A revised implicit equal-weights particle filter

Article

Accepted Version

A Revised Implicit Equal-Weights Particle Filter

Jacob Skauvold[1], Jo Eidsvik[1], Peter Jan van Leeuwen[2,3], and Javier Amezcua[2,3]

[1]Norwegian University of Science and Technology

[2]Data Assimilation Research Centre, University of Reading, UK

[3]National Centre for Earth Observation, UK

January 14, 2019

Abstract

Particle filters are fully non-linear data assimilation methods and as such are highly relevant. While the standard particle filter degenerates for high-dimensional systems, recent developments have opened the way for new particle filters that can be used in such systems.

The implicit equal-weights particle filter (IEWPF) is an efficient approach which avoids filter degeneracy because it gives equal particle weights by construction. The method uses implicit sampling whereby auxiliary vectors drawn from a proposal distribution undergo a transformation before they are added to each particle.

In the original formulation of the IEWPF, the proposal distribution has a gap causing all but one particle to have an inaccessible region in state space. We show that this leads to a systematic bias in the predictions and we modify the proposal distribution to eliminate the gap. We achieved this by using a two-stage proposal method, where a single

variance parameter is tuned to obtain adequate statistical coverage properties of the predictive distribution. We discuss properties of the implicit mapping from an auxiliary random vector to the state vector, keeping in mind the aim of avoiding particle resampling. The revised filter is tested on linear and weakly nonlinear dynamical models in low-dimensional and moderately high-dimensional settings, demonstrating the suiccess of the new methodology in removing the bias.

1 Introduction

Geophysical models involving numerical simulations of processes unfolding in space and time often take the form of state space models with non-linear dynamics and millions of state variables. As the evolution of such systems is sensitive to initial conditions and boundary conditions, which are almost never known precisely, the actual system state is generally uncertain. Model error, failure of the numerical model to faithfully represent the simulated process, also contributes to system state uncertainty. If observations of the modelled system are available, then incorporating information from these into the model through data assimilation can mitigate uncertainty and lead to more accurate predictions.

Data assimilation in a Bayesian setting begins with a prior probability distribution representing background knowledge about the unknown state variables. The relationships between states and observations are represented by conditional probability distributions referred to as the likelihood. Combining the prior distribution and likelihood according to Bayes' theorem yields a posterior distribution of the state conditional on the observations. When this is done over time, data assimilation conditions the dynamical model to data.

Variational data assimilation methods like 3D-Var and 4D-Var use optimi-

sation to locate the posterior mode (Asch et al., 2016; Van Leeuwen et al., 2015; Fletcher, 2017). While variational data assimilation methods do not necessarily characterise the spread of the posterior distribution, an estimate of the posterior covariance is available via the inverse of the Hessian evaluated at the mode. More direct uncertainty quantification is possible with ensemble-based data assimilation methods such as the many variants of the ensemble Kalman filter (EnKF), see e.g. Evensen (2009). However, the EnKF uses linear updating and implicitly assumes that the state distribution and likelihood are Gaussian. This limits the applicability of EnKF variants to only mildly non-linear dynamical models.

Particle filters (PFs), see e.g. Doucet et al. (2000), most of which are based on importance sampling, have no assumptions of linearity or Gaussianity. They work by propagating particles, or model realizations, forward in time via a forecast step and then weighting particles according to the likelihood, so that the resulting weighted ensemble of particles represents the posterior probability density. Some PF variants modify the forecast step by drawing particles from a proposal distribution instead of the forward model (e.g. Doucet et al., 2000; Van Leeuwen, 2009; Morzfeld et al., 2012; Van Leeuwen et al., 2015). This is then accounted for in the weighting step. PFs are appealing in large part because they are free of distributional assumptions and will, given enough particles, correctly sample the posterior distribution even when applied to highly non-linear dynamical models. In practice, when the number of particles is limited, PFs are subject to the curse of dimensionality and can be relied on for correct sampling only when state and observation dimensions are low to moderate. Applied to high-dimensional data assimilation tasks, PFs tend to suffer from filter degeneracy in the form of sample impoverishment. That is, the distribution of particle weights, which is initially uniform, quickly begins to concentrate

around, and eventually collapses onto, a small subset of particles, effectively reducing the ensemble size (Snyder et al., 2008).

There have been several approaches trying to combine strengths from PFs with EnKF approaches. Stordal et al. (2011) constructed a useful Gaussian mixture approximation to the predicted distribution at each step, bridging the EnKF update with a special kind of PF updates. Rezaie and Eidsvik (2012) shrinked the PF update towards the EnKF update, also relying on Gaussian mixture models, and tuned the shrinkage parameter to avoid degeneracy while maintaining reasonable statistical properties. Frei and Künsch (2013) applied a tuning parameter in the exponent of the likelihood part, where parts of the data (with larger variance) are used in an EnKF update, while the remaining part is used in a PF step. In principle, these approaches have the non-linear appeal of PFs, but automatized tuning tends to give results closer to the EnKF output for high dimensional systems and moderate particle sizes (Stordal et al., 2011).

Although in theory filter degeneracy issues can be remedied by increasing the number of particles, computational limitations restrict ensemble sizes to around 100 particles in many data assimilation applications (Van Leeuwen, 2009). What is desired in such cases is a PF variant that is resistant to filter degeneracy and maintains a resonable particle weight distribution when applied to nonlinear dynamical systems. Unlike the standard PF, such a filter might be a viable solution for nonlinear and high dimensional data assimilation despite having to operate with only a moderate number of particles.

The equivalent weights particle filter (EWPF, Van Leeuwen, 2010; Ades and Van Leeuwen, 2013) is a non-linear data assimilation approach which uses a proposal distribution constructed to give equal weights in the update step, thus avoiding particle degeneracy. Depending on the specifics of the proposal density

used, some or all particles may need to be resampled to maintain exact equality between weights.

The implicit equal-weights particle filter (IEWPF), introduced by Zhu et al. (2016), similarly prevents filter degeneracy by constructing the proposal distribution so that the weights are uniform. The IEWPF combines the implicit sampling framework of Chorin et al. (2013) with the equal-weights idea from Ades and Van Leeuwen (2013). By the implicit construction no parameter tuning is required. However, the approach tends to give biased results, particularly for moderate state dimensions, because its construction yields a proposal density for particle updates that is zero on parts of state space.

The new contributions of the current paper are first a demonstration that this bias is systematic, and leads to underestimation of the filter variance. Secondly, we modify the IEWPF to remedy some of the deficiencies of the proposal distribution under the original IEWPF formulation, specifically to eliminate the gap in state space described by Zhu et al. (2016) and to reduce the mismatch between the reported and actual prediction variance of the ensemble representation of the posterior probability density. Our suggested modification achieves this by introducing an additional perturbation of each particle in the update step of the filtering algorithm. Adjusting the scale of this perturbation enables calibration of ensemble spread without compromising particle weight equality. Additionally, the revised IEWPF can be applied to systems of any dimension. This is in contrast to the original IEWPF, which relied on an approximation that is only valid when the state dimension is large.

The new filter is a substantial improvement of the original IEWPF as it provides a way to mitigate the bias in the original method. Still, it should be noted that as with the original IEWPF, the emphasis is on handling non-Gaussianity resulting from a nonlinear dynamical model rather than a non-

Gaussian likelihood. Allowing a nonlinear observation operator does not pose a fundamental problem but part of the analytical development involving the incomplete γ functions would not be possible, and solutions to the nonlinear equations would rely more on iterative methods. Such an extension is outside the scope of the this work.

This paper is organized as follows: In Section 2 the original single-stage IEWPF algorithm is described. In Section 3 the new two-stage IEWPF is presented. In Section 4 a linear example and a non-linear Lorenz96 example are studied.

2 Implicit equal-weights particle filter (IEWPF)

In this section we describe the main ideas and building blocks of the IEWPF algorithm. Some properties and challenges of this algorithm are discussed. A modified version of the filter is then described in Section 3.

2.1 Problem description and background

Consider a dynamical system with an N_x-dimensional state vector $\mathbf{x}^n$, $n = 0, 1, \ldots, n_t$. Set initial distribution $\mathbf{x}^0 \sim N(\boldsymbol{\mu}, \mathbf{B})$, denoting an N_x-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{B}$. Given the state at time t_{n-1} the state at time t_n is given by

$$\mathbf{x}^n = \mathcal{M}(\mathbf{x}^{n-1}) + \mathbf{u}^n, \tag{1}$$

where $\mathcal{M}$ denotes forward integration of the dynamical system, and $\mathbf{u}^n \sim N(\mathbf{0}, \mathbf{Q})$ represents additive model error that we assume to be independent over time.

