Aspects of Particle Filtering in High-Dimensional Spaces

Peter Jan van Leeuwen

Data Assimilation Research Centre (DARC), University of Reading, UK,
p.j.vanleeuwen@reading.ac.uk

Abstract. Nonlinear data assimilation is high on the agenda in all fields of the geosciences as with ever increasing model resolution and inclusion of more physical (biological etc) processes, and more complex observation operators the data-assimilation problem becomes more and more nonlinear. The suitability of particle filters to solve the nonlinear data assimilation problem in high-dimensional geophysical problems will be discussed. Several existing and new schemes will be presented and it is shown that at least one of them, the Equivalent-Weights Particle Filter, does indeed beat the curse of dimensionality and provides a way forward to solve the problem of nonlinear data assimilation in high-dimensional systems.

1 Introduction

There is a growing need for nonlinear data-assimilation methods for high dimensional systems. This is evidenced by the inclusion of more and more nonlinear processes in the systems at hand. Also more and more indirect observations are being utilised, leading quite often to highly nonlinear relations between model space and observation space.

Several nonlinear data-assimilation methods based on Metropolis-Hastings have been generated, including Langevin-sampling, Hybrid Monte-Carlo, and more efficient extensions, but all have in common that they are extremely inefficient: one first has to generate large numbers of samples to converge to the correct posterior probability density function (pdf), and then several samples have to be generated for each independent sample of the posterior. The former problem can be eliminated by using techniques from inverse modelling, like the commonly used variational schemes, e.g. 4Dvar, but the latter remains an unresolved issue.

Particle filters are another branch of nonlinear data assimilation methods, but have long been though of as very inefficient too because of the so-called curse of dimensionality, in which it is claimed that the number of particles needed to generate a few samples from the high-probability areas of the posterior grows exponentially with the dimension of the state vector (Snyder et al, 2007; Van Leeuwen, 2009). This is due to the highly peaked likelihood in high-dimensions, leading to particle weights varying widely, with a few particles having much

higher weight than all the others. Even the so-called 'optimal proposal density' particle filters suffer from this problem. It should be noticed that it is not the dimension of the state space that is the problem, but the dimension of the observation space. The higher this last dimension, the more peeked the likelihood is, and the more unlikely it is for the majority of particles to end up close to all of them (see Ades and Van Leeuwen, 2013).

However, it has recently been shown that this problem can easily be avoided by construction by leading the particles towards the observations and at the same time forcing them into position in state space such that they have weights very close to each other (Van Leeuwen, 2010, 2011, Ades and Van Leeuwen 2013, 2014a,b).

In this short paper we will discuss several existing and new particle filter variants and discuss their relative merits and problems. A very simple numerical example will be used to show that at least one particle filter exists that does beat the curse of dimensionality, the so-called Equivalent-Weights Particle Filter. The paper is closed with a summary and discussion of possible future directions in nonlinear filtering.

2 The standard Particle Filter

Starting from Bayes Theorem:

$$p(x|y) = \frac{p(y|x)}{p(y)} p(x) \tag{1}$$

in which $x \in \Re^d$ is the state vector and $y \in \Re^M$ is the observation vector, we introduce a representation of the prior $p(x)$ as a sum of particles:

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i) \tag{2}$$

to find for the posterior:

$$p(x|y) = \sum_{i=1}^{N} w_i \delta(x - x_i) \tag{3}$$

in which we introduced the likelihood weights

$$w_i = \frac{1}{N} \frac{p(y|x_i)}{\sum_{j=1}^{N} p(y|x_j)} \tag{4}$$

To derive this we used the expansion

$$p(y) = \int p(y|x) p(x) \, dx \tag{5}$$

These weights measure how close each particle is to all observations. The issue in high-dimensional systems, or rather high-dimensional observation spaces, is that independent observations lead to very highly peaked likelihoods. A simple example will illustrate this nicely. Suppose we have two particles, and both are very close to all observations. We assume that the observation errors are independent Gaussian distributed and that the first particle is 0.1 standard deviations away from *all* observations, and the other particle is 0.2 standard deviations from all observations. This is of course highly artificial, but it will illustrate the point. The weight of particle one will be

