SPECIAL
Mathematical Advances in Data Assimilation
COLLECTION

REVIEW

Particle Filtering in Geophysical Systems

PETER JAN VAN LEEUWEN

Institute for Marine and Atmospheric Research Utrecht, Utrecht University, Utrecht, Netherlands, and Department of Meteorology, University of Reading, Reading, United Kingdom

(Manuscript received 23 October 2008, in final form 1 June 2009)

ABSTRACT

The application of particle filters in geophysical systems is reviewed. Some background on Bayesian filtering is provided, and the existing methods are discussed. The emphasis is on the methodology, and not so much on the applications themselves. It is shown that direct application of the basic particle filter (i.e., importance sampling using the prior as the importance density) does not work in high-dimensional systems, but several variants are shown to have potential. Approximations to the full problem that try to keep some aspects of the particle filter beyond the Gaussian approximation are also presented and discussed.

1. Introduction

From a data-assimilation perspective, geophysical systems such as the atmosphere or the oceans are characterized by a large state space, typically on the order of 1 million or (much) more, and by spatial relations between the variables related to the partial differential equations that describe the system. These systems are nonlinear, especially in high-resolution applications. The nonlinearity arises from both the dynamics, in which the nonlinear advection terms can become of dominant order, and from diabatic processes such as deep convection, cloud formation, and precipitation. In this paper we concentrate on physical models of the atmosphere and oceans, although the ideas developed are also of interest to other geophysical systems such as hydrology and atmospheric chemistry. With ever-increasing resolution, the nonlinearities are likely to become so strong that data-assimilation methods based on linearizations, such as the Kalman filter and its ensemble versions, as well as gradient descent methods like four-dimensional variational data assimilation (4D-Var) might fail in one way or another.

In the statistical literature several methods have been proposed to deal with strongly nonlinear systems. Unfortunately, these methods have been developed with the focus primarily upon low-dimensional systems, typically of state dimension 10 or less. The huge dimensions used in geophysical models thus prohibit direct application. These huge dimensions do not pose immediate problems themselves. One can argue that the actual degrees of freedom of these models is far less, so pointing to the dimension of the state vector is misleading. This boils down to the notion of a strange attractor in state space from which the actual model state is unlikely to deviate much. This should be kept in mind when the size of the problem is discussed in the rest of this paper. Unfortunately, we have little notion of what the relevant dimension of our geophysical model is. But even if it is only 1% of the dimension of the state vector, it is still very large, much larger than that used in the statistical literature.

A further complication is that the geophysical fields contain certain spatial relations, loosely called ‘‘balances,’’ which should be preserved to a certain extent in the data-assimilation system. The formulation in the previous sentence is rather vague, since we have no clear idea what the balances are in nonlinear geophysical systems. But we do know from experience that certain balances should be preserved (e.g., in atmospheric data assimilation the data-assimilation step should not introduce strong gravity waves since these tend to ruin the weather forecast). Up until recently this problem was solved by projection of the analyzed field on the

Corresponding author address: Peter Jan van Leeuwen, Department of Meteorology, University of Reading, Earley Gate, P.O. Box 243, Reading RG6 6BB, United Kingdom.
E-mail: p.j.vanleeuwen@reading.ac.uk

DOI: 10.1175/2009MWR2835.1

so-called slow manifold, mainly related to geostrophy. However, as mentioned above, at higher resolution the coherent nonlinear structures follow other balances, which are not well known. In fact, the gravity waves become an important part of the correct solution, and should be correctly initialized, not necessarily to zero.

Although direct application of the methods developed in the statistical literature is not possible, the general idea is that it is possible to adapt some of these methods to geophysical applications. This paper gives an overview of attempts to do so, and ideas that could be fruitfully followed. Of special interest are Monte Carlo methods because their convergence rate is independent of the dimension of the state vector. Of these, the so-called particle filters, which rely on a representation of the model probability density function (pdf) by a discrete set of model states, seem to be quite promising. In the context of data assimilation these model states are referred to as ''ensemble members'' or ''particles.'' We will call them particles in the following. The evolution of the model probability density function is simulated by propagating all particles (model states) forward in time by the nonlinear model equations. When observations of the true geophysical system become available, the particles are changed such that the information present in the observations is incorporated into the swarm of particles. The emphasis of this paper is on these particles filters.

The statistical literature suggests that particle filters based on improvements of basic importance sampling might have potential for the large-scale systems we are interested in. The restriction to these particle filters stems from the size of the ensemble of particles. Typically, given the capabilities of present-day supercomputers and the expected resolution needed to describe the processes of interest, on the order of 50 to maybe 1000 particles can be afforded. Because of this small size, we cannot afford to completely reject many particles, which will happen quite frequently because the large-dimensional state spaces tend to be very empty in terms of probability. For this reason the accept–reject algorithm, and variants thereof, are not considered here.

Another class of Monte Carlo methods is based on the Metropolis–Hastings algorithm (Hastings 1970), with the Gibbs sampler as a simpler version of that algorithm. These methods have been shown to be effective on highly nonlinear small-scale problems, but typically need a large number of samples. The ability of these methods to solve the geophysical data-assimilation problem is still unclear. For instance, Alexander et al. (2005) use ideas from statistical mechanics, in which the state at a certain time is considered as a point in state space. The sequence of points that describes model integration (i.e., a model trajectory), is viewed as a generalized coordinate, to which a generalized momentum is assigned. To the (weak constraint) cost function that is to be minimized in variational problems one adds a ''kinetic energy'' term quadratic in these momenta and a Hamilton dynamical system is obtained. The Hamilton equations give a model trajectory and the momentum of that model trajectory in state space. Starting from a certain initial model trajectory in state space, a sequence of model trajectories is generated by the Hamilton dynamics. This sequence of model trajectories is not independent, like in a particle filter, so more model trajectories than actually needed are generated to find N independent trajectories that describe the posterior pdf. Alexander et al. (2005) describe generalizations to this Hamiltonian Monte Carlo technique that shorten the decorrelation time (i.e., ensure that the model trajectories are independent with less Hamiltonian dynamics steps). The authors mention that fairly large space–time spaces can be handled: in quantum chromodynamics up to 10^7 space–time dimensions. Note that this equals about 100 time steps for a model with a fairly small number—10^5—of variables, and that these methods are in fact smoothers instead of filters. For these reasons we will not discuss them further here.

Several other reviews on particle filtering exist in the literature. For instance, Doucet et al. (2001) present an excellent overview of particle filter theory including several examples from the statistical literature. Wikle and Berliner (2007) give a short introduction, including several analytical examples, and is a good introduction to the field.

Before we start off, a word of caution. This paper is intended be a review of what has been done in geophysical applications of particle filters, and of ideas that have come forward from the statistical literature. I have encountered several articles in which it has becomes clear that the authors, including myself, were not aware of relevant previous literature on the subject. There is no doubt that this review will also miss important contributions.

The paper is organized as follows. The next chapter introduces the basic idea of particle filtering. The idea of Monte Carlo simulation and Bayes's theorem to incorporate observations into the model system is outlined, and its problems are identified. These are in fact twofold. Section 3 describes methods to solve the problem of ensemble degeneracy over time due to repeated application of the algorithm in the assimilation cycle. The second problem, related to the fact that the majority of particles are too far from the observations to properly represent the actual state of the geophysical system, is discussed in section 4, together with possible solutions. Section 5 discusses approximations to the general framework. It is here that the boundary between particle filtering and other solution methods for data-assimilation problems becomes particularly vague. The rather loose definition of particle filtering used here is that particles are used to evolve the pdf forward in time between observations, but excludes ensemble Kalman filters (EnKF), which approximate the prior probability density function by a Gaussian. (Actually, from a maximum entropy point of view, one assumes that one can only estimate the mean and the covariance with any confidence from the ensemble of particles. The maximum entropy formalism leads directly to a Gaussian density in that case.) Section 6 summarizes the results and tries to draw a few general conclusions.

2. The basic particle filter

When the system under study is nonlinear a statistical description in terms of errors in variables of the system becomes cumbersome, and a probabilistic description in terms of stochastic differential equations is more natural. Each variable in the geophysical model is described by a probability density function, and the full state ψ is described by the joint probability density of all variables, denoted by $p(\psi)$. Given this probability density, we want to combine it with observations when they become available, to obtain the probability density of the model given these new observations. To this end we use Bayes's theorem.

a. Bayes's theorem

Bayes's theorem is based on conditional probability densities that can be introduced in the following way. The joint probability of two variables can be written as the probability (Pr) of one of the variables given the other (i.e., the one conditioned on the other), times the probability of the other:

$$\Pr(a \in A, b \in B) = \Pr(a \in A|b \in B)\Pr(b \in B). \tag{1}$$

An example shows that this makes sense: if a is the number of participants of a workshop on data assimilation who are wearing glasses, and b is the number of women participating, then the joint probability (the number of women wearing glasses) is equal to the conditional probability (the number of participants that wear glasses given that they are female) multiplied by the fraction of women in the workshop. For a workshop with 30 participants, of which 2 of the (only…) 3 women wear glasses, we find

$$\frac{2}{30} = \frac{2}{3}\frac{3}{30}, \tag{2}$$

which is indeed an identity. Since

$$\begin{aligned}\Pr(a \in A, b \in B) &= \Pr(a \in A|b \in B)\Pr(b \in B)\\ &= \Pr(b \in B|a \in A)\Pr(a \in A)\end{aligned} \tag{3}$$

we find

$$\Pr(a \in A|b \in B) = \frac{\Pr(b \in B|a \in A)\ \Pr(a \in A)}{\Pr(b \in B)}, \tag{4}$$

which is Bayes's theorem. It is easy to show that when Pr is continuously differentiable Bayes's theorem also holds for probability densities (see e.g., Papoulis 1995). For the data assimilation case we find that the model state ψ given the observations d is

$$p_m(\psi|d) = \frac{p_d(d|\psi)p_m(\psi)}{p_d(d)}, \tag{5}$$

in which the subscript m denotes the probability density of the model, and the subscript d denotes the probability of the observations. From now on we will omit these subscripts and let the argument of the density refer to the domain and functional form of the density.

We thus find that the probability density of the model given the new observations is equal to that of the observations given the model times the probability density of the model prior to the observations being taken into account, divided by the probability density of the observations. Interestingly, the probability density of the model comes in differently from that of the observations. We have to know the probability density of the model prior to the observations, the so-called prior pdf. For the observations we know the observed values, and the functional form of their probability density. Each observation is picked randomly by our measurement from the probability density but we do not know the probability of that specific observation because we do not know where the pdf is located. Happily, we do not need to, because we can use $p(d|\psi)$; that is, the pdf of the observations given the model, as if the model gives the true observation. The observational pdf $p(d)$ in the denominator of (5) is written as

$$p(d) = \int p(d, \psi)\, d\psi = \int p(d|\psi)\,p(\psi)\, d\psi, \tag{6}$$

which shows that the denominator is just the normalization of the pdf of the model given the observations, the posterior pdf. If we use (6) in (5) it becomes apparent that the likelihood $p(d|\psi)$ does not have to be a pdf, in the sense that it does not have to integrate to 1.

The difficulty in applying Bayes's theorem is the probability density of the model. It is a density over a million (or more) dimensional space, which is impossible to store. This raises the question of what we want to know about this pdf. The mean? The mode (maximum)? The covariance? Higher-order moments? Or, if the pdf is multimodal, the positions of these modes?

The idea used in particle filtering is to try to represent the model pdf by a number of random draws, called ensemble members, or particles. After (or during) a previous data-assimilation step a new ensemble of particles is generated. Each particle (model state) is propagated forward in time with the full nonlinear model. This part of the data assimilation consists of trying to represent the forward evolution of the model pdf. Formally the evolution equation of this pdf is given by the Kolmogorov equation (see, e.g., Jazwinski 1970). This equation is solved approximately by solving an ensemble of stochastic partial differential equations. The stochastic terms in these equations represent unknown external and internal terms (or factors) in the model. Unknown terms in external forcing and in the model equations are incorporated by adding random numbers, drawn from a known error density, to the model equations. It is also possible to multiply parts of the model equations by unknown factors, sometimes called multiplicative errors. The latter approach is usually related to unknown model parameters. All particle methods have this forward propagation in common, and differ mainly in the analysis step (i.e., in the way model and observations are combined).

b. *Why use Monte Carlo (or particle) methods?*

We know that the most complete description of the system ψ under study (e.g., the atmosphere or the ocean) is given by its probability density $p(\psi)$, in which ψ is the state vector (i.e., the vector that contains all prognostic variables of the system). Suppose we want to calculate the mean of a function f of the state vector (e.g., the velocity at some point in space time). That mean is given by

$$I(f) = \overline{f(\psi)} = \int f(\psi)\, p(\psi)\, d\psi. \tag{7}$$

In the applications we have in mind, the dimension of the integrand can easily be 1 000 000 000 or more, so a very efficient integration method is needed. A gridpoint-based integration is out of the question because of this high dimension. One way to solve this problem is to introduce N independently drawn random samples, or particles, from the density $p(\psi)$. We denote these particles with subscript i, as ψ_i. The density $p(\psi)$ can be represented by these particles as

$$p(\psi) = \frac{1}{N}\sum_{i=1}^{N} \delta(\psi - \psi_i). \tag{8}$$

The mean of $f(\psi)$ can now be approximated as

$$I(f) \approx I_N(f) = \frac{1}{N}\sum_i f(\psi_i). \tag{9}$$

This estimate is unbiased, and the variance in the estimate due to the random character of the draws is equal to (see, e.g., Robert and Cassela 2004)

$$\mathrm{var}[I_N(f)] = \frac{\mathrm{var}_f}{N}, \tag{10}$$

where

$$\mathrm{var}_f = \frac{1}{N-1}\sum_{i=1}^{N} f^2(\psi_i) - I_N^2(f). \tag{11}$$