Suppose that at times $m \in \{1, 2, \ldots\}$ an observation vector $\mathbf{y}^m \in \mathbb{R}^{N_y}$ is

available. The relationship between the state and observation vectors is

$$\mathbf{y}^m = \mathbf{H}\mathbf{x}^m + \mathbf{v}^m. \tag{2}$$

Here, $\mathbf{H}$ is a size $N_y \times N_x$ linear observation operator and $\mathbf{v}^m \sim N(\mathbf{0}, \mathbf{R})$ represents additive observation error. In this article we will consider observation operators which simply select certain elements of the state vector, but operations like averaging and convolution of state vector elements are also possible. We assume that the error terms are independent over time, and independent of the error terms in the dynamical system model. Furthermore we will assume in the remainder of this article that observations are available at every time step n, so that the above notation may be simplified by letting $m = n$.

The filtering problem consists of estimating the current state $\mathbf{x}^n$ given all available observations up to time n. We denote the set of observations by $\mathbf{y}^{1:n}$. The filtering probability density function is $p(\mathbf{x}^n|\mathbf{y}^{1:n})$, and this is computed sequentially for $n = 1, 2, \ldots$. The PF (Gordon et al., 1993) represents the filtering distribution at every stage n by a size N_e ensemble of state realizations $\mathbf{x}_i^n$, $i = 1 \ldots, N_e$, called particles. Weights w_i^n, $i = 1, \ldots, N_e$, are assigned to each particle. A particle's weight is proportional to the likelihood of all data along its sample path. It is updated sequentially using the multiplicative factor $p(\mathbf{y}^n|\mathbf{x}_i^n)$.

One major problem affecting PF methods is sample degeneracy, also known as sample impoverishment. This happens when the distribution of weight over particles becomes more unequal with every iteration. Eventually this leads to a situation where almost all weight is concentrated on a single particle, so that the effective sample size is much smaller than the nominal ensemble size, and the usefulness of the resulting ensemble is very limited. To avoid this behaviour it is of interest to minimise the variance of the weights with respect to the

filtering distribution. The weights could be reset by including various kinds of resampling of particles at different stages, but this is usually not enough to avoid degeneracy in high-dimensional state space models.

Some PF variants employ importance sampling (Van Leeuwen, 2009), whereby particle updates are drawn from a proposal probability density function, or importance function, $q(\mathbf{x})$. Proposal densities are typically chosen to allow easy sampling and pointwise evaluation. The choice of proposal distribution can also affect the overall efficiency of the algorithm. For consistent results the particle weights are multiplied by the ratio of the target density to the proposal density

$$\frac{p(\mathbf{x}^n|\mathbf{y}^{1:n})}{q(\mathbf{x}^n)}. \tag{3}$$

According to Doucet et al. (2000), among potential importance functions of the form $q(\mathbf{x}^n) = q(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n)$, the one which minimises the particle weight variance is the conditional distribution $p(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n)$, referred to as the optimal proposal density (OPD, Snyder et al., 2015). Ades and Van Leeuwen (2013) showed that a PF using the optimal proposal density as an importance function will degenerate slower than the standard SIR PF, but the exponential dependence on the size of the system remains the same.

With our modeling assumptions, this OPD is Gaussian. At stage n, and for every particle $i = 1 \ldots, N_e$, its mean and covariance matrix, denoted by $\mathbf{x}_i^{n,\mathrm{a}}$ and $\mathbf{P}$ respectively, are given by

$$\mathbf{x}_i^{n,\mathrm{a}} = \mathcal{M}(\mathbf{x}_i^{n-1}) + \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y}^n - \mathbf{H}\mathcal{M}(\mathbf{x}_i^{n-1})), \tag{4}$$

and

$$\mathbf{P} = (\mathbf{Q}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}. \tag{5}$$

2.2 Single-stage IEWPF

The proposal density can be constructed in various ways. We will now discuss the implicit scheme used in the IEWPF. This implicit sampling is realised by centering the proposal distribution on the mode of the OPD for each particle, and adding a random perturbation vector which is pre-multiplied by the square root of the OPD covariance matrix (5) and by a particle-specific scale factor $\alpha_i^{1/2}$.

Mathematically, the updated state of particle i is computed according to

$$\mathbf{x}_i^n = \mathbf{x}_i^{n,a} + \alpha_i^{1/2} \mathbf{P}^{1/2} \boldsymbol{\xi}_i^n, \tag{6}$$

where the random vector $\boldsymbol{\xi}_i^n \in \mathbb{R}^{N_x}$ is drawn from the proposal distribution $q(\boldsymbol{\xi}_i^n)$, which is specified as $N(\mathbf{0}, \mathbf{I}_{N_x})$. With $\alpha_i = 1$ this scheme is equivalent to drawing samples from the OPD. When $\alpha_i \neq 1$, the corresponding sampling distribution is either compressed or extended relative to the OPD. Note that α_i will change over time steps, but for notational convenience we have suppressed the superscript n.

By selecting α_i judiciously one can gain flexibility in the algorithm and avoid particle degeneracy, for instance by aiming for equal weights like we do here. The weight of particle i is given by

$$\begin{aligned} w_i^n &= w_i^{n-1} \frac{p(\mathbf{x}^n|\mathbf{x}_i^{n-1}) p(\mathbf{y}^n|\mathbf{x}^n)}{q(\boldsymbol{\xi}_i^n)} \left\| \frac{\partial \mathbf{x}^n}{\partial \boldsymbol{\xi}_i^n} \right\| \\ &= \frac{1}{N_e} \frac{p(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n) p(\mathbf{y}^n|\mathbf{x}_i^{n-1})}{q(\boldsymbol{\xi}_i^n)} \left\| \frac{\partial \mathbf{x}^n}{\partial \boldsymbol{\xi}_i^n} \right\|, \end{aligned} \tag{7}$$

where it is assumed that $w_i^{n-1} = 1/N_e$ for all particles i. To have equal weights $w_1^n = w_2^n = \ldots = w_{N_e}^n = w_{\text{target}}^n$ the unnormalized log-weights must also be

equal, hence for each particle i the scalar α_i must satisfy

$$(\alpha_i - 1)\boldsymbol{\xi}_i^T \boldsymbol{\xi}_i - 2\log(\alpha_i^{N_x/2}) - 2\log\left(\left|1 + \frac{\partial \alpha_i^{1/2}}{\partial \boldsymbol{\xi}_i}\frac{\boldsymbol{\xi}_i}{\alpha_i^{1/2}}\right|\right) = C - \varphi_i, \quad (8)$$

for a constant C and with

$$\varphi_i = [\mathbf{y}^n - \mathbf{H}\mathcal{M}(\mathbf{x}_i^{n-1})]^T(\mathbf{R} + \mathbf{HQH}^T)^{-1}[\mathbf{y}^n - \mathbf{H}\mathcal{M}(\mathbf{x}_i^{n-1})], \quad (9)$$

so that

$$p(\mathbf{y}^n|\mathbf{x}_i^{n-1}) \propto e^{-\varphi_i/2}. \quad (10)$$

In practice the scale factor α_i is determined numerically by solving

$$\gamma\left(\frac{N_x}{2}, \frac{\alpha_i \boldsymbol{\xi}_i^{n,T}\boldsymbol{\xi}_i^n}{2}\right) = e^{-c_i/2}\gamma\left(\frac{N_x}{2}, \frac{\boldsymbol{\xi}_i^{n,T}\boldsymbol{\xi}_i^n}{2}\right), \quad (11)$$

for α_i, where $\gamma(s,x) = \int_0^x t^{s-1}e^{-t}\mathrm{d}t$ is the lower incomplete gamma function. and we refer to $c_i = \max_j[\varphi_j] - \varphi_i$ as the ith offset. See Appendix for details.

By using the solution of (11) in the update expression (6) one ensures that the unnormalized weight associated with the ith updated state vector $\mathbf{x}_i^n$ is equal to the chosen target weight. The log-weight offsets $c_1, c_2, \ldots, c_{N_e}$ are necessary because the likelihood $p(\mathbf{y}^n|\mathbf{x}_i^{n-1}) \propto \exp(-\varphi_i/2)$ of the current observation given the previous state of the ith particle will differ between particles. For every particle i to reach the target weight we need $c_i \geq 0$. That is, the target unnormalized weight cannot be set larger than the smallest unnormalized weight in the ensemble. Consequently, since the incomplete γ-function is monotonically increasing, we must have $\alpha_i \leq 1$ for every particle i. We therefore expect an updated IEWPF ensemble to have a smaller spread than a sample drawn from the OPD, and this suggests an explanation for the bias in the original IEWPF. The offset c_i in equation (11) is chosen by targeting the smallest unnormalized

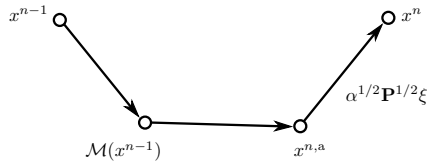

Figure 1: Single-stage IEWPF proposal scheme.

weight in the forecast ensemble. In principle, c_i could be defined differently, targeting for instance the average or median weight. Targeting the smallest weight, i.e. the maximum φ_i, has the advantage of making all offsets non-negative, which guarantees that a solution of (11) exists.

