Particle Filters for nonlinear data assimilation in high-dimensional systems

Peter Jan van Leeuwen *

Data-Assimilation Research Centre (DARC)

June 6, 2016

Abstract

Particle Filters are Monte-Carlo methods used for Bayesian Inference. Bayesian Inference is based on Bayes Theorem that states how prior information about a system, encoded in a probability density function, is updated when new information in the form of observations of that system become available. This process is called data assimilation in the geosciences. This contribution discusses what particle filters are and what the main issue is when trying to use them in the geosciences, in which the data-assimilation problem is typically very high dimensional. An example is numerical weather forecasting, with a state-space size of a billion or more. Then it discusses recent progress made in trying to beat the so-called 'curse of dimensionality', such as localisation and clever ways to slightly change the model equations to obtain better approximations to the posterior probability density via so-called proposal densities. This culminates in a new class of particle filters that is indeed able to provide estimates of the posterior probability density. The emphasis is not on mathematical rigour but on conveying the main new ideas in this rapidly growing field.

1 Introduction

In this chapter we will discuss particle filters and their use in the geosciences. A general review on the application and usefulness of particle filters in geosciences in given in Van Leeuwen (2009), and a general overview of particle filtering is given by the excellent book by Doucet et al, (2001). The book by provides an excellent introduction in the mathematical background of particle filters. Several problems are encountered when trying to apply particle filters to high-dimensional problems, summarised with the so-called 'curse of dimensionality'. However, a lot of progress has been made the last couple of years, and this chapter will discuss some of these new developments, with an emphasis on the geosciences. Because of space limitations the presentation will be focussed on the general ideas and not on mathematical rigour. Excellent text books like those of Del Moral (2004) and Bain and Crisan (2009) can be consulted for that rigour.

First the basic idea behind particle filters is presented, followed by why this basic formulation can never work for large-dimensional systems. We discuss resampling as a way to increase the efficiency of particle filters. Then we will discuss proposal densities, which form the major contribution in this paper. Specifically, we will discuss a wider class of proposal densities than commonly used, which will allow us to avoid the otherwise inevitable filter degeneracy that plagues even the so-called optimal proposal density. We show that this new class allows us an enormous amount of freedom to build particle filters for very high dimensional systems, and present an example of a successful approach that works in systems of any dimension by construction. This is the first example of undoubtately an enormous growth in useful methods for extremely high dimensional systems encountered in the geosciences. To keep the text fluent I kept the literature references to a minimum; a more comprehensive, but slightly

* Peter jan van Leeuwen, Department of Meteorology, University of Reading, United Kingdom
E-mail: p.j.vanleeuwen@reading.ac.uk

outdated, literature list for application of particle filters in the geosciences can be found in Van Leeuwen (2009).

We discuss the basic Particle filter as an importance sampler, and show why straightforward implementation will lead to so-called degeneracy, in which the effective ensemble size reduces to a very small number of particles, and the method fails. At the same time the strength of particle filters will be investigated, namely that particle filters are completely nonlinear, have no problems with model balances after updates (at least not the simple versions), and their performance is not reliant on a good representation of the error covariance of the model state. The latter point has been overlooked by geoscientists in the past, but is actually a major selling point.

While perhaps less obvious to a mathematician, this balance issue is a crucial ingredient in geoscience applications. In short, it means that different model variables are coupled to each other by closely, but not exactly, following so-called 'balance relations'. These relations are closely related to the model attractor. So different model variables are not independent from each other, and a too large random perturbation to a model field will result in breaking these balance relations, pushing the system off the attractor, and resulting in strong adaptation of the system back to the attractor. The real world does only react this way if there is an physical (chemical, biological,...) process that pushes the system away from the attractor, while the filtering process might introduce large pushes to the system that have nothing to do with the physics (chemistry, biology,...). As an example, a strongly unbalanced update of an atmospheric model tends to introduce strong gravity waves in the adjustment to balance that lead to spurious rain generation.

As real-world examples show us again and again, the data-assimilation problem is typically a nonlinear one, especially in the geosciences. The models we use to simulate are almost never linear, and the observation operator that relates model states to observations is quite often nonlinear too. While linearisations have been shown to be very useful to tackle real-world problems, there are several problems that are so nonlinear that these linearisations are just not good enough.

As is well known, Kalman filters either assume that the update is a linear combination between observations and prior estimate, the BLUE, or they assume that both the prior and the likelihood are Gaussian distributed in the model state. Of course, when the system is weakly nonlinear the Kalman filter can be used quite efficiently, and even iterations of the Kalman filter update can be performed. But when the system is highly nonlinear these iterations are unlikely to converge, and if they do, it is unclear to what. Also the interpretation of the ensemble as a measure for the posterior covariance becomes questionable. It is important to realise that the (Ensemble) Kalman filter is not variance minimising for a non-Gaussian posterior pdf!

Variational methods like 4DVar and the Representer method look for the maximum of the posterior probability density function (pdf), or to the minimum of minus the logarithm of this pdf, which amounts to the same state. When the system is linear or Gaussian it is easy to prove that there is indeed one maximum. Also for a weakly nonlinear system variational methods are very useful, and the variational problem can be solved by iteration, sometimes called 'incremental 4DVar'. However, when the problem is highly nonlinear it can be expected that the posterior pdf has several local maxima, and the variational methods will converge to one of them. This is not necessarily the global maximum. Another issue is the lack of covariance information. Even if the inverse of the Hessian, the local curvature of the pdf at the maximum, could be calculated it does not represent the covariance of the full posterior pdf. This means that no estimate of the accuracy (width of posterior pdf) of the maximum is available.

Nonlinear data assimilation is a whole new ball game, especially when the posterior pdf is multi modal. What does the 'best estimate' mean? Is it the mean of the posterior pdf? Well, definitely not when the posterior is bimodal and the two modes have equal probability mass and are of equal shape. In that case the mean will fall between the two peaks. Is the global maximum the best estimate? If the posterior pdf has multiple maxima of equal size the answer is no. Also, when the maximum is related to a relatively small probability mass, it is also not that useful. It becomes clear that the notion of 'best estimate' depends very strongly on the application, and is perhaps not a very useful concept in nonlinear data assimilation.

The solution to the data-assimilation problem is not a best estimate, but the posterior pdf itself. That is exactly what Bayes Theorem tells us, given the prior pdf and the likelihood, we can calculate the posterior pdf, and that is the answer. And the calculation is extremely simple, just a multiplication. So, this little excursion into nonlinear data assimilation learns us that data assimilation is NOT AN

INVERSE PROBLEM, but a multiplication problem. That is the starting point for this chapter on Particle Filters.

2 A simple Particle filter based on Importance Sampling

The particle filters we will discuss here are based on Importance Sampling. The most straight-forward implementation is what is called Basic Importance Sampling here. (In the statistical literature one usually finds Importance Sampling described with a proposal density different from the prior model pdf. However, for pedagogical reasons we present Importance Sampling in the following way.) Basic Importance sampling is straightforward implementation of Bayes Theorem as we will show below.

2.1 Basic Importance Sampling

The basic data assimilation problem is to infer the probability density function (pdf) of a state $x \in \Re^{N_x}$ given a set of observations $y \in \Re^{N_y}$. We have prior information in terms of a prior pdf $p(x)$, and the observations are obtained by measuring the true state of the system via

$$y = H(x_{true}) + \epsilon \tag{1}$$

in which $H(.)$ maps model state x to observation space. ϵ is a measurement error, as all real observations of a real system have these errors.

The idea is to represent the prior pdf by a set of particles $x_i \in \Re^{N_x}$, which are delta functions centred around state vectors x_i, and from which all statistical measures of interest can be calculated, like mean, covariance etc. If one represents the prior pdf by a number of particles, or ensemble members, like in the Ensemble Kalman Filter, so

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

and we use this in Bayes Theorem;

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)\, dx} \tag{3}$$

we find

$$p(x|y) = \sum_{i=1}^{N} w_i \delta_{x_i} \tag{4}$$

in which the weights w_i are given by:

$$w_i = \frac{p(y|x_i)}{\sum_{j=1}^{N} p(y|x_j)} \tag{5}$$

The density $p(y|x_i)$ is the probability density of the observations given the model state x_i, which is often taken as a Gaussian:

$$p(y|x_i) = A \exp\left[-\frac{(y - H(x_i))^2}{2\sigma^2}\right] \tag{6}$$

in which $H(x_i)$ is the measurement operator, which is the model equivalent of the observation y, and σ is the standard deviation of the observation error. When more measurements are available, which might have correlated errors, the above should be the joint pdf of all these measurements.

Weighting the particles just means that their relative importance in the probability density changes. For instance, if we want to know the mean of the function $f(x)$ we now have:

$$\overline{f(x)} = \int f(x)p(x)\, dx \approx \sum_{i=1}^{N} w_i f(x_i) \tag{7}$$

Common examples for $f(x)$ are x itself, giving the mean of the pdf, and the squared deviation from the mean, giving the covariance.

Up to now, we haven't specified what x is. It can be a state vector x^n at a certain time n, or x can be a model trajectory over some time window $(0, n\Delta t)$, so $x = x^{0:n} = (x^0, x^1, ..., x^n)$ over n time steps. Here the superscript is the time index, and the subscript is the sample, or particle.

