observation operator but has difficulties with nonlinear observation operators.

2 Methodology

Localization is used in most operational NWP data assimilation systems, either through a direct scaling of the background error covariance matrix *(e.g. Whitaker and Hamill, 2002)* or by a scaling of the observation error covariance matrix *(Hunt et al. 2007)*. Because the

computation of a background error covariance matrix is not needed for the PF, the latter approach is applied here to develop an effective PF for high-dimensional geophysical systems. Localization reduces the dimensionality of the solution space, thus requiring fewer ensemble members to sample the phase space. Gaussian noise is applied as an additive inflation to prevent particle degeneracy.

2.1 The Standard SIR Particle Filter

There are many variations of the PF *(Stewart and McCarty, 1992; Gordon et al., 1993; Kitagawa, 1996; Hurzeler and Kunsch, 1998; Liu and Chen, 1998)*. In essence it is simply a Monte Carlo estimation of Bayes Theorem, reformulated as a recursion *(Doucet et al., 2001)*,

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \quad , \tag{1}$$

where $p(x_t \mid y_{1:t})$ is the probability of the state x at time t, given all observations y up to time t. We consider the model domain to be described by vectors x with dimension m, the observation domain to be descried by vector y with dimension l, and an ensemble of size k. The term in the numerator can be expressed using the Chapman-Kolmogorov equation as,

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})\, dx_{t-1} \quad , \tag{2}$$

and similarly the term in the denominator can be expressed as,

$$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})\, dx_t \quad . \tag{3}$$

The two factors in the numerator of Equation (1) are sampled using a numerical model f,

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) \approx \frac{1}{k} \sum_{i=1}^{k} \delta\left(\mathbf{x}_t - f\left(\mathbf{x}_{t-1}^{i}\right)\right) \quad , \tag{4}$$

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = g(\mathbf{y}_t \mid \mathbf{x}_t) \quad . \tag{5}$$

The term in eqn. (5) is typically called the likelihood, because the probability of y given x is equivalent to the likelihood of x given y, i.e. $p(\mathbf{y} \mid \mathbf{x}) \equiv \ell(\mathbf{x} \mid \mathbf{y})$. The function g is general and can represent any distribution for the observations.

For the experiments here we generate two experiment cases, each with a different likelihood function. First we use a Gaussian likelihood function corresponding to that used for EnKFs,

$$g(\mathbf{y}_t \mid \mathbf{x}_t) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{R}|}} \exp\left[-\frac{1}{2}(\mathbf{y}_t - h(\mathbf{x}_t))^T \mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\right], \quad (6)$$

where the function h is a general, possibly nonlinear, observation operator mapping from the model state space to the observation space. For the LPF, it is straightforward to generalize to arbitrary non-Gaussian likelihood functions. As an example, we also apply a multivariate Gaussian mixture model (GM_2) following *Fowler and van Leeuwen (2013)* with pdf,

$$p(\mathbf{y}|\mathbf{x}) \propto \nu_w \exp\left[(\mathbf{y}+\nu_1\mathbf{1}-h(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y}+\nu_1\mathbf{1}-h(\mathbf{x}))\right] + (1-\nu_w)\exp\left[(\mathbf{y}+\nu_2\mathbf{1}-h(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y}+\nu_2\mathbf{1}-h(\mathbf{x}))\right]. \quad (7)$$

Let each ensemble member be identified with an index, i. Normalized weights are evaluated for each member,

$$w_t^i = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t^i)}{\sum_{j=1}^{k} p(\mathbf{y}_t \mid \mathbf{x}_t^j)}. \quad (8)$$

Then the posterior is,

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{k} w_t^i \delta(\mathbf{x}_t - f(\mathbf{x}_{t-1}^i)). \quad (9)$$

Based on Liouville's theorem, the evolution of a probability measure in a dynamical system satisfies the property that "the probability of finding trajectories inside the time-variant volume *W(t)* is constant during the evolution of the dynamical system." (Property 2, http://www.ulb.ac.be/di/map/gbonte/ftp/bontempi_fpde.pdf). If the solution manifold expands in some directions, so will the pdf represented by the particles. Thus, the fidelity of the distribution will quickly become insufficient to sample a solution manifold around the true trajectory. A resampling procedure is used to refocus the particles on the densest areas of the distribution at each analysis step. For the experiments here, we use a resampling procedure that resembles resampling with replacement. After resampling we have,

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \frac{1}{k}\sum_{i=1}^{k}\delta\left(\mathbf{x}_t - \mathbf{x}_t^i\right) . \tag{10}$$

2.2 The Transform Interpretation

The PF can be interpreted similarly to the Ensemble Transform Kalman Filter (ETKF) of Bishop et al. (2001). The transform interpretation has been explored by *Reich 2013* and *Metref et al. 2014*. Namely we define the PF solution as a transformation of the background ensemble to the analysis ensemble,

$$\mathbf{X}^a = \mathbf{X}^b\mathbf{T} . \tag{11}$$

where each column of $\mathbf{X}^b$ is a background ensemble member, and each column of $\mathbf{X}^a$ is an analysis ensemble member. The transform function $\mathbf{T}$ in (11) is used to generically refer to either E in (16), or $E^{(j)}$ in (19) in the text below.

Let $\mathbf{b}$ be the vector of background particle indices and $\mathbf{a}$ be the vector of analysis particle indices,

$$\mathbf{b} \in \left\{ \mathbf{z} \in \mathbb{Z}^k \middle| \mathbf{z} = (1,2,3,...,k)^T \right\}, \tag{12}$$

$$\mathbf{a} \in \left\{ \mathbf{z} \in \mathbb{Z}^k \middle| \mathbf{z} = (a_1,a_2,a_3,...,a_k)^T, a_i \in [1,k] \right\}. \tag{13}$$

Here, $\mathbf{b}$ is simply vector composed of an ordered set of indices ranging from 1 to k, while $\mathbf{a}$ is a vector composed of a subset of elements of the vector $\mathbf{b}$, with no particular ordering and possible repetitions. We also extend the definition of $\mathbf{a}$ to allow the vector to vary by grid point. For example, for any grid point p, the vector $\mathbf{a}_p$ may represent the resampling index at point p so that,

$$\mathbf{X}^a(p,i) = \mathbf{X}^b\left(p, a_{p,i}\right), \tag{14}$$

where the arguments indicate the row and column of the matrix $\mathbf{X}$.

We further define the vector $\mathbf{e}_i$, for which only the i^{th} element is nonzero,

$$\mathbf{e}_i \in \left\{ \mathbf{z} \in \mathbb{Z}_2^k \middle| \mathbf{z} = (0,...,0,1_i,0,...,0)^T \right\} . \tag{15}$$

If $\mathbf{e}_i$ are the canonical basis vectors then we can define,

$$\mathbf{E}_{k\times k} = \begin{bmatrix} \mathbf{e}_{a_1} & \mathbf{e}_{a_2} & \cdots & \mathbf{e}_{a_k} \end{bmatrix}. \tag{16}$$

For the standard PF, the indicator matrix **E** is made up of k (not necessarily unique) standard basis vectors $\mathbf{e}_i$, with entries 0 and 1 that we will interpret as weights. Thus the analysis ensemble for the PF is defined simply by the transform,

$$\mathbf{X}^a_{m\times k} = \mathbf{X}^b_{m\times k}\mathbf{E}_{k\times k}. \tag{17}$$

We note that by using this approach, each new analysis member, with index i, maintains the continuity properties of its associated background member, a_i.