In their original formulation of the IEWPF, Zhu et al. (2016) considered the limiting case of (11) when $N_x \to \infty$. This yields a simplified equation for α_i which admits an analytical solution in terms of the Lambert W function (Weisstein, 2002). A feature of this closed-form solution is a gap between branches of the Lambert W function, leaving a region in state space where the proposal density of the filter is zero. The authors used both branches of the solution, one corresponding to $\alpha_i \leq 1$ and one to $\alpha_i \geq 1$, to reduce the bias of the resulting filter. We see here that using both solutions is in fact inconsistent, and only the $\alpha_i \leq 1$ solutions are valid. In this article we do not simplify or approximate equation (11). Instead we resort to numerical solution methods for determining α_i. Although the solutions of (11) obtained in this way do not have a gap between distinct branches, the resulting transformation from $\boldsymbol{\xi}_i^n$ to $\mathbf{x}_i^n$ is not in general bijective (see Section 2.3).

An elementary sketch of the particle movement of the single-stage IEWPF is summarized in Figure 1. Details of the IEWPF implementation are provided in the Appendix.

2.3 Properties of the single-stage IEWPF

The implicit formulation of the IEWPF makes it difficult to study the properties of the resulting particle representation. For instance it is not clear, even in simplified model settings, how to calculate closed form expressions describing how the IEWPF update changes the ensemble mean or variance. In what follows we will nevertheless gain insight in the solutions via the form of the implicit transformation, and by simulating from a Gaussian model where the exact solution is known. In Section 3 we then modify the algorithm and overcome some of the shortcomings of the single-stage IEWPF.

Figure 2 shows solutions of the equal weights equation (11) for seven different offsets c (ignoring the subscript i in this display). When implementing the IEWPF, we require $c > 0$, but here we consider the more general case $c \in \mathbb{R}$. The solutions in Figure 2 are shown in terms of the transformation from $\boldsymbol{\xi}$ to $\alpha^{1/2}\boldsymbol{\xi}$. When $c = 0$ the solution is $\alpha = 1$ which gives the identity transformation. Furthermore the solution α decreases with increasing c, so for $c < 0$ the resulting transformation has the effect of expanding the probability distribution of the perturbation $\boldsymbol{\xi}$, whereas for $c > 0$ the transformation contracts the distribution.

As can be seen in Figure 2 the contracting solutions for $c > 0$ have horizontal asymptotes while the expanding solutions for $c < 0$ have vertical asymptotes. As a consequence, the transformation from $\boldsymbol{\xi}$ to $\alpha^{1/2}\boldsymbol{\xi}$, and hence to $\mathbf{x}^n$, is not defined on the whole domain when $c > 0$, and is not surjective when $c < 0$. Only when $c = 0$ is the transformation bijective. When solving (11) with negative offsets therefore, we are not free to use any proposal distribution for $\boldsymbol{\xi}$ as the range of possible perturbation vectors must be restricted to the appropriate subset of the domain. One could try to achieve this by truncating the proposal distribution at the location of the vertical asymptote when $c < 0$. But this is not a viable modification of the IEWPF, because truncation introduces a

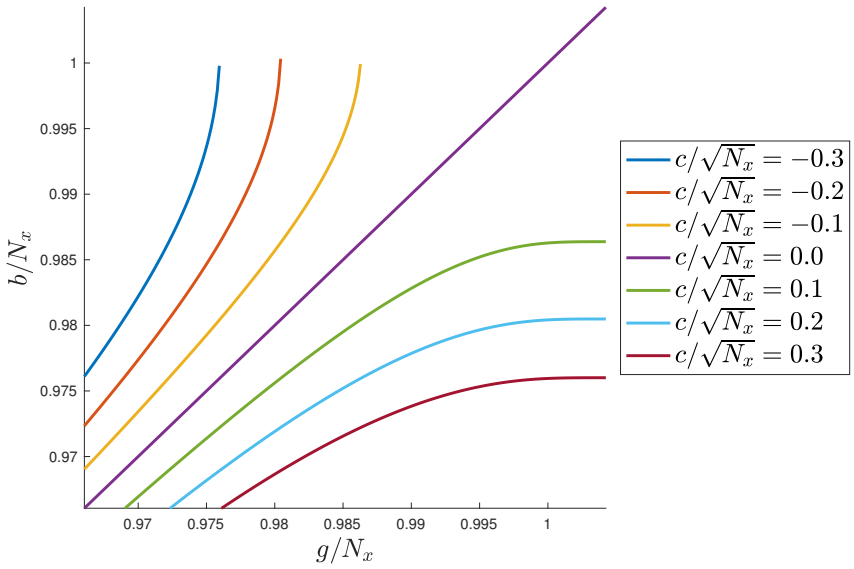

Figure 2: Solutions of (11) for different offsets c shown in terms of $g = \|\boldsymbol{\xi}\|$ and $b = \|\alpha^{1/2}\boldsymbol{\xi}\|$.

particle-dependent normalisation constant into the expression for the particle weights, making them unequal. Here, the requirement that weights be kept equal appears to be in conflict with the requirement that the transformation from $\boldsymbol{\xi}$ to $\mathbf{x}^n$ should be a bijection from $\mathbb{R}^{N_x}$ to $\mathbb{R}^{N_x}$ (Chorin et al., 2010). A theoretical justification of the IEWPF ultimately necessitates the resolution of this conflict, but it is unclear whether it can be resolved.

Considering the OPD and the update expression (6), it is clear that when $\alpha_i < 1$, the IEWPF produces updated particles with a smaller variance than the OPD PF, which is known to be unbiased. Hence we expect underestimation of variance as a consequence of using only contracting solutions of the equal-weights equation. As is illustrated in the following simulation study, the IEWPF does indeed tend to underestimate the variability of the state vector in the long run.

To make the presentation of the IEWPF more concrete before introducing the revised version, we now apply the IEWPF to a test case involving a Gauss-linear model. We revisit the same test case in Section 3.2 after describing the revised IEWPF. A more detailed description of the test case is given in Section 4.1.

We consider a size 100 state vector with initial state $\mathbf{x}^0 \sim N(0, \mathbf{B})$. The transition mechanism is defined by $\mathbf{x}^n \sim N(\mathbf{x}^{n-1}, \mathbf{Q})$, $n = 1, \ldots$. Further, observations are given by $\mathbf{y}^n \sim N(\mathbf{x}^n, \mathbf{R})$. The filtering distribution is then Gaussian and its mean and covariance matrix are provided by the Kalman filter (Kalman, 1960). The covariance matrices $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{R}$ are all diagonal, with constant diagonal entries of 1.0, 0.04 and 0.12 respectively.

When applying the IEWPF as defined by equation (6) and (8) to this model we find that while the ensemble mean matches the KF mean on average, the ensemble spread is too small to match the KF variance in the long run. This means that the IEWPF systematically underestimates the variance of the state (see Figures 3 and 4).

In Figure 3 the results of 1000 independent simulations are visualised for one state variable (component 42) at time $n = 120$. We show the rank histogram of the true variable in the set of $N_e = 25$ particles. This is computed by sorting the particles from smallest to largest by the value of this component, and then determining the position of the true value in this ordering. The procedure is repeated for each simulation. When the true state is unavailable the preferred approach is to carry out ranking in data space, comparing observations with realizations of their model equivalents generated from the ensemble. For a detailed treatment of rank histograms, their use and cautions, see Hamill (2001).

The rank of the true state relative to the ensemble should ideally be uniform, but in Figure 3 we notice few ranks in the middle. The true value is too often

at the extremes of the distribution represented by the 25 particle members. This means that the ensemble is underdispersive, i.e. the variability in the particle set is too small. Figure 4 shows the distribution over 1000 simulations of variance at time $n = 20$, averaged over all 100 entries of the state vector. A corresponding variance distribution for the stochastic EnKF is included for comparison. None of the filters being compared use inflation or localization. The purpose of the comparison is not to show which filter performs better, but rather to demonstrate that the IEWPF systematically underestimates the filtering variance. The variance in the particle representation varies somewhat between the different state vector entries but is mostly between 0.02 and 0.04. In comparison, the variance calculated by the Kalman filter is 0.052 for all state vector entries. Both the IEWPF and the EnKF underestimate the long-run process variability for this example. For the EnKF, variance estimates become more consistent with the KF level when the ensemble size is increased (Figure 4, bottom display). The same is not true of the IEWPF.

In Section 4 we provide further analysis of this example, studying how the filter behaves over time. We also compare results of the single-stage IEWPF with our new algorithm using two stages.