$$w_1 \propto \exp\left[-\frac{1}{2}(y-H(x_1))R^{-1}(y-H(x_1))\right] = \exp(-0.005M) \tag{6}$$

in which M is the number of independent observations, and we explored the independence of the observations. Similarly, the weight of particle two will be

$$w_2 \propto \exp\left[-\frac{1}{2}(y-H(x_2))R^{-1}(y-H(x_2))\right] = \exp(-0.02M) \tag{7}$$

The ratio of these two weights is

$$\frac{w_2}{w_1} = \exp(-0.015M) \tag{8}$$

Assuming the number of independent observations to be 1000, a moderate number for the geosciences, we find that this ratio is about 10^{-7}! Hence we see that even though the two particles are both doing extremely well, the likelihood is so strongly peaked that one particle has negligible weight compared to the other. Clearly something is needed to save the particle filter, and that is the so-called proposal density. This is explained in detail in e.g. Doucet et al. (2001), see also Van Leeuwen (2009) and will not be repeated here. The basic idea is that instead of drawing samples directly from the prior $p(x)$, we draw the samples from another density $q(x|y)$ where we explicitly include a dependence on the observations. We can always do this as long as we adapt the weights, as follows:

$$p(x|y) = \frac{p(y|x)}{p(y)}p(x) = \frac{p(y|x)}{p(y)}\frac{p(x)}{q(x|y)}q(x|y) \tag{9}$$

Using a particle representation for $q(x|y)$, so drawing particles from q instead of from the prior, we find:

$$p(x|y) = \sum_{i=1}^{N}\frac{p(y|x_i)}{Np(y)}\frac{p(x_i)}{q(x_i|y)}\delta(x-x_i) = \sum_{i=1}^{N} w_i\delta(x-x_i) \tag{10}$$

in which the weights now become:

$$w_i = \frac{p(y|x_i)}{N\sum_j p(y|x_j)}\frac{p(x_i)}{q(x_i|y)} \tag{11}$$

These weights consist of a likelihood part, as before, and a part related to the use of the proposal density. The usefulness of the proposal is that we can choose q such that the particles are closer to the observations, so that the likelihood part of the weights are more equal, while at the same time ensuring that p/q does not spoil this gain in efficiency. The different particle filter variants differ in the proposal density used. In the following we will discuss a few recent developments in generating efficient proposal densities.

3 The Implicit Particle Filter

The implicit particle filter was introduced by Chorin and Tu (2009) and has been further detailed in Chorin et al. (2011). It works as follows. Define a function $F(x)$ as minus the logarithm of the posterior pdf:

$$F(x) = -\log(p(x|y, x_i^m)) \tag{12}$$

in which x can be either a state vector at a certain time, or an evolution of the system over a time window $x = (x^1, ..., x^n)^T$. x_i^m is the starting point of the particle i at the start of the time window starting at time m, and m can be equal to $n-1$. Define the minimum of $F(x)$ as

$$\phi_F = \min(F(x)) \tag{13}$$

The basic idea in the implicit particle filter is to draw samples from a pdf $g(\xi)$ from which it is easy to draw, e.g. a multivariate Gaussian. Define $G(\xi)$ as

$$p(\xi) \propto \exp(-G(\xi)) \tag{14}$$

and denote

$$\phi_G = \min(G(\xi)) \tag{15}$$

The relation between the samples ξ_i and the samples of the posterior pdf x_i is *defined* by the solution of

$$F_i(x) - \phi_{F_i} = G(\xi_i) - \phi_{G_i} \tag{16}$$

where we emphasise that ϕ_F and ϕ_G can depend on the particle index i.