For reference in the next section, the components of the analysis matrix $\mathbf{X}^a$ will have the form,

$$\mathbf{X}^a = \begin{bmatrix} x_{1,i}e_{i,1} & x_{1,i}e_{i,2} & \cdots & x_{1,i}e_{i,k} \\ x_{2,i}e_{i,1} & x_{2,i}e_{i,2} & \cdots & x_{2,i}e_{i,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,i}e_{i,1} & x_{m,i}e_{i,2} & \cdots & x_{m,i}e_{i,k} \end{bmatrix}. \tag{18}$$

Here we have used the Einstein tensor notation for the elements, in which $x_{1,i}e_{i,1}$ represents a summation over the index i (i.e. the inner product of row 1 of $\mathbf{X}^b$ and column 1 of **E**). While the summation index could be represented generically by any symbol, we reuse the symbol '*i*' due to its correspondence with the background particle indices as defined above.

2.3 The Localization Approach

Snyder et al. (2008) note that when either the model dimension or observation count is large, the PF requires significantly more particles to give an adequate representation of the system. Localization, as introduced by Houtekamer and Mitchell (1998), reduces both the model and observation dimensions by dividing the problem into a series of sub-domains, thus reducing the required number of particles for accurate filtering. Bengtsson et al. (2003) were among the first to point to spatially local updating, using a local subset of observations, as a solution to difficulties of high-dimensional non-Gaussian filtering. *Lei and Bickel (2011)* introduced the notion of computing local weights in a non-Gaussian filter. The LPF follows *Hunt et al. (2007)* to select nearby observations for independent analysis at each grid point. Nearby grid points thus assimilate nearly identical sets of observations to derive their analyses.

We use the deterministic resampling of *Kitagawa (1996)*, with complexity $O(k)$, adapted for local use as described next. A uniform partition of [0,1] with width $1/k$ is first generated globally, with an offset applied from a uniform distribution over $[0,1/k]$. The same partition is used locally for resampling at each grid point. Cumulative sums of the normalized weights (eqn. 8),

$$\breve{w}_t^j = \sum_{i=1}^{j} w_t^i \,, \tag{19}$$

are compared with the elements of the partition. Traversing $j=1,\ldots,k$ all unassigned particles with index j having a corresponding cumulative sum with index j that surpasses the next element of the partition (ordered monotonically increasing) are assigned as particles of the resampled (analysis) ensemble.

For a given grid point, when the cumulative sums of the particle weights are near one of the partition values, there may be sensitivity in neighboring grid points that lead to discontinuities between local analyses. The analysis ensemble at this grid point consists of a subset of background particle indices (1 through k) with repetitions. To eliminate the discontinuities with neighboring grid points we associate weights of a local transform function T with the particle indices, nominally either 1.0 (full weight) or 0.0 (no weight). This is partially inspired by the "weight interpolation" of *Bowler (2006)*, applied to LETKF by *Yang et al. (2009)*, who found that interpolation of weights was more robust than interpolation of state values. At a single grid point, there are k pieces of background information about the possible system state at that point. In the standard PF, only 1 out of these k pieces of information is retained for each analysis ensemble member, based on the overall agreement with observations. In the LPF we use anywhere from 1 to k members to construct each analysis member based on the agreement with observations within a local radius.

Thus the modified local analysis at a given model grid point p is given as a linear combination of ensemble members at that point. For point p, ensemble member i, a set of N neighbor points N_p, and vector $\mathbf{a}_p$ acting as a resampling index as in (13) but given as a function of the model grid point, we have,

$$\mathbf{X}^{LPF}_{\{p,i\}} = \frac{1}{2}\mathbf{X}^{a}_{\{p,i\}} + \frac{1}{2N}\sum_{n\in N_p} \mathbf{X}^{b}_{\{p,\mathbf{a}(n,i)\}} \,. \tag{20}$$

The subscript indices indicate the row and column of the matrix.

A new transform can then be defined for the LPF at each point in the model domain to generate a set of m indicator matrices, $\mathbf{E}^{(j)}_{k\times k}$, so that for each point (x_p, for p=1,…,m),

$$\left(\mathbf{X}^{a}_{m\times k}\right)^{p} = \mathbf{X}^{b}_{m\times k}\mathbf{E}^{(p)}_{k\times k}. \tag{21}$$

Using the summation tensor notation described in the previous section, the analysis ensemble can be written,

$$\mathbf{X}^{a} = \begin{bmatrix} x_{1,i}e^{(1)}_{i,1} & x_{1,i}e^{(1)}_{i,2} & \cdots & x_{1,i}e^{(1)}_{i,k} \\ x_{2,i}e^{(2)}_{i,1} & x_{2,i}e^{(2)}_{i,2} & \cdots & x_{2,i}e^{(2)}_{i,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,i}e^{(m)}_{i,1} & x_{m,i}e^{(m)}_{i,2} & \cdots & x_{m,i}e^{(m)}_{i,k} \end{bmatrix}. \tag{22}$$

The transform matrix may have any degree of sophistication. We apply a smoothing operator to the ensemble subspace by modifying the weights $\mathbf{e}_{a_i}$ associated with each analysis particle index $\mathbf{a}_{(i)}$ from a binary value to a continuous value between 0 and 1, while maintaining all column sums of equal to one, and call this new transform matrix **W**.

We define the concept of a 'neighbor point' abstractly as a point near the analyzed grid point based on a specified distance metric. Through examination of (20), it is clear that the choice of neighbor points simply informs the weighting of indices, and that the values at these neighbor points otherwise have no impact on the analysis. If there are N neighbor points, then there will be at most min(N+1,k) collocated pieces of background information that can be utilized to construct each analysis ensemble member at this point. An example is given in **Figure 1**. With a sufficiently large set of observations the indices for these neighbor points are calculated from nearly identical observational innovations. Therefore, when there is a sufficiently large ensemble size (k) the difference between the states associated with different particle indices will be small. The transform function T is applied across all background indices (i.e. for particles 1,…,k) at this grid point to compute the analysis.

We summarize that the total smoothing in the resulting analysis is achieved independently at each grid point due to a combination of localization in observation space and the formation of a convex combination of analysis weights in the ensemble space. There is no explicit smoothing in the physical model space such as might occur using a Gaussian smoother applied via a stencil of function values from neighboring grid points. Instead, the

neighbor points simply inform the choice of weights to apply to particle indices at a single grid point. There is an implicit smoothing achieved by applying the same procedure to many contiguous model grid points, each generating a different set of weights that vary only slightly. This is similar to the effect of observation-space localization. However, as with most localization techniques, the more distant information suffers from the poor sampling size of the ensemble. Thus if holding the ensemble size k fixed, a larger radius of neighbor points may lead to noisier results rather than increased smoothing.

2.4 Particle Degeneracy

The particle selection process of the PF reduces the rank of the ensemble. For a linear deterministic system this leads to a rapid collapse of the ensemble and divergence of the filter. For a sufficiently stochastic nonlinear system the members are made distinct after a single forecast step. If the nonlinear system is not sufficiently stochastic, then we must address the ensemble initialization problem at every analysis cycle. *Pazo et al. (2010)* discuss the desirable properties in an initial ensemble, namely the members: (1) should be well-embedded in the attractor, (2) should be statistically equivalent but have enough diversity to represent a significant portion of the phase space, (3) should adequately represent the error between the analysis and true state, and (4) should sample the fastest growing directions in phase space. We wish to avoid particle degeneracy while also engendering some of these qualities. Applying noise to the PF at the sampling step is a common empirical technique. Therefore we employ a simple approach: at each cycle we add Gaussian noise with variance scaled locally to a magnitude matching the analysis error variance and apply this to each analysis member prior to the subsequent ensemble forecast. The amplitude of the additive noise was chosen to conform to the dynamics of the growing error subspace, as estimated by the analysis ensemble spread. We note that this amplitude varies spatially and temporally. The results degraded when departing from this approach. The practice of applying noise to the PF sampling step is a standard technique.