3 Modifying the IEWPF

To address the underestimation of variance by the IEWPF described in the previous section, we now introduce a modified version of the filter. We add a second perturbation vector $\boldsymbol{\eta}_i$, orthogonal to $\boldsymbol{\xi}_i$, to the analysis state $\mathbf{x}_i^{n,a}$. We refer to the filter with two separate perturbation vectors as the two-stage IEWPF, while the original filter with one perturbation vector is referred to as the single-stage IEWPF. Like the single-stage filter, the two-stage case also involves a particle-specific parameter α_i which ensures equal particle weights.

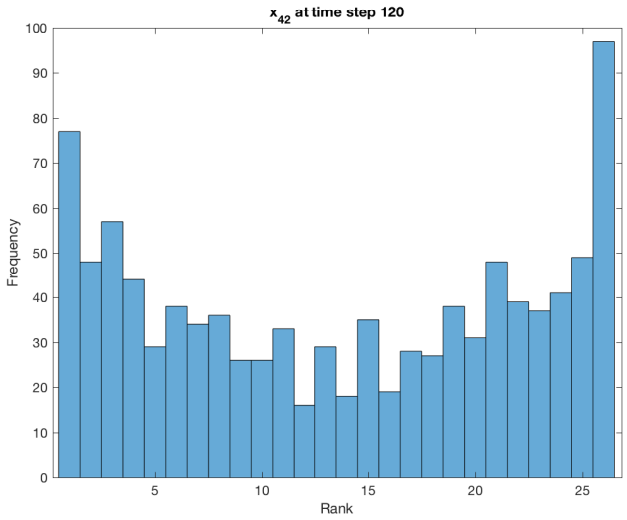

Figure 3: Rank histogram of x_{42}^{120} of true realisation relative to IEWPF ensemble over 1000 simulations. U-shape suggests ensemble is under-dispersed.

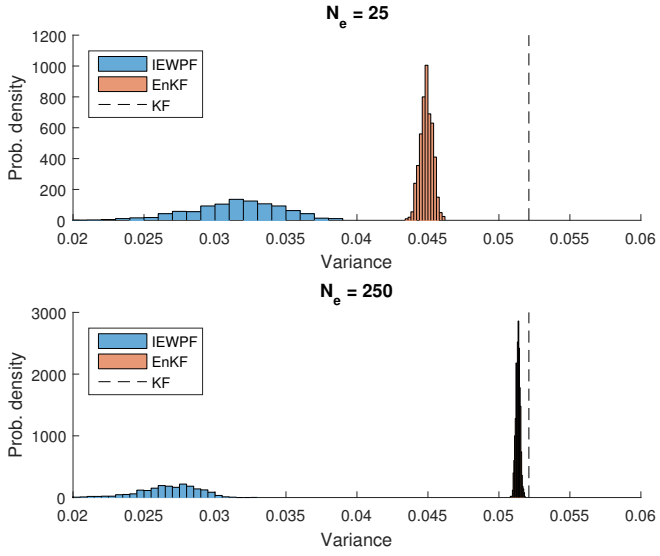

Figure 4: Histograms of estimates of $\mathrm{Var}(\mathbf{x}^{20})$ from 1000 runs of the IEWPF and stochastic EnKF, averaged over $N_x = 100$ elements compared with deterministic KF result. **Top**: $N_e = 25$. **Bottom**: $N_e = 250$.

In the two-stage filter there is an additional parameter β, which is common to all particles and is related to the spread of the ensemble. Note that in the single-stage case, α_i depends on the unnormalised weight of the ith particle in the forecast ensemble as well as the magnitude of the sampled perturbation vector $\boldsymbol{\xi}_i$, i.e. $\alpha_{i,\text{1-stage}} = \alpha_i(\varphi_i, \|\boldsymbol{\xi}_i\|)$. In the two-stage case, α_i will also depend on β and the magnitude of $\boldsymbol{\eta}_i$, i.e. $\alpha_{i,\text{2-stage}} = \alpha_i(\varphi_i, \|\boldsymbol{\xi}_i\|, \beta, \|\boldsymbol{\eta}_i\|)$.

3.1 Two-stage IEWPF

In the two-stage proposal scheme, the updated particle $\mathbf{x}_i^n$ is given by

$$\mathbf{x}_i^n = \mathbf{x}_i^{n,\text{a}} + \beta^{1/2}\mathbf{P}^{1/2}\boldsymbol{\eta}_i + \alpha_i^{1/2}\mathbf{P}^{1/2}\boldsymbol{\xi}_i, \tag{12}$$

where the perturbation vectors $\boldsymbol{\xi}_i, \boldsymbol{\eta}_i \in \mathbb{R}^{N_x}$ are standard multivariate Gaussian random vectors satisfying $\boldsymbol{\xi}_i^T\boldsymbol{\eta}_i = 0$. Requiring orthogonality simplifies the particle weight expression so that the equal-weights equation for α_i has the same form as in the single-stage case. Using perturbation vectors that are not orthogonal would introduce extra terms in the equal-weights equation (see Appendix). The equal-weights equation for the updating scheme (12) is

$$\begin{aligned}&(\alpha_i - 1)\boldsymbol{\xi}_i^T\boldsymbol{\xi}_i - 2\log(\alpha_i^{N_x/2}) - 2\log\left(\left|1 + \frac{\partial\alpha_i^{1/2}}{\partial\boldsymbol{\xi}_i}\frac{\boldsymbol{\xi}_i}{\alpha_i^{1/2}}\right|\right)\\ &= C - \varphi_i - (\beta - 1)\boldsymbol{\eta}_i^T\boldsymbol{\eta}_i.\end{aligned} \tag{13}$$

Note that (13) is identical to the single-stage equal-weights equation (8) with the offset now defined as $c_i = \max_j [D_j] - D_i$ where $D_j = \varphi_j - (1 - \beta)\boldsymbol{\eta}_j^T\boldsymbol{\eta}_j$.

The purpose of the additional perturbation $\boldsymbol{\eta}_i$ and the common scale factor β is to control the spread of the updated particles so that the filter correctly represents the variability of the filtered state. In applications β would be considered a tuning parameter.

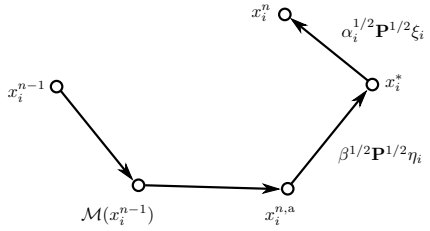

Figure 5: Two-stage IEWPF proposal scheme. Compare with Figure 1.

To determine a suitable value of β, particle ranks or coverage probabilities may be used. Since the spread of the updated ensemble is sensitive to the value of β, different values will produce differently shaped rank distributions and different observed coverage probabilities. This can indicate whether the currently used value of β is suitable and, if it is not, whether the value should be adjusted up or down. An automated search procedure based on some quantitative mismatch criterion—say, the difference between observed and nominal coverage probabilities—is also possible.

A coverage probability is the observed frequency with which a prediction interval covers the predicted quantity. Ideally it should match the interval's nominal confidence level. For instance, an 80% prediction interval for y^n is $(y^n_{(0.1N_e)}, y^n_{(0.9N_e)})$, and on average about 80% of the data vector entries at time n should fall within this interval. We suggest tuning β such that the coverage probabilities at the 50%, 60%, ..., 90% levels all match their respective nominal confidence level reasonably well. This entails running the algorithm for a range of β values, and choosing a value that gives an acceptable calibration (see Section 4 for more details about how this is tuned in practical experiments).

An elementary sketch of the particle movement of the two-stage IEWPF is summarized in Figure 5. Implementation details are provided in the Appendix.

3.2 Properties of the two-stage IEWPF

As for the single-stage IEWPF, it is difficult to study analytical properties of the two-stage IEWPF, even in simplified model settings. Some insight can still be gleaned by simulating from a Gaussian model where the exact solution is known.

In section 2.3 we stated that the single-stage transformation from $\boldsymbol{\xi}$ to $\mathbf{x}$ implied by (6) is only injective for $c \geq 0$ and is only surjective for $c \leq 0$, i.e. it is a bijection only when $c = 0$. In the two-stage case we can think of the map from $\boldsymbol{\xi}$ to $\mathbf{x}$ as depending on $\boldsymbol{\eta}$ and β through c. That is, there is not one map $\psi : \boldsymbol{\xi} \mapsto \mathbf{x}$, but a set $\{\psi_c : c \geq 0\}$ of maps where c is a function of β and $\boldsymbol{\eta}$. The two-stage IEWPF keeps β fixed and draws a random $\boldsymbol{\eta}$, thereby selecting one of the maps ψ_c. Then $\boldsymbol{\xi}$ is drawn subject to the orthogonality constraint. For any point $\mathbf{x} \in \mathbb{R}^{N_x}$ and any $c \geq 0$, there is some combination of $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ with $\boldsymbol{\eta}^T\boldsymbol{\xi} = 0$ such that ψ_c maps $\boldsymbol{\xi}$ onto $\mathbf{x}$. With $\boldsymbol{\eta}$ fixed, there may not exist a $\boldsymbol{\xi}$ that is orthogonal to $\boldsymbol{\eta}$ and is mapped onto $\mathbf{x}$. Introducing a second perturbation vector to randomize the selection of a map is thus a way to ensure that state space is covered by the proposal distribution.