The interesting point for us is that the rate of convergence of the estimate $I_N(f)$ is independent of the dimension of the integrand. Furthermore, we just saw that the standard deviation of the estimate decreases as $1/\sqrt{N}$. In contrast, gridpoint-based integration has the property that the rate of convergence decreases as the dimension of the integrand increases.

c. *. . .and perhaps why not?*

Obviously, for a million-dimensional system, the number of particles should also be large. How large is not exactly clear. A rule of thumb is that this number should be equal to the degrees of freedom of the system. However, in these large-dimensional systems it is very hard to get an idea of the order of magnitude of the number of degrees of freedom. Furthermore, that number will no doubt depend on the actual state of the system. In practice, we cannot afford to run more than 50–1000 model states forward in time. Is particle filtering feasible at all? Experience shows that when assuming that the pdf is Gaussian, the so-called EnKF (and its variants) can, with a few tricks, give good results (e.g., in weather forecasting; see, e.g., Kalnay 2004; Evensen 2006). The hope is that, probably also with a few tricks, the non-Gaussian problem can be solved in this way too. This is still to be proven, and that problem is at the heart of this paper.

d. *The basic importance of sampling*

As mentioned in the introduction, almost all methods we will discuss here are based on importance sampling. The most straight-forward implementation is what we call basic importance sampling. (In the statistical literature one usually finds importance sampling described with a proposal density different from the prior model pdf; see section 4a. However, for pedagogical reasons we present importance sampling in the following way.) Basic importance sampling is a straightforward implementation of Bayes's theorem. Using the particle description of the prior pdf given by (8) in (5) gives the following:

$$p(\psi|d) = \sum_{i=1}^{N} w_i \delta(\psi - \psi_i), \tag{12}$$

where the weights w_i are given by

$$w_i = \frac{p(d|\psi_i)}{\sum_{j=1}^{N} p(d|\psi_j)}. \tag{13}$$

The density $p(d|\psi_i)$ is the probability density of the observations given the model state ψ_i, which is often taken as a Gaussian:

$$p(d|\psi_i) = A \exp\left\{-\frac{1}{2}[d - H(\psi_i)]^{\mathrm{T}} R^{-1} [d - H(\psi_i)]\right\}, \tag{14}$$

where $H(\psi_i)$ is the measurement operator, which is the model equivalent of the observation d, and R is the error covariance of the observations.

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 $f(\psi)$ we now have

$$\overline{f(\psi)} = \int f(\psi) p(\psi)\, d\psi \approx \sum_{i=1}^{N} w_i f(\psi_i). \tag{15}$$

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

The order of operations in importance sampling is as follows (see Fig. 1):

1) Sample N particles ψ_i from the initial model probability density $p(\psi^0)$, 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 $p(\psi^n|\psi_i^{n-1})$ for each i (i.e., for each particle ψ_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 (13) 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 1 and repeat steps 2 and 3 until all observations up to the present have been processed.

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.

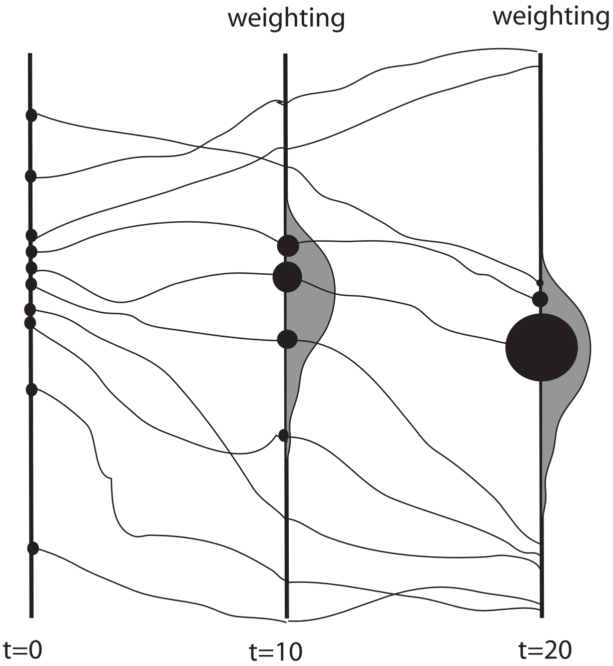

FIG. 1. The standard particle filter with importance sampling. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time 0. At time 10 the likelihood is displayed together with the new weights of each particle. At time 20 only 3 members have weights different from zero: the filter has become degenerate.

However, there is more. Even if the particles do follow the observations in time, the weights will still 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 Fig. 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 (re)invented, see later on. In a naive example, van Leeuwen and Evensen (1996) applied importance sampling in a two-layer quasigeostrophic model of the perturbed baroclinic jet with a state dimension of 6400. Unaware of the statistical literature, they called the method the direct ensemble method. The 500-member data-assimilation experiment was indeed divergent.

The method has been applied successfully in an atmospheric chemistry model to infer the trend in the global OH concentration by Krol et al. (1998). In a global atmospheric chemistry model with online fluid dynamics the global OH concentration, its trend, and the initial methylchloroform (MCF) concentration were estimated from MCF concentration observations in the period 1975–95. After 200 model integrations the statistics of mean and covariance converged to their final value. It should be mentioned that only three parameters were estimated, but the measurement operator (i.e., obtaining the model equivalent of the observations by running the full model) was rather complex. This is usually the case in parameter estimation. Vossepoel and van Leeuwen (2007) estimated the lateral mixing coefficients for temperature, salinity, and momentum in the global ocean general circulation model Océan Parallélisé (OPA) of 2° resolution with meridional refinement to 0.5° in the tropics. Observations were obtained from a model run with mixing coefficients derived from altimeter sea surface height variability observations. About 10 000 coefficients were estimated using 128 members. The method worked, but showed that more particles were needed for convergence. [Actually, to obtain good estimates the observations had to be assimilated locally (i.e., only observations within a 5° radius were taken into account for each grid point, reducing the number of observations per grid point to about 21). This is a form of localization, to be discussed later.]

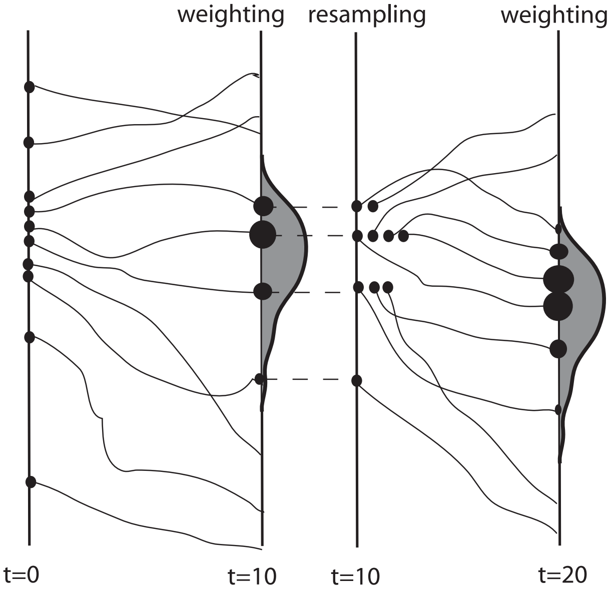

FIG. 2. The particle filter with resampling, also called SIR. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal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

3. Reducing the variance in the weights

Several methods exists to reduce the variance in the weights (see, e.g., Doucet et al. 2001). We discuss here sequential importance resampling, the marginal particle filter and hierarchical models, because these are among the few that can be applied directly to large-dimensional problems. The first two methods ''break with the past'' in that they get rid of the weights of the particles accumulated during previous assimilation steps. In resampling methods the posterior ensemble is resampled so that the weights become equal. In the marginal particle filter the past is integrated out. Both methods do not change the position of the particles in state space. In the next section methods are discussed that do change the positions of the prior particles in state space to improve the likelihood of the particles. In hierarchical models one tries to break up the full assimilation problem in a sequence of easier to solve smaller assimilation problems, using the concept of conditional probability densities.

a. 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. Although the idea is old (Metropolis and Ulam 1944), it was reintroduced in the statistical literature by Gordon et al. (1993). To restore the total number of particles N, identical copies of high-weight particles are formed. The higher the weight of a particle is, the more copies are generated, such that the total number of particles becomes N again. Sequential importance resampling (SIR) does the above and makes sure that the weights of all posterior particles are equal again, to $1/N$. Several resampling algorithms exist of which we discuss four. The last one is a special application of Metropolis–Hastings, which uses a chainlike procedure to resample the particles.

SIR is identical to basic importance sampling but for a resampling step after the calculation of the weights. The ''flowchart'' reads (see Fig. 2) as

1) Sample N particles ψ_i from the initial model probability density $p(\psi^0)$.
2) Integrate all particles forward in time up to the measurement time [so, sample from $p(\psi^n|\psi_i^{n-1})$ for each i].
3) Calculate the weights according to (13) and attach these weights to each corresponding particle. Note that the particles are not modified, only their relative weight is changed!
4) Resample the particles such that the weights are equal to $1/N$.
5) Repeat steps 2, 3, and 4 sequentially until all observations up to the present have been processed.

1) Probabilistic resampling

The resampling can be performed in many ways. The most straightforward method is to directly sample randomly from the density given by the weights (Gordon et al. 1993). Since this density is discrete and one-dimensional this is an easy task. However, because of the random character of the sampling, so-called sampling noise is introduced.

Kim et al. (2003) applied this method to the one-dimensional double-well problem and found superior results over basic importance sampling. They called their method the weight resampling filter. The best results for the double-well problem were achieved with their parametric resampling filter, which we discuss as the maximum entropy filter in section 5. Zhou et al. (2006) estimated a 684-dimensional state vector in their land surface dynamics model using 1 satellite-derived brightness observation every day for 28 days. They compared the results of the particle filter with probabilistic resampling with those of the EnKF and found that although the particle filter gave better results, it needed 800 particles to converge versus 80 for the EnKF. They mention that when more measurements are used, or when the state space increases, much more particles are needed in the particle filter.

A special case of particle filtering is parameter estimation. Kivman (2003) used the particle filter with probabilistic resampling to estimate model parameters sequentially. The method was compared with the EnKF on the Lorenz 63 model with 250 and 1000 particles. In this highly nonlinear system the particle filter was superior to the EnKF, as can be expected. Interestingly, in some experiments the EnKF did converge, but to the wrong parameter values, showing the need for non-Gaussian estimators in highly nonlinear systems.

Losa et al. (2003) used particle filtering with probabilistic resampling in a combined state-parameter estimation problem. The ecosystem model used had 9 state variables (compartments) and 29 model parameters, of which 15 were estimated and 14 held fixed. Using 1000 particles the authors obtained a converged solution (in which some parameters showed seasonal variations, pointing to serious model problems). The particles were drawn from the combined model state–parameter space, and each time step the model noise was changed, while the parameters were held fixed until the new observations. Then the particles were weighted and resampled probabilistically. After resampling identical particles are present in the posterior ensemble with identical values for the parameters. For the model state this is not problematic because of the time-variable model noise, but the spread in parameter space is lost. To overcome this problem one could try to sample from a smoothed approximation of the parameter pdf (see e.g., West 1993). Instead, the authors used a simpler scheme, in which the mean and variance of each parameter were determined, and new parameters were chosen from a uniform pdf with the same mean and variance. To avoid negative parameter values, the pdf was restricted from below.

The continuation of this work is described in Brasseur et al. (2005), in which also the model-noise amplitude in the ecosystem was estimated together with the model parameters. With a 1000-member ensemble the strong seasonal variations in the parameters disappeared, and all seasonal variations moved toward the model-noise amplitude. In this experiment another variant of resampling was used in which the new parameters were chosen from the uniform density bounded by [nearest smaller value, nearest higher value].

2) Residual sampling

To reduce the sampling noise, residual sampling can be applied. In this resampling method all weights are multiplied by 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 are drawn randomly from this resulting distribution. This method was introduced by Lui and Chen (1998), who report a substantial reduction in sampling noise and thus an improved performance of the filter in their small-scale applications. Van Leeuwen (2003a) compared this method with the EnKF in a one-dimensional Korteweg–de Vries (KdV) equation with a state dimension of 100. With an ensemble size of 250 it was shown that the particle filter is superior to the EnKF in the specific parameter regime chosen. The problem with the EnKF was that when the update during the assimilation step is large, some the EnKF particles obtained negative values for the model variable. This leads to unstable behavior of the KdV equation in this parameter regime. The reason for the negative values is that the EnKF assumes the model variable to be approximately Gaussian distributed, while it is positive definite. (It should be mentioned that the EnKF might perform much better with a few improvements over the basic formulation.) On the other hand, the particle filter did show filter divergence more often than the EnKF for some settings, but when both the EnKF and the particle filter became degenerate the particle filter did recover more easily. Van Leeuwen (2003a) also used this method with 512 particles to estimate the evolution of a five-layer quasi-geostrophic model of the ocean area around South Africa, governed by highly nonlinear dynamics. The dimension of the state vector was about 200 000.

To avoid filter divergence the tails of the observational pdf had to be widened from a Gauss to a Lorentz (Cauchy) density, the rationale being that the Gaussian has too narrow tails to represent observational errors properly.

3) Stochastic universal sampling

While residual sampling reduces the sampling noise, it can be shown that stochastic universal sampling has the lowest sampling noise (Kitagawa 1996; see also Robert and Cassela 2004). In this method all weights are put after each other on a line [0, 1]. Then a random number is drawn from a uniform density on [0, 1/N]. Then N line pieces starting from the random number, and with interval length 1/N are laid on the line [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.