We caution that EnKFs have fundamentally different behavior than the general PF, in that the former maintain a forcing term that drives the DA system toward the observations even when the forecast may start far from the true state. The general PF with resampling essentially requires random chance to generate a state with high enough probability to be propagated through the resampling step. As the system size increases the probability of such an event declines.

2.5 Computational Complexity

A data assimilation system is comprised of many components. We simplify the cost analysis in order to gain an approximate relative measure of the algorithms presented here. Let m be the model dimension, l be the observation dimension, and $\overline{l_i}$ be the average local observation dimension. The total cost (C_T) of an analysis cycle is equal to the overhead (C_H) of the assimilation system plus m times the average local cost (C_L) of the assimilation method plus k times the cost of one model forecast (C_M) of duration t,

$$C_T(k,l,m) = C_H(k,l,m) + m \cdot C_L\left(k,\overline{l_i}\right) + k \cdot C_M(\tau,m) \tag{23}$$

We will assume that between the two methods the overhead and model costs are approximately equal. The primary difference in cost between the two systems is then the average local cost,

$$C_L^{LPF} = O\left(k\overline{l_i}\right), \tag{24}$$

$$C_L^{LETKF} = O\left(k^2\overline{l_i} + k^3\right). \tag{25}$$

If as is typically the case, the system size m is large and the ensemble size k is small, then

$$C_T(k,l,m) \approx C_H(k,l,m) + O(m), \tag{26}$$

and the difference in cost between LETKF and LPF is small. However for large k, we see that the average local cost of LETKF,

$$C_T^{LETKF}(k,l,m) = C_H(k,l,m) + O\left(mk^3\right), \tag{27}$$

exceeds that of the LPF,

$$C_T^{LPF}(k,l,m) = C_H(k,l,m) + O(mk). \tag{28}$$

Subtracting the overhead costs, in this case the LPF is a factor of k^2 cheaper than LETKF.

2.6 Data Assimilation Methods

We enumerate the benefits of the LPF versus the benchmark LETKF, an ensemble square root filter that performs its analysis in the ensemble space at each grid point using a geospatially

local selection of observations. The LETKF approach is very efficient as long as the ensemble size is small relative to the number of observations and the model dimension.

We use LETKF as a proxy for a general EnKF. *Nerger (2015)* gives a comparison between LETKF and the Ensemble Square Root Filter (ESRF) of *Whitaker and Hamill (2002)*, while *Tippett et al. (2003)* indicate that the ESRF is identical to the Ensemble Adjustment Kalman Filter (EAKF) of *Anderson (2001)* when using serial single observation processing.

2.7 Experiment Design

We demonstrate the algorithms on the Lorenz-96 system *(Lorenz 1996)*, composed of m=40 grid points, using Lorenz's original forcing, F=8.0. The Lorenz-96 system has frequently been used to demonstrate PF and other data assimilation algorithms for the geosciences *(Nakano et al, 2007; van Leeuwen, 2010; Lei and Bickel, 2011; Penny 2014)*. Observations are sampled from a nature run of Lorenz-96 after running the model for 14,400 timesteps to allow the model to settle on the attractor. Gaussian noise is added to each observation with a standard deviation of 0.5. For various experiments, the Lorenz-96 system is observed either at every 0.05 or 0.5 timesteps, reflective of a 6-hour and 60-hour forecast, respectively, based on Lorenz's original description of the system. Observations are sampled randomly on the interval [0,m], and a linear interpolation is used for the observation operator. The last experiment case uses a bimodal Gaussian mixture distribution to represent observational error.

3 Results

The standard SIR PF performs poorly with any ensemble size *O(m)*. For example, using 1500 particles and 20 randomly chosen observations per analysis cycle leads to rapid filter divergence for the L96 system, even in a relatively linear regime (dt=0.05) of the system (**Figure 2**). On the contrary, LETKF performs well even with few ensemble members and few observations per cycle (k=20, l=10). Localization is consistent between each method, using r=2 gridpoints. For a given ensemble size, increasing the localization radius degraded the accuracy of both methods. Penny (2014) addressed the impact of varying localization radius on LETKF for this L96 system. It was shown that for a fixed ensemble size, the analysis errors reduce when the localization radius decreases, and for a fixed localization radius, the analysis errors reduce when the ensemble size increases. The LPF showed similar behavior to LETKF in this regard. To explore the relative advantages of each approach, we will describe a

series of cases in which the LETKF outperforms the LPF, and in which the LPF outperforms LETKF.

3.1 Case 1: typical forecast lengths (dt=0.05, or 6-hr)

Lorenz (1996) introduced the dt=0.05 timescale as being comparable to the error doubling taking place over 6 hours in the operational forecasting systems of the early 1990's. In this relatively linear timescale of the L96 system, LETKF clearly outperforms the LPF at a given ensemble size. This is expected as EnKFs take advantage of the Gaussian/linear assumption. When the experiment parameters match such assumptions (even loosely), LETKF performs quite well. However, using localization, the LPF can perform adequately (i.e. avoid filter divergence) in a similar parameter regime (**Figure 3**). Thus for this case, we find that LETKF attains higher accuracy than the LPF, but the LPF improves upon the accuracy and stability of the standard SIR PF for a given ensemble size.

3.2 Case 2: long forecast lengths (dt=0.50, or 60-hr)

To increase the degree of nonlinearity in a data assimilation system using L96, it is typical to increase the analysis cycle length (e.g. *Lei and Bickel, 2011*). The LPF has superior performance for more nonlinear regimes of the L96 system (e.g. dt=0.5) provided there are many ensemble members, e.g. $O(100)$. Using 80 observations per cycle and 100 ensemble members, for example, LETKF produces occasional errors that propagate eastward (along the positive x-direction). The LPF does not produce such effects, and the errors are generally lower than with LETKF (**Figure 4**). We consider this a relevant scenario because the majority of observations in operational weather forecasting are discarded *(Ochatta et al., 2005)*.

Exploring a more complete parameter space, we examine the forecast error for LETKF over a range of observation coverage (l=2,…,80 per analysis cycle) and ensemble sizes (k=10,…,400), and compare the relative difference to LPF. **Figure 5** shows the average absolute error over 600 analysis cycles of length dt=0,5 for 1600 different parameter combinations of observation coverage (l) and ensemble size (k). The LPF is more accurate than LETKF when using many observations (e.g. $l > 20$) and large ensemble sizes (e.g. $k >$ 100-200). When using fewer than 20 observations per cycle in this case, both LETKF and LPF experience filter divergence. Due to the unconstrained nature of the LPF as described in section 2.4, large errors occur more frequently than for LETKF in this parameter regime.

Further, when examining the computational cost of the LPF versus LETKF, the relative costs reflect the analytical assessment given above in section 2.5. Namely, the elapsed time of the LETKF experiments grows with the cube of the ensemble size, while the elapsed time of the LPF is significantly lower at large ensemble sizes (**Figure 6**).