A practical way to implement the particle filter is to calculate the one time or the trajectory sequentially over time, which is where the name 'filter' comes from. The idea is to write the prior density as

$$p(x^{0:n}) = p(x^n|x^{0:n-1})p(x^{0:n-1}) \tag{8}$$

Using that the state vector evolution is Markov, i.e. to predict the future we only need the present, not the past, we can write:

$$p(x^{0:n}) = p(x^n|x^{n-1})p(x^{n-1}|p(x^{n-2})...p(x^1|x^0)p(x^0) \tag{9}$$

Before we continue it is good to realise what the so-called transition densities $p(x^n|x^{n-1})$ actually mean. Consider a model evolution equation given by:

$$x^n = f(x^{n-1}) + \beta^n \tag{10}$$

in which $\beta^n \in \Re^{N_x}$is a random term or factor in the model equation that describes the error in the model equation. The idea is that the model is not perfect, i.e. any numerical model used in the geosciences that is used to simulate the real world has errors (and these tend to be significant!). These errors are unknown (otherwise we would include them as deterministic terms in the equations) but we assume we are able to say something about their statistics, e.g. their mean, covariance, etc. Typically one assumes the errors in the model equations are Gaussian distributed with zero mean and known covariance, but that is not always the case. To draw from such a transition density $p(x^n|x^{n-1})$ means to draw β^n from its density and evaluate the model equation given above. In fact, for normal, or Gaussian, distributed model errors β^n with mean zero and covariance Q, we can write:

$$p(x^n|x^{n-1}) = N(f(x^{n-1}), Q) \tag{11}$$

Note that we assume the model errors are additive in this chapter. Multiplicative model errors in which the size of the random forcing is dependent on the state x can be accounted for too, but we use additive model errors here for simplicity.

Let us now continue with Importance Sampling. If we also assume that the observations at different times, conditional on the states at those times, are independent, which is not necessary for the formulation of the theory, but keeps the notation so much simpler, we have for the likelihood:

$$p(y^{1:n}|x^{0:n}) = p(y^n|x^n)...p(y^1|x^1) \tag{12}$$

where we used that y^j is not dependent on x^k with $j \neq k$ when x^j is known. The posterior density can now be written as:

$$\begin{aligned} p(x^{0:n}|y^{1:n}) &= \frac{p(y^{1:n}|x^{0:n})p(x^{0:n})}{p(y^{1:n})} \\ &= \frac{p(y^n|x^n)...p(y^1|x^1)p(x^n|x^{n-1})...p(x^1|x^0)p(x^0)}{p(y^n), ..., p(y^1)} \\ &= \frac{p(y^n|x^n)p(x^n|x^{n-1})}{p(y^n)} ... \frac{p(y^1|x^1)p(x^1|x^0)p(x^0)}{p(y^1)} \end{aligned} \tag{13}$$

Realising that the last ratio in this equation is actually equal to $p(x^{0:1}|y^1)$ we find the following sequential relation:

$$p(x^{0:n}|y^{0:n}) = \frac{p(y^n|x^n)p(x^n|x^{n-1})}{p(y^n)} p(x^{0:n-1}|y^{1:n-1}) \tag{14}$$

This expression allows us to find the full posterior with the following sequential scheme(see figure 1):

1 Sample N particles x_i from the initial model probability density $p(x^0)$, in which the superscript 0 denotes the time index.

2 Integrate all particles forward in time up to the measurement time. In probabilistic language we denote this as: sample from $p(x^n|x_i^{n-1})$ for each i,
i.e. for each particle x_i run the model forward from time $n-1$ to time n using the nonlinear model equations. The stochastic nature of the forward evolution is implemented by sampling from the density that describes the random forcing of the model.

3 Calculate the weights according to (5), normalise them so their sum is equal to 1, and attach these weights to each corresponding particle. Note that the particles are not modified, only their relative weight is changed!

4 Increase n by one and repeat 2 and 3 until all observations have been processed.

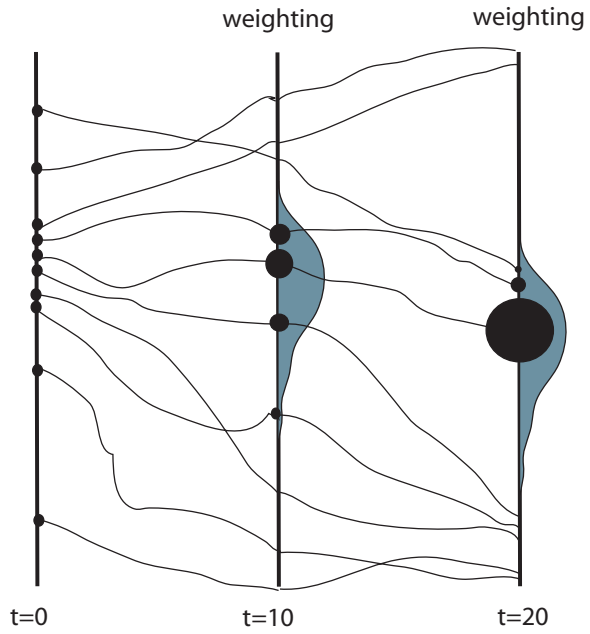

Figure 1: *The standard particle filter with Importance Sampling. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time 0. At time 10 the likelihood is displayed together with the new weights of each particle. At time 20 only 2 members have weights different from zero: the filter has become degenerate.*

2.2 Why particle filters are so attractive

Despite the problems discussed just now, their advantages compared to traditional methods should not be underestimated. First of all, they do solve the complete nonlinear data assimilation problem, see the discussion at the beginning of this chapter.

Furthermore, the good thing about importance sampling is that the particles are not modified, so that dynamical balances are not destroyed by the analysis. The bad thing about importance sampling is that the particles are not modified, so that when all particles move away from the observations they are not pulled back to the observations. Only their relative weights are changed.

And finally it is stressed how simple this scheme is compared to traditional methods like 3- or 4DVar and (Ensemble) Kalman filters. The success of these scheme depends heavily on the accuracy and error covariances of the model state vector. In 3- and 4DVar this leads to complicated covariance structures to ensure balances etc. In Ensemble Kalman filters artificial tricks like covariance inflation and localisation are needed to get good results in high dimensional systems. Particle filters do not have these difficulties.

However, there is (of course) a drawback. Even if the particles manage to follow the observations in time, the weights will differ more and more. Application to even very low-dimensional systems shows

that after a few analysis steps one particle gets all the weight, while all other particles have very low weights (see figure 1, at $t = 20$). That means that the statistical information in the ensemble becomes too low to be meaningful. This is called *filter degeneracy*. It has given importance sampling a low profile until resampling was invented, see the next section.

3 Reducing the variance in the weights

Several methods exists to reduce the variance in the weights, and we discuss Sequential Importance Resampling here. See Van Leeuwen(2009) for other methods. In resampling methods the posterior ensemble is resampled so that the weights become more equal (Gordon et al., 1993). In the next section after this one methods are discussed that do change the positions of the prior particles in state space to improve the likelihood of the particles, i.e. to bring them closer to the observations before the weighting with the likelihood is applied.

3.1 Resampling

The idea of resampling is simply that particles with very low weights are abandoned, while multiple copies of particles with high weight are kept for the posterior pdf in the sequential implementation. In order to restore the total number of particles N, identical copies of high-weight particles are formed. The higher the weight of a particle the more copies are generated, such that the total number of particles becomes N again. Sequential Importance Re-sampling (SIR) does the above, and makes sure that the weights of all posterior particles are equal again, to $1/N$.

Sequential Importance Re-sampling is identical to Basic Importance Sampling but for a resampling step after the calculation of the weights. The 'flow chart' reads (see figure 2):

1 Sample N particles x_i from the initial model probability density $p(x^0)$.

2 Integrate all particles forward in time up to the measurement time (so, sample from $p(x^n|x_i^{n-1})$ for each i)

3 Calculate the weights according to (5) and attach these weights to each corresponding particle. Note that the particles are not modified, only their relative weight is changed!

4 Re-sample the particles such that the weights are equal to $1/N$.

5 Repeat 2, 3 and 4 sequentially until all observations have been processed.

It is good to realise that the resampling step destroys the smoother character of the method. All particles that are not chosen in the resampling scheme are lost, and their evolution is broken. So the smoother estimate is build of of lesser and lesser particles over time, until it consists of only one particle, loosing again all statistical meaning.

The resampling can be performed in many ways, and we discuss the most used.

1) *Probabilistic resampling*

Most straightforward is to directly sample randomly from the density given by the weights. Since this density is discrete and one-dimensional this is an easy task. However, due to the random character of the sampling, so-called sampling noise is introduced. Note that this method is actually generalised Bernoulli for those versed in sampling techniques..

2) *Residual Sampling*

To reduce the sampling noise Residual Sampling can be applied. In this re-sampling method all weights are multiplied with the ensemble size N. Then n copies are taken of each particle i in which n is the integer part of Nw_i. After obtaining these copies of all members with $Nw_i \geq 1$, the integer parts of Nw_i are subtracted from Nw_i. The rest of the particles needed to obtain ensemble size N are than drawn randomly from this resulting distribution.

3) *Stochastic Universal Sampling*

While Residual Sampling reduces the sampling noise, it can bee shown that Stochastic Universal Sampling has lowest sampling noise. In this method all weights are put after each other on the unit interval $[0, 1]$. Then a random number is drawn from a uniform density on $[0, 1/N]$, and N line pieces starting from the random number, and with interval length $1/N$ are laid on the line

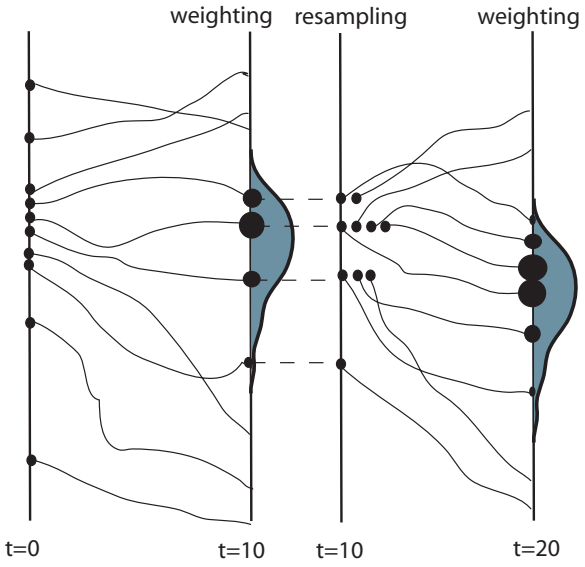

Figure 2: *The Particle Filter with Resampling, also called Sequential Importance Resampling. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. At time 10 the particles are weighted according to the likelihood, and resampled to obtain an equal-weight ensemble.*

$[0, 1]$. A particle is chosen when one of the end points of these line pieces falls in the weight bin of that particle. Clearly, particles with high weights span an interval larger than $1/N$ and will be chosen a number of times, while small weight particles have a negligible change of being chosen.

