

A new formulation of vector weights in localized particle filters

Zheqi Shen,^a Youmin Tang^{a,b*} and Xiaojing Li^{a,b}

^aState Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, Hangzhou, China

^bEnvironmental Science and Engineering, University of Northern British Columbia, Prince George, Canada

*Correspondence to: Y. Tang, Environmental Science and Engineering, University of Northern British Columbia, Prince George, Canada. E-mail: ytang@unbc.ca

Particle filters (PFs) constitute a sequential data assimilation method based on the Monte Carlo approximation of Bayesian estimation theory. Standard PFs use scalar weights derived from the likelihood of the approximate posterior probability density functions (PDFs) of observations and use resampling schemes to generate posterior particles. However, the scalar weights approach interferes with the localization algorithm and often results in filter degeneracy.

Recently, a localized particle filter (LPF) was developed by extending the scalar weights of PFs to vector weights, which produces various (local) posterior PDFs for different model grids and variables. With a sampling and merging approach in the resampling, a LPF can effectively solve the filter degeneracy problem and offer a practical, efficient algorithm for localization. However, this algorithm assumes the variations in the weights of a state variable of neighbouring grids to be continuous and uses a spatially linear interpolation of PF weights to determine the local weights.

In this paper, we first analyse the possible concerns associated with the linear continuity of PF weights. This assumption is found to challenge the theoretical properties of nonlinear and non-Gaussian variations in weights and alleviate the intrinsic spatial variations of PF weights. On this basis, we propose a new algorithm to produce vector weights for PFs for neighbouring grids. Numerical experiments using the Lorenz '96 model show that our new LPF performs better than the existing LPF algorithm, indicating the advantages and potential applications of this new algorithm of vector weights in the field of data assimilation.

Key Words: data assimilation; particle filter; localization

Received 05 February 2017; Revised 04 August 2017; Accepted 12 October 2017; Published online in Wiley Online Library

1. Introduction

In the data assimilation community, particle filters (PFs) have recently attracted widespread attention. PFs are sequential Monte Carlo methods based on particle representation of probability density functions (PDFs), which can be applied to any state-space model with no Gaussian assumption. The particles are a discrete set of model states, which approximately represent the model PDF using a weighted combination. Particles are essentially equivalent to the ensemble members in the ensemble-based filters, such as the ensemble Kalman filter (EnKF). In the analysis stage of PFs, the weights of particles are modified according to the observation, such that the observed information is incorporated into the swarm of particles to produce the posterior PDF from the prior PDF. The resampling technique, which removes the particles with very low weights while duplicating those with high weights, is frequently employed in standard PFs to suppress filter degeneracy, wherein almost every particle has negligible weight and contributes little

to representing the PDFs. The resampling scheme cannot prevent filter degeneracy with limited computational resources, especially when it considers high-dimensional systems. Snyder *et al.* (2008) showed that the ensemble size required for a successful PF (standard PF) scales exponentially with the dimension of the system. This is apparently impossible for the data assimilation of large oceanic or atmospheric models.

Many efforts have been made to develop a variant of PF that works for high-dimensional systems. For example, Papadakis *et al.* (2010) proposed a weighted ensemble Kalman filter (WEnKF), which uses the analysis of the PDF of the EnKF as the proposal density from which the particles are drawn. Similarly, Nakano (2014) also developed a hybrid algorithm using the ensemble transform Kalman filter (ETKF) as the proposal density. Ades and van Leeuwen (2015) and Zhu *et al.* (2016) proposed the equal-weights PF and implicit equal-weights PF, the bases of which are to use well-designed proposal densities (directly or implicitly) to ensure that the majority of particles have almost equivalent

weights at the time of observation. Additionally, Chorin *et al.* (2010) developed an implicit PF that uses gradient descent minimization combined with random maps to find the regions of high probabilities, avoiding the calculation of Hessians. Frei and Knsch (2013) and Shen and Tang (2015) combined the analysis schemes of the standard PF and EnKF and developed the so-called ensemble Kalman PF (EnKPF), which can be regarded as a blend of both methods. The particles are drawn from a well-prescribed proposal density which is close to that of the observations, such that their weights are relatively large and have similar magnitudes. However, this approach is often inefficient for high-dimensional systems in which the weights can still vary considerably, as discussed below.

In the aforementioned PFs, the weights are computed as scalars, as in the standard PF. In other words, a scalar value corresponding to the likelihood of observation is used to represent the weight for the whole particle. It should be noted that a particle is typically a vector in a multiple-dimensional model state (e.g. spatial grids). Thus, the scalar approach implies that the differences between the predictions and observations at different spatial grids or observational sites are accumulated to produce a scalar weight. Consequently, the weights can vary considerably for different particles, even reaching significant differences of orders of magnitude for a high-dimensional system. This is likely to result in the filter degeneracy.

A potential improvement to PFs is to extend the scalar weights to vector weights, which can well consider the different influences of observations on each model grid in determining the weights. This approach also implies that the role of the observation in determining the weights can be localized, as suggested in van Leeuwen (2003) and van Leeuwen (2009). Towards the goal of developing a computationally efficiency scheme of the PF for high-dimensional systems, two major efforts have been made in previous studies, resulting in two localized PF (LPF) schemes.

Poterjoy (2016) first proposed the LPF method (called P2016 hereafter) using the framework similar to the localized ensemble adjustment Kalman filter (EAKF; Anderson, 2003), in which localization is done implicitly by multiplying the covariance matrix by a distance-dependent function, which decays to zero beyond a certain distance. Almost simultaneously, Penny and Miyoshi (2016) proposed another LPF, which used a framework similar to the Local Ensemble Transform Kalman Filter (LETKF; Hunt *et al.*, 2007), and performed localization explicitly by considering only the observations in a region surrounding the analysis grid.

The above two localization schemes represent different strategies in developing computationally efficient PF algorithms for high-dimensional systems. Both methods apply localization in the background covariance. However, Hunt *et al.* (2007) found that similar effects can be achieved by multiplying the entries in the background covariance (B-localization) or in the inverse observation-error covariance matrix (R-localization) by the localization factor which decays from one to zero as the distance of the observations from the analysis grid point increases. It is interesting to compare the two algorithms but this is beyond the scope of this work.

In this study, we focus on the improvement of the P2016 LPF algorithm and wish to develop a finer algorithm in the framework of implicit localization scheme. Specifically, P2016 LPF computes the likelihood at each observational site and uses an interpolation method to obtain varied weights for different model grids. The interpolation coefficients are determined by the distance of the model grid point to the observation. In addition, it also uses a sampling and merging approach to implement the resampling procedure for vector weights. The performance of the LPF is tested with the Lorenz '96 model in P2016. As reported in Nakano *et al.* (2007), a standard PF requires over 30 000 particles to obtain a non-degenerate result superior to that of an EnKF. P2016 showed that the LPF can prevent filter degeneracy with tens of particles, and beat the EAKF with no more than one hundred particles, especially when the measurement operator is nonlinear.

In this study, a new formulation of vector weights is proposed by improving the P2016 interpolation scheme. In the framework of vector weights, each model grid has a corresponding weight computed using the products of likelihoods of each single scalar observation. Borrowing the idea of R-localization in Hunt *et al.* (2007) and Chen and Reich (2015), the contribution of each observation can be localized by inflating the observation-error covariance according to the distance from the observation site. Accordingly, a factor which is inversely proportional to this distance can function as the interpolation coefficient for the contribution of each single observation. In short, the interpolation in our proposed scheme is plotted in a state space rather than in a weight (probability) space. We give the details in section 3.

