

Implicit particle methods and their connection with variational data assimilation

Ethan Atkins¹, Matthias Morzfeld² and Alexandre J. Chorin^{1,2}

Department of Mathematics

¹University of California, Berkeley

and

²Lawrence Berkeley National Laboratory

Abstract

The implicit particle filter is a sequential Monte Carlo method for data assimilation that guides the particles to the high-probability regions via a sequence of steps that includes minimizations. We present a new and more general derivation of this approach and extend the method to particle smoothing as well as to data assimilation for perfect models. We show that the minimizations required by implicit particle methods are similar to the ones one encounters in variational data assimilation and explore the connection of implicit particle methods with variational data assimilation. In particular, we argue that existing variational codes can be converted into implicit particle methods at a low cost, often yielding better estimates, that are also equipped with quantitative measures of the uncertainty. A detailed example is presented.

1 Introduction

The goal in data assimilation is to estimate the state of a system by combining information from incomplete and noisy observations of this state with information from a possibly uncertain numerical model. This can be done by analyzing the conditional probability density function (pdf) for the state given the observations [5, 12, 16, 27]. If the model is linear and the observations are linear functions of the state and if, in addition, all error statistics are Gaussian, then the state conditioned on the data is also Gaussian. In this case, all one needs to know is the mean and covariance of the state and both can be computed by the Kalman filter [25, 26]. However, many problems are nonlinear and non-Gaussian, and methods that assume a nearly linear model or nearly Gaussian errors, such as the ensemble Kalman filters [14, 16], can perform poorly if these assumptions are violated [36].

For that reason we focus on variational data assimilation and particle methods, which do not require Gaussianity or linearity approximations. In variational data assimilation one finds the most likely state given the observations, i.e. the mode of the conditional pdf, through minimization of a suitable cost function [1, 11, 47, 48]. While there is no guarantee that the most likely state is found (the minimization may not converge to the global minimum), variational methods have proven effective in many applications and they are widely used in geophysical data assimilation, e.g. in numerical weather prediction.

Particle methods assimilate the data via Monte Carlo importance sampling [12, 22, 43]. Most particle methods first sample a given importance function and then use the data to assign weights to each sample, so that the weighted samples, called particles in this context, form an empirical estimate of the conditional pdf. The difficulty is that the importance function and the conditional pdf can become nearly mutually singular as the dimension of the state space becomes large, which leads to a representation of the conditional pdf by a single and often uninformative particle [2, 45]. This effect is known as sample impoverishment and has been an obstacle to the application of particle methods to geophysical data assimilation, where the state dimension is typically large.

Sample impoverishment can be delayed or even prevented if the overlap between the importance function and the conditional density is increased, and much effort has been invested to find an importance function that can work in large dimensional problems, particularly in geophysical applications [3, 13, 24, 49, 50]. The implicit particle filter [7, 8, 39] attempts to prevent sample impoverishment by focussing the particles to regions of high probability. These regions are identified through particle-by-particle minimizations. Since the minimization for each particle of an implicit particle filter is similar to the minimizations one encounters in variational data assimilation, one can expect a link between these two approaches. We will describe this link in this paper.

The paper is structured as follows. In section 2, we review how to sample a given pdf using implicit sampling by first finding the mode of the pdf and then generating samples in the neighborhood of this mode. In section 3, we apply implicit sampling to the conditional pdf for data assimilation to derive the implicit particle smoother that assimilates all available observations at the same time, and the implicit particle filter that assimilates the data sequentially. In section 4, we make the connection between these implicit particle methods and variational data assimilation, and show how existing variational codes can be used for the efficient implementation of implicit

particle methods. In section 5 we present an application of implicit particle methods and discuss their variational aspects. Conclusions are offered in section 6.