The weight of the particle i using proposal density $g(\xi)$ is given by:

$$w_i = \frac{p(x_i|y)}{q(x_i|y)} = \frac{p(x_i|y)}{g(\xi_i|y)} \left|J(\xi_i)\right| \tag{17}$$

in which

$$J(\xi) = \det\left(\frac{\partial x}{\partial \xi}\right) \tag{18}$$

Using the expression for the posterior and the relation between x and ξ we find for the weights:

$$w_i = \frac{p(x_i|y)}{q(x_i|y)} \propto \frac{\exp(-F_i(x)))}{\exp(-F_i(x) + \phi_{F_i} - \phi_{G_i})} \left|J(\xi_i)\right| = \exp(-\phi_{F_i} + \phi_{G_i}) \left|J(\xi_i)\right| \tag{19}$$

Several proposal densities have been explored in the literature. Here we focus on the so-called random map method (Morzfelt et al, 2013) which takes ξ to be Gaussian distributed $N(0, I)$, so $\phi_G = 0$, and writing

$$x_i = argmin(F_i(x)) + \lambda(\xi_i)\xi_i \tag{20}$$

in which $\lambda(x_i)$ a scalar function of ξ_i. In case $F_i(x)$ is quadratic in x λ is scalar constant and the Jacobian $|J(\xi)|$ is constant over the particles, so can be dropped, leading to:

$$w_i = \exp(-\phi_{F_i}) \tag{21}$$

This is typically the case when observations and model errors are Gaussian distributed and H is linear for a filter, and for a smoother with the additional constraint that the model is linear.

To summarise the procedure is, for each particle:

1) Calculate $argmin(F_i(x))$ (not necessary, but typically done for efficiency)
2) Draw ξ_i from $g(\xi)$
3) Solve for the corresponding x_i, from $F_i(x) - \phi_{F_i} = G(\xi_i) - \phi_{G_i}$
3) Evaluate the weight w_i

In can be shown that this method is very similar to the so-called optimal proposal density when observations are present every time step (see e.g. Ades and Van Leeuwen, 2013). Assuming Gaussian observation errors and a linear observation operator $H(x)$ it is easy to show that the weights are equal to:

$$w_i \propto \exp\left[-\frac{1}{2}(y^n - Hf(x_i^{n-1}))^T(HQH^T + R)^T(y^n - Hf(x_i^{n-1}))\right] \tag{22}$$

in which n is the time index. Ades and Van Leeuwen (2013) show that these minus the logarithm of these weights are non-central χ^2 distributed with variance proportional to the number of independent observations M. Hence, for high-dimensional observation systems the implicit particle filter is expected to be degenerate, which is confirmed by our experiments presented later.

Another potential issue is that the Jacobian $|J(\xi)|$ has to be nonzero, which means that $F_i(x)$ has to be unimodal, at last close to the position of the state that produces the maximum weight for each particle.

4 The Equivalent-weights Particle Filter

The Equivalent-Weights Particle filter introduced in Van Leeuwen (2010) and investigated in detail in Ades and Van Leeuwen (2013, 2014a, 2014b) works as follows.

First, we calculate the state for which $F_i(x)$ is minimal, in which $F_i(x)$ is defined as before as:

$$F_i(x) = -\log(p(x|y, x_i^m) \tag{23}$$

in which x_i^m is the starting point of the particle i at the start of the time window starting at time m, and m can be equal to $n-1$.

After we have done this for each particle we rank the particles in acceding order of $min(F_i(x))$. We choose the number of particles we'd like to keep in the ensemble, let's say 80%. We then set a target weight as the value of $w_{target} = \exp[-min(F_i(x))]$ for that particle that ranks as the 80% particle in the ranking. For instance, if we have 100 particles we rank them and set the target weight as the value of $\exp[-min(F_i(x))]$ for the 80^{th} particle in the ranking.

The next step is to solve for each particle x_i for which its weight $\exp[-min(F_i(x))$ is larger than the target weight (or $min(F_i(x))$ is smaller than $-\log(w_{target})$) as:

$$F_i(x) = -\log(w_{target}) \tag{24}$$

This is the case for 80% of the particles by this construction. The other 20% of the particles cannot reach this target weight: no matter how we move them in state space their weight will be smaller than the target weight. These particles will come back via a resampling step later.