This paper is organized as follows. In section 2, the standard PF with scalar weights is introduced as a Monte Carlo approximation of the Bayesian recursive filter. In section 3, we extend the scalar weights in the standard PF, obtaining a new formulation for vector weights. The sampling and merging technique is then introduced and is shown to generate particles with local PDFs, while barely violating the continuity of the state field. Numerical experiments are conducted in section 4 to compare the new LPF with the P2016 LPF and the EAKF. The conclusions are presented in the last section.

2. Standard PF with scalar weights

The PF algorithm is based on the Bayesian estimation theory, which can be considered to be a version of the Monte Carlo realization of the Bayesian recursive filter. The Bayesian filter aims to provide the full PDF of a model state instead of giving only the state mean and covariance, as in the EnKF. If an initial PDF $p(x_0)$ is available, the filter estimates the posterior PDFs $p(x_k|Y_k)$ sequentially using the observational data at different time steps as $Y_k = \{y_1, y_2, \dots, y_k\}$, according to Bayes' rules.

A Bayesian recursive filter consists of two stages: a prediction step and a correction (or analysis) step.

1. The prediction step corresponds to the model integration. From a probabilistic perspective, the prior PDF at time step t_k (denoted $p(x_k|Y_{k-1})$) is derived from the posterior PDF at t_{k-1} (denoted $p(x_{k-1}|Y_{k-1})$) via:

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|Y_{k-1}) dx_{k-1}, \quad (1)$$

in which $p(x_k|x_{k-1})$ is the transitional PDF, implying the probabilistic effect of the model integration, and need not be expressed explicitly.

2. The analysis step uses the observation at the current time, written as y_k , to update the prior PDF $p(x_k|Y_{k-1})$ and generate the posterior PDF $p(x_k|Y_k)$, which is done using Bayes' rule:

$$p(x_k|Y_k) = \frac{p(y_k|x_k) p(x_k|Y_{k-1})}{A}, \quad (2)$$

in which $p(y_k|x_k)$ represents the likelihood, i.e. the probability that the forecasted x_k at time step t_k coincidences with y_k . The denominator A is expressed as

$$A = p(y_k|Y_k) = \int p(y_k|x_k) p(x_k|Y_{k-1}) dx_k,$$

which is typically regarded as a normalization factor.

The standard PFs use the Monte Carlo method to perform the Bayesian filter. Here, a particle is similar to an ensemble member in the EnKF, which is an independent model output. With a PF,

a combination of particles is used to approximate the posterior PDFs in Eqs (1) and (2), written as

$$p(x_k|Y_k) = \sum_{n=1}^N w_k^n \delta(x_k - x_k^n), \quad (3)$$

$$p(x_{k-1}|Y_{k-1}) = \sum_{n=1}^N w_{k-1}^n \delta(x_{k-1} - x_{k-1}^n), \quad (4)$$

where x_k is the model state at the time step t_k , x_k^n refers to the n th particle and N is total the number of particles. δ represents the Dirac delta function, which is 0 everywhere except zero, and the integration of δ over any interval including zero is 1. w_k^n is a scalar weight corresponding to x_k^n , satisfying $\sum_{n=1}^N w_k^n = 1$.

Substituting Eq. (4) into Eq. (1):

$$\begin{aligned} p(x_k|Y_{k-1}) &= \int p(x_k|x_{k-1}) \sum_{n=1}^N w_{k-1}^n \delta(x_{k-1} - x_{k-1}^n) dx_{k-1} \\ &= \sum_{n=1}^N w_{k-1}^n \int p(x_k|x_{k-1}) \delta(x_{k-1} - x_{k-1}^n) dx_{k-1} \\ &= \sum_{n=1}^N w_{k-1}^n p(x_k|x_{k-1}^n). \end{aligned}$$

If the prior particles at t_k are proposed by a given PDF $q(x)$, the above equation can be written as

$$p(x_k|Y_{k-1}) = \sum_{n=1}^N \frac{w_{k-1}^n p(x_k^n|x_{k-1}^n)}{q(x_k^n)} \delta(x_k - x_k^n).$$

Using Eq. (2), the posterior PDF can be given as

$$p(x_k|Y_k) = \sum_{n=1}^N w_{k-1}^n \frac{p(y_k|x_k^n) p(x_k^n|x_{k-1}^n)}{A q(x_k^n)} \delta(x_k - x_k^n).$$

When compared with Eq. (3),

$$w_k^n \propto w_{k-1}^n \frac{p(y_k|x_k^n) p(x_k^n|x_{k-1}^n)}{q(x_k^n)}. \quad (5)$$

Here, the proportional symbol is used to show that a normalization operation is required to update the weights.

Gordon *et al.* (1993) proposed the bootstrap PF, in which $q(x) = p(x|x_{k-1}^n)$, such that Eq. (5) can be simplified by

$$w_k^n \propto w_{k-1}^n p(y_k|x_k^n). \quad (6)$$

To reduce the chance of filter degeneracy, PFs employ certain forms of resampling or selection steps after the updated weights are calculated. The resampling scheme essentially removes the particles with very small weights and duplicates those with large weights. For the scalar weights, a deterministic resampling scheme was developed by Kitagawa (1996), in which the weights are compared with multiples of $1/N$ to decide the number of copies of each particle to be included in the resampled analysis ensemble. The bootstrap PF with resampling is also known as the sequential importance resampling PF (SIR-PF), which uses the scalar weights determined by

$$\begin{aligned} w_k^n &\propto p(y_k|x_k^n) \\ &= \frac{1}{\sqrt{2\pi|R|}} \exp\left[-\frac{1}{2}\{y_k - h(x_k^n)\}^T \mathbf{R}^{-1} \{y_k - h(x_k^n)\}\right]. \end{aligned} \quad (7)$$

Here, y_k is a vector containing all the observations at different sites at time steps of t_k , \mathbf{R} is the corresponding error covariance matrix and $|R|$ is its determinant, and the measurement function h maps the model states onto those observational sites. Equation (7) essentially assumes that the observational errors are Gaussian. If they are not Gaussian in practice, Eq. (7) uses a Gaussian approximation for the actual PDF.

3. Particle filter with vector weights

3.1. New formulation of vector weights

In SIR-PF, the important particles are duplicated while others are deleted. This will often result in particle impoverishment due to the rapid decrease in the diversity of the particles in the resampling step for reasons such as insufficient lengths of assimilation windows or weak nonlinearities. As a result, a large number of particles are required to prevent both filter degeneracy and particle impoverishment. Even worse, Bengtsson *et al.* (2003) showed that the number of particles has to grow exponentially with the number of state dimensions to avoid filter degeneracy, making this PF very impractical for large models.

The main reason why filter degeneracy is very likely to occur in standard PFs is the scalar form of its weights. As shown in Eq. (7), one scalar value is used as the weight of each particle, ignoring the fact that each particle actually represents a multiple set of entries of the model state vector. This equation also implies that the scalar weight acts globally, i.e. every different entry of the state vector has the same weight, and only one posterior PDF is used to describe the probability features of the whole state space. This provides an intuitive explanation for filter degeneracy and particle impoverishment problems. When the more statistical information is incorporated into this scalar with the increase in the dimension of the state vector, it becomes increasingly difficult to approximate the posterior PDF using a finite number of particles. Consequently, it is important to introduce vector weights and to localize the roles of each observation in determining those weights. In the rest of this section, we will extend the scalar weights of standard PFs to vector weights.