3.2 Is resampling enough?

Snyder et al. (2008) prove that resampling will not be enough to avoid filter collapse, The problem is related to the large number of observations, which make the likelihood peak in only a very small portion of the observation space. The conclusion is that more is needed than simple resampling to solve the degeneracy problem.

3.3 Convergence of particle filters to the target density

Several convergence results for particle filters exist, with varying conditions on the transition probabilities and likelihood functions. A basic result is due to Del Moral and Guionnet (2001), who show that

$$\sup_{n\geq 0} E\left(|p^N(x^n|y^n) - p(x^n|y^n)|\right) \leq \frac{C}{N^{\alpha/2}} \tag{15}$$

for $\alpha \leq 1$. See details on constants C and α in Del Moral and Guionnet (2001). Here $p^N(.|.)$ denotes the particle representation of the full density $p(.|.)$. This result essentially shows that the system has limited memory due to the model noise and the resampling step (so the interaction between the particles), and approximation errors tend to dissipate.

More can be found in text books like Del Moral (2004) and Bain and Crisan (2009). Recent work (e.g. Van Handel (2009), Tong and Van Handel, 2012) connected the stability of nonlinear filters to standard concepts in linear filtering, specifically stability, observability, and detectability. Noteworthy is also the work by Le Gland et al. (2011) on the convergence properties of the Ensemble Kalman Filter, which applies a Gaussian assumption on the prior each time observations become available.

4 Localisation in Particle Filtering

One way to reduce the number of observations is to allow them to influence only that part of the model state close to them. Close in this case relates to the physical distance. The rationale behind this is that there is no direct instantaneous physical relation between the temperature in say Reading in the UK, and the temperature in New York. So a temperature observation in Reading should only influence the model state in the neighbourhood of Reading.

This is an idea from Ensemble Kalman Filtering, called localisation. There, it is needed because information between different grid points is encoded in the ensemble covariance. Since the number of ensemble members in high-dimensional systems is low, the covariance estimate is rather noisy. The true covariance between the temperature in Reading and New York is zero, but the ensemble estimate will display spurious correlations that have to be suppressed. That is done in the localisation procedure, of which several variants exist, but they all ensure that observations have a finite spatial influence.

Our basic motivation is different in particle filtering as ensemble-based covariance estimates are not needed. In particle filtering it is just used to avoid filter degeneracy by reducing the number of observations at each grid point. This idea was first discussed by Bengtsson et al, (2003), and Van Leeuwen (2003).

The theoretical justification for localisation needs further work, but important first steps have been set by Rebeschini and Van Handel (2015), who show that when the system has decaying correlations in space localisation can keep filter stability. The idea is that the decaying correlations lead to a limited spatial memory of the system, so that approximation errors tend to dissipate in space. They proof that localisation can beat the curse of dimensionality for a so-called block particle filter, in which each grid point is part of a certain block, and only observations inside that block are used for updates of the prior pdf at that grid point. As noted by the authors, this will lead to large approximation errors at the boundaries. Furthermore, as pointed out by the authors and earlier practitioners, like Bengtsson et al, (2003), and Van Leeuwen (2003), a fundamental problem is that the discontinuous boundaries can lead to unrealistic model states, potentially ruining the model forecasts. Another interesting recent work in this area is Beskos et al. (2014), who discuss the convergence rate a particle filter that updates a sequence of artificial targets, and these targets can be chosen as marginal posterior pdf's, making a direct connection with localisation.

On the practical side, several methods have been developed, e.g. by Cotter and Reich (2015), Poterjoy (2015), and Penny and Miyoshi (2015). No theoretical justification of these methods exists, and the algorithms typically rely heavily on spatial smoothing to ensure realistic model fields, loosing part of the full nonlinearity. It is encouraging that both practitioners and mathematicians are both interested in this approach, no doubt leading to rapid progress in the near future.

Due to space limitations localisation will not be discussed further here, apart from mentioning that localisation alone is unlike to solve the degeneracy problem for several geoscience problems with high observational density, like numerical weather prediction, because the physical decorrelation lengthscale will be larger than the localisation radius needed to avoid too many observations inside the localisation area. As a rule of thumb, 10 independent observations is typically too much to prevent filter degeneracy, and e.g. high-density radar observations of rain fall intensity are known to cause problems.

5 The proposal density

In this part we will concentrate on recent developments in using the so-called proposal transition density in solving the degeneracy problem. Related to decreasing the variance of the weights is to make sure that all model integrations end up close to the new observations, or, more precisely, ensuring that all posterior particles have similar weights.

First, we discuss what a proposal density is in particle filtering, and how it can be useful. This is then illustrated with using an Ensemble Kalman Filter as proposal density. This is followed by a discussion of more traditional methods of which we choose the Auxiliary Particle Filter.

Next, we discuss methods that change the model equations by bringing information on where the future observations are directly into the model equations. We start with the so-called Optimal proposal density and show that that idea doesn't work in high-dimensional spaces with large numbers of independent observations. The optimal proposal density is a one-time-step scheme, assuming observations every time step. The so-called Implicit Particle Filter extends this to multiple time steps between

observations. It is shown that the implicit particle filter can be interpreted as a weak-constraint 4DVar on each particle, with fixed initial condition. We will show that when the number of independent observations is large also this filter will be problematic.

The major breakthrough in the field is the extension of the class of proposal densities to situations where each new particle is informed not only by the weights of the other particles as in a resampling scheme, but also by their relative positions in state space. That will allow us to generate a method that will not be degenerate by construction, the Equivalent-Weights Particle filter. Its working is illustrated on a 65,000 dimensional barotropic vorticity model of atmospheric or oceanic flow, hinting that particle filters are now mature enough to explore in e.g. operational numerical weather prediction settings.

We are now to discuss a very interesting property of particle filters that has received little attention in the geophysical community. We start from Bayes:

$$p(x^{0:n}|y^{0:n}) = \frac{p(y^n|x^n)p(x^n|x^{n-1})}{p(y^n)} p(x^{0:n-1}|y^{1:n-1}) \tag{16}$$

To simplify the analysis, and since we concentrate on a filter here, let us first integrate out the past, to get:

$$p(x^n|y^{0:n}) = \frac{p(y^n|x^n)}{p(y^n)} \int p(x^n|x^{n-1}) p(x^{n-1}|y^{1:n-1}) \, dx^{n-1} \tag{17}$$

This expression does not change when we multiply and divide by a so-called proposal transition density $q(x^n|x^{n-1}, y^n)$, so:

$$p(x^n|y^{0:n}) = \frac{p(y^n|x^n)}{p(y^n)} \int \frac{p(x^n|x^{n-1})}{q(x^n|x^{n-1}, y^n)} q(x^n|x^{n-1}, y^n) p(x^{n-1}|y^{1:n-1}) \, dx^{n-1} \tag{18}$$

As long as the support of $q(x^n|x^{n-1}, y^n)$ is equal to or larger than that of $p(x^n|x^{n-1})$ we can always do this. This last condition makes sure we don't divide by zero. Let us now assume that we have an equal-weight ensemble of particles from the previous analysis at time $n-1$, so

$$p(x^{n-1}|y^{1:n-1}) = \sum_{i=1}^{N} \frac{1}{N} \delta_{x_i^{n-1}} \tag{19}$$

Using this in the equation above gives:

$$p(x^n|y^{0:n}) = \sum_{i=1}^{N} \frac{1}{N} \frac{p(y^n|x^n)}{p(y^n)} \frac{p(x^n|x_i^{n-1})}{q(x^n|x_i^{n-1}, y^n)} q(x^n|x_i^{n-1}, y^n) \tag{20}$$

As a last step, we run the particles from time $n-1$ to n, i.e. we sample from the transition density. However, instead of drawing from $p(x^n|x_i^{n-1})$, so running the original model, we sample from $q(x^n|x_i^{n-1}, y^n)$, so from a modified model. Let us write this modified model as

$$x^n = g(x^{n-1}, y^n) + \hat{\beta}^n \tag{21}$$

so that we can write for the transition density, assuming $\hat{\beta}^n$ is Gaussian distributed with covariance $\hat{Q}$:

$$q(x^n|x^{n-1}, y^n) = N(g(x^{n-1}, y^n), \hat{Q}) \tag{22}$$

Drawing from this density leads to:

$$p(x^n|y^{0:n}) = \sum_{i=1}^{N} \frac{1}{N} \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)} \delta(x^n - x_i^n) \tag{23}$$

so the posterior pdf at time n can be written as:

$$p(x^n|y^{1:n}) = \sum_{i=1}^{N} w_i \delta_{x_i^n} \tag{24}$$

with weights w_i given by:

$$w_i = \frac{1}{N} \frac{p(y^n|x_i^n)}{p(y^n)} \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)} \tag{25}$$

We recognise the first factor in this expression as the likelihood, and the second as as a factor related to using the proposal transition density instead of the original transition density to propagate from time $n-1$ to n, so it is related to the use of the proposed model instead of the original model. Note that because the factor $1/N$ and $p(y^n)$ are the same for each particle and we are only interested in relative weights, we will drop them from now on, so

$$w_i = p(y^n|x_i^n) \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)} \tag{26}$$