Then we add to each particle a small random perturbation and recalculate the weight for each particle. Since the perturbation is small the weights will not change much, meaning that 80% of the particles will all have very similar, or equivalent, weights. For details see Ades and Van Leeuwen (2014a).

Finally a resampling step is performed in which N particles are drawn from the weighted ensemble of 80% of the particles. This is the whole scheme, which can be summarised as:

1) Calculate $argmin(F_i(x))$
2) Determine w_{target}
3) Solve for x_i^* from $F_i(x) = -\log(w_{target})$
4) Draw η_i from mixture density with small amplitude and write $x_i = x_i^* + \eta_i$.
5) Evaluate final weight

This scheme is not degenerate by construction, as has been shown in e.g. Ades and Van Leeuwen (2014a) who used this scheme in a 65,000 dimensional barotropic vorticity model.

5 Another non-degenerate scheme

Using the scheme above we can easily derive new variants of the EWPF that are not degenerate by construction. We present one example here. The comparison of the two methods discussed above shows that the number of calculations is identical. The implicit particle filter (IPF) first draws the random vector ξ before solving an equation for x_i, while the EWPF first solves for x^* and than adds η. The advantage of the IPF is that the draw is simply done for a Gaussian, but the disadvantage is that the weights are degenerate.

Would it be possible in the EWPF to draw η from a Gaussian before solving for x^* ($= x_i^n$ in that case)? This would mean to replace the procedure above with:

1) Calculate $argmin(F_i(x))$
2) Determine w_{target}
3) Draw ξ_i from $g(\xi) \propto \exp(-G(\xi))$
4) Solve for x_i^* from $F_i(x) = -\log(w_{target}) + G(\xi_i)$
5) Evaluate weight

(Note that step 4 is easily done with the existing software for the Equivalent-Weights Particle Filter by simply replacing $-\log(w_{target})$ with $-\log(w_{target}) + G(\xi_i)$.) The weights will become:

$$\begin{aligned} w_i &= \frac{p(x_i|y)}{q(x_i|y)} \\ &\propto \frac{\exp(-F_i(x_i)))}{\exp(-G(\xi_i))} \, |J(\xi_i)| \\ &= \frac{\exp(-F_i(x_i)))}{\exp(-F_i(x_i) + \log(w_{target}))} \, |J(\xi_i)| \\ &= \exp(-w_{target}) \, |J(\xi_i)| \end{aligned} \tag{25}$$

If $F_i(x)$ is a quadratic function of x_i, as we usually assume via Gaussian errors in observations, H linear, and Gaussian model errors, the Jacobian is constant and drops out, and the weights are all equal again!

Finally, the difference between this scheme and the IPF is that we have a small extra step to calculate w_{target} that ensures that all ϕ_{F_i} are equal to avoid degeneracy.

This is one of the ways to avoid degeneracy in particle filtering, and no doubt many more will be developed over the coming years.

6 A simple numerical example

To test the ideas the standard particle filter, Equivalent-Weights Particle Filter and the Implicit Particle Filter are comparer on a simple model given by:

$$x^1 = x^0 + \eta \tag{26}$$

The distribution of x^0 is taken as independent Gaussian for simplicity. The state vector is d-dimensional and we will investigate the performance of the filters when d increases. Finally, the model errors are also independent Gaussian. The state x^1 is observed as

$$y = x^1 + \epsilon \tag{27}$$

in which the observation errors are also taken independent and Gaussian.

While this system is completely Gaussian, so linear, so the traditional linear data-assimilation methods can be used, we will explore the particle filter performance here. The idea is that if the particle filters fail in this simple linear case it is unlikely they will perform better in a nonlinear setting.

Several experiments were performed with different values for initial, model, and observational errors. Here we report on the following experimental settings, mentioning that other settings give similar results: the initial mean is 0 for each variable, the initial variance is 1 for each variable, the model variance is 0.01, and the observation error variance is 0.16. The number of ensemble members or particles is 10 in all experiments. We set the number of particles kept in the equivalent-weights procedure equal to 80%.