3.3 Case 3: Non-Gaussian observation error.

The previous section examined the impacts of nonlinearity and non-Gaussianity on the forecast. We now examine the impacts of non-Gaussian observation error. Using a multivariate Gaussian mixture model (GM_2) following *Fowler and van Leeuwen (2013)*, we apply a corresponding random error to the observation and compare the impacts on LETKF and LPF. We use the LETKF without modification, but modify the likelihood function of LPF to the definition of GM_2 as in section 2.1, Eqn. (7). Using n_w=0.1, n_1=-1, n_2=1, we create a bimodal distribution biased toward the second Gaussian mode. The analysis cycle is dt=0.05 (6 hr) as in experiment case 1, section 3.1. **Figure 7**, compares LETKF and LPF using k=100 ensemble members and l=80 observations. An additional results is given for LPF with l=20 observations. The introduction of a strong non-Gaussianity in the observation error distribution disrupts LETKF and eventually creates errors that propagate throughout the entire domain. Using the same ensemble size and observation count, the LPF gains significant advantage in its ability to explicitly account for the non-Gaussian error structure of the observations. Even reducing the observation count by 75%, the LPF maintains its advantage.

3.4 Case 4: Examining the impact of spatial smoothing.

We consider the example of LPF applied to L96 with the analysis cycle dt=0.5 (producing a relatively non-linear error growth), a localization radius of 2 grid points, k=100 ensemble members, l=40 observations per cycle, and observation error of 0.5. By applying ensemble smoothing of the weights, we find that the mean absolute forecast error (averaged over the model domain) is reduced versus a non-smoothed analysis (**Figure 8**). Increasing the search radius to 2 gridpoints for the same example case does not produce any clear benefits.

Considering the impact on the ensemble, we use the smoothed analysis the mean effective ensemble size N_{eff},

$$N_{eff} = \left[\sum_{i=1}^{j} \left(w_t^i \right)^2 \right]^{-1}, \tag{29}$$

with w_t^i defined as in (8), calculated at each gridpoint and averaged. We find that the effective ensemble size is increased when using the smoothed versus the non-smoothed approach (**Figure 9**).

4 Conclusions

The Local Particle Filter (LPF) has been shown to outperform a state of the art ensemble Kalman filter (i.e. LETKF) in scenarios that violate the Gaussian/linear assumptions of the Kalman filter. We showed the advantage of the LPF when forecast is more non-linear (via longer analysis cycles, or less frequent observations), and when observation error is non-Gaussian (using a bimodal error distribution). Further, upon transitioning to large ensembles the LPF has a significant cost savings relative to LETKF.

The LPF maintains many of the attractive qualities that give Particle Filters (PFs) advantages over standard EnKFs. While the PF provides a means of assimilating observations with non-Gaussian errors (e.g. precipitation, sea-ice concentration), we caution that the covariances utilized by the EnKF play a critical role in constraining the unobserved variables. Thus while the LPF is not optimal for all possible data assimilation scenarios, there is great potential for the LPF to be combined with more traditional approaches to create adaptive hybrid systems that can avoid catastrophic filter divergence and manage multi-modal forecast distributions, nonlinear observation operators, non-Gaussian observations.

We found that a large number of ensemble members (or particles) and observations are sufficient for the LPF to match or surpass the accuracy of LETKF. We found large ensemble sizes a relevant scenario for realistic systems running on large supercomputers such as the K computer at RIKEN. The use of large observation sets is relevant in operational weather forecasting as much of the dense satellite data is currently discarding in a thinning process. Further, in this parameter regime the LPF has significantly lower computational cost than LETKF.

In a realistic system, some mechanism is needed to drive the ensemble toward the observations in the event of the ensemble drifting away from the true state. The PF itself has no inherent mechanism to do this other than the brute force generation of more particles. There are many techniques in the PF literature for managing filter divergence, but none of them are foolproof. *Atkins et al. (2013)* presented a promising extension of the use of an importance density that may connects effectively with the existing infrastructure of

variational solvers used by most operational centers. Another popular mechanism to achieve this is regularization, which uses a kernel to sample from a continuous distribution at the resampling stage.

Finally, while the inflation mechanism used here was effective for the Lorenz-96 system, it is not adequate for more realistic atmospheric or oceanic models. For such systems, either geospatially correlated noise or stochastic physics parameterizations may be capable of performing the same function. Stochastic physics parameterizations are an active area of research, and are under development for a number of operational center models, including NCEP *(Hou et al., 2006; Hou et al, 2010; Kolczynski et al., 2015),* ECMWF *(Berner et al., 2009; Weisheimer et al., 2014; Watson et al., 2015)*, and the Met Office *(Tennant et al., 2011; Sanchez et al., 2014; Shutts and Paleres, 2014; Shutts 2015)*.

Acknowledgements

We gratefully acknowledge the Japan Society for the Promotion of Science (JSPS) whose FY2013 fellowship supported this work. We would also like to thank the RIKEN Advanced Institute for Computational Science (AICS) for hosting Penny.

Appendix

We provide pseudo-code for the LPF:

```
% INPUTS:
% k   :: ensemble size
% m   :: model dimension
% Yo  :: vector of global observations
% yo  :: vector of local observations
% Xb  :: background ensemble arranged as a matrix
% R   :: observation error covariance matrix
% HXb :: Xb mapped from model space to observation space
%
% FUNCTIONS:
% obsloc :: finds observations local to a grid point
% cumsum :: cumulative sum of array
% sort   :: sorting of array in descending order
% repmat :: repeat array to form a matrix
%
% OUTPUTS:
% Neff :: effective ensemble sample size
```

```
% Xa   :: analysis ensemble arranged as a matrix

%
% Specify minimum of distribution tails
%
mintail = epsilon

%
% Setup global comb for resampling with replacement
%
interval = 1/k
start = interval * random_number
selection_points = [start : interval : start+(k-1)*interval]

%
% Loop over each grid point
%
for mi=1:m
    % Find the observation points within range of focal grid point:
    yo = obsloc(mi,Yo,local_radius)

    % Update (calculate particle weights) using desired distribution
    % (e.g. Gaussian shown here)
    for ki=1:k
       likelihood(ki) =
               exp( -0.5* (yo-HXb(:,ki))' * R^{-1} * (yo-HXb(:,ki)) )
    end

    % Protect against numerical representation problems in the tails
    likelihood = likelihood + mintail

    %
    % Normalize the weights
    %

    % Compute the weights
    weights = likelihood/sum(likelihood)

    % Calculate effective ensemble sample size
    Neff(mi) = 1/sum(weights.^2)
```

```
1
2       % Form cumulative distribution
3       [wts, wi] = sort(weights)
4       weight  = cumsum(wts)
5
6       %
7       % Apply the comb to resample analysis members
8       %
9       j=1
10      for i=1:k
11          while selection_points(i) >= weight(j)
12              j=j+1
13          end
14
15          % Specify the resampling index rs
16          rs(mi,i)=wi(j)
17
18          % Assign background value as global analysis
19          Xa(mi,i) = Xb(mi,rs(mi,i))
20      end
21
22  end % loop over model grid points
23
24  %
25  % Apply smoothing by weights in ensemble space.
26  %
27  X = 0
28  for ki=1:k
29      for mi=1:m
30          for ni={“set of all Np neighbor points”}
31              X(mi,ki) = X(mi,ki) + Xa(mi,ki) + Xb(mi,rs(ni,ki))
32          end
33          Xa(mi,ki) = X(mi,ki)/(2*Np);
34      end
35  end
36
37  %
38  % Apply additive inflation
39  %
40
41  % Compute the analysis ensemble spread to
```

```
% determine amplitude of inflation needed
if (mean(Neff) > k/2)
    Xstd=std(Xa,2)
else
    % add a little more noise to protect
    % against ensemble collapse
    Xstd=max(std(Xa,2),maximum_obs_error)
end

% Compute Gaussian random values with standard
% deviation equal to analysis ensemble spread
rmat=randn(m,k).*repmat(Xstd,1,k)

% Apply additive inflation (and remove sample mean)
Xa = Xa + rmat-repmat(mean(rmat,2),1,k)
```

References

Ades, M., P.J. van Leeuwen, 2013: An exploration of the equivalent weights particle filter. *Q. J. R. Met. Soc.*, **139** (672), 820–840.