Since the proposal distribution of the additional perturbation vector is zero-mean, the expectation of the state vector is the same under the two-stage update scheme as under the single-stage scheme. Hence, the modification does not induce a bias in the ensemble mean.

We return now to the Gauss-linear model from Section 2.3. A more detailed description of the test case is given Section 4.1. This time we apply the two-stage IEWPF to the Gauss-linear test case. Results of this method and that of the Kalman filter are shown in Figures 6 and 7. As in Section 2.3, these are the results of 1000 independent simulations, and the results are presented for time $n = 120$. Also as in Section 2.3, the ensemble size is $N_e = 25$.

Figure 6 shows the rank histograms for the true value of state vector entry 42. The rank histogram for $\beta = 0.05$ is clearly U-shaped. As β increases to 0.25 and 0.30 the rank distribution becomes more uniform. The rank histogram for $\beta = 0.5$ is indistinguishable from a uniform distribution given the sampling error and the calibration is better than when using the single-stage approach as shown in Figure 3.

Figure 7 shows the distributions of average variances produced by the two-stage IEWPF for β set to 0.05, 0.25, 0.30 and 0.50. The average is taken over all elements of the state vector. The display also shows the Kalman filter variance estimate as a thin, vertical line. Ideally the IEWPF should produce an ensemble whose variance matches the KF variance. Of the four β-values considered, 0.3 and 0.5 come closest to realizing this, showing a clear improvement over the variance distribution of the single-stage IEWPF in Figure 4. Judging by Figure 7, the optimal value of β in terms of variance calibration seems to lie closer to 0.3 than to 0.5 in this case. Yet Figure 6 shows a more uniform rank distribution for $\beta = 0.5$ than for $\beta = 0.3$. It is important to keep in mind, however, that comparing the rank histograms in terms of their degree of departure from uniformity is less precise than comparing the more concentrated variance histograms in terms of their locations along the horizontal axis. Figure 7 is therefore probably a better guide to identifying the optimal value of β. On the other hand, it cannot be ruled out that the discrepancy between the two figures has a different cause, such as the updated particles having a non-Gaussian distribution.

4 Numerical experiments

We present two synthetic test cases for assessing the performance of the IEWPF algorithms described in sections 2 and 3. The first is a Gauss-linear test case

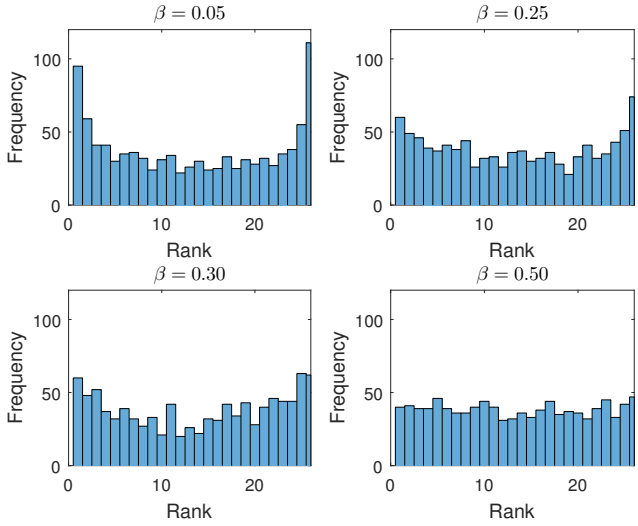

Figure 6: Rank histogram of x_{42}^{120} of true realisation relative to IEWPF ensemble over 1000 simulations. Results are for the two-stage IEWPF using four different values of β.

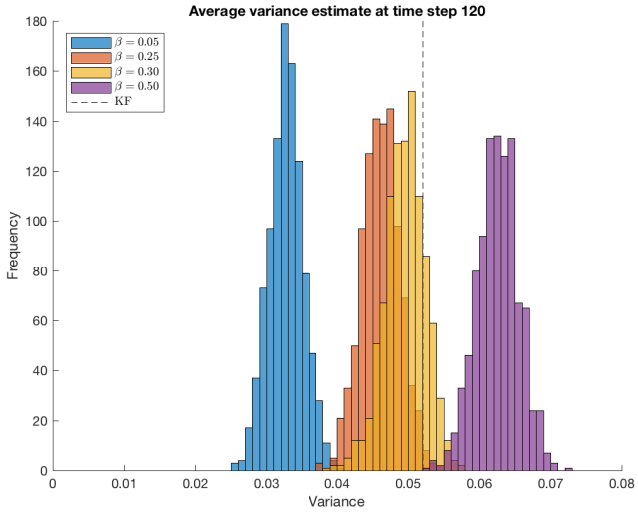

Figure 7: Histograms of two-stage IEWPF estimates of the variance of $\mathbf{x}^{120}$, averaged over all elements, for four different values of β, based on 1000 simulations each.

where the dynamical system evolves according to a linear model and the system state is observed directly except for an additive observation error term. In the second case the state evolves according to the Lorenz96 model (Lorenz, 1995) and we observe every second element of the state vector. We assume Gaussian probability distributions for the initial state, model errors and observation errors as described in section 2.1. In the Gauss-linear case the filtering distribution is analytically available via the Kalman filter under these assumptions, and we will make use of this to judge the quality of the estimates produced by the single-stage and two-stage IEWPFs.

4.1 Gauss-linear model

This is the Gauss-linear test case referred to in Sections 2.3 and 3.2. We re-use the model and observation equations from Section 3.1 of Zhu et al. (2016):

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \mathbf{u}^n, \tag{14}$$

$$\mathbf{y}^n = \mathbf{x}^n + \mathbf{v}^n, \tag{15}$$

$$\mathbf{u}^n \sim N(0, \mathbf{Q}), \quad \mathbf{v}^n \sim N(0, \mathbf{R}), \quad \mathbf{x}^0 \sim N(0, \mathbf{B}),$$

$$N_x = 100, \qquad n_t = 120, \qquad \mathbf{Q} = 0.04\mathbf{I}, \qquad \mathbf{R} = 0.12\mathbf{I}, \qquad \mathbf{B} = \mathbf{I}.$$

The filtering probability density $p(\mathbf{x}^n|\mathbf{y}^1, \ldots, \mathbf{y}^n)$ is Gaussian with parame-

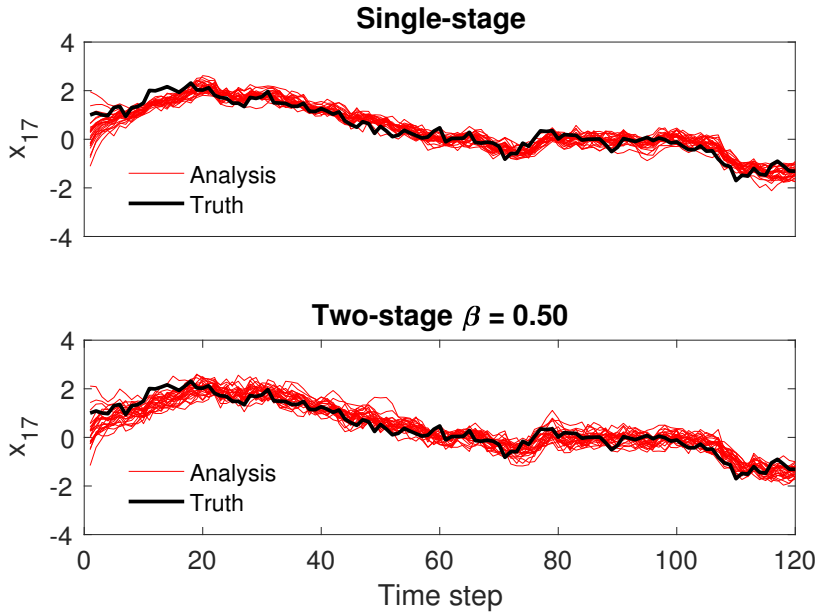

Figure 8: Example trajectories of x_{17} under Gauss-linear model. **Top**: Single stage IEWPF. **Bottom**: Two-stage IEWPF with $\beta = 0.5$. In both panels analysis ensemble members are shown in red and the true model trajectory in black.

ters $\boldsymbol{\mu}^n$ and $\mathbf{P}^n$, given recursively via the Kalman filter:

$$\begin{aligned}\boldsymbol{\mu}^n &= \boldsymbol{\mu}^{n-1} + (\mathbf{P}^{n-1} + \mathbf{Q})(\mathbf{R} + \mathbf{P}^{n-1} + \mathbf{Q})^{-1}(\mathbf{y}^n - \boldsymbol{\mu}^{n-1}) \\ \mathbf{P}^n &= \mathbf{P}^{n-1} + \mathbf{Q} - (\mathbf{P}^{n-1} + \mathbf{Q})(\mathbf{R} + \mathbf{P}^{n-1} + \mathbf{Q})^{-1}(\mathbf{P}^{n-1} + \mathbf{Q}),\end{aligned} \tag{16}$$

where $\boldsymbol{\mu}^0 = \mathbf{0}$ and $\mathbf{P}^0 = \mathbf{B}$.