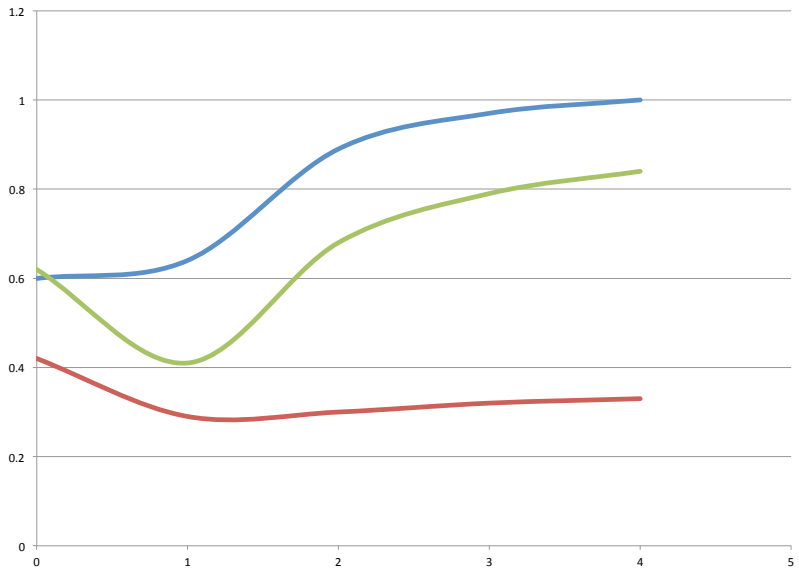

Fig. 1. *Effective ensemble size for SIR (blue), EWPF (red), and IPF (green) as function of the power of the state dimension, so* 10^0 *to* 10^4.

Figure 1 shows the effective ensemble size, defined as

$$N_{eft} = \frac{1}{\sum w_i^2} \tag{28}$$

for the standard particle filter with proposal density equal to the prior, the Equivalent-Weights Particle Filter (EWPF) and the Implicit Particle Filter (IPF), which is equal to a particle filter using the so-called Optimal Proposal density. The results shown are averages over 1000 experiments. We can clearly see that apart from a state dimension of 1, all filters are degenerate, except for the EWPF, in which we find constant effective ensemble sizes of 8 out of 10, identical to the percentage of particles kept in the equivalent-weights procedure.

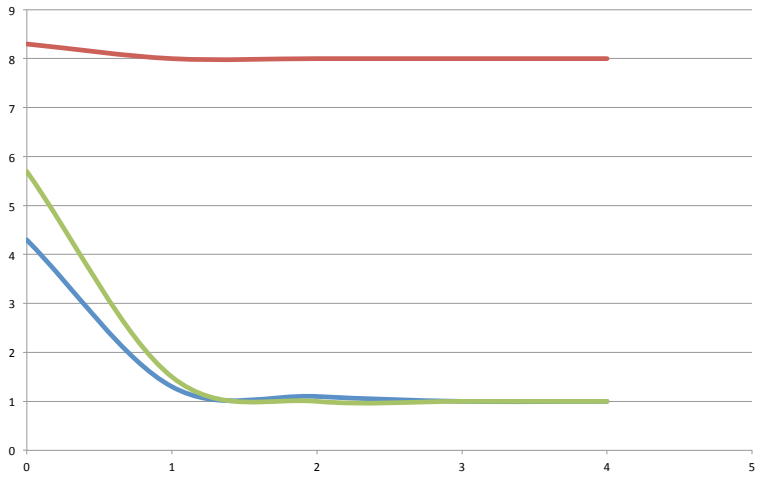

Fig. 2. *State-space averaged root-mean-square error of ensemble mean for SIR (blue), EWPF (red), and IPF (green) as function of the power of the state dimension, so* 10^0 *to* 10^4.