Atkins, E., M. Morzfeld, A.J. Chorin, 2013: Implicit Particle Methods and their Connection with Variational Data Assimilation. *Mon. Wea. Rev.*, **141**(6), 1786-1803.

Berner, J., G. J. Shutts, M. Leutbecher, and T. N. Palmer, 2009: A spectral stochastic kinetic energy backscatter scheme and its impact on flow-dependent predictability in the ECMWF ensemble prediction system. *J. Atmos. Sci.*, **66**, 603–626. doi: http://dx.doi.org/10.1175/2008JAS2677.1

Beskos, A., D. Crisan, A. Jasra, 2012: On the Stability of Sequential Monte Carlo Methods in High Dimensions. arXiv:1103.3965v2 [stat.CO].

Bocquet, M., C.A. Pires, L. Wu, 2010: Beyond Gaussian Statistical Modeling in Geophysical Data Assimilation. *Mon. Wea. Rev.*, **138**, 2997-3023. doi: 10.1175/2010MWR3164.1

Bowler, N., 2006: Comparison of error breeding, singular vectors, random perturbations and ensemble Kalman filter perturbation strategies on a simple model. *Tellus A* **58**: 538–548.

Doucet, A., N. De Freitas and N.J. Gordon, 2001: An introduction to Sequential Monte Carlo Methods, in *SMC in Practice*. http://www.stats.ox.ac.uk/~doucet/smc_resources.html

Evenson, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res*., **99**, 10,143–10,162.

Fowler, A., P.J. van Leeuwen, 2013: Observation impact in data assimilation: the effect of non-Gaussian observation error. *Tellus A*, **65**, 20035, http://dx.doi.org/10.3402/tellusa.v65i0.20035

Gordon, N.J., D. Salmond and A.F.M. Smith, 1993: Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proc. F*., vol. 140, p. 107, 1993.

Hoffman, R.N. and E. Kalnay, 1983: Lagged Average Forecasting: an Alternative to Monte Carlo Forecasting. *Tellus*, **35**, 100-118.

Hou, D., Z. Toth, Y., Zhu, 2006: A Stochastic Parameterization Scheme Within NCEP Global Ensemble Forecast System. *Am. Met. Soc.*, 18th Conference on Probability and Statistics in the Atmospheric Sciences. http://ams.confex.com/ams/pdfpapers/101401.pdf

Hou, D., Z. Toth, Y. Zhu, W. Yang, R. Wobus, 2010: A Stochastic Total Tendency Perturbation Scheme Representing Model- Related Uncertainties in the NCEP Global Ensemble Forecast System. *NOAA/NCEP/EMC*, http://www.emc.ncep.noaa.gov/gmb/yzhu/gif/pub/Manuscript_STTP_Tellus_A_HOU-1.pdf

Hurzeler, M. and H. Kunsch, 1998: Monte Carlo approximations for general state-space models. *JCGS*.

Jardak, M., I.M. Navon, M. Zupanski, 2010: Comparison of sequential data assimilation methods for the Kuramoto-Sivashinsky equation. *Int. J. Num. Meth. in Fluids*, **62** (4), 374-402.

Kitagawa, G., 1996: Monte Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, vol. **5**, no. 1, pp. 1–25.

Kolczynski, W., P. Pegion, T. Hamill, J. S. Whitaker, D. Hou, Y. Zhu, and X. Zhou. 2015: Investigating a New Stochastic Physics Suite for Use in the NCEP Global Ensemble. *Am. Met. Soc.*, 27th Conference On Weather Analysis And Forecasting/23rd Conference On Numerical Weather Prediction.

https://ams.confex.com/ams/27WAF23NWP/webprogram/Paper273838.html

Lei J., and P. Bickel, 2011: A Moment Matching Ensemble Filter for Nonlinear Non-Gaussian Data Assimilation. *Mon. Wea. Rev*., **139**, 3964-3973.

Lien, G.-Y., E. Kalnay, and T. Miyoshi, 2013: Effective assimilation of global precipitation: Simulation experiments. *Tellus A*, **65**, 19915. doi:10.3402/tellusa.v65i0.19915

Lien, G.-Y., T. Miyoshi, and E. Kalnay, 2015: Assimilation of TRMM Multisatellite Precipitation Analysis with a low-resolution NCEP Global Forecasting System. *Mon. Wea. Rev.*, doi:10.1175/MWR-D-15-0149.1 (accepted and available at early online releases)

Liu, J.S. and R. Chen, 1998: Sequential Monte Carlo methods for dynamic systems. *JASA*.

Miyamoto, Y., Y. Kajikawa, R. Yoshida, T. Yamaura, H. Yashiro, and H. Tomita, 2013: Deep moist atmospheric convection in a subkilometer global simulation, Geophys. Res. Lett., 40, doi:10.1002/grl.50944.

Miyoshi, T., K. Kondo, and T. Imamura, 2014: The 10240-member ensemble Kalman filtering with an intermediate AGCM. Geophys. Res. Lett., 41, 5264-5271, doi:10.1002/2014GL060863.

Miyoshi, T., K. Kondo, and K. Terasaki, 2015: Big Ensemble Data Assimilation in Numerical Weather Prediction. Computer, in press.

Nakano, S., G. Ueno, T. Higuchi, 2007: Merging Particle Filter for Sequential Data Assimilation. *Nonlin. Pr. Geophys*., **14**, 395-408.

Nerger, L., 2015: On Serial Observation Processing in Localized Ensemble Kalman Filters. *Mon. Wea. Rev.*, **143**, 1554-1567. doi: 10.1175/MWR-D-14-00182.1

Ochatta, T., C. Gebhardt, D. Saupe, W. Wergen, 2005: Adaptive thinning of atmospheric observations in data assimilation with vector quantization and filtering methods. *QJRMS*, **131**, 3427-3437.

Ott, E., B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D.J. Patil, J. A. Yorke, 2004: A Local Ensemble Kalman Filter for Atmospheric Data Assimilation. *Tellus A*.

Pazo, D., M.A. Rodriguez, J.M. Lopez, 2010: Spatio-temporal evolution of perturbations in ensembles initialized by bred, Lyapunov, and singular vectors. Tellus 62A, 10–23.

Metref, S., E. Cosme, C. Snyder, and P. Brasseur, 2014: A non-Gaussian analysis scheme using rank histograms for ensemble data assimilation. *Nonlin. Processes Geophys.*, **21**, 869–885.

Penny, S. G., E. Kalnay, J.A. Carton, B.R. Hunt, K. Ide, T. Miyoshi, and G.A. Chepurin, 2013: The local ensemble transform Kalman filter and the running-in-place algorithm applied to a global ocean general circulation model, *Nonlin. Processes Geophys.*, **20**, 1031-1046, doi:10.5194/npg-20-1031-2013.

Penny, S.G., 2014: The Hybrid Local Ensemble Transform Kalman Filter. *Mon. Wea. Rev.*, **142**, 2139–2149. doi: http://dx.doi.org/10.1175/MWR-D-13-00131.1

Reich, S., 2013: A nonparametric ensemble transform method for Bayesian inference. SIAM *J. Sci. Comput.*, **35**, A2013–A2024, doi:10.1137/130907367.