We compare results of the single-stage and two-stage IEWPF, using the KF filtering distribution (16) as a reference solution. The number of particles is $N_e = 25$, and we run the algorithm for 1000 simulations.

Example trajectories of the single-stage and two-stage IEWPF algorithms are shown in Figure 8. Both follow the true state pretty well, but the single-stage results (top display) have less variability.

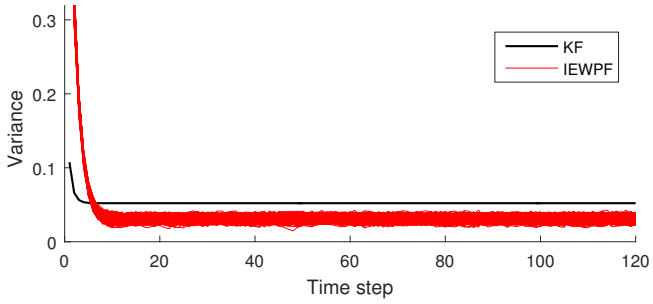

Figure 9: Comparison of posterior variance estimates from the Kalman filter and the single-stage IEWPF for state vector entry 42 in the Gaussian linear model case with $N_x = 100$ and $N_e = 25$. The IEWPF variance curves show the result of 1000 repetitions of the filtering task.

Figure 9 shows variance results for the single-stage IEWPF over the assimilation steps. Initial filtering variance, specified through the background error covariance matrix $\mathbf{B}$, is 1. The variance of the KF filtering distribution decreases quickly before stabilising, while the IEWPF ensemble variance takes longer to stabilise, and does so at a lower variance level. Comparing the IEWPF and KF variance estimates, it is clear that the IEWPF overestimates the filtering variance early on, and underestimates it in the long run.

Figure 10 shows average two-stage IEWPF variance estimates over the assimilation steps. As in the single-stage case (Figure 9), variability is still overestimated at the beginning of the time interval, but the long-run KF variance can now be matched quite well by an appropriate choice of β.

4.2 Lorenz96 model

We study the performance of the single and two-stage IEWPF using the model presented in Section 3.2 of Zhu et al. (2016). The dynamical model is given by

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_{i-2}x_{i-1} + x_{i-1}x_{i+1} - x_i + F, \qquad i = 1, \ldots, N_x, \tag{17}$$

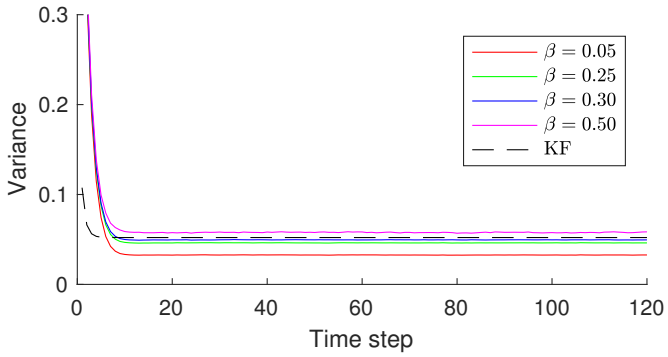

Figure 10: Evolution of variance estimates from the two-stage IEWPF with four different values of the common scale factor β. The solid curves show variance estimates averaged over 1000 independent simulations. The Kalman filter variance estimate is included for comparison.

where the indices wrap around so that x_{N_x+1} is identified with x_1. Letting $\mathbf{x}^n = \mathbf{x}(t_n)$ with $t_n = n\Delta t$, the model equation can be written as

$$\mathbf{x}^n = \mathcal{M}(x^{n-1}) + \mathbf{u}^n, \qquad \mathbf{u}^n \sim N(0, \mathbf{Q}), \qquad n = 1, \ldots, n_t,$$

where $\mathcal{M}$ denotes integration of equation (17) by a fourth order Runge-Kutta scheme and

$$N_x = 40, \qquad n_t = 300, \qquad F = 8, \qquad \Delta t = 0.05.$$

Observations are gathered at every time step t_n, $n = 1, \ldots, n_t$, which means that here Δt is both the integration time step of the numerical solution of (17) and the time between successive observation time points. Data are related to the state vector by

$$\mathbf{y}^n = \mathbf{H}\mathbf{x}^n + \mathbf{v}^n, \qquad \mathbf{v}^n \sim N(0, \mathbf{R}),$$

where $\mathbf{H}$ is a selection matrix which picks out every second element of the state vector, so that $\mathbf{H}\mathbf{x}^n = (x_2^n, x_4^n, \ldots, x_{N_x}^n)^T$. The remaining model parameters are specified as follows:

$$\mathbf{B} = \text{tridiag}(0.25, 1, 0.25), \quad \mathbf{Q} = \text{tridiag}(0.025, 0.10, 0.025), \quad \mathbf{R} = 0.16I,$$

where $\text{tridiag}(a_1, a_2, a_3)$ is a tridiagonal matrix with a_1 in every entry of the first subdiagonal, a_2 on the main diagonal and a_3 on the first superdiagonal. The nonlinearity in this data assimilation test case is weak due to the high frequency of observations. Increasing the time between updates would give a more severe test of the filter. A weakly nonlinear test case is still suitable for demonstrating that the IEWPF ensemble spread can be controlled through the choice of β.

We run the single-stage and two-stage IEWPF variants on this test case, with $N_e = 100$ in both cases. Figure 11 shows results of the two-stage IEWPF using $\beta = 0.7$, where we plot the filtering distribution over time along with the truth. This is done for two entries of the state vector (component 1 and 2). The ensemble tracks the reference state and covers the truth reasonably well. The bottom display shows the estimated variances of component 1 and 2. Because the observations provide much more information about the second entry, this has smaller variance over time.

In Figure 12 we plot coverage probabilities at one time step. These are plotted for different β parameters and for different confidence levels. The tuning procedure tells us that a value of β near 1 is useful in this example because it gives the best predictive performance, and any value in the range $0.7 - 1.2$ would be acceptable. Figure 13 compares rank histograms of one run each of the single-stage IEWPF and the two-stage IEWPF with $\beta = 0.7$, the latter being the same run used to make Figure 11. The single-stage rank histogram has a clear U-shape while the two-stage rank histogram does not, suggesting

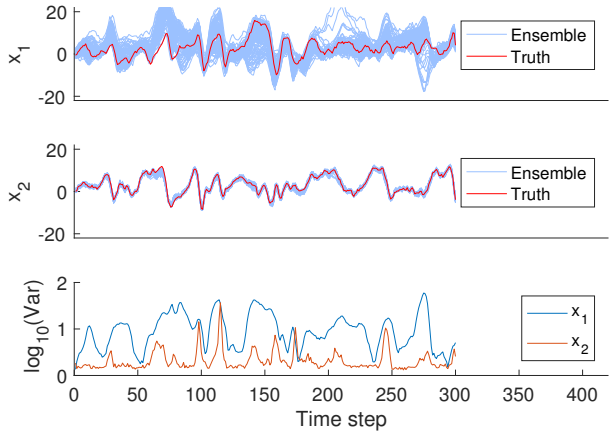

Figure 11: Time evolution of true realisations and an IEWPF ensemble. The components shown are x_1 (not directly observed) and x_2 (directly observed) along with ensemble variances. This ensemble was obtained from the two-stage IEWPF with $\beta = 0.7$ applied to the standard Lorenz96 case with $N_x = 40$.

that the two-stage IEWPF is better calibrated.

Finally, to test the two-stage IEWPF in a setting that is both weakly non-linear and where N_x is much larger than N_e, we run the Lorenz96 case with $N_x = 1000$, $N_e = 25$ and $\beta = 0.75$. Remaining parameter values are unchanged. Figure 14 shows two components of the estimated and true model trajectories in this moderately high-dimensional test case. The top and bottom displays show particle trajectories for an observed component and an unobserved component of the state vector respectively. Filter behaviour is not noticeably different from the lower dimensional case of Figure 11. The variance is clearly larger for the unobserved state. For both variables, coverage is reasonable, and no bias effects are apparent. As is common in Lorenz models, the state is sometimes very uncertain; for instance at time steps 120–150, and especially so for the unobserved state. Even though the state dimension is much larger here, it seems that β can be in the same range ($\beta = 0.75$ in this plot as opposed to $\beta = 0.7$ in the $N_x = 40$ case).