Figure 2 shows the root-mean-square error (RMSE) of the ensemble mean from the truth for each method, averaged over state space. The results shown are again averages over 1000 experiments. One would expect that the RMSE is slightly smaller that the observation error, which is 0.4 in this case. Only the EWPF is able to do this, all other methods fail and return a RMSE close to that of the prior. This is consistent with the effective ensemble size results above.

One might wonder why the IPF does not do much better than the standard particle filter. The reason is simply that the members are moved to better positions but the filter is degenerate, so the mean only consist of the best particle, which does depend on the initial particle position, so the RMSE will converge to that of the prior. Note that, for this specific case in which the prior at time zero is a Gaussian one could take the drawing from that Gaussian into the sampling via the proposal q, in which case all samples would have equal weight as this system is linear. In that case the IPF reduces to the Ensemble Kalman Smoother (Evensen and Van Leeuwen, 2000). The point here is that, in general, the prior at time zero is non-Gaussian as it arises from a sequential application of the algorithm so the starting point at time zero is a number of particles with unknown distribution.

7 More realistic high-dimensional applications

We have applied the EWPF to several high-dimensional problems. Ades and Van Leeuwen (2014a) applied the method to a 65,000 dimensional barotropic vorticity model and found that the particle filter is indeed not degenerate, and quite robust. For instance, figure 2 shows how the histogram from 32 particles captures the main features of the histogram generated using 512 particles.

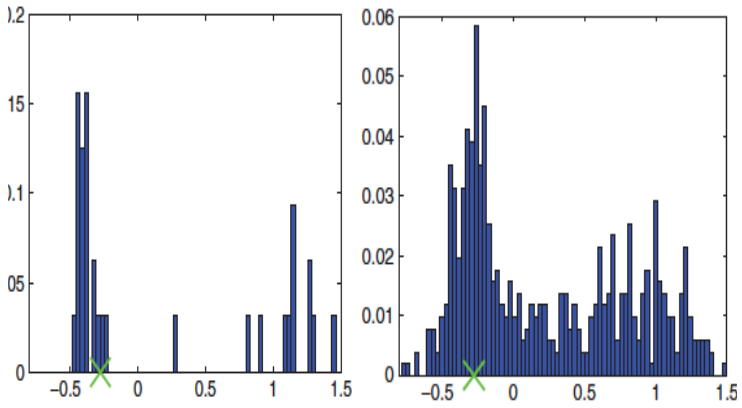

Fig. 3. *Marginal posterior pdf of an unobserved point using 32 and 512 particles. The green cross denotes the truth value for the vorticity at this point. Note the similarity between the main features of the pdf.*

In this experiment only half of the state was observed, and of the observed part only every other point. The pdfs shown in figure 3 are for a point in the middle of the unobserved half, where nonlinearities can grow and non-Gaussian pdfs are common.

Finally we show first results from an application of the EWPF to the climate model HadCM3, with about 2 million state variables. In this experiment only the Sea-Surface Temperature (SST) was observed every day. The initial results are encouraging, although not all problems have been solved. Figure 4 shows rank histograms of atmospheric surface pressure, the oceanic and atmospheric velocity fields, and and the ocean temperature in the first layer (which is different from the SST). Is shows that for several of the model variables the rank histograms are flat, indicating a proper ensemble, for others the rank histograms are U-shaped, indicating an under-dispersive ensemble. Ideally all marginal histograms would be flat, but it will be clear that this is hard to achieve in general. The main point

here is, however, that, again, the EWPF is not degenerate and can be fine-tuned further.