Finally, let us formulate an expression for the weights when multiple model time steps are present between observation times. Assume the model needs m time steps between observations. This means that the ensemble at time $n-m$ is an equal weight ensemble, so

$$p(x^{n-m}|y^{1:n-m}) = \sum_{i=1}^{N} \frac{1}{N} \delta_{x_i^{n-m}} \tag{27}$$

We will explore the possibility of a proposal density at each model time steps, so for the original model we write

$$p(x^n|y^{0:n}) = \frac{p(y^n|x^n)}{p(y^n)} \int \prod_{j=n-m+1}^{n} p(x^j|x^{j-1}) p(x^{n-m}|y^{1:n-m}) \, dx^{n-m:n-1} \tag{28}$$

and introducing a proposal transition density at each time step we find:

$$p(x^n|y^{0:n}) = \frac{p(y^n|x^n)}{p(y^n)} \int \prod_{j=n-m+1}^{n} \frac{p(x^j|x^{j-1})}{q(x^j|x^{j-1}, y^n)} q(x^j|x^{j-1}, y^n) p(x^{n-m}|y^{1:n-m}) \, dx^{n-m:n-1} \tag{29}$$

Using the expression for $p(x^{n-m}|y^{1:n-m})$ above and choosing randomly from the transition proposal density $q(x^j|x^{j-1}, y^n)$ at each time step leads to:

$$w_i = p(y^n|x_i^n) \prod_{j=n-m+1}^{n} \frac{p(x_i^j|x_i^{j-1})}{q(x_i^j|x_i^{j-1}, y^n)} \tag{30}$$

5.1 Example: the EnKF as proposal

As an example we will explore this technique with the Gaussian of the EnKF as the proposal density. First we have to evaluate the prior transition density. Since we know the starting point of the simulation, x_i^{n-1}, and its end point, the posterior EnKF sample x_i^n, and we know the model equation, written formally as:

$$x_i^n = f(x_i^{n-1}) + \beta_i^n \tag{31}$$

we can determine β_i^n from this equation directly. We also know the distribution from which this β_i^n is supposed to be drawn, let us say a Gaussian with zero mean and covariance Q. We then find for the transition density:

$$p(x_i^n|x_i^{n-1}) \propto \exp\left[-1/2 \left(x_i^n - f(x_i^{n-1})\right) Q^{-1} \left(x_i^n - f(x_i^{n-1})\right)\right] \tag{32}$$

This will give us a number for each $[x_i^{n-1},\ x_i^n]$ combination.

Let us now calculate the proposal density $q(x_i^n|x_i^{n-1}, y^n)$. This depends on the ensemble Kalman filter used. For the Ensemble Kalman filter with perturbed observations the situation is as follows. Each particle in the updated ensemble is connected to those before analysis as:

$$x_i^n = x_i^{n,old} + K^e \left(y + \epsilon_i - H(x_i^{n,old})\right) \tag{33}$$

in which ϵ_i is the random error drawn from $N(0, R)$ that has to be added to the observations in this variant of the ensemble Kalman filter. K^e is the ensemble Klman gain, i.e. the Kalman gain using

the prior error covariance calculated from the prior ensemble. The particle prior to the analysis comes from that of the previous time step through the stochastic model:

$$x_i^{n,old} = f(x^{n-1}) + \beta_i^n \tag{34}$$

Combining these two gives:

$$x_i^n = f(x_i^{n-1}) + \beta_i^n + K^e \left(y + \epsilon_i - H(x_i^{n-1}) - H(\beta_i^n))\right) \tag{35}$$

or

$$x_i^n = f(x_i^{n-1}) + K^e \left(y - H(f(x_i^{n-1}))\right) + (1 - K^e H)\beta_i^n + K^e \epsilon_i \tag{36}$$

assuming that H is a linear operator. The right-hand side of this equation has a deterministic and a stochastic part. The stochastic part provides the transition density going from x_i^{n-1} to x_i^n. Assuming both model and observation errors to be Gaussian distributed and independent we find for this transition density:

$$q(x_i^n | x_i^{n-1} y^n) \propto \exp\left[-1/2 \left(x_i^n - \mu_i^n\right)^T \Sigma_i^{-1} \left(x_i^n - \mu_i^n\right)\right] \tag{37}$$

in which μ_i^n is the deterministic 'evolution' of x, given by:

$$\mu_i^n = f(x_i^{n-1}) + K^e \left(y - H(x_i^{n-1})\right) \tag{38}$$

and the covariance Σ_i is given by:

$$\Sigma_i = (1 - K^e H) Q (1 - K^e H)^T + K^e R K^{eT} \tag{39}$$

where we assumed that the model and observation errors are uncorrelated. It should be realized that x_i^n does depend on all $x_j^{n,old}$ via the Kalman gain, that involves the error covariance P^e. Hence we have calculated $q(x_i^n | P^e, x_i^{n-1}, y^n)$ instead of $q(x_i^n | x_i^{n-1}, y^n)$, in which P^e depends on all other particles. The reason why we ignore the dependence on P^e is that in case of an infinitely large ensemble P^e would be a variable that depends only on the system, not on specific realizations of that system. This is different from the terms related to x_i^n, that will depend on the specific realization for β_i^n even when the ensemble size is 'infinite'. (Hence another approximation related to the finite size of the ensemble comes into play here and at this moment it is unclear how large this approximation error is.)

The calculation of $p(x^n | x^{n-1})$ and $q(x_i^n | x_i^{n-1} y^n)$ look like very expensive operations. By realizing that Q and R can be obtained from the ensemble of particles, computationally efficient schemes can easily be derived.

We can now determine the full new weights. Since the normalization factors for the transition and the posterior densities are the same for all particles the weights are easily calculated. The procedure now is as follows (see figure 3):

1 Run the ensemble up to the observation time

2 Perform a (local) EnKF analysis of the particles

3 Calculate the proposal weights $w_i^* = p(x_i^n | x_i^{n-1}) / q(x_i^n | x_i^{n-1} y^n)$

4 Calculate the likelihood weights $w_i = p(y^n | x_i^n)$

5 Calculate the full relative weights as $w_i = w_i * w_i^*$ and normalize them.

6 Resample

It is good to realize that the EnKF step is only used to draw the particles close to the observations. This means that when the weights are still varying too much, one can do the EnKF step with much smaller observational errors. This might look like overfitting but it is not since the only thing we do in probabilistic sense is to generate particles to those positions in state space where the likelihood is large.

Finally, other variants of the EnKF, like the adjusted and the transform variants can be used too, as detailed in Van Leeuwen (2009). The efficiency of using the EnKF as proposal is under debate at the moment. The conclusions so far seem to be that using the EnKF as proposal in high-dimensional systems does not work. What has not been tested, however, is to use EnKF proposals with smaller observation matrix R, and more possibilities are still open, like using localisation (see later on).

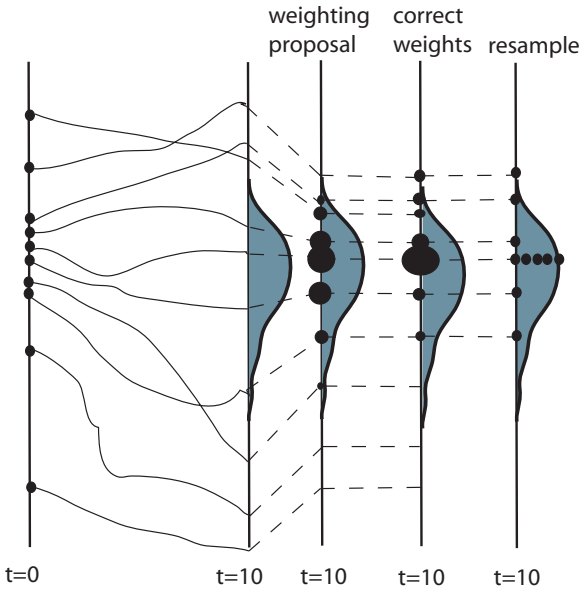

Figure 3: *The particle filter with proposal density. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. At time 10 the particles are brought closer to the observations by using e.g. the EnKF. Then they are weighted with the likelihood and these weights are corrected for the artificial EnKF step.*

5.2 The Auxiliary Particle Filter

In the auxilary particle filter the ensemble at time $n-1$ is weighted with information of the likelihood at time n, see Pitt and Shephard, (1999). In this method one generates a representation of each particle at the time of the new observation, e.g. by integrating each particle from time $n-1$ to time n using zero model noise. (Depending on the complexity of the stochastic model integrator this can save considerable time.) Then the particles are weighted with the observations, and a new resampled ensemble is integrated from $n-1$ to arrive closer to the observations. A flow chart reads (see figure 4):

1 Integrate each particle from $n-1$ to n with simplified dynamics (e.g. without model noise), producing the a representation of the proposal density $q(x^n|x_i^{n-1}, y^n)$.

2 Weight each particle with the new observations as

$$\beta_i \propto p(y^n|x_i^n) w_i^{n-1} \tag{40}$$

These weights are called the 'first-stage weights' or the 'simulation weights'.

3 Resample the particles i at time $n-1$ with these weights, and use this resampled particles j_i as a representation of the proposal density by integrating each forward to n with the full stochastic model, so choosing from $q(x^n|x_{j_i}^{n-1}, y^n)$. Note that j_i connects the original particle i with its new position in state space, that of particle j.

4 Re-weight the members with weights

$$w_i^n = \frac{1}{A} p(y^n|x_i^n) \frac{p(x_i^n|x_{j_i}^{n-1})}{q(x_i^n|x_{j_i}^{n-1}, y^n)\beta_{j_i}} \tag{41}$$

in which A is the normalization factor. A resampling step can be done, but is not really necessary because the actual resampling is done at step 3.