Sanchez, C., K.D. Williams, G. Shutts, M. Collins, 2014: Impact of a Stochastic Kinetic Energy Backscatter scheme across time-scales and resolutions. *Q. J. R. Meteorol. Soc.,* **140**, 2625 – 2637.

Shutts, G., Pallarès, A.C., 2014: Assessing parametrization uncertainty associated with horizontal resolution in numerical weather prediction models. *Phil. Trans. R. Soc. A,* **372**: 20130284. http://dx.doi.org/10.1098/rsta.2013.0284

Shutts, G., 2015: A stochastic convective backscatter scheme for use in ensemble prediction systems. *Q. J. R. Meteorol. Soc.*, in press, doi:10.1002/qj.2547

Snyder, C., T. Bengtsson, P. Bickel, J. Anderson, 2008: Obstacles to High-Dimensional Particle Filtering. *Mon. Wea. Rev.*, **136**, 4629-4640.

Snyder., C., T. Bengtsson, M.. Morzfeld, 2015: Performance bounds for particle filters using the optimal proposal. *Mon. Wea. Rev.*, in press, doi: http://dx.doi.org/10.1175/MWR-D-15-0144.1

Stewart, L., P. McCarty, 1992: The use of Bayesian Belief Networks to fuse continuous and discrete information for target recognition and discrete information for target recognition, tracking, and situation assessment, in *Proc. SPIE* Signal Processing, Sensor Fusion and Target Recognition,, vol. 1699, pp. 177-185.

Tennant, W.J., G. J. Shutts, A. Arribas, and S. A. Thompson, 2011: Using a Stochastic Kinetic Energy Backscatter Scheme to Improve MOGREPS Probabilistic Forecast Skill. *Mon. Wea. Rev.*, **139**, 1190–1206. doi: http://dx.doi.org/10.1175/2010MWR3430.1

van Leeuwen, P. J., 2003: A variance-minimizing filter for large-scale applications. *Mon. Wea. Rev*., **131**, 2071–2084.

van Leeuwen, P. J., 2010: Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Q.J.R.Meteorol.Soc.*, **136**, 1991-1999.

Watson, P.A.G., H. M. Christensen, T. N. Palmer, 2015: Does the ECMWF IFS Convection Parameterization with Stochastic Physics Correctly Reproduce Relationships between Convection and the Large-Scale State? *J. Atmos. Sci.*, **72**, 236-242. doi: 10.1175/JAS-D-14-0252.1

Weisheimer, A., S. Corti, T. Palmer, F. Vitart, 2014: Addressing model error through atmospheric stochastic physical parametrizations: impact on the coupled ECMWF seasonal forecasting system. *Phil. Trans. R. Soc. A,* **372**: 20130290. http://dx.doi.org/10.1098/rsta.2013.0290

Xiong, X., I.M. Navon, B. Uzunoglu, 2006: A note on the particle filter with posterior Gaussian resampling. *Tellus A*, **58** (4), 456-460.

Yang, S.-C., E. Kalnay, B.R. Hunt, 2009: Weight interpolation for efficient data assimilation with the Local Ensemble Transform Kalman Filter. *Q. J. R. Meteorol. Soc.* **135**: 251–262.

Yang, S.-C., E. Kalnay, and B.R. Hunt, 2012a: Handling nonlinearity in Ensemble Kalman Filter: Experiments with the three-variable Lorenz model. *Mon. Wea. Rev.*, **in press**. Available online (dx.doi.org/10.1175/MWR-D-11-00313.1).

Yang, S.-C., E. Kalnay, and T. Miyoshi, 2012b: Improving EnKF spin-up for typhoon assimilation and prediction, *Wea. Forecasting*, **27**, 878-897.

Figures

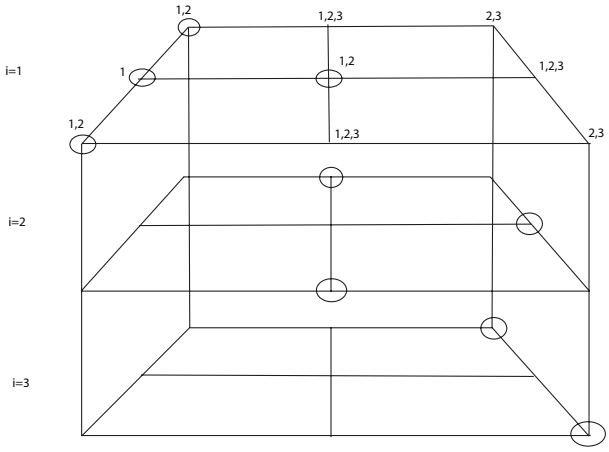

Figure 1. A hypothetical example depicting the construction of a single analysis member. Each level represents a different background ensemble member (particle), with a model space composed of a 3x3 grid. The nodes of the grid are circled if the member is chosen for the construction of analysis member 1 by the LPF. The numerals indicate the ids for the background members that will be averaged at the corresponding node, in this case based on the immediately adjacent neighbor points of that node.

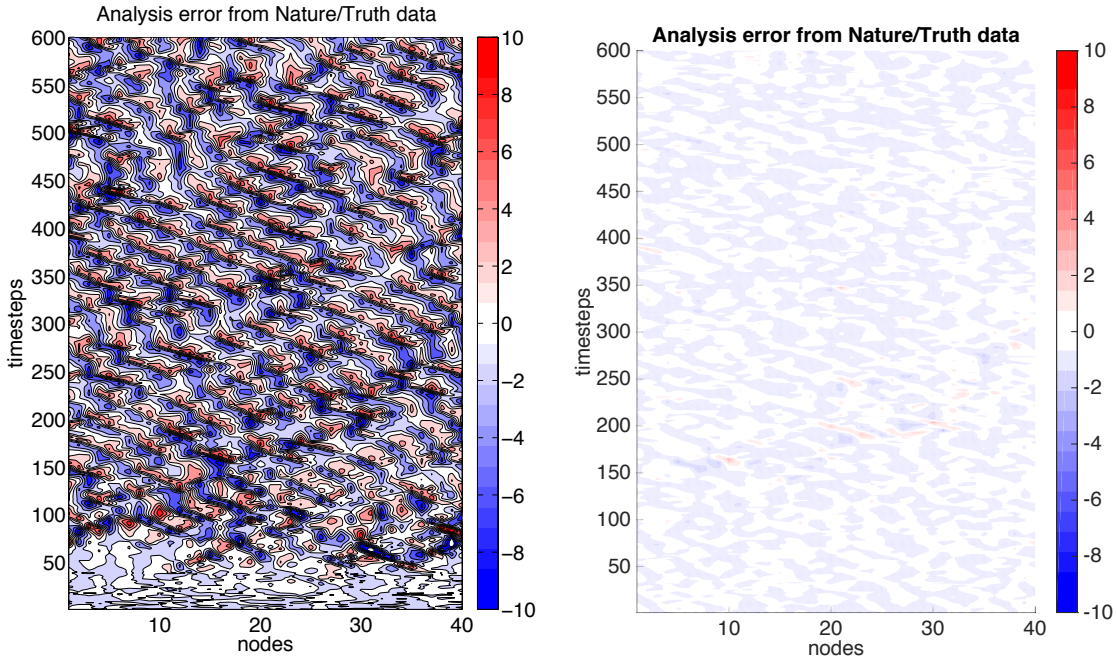

Figure 2. Analysis error using an analysis cycle window length dt=0.05 (6-hr) for (a) the standard SIR PF using k=1500 particles with l=20 observations per analysis cycle, and (b) LETKF with localization radius r=2, k=20 ensemble members, and l=10 observations per analysis cycle, sampled randomly on the domain.