2 Implicit sampling

Importance sampling is a Monte Carlo method that samples a hard-to-sample pdf p using an easy-to-sample pdf p_0 [5,12,20,23,29]. In this context, the density p we want to sample, but cannot sample directly, is called the target density and the density p_0 we actually use to obtain a sample is called the importance density (or importance function). Suppose we are interested in the pdf p of a d -dimensional, continuous random variable x . Importance sampling generates samples $X \in R^d$ (we use capital letters to denote realizations of random variables) from the known importance function p_0 with weights

$$w(X) = \frac{p(X)}{p_0(X)}, \quad (1)$$

such that the weighted samples form an empirical estimate of the target pdf p . This empirical estimate approximates the target pdf weakly, i.e.

$$\hat{E}_M(u) = \frac{\sum_{j=0}^M u(X_j)w(X_j)}{\sum_{j=0}^M w(X_j)}, \quad (2)$$

converges almost surely to the expected value, $E_p[u(x)]$, of a function u with respect to the density p , as the number of samples, M , approaches infinity. It should be clear that the support of p_0 must include the support of p . Moreover, importance sampling works even if the target pdf is known only up to a multiplicative constant, because this constant can be eliminated by scaling the weights so that their sum equals one.

The efficiency of importance sampling depends on the choice of the importance function. For example, samples with a small weight contribute very little to the approximation of the expected value in (2), so that the computational effort spent on generating these low probability samples is mostly wasted. To avoid spending computation time on low probability samples, one needs to find an importance function p_0 such that the variance of the weights in (1) is small, i.e. all samples contribute equally to the sum in (2). This means in particular that the importance function must be large in the regions where the target density is large. Implicit sampling is an importance

sampling method that defines the importance function implicitly by an algebraic equation. We will now show that this importance function is large where p is large, i.e. that the samples we obtain have a high probability.

We write the pdf we are interested as $p \propto e^{-F(x)}$ (this is natural in data assimilation, see section 3.1) and, for a moment, assume that

$$F(x) = -\log p(x), \quad (3)$$

is convex (we will relax this assumption later on). The region where p is large, and where the high-probability samples lie, is the neighborhood of the mode of p . Using the log-transformation (3), we can identify this region through minimization of F , and define

$$\phi_F = \min F.$$

To obtain a sample in the high-probability region, we pick a reference variable $\xi \sim g$, with a known pdf $g \propto e^{-G(\xi)}$, and which is easy to sample. We then map the high probability region of the reference variable ξ to the high probability region of X . This can be done by solving the algebraic equation

$$F(X) - \phi_F = G(\xi) - \phi_G, \quad (4)$$

where $G = -\log g$ is chosen to be convex and $\phi_G = \min G$. Note that the above scalar equation is underdetermined (it connects the d elements of x to the d elements ξ) and solvable since F and G are infinite at $\pm\infty$, so that the left and right hand sides of (4) both range from $[0, \infty)$. We can thus find a sample X by solving (4) with a one-to-one and onto mapping

$$\psi : \xi \rightarrow X. \quad (5)$$

A sample of the reference density ξ is likely to lie near the mode of g , so that the right hand side of (4) is likely to be small. Equation (4) and the mapping ψ thus imply that, for a high-probability sample of ξ , the function $F(X)$ is close to its minimum ϕ , which implies that X is in the region where p is large. The map ψ thus maps the high-probability region of the reference variable ξ to the high-probability region of X , so that, with a high probability, we obtain a high probability sample.

The reference variable ξ and the map ψ in (5) define the importance function

$$p_0(X(\xi)) \propto \frac{\exp(-G(\xi))}{|J(\xi)|},$$

where $J = \det(\partial X/\partial \xi)$ is the Jacobian of ψ . Using (4), the importance function can be written in terms of $X = \psi(\xi)$

$$p_0(X) \propto \frac{\exp(-F(X) + \phi_F - \phi_G)}{|J(X)|}, \quad (6)$$

and, by using (1), we find that the weight of the sample X is

$$w(X) \propto e^{-\phi_F + \phi_G} |J(X)|. \quad (7)$$

The variability in the weights is induced by the Jacobian of the map (the term involving the ϕ 's is constant among the samples and can be removed by scaling the weights so that their sum equals one). The only requirement on ψ is that it solves the undetermined equation (4). We thus have a lot of freedom in choosing this map and we can use this freedom to construct a map that keeps the variance of the weights small, and whose Jacobian is easy to compute. Various ways of doing this have been presented in [7, 8, 39] and we will review one of these maps in the example in section 5. What is important here is to realize that solving the algebraic equation (4) is cheap, once the minimum of F is found.