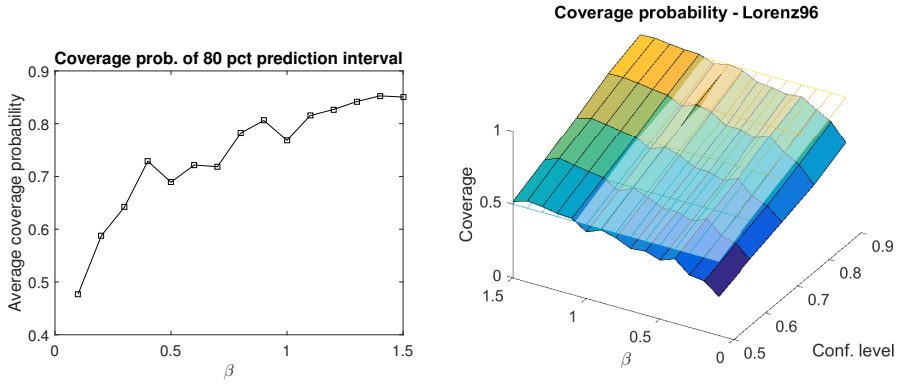

Figure 12: Empirical coverage probability as a function of prediction interval confidence and the value of the tuning parameter β. Best calibration is achieved for $\beta \approx 1$. **Left:** Average empirical coverage probability of 80 per cent prediction intervals constructed from two-stage IEWPF ensembles with a range of β values. **Right:** Coverage probabilities as a function of β for a range of confidence levels.

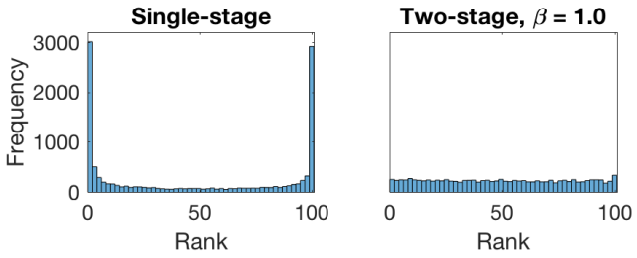

Figure 13: Rank histograms for one run of the single-stage IEWPF and one run of the two-stage IEWPF with $\beta = 0.7$ on the Lorenz96 model test case. Ranks are aggregated over all steady-state time steps and all state elements.

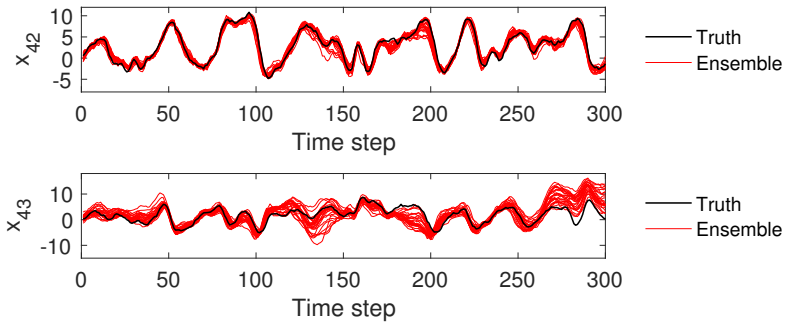

Figure 14: Two components of the trajectories of the ground truth and two-stage IEWPF ensemble in the high-dimensional Lorenz96 case with $N_x = 1000$ and $N_e = 25$. The components shown are x_{42} (directly observed) and x_{43} (not directly observed). The common scale factor in the two-stage IEWPF was set to $\beta = 0.75$ in this case.

5 Conclusion

In this paper we have presented a modification to the Implicit Equal Weight Particle Filter (IEWPF). The suggested approach is applicable to data assimilation in both low and high-dimensional state space models. When applied to a weakly nonlinear dynamical model, the revised IEWPF performed reasonably well even in the $N_x \gg N_e$ case.

A MATLAB implementation of the IEWPF algorithm ran in seconds to minutes when applied to the Gauss-linear and Lorenz96 data assimilation test cases in Section 4. The runtime increases in proportion with the number of time steps. It is not sensitive to the state dimension, but the numerical solution of the equal-weights equation may require more iterations to achieve convergence for very high-dimensional cases. The IEWPF update can be carried out in parallel for each particle once the weight-offsets for the whole ensemble have been determined, making the algorithm easy to parallelize. The memory requirements of the IEWPF are largely determined by the need to store the model error and observation error covariance matrices.

When using the revised filtering method, particles are updated according to a two-stage proposal scheme which draws two separate and orthogonal perturbation vectors from the proposal distribution. By using two stages, we are able to eliminate the gap in the proposal distribution of the original IEWPF, ensuring that the proposal distribution is nonzero everywhere in state space. The random perturbations are scaled to keep weights equal and to achieve the correct ensemble spread. Accurately adjusting the spread requires tuning of the corresponding scale parameter. In our setting this is a single parameter which we propose to specify through the use of coverage probabilities or rank histograms. Other approaches might be possible here, for instance a criterion guided by the distribution of weights in the optimal particle density proposal. We leave this for future work. Note that we have chosen to keep the tuning parameter fixed throughout the data assimilation period. One could also adjust this parameter dynamically, so that different values can be used at different assimilation times as is done in adaptive inflation for EnKFs.

The updating schemes described in this paper are constructed so that the distribution of particle weights will be uniform. An alternative would be to fix some proportion of the weights, allowing the rest to vary. This has the possible benefit of balancing contracting and expanding solutions of the equal-weights equation. Another possibility is to select multiple target weights, so that the overall distribution of weights is uniform within sub-ensembles of particles, but may differ between sub-ensembles.

References

Ades, M. and Van Leeuwen, P. J. (2013). An exploration of the equivalent weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 139(672):820–840.

Asch, M., Bocquet, M., and Nodet, M. (2016). *Data assimilation: methods, algorithms, and applications*, volume 11. SIAM, Philadelphia.

Brookes, M. (2011). The matrix reference manual. *URL http://www.ee.imperial.ac.uk*.

Chorin, A., Morzfeld, M., and Tu, X. (2010). Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Science*, 5(2):221–240.

Chorin, A. J., Morzfeld, M., and Tu, X. (2013). A survey of implicit particle filters for data assimilation. In *State-Space Models*, pages 63–88. Springer.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Evensen, G. (2009). *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media.

Fletcher, S. J. (2017). *Data Assimilation for the Geosciences: From Theory to Application*. Elsevier, Amsterdam.

Frei, M. and Künsch, H. R. (2013). Bridging the ensemble kalman and particle filters. *Biometrika*, 100(4):781–800.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140(2), pages 107–113. IET.

Hamill, T. M. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Lorenz, E. N. (1995). Predictability: A problem partly solved, vol. i. In *Proceedings of Seminar at ECMWF*.

Morzfeld, M., Tu, X., Atkins, E., and Chorin, A. J. (2012). A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066.

Rezaie, J. and Eidsvik, J. (2012). Shrinked (1- α) ensemble kalman filter and α gaussian mixture filter. *Computational Geosciences*, 16(3):837–852.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640.

Snyder, C., Bengtsson, T., and Morzfeld, M. (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750–4761.

Stordal, A. S., Karlsen, H. A., Nævdal, G., Skaug, H. J., and Vallès, B. (2011). Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. *Computational Geosciences*, 15(2):293–305.

Van Leeuwen, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12):4089–4114.

Van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999.

Van Leeuwen, P. J., Cheng, Y., and Reich, S. (2015). *Nonlinear data assimilation*. Springer.

Weisstein, E. W. (2002). Lambert W-function. From MathWorld—A Wolfram Web Resource. `http://mathworld.wolfram.com/LambertW-Function.html` Last accessed on 1 August 2018.

Zhu, M., Van Leeuwen, P. J., and Amezcua, J. (2016). Implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 142(698):1904–1919.