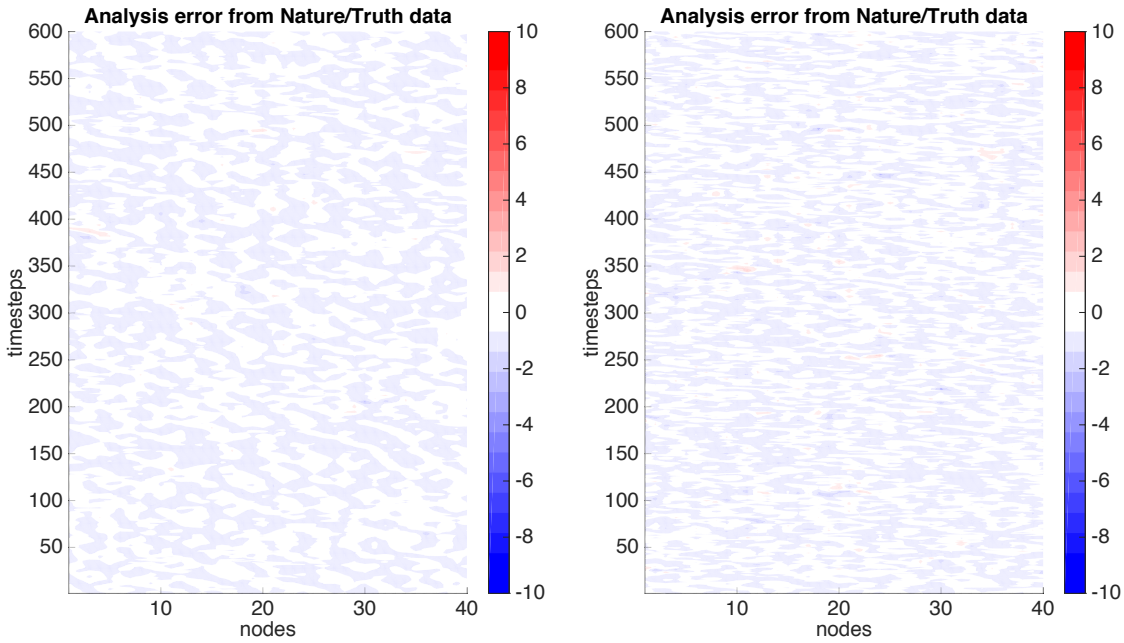

Figure 3. Analysis error for (a) LETKF and (b) LPF, using an analysis cycle window length dt=0.05 (6-hr), localization radius r=2 grid points, k=40 ensemble members, and l=20 observations sampled randomly on the domain (prescribed observation locations and errors are identical for both methods).

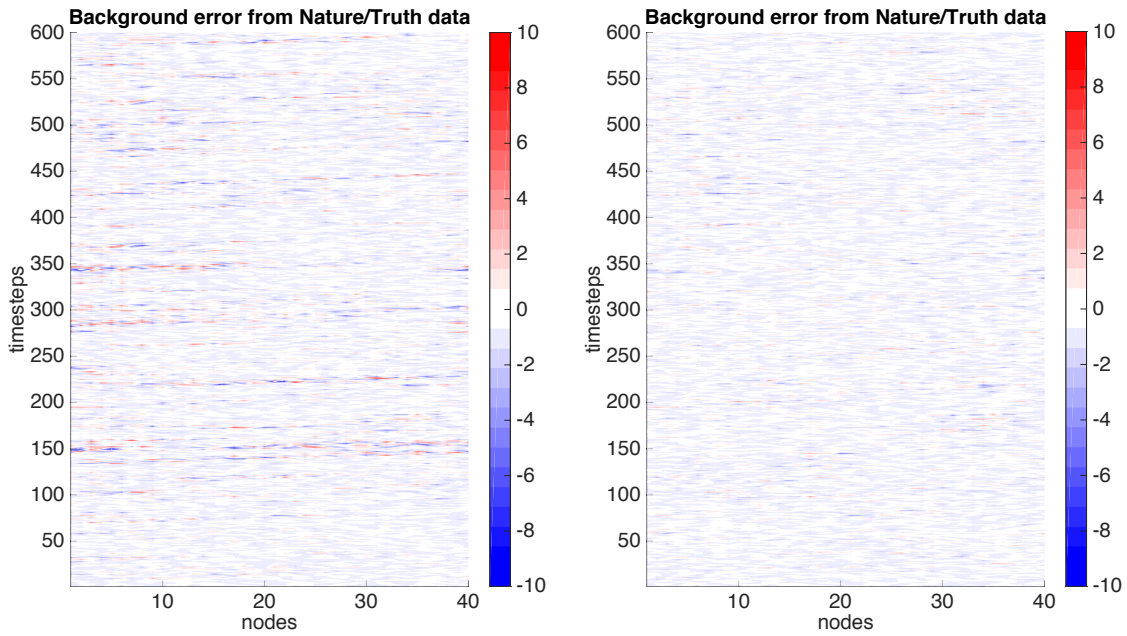

Figure 4. Forecast error for (a) LETKF and (b) LPF, using an analysis cycle window length dt=0.5 (60-hr), localization radius r=2 grid points, k=100 ensemble members, and l=80 observations sampled randomly on the domain (observation locations and errors are identical for both methods).

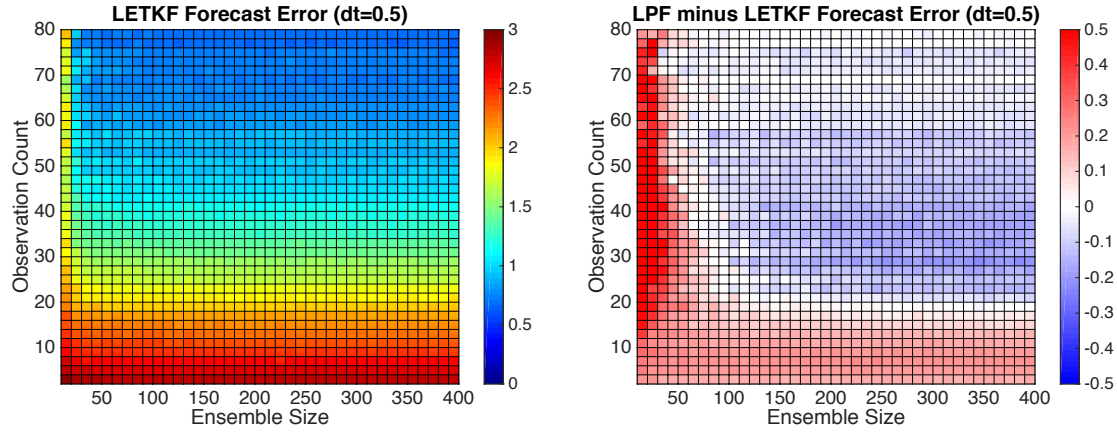

Figure 5. Forecast error for (a) LETKF and (b) LPF minus LETKF. LPF reduces error in highly sampled cases with larger observation coverage. The LPF increases error in poorly sampled cases and with low observation coverage. Each cell represents one experiment case; absolute errors are averaged over the entire domain for 600 analysis cycles for each case.