What is left to do is to choose a reference variable ξ . Equation (6) implies that the closer the pdf of the reference variable resembles the target density p , the more the importance function p_0 also resembles the target density. It is thus desirable to choose such a reference variable, however that might be impractical (because we typically do not know the target pdf in advance and because it is hard to sample the target pdf). In practice one should choose a reference density that is easy to sample and easy to minimize. For example, in [7, 8, 39], a Gaussian reference variable, $\xi \sim \mathcal{N}(0, I)$, was used and yielded good results (we denote a Gaussian variable with mean μ and covariance matrix Σ by $\mathcal{N}(\mu, \Sigma)$ and use I for the identity matrix of appropriate dimensions). A Gaussian reference variable does not imply that the target density is approximated by a Gaussian, since it is clear from (6) that the importance density is generally not Gaussian even if ξ is. Instead, each sample X is a function of a Gaussian reference sample.

We now relax the assumption that F is convex. If F is U -shaped, then the above construction works without modification. A scalar function F is called U -shaped if it is at least piecewise differentiable, its first derivative vanishes at a single point which is a minimum, F is strictly decreasing on one side of the minimum and strictly increasing on the other, and $F(X) \rightarrow \infty$ as $|X| \rightarrow \infty$; in the d -dimensional case, F is U -shaped if it has a single minimum and each intersection of the graph of the function $y = F(X)$ with

a vertical plane through the minimum is U -shaped in the scalar sense. If F is not U -shaped, but has only one minimum, one can replace it by a U -shaped approximation, say F_0 , and then apply implicit sampling as above. The error one makes by this approximation can be accounted for through reweighting [7]. If F has multiple minima (the target pdf p has more than one mode), then one can find local U -shaped approximations at each local minimum and apply implicit sampling to each local approximation. The errors one makes can be accounted for by reweighting of the samples.

3 Implicit sampling for data assimilation

We now apply implicit sampling to the conditional pdf for data assimilation and derive three implicit particle methods. Our derivation is more general than the ones presented in [7, 8, 39] and highlights the variational aspects of the implicit particle methods.

3.1 Problem formulation

We start with a review of the data assimilation problem to set up notation and terminology. In data assimilation, one is given an uncertain numerical model of a system and a stream of noisy data about its state, and one wants to use both to estimate the state of the system. The numerical model is a Markovian state space model

$$x_{j+1} = R_j(x_j) + G_j(x_j)Z_j, \quad (8)$$

where $j = 0, 1, 2, \dots$ can be thought of as discrete time; the state, x_j , is a d -dimensional real vector, R_j respectively G_j are given d -dimensional vector functions, respectively real $d \times d$ matrices and the Z_j 's are d -dimensional random variables. In geophysical applications, the numerical model often comes from discretizations of stochastic differential equations, in which case the Z_j 's are random vectors whose elements are independent normal variates [30], and we assume the Z_j 's to be Gaussian from now on. We assume further that at time $j = 0$ the pdf for the state x_0 is known and that the matrices G_j have full-rank. How to relax the latter assumption is described in [38].

The data

$$y_k = h(x_{n_k}) + V_k, \quad (9)$$

indexed by $k = 1, 2, \dots$, are regularly spaced, noisy measurements of the state, taken at times $n_k = kr$, where $r \geq 1$ is a positive integer (it is an easy

exercise to consider also the case when observations are irregularly spaced in time). In the above equation, h is a b -dimensional vector function and V_k is a b -dimensional random variable with a known pdf. We assume that the random variables V_k are independent of each other and also independent of the model noise Z_j . For notational convenience, we will write $x_{0:k}$ for the sequence of vectors x_0, \dots, x_k .