To simplify the notation, we ignore the subscript for time step (k) in the following sections and consider only one analysis step. Assuming that the observational vector y contains m components, written as $\{y_1, y_2, \dots, y_m\}$, which represent the single scalar observational dataset at each site, the scalar weights in Eq. (7) can be written as

$$\begin{aligned} w^n &\propto \exp\left[-\frac{1}{2} \sum_{i=1}^m \{y_i - h_i(X^n)\}^2 / \sigma_i^2\right] \\ &= \prod_{i=1}^m \exp\left[-\frac{1}{2} \{y_i - h_i(X^n)\}^2 / \sigma_i^2\right]. \end{aligned} \quad (8)$$

where σ_i^2 is the variance of y_i , and h_i is the measurement function that maps the model states onto the observational site corresponding to y_i . Here, we assume that the individual scalar observational data points are uncorrelated with each other, such that the \mathbf{R} in Eq. (7) is a diagonal matrix. Equation (8) computes the global scalar weights by combining the effects of each observation. Similarly, we can develop the local weights for each entry of the state vector $X^n = \{x_1^n, \dots, x_j^n, \dots, x_L^n\}$ via slight modifications to Eq. (8). To avoid confusion, we use X^n to denote the model states as vectors with L entries hereafter. In addition, the corresponding vector weights are written as $W^n = \{w_1^n, \dots, w_j^n, \dots, w_L^n\}$. The weights for each entry x_j can be given by extending Eq. (8) and using some localization techniques.

In the ensemble-based Kalman filters (EnKF, EAKF, etc), localization techniques are used to suppress the spurious correlations between distant locations, which are generally introduced by insufficient ensemble members. As mentioned,

these localizations are either done explicitly (Hunt *et al.*, 2007) or implicitly (Anderson, 2003). The former strategy is also referred to as local analysis, in which only the observations from the region surrounding the localization analysis are taken into consideration. The latter strategy is referred to as the covariance localization, in which a continuous function whose values are inversely proportional to the distances from the observation sites is employed to generate the multiplication factors for the state increments, i.e.

$$l_{ij} = \Omega(d_{ij}; c) = \begin{cases} -\frac{1}{4}\left(\frac{d_{ij}}{c}\right)^5 + \frac{1}{2}\left(\frac{d_{ij}}{c}\right)^4 + \frac{5}{8}\left(\frac{d_{ij}}{c}\right)^3 - \frac{5}{3}\left(\frac{d_{ij}}{c}\right)^2 + 1, & \text{for } 0 \leq d_{ij} \leq c, \\ \frac{1}{12}\left(\frac{d_{ij}}{c}\right)^5 - \frac{1}{2}\left(\frac{d_{ij}}{c}\right)^4 + \frac{5}{8}\left(\frac{d_{ij}}{c}\right)^3 + \frac{5}{3}\left(\frac{d_{ij}}{c}\right)^2 - 5\left(\frac{d_{ij}}{c}\right) + 4 - \frac{2}{3}\left(\frac{d_{ij}}{c}\right)^{-1}, & \text{for } c \leq d_{ij} \leq 2c, \\ 0, & \text{for } d_{ij} \geq 2c, \end{cases} \quad (9)$$

where d_{ij} represents the distance between the observational site values y_i and the locations of the model grid values x_j . The parameter c is related to the decorrelation length. The value of l_{ij} is equal to zero when d_{ij} is larger than $2c$. Additionally, $l_{ij} = 1$ for $d_{ij} = 0$, decreasing monotonically with d_{ij} until it reaches 0 at $d_{ij} = 2c$. The function Ω in Eq. (9) is referred to as the Gaspari–Cohn (GC) function (Gaspari and Cohn, 1999), which is frequently used in the localization schemes of the EnKF.

The GC function is a compact-supported function and can also be used in PFs. P2016 updates the local weight w_j^n using a single observation y_i via a term $(\hat{w}_j^n)_{y_i}$, which combines the value at 1 and the likelihood of y_i in Eq. (8) and is weighted by l_{ij} as follows:

$$(\hat{w}_j^n)_{y_i} = p(y_i|X_j^n) l_{ij} + (1 - l_{ij}), \quad (10)$$

where $p(y_i|X_j^n)$ is proportional to the i th term of the product in Eq. (8). Equation (10) is repeatedly applied to each single observation y_i until all the observations are assimilated, leading to the local weight

$$w_j^n = \prod_{i=1}^m \{p(y_i|X_j^n) l_{ij} + (1 - l_{ij})\}. \quad (11)$$

Equation (10) can be understood as a linear interpolation between the likelihoods $p(y_i|X_j^n)$ and 1 to determine the weight for a given grid j , which inherently assumes that the likelihoods are continuous in space. This can cause concerns since they incorporate the state variables with observations by nonlinear functions as in Eq. (8). Next, we will consider a new formulation by removing this assumption.

We assume that each component x_j corresponds to the model state at a certain model grid point whose distances to each observational site values of y_i are l_{ij} , for $i = 1, \dots, m$. The weight for x_j^n can then be given by

$$w_j^n \propto \exp \left[-\frac{1}{2} \sum_{i=1}^m l_{ij} \{y_i - h_i(X^n)\}^2 / \sigma_i^2 \right] = \prod_{i=1}^m \exp \left[-\frac{1}{2} l_{ij} \{y_i - h_i(X^n)\}^2 / \sigma_i^2 \right]. \quad (12)$$

By comparing this equation with Eq. (8), it can be found that l_{ij} is applied to each term in the exponent.

We can write the weights up to the single observation y_{i-1} as $(w_j^n)_{y_{i-1}}$, and write each term in the product in Eq. (12) by $(\hat{w}_j^n)_{y_i}$, i.e.

$$(\hat{w}_j^n)_{y_i} = \exp \left[-\frac{1}{2} l_{ij} \{y_i - h_i(X^n)\}^2 / \sigma_i^2 \right], \quad (13)$$

such that the vector weights can be determined recursively by assimilating each single observation into the weights by

$$(W^n)_{y_i} = (W^n)_{y_{i-1}} (\hat{W}_{y_i}), \text{ for } i = 1, \dots, m \quad (14)$$

and $W^n = (W^n)_{y_m}$ if all the available data are assimilated sequentially. From Eq. (13), the range of influence of y_i is restricted to the vicinity of the observational site, since $(\hat{w}_j^n)_{y_i} = 1$ holds for every $n = 1, 2, \dots, N$ if $l_{ij} = 0$. Additionally, the variance of the observation is inflated to σ_i^2 / l_{ij} , implying that the uncertainty of y_i increases as the model grid points move away from the observational site. This is similar to the R-localization idea in Hunt *et al.* (2007), which is shown to have a similar effect to the localization in background covariance. Likewise, a normalization step is still required for each x_j to satisfy that $\sum_{n=1}^N w_j^n = 1$, for every $j = 1, 2, \dots, L$.

3.2. Resampling scheme for vector weights

Equation (12) can produce a scalar weight w_j^n for the j th entry of the state vector. A vector weight can be obtained by assembling these entries over the model grids. Using the vector weights to resample particles in the analysis step is a challenge. Apparently, the resampling scheme as used in SIR-PF cannot be directly applied since the different scalar weights result in different particles being selected in different areas of the domain when constructing a new particle. This will cause the model state to be discontinuous in both space (analysis step) and time (prediction step).

To solve the problem of discontinuities, P2016 use a sampling and merging strategy. As soon as each y_i is assimilated to update the local weights, the prior particles are merged with the resampled particles to generate samples locally from the local distributions. In this study, we will adopt this strategy with some important modifications to fit the new formulation of the weights.

The most important features for the sampling and merging strategy in P2016 include the following:

- (i) single observations are assimilated one by one to update local weights;
- (ii) the sampling and merging strategy is invoked every time a single observation y_i is assimilated;
- (iii) only the scalar weights at the observational site values y_i are used in the standard resampling schemes to determine the resampled particles;
- (iv) the resampled particles are merged with the prior particles using the parameters r_1 and r_2 , which vary with the distance from y_i ;
- (v) r_1 and r_2 are computed to match the local PDF for each model grid.