Appendix

Single-stage IEWPF equal-weights equation

Suppose we have a forecast ensemble $\{\mathcal{M}(\mathbf{x}_i^{n-1})\}_{i=1}^{N_e}$ with equal weights at the previous time step, and we want to update this ensemble with respect to the observation $\mathbf{y}^n$. We sample the perturbation vector $\boldsymbol{\xi}_i^n$ from the proposal distribution $q(\boldsymbol{\xi}_i^n)$. The updated weight of particle i is

$$w_i^n = \frac{p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1},\mathbf{y}^n)p(\mathbf{y}^n|\mathbf{x}_i^{n-1})}{q(\boldsymbol{\xi}_i^n)} \left\| \frac{\partial \mathbf{x}_i^n}{\partial \boldsymbol{\xi}_i^n} \right\|$$

Taking $-2\log$ of both sides gives

$$-2\log w_i^n = -2\log p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1},\mathbf{y}^n) - 2\log p(\mathbf{y}^n|\mathbf{x}_i^{n-1}) + 2\log q(\boldsymbol{\xi}_i^n) - 2\log\left(\left\| \frac{\partial \mathbf{x}_i^n}{\partial \boldsymbol{\xi}_i^n} \right\|\right)$$

and using that $p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1},\mathbf{y}^n) \propto \exp(-\frac{1}{2}(\mathbf{x}_i^n - \mathbf{x}_i^{n,\mathrm{a}})^T\mathbf{P}^{-1}(\mathbf{x}_i^n - \mathbf{x}_i^{n,\mathrm{a}}))$ and $p(\mathbf{y}^n|\mathbf{x}_i^{n-1}) \propto \exp(-\frac{1}{2}\varphi_i)$ this becomes

$$\begin{aligned}
-2\log w_i^n &= (\mathbf{x}_i^n - \mathbf{x}_i^{n,\mathrm{a}})^T\mathbf{P}^{-1}(\mathbf{x}_i^n - \mathbf{x}_i^{n,\mathrm{a}}) + \varphi_i - (\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n - 2\log\left(\left\|\frac{\partial \mathbf{x}_i^n}{\partial \boldsymbol{\xi}_i^n}\right\|\right) \\
&= (\alpha_i - 1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n + \varphi_i^n - 2\log\left(\left\|\frac{\partial \mathbf{x}_i^n}{\partial \boldsymbol{\xi}_i^n}\right\|\right) \\
&= (\alpha_i - 1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n + \varphi_i^n - 2N_x\log\alpha_i^{1/2} \\
&\quad - 2\log\left(\left|1 + \frac{\partial \alpha_i^{1/2}}{\partial \boldsymbol{\xi}_i^n}\frac{\boldsymbol{\xi}_i^n}{\alpha_i^{1/2}}\right|\right) - 2\log\left\|\mathbf{P}^{1/2}\right\|
\end{aligned}$$

where we have used the single-stage IEWPF update scheme (6) and rewritten the determinant of the Jacobian using Sylvester's determinant lemma (Brookes, 2011). Equating the negative log-weight with a constant C now gives

$$(\alpha_i - 1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n + \varphi_i^n - 2N_x\log\alpha_i^{1/2} - 2\log\left(\left|1 + \frac{\partial \alpha_i^{1/2}}{\partial \boldsymbol{\xi}_i^n}\frac{\boldsymbol{\xi}_i^n}{\alpha_i^{1/2}}\right|\right) - 2\log\left\|\mathbf{P}^{1/2}\right\| = C$$

or

$$(\alpha_i - 1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n - 2N_x\log\alpha_i^{1/2} - 2\log\left(\left|1 + \frac{\partial \alpha_i^{1/2}}{\partial \boldsymbol{\xi}_i^n}\frac{\boldsymbol{\xi}_i^n}{\alpha_i^{1/2}}\right|\right) = c_i \tag{18}$$

where $c_i = C - \varphi_i$ and $2\log\left\|\mathbf{P}^{1/2}\right\|$ has been absorbed into C since it is the same for all particles. If it is assumed that α_i depends on $\boldsymbol{\xi}_i^n$ only through $g_i = (\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n$, then the above equation simplifies to

$$(\alpha_i - 1)g_i - 2\log\left(\alpha_i^{N_x/2-1}\frac{\partial \alpha_i g_i}{\partial g_i}\right) = c_i.$$

Writing $b_i = \alpha_i g_i$, this is

$$b_i - g_i + 2\log g_i^{N_x/2-1} - 2\log\left(b_i^{N_x/2-1}\left|\frac{\partial b_i}{\partial g_i}\right|\right) = c_i.$$

Separating the terms involving b_i and g_i gives

$$\log\left(\exp\left(\frac{b_i}{2}\right) b_i^{N_x/2}\left|\frac{\partial b_i}{\partial g_i}\right|\right) = \log\left(\exp\left(\frac{g_i}{2}\right) g_i^{N_x/2-1}\right) - \frac{c_i}{2}$$
$$\exp\left(-\frac{b_i}{2}\right) b_i^{N_x/2-1}\left|\frac{\partial b_i}{\partial g_i}\right| = \exp\left(-\frac{g_i}{2}\right) g_i^{N_x/2-1} \exp\left(-\frac{c_i}{2}\right)$$

which, when integrated from $g_i = 0$ to $g_i = \tilde{g}_i$, yields the single-stage equal-weights equation (11).

Two-stage IEWPF equal-weights equation

In the two-stage IEWPF we draw two orthogonal perturbation vectors $\boldsymbol{\xi}_i^n$ and $\boldsymbol{\eta}_i^n$ from the proposal distribution $q(\boldsymbol{\xi}_i^n, \boldsymbol{\eta}_i^n)$, and use them to compute the updated particle position according to the two-stage update scheme (12).

Orthogonal pairs of multivariate normal perturbation vectors are generated as follows:

1. Generate $\boldsymbol{\eta}$ and $\mathbf{z}$ by sampling from the standard N_x-variate Gaussian distribution.
2. Decompose $\mathbf{z}$ into two components $\mathbf{z} = \mathbf{z}_\parallel + \mathbf{z}_\perp$ where $\mathbf{z}_\parallel$ is parallel to $\boldsymbol{\eta}$ and $\mathbf{z}_\perp$ is orthogonal to $\boldsymbol{\eta}$.
3. Let $\boldsymbol{\xi} = \sqrt{(\mathbf{z}^T\mathbf{z})/(\mathbf{z}_\perp^T\mathbf{z}_\perp)}\mathbf{z}_\perp$, so that $\boldsymbol{\xi}^T\boldsymbol{\xi} = \mathbf{z}^T\mathbf{z}$.

For a pair $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathbb{R}^{N_x}$ satisfying $\boldsymbol{\xi}^T\boldsymbol{\eta} = 0$, we have

$$q(\boldsymbol{\xi}, \boldsymbol{\eta}) \propto \exp\left(-\frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}\right) \exp\left(-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}\right) I(\boldsymbol{\xi}^T\boldsymbol{\eta} = 0), \tag{19}$$

where $I(\boldsymbol{\xi}^T\boldsymbol{\eta} = 0)$ is an indicator function that is equal to one if $\boldsymbol{\xi}^T\boldsymbol{\eta} = 0$ and is equal to zero if $\boldsymbol{\xi}^T\boldsymbol{\eta} \neq 0$.

Under the two-stage scheme, assuming that since β is shared between particles the Jacobian matrix of the map from $\boldsymbol{\eta}_i^n$ to $\mathbf{x}_i^n$ can be omitted, the expression

for the weight of particle i is

$$w_i^n = \frac{p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1},\mathbf{y}^n)p(\mathbf{y}^n|\mathbf{x}_i^{n-1})}{q(\boldsymbol{\xi}_i^n,\boldsymbol{\eta}_i^n)}\left\|\frac{\partial \mathbf{x}_i^n}{\partial \boldsymbol{\xi}_i^n}\right\|$$

and taking $-2\log$ and proceeding as in the single-stage case now gives

$$\begin{aligned}(\alpha_i-1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n + (\beta-1)(\boldsymbol{\eta}_i^n)^T\boldsymbol{\eta}_i^n + \varphi_i^n - 2N_x\log\alpha_i^{1/2} \\ -2\log\left(\left|1+\frac{\partial\alpha_i^{1/2}}{\partial\boldsymbol{\xi}_i^n}\frac{\boldsymbol{\xi}_i^n}{\alpha_i^{1/2}}\right|\right) - 2\log\left\|\mathbf{P}^{1/2}\right\| = C\end{aligned}$$

and

$$(\alpha_i-1)(\boldsymbol{\xi}_i^n)^T\boldsymbol{\xi}_i^n - 2N_x\log\alpha_i^{1/2} - 2\log\left(\left|1+\frac{\partial\alpha_i^{1/2}}{\partial\boldsymbol{\xi}_i^n}\frac{\boldsymbol{\xi}_i^n}{\alpha_i^{1/2}}\right|\right) = C-\varphi_i-(\beta-1)(\boldsymbol{\eta}_i^n)^T\boldsymbol{\eta}_i^n$$

which matches equation (18) if we let

$$c_i = C-\varphi_i-(\beta-1)(\boldsymbol{\eta}_i^n)^T\boldsymbol{\eta}_i^n = C+(1-\beta)(\boldsymbol{\eta}_i^n)^T\boldsymbol{\eta}_i^n-\varphi_i. \tag{20}$$

Consequently α_i can be determined in the two-stage case by solving equation (11) with offset c_i given by (20).