At time $n_m = n \cdot m$, $m \geq 1$, we have collected m observations $y_{1:m}$, and everything we know about the state trajectory $x_{0:n_m}$ is contained in the conditional pdf

$$p(x_{0:n_m}|y_{1:m}) = p(x_0) \frac{\prod_{j=1}^{n_m} p(x_j|x_{j-1}) \prod_{j=1}^m p(y_j|x_{n_j})}{p(y_{1:m})}. \quad (10)$$

Since we know $p(x_0)$, and can read $p(x_j|x_{j-1})$ and $p(y_j|x_j)$ from equations (8) and (9), we know this pdf up to the normalization constant $p(y_{1:m})$.

3.2 The implicit particle smoother

To assimilate all m available observations, we can apply implicit sampling to the conditional pdf in (10). Since an importance sampling scheme that assimilates more than one observation at a time is often called a particle smoother [12], we will call this method the implicit particle smoother.

The target pdf is the conditional pdf in (10), so that the function F of implicit sampling is

$$F(x_{0:n_m}) = -\log(p(x_{0:n_m}|y_{1:m})).$$

If V_k in (9) is Gaussian with mean zero and covariance matrix Q , then this F is

$$\begin{aligned} F(x_{0:n_m}) &= -\log(p(x_0)) \\ &+ \frac{1}{2} \sum_{j=0}^{n_m-1} (x_{j+1} - R_j(x_j))^T \Sigma_j^{-1} (x_{j+1} - R_j(x_j)) \\ &+ \frac{1}{2} \sum_{j=1}^m (y_j - h(x_{n_j}))^T Q^{-1} (y_j - h(x_{n_j})) + C, \end{aligned} \quad (11)$$

where $\Sigma_j = G_j(x_j)^T G_j(x_j)$, and where the value of the constant C is irrelevant (it will drop out in the normalization of the weights). We find the minimum ϕ_F of F using standard techniques, such as Newton's methods, quasi Newton methods or gradient descent (see e.g. [18, 40, 42]) and choose

a Gaussian reference variable $\xi \sim \mathcal{N}(0, I)$. In this case the algebraic equation (4) becomes

$$F(X) - \phi_F = \frac{1}{2} \xi^T \xi, \quad (12)$$

which we solve with a suitable mapping ψ (see [7, 8, 39]) for M independent realizations of ξ to obtain M weighted samples (particles), with weights given by (7). The M particles form an empirical estimate of the conditional pdf $p(x_{0:n_m})$, so that we have successfully assimilated the data. To compute a concrete state estimate, we can compute, for example, the weighted sample average to approximate the conditional mean $E(x_{0:n_m} | y_{1:m})$, which, in turn, is the minimum mean squared error estimate [5].

3.3 The implicit particle filter

Suppose we have assimilated m observations, for example by using the implicit particle smoother, and that a new observation y_{m+1} is now available. One can of course assimilate this observation by redoing the calculations of the previous section with $p(x_{0:n_{m+1}} | y_{1:m+1})$ replacing $p(x_{0:n_m} | y_{1:m})$, however this approach becomes impractical as we collect more and more data.

Alternatively, we can assimilate the data sequentially using the recursive formula for the conditional pdf (see [12])

$$p(x_{0:n_{m+1}} | y_{1:m+1}) = p(x_{0:n_m} | y_{1:m}) \frac{p(x_{n_m+1:n_{m+1}} | x_{n_m}) p(y_{m+1} | x_{n_{m+1}})}{p(y_{m+1} | y_{1:m})}.$$

Given a set of M weighted samples $\{X_{0:n_m}^k, w^k\}$ (particles), $k = 1, \dots, M$, that form an empirical estimate of the conditional pdf $p(x_{0:n_m} | y_{1:m})$, the goal is to update each particle to time n_{m+1} , by generating a sample $X_{n_m+1:n_{m+1}}$ using an importance function p_0 , and putting

$$\{X_{0:n_{m+1}}^k, \hat{w}^k\} = \{(X_{0:n_m}^k, X_{n_m+1:n_{m+1}}^k), \hat{w}^k\},$$

with updated weights

$$\hat{w}^k = w^k \frac{p(X_{n_m+1:n_{m+1}}^k | X_{n_m}^k) p(y_{m+1} | X_{n_{m+1}}^k)}{p_0(X_{n_m+1:n_{m+1}}^k)}. \quad (13)$$

The assimilation of data using the above sequential importance sampling approach is known as particle filtering (as opposed to the particle smoother, which does not operate sequentially).