4) Monte Carlo Metropolis–Hastings

An enormous amount of literature exists on the Metropolis–Hastings algorithm (Hastings 1970, see also Robert and Cassela 2004). It is unusual to describe it as a resampling method, but for the application we have in mind it is. Here we restrict ourselves to a straightforward application to particle filtering (see, e.g., Dowd 2007, for a geophysical application). The algorithm works as follows:

1) Sample N particles ψ_i from the initial model probability density $p(\psi^0)$.
2) Sample from $p(\psi^n|\psi_i^{n-1})$ once for each i (i.e., run the N particles forward in time).
3) Calculate the weights according to (13).
4) We do not automatically accept all particles as members of the new ensemble, but use the Metropolis–Hastings algorithm, as follows:
   - Choose one particle (e.g., ψ_1^n) as a member of the new ensemble.
   - Particle 2 is also a member of the new ensemble if either

$$w_2 > w_1 \tag{16}$$

   or it is accepted with probability

$$p = \frac{w_2}{w_1}. \tag{17}$$

   - The latter condition is implemented by choosing a random number ν from a uniform density on [0, 1], and accepting particle 2 when $\nu < p$. When it is not accepted we duplicate particle 1.
5) Repeat steps 4 and 5 using the last chosen particle as particle 1, and the new particle as particle 2.

5) Regularization

A potential problem with resampling is that several particles will be identical after the resampling step. If the model has no error, these particles will remain identical during forward integration, leading eventually to filter divergence. Also when the model error is relatively small, or the dynamics of the model is itself degenerate (i.e., a strong low-dimensional attractor is close by), the filter can become degenerate. (On can argue, however, that for realistic atmospheric and oceanic applications model errors are substantial.) To avoid this, some "jitter" has to be applied to the resampled particles. In the statistical literature this action has been termed *regularization*, and one speaks of regularized particle filters. Several regularization algorithms have been proposed. The central idea is to resample the particles from a continuous representation of the pdf. Typically kernel density smoothers are used, in which the sum of delta functions is replaced by a sum of smooth functions:

$$p(\psi) = \sum_{i=1}^{N} c_i g_i(\psi), \tag{18}$$

where

$$\sum_i c_i = 1 \tag{19}$$

and where $g_i(\psi)$ is the smooth pdf centered around particle ψ_i. Popular choices for $g_i(\psi)$ are the Gaussian density and the Epanechnikov kernel. The width of these kernels should be as large as possible to avoid degeneracy, and as small as possible to avoid extra statistical noise. An overview is given by Musso et al. (2001). One can show from density estimation theory (see, e.g., Silverman 1986), that the optimal choice for the kernel for equal weight particles is the Epanechnikov kernel, given by

$$g_i(\psi) = \frac{n+2}{2\gamma}(1 - |\psi - \psi_i|^2) \quad \text{for} \quad |\psi - \psi_i| < 1, \tag{20}$$

and zero otherwise, where n is the dimension of ψ and γ is the volume of a unit sphere in R^n.

Pham (2001) proposes to sample new particles from a Gaussian density centered around particles with multiple identical copies, found from probabilistic resampling. Applications to the Lorenz 63 model show that some 200 particles are needed for good performance. As an extreme case Xiong et al. (2006), probably unaware of the work by Pham (2001), draw their new particles from a Gaussian posterior density with mean and covariance found from the complete weighted ensemble. This makes sense because the posterior pdf will be much more Gaussian than the prior pdf because the prior is multiplied by a Gaussian likelihood. In their example of the Lorenz 63 model the method produces results that are slightly better than the EnKF, although the authors mention that this result will depend on the model parameters chosen.

6) Is resampling enough?

The future for resampling in large-scale geophysical applications is limited according to a growing number of authors. Budhiraja et al. (2007) state (without proof) that a particle filter with a suitable resampling scheme might work in systems of dimension up to 15.

Bengtsson et al. (2008) and Snyder et al. (2008) give arguments why resampling the prior pdf is not enough to avoid filter divergence in large-scale applications. The problem is related to the small number of particles that are typically found in those areas where the likelihood is large. This especially happens if a large number of (independent) observations is present, which makes the likelihood peak in only a very small portion of the observation space. So, via the likelihood the curse of dimensionality comes back into the problem. The importance of these papers allows us to spend some more time on them. Assuming that the observations are identically distributed and independent the weights can be written as

$$w_i = \frac{p(d|\psi_i)}{\sum_k p(d|\psi_k)} = \frac{\prod_{j=1}^{N_y} p(d_j|\psi_i)}{\sum_k \prod_{j=1}^{N_y} p(d_j|\psi_k)}, \qquad (21)$$

where N_y is the total number of observations. Now introduce $\phi(x_j) = -\log p(x_j)$ to find

$$w_i = \frac{\exp\left[-\sum_{j=1}^{N_y} \phi(d_j|\psi_i)\right]}{\sum_k \exp\left[-\sum_{j=1}^{N_y} \phi(d_j|\psi_k)\right]}. \qquad (22)$$

If the observation pdf is Gaussian, $\phi(d_j|\psi_k) = ½[d_j - H_j(\psi_k)]^{\mathrm{T}} R^{-1}[d_j - H_j(\psi_k)]$. Snyder et al. (2008) and Bengtsson et al. (2008) now concentrate on the largest weight. The idea is to show that in order to keep the largest weight much smaller than 1 the number of particles has to grow exponentially with "the size of the problem." We come back to this in a minute. The largest weight can be taken without loss of generality as that of the first member, and by dividing numerator and denominator by the numerator we find for this largest weight

$$\begin{aligned} w_1 &= \left\{1 + \sum_{k=2}^{N} \exp\left[-\sum_{j=1}^{N_y} \phi(d_j|\psi_k) - \phi(d_j|\psi_1)\right]\right\}^{-1} \\ &= \left\{1 + \sum_{k=2}^{N} \exp[-\tau(S_k - S_1)]\right\}^{-1}, \end{aligned} \qquad (23)$$

where τ is the square root of the variance (in particles k) of $\sum_{j=1}^{N_y} \phi(d_j|\psi_k)$ and S_i is the normalized value:

$$S_i = \frac{1}{\tau}\left\{\sum_{j=1}^{N_y} \phi(d_j|\psi_k) - E\left[\sum_{j=1}^{N_y} \phi(d_j|\psi_k)\right]\right\}. \qquad (24)$$

With these expressions the largest weight can be written as

$$w_1 = \frac{1}{1+T}, \qquad (25)$$

where T is given by

$$T = \sum_{k=2}^{N} \exp[-\tau(S_k - S_1)]. \qquad (26)$$

Filter divergence occurs when T approaches zero, so one has to look at the expectation of T. To find this expectation it is assumed that the S_i are $N(0, 1)$ distributed. This is a weak point in the analysis, because a rigorous proof on the distribution of the S_i is lacking. However, the end result fits quite well with numerical experiments, so the assumption has serious backup. When the S_i are indeed Gaussian, it is easy to show that as $N \to \infty$ the S_i tends to $-\sqrt{2\log N}$. After a few manipulations and assumptions the authors arrive at

$$E[T] \approx \frac{\sqrt{2\log N}}{\tau}. \qquad (27)$$

To prevent the filter from diverging, the number of particles $N \gg \exp(\tau^2/2)$. Now recall that τ is the square root of the variance of $\sum_{j=1}^{N_y} \phi(d_j|\psi_k)$. τ depends on the variability of the prior and on the characteristics of the observations. It can be considered as an effective dimension of the system, but in general it is unclear what the relation to the state dimension is. However, it is reasonable to assume that it will grow with the state dimension, and the ensemble size should grow exponentially with its square. As a rough order of magnitude one might use the number of independent observations for τ^2. So 10 independent observations lead to $N \gg 150$, and 50 independent observations need $N \gg 10^{11}$. These results are consistent with the findings of Zhou et al. (2006) and van Leeuwen (2003a) described above. This leads to the conclusion that a particle filter with resampling will diverge for large-scale applications. More is needed, and section 4 deals with extensions of the simple resampling technique.

b. The marginal particle filter

The marginal particle filter, introduced by Klaas et al. (2005), uses the fact that in basic importance sampling the joint-in-time pdf is updated using a new observation, while in the filtering problem we are only interested in the marginal pdf at the analysis time. That basic importance sampling is indeed working on the joint pdf is immediately clear when we realize that for each particle all weights from all observations are just multiplied with each other to obtain the weight of the particle at the present time. In the marginal particle filter the past is integrated out. This idea is exploited as follows. The marginal prior pdf can be obtained from the joint pdf by integration:

$$\begin{aligned} p(\psi^n|d^{1:n-1}) &= \int p(\psi^n, \psi^{n-1}|d^{1:n-1})\, d\psi^{n-1} \\ &= \int p(\psi^n|\psi^{n-1})p(\psi^{n-1}|d^{1:n-1})\, d\psi^{n-1}. \end{aligned} \tag{28}$$

If we now introduce the particle representation of the pdf at time $n - 1$ we find that each particle of the prior pdf at time n can be obtained from

$$\psi_j \sim \sum_i w_i^{n-1} p(\psi^n|\psi_i^{n-1}), \tag{29}$$

where the $\sim$ denotes that we have to sample from the density at the right-hand side. So, we have to propagate all particles ψ_i^{n-1} forward in time and construct the weighted sum as indicated to obtain particle ψ_j^n. This has to be done for each j, so N^2 integrations have to be performed. This is exactly equivalent to the particle filter with resampling, since instead of resampling at $n - 1$, a weighted sample is produced at time n. In this way the stochastic noise introduced by the resampling step is avoided. The new ensemble at time n is then confronted with the observations and new weights w^n are obtained from the likelihood, as

$$w_j^n \approx p(d^n|\psi_j^n) \tag{30}$$

as usual. Klaas et al. (2005) apply the method to a one-dimensional highly nonlinear model. They show that the marginal particle filter is much less sensitive to outliers in observations, especially in combination with the auxiliary particle filter, to be discussed in the next section. Finally, the authors discuss techniques from N-body learning that can reduce the number of integrations to N log$N$, or even smaller for special problems. No applications to geophysical systems have been described yet.

c. Hierarchical models

In some situations it is possible to formulate the full assimilation problem in terms of a sequence of easier-to-handle assimilation problems. The basic idea is the following. Suppose the state vector contains two parts ψ and ϕ (or ψ is the vector containing the model variables and ϕ denotes unknown parameters in the model). The pdf of the state vector can be written as

$$p(\psi, \phi) = p(\psi|\phi)p(\phi). \tag{31}$$

If both $p(\phi)$ and $p(\psi|\phi)$ are easy to draw from, this equation allows us to generate samples from $p(\psi, \phi)$.

In a particle implementation this decomposition can be used as follows. Assume, for example, that ϕ denotes unknown model parameters. Draw N sets of model parameters ϕ_i, and for each parameter set evaluate $p(\psi|\phi_i)$, which is in this case a model run with parameters equal to ϕ_i. This then gives us an ensemble of particles drawn from $p(\psi, \phi)$.

Two advantages of the method can be highlighted. First, the conditional densities have smaller dimension than the joint densities. Second, different algorithms can be used for different parts of the hierarchy.

Wikle and Berliner (2007) explain the method in general, and Berliner and Wikle (2007) use an example in which velocity and pressure observations are available and the model is "stochastic geostrophic," that is,

$$\rho f u = \beta_{u1} \frac{\partial P}{\partial x} + \beta_{u2} \frac{\partial P}{\partial y} \tag{32}$$

and a similar equation for υ. Clearly, if the field is exactly geostrophic $\beta_{u1} = 0$ and $\beta_{u2} = -1$, but now these two parameters are to be estimated. So in this case we want to estimate

$$p(\mathbf{u}, P, \beta|d_{\mathbf{u}}, d_p) \sim p(d_{\mathbf{u}}|\mathbf{u})p(\mathbf{u}|P, \beta)p(\beta)p(d_p|P)p(P). \tag{33}$$

We assume the densities of the pressure P, and of the βs to be known. One starts with drawing samples P_i from $p(P)$, evaluates the likelihoods $p(d_p|P)$, and one resamples the P_i to give them equal weight, then draw samples for the βs. With these in hand we generate samples of the velocities, using the stochastic geostrophy model. These are then confronted with the observations in $p(d_{\mathbf{u}}|\mathbf{u})$, leading again to a weighted ensemble of velocity, pressure, and parameter fields. This ensemble of particles represents the posterior pdf, as we wanted. So, instead of drawing particles from $p(\mathbf{u}, P, \beta)$ we reduced the dimension of the system and only had to draw from $p(P)$ and $p(\beta)$.

Although this sounds simple, it does not work this way in practice. For instance, Berliner and Wikle (2007) had to use an EOF decomposition of the pressure field and estimate the coefficients of those fields. Also other smart changes in the formulation of the problem were needed to obtain a stable solution. Furthermore, one must be able to draw samples from the prior densities, in this case pressure and parameters, enforcing relatively simple pdfs for these. Wikle and Berliner (2007) argue that basic importance sampling, as outlined above, should be used only for low-dimensional problems. In higher-dimensional problems they propose Markov chain Monte Carlo (MCMC) methods, like the Gibbs sampler, Metropolis–Hastings, and slice samplers. This goes beyond particle filtering and is not discussed here. The interested reader is referred to Robert and Cassela (2004).

To conclude, hierarchical models provide a systematic way to reduce the complexity of the assimilation problem by highlighting specific dependencies between model variables. This can be very useful, but it is unclear how to exploit this efficiently in high-dimensional problems. However, as Berliner and Wikle (2007) state, in combination with other techniques it might help in solving the particle filter problem, and dismissing particle filters altogether is premature.

4. Improvement of the likelihood

Related to decreasing the variance of the weights is ensuring that all model integrations end up close to the new observations. First we discuss the application of a proposal density that allows one to sample from a density that is conditioned on the new observations, so it is much closer to these observations than the prior density. As an example the EnKF is chosen as the proposal density. Then we discuss the auxiliary particle filter, the backtracking particle filter, and the guided SIR. In both the auxiliary particle filter and the guided SIR future observations are used directly to improve the likelihood, while the backtracking filter ''tries again.''