The name 'auxiliary' comes from the introduction of the member index j_i in the formulation. This member index keeps track of the relation between the first-stage weights and the particle sample at $n-1$.

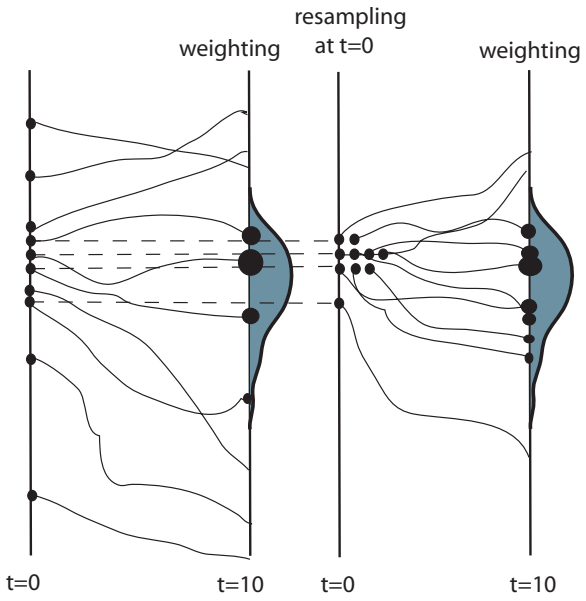

Figure 4: *The Auxiliary Particle Filter. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. At time 10 the particles are weighted according to the likelihood. These weights are used at time 0 to rerun the ensemble up to time 10.*

It should be noted that 2N integrations have to be performed with this method, one ensemble integration to find the proposal, and one for the actual pdf. If adding the stochastic noise is not expensive step 1 can be done with the stochastic model, which comes down to doing Sequential Importance Resampling twice. However, one could also use a simplified model for the first set of integrations. A geophysical example would be to use a quasi-geostrophic model for the first set of integrations, and the full model for the second. One can imagine to do it even more times, zooming in into the likelihood, but at a cost of performing more and more integrations of the model. Figure 4 displays how the method works.

5.3 Including future observations in the model equations

So far we have discussed proposal density applications in which the model equations were not changed directly. Of course, in e.g. the auxiliary particle filter one could use a different model for the first set of integrations to obtain the first-stage weights, but the future observations were not used directly in the model equations. However, much more efficient scheme's can be derived that change the model equations such that each particle is pulled towards the future observations at each time step. By keeping track of the weights associated with this it can be assured that the correct problem is solved, and the particles are random samples from the posterior pdf.

As mentioned before, the idea of the proposal transition density is that we draw samples from that density instead of from the original model. Furthermore, these samples can be dependent on the future observations. To see how this works, let us write the stochastic model equation as:

$$x_i^n = f(x_i^{n-1}) + \beta_i^n \tag{42}$$

First we have to understand how this equation is related to the transition density $p(x_i^n|x_i^{n-1})$. The probability to end up in x_i^n starting from x_i^{n-1} is related to β_i^n. For instance, if $\beta_i^n = 0$, so no model error, a perfect model, this probability is 1 if the x_i^n, x_i^{n-1} pair fulfils the perfect model equations, and zero otherwise. So, in this case $p(x_i^n|x_i^{n-1})$. would be a delta function centred on $f(x_i^{n-1})$. However, the more realistic case is that the model error is nonzero. The transition density will now depend on the distribution of the stochastic random forcing. Assuming Gaussian random forcing with mean zero

and covariance Q, so $\beta_i^n \sim N(0,Q)$, we find

$$p(x_i^n|x_i^{n-1}) \propto N(f(x_i^{n-1}),Q) \tag{43}$$

As mentioned above, we will not use the normal model equation for each particle, but a modified model equation, one that 'knows' about future observations, and actually draws the model to those observations. Perhaps the simplest example is to add a term that relaxes the model to the future observation, like

$$x_i^n = f(x_i^{n-1}) + \beta_i^n + K^n(y^{n+m} - H(x_i^{n-1}) \tag{44}$$

in which $n+m$ is the next observation time. Note that the observation operator H does not contain any model integrations, it is just the evaluation of x_i^{n-1} in observation space. The reason is simple, we don't have x_i^{n+m} yet. Clearly, each particle i will now be pulled towards the future observations, with relaxation strength related to matrix K^n. In principle, we are free to choose K^n, but it is reasonable to assume that it is related to the error covariance of the future observation R, and that of the model equations Q. We will show possible forms in the examples discussed later.

With the simple relaxation, or other techniques, we have ensured that all particles end up closer to the observations. But we can't just alter the model equations, we have to compensate for this trick. This is why the proposal density turns up in the weights. Each time step the weight of each particle changes with

$$w_i^n = \frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1},y^n)} \tag{45}$$

between observation times. This can be calculated in the following way. Using the modified model equations, we know x_i^{n-1} for each particle, that was our starting point, and also x_i^n. So, assuming the model errors are Gaussian distributed, this would become

$$p(x_i^n|x_i^{n-1}) \propto \exp\left[-\frac{1}{2}\left(x_i^n - f(x_i^{n-1})\right)^T Q^{-1}\left(x_i^n - f(x_i^{n-1})\right)\right] \tag{46}$$

The proportionality constant is not of interest since it is the same for each particle, and drops out when the relative weights of the particles are calculated. Note that we have all ingredients to calculate this, and $p(x_i^n|x_i^{n-1})$ is just a number.

For the proposal transition density we use the same argument, to find:

$$\begin{aligned} q(x_i^n|x_i^{n-1}y^n) &\propto \exp\left[-\frac{1}{2}\left(x_i^n - f(x_i^{n-1}) - K^n(y^n - H(x^{n-1})\right)^T Q^{-1}\left(x_i^n - f(x_i^{n-1}) - K^n(y^n - H(x^{n-1}))\right)\right] \\ &= \exp\left[-\frac{1}{2}{\beta_i^n}^T Q^{-1}\beta_i^n\right] \end{aligned} \tag{47}$$

Again, since the did choose β to propagate the model state forward in time, we can calculate this and it is just a number. In this way, any modified equation can be used, and we know, at least in principle, how to calculate the appropriate weights.

5.4 The Optimal proposal density

In the literature the so-called optimal proposal density is described (e.g. Doucet et al, 2000). It is argued that taking $q(x^n|x^{n-1},y^n) = p(x^n|x^{n-1},y^n)$ results in optimal weights, although the proof of this has been lacking until recently (Snyder et al., 2015). While this is true, in a later section we will extend the class of particle filters and show that non-degenerate particle filters for systems with arbitrary large dimensions can be constructed.

The following develops the argument put forward by Ades and Van Leeuwen (2013) that the even the optimal proposal density cannot avoid the curse of dimensionality. Assume observations every time step, and a resampling scheme at every time step, so that a equal-weighted ensemble of particles is present at time $n-1$. Furthermore, assume that model errors are Gaussian distributed $N(0,Q)$ and observation errors are Gaussian distributed according to $N(0,R)$. First, using the definition of conditional densities we can write:

$$p(x^n|x^{n-1},y^n) = \frac{p(y^n|x^n)p(x^n|x^{n-1})}{p(y^n|x^{n-1})} \tag{48}$$

where we used $p(y^n|x^n, x^{n-1}) = p(y^n|x^n)$. Using this proposal density gives posterior weights:

$$\begin{aligned} w_i &= p(y^n|x_i^n)\frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_i^{n-1}, y^n)} \\ &= p(y^n|x_i^n)\frac{p(x_i^n|x_i^{n-1})}{p(x_i^n|x_i^{n-1}, y^n)} \\ &= p(y^n|x_i^{n-1}) \end{aligned} \tag{49}$$

The latter can be expanded as:

$$w_i = \int p(y^n, x^n|x^{n-1})\, dx^n = \int p(y^n|x^n)p(x^n|x^{n-1})\, dx^n \tag{50}$$

in which we again used $p(y^n|x^n, x^{n-1}) = p(y^n|x^n)$. Using the Gaussian assumptions mentioned above (note, the state is never assumed to be Gaussian), we can perform the integration to obtain:

$$w_i \propto \exp\left[-\frac{1}{2}\left(y^n - Hf(x_i^{n-1})\right)^T (HQH^T + R)^{-1}\left(y^n - Hf(x_i^{n-1})\right)\right] \tag{51}$$

Note that we have just calculated the probability density of $p(y^n|x_i^{n-1})$.

To estimate the order of magnitude of the first two moments of the distribution of $y - Hf(x_i^{n-1})$ it is expanded to $y - Hx_t^n + H\left(x_t^n - f(x_i^{n-1})\right)$ in which x_t^n the true state at time n. If we now use $x_t^n = f(x_t^{n-1}) + \beta_t^n$ this can be expanded further as $y - Hx_t^n + H\left(f(x_t^{n-1}) - f(x_i^{n-1})\right) + H\beta_t^n$. To proceed we make the following restrictive assumptions that will nevertheless allow us to obtain useful order-of-magnitude estimates. Let us assume that both the observation errors R and the observed model errors HQH^T are uncorrelated, with variances V_y and V_β, respectively, to find:

$$-\log(w_i) = \frac{1}{2(V_\beta + V_y)}\sum_{j=1}^{M}\left[y_j - H_j x_t^n + H_j\beta_t^n + H_j\left(f(x_t^{n-1}) - f(x_i^{n-1})\right)\right]^2 \tag{52}$$

The variance of w_i arises from varying ensemble index i. Clearly the first three terms are given, and we introduce the constant $\gamma_j = y_j^n - H_j x_t^n + H_j\beta_t^n$. To proceed with our order of magnitude estimate we assume that the model can be linearised as $F(x_i^{n-1}) \approx Ax_i^{n-1}$., leading to:

$$-\log(w_i) = \frac{1}{2(V_\beta + V_y)}\sum_{j=1}^{M}\left[\gamma_j + H_j A(x_t^{n-1} - x_i^{n-1})\right]^2 \tag{53}$$

A following step in our order of magnitude estimate is to assume $x_t^{n-1} - x_i^{n-1}$ to be Gaussian distributed. In that case the expression above is non-central χ_M^2 distributed apart from a constant. This constant comes from the variance of $\gamma_j + H_j A(x_t^{n-1} - x_i^{n-1})$, which is equal to $H_j A P^{n-1} A^T H_j^T = V_x$, in which P^{n-1} is the covariance of the model state at time $n-1$. Hence we find:

$$-\log(w_i) = \frac{V_x}{2(V_\beta + V_y)}\sum_{j=1}^{M}\frac{\left[\gamma_j + H_j A(x_t^{n-1} - x_i^{n-1})\right]^2}{V_x} \tag{54}$$

Apart from the constant in front the expression above is non-central χ_M^2 distributed with variance $a^2 2(M + 2\lambda)$ where $a = V_x/(2(V_\beta + V_y)$ and $\lambda = (\sum_j \gamma_j^2)/V_x$.