For an efficient particle filter, we need to find an importance function p_0 that closely resembles the functions $p(X_{n_i+1:n_{i+1}}^k | X_{n_i}^k) p(y_{i+1} | X_{n_{i+1}}^k)$ for each

particle. We can achieve this by applying implicit sampling to each particle, and we will call this approach the implicit particle filter. Thus, we define M functions F^k by

$$F^k(x_{n_m+1:n_{m+1}}) = -\log(p(x_{n_m+1:n_{m+1}}|X_{n_m}^k)p(y_{m+1}|x_{n_m+1})) \quad (14)$$

For Gaussian observation noise, $V_k \sim \mathcal{N}(0, Q)$, these functions F^k become

$$\begin{aligned} F^k(x_{n_m+1:n_{m+1}}) &= \frac{1}{2}(x_{n_m+1} - R_{n_m}(X_{n_m}^k))^T \Sigma_{n_m}^{-1}(x_{n_m+1} - R_{n_m}(X_{n_m}^k)) \\ &\quad + \frac{1}{2} \sum_{j=n_m+1}^{n_{m+1}-1} (x_{j+1} - R_j(x_j))^T \Sigma_j^{-1}(x_{j+1} - R_j(x_j)) \\ &\quad + \frac{1}{2}(y_{m+1} - h(x_{n_{m+1}}))^T Q^{-1}(y_{m+1} - h(x_{n_{m+1}})) + C \end{aligned} \quad (15)$$

where C is a constant whose value is irrelevant. We find the minima ϕ_k of each of these F_k 's using standard techniques, such as Newton's method, quasi Newton methods or gradient descent (see e.g. [18, 40, 42]). We then pick a Gaussian reference variable $\xi \sim \mathcal{N}(0, I)$ and obtain M samples, $X_{n_m+1:n_{m+1}}^k$, by solving the M equations

$$F^k(X_{n_m+1:n_{m+1}}^k) - \phi^k = \frac{1}{2} \xi^T \xi, \quad (16)$$

with a suitable mapping ψ (see [7, 8, 39]). The update equation for the weights can be obtained by combining (7) with (13):

$$\hat{w}^k = w^k e^{-\phi^k} J(X_{n_m+1:n_{m+1}}^k) \quad (17)$$

where J is the Jacobian of ψ . We append the M samples $X_{n_m+1:n_{m+1}}^k$ to the M particles we already had, and replace their weight with the updated weight from (17). We thus obtain M updated particles that approximate the conditional pdf $p(x_{0:n_{m+1}}|y_{1:m+1})$ at time n_{m+1} . As a concrete state estimate, we can use, for example, the weighted sample average to approximate the conditional mean. Before the next observation is assimilated, the weights can be eliminated by resampling, using one of the algorithms in e.g. [12, 32, 37, 44].

Note that the term $\exp(-\phi^k)$ in (17) induces additional variability into the weights when compared to the implicit particle smoother in section 3.2, where the variability of the weights is due to only the Jacobian. The additional factor appears here because we apply implicit sampling to M different functions F^k which arise because of the sequential problem formulation (for

the implicit particle smoother, we applied implicit sampling to one function F). The functions F^k however differ only in the position of each particle, $X_{n_m}^k$, at time n_m (see equations (14) and (15)). For that reason, the minima ϕ^k of these functions should not vary too much from particle to particle, so that the variance induced by this term can be expected to be small. The numerical experiments in section 5, as well as those in [7, 39] confirm this statement, however a rigorous analysis of the variance of the weights of the implicit particle filter has not been reported.

3.4 The implicit particle smoother for perfect models

If model errors are small compared to observation errors, one can put

$$G_j(x_j) = 0,$$

in (8), so that the state trajectory, $x_{1:n_m}$, is a deterministic function of the initial condition x_0 . This assumption is often called the perfect model assumption and our goal is to find an initial state that is compatible with the available data y_k , $k = 1, \dots, m$.