As a bit of a side track, before we go into these methods, we shortly discuss smart ways to generate the prior ensemble based on the dynamics of the system at hand. One could use the singular vectors of the linearized model equations as particles (Chorin and Krause 2004). The idea is to use the singular vectors to explore the state space only in the direction of the growing eigenmodes of the linearized model operator. The eigenmodes are related to the Liapounov exponents of the system. However, Bowler (2006) compared the impact of singular vectors, error breeding, random perturbations, and EnKF strategies on the forecasts (so, on the quality of the next prior) for the 40-variable Lorenz-95 model. It turned out that the breeding perturbations were too similar to each other to show a reasonable spread. The singular vectors denote the fastest growing structures, but there is no guarantee that the actual errors will project significantly on those structures. Looking at their structure, they tend to be very localized in space, so that a large number of them is needed to perturb all grid points. The random perturbations and the EnKF scheme were superior. It must be mentioned that the 40-variable Lorenz-95 model is of much smaller dimension than the applications we have in mind in this paper, but the problems with the error-breeding and singular-vector ensembles seem to be general. Several attempts have been performed on the ENSO system with a Kalman filter using a reduced EOF space (e.g., Ballabrera-Poy et al. 2001), and again it was found that the system tends to grow out of the reduced space and the assimilation fails after some time.

a. The proposal density

We now discuss a very interesting property of particle filters that has received little attention in the geophysical community. This has to do with the following. Typically, we are interested in evaluating at time n integrals like

$$\begin{aligned}\overline{f(\psi^n)} &= \int f(\psi^n)p(\psi^n|d^n)\,d\psi^n\\ &= \frac{1}{A}\int f(\psi^n)p(d^n|\psi^n)p(\psi^n)\,d\psi^n,\end{aligned} \tag{34}$$

where A is a normalization factor. This integral can be written as

$$\overline{f(\psi^n)} = \frac{1}{A}\int f(\psi^n)p(d^n|\psi^n)\frac{p(\psi^n)}{q(\psi^n)}q(\psi^n)\,d\psi^n, \tag{35}$$

where $q(\psi^n)$ is another probability density. Obviously, the support of q has to be equal to or larger than that of p to avoid division by 0. The density q is called the *proposal density*, and we could sample from q instead of from p to approximate the integral. Using a particle representation of q, we obtain for the mean of f in the posterior

$$\overline{f(\psi^n)} = \frac{1}{A}\sum_i f(\psi_i^n)p(d^n|\psi_i^n)\frac{p(\psi_i^n)}{q(\psi_i^n)} = \sum_i w_i f(\psi_i), \tag{36}$$

where the weights are given by

$$w_i = \frac{1}{A}p(d^n|\psi_i^n)\frac{p(\psi_i^n)}{q(\psi_i^n)}. \tag{37}$$

The interest comes from the fact that we can try to put information of the observations d^n in the density q such that the particles ψ_i^n, which are sampled from q, are close to these observations. This opens a whole new area of possible improvements of efficiency of particle filters. When the observations are included we have

$$\overline{f(\psi^n)} = \frac{1}{A}\sum_i f(\psi_i^n)p(d^n|\psi_i^n)\frac{p(\psi_i^n)}{q(\psi_i^n|d^n)} = \sum_i w_i f(\psi_i), \tag{38}$$

with

$$w_i = \frac{1}{A}p(d^n|\psi_i^n)\frac{p(\psi_i^n)}{q(\psi_i^n|d^n)}. \tag{39}$$

(As mentioned before, in the statistical literature one tends to find derivations of importance sampling with the proposal density included from the beginning. For pedagogical reasons the proposal density is introduced here.)

Although this sounds good, a close inspection of the equations above reveals a difficulty. The ψ_i are drawn from q, so we can infer their probability. But how should we determine $p(\psi_i)$? And, similarly, how can we evaluate $q(\psi_i|d^n)$? Only when the shape of the prior and proposal pdf are known can these probabilities be determined. This might be possible for the proposal, which we are free to choose, but not for the prior. To solve this problem we have to go to a slightly different formulation of the assimilation problem (see, e.g., Doucet et al. 2001). We first derive the equations without proposal density for clarity.

We can expand $p(\psi^n)$ in Bayes's theorem to find

$$p(\psi^n|d^n) = \frac{p(d^n|\psi^n)\int p(\psi^n\psi^{n-1})\,d\psi^{n-1}}{\int p(d^n|\psi^n)p(\psi^n)\,d\psi^n}, \tag{40}$$

which can be rewriten as

$$p(\psi^n|d^n) = \frac{p(d^n|\psi^n)\int p(\psi^n|\psi^{n-1})p(\psi^{n-1})\,d\psi^{n-1}}{\int p(d^n|\psi^n)p(\psi^n)\,d\psi^n}. \tag{41}$$

Assuming a (resampled) particle representation of $p(\psi^{n-1})$ [i.e., $p(\psi^{n-1})$ is a sum of delta functions] we find

$$p(\psi^n|d^n) = \frac{p(d^n|\psi^n)}{\int p(d^n|\psi^n)p(\psi^n)\,d\psi^n}\frac{1}{N}\sum_i p(\psi^n|\psi_i^{n-1}). \tag{42}$$

Note that we have assumed that the particles at $n-1$ all have the same weight (e.g., from a resampling step at $n-1$). This is not essential for the following, but is assumed here for clarity. The procedure is that we now draw a sample from $\sum_i p(\psi^n|\psi_i^{n-1})$, which is equal to choosing the stochastic forcing for the model integration starting at ψ_i^{n-1} to form ψ_i^n. Sampling N particles from this density gives a posterior that is a weighted sum of delta functions with weights as before:

$$w_i = \frac{p(d^n|\psi_i^n)}{\sum_j p(d^n|\psi_j^n)}. \tag{43}$$

Let us now introduce the proposal density again. The posterior density is written as

$$p(\psi^n|d^n) = \frac{1}{A}p(d^n|\psi^n)\frac{1}{N}\sum_i \frac{p(\psi^n|\psi_i^{n-1})}{q(\psi^n|\psi_i^{n-1}d^n)}q(\psi^n|\psi_i^{n-1}d^n), \tag{44}$$

where A is the normalization factor:

$$A = \int p(d^n|\psi^n)p(\psi^n)\,d\psi^n. \tag{45}$$

Drawing from the transition density $q(\psi^n|\psi_i^{n-1}d)$, that is, performing a model integration from each of the ψ_i^{n-1} gives us the sample of the posterior density, with weights:

$$w_i \propto p(d^n|\psi_i^n)\frac{p(\psi_i^n|\psi_i^{n-1})}{q(\psi_i^n|\psi_i^{n-1}d^n)}. \tag{46}$$

It is important to note that we never actually calculate the normalization constant A. The important observation is that it does not depend on a specific ψ_i, so it is the same for all particles. In practice one calculates the weights up to a normalization constant, and normalizes these weights by their sum.

This formulation of Bayes's theorem gives us the possibility to explore the proposal density. 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. We will follow the presentation by Papadakis (2007). Since we know the starting point of the simulation, ψ_i^{n-1}, and its end point, the posterior EnKF sample ψ_i^n, and we know the model equation, written formally as

$$\psi_i^n = f(\psi_i^{n-1}) + \epsilon_i^n, \tag{47}$$

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

$$p(\psi_i^n|\psi_i^{n-1}) \propto \exp\left\{-\frac{1}{2}[\psi_i^n - f(\psi_i^{n-1})]Q_m^{-1}[\psi_i^n - f(\psi_i^{n-1})]\right\}. \tag{48}$$

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

Let us now calculate the proposal density $q(\psi_i^n|\psi_i^{n-1}d^n)$. This depends on the EnKF used. For the EnKF with perturbed observations (see Evensen 2006) the situation is as follows. Each particle in the updated ensemble is connected to those before analysis as

$$\psi_i^n = \psi_i^{n,\mathrm{old}} + K^e[d + \delta_i - H(\psi_i^{n,\mathrm{old}})], \tag{49}$$

where δ_i is the random error that has to be added to the observations in this variant of the EnKF. Here K^e is the ensemble Kalman 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:

$$\psi_i^{n,\mathrm{old}} = f(\psi_i^{n-1}) + \epsilon_i^n. \tag{50}$$

Combining these two gives

$$\psi_i^n = f(\psi_i^{n-1}) + \epsilon_i^n + K^e\{d + \delta_i - H[f(\psi_i^{n-1})] - H(\epsilon_i^n)\} \tag{51}$$

or

$$\begin{aligned}\psi_i^n = {} & f(\psi_i^{n-1}) + K^e\{d - H[f(\psi_i^{n-1})]\} \\ & + (1 - K^e H)\epsilon_i^n + K^e\delta_i,\end{aligned} \tag{52}$$

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 ψ_i^{n-1} to ψ_i^n. Assuming both model and observation errors to be Gaussian distributed and independent we find for this transition density:

$$q(\psi_i^n|\psi_i^{n-1}d^n) \propto \exp[-\frac{1}{2}(\psi_i^n - \mu_i^n)^{\mathrm{T}}\Sigma_i^{-1}(\psi_i^n - \mu_i^n)], \tag{53}$$

where μ_i^n is the deterministic "evolution" of ψ given by

$$\mu_i^n = f(\psi_i^{n-1}) + K^e\{d - H[f(\psi_i^{n-1})]\} \tag{54}$$

and the covariance Σ_i is given by

$$\Sigma_i = (1 - K^e H)Q_m(1 - K^e H)^{\mathrm{T}} + K^e R K^{e\mathrm{T}}, \tag{55}$$

where we assumed that the model and observation errors are uncorrelated. It should be realized that ψ_i^n does depend on all $\psi_j^{n,\mathrm{old}}$ via the Kalman gain, that involves the error covariance P^e. Hence, we have calculated $q(\psi_i^n|P^e\psi_i^{n-1}d)$ instead of $q(\psi_i^n|\psi_i^{n-1})$, 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 ψ_i^n, that will depend on the specific realization for ϵ_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.)

For a square root filter the situation is a bit different because the observations are not perturbed. Instead, an updated ensemble is found as

$$\psi_i^n = \sum_{j=1}^{N}(\psi_j^{n,\mathrm{old}} - \overline{\psi^{n,\mathrm{old}}})\mathbf{W}_{ji} + \overline{\psi^{n,\mathrm{old}}}, \tag{56}$$

where $\mathbf{W}_{ij}$ is a symmetric transition matrix dependent on ensemble covariance, innovation, and observation error covariance. Note that the $\mathbf{W}_{ij}$ can be considered as weights (i.e., how strong each old member j counts in forming the new member i). As before, $\psi_j^{n,\mathrm{old}}$ is the particle j at time n before the Kalman filter analysis step. Using the stochastic model evolution equation, we obtain

$$\psi_i^n = \sum_{j=1}^{N}\left[f(\psi_j^{n-1}) + \epsilon_j^n - \overline{\psi^{n,\mathrm{old}}}\right]\mathbf{W}_{ji} + \overline{\psi^{n,\mathrm{old}}} \tag{57}$$

or

$$\psi_i^n = \sum_{j=1}^{N}\left[f(\psi_j^{n-1}) - \overline{\psi^{n,\mathrm{old}}}\right]\mathbf{W}_{ji} + \overline{\psi^{n,\mathrm{old}}} + \sum_{j=1}^{N}\epsilon_j^n\mathbf{W}_{ji}. \tag{58}$$

Again a deterministic and a stochastic part can be found at the right-hand side. As above, the deterministic part shows dependence not only on ψ_i^{n-1}, but also on all other ψ_i^{n-1}. The argument used there is repeated here: in case of an infinite ensemble size this term would be a system variable not dependent on specific realizations. Also, the stochastic term only depends on i when the ensemble size becomes very large.

The transition density can be found as

$$q(\psi_i^n|\psi_i^{n-1}d^n) \propto \exp\left[-\frac{1}{2}(\psi_i^n - \mu_i^n)^{\mathrm{T}}\Sigma_i^{-1}(\psi_i^n - \mu_i^n)\right], \tag{59}$$

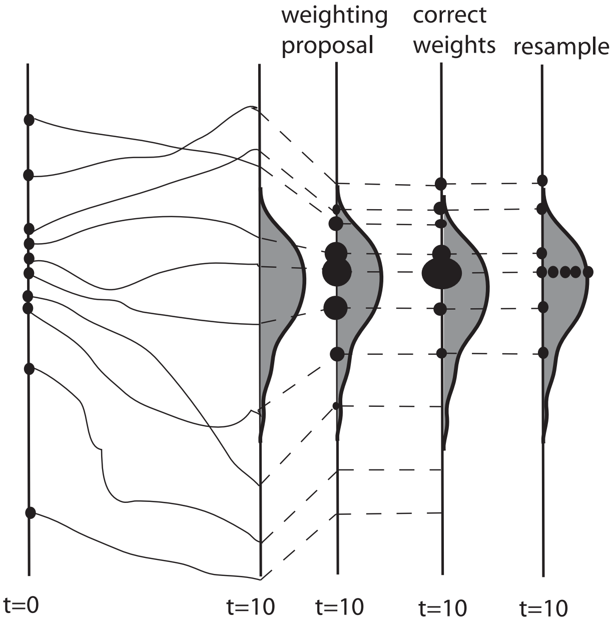

FIG. 3. The particle filter with proposal density. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight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

where now

$$\mu_i^n = \sum_{j=1}^{N} \left[f(\psi_j^{n-1}) - \overline{\psi^{n,\text{old}}}\right]\mathbf{W}_{ji} + \overline{\psi^{n,\text{old}}} \tag{60}$$

and the covariance Σ_i is given by

$$\Sigma_i = \sum_{j=1}^{N} \mathbf{W}_{ij} Q_m \mathbf{W}_{ij}^{\mathrm{T}}. \tag{61}$$

The calculation of $p(\psi_i^n|\psi_i^{n-1})$ and $q(\psi_i^n|\psi_i^{n-1}d^n)$ looks very expensive. By realizing that Q_m 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 in (46) are easily calculated. The procedure now is as follows (see Fig. 3):

1) Run the ensemble up to the observation time n.
2) Perform a (local) EnKF analysis of the particles.
3) Calculate the relative weights $w_i^* = p(\psi_i^n|\psi_i^{n-1})/q(\psi_i^n|\psi_i^{n-1}d^n)$ using (48, 53) or (48, 59).
4) Calculate the relative weights $w_i = p(d^n|\psi_i^n)$.
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 lower observational errors. This might look like overfitting but it is not because the only thing we do in probabilistic sense is to "drag" particles to those positions in state space where the likelihood is large. Any other proposal density that does this could be used instead.