The state vectors and weights up to the observation point y_{i-1} are written as $(X^n)_{y_{i-1}}$ and $(W^n)_{y_{i-1}}$ for $n = 1, \dots, N$, respectively; the sampling and merging procedure for the single observation y_i can be briefly described as follows.

1. Compute and normalize the scalar weights according to y_i by

$$\tilde{w}^n = \exp \left[-\frac{1}{2} \{y_i - h_i(X^n)\}^2 / \sigma_i^2 \right], \quad \text{for } n = 1, \dots, N. \quad (15)$$

2. Apply the standard resampling scheme with $\{\tilde{w}^n, n = 1, \dots, N\}$ to determine the resampling indices k_1, k_2, \dots, k_N .
3. Compute $(\hat{w}_j^n)_{y_i}$ using Eq. (13), update the vector weights using Eq. (14), and compute the normalization factor for each j using $S_j = \sum_{n=1}^N (w_j^n)_{y_i}$.

4. Merge the prior particles with the resampled particles using the formula

$$(x_j^n)_{y_i} = (\bar{x}_j)_{y_i} + r_{1,j} \{(x_j^{k_n})_{y_{i-1}} - (\bar{x}_j)_{y_i}\} + r_{2,j} \{(x_j^n)_{y_{i-1}} - (\bar{x}_j)_{y_i}\}, \quad (16)$$

where $(\bar{x}_j)_{y_i} = \sum_{n=1}^N \frac{(w_j^n)_{y_i}}{S_j} x_j^n$ is the weighted sum of the prior particles. The parameters $r_{1,j}$ and $r_{2,j}$ determine the portions of the resampled and prior particles and they can be given as

$$c_j = \frac{N(1 - l_{ij})}{l_{ij} S_j}, \quad (17)$$

$$r_{1,j} = \frac{(\sigma_j)_{y_i}^2}{\sqrt{\frac{1}{N-1} \sum_{n=1}^N [(x_j^{k_n})_{y_{i-1}} - (\bar{x}_j)_{y_i} + c_j \{(x_j^n)_{y_{i-1}} - (\bar{x}_j)_{y_i}\}]^2}}, \quad (18)$$

$$r_{2,j} = c_j \times r_{1,j}, \quad (19)$$

where $(\sigma_j)_{y_i}^2 = \sum_{n=1}^N \frac{(w_j^n)_{y_i}}{S_j} \{x_j^n - (\bar{x}_j)_{y_i}\}^2$ is the weighted variance of the prior particles with the local weights.

The derivations of Eqs (17)–(19) can be found in Appendix A of P2016. These formulae are derived to adjust the particles locally in order to make the local members $\{x_j^n, n = 1, \dots, N\}$ have the same means and variances as the weighted means and weighted variances of the prior particles, when using local weights. These adjustments are accomplished gradually by applying Eq. (16) whenever a single observation is assimilated to update the weights. Using the sampling and merging strategy, both P2016 LPF and the new LPF linearly combine the old and localized particles to generate the new particles, potentially breaking the nonlinear balances in the model. However, the continuity of the model fields is barely violated because each adjustment involves the continuous GC function in Eq. (17).

Similar to P2016, we also use a special treatment for the weights in Eq. (15) to stabilize the algorithm. In step 1, when \tilde{w}^n ($n = 1, 2, \dots, N$) is computed using Eq. (15), it is modified by

$$\tilde{w}^n = (\tilde{w}^n - 1)\alpha + 1 \quad (20)$$

before the normalization step. This modification actually sets a threshold value for the weights. Here, we assume that α is slightly smaller than 1, e.g. $\alpha = 0.98$. Thus, if at least one weight in the set $\{\tilde{w}^n\}$ is not too small, the weights will barely change with Eq. (20). Conversely, if all \tilde{w}^n are very small (e.g. $\tilde{w}^n < 0.01$ for all n), Eq. (20) will make all the weights almost identical. This treatment can reduce the chance of the filter degeneracy caused by particle divergence, in which all the weights are so small that their relative magnitudes are meaningless. Equation (20) is also applied to the vector weights for each component as $\{w_j^n, n = 1, \dots, N\}$ in step 3 before the sum S_j is computed.

Furthermore, P2016 uses the Kernel Density Distribution Mapping (KDDM) approach proposed by McGinnis *et al.* (2015) to match the high-order moments. However, we ignore the KDDM procedure in the following experiment, since it shows only minor improvements to the PF results in the numerical experiment using Lorenz '96 model, and we mainly focus on the different formulations of both localized PFs.

4. Numerical experiments

In this section, we use the Lorenz '96 model as the test to compare the performances of our new localized PF with P2016 LPF and the sequential EAKF by Anderson (2003).

The simplified model proposed by Lorenz (1996) represents an atmospheric variable x at L with equally spaced points around a circle with a constant latitude. The j th component is propagated forward in time according to the differential equation

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F, \quad (21)$$

where $j = 0, 1, \dots, L - 1$ represents the spatial coordinates. The boundary conditions are cyclical, i.e., $x_{L-1} = x_{-1}$, $x_L = x_0$, and $x_{L+1} = x_1$. In this study, L is set to 36; that is, the dimension of the state vector is 36. This specification implies that the distance between the two adjacent grid points roughly represents a length of 10° of longitude. F is a constant external forcing term that dominates the system dynamics and is set to 8. The time unit equals 5 days in this model.

The dynamical model is integrated one step forward using the Runge–Kutta fourth-order method. One time step is set to be 0.05 (i.e. 6 h). To generate the data for the experiment, we run this model from the initial conditions

$$x_j = \begin{cases} 1 & \text{for } j \neq 1, \\ 1.0001 & \text{for } j = 1. \end{cases}$$

We integrate the model for 14 400 steps (or 10 years) to allow the fluctuations in the system to develop sufficiently. After that, the data are generated at every 4 timesteps with errors having a standard deviation of 1. One of every two sites is observed. That is, we can only observe x_j if j is an odd number. The initial conditions for the data assimilation experiment are given by perturbing the true state vector with random Gaussian noises with a mean value 0 and variance of 2.0. To make the experiment more generalizable and complicated, we assume that the observation is nonlinearly related to the state space, i.e. a nonlinear measure function $y = h(X)$. In the first case, we use $h(X) = |X|$ in the experiment.

First, we perform the sensitivity experiments to tune the localization parameters for the different ensemble sizes for each method. The performance of each method is evaluated against the true values using the average root-mean-square errors (RMSE) over all 36 variables. The RMSE is defined as

$$\sqrt{\frac{1}{36} \sum_{j=1}^{36} (x_j - x_j^{\text{true}})^2}$$

and is averaged over 10 000 model steps, i.e. 2500 data assimilation cycles. The EAKF, P2016 LPF and the new LPF are performed using the same initial ensemble members/particles. Ensemble sizes of $N = 10, 20, 40, 80, 160$ and 320 are considered, with localization parameters chosen from a range of $c \in \{1, 2, 4, 8, 16\}$.