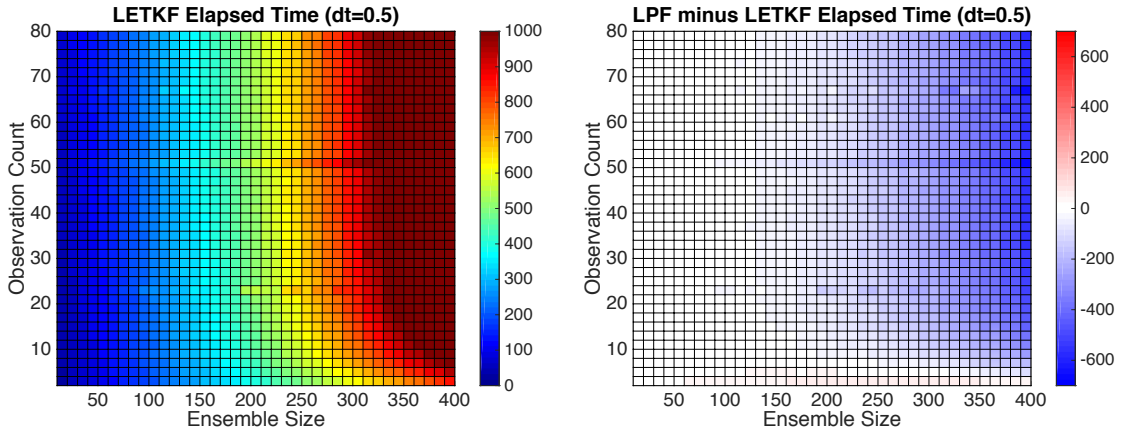

Figure 6. Elapsed time in seconds for (a) LETKF and (b) LPF minus LETKF.

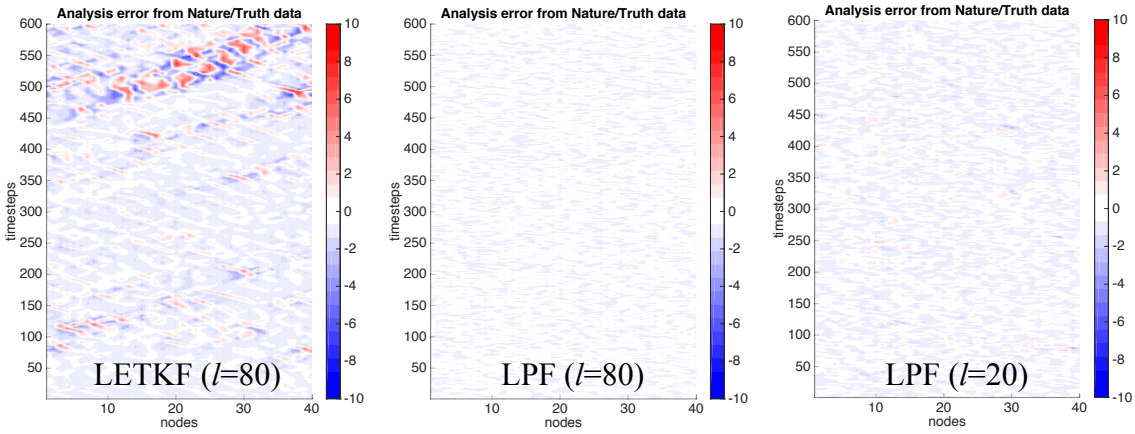

Figure 7. Analysis error for (a) LETKF and (b) LPF, using l=80 observations and k=100 ensemble members. The observations used between (a) and (b) are identical. In (c), the number of observations for LPF is reduced to l=20, but improvement in accuracy versus LETKF remains.

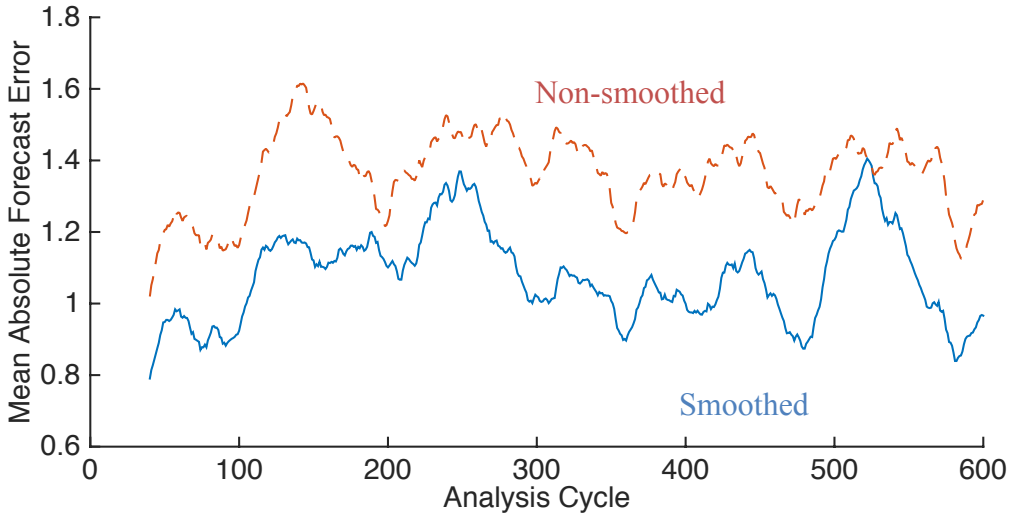

Figure 8. 40-cycle moving average of the mean absolute forecast error, comparing the Non-smoothed (dashed red) and Smoothed (solid blue) LPF analyses. The ensemble space smoothing improves the forecast accuracy.

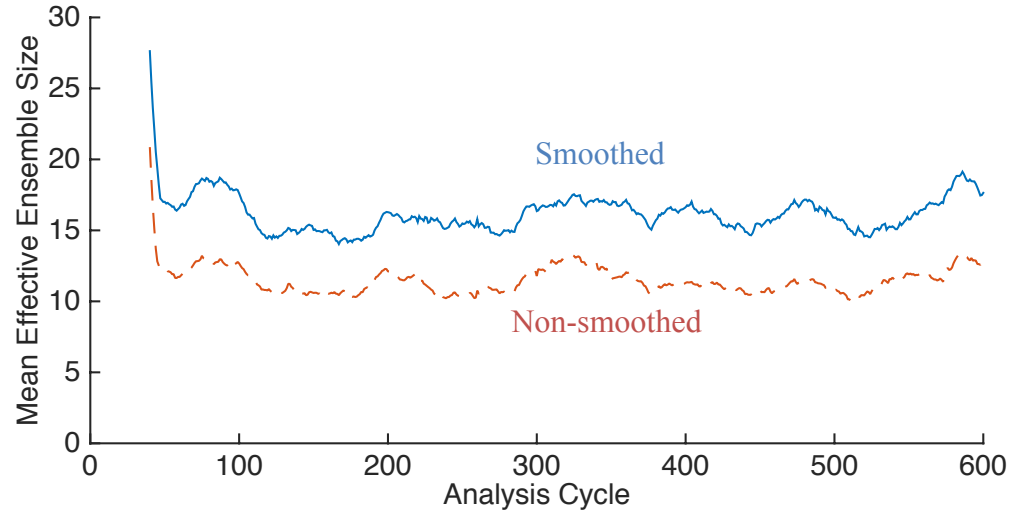

Figure 9. Illustrating the impact of ensemble space smoothing on the effective ensemble size N_{eff}.. Here we show the 40-cycle moving average of N_{eff} for the Smoothed (solid blue) and Non-smoothed (dashed red) cases.