We can estimate λ by realising that for a large enough number of observations we expect $\sum_j (y_j^n - H_j x_t^n)^2 \approx MV_y$, and $\sum_j (y_j^n - H_j x_t^n) \approx 0$. Furthermore, when the dimension of the system under study is large we expect $\sum_j (H_j\beta_t^n)^2 \approx MV_\beta$. Combining all these estimates we find that the variance of $-\log(w_i)$ can be estimated as

$$\frac{M}{2}\left(\frac{V_x}{V_\beta + V_y}\right)^2\left(1 + 2\left(\frac{V_\beta + V_y}{V_x}\right)\right) \tag{55}$$

This expression shows that the only way to keep the variance of $-\log(w_i)$ low when the number of independent observations M is large is to have a very small variance in the ensemble: $V_x \approx (V_\beta + V_y)/M$.

Clearly, in when the number of observations is large (10 million in typical meteorological applications), this is not very realistic. This expression has been tested in several applications and holds within a factor 0.1 in all tests (Ades and Van Leeuwen, 2013).

It should be mentioned that a large variance of $-\log(w_i)$ does not necessarily mean that the weights will be degenerate because the large variance could be due to a few outliers. However, we have shown that $-\log(w_i)$ is approximately non-central χ^2_M distributed for a linear model, so the large variance is not due to outliers but intrinsic in the sampling from such a distribution. Furthermore, there is no reason to assume that this variance will behave better for nonlinear models, especially because we didn't make any assumptions on the divergent or contracting characteristics of the linear model.

From this analysis we learn two things: it is the number of independent observations that determines the degeneracy of the filter, and the optimal proposal density cannot be used in systems with a very large number of independent accurate observations, as confirmed later by Rebeschini and Van Handel (2015).

Recently, an elegant proof that does not make any of the assumptions above on the order of magnitude of the estimates has been provided by Snyder et al. (2015). The importance of this article is that they show that for this class of particle filters the optimal proposal density does have least variance in the weights, so is indeed optimal in this sense. As will be discussed in a later section, fortunately the class of particle filters can be extended such that the curse of dimensionality does not apply.

5.5 The Implicit Particle Filter

In 2009 Chorin and Tu introduced the implicit particle filter. Although the paper is not very clear, the theory and the application are intertwined, discussions with them and later papers (Chorin et al., 2010; Morzfield et al., 2012) explain the method in more detail. Although they do not formulate their method in terms of a proposal density, to clarify the relation with the other particle filters this is the way it is presented here. In fact, as we shall see, it is closely related to the 'optimal proposal density' discuss before when the observations are available at every time step.

The proposed samples are produced as follows. Assume we have m model time steps between observations. Draw a random vector ξ_i of length the size of the state vector times m. each element of ξ_i is drawn from $N(0,1)$. The actual samples are now constructed by solving

$$-\log(p(y^n|x^n)p(x^{n-m+1:n}|x_i^{n-m})) = \frac{\xi_i^T\xi_i}{2} + \phi_i \tag{56}$$

for each particle x_i. The term ϕ_i is included to ensure that the equation above has a solution, so $\phi_i \geq \min(-\log(p(y^n|x^n)p(x^{n-m+1:n}|x_i^{n-m})))$. Note that this can be written as

$$p(y^n|x^n)p(x^{n-m+1:n}|x_i^{n-m}) = A\exp\left(-\frac{\xi^T\xi}{2} - \phi_i\right) \tag{57}$$

for later reference. One can view this step as drawing from the proposal density $q_x(x^{n-m+1:n}|x_i^{n-m}, y^n)$ via the proposal density $q_\xi(\xi)$, where we introduced the subscript to clarify the shape of the pdf. These two are related by a transformation of the probability densities as

$$q_x(x^{n-m+1:n}|x_i^{n-m}, y^n)dx^{n-m:n} = q_\xi(\xi)d\xi \tag{58}$$

so that

$$q_x(x^{n-m:n}|x_i^{n-m}, y^n) = q_\xi(\xi)J \tag{59}$$

in which J is the Jacobian of the transformation $\xi \rightarrow x$. We can now write the weights of this scheme as:

$$\begin{aligned} w_i &= p(y^n|x_i^n)\frac{p(x_i^{n-m+1:n}|x_i^{n-m})}{q_x(x_i^{n-m+1:n}|x_i^{n-m}, y^n)} \\ &= p(y^n|x_i^n)\frac{p(x^{n-m+1:n}|x_i^{n-m})}{Jq_\xi(\xi_i)} \end{aligned} \tag{60}$$

Using (57) we find that the weights are given by:

$$w_i = A\frac{\exp(-\phi)}{J} \tag{61}$$

To understand better how this works lets consider the case of observations every model time step, and Gaussian observation errors, Gaussian model equation errors, and linear observation operator H. In that case we have

$$\begin{aligned} -\log(p(y^n|x^n)p(x^{n-m+1}|x_i^{n-m})) &= \frac{1}{2}\left(y^n - Hx_i^n)\right)^T R^{-1}\left(y^n - Hx_i^n\right) \\ &+ \frac{1}{2}\left(x^n - f(x_i^{n-1})\right)^T Q^{-1}\left(y^n - f(x_i^{n-1})\right) \\ &= \frac{1}{2}\left(x^n - \hat{x}_i^n)\right)^T P^{-1}\left(x^n - \hat{x}_i^n\right) + \phi_i \end{aligned} \tag{62}$$

in which $\hat{x}_i^n = f(x_i^{n-1}) + K(y^n - Hf(x_i^{n-1}))$, the maximum of the posterior pdf, and $P = (1-KH)Q$, with $K = QH^T(HQH^T + R)^{-1}$. Comparing this with (57) we find $x^n = P^{1/2}\xi$, so J is a constant, and

$$\begin{aligned} \phi_i &= \min(-\log(p(y^n|x^n)p(x^{n-m+1:n}|x_i^{n-m}))) \\ &= \frac{1}{2}\left(y^n - Hf(x_i^{n-1})\right)^T (HQH^T + R)^{-1}\left(y^n - Hf(x_i^{n-1})\right) \end{aligned} \tag{63}$$

and finally $w_i \propto \exp(-\phi_i)$. Comparing with the optimal proposal density we see that when observations are present at every time step the implicit particle filter is equal to the optimal proposal density, with the same degeneracy problem.

5.6 The Equivalent-Weights Particle Filter

At the beginning of this chapter we discussed a very simple nudging scheme to pull the particles towards future observations, written as:

$$x_i^j = f(x_i^{j-1}) + \beta_i^j + K^j(y^n - H(x_i^{j-1}) \tag{64}$$

in which n is the next observation time. Unfortunately, exploring the proposal density by simply nudging will not avoid degeneracy in high-dimensional systems with a large number of observations. Also more complicated scheme's, such as running a 4DVar on each particle, which is essentially what Chorin et al. (2009) propose, is likely to lead to strongly varying weights for the particles because its close relation to the optimal proposal density. (However, it must be said that no rigourous proof exist of this statement!!!) We can expect to have to do something more optimal.

To start, let us recall the expression we found for the proposal density weights in the section on proposal densities:

$$w_i = p(y^n|x_i^n)\prod_{j=n-m+1}^{n}\frac{p(x_i^j|x_i^{j-1})}{q(x_i^j|x_i^{j-1}, y^n)} \tag{65}$$

We will use a modified model as explained in section 5.1 for all but the last model time step. The last time step will be different, as it explores a new class of particle filters to avoid filter degeneracy. The essential ingredient is that we write the proposal density of the last time step before observations as:

$$q(x^n|x_{1:N}^{n-1}, y^n) \tag{66}$$

in which $x_{1:N}^{n-1} = (x_1^{n-1}, ..., x_N^{n-1})$, so we allow a stronger dependence of each new particle on all previous particles. This dependency is typically present due to the resampling step, but here we allow for a stronger dependency. This is allowed as we can write:

$$p(x^n) = \int p(x^n|x^{n-1})p(x^{n-1})\,dx^{n-1} \approx \sum_{i=1}^{N} w_i^{n-1}\frac{p(x_i^n|x_i^{n-1})}{q(x_i^n|x_{1:N}^{n-1}, y^n)}q(x_i^n|x_{1:N}^{n-1}, y^n) \tag{67}$$

in which we assumed an ensemble x_i^{n-1} with weights w_i^{n-1} as representation of $p(x^{n-1})$.

This freedom to make the proposal density dependent on all previous particles allows us to develop particle filters that are not degenerate by construction. The following gives an example of such an algorithm. A more recent example can be found in Zhu et al. (2016).