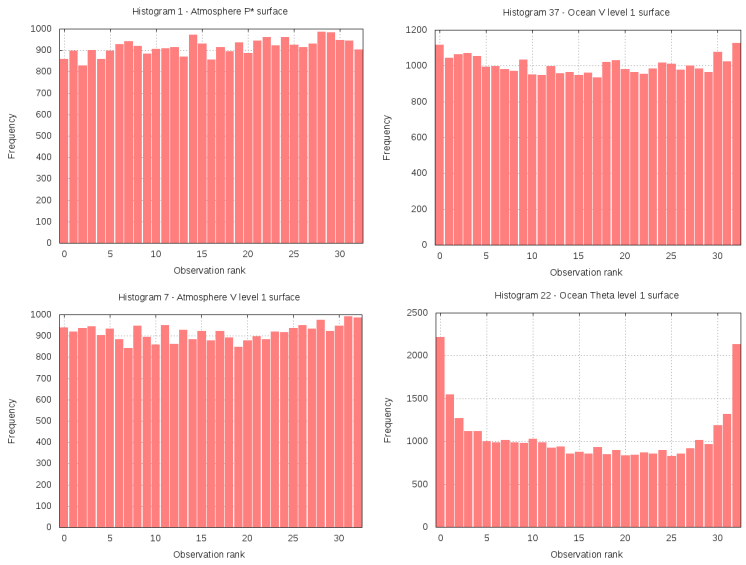

Fig. 4. *Rank histogram showing how the truth ranks in the ensemble. For a proper ensemble the rank histogram should be flat. This is indeed the case for the atmospheric surface pressure and the oceanic and atmospheric surface meridional velocity fields, but the ocean temperature in the first ocean layer shows signs of an under-dispersive ensemble.*

8 Conclusions

High-dimensional spaces poses a few counter intuitive features. One of those directly related to nonlinear filtering in these spaces is that the likelihood is extremely narrow when the number of independent observations is large. A simple example with only 1000 independent observations showed catastrophic filter collapse when the proposal density is taken equal to the prior. New developments like the implicit particle filter try to avoid that problem by exploring the so-called optimal proposal density. It has been shown theoretically that this filter and its variants also suffer from ensemble collapse in Ades and Van Leeuwen (2013), and that finding is confirmed in an very simple experiment in this paper. The only particle filter that is insensitive to the dimension of the state or, rather, of the observation space is the Equivalent-Weights Particle Filter, which avoids ensemble collapse by construction. The simple experiment has confirmed this, as have more realistic systems, such as the barotropic vorticity equation model and the climate model HadCM3.

We have also shown that it is easy to combine the EWPF scheme with other schemes like the IPF and formulate other non-degenerate schemes. One undesirable feature of the EWPF is that one has to set the percentage of particles kept in the equivalent weight step, and the results do depend on this (se e.g. Ades and Van Leeuwen, 2012). Ideally one would get rid of this step and manages to find non-degenerate particles without resampling. There is room for good ideas to explore these methods further.

References

Ades, M. and P. J. Van Leeuwen, 2013: An exploration of the equivalent weights particle filter. Quartely Journal of Meteorology, 139, 820-840.

Ades, M. and P. J. Van Leeuwen, 2014a: The equivalent weights particle flter in a high 662 dimensional system. Quarterly Journal of Meteorology, accepted.

Ades, M. and P. J. Van Leeuwen, 2014b: The effect of the equivalent-weights particle flter on dynamical balance in a primitive equation model. Monthly Weather Review, accepted.

Chorin AJ, Morzfeld M, Tu X. 2010. Interpolation and iteration for nonlinear filters. Communications in Applied Mathematics and Computational Science 5: 221 240.

Chorin AJ, Tu X. 2009. Implicit sampling for particle filters. Proceedings of the National Academy of Sciences 106(41): 17 249-17254.

Doucet, A., N. de Freitas, and N. Gordon, 2001: Sequential Monte-Carlo Methods in Practice. 686 Springer-Verlag.

Evensen, G. and P.J. van Leeuwen 2000: n Ensemble Kalman smoother for nonlinear dynamics, Monthly Weather Review, 129, 709-728.

Morzfeld M, Tu X, Atkins E, Chorin AJ. 2012. A random map implementation of implicit filters. Journal of Computational Physics 231: 2049-2066.

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

Van Leeuwen PJ. 2009. Particle filtering in geophysical systems. Monthly Weather Review 137: 4089-4114.

Van Leeuwen PJ. 2010. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. Quarterly Journal of the Royal Meteorological Society 136: 1991-1999.