Figure 1 shows that the optimal localization parameters used by each method are different for the different ensemble sizes. When the ensemble size is small, e.g. $N = 10$, a relatively small c should be used, which implies that each single observation can update very few model variables. If a larger c is used for a small ensemble size, this would result in large errors, either due to the spurious correlation in the EAKF or due to the filter degeneracy in the LPFs. A large ensemble size can effectively alleviate spurious correlations in EAKF, allowing us to use a larger c to update sufficient variables with one single observation. In LPFs, the filter degeneracy caused by a large c with small ensembles can be understood by Eqs (11) and (12). As shown in these equations, the localization weights are determined by adding l_{ij} to each term of the scalar weights in Eq. (8). While the parameter c is larger, the (local) vector weights more resemble the (global) scalar weights, which requires a very large number of particles to prevent filter degeneracy. In the extreme case of $c \rightarrow \infty$, each l_{ij} tends to the value of 1, and the local vector weights in Eq. (12) turn to the global scalar weights completely. When the ensemble size is small, a small c is preferable, such that the sampling and merging procedure will not delete and duplicate large portions of the

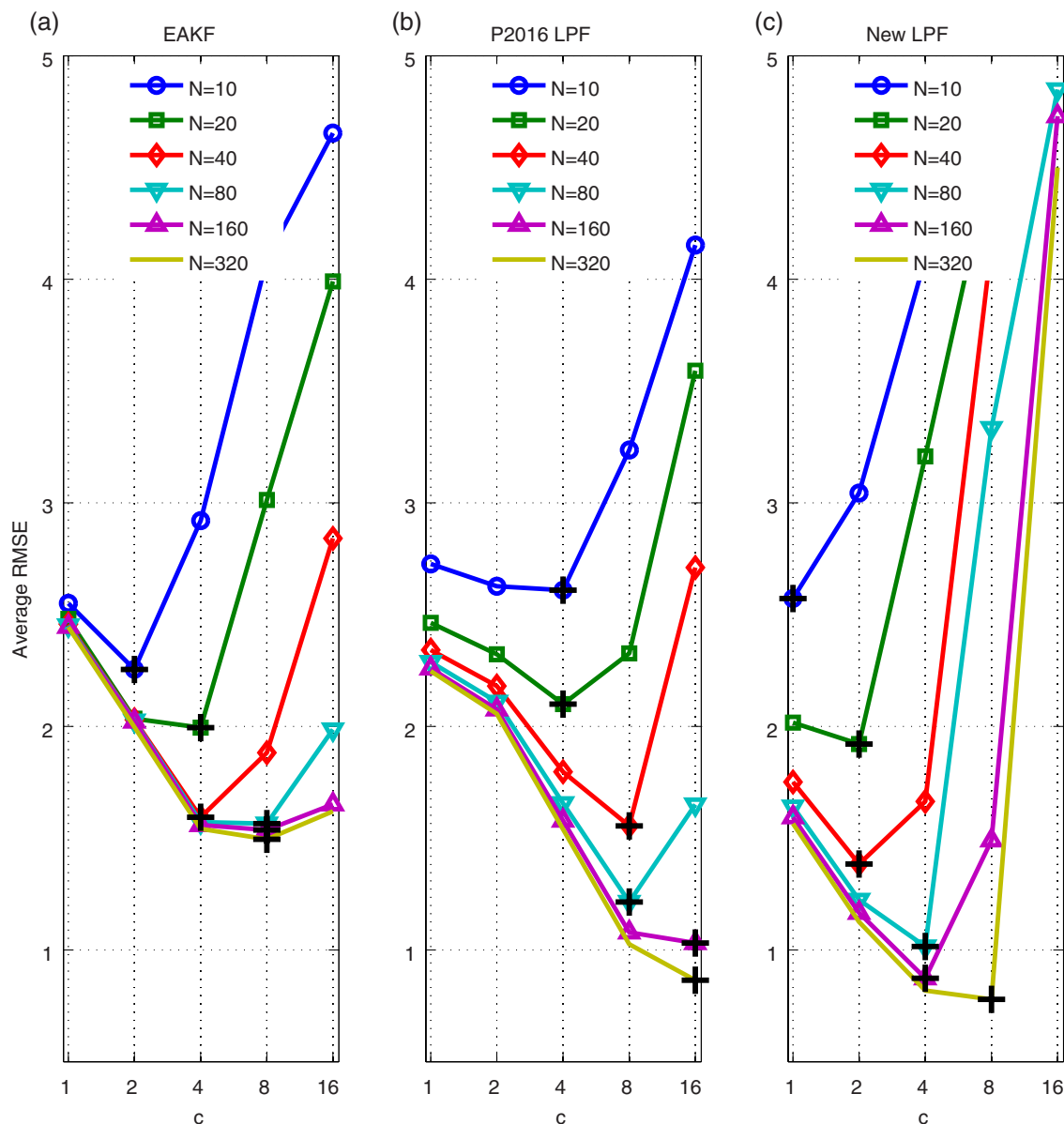


Figure 1. The average RMSE values over the data assimilation period with different ensemble sizes and different localization parameters for (a) EAKF, (b) P2016 LPF and (c) the new LPF, where the + indicates those c values corresponding to the minimal average RMSE. [Colour figure can be viewed at wileyonlinelibrary.com].

particles, thus maintaining the diversity of ensemble. Otherwise, the filter degeneracy is very likely to happen, making the errors very large, even intractable, such as the $N = 10$ and $N = 20$ cases for the new LPF in Figure 1.

However, such degeneracy can be prevented using suitable localization parameters in the new LPF. At least we can use $c = 0.5$ or 1 to fully localize the observations, such that filter degeneracy would not happen because the sampling and merging is done locally. Larger c can be used to assimilate more observations to improve the quality of the analyses, but it also brings a higher chance of producing filter degeneracy. The RMSE or ensemble spread can be computed to monitor whether filter degeneracy happens. An interesting property of the new LPF is that the filter degeneracy no longer occurs for any value of c smaller than a criterion value. This value can be obtained by the sensitivity experiments as is done in Figure 1. For example, in Figure 1(c), this value is approximately $c = 2$ for $N = 10, 20$, $c = 8$ for $N = 40, 80$ and $c = 16$ for $N = 160, 320$; beyond these values the RMSE becomes extremely large. If a criterion value of c is able to avoid filter degeneracy, this suggests that the diversity of the particles is sufficient, and so a smaller c with a corresponding lower chance of filter degeneracy will be safe. In practice, the localization parameter which leads to optimal LPF performance can be tuned between the criterion value and a minimum value corresponding to full localization.

This interesting finding may have practical significance for LPF applications.

Figure 2 shows the absolute analysis errors of all the variables for the last 1000 model steps. The EAKF, P2016 LPF and the new LPF are performed with 80 ensemble members or particles and these localization parameters given in Figure 1 are also used to achieve the optimal performance of each method. The inflation factor γ for the EAKF is tuned by the sensitivity experiments, and the value 1.03 is adopted in the data assimilation. The stabilizing factor α (Eq. (20)), when set to 0.98, is successful for both LPFs. The analysis errors clearly show that the errors of the LPF results are smaller than those of the EAKF results for most variables at most model steps; this strongly indicates that both the LPFs can significantly outperform the EAKF with affordable ensemble sizes. Furthermore, the analysis errors of the new LPF seem smaller than those of the P2016 LPF.

To further demonstrate the advantage of the new LPF, Figure 3 shows the corresponding RMSEs at each analysis step for each method. Due to the nonlinearity of the Lorenz '96 model, there are many peaks in the RMSE, corresponding to the growth of errors between the analysis steps. However, the new LPF outperforms the other two methods since the RMSEs of the new LPF are the smallest for most of the time. In addition, the average RMSE for the whole data assimilation period is also calculated for each method, by which the advantage of the new LPF can be further