Van der Merwe et al. (2000) use the unscented Kalman filter (UKF) as the proposal density. The UKF, introduced by Julier and Uhlmann (1997), is an EnKF in which the initial particles are chosen as in a square root filter, capturing the mean and the covariance of the prior density, but with the possibility to use a scaling parameter to change the distance of the particles to the mean without changing the covariance. Van der Merwe et al. (2000) use in their implementation of the so-called unscented particle filter a Gaussian density around each particle, so, in fact, their proposal is a Gaussian mixture model (see next section). Each Gaussian is updated using all observations, the result of which is again a Gaussian distribution $q(\psi^n|\psi_i^{n-1}d^n)$. The proposal density value $q(\psi_i^n|\psi_i^{n-1}d^n)$ is found by drawing a ψ_i^n from that Gaussian density and evaluating its probability. Note the difference with the EnKF approach: in the EnKF the posterior is represented by one Gaussian density, while in the UPF a Gaussian density is centered around each particle. Because the UKF is used on each particle separately the method needs about N^2 integrations, so it is not feasible in the large-scale geophysical context. Van der Merwe et al. (2000) show in small-scale highly nonlinear examples that the unscented particle filter outperforms the particle filter with resampling and particle filters with the extended Kalman filter as proposal.

As another example the marginal particle filter with proposal density is briefly discussed. The marginal particle filter can easily be expanded by utilizing the proposal density as

$$\psi_j \sim \sum_i w_i^{n-1} q(\psi^n|\psi_i^{n-1}d^n) \tag{62}$$

and forming weights at time n as

$$w_j^n \approx p(d^n|\psi_j^n)\frac{\sum_i w_i^{n-1} p(\psi^n|\psi_i^{n-1})}{\sum_i w_i^{n-1} q(\psi^n|\psi_i^{n-1}d^n)}. \tag{63}$$

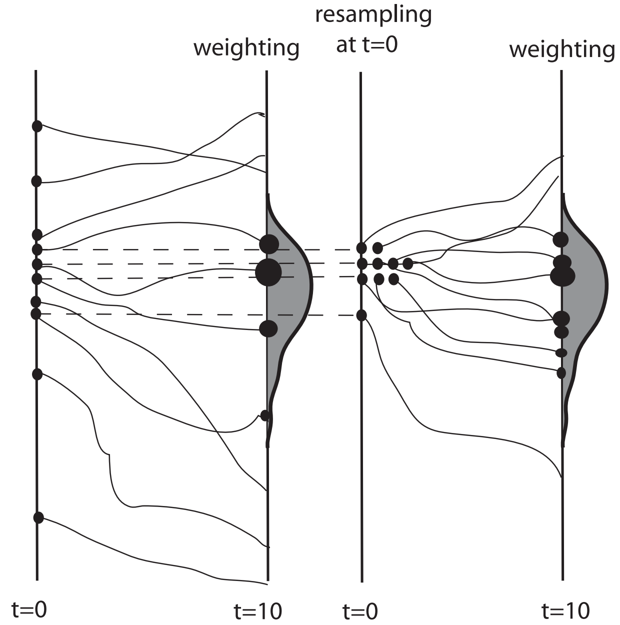

FIG. 4. The auxiliary particle filter. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. 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.

Klaas et al. (2005) also discuss a combination of this filter with the auxiliary particle filter (see the next paragraph) and apply the method to a one-dimensional highly nonlinear model. Again they find that the marginal particle filter is much less sensitive to outliers in observations, especially in combination with the auxiliary particle filter.

b. *Auxiliary particle filter*

In the auxilary particle filter introduced by Pitt and Shephard (1999), the ensemble at time $n - 1$ is weighted with information from the likelihood at time n. 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 0 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 flowchart reads (see Fig. 4):

1) Integrate each particle from $n - 1$ to n with simplified dynamics (e.g., without model noise or omitting some computationally expensive part of the model), producing a representation of the proposal density $q(\psi^n)$ at the measurement time.
2) Weight each particle with the new observations as

$$w_i \propto q(d^n|\psi_i^n). \tag{64}$$

These weights are called the "first-stage weights" or the "simulation weights."
3) Resample the particles at time $n - 1$ with these weights, and use this ensemble as a representation of the proposal density by integrating it forward to n with the full stochastic model.
4) Reweight the members with weights:

$$w_i \propto \frac{p(d^n|\psi_i^n)}{q(d^n|\psi_i^n)}. \tag{65}$$

A resampling step can be done but is not really necessary because the actual resampling is done at step 3.

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 the SIR twice. One can imagine doing it even more times, zooming into the likelihood, but at a cost of performing more and more integrations of the model.

The name "auxiliary" comes from the introduction of a member index in the original formulation by Pitt and Shephard (1999), but that is not needed explicitly. Implicitly, the member index keeps track of the relation between the first-stage weights and the particle sample at $n - 1$, so that a formal treatment of the characteristics of the method is facilitated.

c. *The backtracking particle filter*

One possibility to solve the problem of filter divergence (i.e., when all members obtain very low weights, such that none of the particles follows the observations) is to go back to the time when the particle filter did work properly, and try again. This "trying again" can be done in several ways, and Spiller et al. (2008) describe four methods to do so.

The first idea is to duplicate each particle at time $n - b$ when the filter still performed fine, and propagate $2N$ particles up to time $n + 1$, one time step after the divergence. Then the ensemble size returns to N again until a new divergence is detected.

Another possibility, called cloud expansion, is to go back to $n - b$ and just rerun the N particles with different realizations of the model noise, or, if that does not work, add N particles each with zero mean Gaussian random noise centered around the original N particles.

These $2N$ particles are then used until time $n + 1$, after which one proceeds again with N particles.

A more serious change is directed doubling, in which one goes back to $n - b$ and uses the ensemble at that time before resampling. Take the M particles with the largest weights (typically, $M = N/10$) and add new particles along the line between each of these particles and the observations. (With more than one observation at a time this becomes more complicated, but one can probably generate efficient schemes that do this in an approximate way.) To preserve the mean of the ensemble, add the same number of particles on the same line at the other side of each of the M original particles. The number of extra particles is such that the total size of the new ensemble becomes $2N$. This ensemble is propagated up to time $n + 1$, after which the size is brought back to N. A potential problem with this method in high-dimensional systems with implicit balances is that it is unclear how to generate these new particles because they will not all be close to an original particle. In the application used by Spiller et al. (2008) in which the position of two point vortices in a 2D domain has to be found when a tracer is observed, this is not a problem.

Finally, Spiller et al. (2008) use doubling with perturbed observations, in which one goes back to $n - b$, perturbs the observations $2N$ times, and calculates $2N$ particles using Bayes's theorem. This step reminds us of the perturbed-observation implementation of the EnKF. Then these $2N$ particles are used up to time $n + 1$, after which one returns back to N particles.

Spiller et al. (2008) showed that each of the variants of the backtracking particle filter worked better than the extended Kalman filter (EKF) and than the particle filter with resampling in their highly nonlinear Lagrangian tracking problem with $N = 500$. A possible disadvantage is that the filters typically need $2N$ model integrations, similar to the auxilary particle filter.

d. The guided SIR

In this method, introduced by van Leeuwen (2002), we use the fact that observations are available not at every time step, but at say each L time steps. This means that the model is stepping forward L time steps before new observations are available. This is the common situation in geophysical systems. The idea is to weight and resample the ensemble of particles using future observations at the present time. This will guide the particles into the direction of these future observations. To avoid drawing the particles to the observations too strongly too early, one increases the error covariance of the observations by a factor of >1, typically a factor 10–100 is used. A possible implementation is as follows (see Fig. 5):

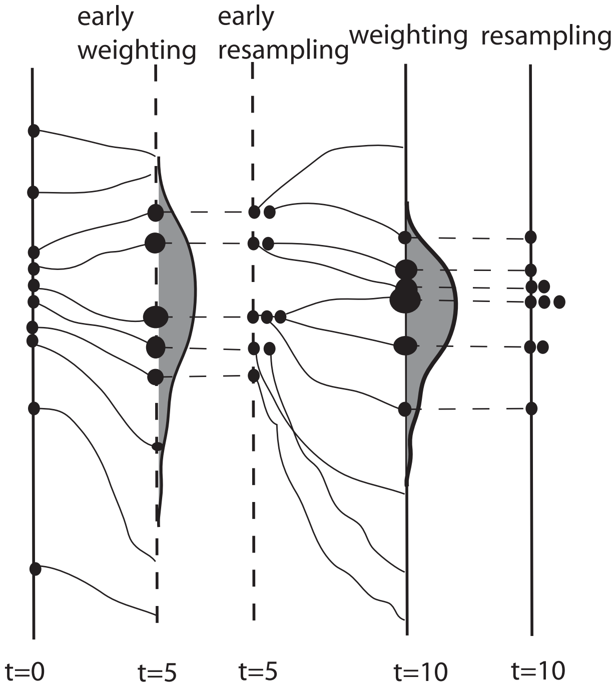

FIG. 5. The guided particle filter. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. Pseudo-observations are introduced at time 5. These are equal to those at time 10, but with larger observational errors, so wider likelihood. The particles are weighted with that likelihood and resampled. At time 10 the particles are weighted according to the true likelihood, and these weights are corrected for the artificial weights from the pseudo-observations at time 5.

1) Run the particles from the previous analysis time $n - L$ to some predefined time $n - L + k$ in between measurement times, where $k < L$.
2) Calculate the weights using the observations of the next measurement time with the observational error covariance multiplied by $1 + 10(L - k)/L$, for example, as

$$w_i \propto p(d^n|\psi_i^{n-L+k}) \quad \text{with} \quad \sum_i w_i = 1. \tag{66}$$

3) Resample the particles according to these weights. This ensures that the particles are on the good track toward the next observations.
4) Reweight this new ensemble to compensate for the extra weights that have been introduced in step 3, *but do not resample*, as

$$\alpha_i \propto [p(d^n|\psi_i^{n-L+k})]^{-1} \quad \text{with} \quad \sum_i \alpha_i = 1. \tag{67}$$

So now each particle i has weight α_i, but is still on the good track toward the observations.

5) Propagate this ensemble forward in time to the next predefined time $n - L + 2k$, again before the actual measurement time (so also $2k < L$).
6) Repeat step 2, using as weights

$$w_i \propto \alpha_i p(d^n|\psi_i^{n-L+2k}) \quad \text{with} \quad \sum_i w_i = 1, \qquad (68)$$

with the observational error covariance multiplied by, for example, $1 + 10(L - 2k)/L$.
7) Repeat steps 3, 4, 5, and 6 until the ensemble reaches the true measurement time.
8) Calculate the weights at the measurement time, as

$$w_i \propto p(d^n|\psi_i^n) \quad \text{with} \quad \sum_i w_i = 1. \qquad (69)$$

9) Resample the particles according to these weights (i.e., without corrections) since they are not needed at this step.

Note that all we did in this procedure is to guide the particles toward the observations via resampling at intermediate steps (step 3). To avoid a bias we compensated for the extra weights introduced by the intermediate resampling in step 4.

A few comments are presented here:

1) One could do a resampling at step 4 (A. Doucet 2003, personal comunication).
2) One can choose k differently in each cycle of step 7.
3) One can use modifications of the true observations at time n. For instance, when a daily cycle is present and observations are only once a day, one could introduce a daily variation in the pseudo-observations, too. Note that any pseudo-observation can be used, as long as it brings the ensemble closer to the true observations at time n.

5. Approximations

Even with all ideas from the statistical literature presented above, it is difficult to avoid filter degeneracy in geophysical problems with state vectors of size 1 000 000 000 or larger. To this end, several approximations to the SIR filter have been proposed. We discuss here the merging particle filter, localization, as used in the EnKF, kernel dressing, in which each particle is "dressed" with a usually Gaussian pdf, and the maximum entropy particle filter.

Also, a considerable number of papers have appeared recently that approach the problem from the EnKF side, that is, one tries to introduce non-Gaussian elements in the EnKF. Unfortunately, the adaptations to the EnKF or its square root variants are rather ad hoc. Nonetheless, interesting results have been achieved that are of interest in trying to solve the full nonlinear problems. An example is presented by Harlim and Hunt (2007), who widen the prior pdf in a variational formulation of the Kalman filter update. This is done routinely in operational centers now by including a covariance inflation factor in the prior covariance, but Harlim and Hunt change the shape of the prior pdf from a Gauss to a pdf with much broader tail. Obviously, the Kalman filter equations do not apply anymore, but the wider pdf can be incorporated in the variational formulation quite easily.