In this example, the last time step consists of two stages: first perform a deterministic time step with each particle that ensures that most of the particles have equal weight, and then add a very small

random step to ensure that Bayes theorem is satisfied, see Van Leeuwen (2010, 2011) for details. There are again infinitely many ways to do this. For the first stage we write down the weight for each particle using only a deterministic move, so ignoring the proposal density q for the moment:

$$-\log w_i = -\log w_i^{n-1} + \frac{1}{2}(y^n - Hx_i^n)^T R^{-1}(y^n - Hx_i^n) + \frac{1}{2}(x_i^n - f(x_i^{n-1}))^T Q^{-1}(x_i^n - f(x_i^{n-1})) \quad (68)$$

in which w_i^{n-1} is the weight accumulated over the previous time steps between observations, so the p/q factors from each time step. If H is linear, which is not essential but as we will assume for simplicity here, this is a quadratic equation in the unknown x_i^n. All other quantities are given. We calculate the minimum of this function for each particle i, which is simply given by

$$-\log w_i = -\log w_i^{n-1} + \frac{1}{2}\left(y^n - Hf(x_i^{n-1})\right)^T (HQH^T + R)^{-1}\left(y^n - Hf(x_i^{n-1})\right) \quad (69)$$

For N particles this given rise to N minima. Next, we determine a target weight as the weight that 80% of the particles can reach, i.e. 80% of the minimum $-\log w_i$ is smaller than the target value. (Note that we choose another percentage, see e.g. Ades and Van Leeuwen (2013), who investigate the sensitivity of the filter for values between 70% and 100%.) Define a quantity $C = -\log w_{target}$, and solve for each particle with a minimum weight larger than the target weight

$$C = -\log w_i^{n-1} + \frac{1}{2}\left(y^n - Hf(x_i^{n-1})\right)^T (HQH^T + R)^{-1}\left(y^n - Hf(x_i^{n-1})\right) \quad (70)$$

So now we have found the positions of the new particles x_i^n such that all have equal weight. The particles that have a larger minimum than C will come back into the ensemble via a resampling step, to be discussed later.

The equation above has an infinite number of solutions for dimensions larger than 1. To make a choice we assume

$$x_i' = f(x_i^{n-1}) + \alpha_i K[y^n - Hf(x_i^{n-1})] \quad (71)$$

in which $K = QH^T(HQH^T + R)^{-1}$, Q is the error covariance of the model errors, and R is the error covariance of the observations. Clearly, if $\alpha = 1$ we find the minimum back. We choose the scalar α_i such that the weights are equal, leading to

$$\alpha_i = 1 - \sqrt{1 - b_i/a_i} \quad (72)$$

in which $a_i = 0.5x_i^T R^{-1} HKx$ and $b_i = 0.5x_i^T R^{-1} x_i - C - \log w_i^{n-1}$. Here $x = y^n - Hf(x_i^{n-1})$, C is the chosen target weight level, and w_i^{rest} denotes the relative weights of each particle i up to this time step, related to the proposal density explained above.

Of course, this last step towards the observations cannot be fully deterministic. A deterministic proposal would mean that the proposal transition density q can be zero while the target transition density p is non zero, leading to division by zero, because for a deterministic move the transition density is a delta function. The proposal transition density could be chosen a Gaussian, but since the weights have q in the denominator a draw from the tail of a Gaussian would lead to a very high weight for a particle that is perturbed by a relatively large amount. To avoid this q is chosen in the last step before the observations as a mixture density

$$q(x_i^n | x_i') = (1 - \gamma)U(-a, a) + \gamma N(0, a^2) \quad (73)$$

in which x_i' the particle before the last random step, and γ and a are small. By choosing γ small the change of having to choose from $N(0, a^2)$ can be made as small as desired. For instance, it can be made dependent on the number of particles N.

To conclude, the almost-equal-weight scheme consists of the following steps:

1 Use the modified model equations for each particle for all time steps between observations.

2 Calculate, for each particle i for each of these time steps

$$w_i^j = w_i^{j-1} \frac{p(x_i^j | x_i^{j-1})}{q(x_i^j | x_i^{j-1}, y^n)} \quad (74)$$

3 At the last time step before the observations calculate the maximum weights for each particles and determine $C = -\log w_{target}$

4 Determine the deterministic moves by solving for α_i for each particle as outlined above.

5 Choose a random move for each particle from the proposal density (73).

6 Add these random move to each deterministic move, and calculate the full posterior weight.

7 Resample, and include the particles that have been neglected from step 4 on.

Finally, it is stressed again that we do solve the fully nonlinear data assimilation problem with this efficient particle filter, and the only approximation is in the ensemble size. All other steps are completely compensated for in Bayes Theorem via the proposal density freedom.

5.7 Application to the barotropic vorticity equations

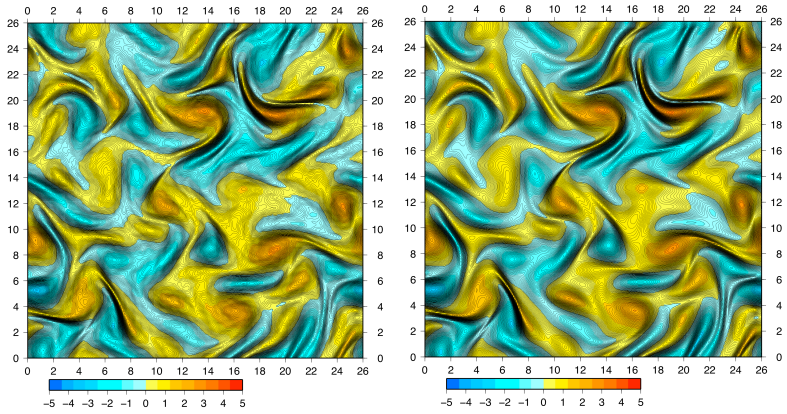

Figure 5: *Snap shot of the vorticity field of the truth (right) and the particle filter mean (left) at time 25. Note the highly chaotic state of the fields, and the close to perfect tracking.*

Here a few results using the new particle filter with almost equal weights are shown, see Van Leeuwen and Ades (2012). Figure 5 shows the application of the method to the highly chaotic barotropic vorticity equation, governed by:

$$\begin{aligned} \frac{\partial q}{\partial t} - \frac{\partial \psi}{\partial y}\frac{\partial q}{\partial x} + \frac{\partial \psi}{\partial x}\frac{\partial q}{\partial y} &= \beta \\ q = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} & \end{aligned} \tag{75}$$

in which q is the vorticity field, ψ is the streamfunction, and β is a random noise term representing errors in the model equations. It was chosen from a multivariate Gaussian with mean zero, variance 0.01, and decorrelation lengthscale 4 gridpoints. The equations are implemented on a 256 X 256 grid, using a semi-Lagrangian scheme with time step $\Delta t = 0.04$, grid spacing $\Delta x = \Delta y = 1/256$, leading to a state dimension of close to 65,000. The vorticity field was observed every 50 time steps on every gridpoint. The decorrelation time scale of this system is about 25 time steps, so, even though the full state is observed, this is a very hard highly nonlinear data-assimilation problem. The observations were

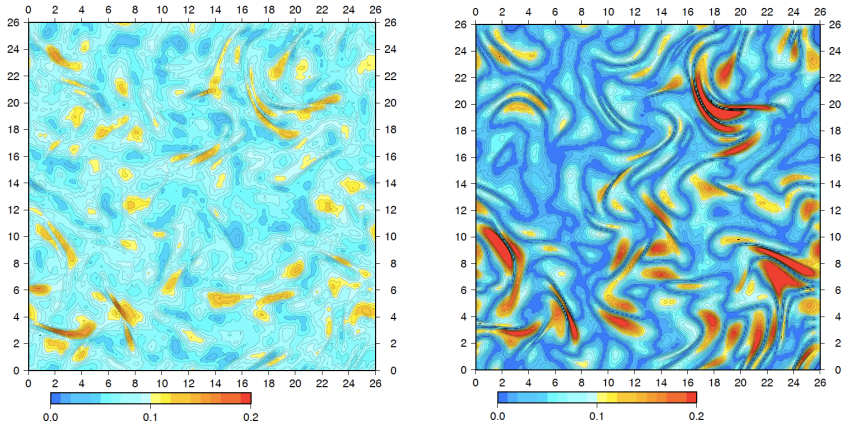

Figure 6: *Snap shot of the absolute value of the mean-truth misfit and the standard deviation in the ensemble. The ensemble underestimates the spread at several locations, but averaged over the field it is slightly higher, 0.074 versus 0.056.*

obtained from a truth run and independent random measurement noise with standard deviation 0.05 was added to each observation.

Only 24(!) particles were used to track the posterior pdf. In the application of the new particle filter we chose $K = 0.1$ in the nudging term (except for the last time step before the new observations, where the 'almost equal weight' scheme was used, as explained above), multiplied by a linear function that is zero half way the two updates and growing to one at the new observation time. The random forcing was the same as in the original model. This allows the ensemble to spread out due to the random forcing, and pulling harder and harder towards the new observation the closer to the new update time.

Figure 6 shows the difference between the mean and the truth after 50 time steps and the ensemble standard deviation compared to the absolute value of the mean-truth misfit. Clearly, the truth is well represented by the mean of the ensemble. Figure 3 shows that although the spread around the truth is underestimated at several locations, it is over estimated elsewhere,

Finally, figure 7 shows that the weights are distributed as they should: they display small variance around the equal weight value 1/24 for the 24 particles. Note that the particles with zero weight had too small weight to be included in the almost equal weight scheme, and will be resampled from the rest.

Because the weights vary so little the weights can be used back in time, generating a smoother solution for this high-dimensional problem with only 24 particles.