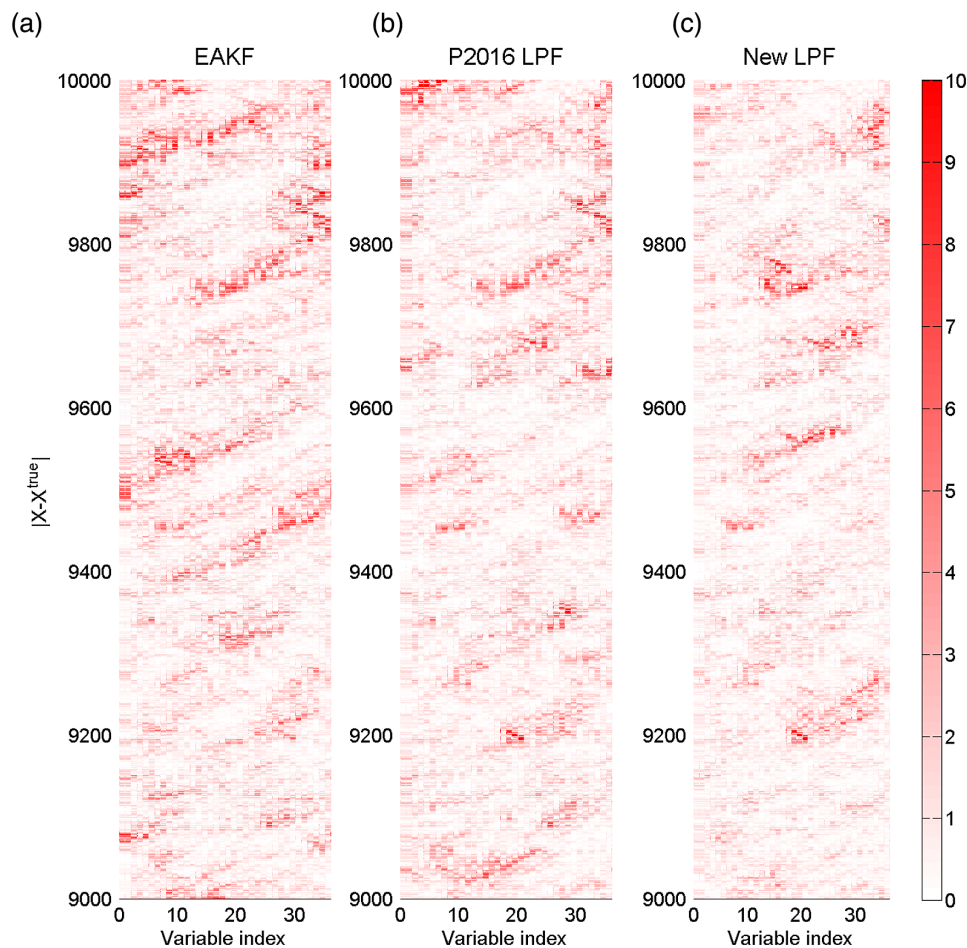


Figure 2. The absolute analysis errors of the 36 model variables for the last 1000 model steps using (a) the EAKF, (b) the P2016 LPF and (c) the new LPF. [Colour figure can be viewed at wileyonlinelibrary.com].

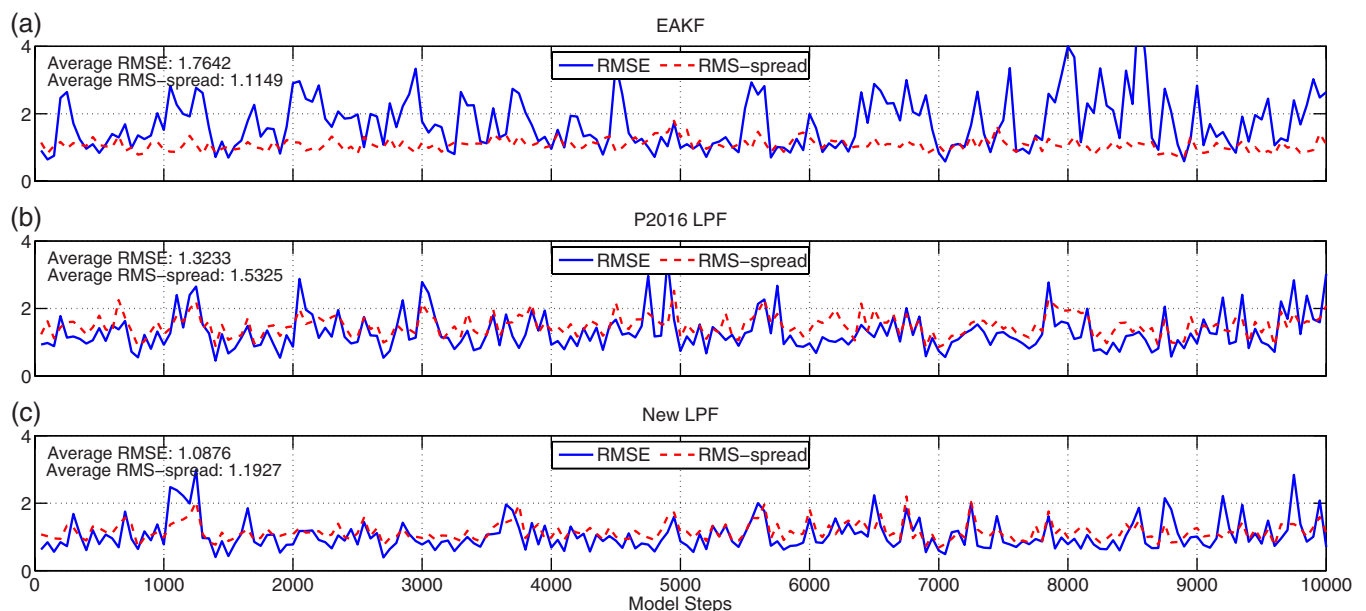


Figure 3. The RMSE and RMS spreads over all the variables against the model steps of the analysis results for (a) the EAKF, (b) the P2016 LPF and (c) the new LPF. [Colour figure can be viewed at wileyonlinelibrary.com].

confirmed. Moreover, we also compute the ensemble spread for each variable at each model step. The root-mean-square (RMS) spread over the 36 variables is shown in Figure 3 as a measure of the uncertainty of the analysis ensemble. Figure 3(c) shows a very impressive result in that the RMS spread is highly correlated with the RMSE, which indicates the consistency of the new LPF.

Figure 4 is a scatterplot of the average RMS spread against the RMSE for each of 36 model variables for three assimilation methods. The average RMS spread is obtained by averaging the

ensemble spread of each time step over the entire assimilation period (10 000 steps), and the RMSE of each variable is also calculated for the same period. Figure 4 is similar to those figures which were frequently used in the literature to evaluate ensemble forecast systems (e.g. Cheng *et al.*, 2010), showing the relationship of RMSE and RMS spread for each variable. Comparison among the three methods reveals that the new LPF results in the best relationship between the RMSE and RMS spread, indicating the superiority of this new methods in producing

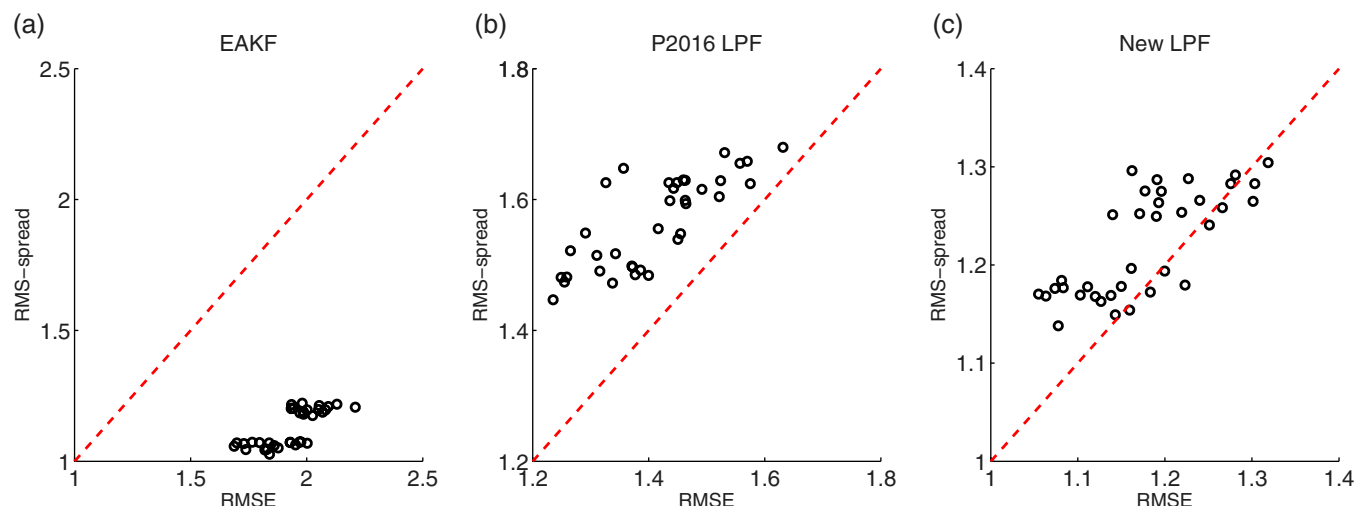


Figure 4. The scatterplot of the average RMS spread against the RMSE for each model variable, for (a) the EAKF, (b) the P2016 LPF and (c) the new LPF; the dashed line indicates when they are equal. [Colour figure can be viewed at [wileyonlinelibrary.com](#)].