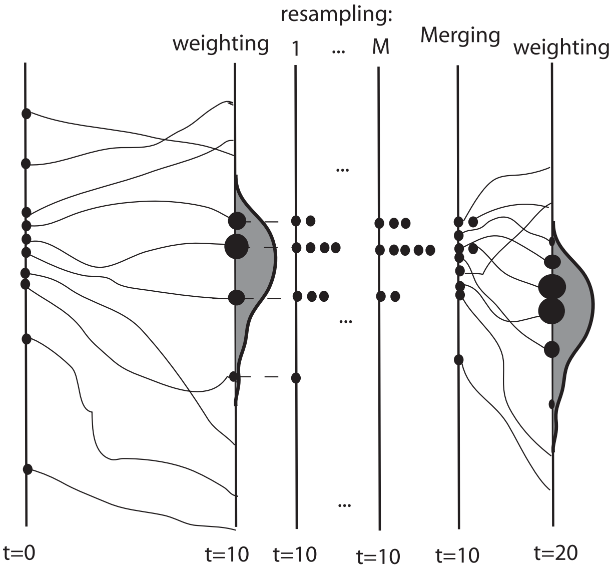

FIG. 6. Merging particle filter. The model variable runs along the vertical axis, the weight of each particle corresponds to the size of the bullets on this axis. The horizontal axis denotes time, with observations at a time interval of 10 time units. All particles have equal weight at time zero. The particles are resampled according to the true likelihood M times at time $t = 10$. The new particles are averages over M of these resampled particles. Note that the new particles are not identical copies of the old ones, so new blood enters the ensemble.

a. The merging particle filter

In this method, introduced by Nakano et al. (2007), linear combinations of the particles are taken at the measurement time to reduce the variance in the weights. These linear combinations are taken such that the mean and the covariance of the ensemble are kept, at the costs of a less correct description of the higher-order moments. In this way the algorithm is still variance minimizing for nonlinear models, unlike the EnKF.

The procedure is as follows (see Fig. 6):

1) Sample N particles ψ_i from the initial model probability density $p(\psi^{n-1})$ (or have them from a previous assimilation step).

2) Integrate all particles forward in time up to the measurement time [so, sample from $p(\psi^n|\psi_i^{n-1})$ for each i].
3) Calculate the weights according to (13) and attach these weights to each particle. Note that the particles are not modified, only their relative weight is changed!
4) Resample the particles such that the weights are equal to $1/N$. So far, the algorithm is identical to the SIR.
5) Repeat step 4 $M - 1$ times, with $M > 2$ to form M ensembles of size N.
6) Merge M particles:

$$\psi_i^n = \sum_{j=1}^{M} \alpha_j \psi_{j,i}^n, \qquad (70)$$

where $\psi_{j,i}^n$ denotes particle $i \in [1, 2, \ldots, N]$ in ensemble $j \in [1, 2, \ldots, M]$. To ensure the correct mean and covariance of the ensemble choose

$$\sum_{j=1}^{M} \alpha_j = 1 \quad \text{and} \quad \sum_{j=1}^{M} \alpha_j^2 = 1 \qquad (71)$$

as proven in Nakano et al. (2007), and which is easily verified.
7) Then repeat step 2.

Nakano et al. (2007) compare the performance of the merging particle filter to the particle filter with resampling and to the EnKF for the Lorenz-63 and -96 models. They note that the EnKF works best with a low number of particles, but increasing the number of particles causes the merging particle filter to take over. Only with a very high number of particles is the particle filter with resampling superior. The breakpoint depends on the linearity of the problem. The authors argue that for strongly non-Gaussian pdfs the merging particle filter will perform less well because only the mean and the covariance of the ensemble are optimized, at the cost of higher-order moments. The difference with the EnKF lies in the fact that in the EnKF approximations are made *before* the analysis step. In the merging particle filter, approximations are made after the analysis step, and only the mean and the covariance remain unchanged (up to sample noise). Hence, the merging particle filter is truly variance minimizing, while the EnKF and its variants are not.

Finally, note that the positions of the particles in state space have changed, unlike a particle filter with direct resampling. This is both good and bad. Good because the spread in the ensemble is enhanced, bad because nonlinear balances in each particle are destroyed to some extent.

b. Gaussian resampling

Xiong et al. (2006) propose to use Gaussian resampling from the posterior pdf as obtained from the particle filter. It is interesting to compare this idea with the merging particle filter. In the merging particle filter the α_j are determined analytically. In Gaussian resampling new particles have to be drawn from the covariance of the weighted ensemble of particles. Clearly, the latter is much more expensive. Gaussian resampling in combination with weights from the particle filter is superior to EnKF-like methods because the posterior is truly variance minimizing, while the EnKF-like methods are not. On the other hand, the particles are changing position in state space, but not toward the observations. In EnKF-like methods all particles are drawn toward the observations. (Note that this is not necessarily true for multimodal likelihoods.)

c. Weights from reduced-dimension models

To avoid large variance in the weights one can reduce the dimension of the assimilation problem by performing the importance-sampling step in a reduced space. For instance, Berliner and Wikle (2007) propose to use the dominant EOFs of the model system and perform the importance sampling on the EOF amplitudes of the particles, given the EOF amplitudes of the observations. The procedure is to first determine the dominant EOFs of the system (e.g., from a long model run, or from a long observational dataset, or from the ensemble of particles at the observation time). Then run the particles up to observation time, determine the EOF amplitudes of both particles and observations, do the importance sampling on the EOF amplitudes, and use these weights as the weights of the full model particles.

More realistic model reduction might be possible using dynamical modes from dynamical system theory. This is one of the areas where data assimilation and dynamical system theory meet directly, and this author expects fruitful collaboration.

d. Localization

It is common practice that the EnKF shows degeneracy when the state space size is much larger than the number of ensemble members. A remedy used in that method is to update locally (i.e., to update the state vector at a certain grid point only the observations in the immediate neighborhood are included in the analysis step). This operation is called localization, and is common practice nowadays in meteorology and oceanography. It was introduced by Houtekamer and Mitchell (1998), and first used operationally by Brusdal et al. (2003) and Ott et al. (2004). The rationale for this procedure comes from the idea that an observation at one location is not expected to have anything to do with the state of the system at the other side of the globe. This neglected correlation can be wrong, for example, for very long-scale phenomena, like long-wavelength waves, and one can imagine localization in spectral space, or in the space spanned by EOFs.

One advantage of localization is that spurious long-range correlations can be suppressed. Because of the relatively small ensemble size we can typically afford, it is hard to obtain zero correlation when the actual correlation is zero. So, correlations close to zero are suspicious, and one can argue to ignore them altogether. Another advantage is that since the update of the ensemble is performed over only part of the full model domain, the size of the state vector is strongly reduced, or the effective ensemble size is greatly increased. Finally, in a direct application of the EnKF without localization a new ensemble will consists of ensemble members that are (almost) a linear combination of the old members, and the space spanned by the ensemble remains the same. Using localization gives ''new blood'' in the ensemble because new members consist of different linear combinations of old members in different parts of the domain.

These advantages motivate the search for a localization procedure in particle filtering. One can calculate the weights locally (i.e., make the weights space dependent and use only observations close to the spatial point). So, it is easy to localize the importance sampling filter. This is less so for more sophisticated filters that employ resampling. In the resampling step low-weight particles are abandoned and high-weight particles are duplicated. However, with local weights, different particles are selected in different parts of the domain. The problem now is that we have to have continuous (in space) model fields to propagate forward in time with the model equations. Just constructing a new particle that consists of one particle in one part of the model domain and another particle in another domain will lead to problems at the boundary between these two. This is less of a problem in the EnKF because the error covariances used there tend to smooth the solution in space. In the SIR, for example, this smoothing is not present.

The above problem is less present in parameter estimation. In most (but not all) model systems parameters can be discontinuous without causing serious problems. An example in provided by Vossepoel and van Leeuwen (2007), as discussed in section 2. To estimate the roughly 10 000 parameters with 128 particles, localization was used, reducing the number of parameters in each batch to about 21 and leading to converged results. Terwisscha van Scheltinga et al. (2009, manuscript submitted to *Ocean Modell.*) extended the method to estimate some 28 000 sea ice strength parameters during a 2-yr assimilation experiment in a finite-element model of the Arctic Ocean, both in space and time. Real sea ice concentration observations were used from the Special Sensor Microwave Imager (SSM/I) satellite instrument together with sea ice drift observations from the Quick Scatterometer (QuikSCAT). Only 32 particles were used to obtain results that converged when the localization area was made small enough and observation errors were large enough. A strong seasonal variation of the sea ice strength values was found, related to the growth or decline of sea ice. These results point to serious problems in the sea ice model, which need further investigation.

When model fields are estimated the smoothness issue is very serious. Several methods to avoid this problem come to mind, which we next discuss briefly.

1) SPATIAL INTERPOLATION

The problem with spatial interpolation is that we have a priori no idea which particles one has to put together in different parts of the domain to generate one new particle over the full domain, so that potentially very large gradients can occur, and interpolation will still yield ''wild'' model fields. Van Leeuwen (2003b) used this method with a multilayer quasigeostrophic model of the ocean area around South Africa and did not encounter these problems, but the author expresses that one can expect spurious gravity wave generation in primitive equation models. So, unless we find efficient ways to generate new global particles that are as smooth as possible given the favorite particles in different parts of the domain, this does not work.

2) HIERARCHICAL BAYES

From a more fundamental point of view, one could use hierarchical Bayes in which part of the state vector is conditioned on other parts. The idea is that the posterior pdf can be written as

$$p(\psi|d) = p(\psi_{\text{in}}|\psi_{\text{out}}d)p(\psi_{\text{out}}|d), \qquad (72)$$

where the subscripts in and out denote inner and outer, respectively. In this way the solution at a specific area, the inner area, is conditioned on the solution in the rest of the domain. To implement localization one assumes that the inner domain is only dependent on that part of the outer domain that is relatively close. The same is assumed for the observations: only those close to the inner domain are relevant for the solution there. With these assumption we can write for any inner domain *i*:

$$p(\psi|d) = p(\psi^i_{\text{in}}|\psi^i_{\text{out}}d^i)p(\psi_{\text{out}}|d). \qquad (73)$$

The outer area is build up of other inner areas around other points in state space. In this way a hierarchy of conditional pdfs is created:

$$p(\psi|d) = \prod_{i=1}^{M} p(\psi_{\text{in}}^{i}|\psi_{\text{out}}^{i}d^{i})p(\psi_{\text{out}}^{M}|d^{M}), \qquad (74)$$

where $M + 1$ denotes the number of inner areas in the total domain. To use this we can start from the back [i.e., $p(\psi_{\text{out}}^{M}|d^{M})$] and work from there through the whole domain. The difficulty in applying this formalism is in how to choose the conditional densities $p(\psi_{\text{in}}^{i}|\psi_{\text{out}}^{i}d^{i})$. Here we want to introduce concepts like smoothness, but it is not straightforward to put these concepts in probabilistic form. A simplified approach that includes localization follows.

First local weighting is performed (i.e., only observations relatively close to each model grid point are taken into account to calculate the local weights). Then we use the following procedure:

1) Start from a point somewhere in the domain and choose the particle with highest weight. Decrease that weight by $1/N$ (because the particle is chosen deterministically).
2) Walk to a next grid point and use the same particle if the weight is still high (i.e., $w_i > 1/N$). If the weight is low $w_i < 1/N$ choose another particle there which has high weight, and is close to the original particle at the previous grid point. This is a form of inplementing the conditional density using as outer domain only the point next to the inner point.
3) Repeat step 2 until all grid points have been visited and a particle is attached to each grid point. This is the first new member.
4) Repeat steps 1, 2, and 3 until N particles have been created.

Experience has shown us that this procedure works nicely for the first few new particles, but generates less smooth particles toward the end, because less particles can be chosen from at a breakpoint where w_i becomes smaller than 1.

3) Using an approximate smooth solution

As previously mentioned, one needs guidance on how to choose the different particles in different parts of the domain to generate smooth global particles. One could use another approximate solution that is smooth to provide this guidance (e.g., the EnKF solution). In that case step 2 above is replaced by the following:

- Walk to a next grid point and use the same particle if the weight is still high (i.e., $w_i > 1/N$). If the weight is low ($w_i < 1/N$), choose another particle there that has high weight and is close to one chosen EnKF member.

One could choose this EnKF member to have the highest global weight for the first global particle, and move to the EnKF member lower in weight rank for the next global particle, and so on. The advantage of this procedure is that when the likelihood of all members is very low in a certain part of the domain one can fall back on the EnKF solution, which is likely to be closer to the observations there.

4) Discontinuous forcing

Instead of trying to smooth discontinuous state vectors, one can try to put the discontinuity in the forcing. Because the forcing is integrated at each time step the influence on the model fields themselves is rather smooth. This could be implemented as follows:

1) Integrate the particles forward from time $n - 1$ to the new observations at time n. Make sure that the random forcing used in each particle is either stored or repeatable.
2) Do local weighting, and attach these weights to the random forcing field used between time $n - 1$ and time n.
3) Resample the forcing fields used, and use these to integrate the particles again from time $n - 1$ to time n.

This method assumes that the random forcing fields are causing the large weight variance, and not the particles at $n - 1$. The latter is changed by the auxiliary particle filter. The method has in common with that method that it needs $2N$ model integrations. In fact, one can imagine a combination of these two. An advantage is that a smooth transition in time is present at measurement times, so the method can be considered a smoother in that aspect.

e. Kernel dressing

Another approximation is to dress each particle with a continuous pdf to approximate the true continuous pdf using a standard kernel technique (see, e.g., Silverman 1986). Usually a Gaussian pdf is chosen since that has only two parameters to be estimated. In the Gaussian case the mean of the Gaussian is the particle state, and the covariance can be taken as a factor smaller than 1 times the covariance of the full ensemble. This results in a so-called Gaussian mixture model.