6 Conclusions

To try to solve strongly nonlinear data-assimilation problems we discussed Particle filters in this chapter. While they have a few strong assets, i.e. their full nonlinearity, the simplicity of their implementation (although this tends to be lost in more advanced variants), the fact that balances are automatically fulfilled (although, again, more advanced methods might break this), and, quite importantly, that their behaviour does not depend on a correct specification of the model state covariance matrix.

We have also seen the weaknesses in terms of efficiency, the filter degeneracy problem, that plagues the simpler implementations. However, recent progress seems to suggest that we are quite close to

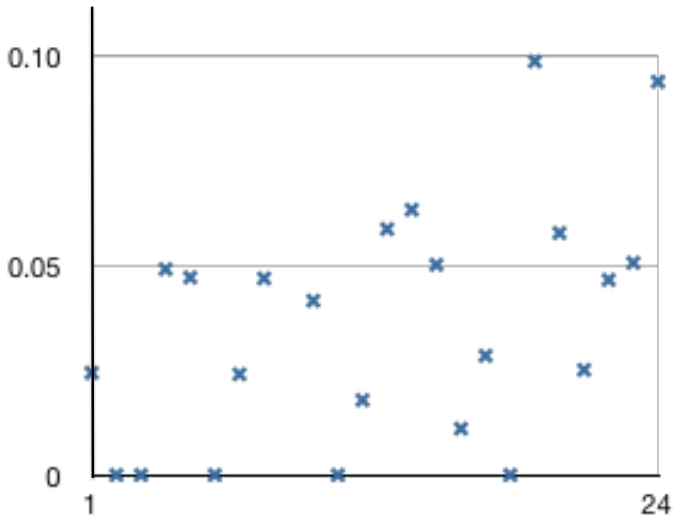

Figure 7: *Weights distribution of the particles before resampling. All weights cluster around 0.05, which is close to 1/24 for uniform weights (using 24 particles). The 5 particles with weights zero will be resampled. Note that the other particles form the smoother estimate.*

solving this problem with developments like the Implicit Particle Filter and the Equivalent-Weights Particle Filter. Also, the approximations are becoming more advanced too, and perhaps we don't need a fully nonlinear data assimilation method for real applications.

There is a wealth of new approximate particle filters that typically shift between a full particle filter and an ensemble Kalman filter, depending on the degeneracy encountered. Especially Gaussian mixture models for the prior are popular. I have refrained from trying to give an overview here, there is just too much going on in this area. A brief discussion is given in Van Leeuwen (2009), again not completely up to date. In a few years time we will have learned what is useful and what is not.

Specifically I'd like to mention the he Rank Histogram Filter of Anderson (2010). It approximates the prior ensemble in observation space with a histogram, assuming Gaussian tails at both end members. It then performs Bayes theorem and multiplies this prior with the likelihood to form the posterior. Samples from this posterior are generated as follows. First, the cumulative probability of the posterior at each prior particle is calculated by integrating the posterior over the regions between the prior particles. We want the posterior particles to have equal probability $1/(N+1)$, so cumulative probability $n/(N+1)$ for ordered particle n. So the position of each new particle is found by integrating the posterior pdf $1/(N+1)$ further from the previous new member. As Anderson shows, this entails solving a simple quadratic equation for each particle, with special treatment of the tails.

A few comments are in place. First, the prior is not assumed to the Gaussian, and also the likelihood can be highly non-Gaussian,which is good. However, a potential problem is that the procedure above is performed on each dimension separately, and it is unclear how to combine these dimensions into sensible particles. Localisation has to be applied to keep the numerical calculations manageable, and inflation is also needed to avoid ensemble collapse. Also, as far as I can see, when the observations are correlated the operations explained above have to be done in a higher dimensional space, making the method more complicated. Finally, the method interpolates in state space, which potentially leads to unbalanced states. Anderson applied the method to an 28,000 dimensional atmospheric model with very promising results.

A word of caution is needed. The contents of this chapter expresses my present knowledge of the field, and no doubt misses important contributions. It is quite heavily biased towards geophysical applications, and the mathematical treatment is consequently poor. A few references have been provided

to partly correct this bias. Also, the field is developing so rapidly that it is becoming extremely hard to keep track of all interesting work.

Finally, it must be said that the methods discussed above have a strong bias to state estimation. One could argue that this is fine for prediction purposes, but for model improvement (and thus indirectly forecasting) parameter estimation is of more interest (and what about parameterisation estimation...). Unfortunately no efficient Particle Filter schemes exist for that problem. This is a growing field that needs much more work, both from mathematicians and from practitioners.

References

Ades, M., and P. J. van Leeuwen. An exploration of the equivalent weights particle filter. *Quart. J. Roy. Meteorol. Soc.*, **139**, 820840, 2013.

Anderson, J.L., and S.L. Anderson, A Monte-Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts, *Monthly Weather Rev.*, **127**, 2741-2758, 1999.

Bain, A. and D Crisan, *Fundamentals of Stochastic Filtering*, Springer, doi: 10.1007/978-0-387-76896-0, 2009.

Bengtsson, T., C. Snyder, and D. Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. J. Geophys. Res., **108**, 87758785, 2003.

Berliner, L.M., and C.K. Wikle, Approximate importance sampling Monte Carlo for data assimilation, *Physica D*, **230**, 37-49, 2007.

Beskos A, Crisan D, Jasra A, On the stability of sequential Monte Carlo methods in high dimensions, *Ann. of Appl. Probab.*, **24**, 1396-1445, 2014.

Chorin A.J., X. Tu Implicit sampling for particle filters. *PNAS* **106**, 17249-17254. 2009.

Del Moral, P. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications.*, Springer, Berlin, 2004.

Del Moral , P., and A. Guionnet, On the stability of interacting processes with applications to filtering and generic algorithms, *Ann. Inst. H. Poincar, Probab. et Stat.* **37**,155194, 2001.

Doucet, A., N. De Freitas, and N. Gordon, *Sequential Monte-Carlo methods in practice*, Springer, Berlin, 2001.

Eyink, G., and S. Kim, A maximum entropy method for particle filtering, *J. Stat. Phys.*, **123**, 1071-1128, doi: 10,1007/s10955-006-9124-9, 2006.

Gordon,N.J., D.J. Salmond, and A.F.M. Smith., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings-F*, **140**, 107-113, 1993.

Kim, S., G.L. Eyink, J.M. Restrepo, F.J. Alexander, and G. Johnson, Ensemble filtering for nonlinear dynamics, *Monthly Weather Rev.*, **131**, 2586-2594, 2003.

Le Gland, F., V. Monbet, V-D. Tran, Large sample asymptotics for the ensemble Kalman Filter, *The Oxford Handbook of Nonlinear Filtering*, 598-631, 2011.

Nakano, S., G. Ueno, and T. Higuchi, Merging particle filter for sequential data assimilation, *Nonlin. processes Geophys.*, **14**, 395-408, 2007.

Penny, S.G., and T. Miyoshi. A local particle filter for high dimensional geophysical systems. *Nonlin. Processes Geophys. Discuss.*, 2015. 10.5194/npgd-2-1631-2015.

Pitt, M.K., and N. Shephard, Filtering via simulation: Auxilary particle filters, *J. American Stat. Ass.*, **94**, 590-599, 1999.

Poterjoy, J., A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Rev.*, 10.1175/MWR-D-15-0163.1, 2016.

Rebeschini P. and R. van Handel, Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.*, **25**, 2809-2866, 2015.

Reich, S., and C. Cotton. *Probabilistic Forecasting and Bayesian Data Assimilation.* Cambridge University Press, 2015.

Snyder,C., T. Bengtsson, P. Bickel, and J. Anderson "Obstacles to high-dimensional particle filtering", *Mon. Wea. Rev.*, **136**, 4629-4640, 2008.

Snyder, C., T. Bengtsson, and M. Morzfeld. Performance bounds for particle filters using the optimal proposal. *Mon. Wea. Rev.*, 10.1175/MWR-D-15-0144.1, 2015.

Spiller, E.T., A. Budhiraja, K. Ide, and C.K.R.T Jones, Modified particle filter methods for assimilating Lagrangian data into a point-vortex model, *Physica D*, **237**, 1498-1506, 2008.

Tong, X.T. and R. Van Handel, Ergodicity and stability of the conditional distributions of nondegenerate Markov chains. *Ann. Appl. Probab.*, **22**, 1495-1540, 2012.
Van Handel, R., When do nonlinear filters achieve maximal accuracy? *SIAM J. Control Optim.*, **48**, 3151-3168, 2009.
Van Leeuwen, P.J.,Ensemble Kalman Filters, Sequential Importance Resampling and beyond, *ECMWF workshop on the role of the upper ocean in medium and extended range forecasting, 13-15 November 2002.*, ECMWF, Reading, UK, 46-56, 2002.
Van Leeuwen, P.J. Nonlinear ensemble data assimilation for the ocean. In *Recent Developments in data assimilation for atmosphere and ocean*, pages 265286. ECMWF Seminar Series, 2003.
Van Leeuwen, P.J. Particle Filtering in Geosciences, *Monthly Weather Rev.*, **137**, 4089-4114, 2009.
Van Leeuwen, P.J. , Nonlinear Data Assimilation in geosciences: an extremely efficient particle filter, *Quart. J. Roy. Meteor. Soc..*, **136**, 1991-1996, 2010.
Van Leeuwen, P.J: Efficient non-linear Data Assimilation in Geophysical Fluid Dynamics *Computers and Fluids*, doi:10,1016/j.compfluid.2010.11.011, 2011.
Xiong, X, I.M. Navon, and B. Uzunoglu, A note on the particle filter with posterior Gaussian resampling, *Tellus*, **58A**, 456-460, 2006.
Zhu, M., P. J. van Leeuwen, and J. Amezcua. Implicit equal-weights particle filter. *Quart. J. Roy. Meteorol. Soc.*, doi:10.1002/qj.2784, 2016.