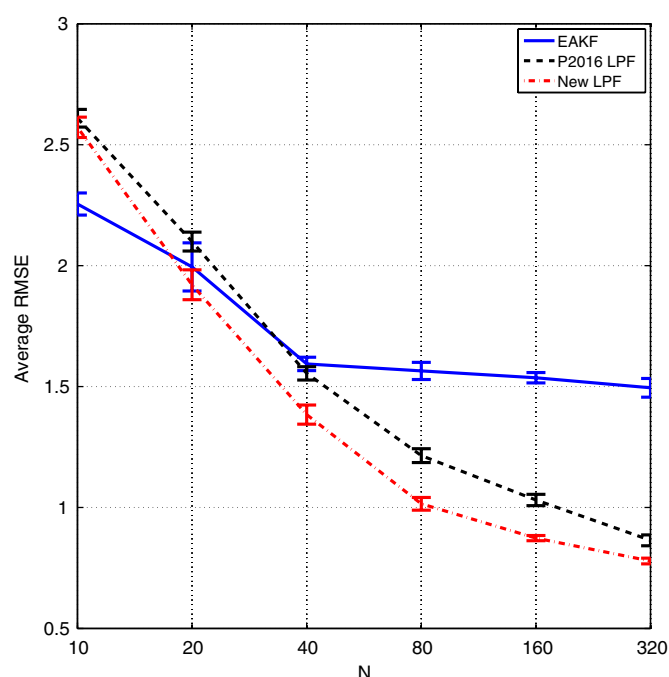


Figure 5. The average RMSEs of the data assimilation results as a function of ensemble size N for the EAKF, P2016 LPF and new LPF. [Colour figure can be viewed at [wileyonlinelibrary.com](#)].

good ensemble members (particles) and quantifying the uncertainty.

In the following experiment, we computed the RMSEs of all 36 variables of the state estimates and averaged these values over the assimilation period to measure the performances of each method against the different ensemble sizes, which is the same as the average RMSE in Figure 3. Figure 5 shows the average RMSEs as a function of ensemble size. To relieve the impact of the random initial ensembles, the experiment is repeated 10 times with different initial ensembles, and the mean and standard deviation of the average RMSEs are shown as solid lines and error bars in this figure. As seen here, the average RMSE of the EAKF seems to approach a saturated value when the ensemble size increases but cannot be significantly improved using more ensemble members. This is most likely because a Gaussian distribution is assumed to derive the EAKF analysis scheme, limiting its performance for nonlinear and non-Gaussian systems. Consistent with the conclusion in P2016, the P2016 LPF can outperform the EAKF with tens of particles. In addition, a significant improvement of the new LPF over P2016 LPF can be

observed in Figure 5. Another interesting feature in this figure is that the RMSE of both LPFs can be further decreased as the number of particle increases. This differs from the EAKF, since its RMSE tends to saturate to a steady value when the ensemble size increases beyond a certain number. The error bars measure the sampling errors, indicating the differences among these experiments in Figure 5 are statistically significant.

To further compare the EAKF, P2016 LPF and the new LPF, we consider other measurement functions. In the preceding experiments, a nonlinear measurement function is used by $h(X) = |X|$. Here we use two measure functions, i.e. $h(X) = X$ and $h(X) = \log|X|$. The former is a linear function, whereas the latter is a nonlinear function with more nonlinearity than $h(X) = |X|$. We repeat the above assimilation experiments as shown in Figure 5, and calculate the average RMSE, as a function of the ensemble sizes, for each method. It was found that when the measurement function is linear, i.e. $h(X) = X$, both LPFs require a very large ensemble to beat the EAKF, i.e. $N = 320$ for P2016 LPF and $N = 160$ for the new LPF. However, while the measurement function is highly nonlinear, i.e. $h(X) = \log|X|$, both LPFs easily show the advantage over EAKF, even using a very small ensemble ($N = 10$). This is because the theory of EAKF has an inherent Gaussian assumption, such that it is less effective when the system is highly nonlinear and non-Gaussian (e.g. Tang *et al.*, 2014). Meanwhile, although the observational error is still assumed to be Gaussian, the Gaussian assumption of prior probability density function is removed in LPF. In addition, Figure 6 evidently shows that the new LPF outperforms the P2016 LPF in almost each case, further confirming the advantage of the new formulation of vector weights.

5. Conclusions

It is generally acknowledged that PFs can assimilate non-Gaussian information into nonlinear dynamical models. However, applying PFs in high-dimensional atmospheric or oceanic models remains an open problem. Very recently, localized PFs were developed to efficiently overcome the problem of filter degeneracy. The key function of the LPF is to extend scalar weights to vector weights.

In this paper, we analysed the potential problems of the current LPF algorithm. It was argued that the formulation of the vector weights contains an implicit assumption that the likelihoods of the observations or weights are continuous in state space. Thus, we considered a new formulation of vector weights for the LPF, which removes this assumption. This new formulation is based on continuity about the analysis points in state space, which has a clear statistical interpretation and achieves localization for the PF.