The idea is to represent the prior pdf as

$$p(\psi) = \sum_{i=1}^{N} \frac{1}{N} N(\psi_i, \Sigma_i). \qquad (75)$$

The $N(\psi_i, \Sigma)$ stands for the Gaussian (normal) density with mean particle i, ψ_i, and covariance Σ_i. Anderson and Anderson (1999) use a simplification of this general method because of Fukunaga (1972) in which all Σ_i are taken as a factor α times the full covariance of the prior ensemble. They apply this method to the Lorenz-63 and -80 models. The factor α is taken as a tuning parameter that takes care of the fact that the spread of the Gaussian pdf around each particle is smaller than that of the full ensemble, while at the same time it should prevent ensemble collapse by the filter being overconfident, showing up as a too small prior error covariance.

Using Bayes's theorem and assuming Gaussian observational errors, leads to a posterior pdf, which is again a sum of Gaussian pdfs. It is easy to show that the posterior pdf can be written as

$$p(\psi|d) = \sum_i c_i N(\nu_i, \Sigma_{\text{new}}), \tag{76}$$

where ν_i is the mean of the new Gaussian pdf i, Σ_{new} is the new posterior covariance (identical for each Gaussian), and c_i is a normalization constant that comes from the product of each prior Gaussian with the likelihood. Here c_i is given by

$$c_i = \frac{1}{(2\pi)^{d/2}|\alpha\Sigma + R|} \exp\left\{-\frac{1}{2}[H(\psi_i) - d]^{\mathrm{T}}(\alpha H\Sigma H^{\mathrm{T}} + R)^{-1}[H(\psi_i) - d]\right\}, \tag{77}$$

where R is the error covariance of the observations. The relative weight of each Gaussian in the posterior pdf is given by this c_i as

$$w_i = \frac{c_i}{\sum_j c_j}. \tag{78}$$

Interestingly, these weights remind us of the simple importance sampling in (13), but now the prior error covariance (multiplied by a factor of α) is added to the error covariance of the observations. This means that the likelihood has been broadened by the extra prior error covariance and that filter degeneracy is less likely to occur. The procedure is depicted in Fig. 7.

To better understand what happens, consider the simple scalar case in which the prior is distributed as $N(\mu, P)$ and the likelihood is $N(d, R)$. Bayes's theorem tells us that the posterior is obtained by multiplying these two. Concentrating on the exponents we have

$$\frac{(\psi - \mu)^2}{P} + \frac{(d - \psi)^2}{R} = \tag{79}$$

$$\left(\frac{1}{P} + \frac{1}{R}\right)\psi^2 - 2\left(\frac{\mu}{P} + \frac{d}{R}\right) + \frac{\mu^2}{P} + \frac{d^2}{R} = \cdots = \tag{80}$$

$$\frac{(\psi - \mu_n)^2}{P_n} + \frac{(d - \mu)^2}{R_n}, \tag{81}$$

where

$$\mu_n = \frac{R\mu}{P + R} + \frac{Pd}{P + R} \tag{82}$$

and

$$\frac{1}{P_n} = \frac{1}{P} + \frac{1}{R}, \tag{83}$$

and finally

$$R_n = P + R. \tag{84}$$

Hence, as a function of ψ the pdf has smaller variance, while as a function of the observations d the pdf is wider. So the posterior pdf around each particle is narrower than the prior pdf around that particle, and the likelihood

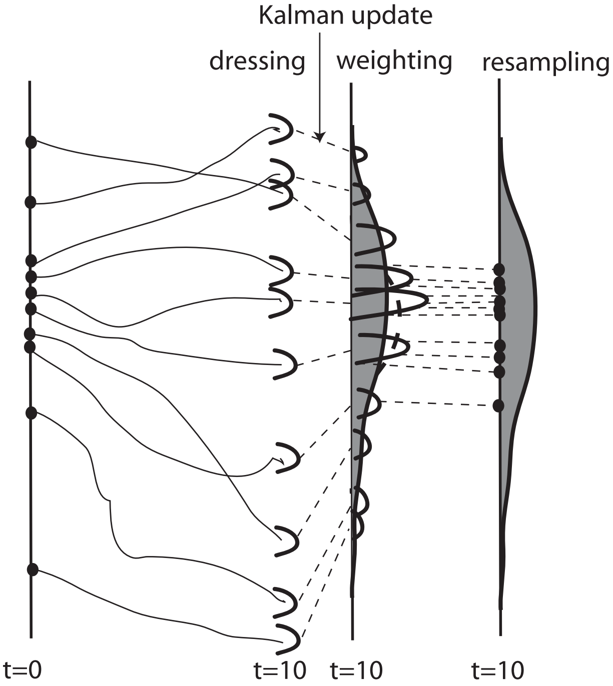

FIG. 7. Kernel dressing. At analysis time each particle is dressed with a kernel (e.g., a Gausssian). Then each kernel is updated using the Kalman filter update. The next step is weighting the kernel with the likelihood of the Kalman posterior. Finally, samples are drawn from this new posterior density. Note that the new particles are not identical copies of the old particles, but drawn from the continuous posterior pdf.

is wider. Furthermore, the posterior mean has moved toward the observation. Note that, since the prior variance is smaller than the variance of the full ensemble, the prior is not affected that much by the observation. That is taken care of by the relative weights of the individual Gaussians in the mixture. In a true particle filter the prior pdf is a sum of delta functions, so the prior variance of each particle P is infinitely small. In that case the posterior pdf remains a delta function, the mean does not move, and the likelihood does not change. Herewith we can understand the smooth transition from continuous representations of the pdf to particle representations.

Anderson and Anderson (1999) generate the new ensemble that represents the posterior pdf by first drawing randomly from the distribution of w_i (i.e., choose a specific Gaussian) and then drawing a new particle randomly from that specific Gaussian.

A few problems exist when trying to apply this method to large-scale problems. First, manipulating the large error covariances becomes problematic, and second, sampling from a large-dimensional pdf, even if it is a Gaussian, is hard.

Bengtsson et al. (2003) combine this approach with localization (see section 5d) to try to solve these problems. Furthermore, they estimate local (in state space) error covariances around each prior particle instead of using the same fraction of the full error covariance. Bengtsson et al. (2003) use the following steps:

1) First choose randomly L particles (typically 10–40) from the full N (typically 100) ensemble to represent the centers of the Gaussians in the Gaussian mixture.
2) Choose the M nearest particles (typically 25) to each center to determine the local (in state space) error covariance for that Gaussian. Bengtsson et al. (2003) use the Euclidian distance, but other choices can be made.
3) Calculate the weights w_i according to (78).
4) Choose randomly one of the Gaussians from the distribution of w_i.
5) From that Gaussian choose one of the M members with probability $1/M$, call that member ψ_I and update each of the M members similarly to the update used in the EnKF (Evensen 1994; Burgers et al. 1998):

$$\psi_i^{\text{new}} = \psi_i + K[d + \epsilon_i - H(\psi_I)], \tag{85}$$

where ϵ_i is chosen from $N(0, R)$.
6) Repeat steps 4 and 5 until N new members have been chosen.

With this approach, the computational efficiency is similar to that of the EnKF. (Note the simularity of this method with the particle filter with the EKF or the UKF as proposal.) Localization can now be implemented by using the above procedure at each grid point, using only observations close to that grid point by putting correlations over large spatial differences equal to zero. A problem with this approach is, however, that since different Gaussians are chosen at different grid points the new global particles will not be smooth. Note that we have encountered the similar problem with the resampling scheme with localization. The solution proposed by Bengtsson et al. (2003) is the following. First, each particle is spitted in two parts, one inside $\psi^{(i)}$ and one outside $\psi^{(o)}$ the local neighborhood used in the localization. The posterior density for state vector ψ can be written as

$$p(\psi|d) = p(\psi^{(i)}|\psi^{(o)}, d)p(\psi^{(o)}|d). \tag{86}$$

Bengtsson et al. (2003) propose to use the nonlocal EnKF particles to sample $p(\psi^{(o)}|d)$, and take a linear combination of that particle with the local particle obtained with the local Gaussian mixture model inside the local neighborhood. In this way a hybrid method is obtained that to some extent preserves non-Gaussianity locally. For details of this linear combination the reader is referred to Bengtsson et al. (2003). They show that when applying the above to the Lorenz-63 and -80 models the local variant is outperformed by the EnKF, but the hybrid method outperforms the EnKF in the higher dimensional Lorenz-80 model.

Hoteit et al. (2008), probably unaware of Bengtsson et al. (2003), successfully used the Gaussian mixture method with 50 particles on a full-blown ocean general circulation model with about 1.5 million prognostic variables in an identical twin setting, without localization. To be able to handle the covariances a low-rank approximation was made. The parameter α used for the size of the kernel error covariances relative to the full error covariance is taken as 0.16.

It is interesting to try to understand why Hoteit et al. (2008) do not need any localization to avoid filter divergence. The reason seems to be the implicit covariance inflation in the resampling step. After showing that naive resampling of the posterior Gaussian mixture gives rise to a biased resampled posterior, they propose a correction to the posterior error covariances in each Gaussian of the mixture. It turns out that this correction is such that the resampling works as if the mixture behaves like a sum of delta functions (i.e., like a particle filter). Since this is undesirable from the viewpoint of filter divergence, Hoteit et al. (2008) decide to put more weight on avoidance of filter divergence, at the cost of allowing for a biased posterior pdf. The outcome of this careful discussion is that the scaling factor α becomes a tuning factor, just like in Anderson and Anderson.

Furthermore, Hoteit et al. (2008) allow for a "partial resampling step," in which noise is added to the particles without performing a true resampling, so without changing the weights of the particles. (The word jitter might be more appropriate because the weights remain unchanged.) The full implementation becomes a few "partial resampling" filter steps, followed by an actual resampling filter step. Finally, it might be important that the first assimilation step provided a large error reduction (i.e., the mean of the ensemble came relatively close to the actual "truth."). Since the model errors were neglected, a perfect model is assumed and the filter did not have too much difficulty is keeping close to the truth. Nonetheless, the model resolution of ¼°, the 5-day assimilation interval, and the fact that only sea surface height fields were assimilated make the filtering problem highly nonlinear, and the model evolution is expected to be sensitive to small deviations from the truth.

Miller et al. (1999) tried to study the evolution of the full pdf in data assimilation in small-dimensional strongly nonlinear systems. They used the Epanechnikov kernel instead of a Gaussian kernel. They apply this method to the double-well problem, the Lorenz-63 model, and to a truncated spectral barotropic model with dimension 44. Since the goal was to try to simulate the evolution of the full pdf, 10 000 particles were used.

f. The maximum entropy particle filter

In the maximum entropy particle filter one tries to incorporate the pdf of the model without observations in the particle filter. In this way the degrees of freedom in the data-assimilation problem can be greatly reduced, and the potential of filter divergence might reduce, too. The basic idea is that the pdf without observations is given by a background pdf q. One can take for q a Gaussian mixture, but there is no need to do so. The background pdf is used to describe the "model climatology," so, instead of Gaussian mixture models that dress each particle with a separate pdf, one tries to describe the model climatology by this background pdf.

The method was first put forward by Kim et al. (2003) and Eyink and Kim (2006) and has received little attention so far in the geophysical literature. However, since it is one of the methods in between the EnKF methods and the full particle filter, this might change. Also in maximum entropy filters an ensemble of particles is run up to the next observation. In the EnKF and its variants, the assumption is that the prior can be approximated by a Gaussian, and the analysis is performed as if the pdf is Gaussian. In the maximum entropy filter the assumption is that the prior pdf is as close as possible to q, but at the same time it enforces that the mean and the covariance of the pdf in observation space is obtained directly from the ensemble. Actually, it is easy to show that when q is a simple Gaussian, the maximum entropy filter is just an EnKF. To determine the prior pdf from the ensemble we do the following. The closeness to q is expressed as the relative entropy:

$$E(p, q) = \int p \log\left(\frac{p}{q}\right)\, d\psi. \tag{87}$$

Furthermore, we use the constraints that p is a pdf that integrates to 1:

$$\int p(\psi)\, d\psi = 1, \tag{88}$$

that the mean of the observation operator η is found from the ensemble:

$$\int H(\psi)p(\psi)\, d\psi = \frac{1}{N}\sum_i H(\psi_i) = \eta, \tag{89}$$

and that the covariance *function* of the observation operator Σ is found from the ensemble:

$$\int H(\psi)H^{\mathrm{T}}(\psi)p(\psi)\, d\psi = \frac{1}{N-1}\sum_i H(\psi_i)H^{\mathrm{T}}(\psi_i) = \Sigma. \tag{90}$$

The constraints are built into the minimization through Lagrangian multipliers θ, λ, and Λ, so we minimize

$$\begin{aligned}\tilde{E}(p, q) = \int p \log\left(\frac{p}{q}\right) d\psi + \theta\left[1 - \int p(\psi)\, d\psi\right] \\ + \lambda\left[\eta - \int H(\psi)p(\psi)\, d\psi\right] \\ + \Lambda^{\mathrm{T}}\left[\Sigma - \int H(\psi)H^{\mathrm{T}}(\psi)p(\psi)\, d\psi\right],\end{aligned} \tag{91}$$

with respect to p, θ, λ, and Λ. This leads to

$$\int\left[\log\frac{p}{q} + 1 + \theta + \lambda H(\psi) + \frac{1}{2}H(\psi)\Lambda H^{\mathrm{T}}(\psi)\right]\delta p\, d\psi = 0, \tag{92}$$

so that

$$p(\psi, \lambda, \Lambda) = \frac{1}{Z(\lambda, \Lambda)}\exp\left[\lambda H(\psi) + \frac{1}{2}H(\psi)\Lambda H^{\mathrm{T}}(\psi)\right]q(\psi), \tag{93}$$

where Z is the normalization factor. The Lagrangian multipliers are found from maximizing the expected value of p, or rather $\log p$, so

$$\max\left(\eta\lambda + \frac{1}{2}\Sigma^{\mathrm{T}}\Lambda - \log Z\right). \tag{94}$$

Note that the normalization factor Z has to be determined explicitly. However, Eyink and Kim (2006) derive analytical expressions for Z when q is a Gaussian mixture.