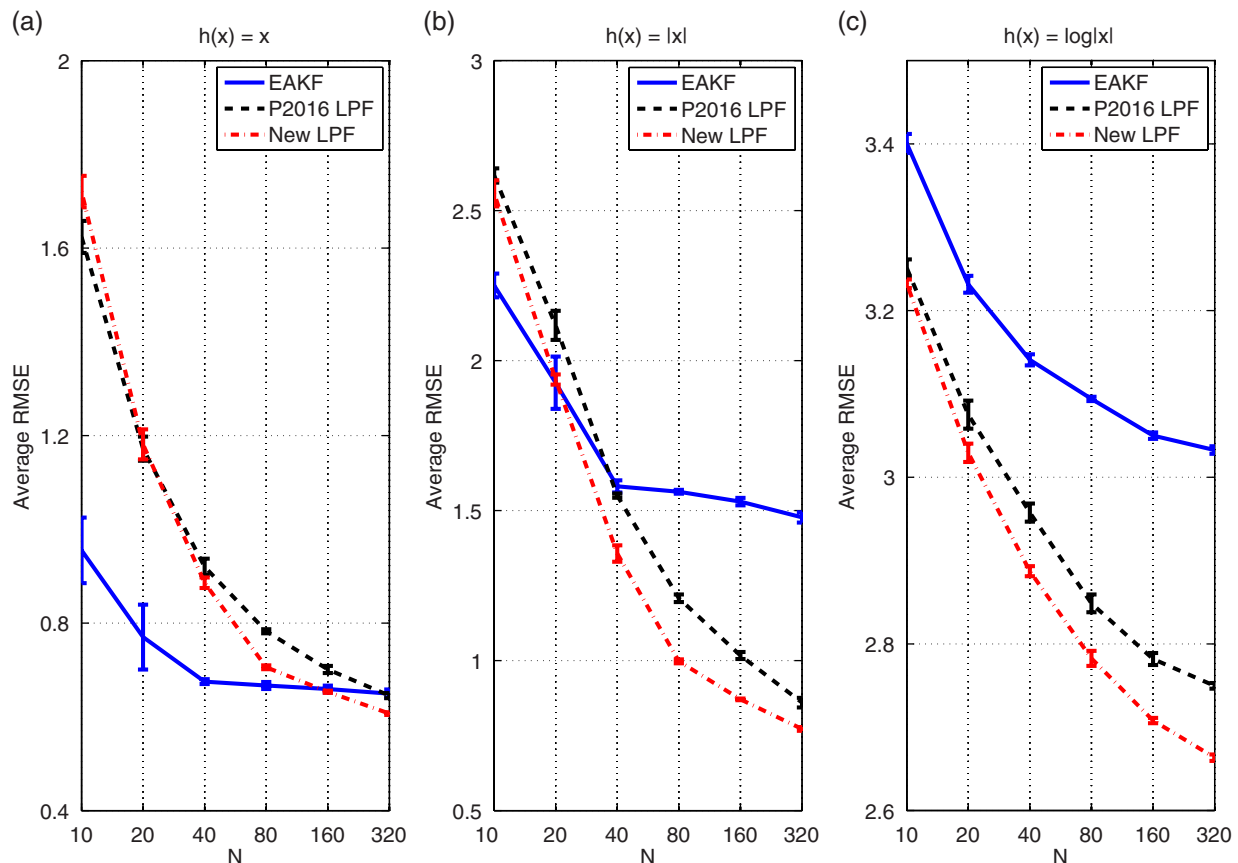


Figure 6. The average RMSE of the data assimilation results as a function of ensemble size N , for the EAKF, P2016 LPF and new LPF, using measurement functions (a) $h(X) = X$, (b) $h(X) = |X|$ and (c) $h(X) = \log|X|$. [Colour figure can be viewed at wileyonlinelibrary.com.]

Using the Lorenz '96 model, we conducted assimilation experiments using the EAKF, P2016 LPF and the newly proposed LPF. In these experiments, all settings, including sampling and merging strategy approaches, were kept the same. Thus, the differences in these experimental results are due to the differences between the algorithms. It was found that, when the system is highly nonlinear (non-Gaussian), either in its nonlinear forecast model or its nonlinear measurement function, the newly proposed LPF shows better results than those of the P2016 LPF, both of which beat those of the EAKF. This shows the large potential of the LPF algorithm for application in realistic atmospheric and oceanic models. However, this experiment uses only a simple Lorenz model and lacks sufficient complexity and higher dimensions. Further studies are needed, using more complicated models to develop an LPF assimilation system appropriate for coupled general circulation models; this work is under way. Nevertheless, this work improves the LPF algorithm, which is likely the most practically efficient and encouraging of all PF schemes, and supports its theoretical significance and practical importance.

Acknowledgements

This study benefited from constructive discussions with Dr J. Poterjoy, who also generously provided the MATLAB codes of his localized PF algorithm.

This work is supported by National Key R&D Program of China (2016YFC1401701), the National Program on Global Change and Air–Sea Interaction (GASI-IPOVAI-06), and grants from Basic Science Foundation of the Second Institute of Oceanography (JG1619), the National Natural Science Foundation of China (41621064, 41606012, 41690124).

References

Ades M, Van Leeuwen PJ. 2015. The equivalent-weights particle filter in a high-dimensional system. *Q. J. R. Meteorol. Soc.* **141**: 484–503, <https://doi.org/10.1002/qj.2370>.

- Anderson JL. 2003. A local least squares framework for ensemble filtering. *Mon. Weather Rev.* **131**: 634–642.
- Bengtsson T, Snyder C, Nychka D. 2003. Toward a nonlinear ensemble filter for high-dimensional systems. *J. Geophys. Res.* **108**: 8775, <https://doi.org/10.1029/2002JD002900>, D24.
- Chen Y, Reich S. 2015. *Assimilating Data Into Scientific Models: An Optimal Coupling Perspective*, Vol. 2. Springer-Verlag: New York, NY.
- Cheng Y, Tang Y, Jackson P, Chen D, Deng Z. 2010. Ensemble construction and verification of the probabilistic ENSO prediction in the LDEO5 model. *J. Clim.* **23**: 5476–5497.
- Chorin AJ, Morzfeld M, Tu X. 2010. Implicit particle filters for data assimilation. *Commun. Appl. Math. Comput. Sci.* **5**: 221–240.
- Frei M, Kunsch HR. 2013. Bridging the ensemble Kalman and particle filters. *Biometrika* **100**: 781–800.
- Gaspari G, Cohn S. 1999. Construction of correlation functions in two and three dimensions. *Q. J. R. Meteorol. Soc.* **125**: 723–757.
- Gordon NJ, Salmond DJ, Smith AF. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar Signal Process.* **140**: 107–113, <https://doi.org/10.1049/ip-f-2.1993.0015>.
- Hunt BR, Kostelich EJ, Szunyogh I. 2007. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D* **230**: 112–126.
- Kitagawa G. 1996. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *J. Comput. Graph. Stat.* **5**: 1–25.
- Lorenz E. 1996. 'Predictability: A problem partly solved'. In *Proceedings of Seminar on Predictability*, Vol. 1. ECMWF: Reading, UK.
- McGinnis S, Nychka D, Mearns LO. 2015. A new distribution mapping technique for bias correction of climate model output. In *Machine Learning and Data Mining Approaches to Climate Science*, Lakshmanan V, Gilleland E, McGovern A, Tingley M. (eds.). Springer: Berlin: 91–100.
- Nakano S. 2014. Hybrid algorithm of ensemble transform and importance sampling for assimilation of non-Gaussian observations. *Tellus A* **66**: 21429, <https://doi.org/10.3402/tellusa.v66.21429>.
- Nakano S, Ueno G, Higuchi T. 2007. Merging particle filter for sequential data assimilation. *Nonlinear Processes Geophys.* **14**: 395–408.
- Papadakis N, Mémén E, Cuzol A, Gengembre N. 2010. Data assimilation with the weighted ensemble Kalman filter. *Tellus A* **62**: 673–697.
- Penny S, Miyoshi T. 2016. A local particle filter for high-dimensional geophysical systems. *Nonlinear Proc. Geophys.* **23**: 391–405.
- Poterjoy J. 2016. A localized particle filter for high-dimensional nonlinear systems. *Mon. Weather Rev.* **144**: 59–76.

- Shen Z, Tang Y. 2015. A modified ensemble Kalman particle filter for non-Gaussian systems with nonlinear measurement functions. *J. Adv. Model. Earth Syst.* **7**: 50–66.
- Snyder C, Bengtsson T, Bickel P, Anderson J. 2008. Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.* **136**: 4629–4640.
- Tang Y, Ambadan J, Chen D. 2014. Nonlinear measurement function in the ensemble Kalman filter. *Adv. Atmos. Sci.* **31**: 551–558.
- Van Leeuwen PJ. 2003. Nonlinear ensemble data assimilation for the ocean, Seminar on recent developments in data assimilation for atmosphere and ocean, 8–12 September 2003, ECMWF, Reading, UK, 265–286.
- van Leeuwen PJ. 2009. Particle filtering in geophysical systems. *Mon. Weather Rev.* **137**: 4089–4114.
- Zhu M, Van Leeuwen PJ, Amezcua J. 2016. Implicit equal-weights particle filter. *Q. J. R. Meteorol. Soc.* **142**: 1904–1919, <https://doi.org/10.1002/qj.2784>.