Now that we have found the prior pdf, the analysis is discussed. The idea is to use Bayes's theorem to update λ and Λ, which, for Gaussian error statistics for the observations and the maximum entropy density (93), leads us to

$$\lambda^{\text{new}} = \lambda + R^{-1}d \tag{95}$$

and

$$\Lambda^{\text{new}} = \Lambda - R^{-1}. \tag{96}$$

The result is a maximum entropy density of the same form as (93), but with the new parameters λ^{new} and Λ^{new}, and a new normalization factor, as can be verified easily. The last step is to sample from this new pdf. This is not so easy, but Eyink and Kim (2006) discuss feasible ways to do this.

The authors also discuss the mean-field filter, in which only the mean of the ensemble is updated. The algorithm closely follows that of the maximum entropy particle filter and is not discussed here. Obviously, the update step is much cheaper, but less accurate.

Interesting is that when the background probability is a single Gaussian, and the mean and the covariance of the ensemble are taken as constraints in the prior maximum entropy pdf, the method is identical to the EnKF. Thus, it can be considered as an extension to that method.

The methods were tested on the double-well and the Lorenz-63 problems. With a well-chosen background pdf q, both methods outperformed the EnKF and the SIR for ensemble sizes 10–100. The conclusion is that when knowledge of this background pdf is available use it, and the maximum entropy particle filter is one of the methods to do so.

6. Summary and conclusions

It is not easy to write a concluding section on the strongly developing field of particle filtering for geophysical applications. It has become clear that simple importance sampling, even with resampling, is not enough to be applicable for large-scale geophysical applications with $O(100)$ particles. The weights of the different particles vary too much to maintain a reasonable description of even the lowest-order moments of the model pdf. The paper by Snyder et al. (2008) gives a firm justification for this statement, based on their idea that for a large number of observations the weights will tend to diverge, no matter how close the particles are. Or, to put it more directly: each observation that has important information on the pdf will increase the variance in the weights of the particles, so a large number of independent observations will always tend to increase the variance in the weights significantly (A. Lorenc 2009, personal communication).

Van Leeuwen (2003a) argues that the usual Gaussian tails for observation pdfs are too small and broader pdfs should be used, but that is probably not enough to avoid filter divergence. This means that other more sophisticated variants of particle filtering have to be tried. Among these, using another proposal density than the prior seems promising, also because it has been used with a lot of success in the statistical literature. The strength comes from the use of the new observations to generate particles with high likelihood directly, at the expense of extra calculations to compensate for the artificially introduced weights. But, it must also be said that in the statistical literature it is not considered the Holy Grail (see e.g., Doucet et al. 2001).

Another method that also uses the new observations is the guided particle filter, in which the particles are confronted with the observations before the actual measurement time to guide them in the right direction. Little experience does exist with this method on large-scale problems.

In another class of particle filters a first shot of particles is done to probe where the observations are, followed by a new run of particles from the same initial condition. Of these methods, the auxilary particle filter used the new weights to resample the ensemble at initial time, while the backtracking particle filter moves back a few measurement times and proceeds from there with a larger ensemble.

A large part of research in particle filtering is devoted to trying to introduce extra approximations to importance (re)sampling. Among these are methods that try to conserve only the mean and the covariance of the posterior pdf. In Gaussian resampling this is done directly, while in the merging particle filter linear combinations of particles are created such that the mean and the covariance are preserved. Kernel dressing methods, like Gaussian mixture models, in which each particle is dressed with a continuous pdf, have been proposed and interesting results have been obtained for large-scale systems. One can also fight the degeneracy by combining different observations in batches. For instance, amplitudes of EOFs could be assimilated. By doing this part of the information present in the observations is lost, but the dominant variability in them is kept.

Of a completely different nature is localization. Localizations has been crucial in the development of large-scale applications of EnKFs. Is was argued that localization is not straightforward in combination with resampling due to the necessary ''gluing together'' of different particles. Several ideas to implement this gluing have been discussed, but none of them seems to work in primitive equation models where too-strong gradients induced by the gluing procedure tend to lead to spurious gravity wave formation and degradation of the update. Finally, the maximum entropy filter was discussed in which use is made of a background pdf to which the model relaxes when no observations are present. Interestingly, this builds a bridge to dynamical system theory, since it tries to use the concept of the attractor of the dynamical system described by the model.

After reviewing all these methods it is difficult to say what is the most promising research direction of the field. All methods discussed have their merits, and perhaps combinations of them will help us to make progress. Also combinations with 4D-Var might be fruitful. For instance, Lorenc (2003) shows how particles can be used to bring flow-dependent information in the background covariance in strong-constraint applications. Particle filtering might also help to construct weak-constraint variational methods. And vice versa, techniques developed for 4D-Var might help to construct efficient particle filters. At this moment the field is so young and exciting that anything can happen!

Acknowledgments. The author thanks numerous colleagues for stimulating discussions, especially Arnoud Doucet and Marc Bocquet, and the reviewers. Special thanks are given to Kayo Ide, who pushed me into this endeavor.

REFERENCES

Alexander, F. J., G. L. Eyink, and J. M. Restrepo, 2005: Accelerated Monte Carlo for optimal estimation of time series. *J. Stat. Phys.,* **119,** 1331–1345, doi:10.1007/s10955-005-3770-1.

Anderson, J. L., and S. L. Anderson, 1999: A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Mon. Wea. Rev.,* **127,** 2741–2758.

Ballabrera-Poy, J., A. J. Busalacchi, and R. Murtugudde, 2001: Application of a reduced-order Kalman filter to initialize a coupled atmosphere–ocean model: Impact on the prediction of El Niño. *J. Climate,* **14,** 1720–1737.

Bengtsson, T., C. Snyder, and D. Nychka, 2003: Toward a nonlinear ensemble filter for high-dimensional systems. *J. Geophys. Res.,* **108,** 8775, doi:10.1029/2002JD002900.

——, P. Bickeel, and B. Li, 2008: Curse-of-dimensionality revisited: Collapse of the particle filter in very large systems. *Probability and Statistics: Essays in Honor of David A. Freedman,* D. Nolan and T. Speed, Eds., Vol. 2, Institute of Mathematical Statistics, 316–334.

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

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

Brasseur, P., and Coauthors, 2005: Data assimilation for marine monitoring and prediction: The MERCATOR operational assimilation systems and the MERSEA developments. *Quart. J. Roy. Meteor. Soc.,* **131,** 3561–3582.

Brusdal, K., J. M. Brankart, G. Halberstadt, G. Evensen, P. Brasseur, P. J. van Leeuwen, E. Dombrowsky, and J. Verron, 2003: A demonstration of ensemble-based assimilation methods with a layered OGCM from the perspective of operational ocean forecasting systems. *J. Mar. Syst.,* **40–41,** 253–289.

Budhiraja, A., L. Chen, and C. Lee, 2007: A survey of numerical methods for nonlinear filtering problems. *Physica D,* **230,** 27–36.

Burgers, G., P. J. van Leeuwen, and G. Evensen, 1998: Analysis scheme in the ensemble Kalman filter. *Mon. Wea. Rev.,* **126,** 1719–1724.

Chorin, A. J., and P. Krause, 2004: Dimensional reduction for a Bayesian filter. *Proc. Natl. Acad. Sci. USA,* **101,** 15 013–15 017, doi:10.1073/pnas.0406222101.

Doucet, A., N. de Freitas, and N. Gordon, 2001: *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag, 581 pp.

Dowd, M., 2007: Bayesian statistical data assimilation for ecosystem models using Markov chain Monte-Carlo. *J. Mar. Syst.,* **68,** 439–456.

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

——, 2006: *Data Assimilation: The Ensemble Kalman Filter*. Springer, 280 pp.

Eyink, G., and S. Kim, 2006: A maximum entropy method for particle filtering. *J. Stat. Phys.,* **123,** 1071–1128, doi:10.1007/s10955-006-9124-9.

Fukunaga, K., 1972: *Introduction to Statistical Pattern Recognition*. Academic Press, 369 pp.

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

Harlim, J., and B. R. Hunt, 2007: A non-Gaussian ensemble filter for assimilating infrequent noisy observations. *Tellus,* **59A,** 225–237.

Hastings, W. K., 1970: Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika,* **57,** 97–109.

Hoteit, I., D. T. Pham, G. Triantafyllou, and G. Korres, 2008: A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Mon. Wea. Rev.,* **136,** 317–334.

Houtekamer, P. L., and H. L. Mitchell, 1998: Data assimilation using an ensemble Kalman filter technique. *Mon. Wea. Rev.,* **126,** 796–811.

Jazwinski, A. H., 1970: *Stochastic Processes and Filtering Theory*. Academic Press, 376 pp.

Julier, S. J., and J. K. Uhlmann, 1997: A new extension of the Kalman filter to nonlinear systems. *Signal Processing, Sensor Fusion, and Target Recognition VI,* I. Kadar, Ed., International Society for Optical Engineering (SPIE Proceedings, Vol. 3068), 182–193.

Kalnay, E., 2004: *Atomospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 341 pp.

Kim, S., G. L. Eyink, J. M. Restrepo, F. J. Alexander, and G. Johnson, 2003: Ensemble filtering for nonlinear dynamics. *Mon. Wea. Rev.,* **131,** 2586–2594.

Kitagawa, G., 1996: Monte-Carlo filter and smoother for non-Gaussian non-linear state-space models. *J. Comput. Graph. Stat.,* **5,** 1–25.

Kivman, G. A., 2003: Sequential parameter estimation for stochastic systems. *Nonlinear Processes Geophys.,* **10,** 253–259.

Klaas, M., N. de Freitas, and A. Doucet, 2005: Towards practical N^2 Monte Carlo: The marginal particle filter. *Proc. 21st Annual Conf. on Uncertainty in Artificial Intelligence (UAI-05),* Arlington, VA, AUAI Press, 308–315.

Krol, M., P. J. van Leeuwen, and J. Lelieveld, 1998: Global OH trend inferred from methylchloroform measurements. *J. Geophys. Res.,* **103D,** 10 697–10 711.

Lorenc, A., 2003: The potential of the Ensemble Kalman filter for NWP: A comparison with 4D-VAR. *Quart. J. Roy. Meteor. Soc.,* **129,** 3183–3203.

Losa, S. N., G. A. Kivman, J. Schroeter, and M. Wenzel, 2003: Sequential weak constraint parameter estimation in an ecosystem model. *J. Mar. Syst.,* **43,** 31–49.

Lui, J. S., and R. Chen, 1998: Sequential Monte-Carlo methods for dynamical systems. *J. Amer. Stat. Assoc.,* **90,** 567–576.

Metropolis, N., and S. Ulam, 1944: The Monte Carlo method. *J. Amer. Stat. Assoc.,* **44,** 335–341.

Miller, R. N., E. F. Carter, and S. T. Blue, 1999: Data assimilation into stochastic models. *Tellus,* **51A,** 167–194.

Musso, C., N. Oudjane, and F. Le Grand, 2001: Improving regularized particle filters. *Sequential Monte-Carlo Methods in Practice,* A. Doucet, N. de Freitas, and N. Gordon, Eds., Springer-Verlag, 247–271.

Nakano, S., G. Ueno, and T. Higuchi, 2007: Merging particle filter for sequential data assimilation. *Nonlinear Processes Geophys.,* **14,** 395–408.

Ott, E., B. R. Hunt, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalney, D. J. Patil, and J. A. Yorke, 2004: A local ensemble Kalman filter for atmospheric data assimilation. *Tellus,* **56,** 415–428.

Papadakis, N., 2007: Assimilation de donnees images: Application au suivi de courbes et de champs de vecteurs. Ph.D. thesis, University of Rennes I, 240 pp.

Papoulis, A., 1995: *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 685 pp.

Pham, D. T., 2001: Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Mon. Wea. Rev.,* **129,** 1194–1207.

Pitt, M. K., and N. Shephard, 1999: Filtering via simulation: Auxilary particle filters. *J. Amer. Stat. Assoc.,* **94,** 590–599.

Robert, C. P., and G. Cassela, 2004: *Monte-Carlo Statistical Methods*. Springer-Verlag, 645 pp.

Silverman, B. W., 1986: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 175 pp.

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

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

van der Merwe, R., A. Doucet, J. F. G. de Freitas, and E. Wan, 2000: The unscented particle filter. Tech. Rep. CUED/F-INFENG/TR 380, Cambridge University Statistical Department, 45 pp.

van Leeuwen, P. J., 2002: Ensemble Kalman filters: Sequential importance resampling and beyond. *Proc. ECMWF Workshop on the Role of the Upper Ocean in Medium and Extended Range Forecasting,* Reading, United Kingdom, ECMWF, 46–56.

——, 2003a: A truly variance-minimizing filter for nonlinear dynamics. *Mon. Wea. Rev.,* **131,** 2071–2084.

——, 2003b: Nonlinear ensemble data assimilation for the ocean. *Proc. Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean,* Reading, United Kingdom, ECMWF, 265–286.

——, and G. Evensen, 1996: Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Wea. Rev.,* **124,** 2898–2913.

Vossepoel, F. C., and P. J. van Leeuwen, 2007: Parameter estimation using a particle method: Inferring mixing coefficients from sea-level observations. *Mon. Wea. Rev.,* **135,** 1006–1020.

West, M., 1993: Approximating posterior distributions by mixtures. *J. Roy. Stat. Soc.,* **55,** 409–422.

Wikle, C. K., and L. M. Berliner, 2007: A Bayesian tutorial for data assimilation. *Physica D,* **230,** 1–16.

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

Zhou, Y., D. McLaughlin, and D. Entekhabi, 2006: Assessing the performance of the ensemble Kalman filter for land surface data assimilation. *Mon. Wea. Rev.,* **134,** 2128–2142.