The implicit particle smoother in section 3.2 can be easily adapted to this situation by applying implicit sampling to the conditional pdf $p(x_0|y_{1:m})$. Using Bayes' theorem, the fact that the observations y_k are independent of each other, and that $x_{1:n_m}$ is a deterministic function of x_0 , we can rewrite this conditional pdf as

$$p(x_0|y_{1:m}) \propto p(x_0) \prod_{j=1}^m p(y_j|x_{n_j}),$$

where the factors $p(y_j|x_{n_j})$ are specified by the observation equation (9). The pdf $p(x_0)$ is called the prior density and is often chosen to be Gaussian. However, the conditional pdf is generally far from being Gaussian, because h is generally nonlinear and the x_{n_j} 's are nonlinear functions of x_0 (see (8)).

For implicit sampling of $p(x_0|y_{1:m})$, we define

$$F(x_0) = -\log(p(x_0|y_{1:m})),$$

which for a Gaussian observation noise, $V_k \sim \mathcal{N}(0, Q)$, becomes

$$F(x_0) = -\log(p(x_0)) + \sum_{j=1}^m (h(x_{n_j}) - y_j)^T Q^{-1} (h(x_{n_j}) - y_j) + C, \quad (18)$$

where the value of the constant C is irrelevant. With this F , we can find M samples from $p(x_0|y_{1:n_m})$ by first minimizing F and then solving (12) repeatedly for M realizations of ξ . We can solve this scalar equation efficiently using e.g. random maps as in [39], or one of the methods in [7]. What is important to realize here is that sampling is fast, once the minimum of F has been found.

Finally, we want to point out that the above implicit smoothing algorithm above can be modified to assimilate data sequentially, i.e. assimilate $k < m$ observations at a time. We can assimilate the first k observations, $y_{1:k}$, by implicitly sampling $p(x_0|y_{1:k})$ and use the results to construct an empirical approximation of a “prior” density for x_{n_k} . With that prior, we repeat the same steps to assimilate the next set of observations $y_{k+1:2k}$ by implicitly sampling $p(x_{n_k}|y_{k+1:2k})$ etc. until all available observations are assimilated. Note that the method naturally keeps track of the uncertainty, whereas 4D-Var codes often use ad-hoc approximations to update the covariance matrices [28]. A sequential approach for data assimilation for perfect models is important in many applications with very large data sets, e.g. in numerical weather prediction or geomagnetics [19], however the details, as well as numerical tests for sequential implicit sampling for this problem are deferred a future paper.

4 Connection with variational data assimilation

Variational data assimilation methods find the most likely state trajectory, given the available observations, i.e. the mode of the conditional pdf $p(x_{0:n_m}|y_{1:m})$. We now make the connection between variational methods and the implicit particle filter and smoother, and show how existing codes for variational data assimilation can be used for efficient implementation of these implicit particle methods. We distinguish between weak and strong constraint variational methods.

4.1 Connection with strong constraint 4D-Var

Strong constraint 4D-Var, see e.g. [9–11, 41, 46, 47], finds the mode of the conditional pdf $p(x_0|y_{1:n_m})$, where x_0 is the unknown initial condition of the discrete model (8), by minimization of a suitable cost function. Specifically, if the pdf $p(x_0)$, which is often called the prior density, is Gaussian and if the observation noise is also Gaussian, the strong constraint 4D-Var cost

function is

$$\mathcal{J}_s(x_0) = (x_0 - x_b)^T B^{-1}(x_0 - x_b) + \sum_{j=1}^m (h(x_{n_j}) - y_j)^T Q^{-1}(h(x_{n_j}) - y_j), \quad (19)$$

where $x_b \in R^d$, called the background state, is the mean of $p(x_0)$ and $B \in R^{d \times d}$ is the covariance matrix of the background state.

If the observation operator h is linear, the gradient of the cost function \mathcal{J}_s can be found using the adjoint method (see e.g. [47]). With this gradient, we can minimize \mathcal{J}_s efficiently using e.g. gradient descent or quasi Newton methods. In the general case (h not linear), one can linearize h along a state trajectory and use this linearization along with the adjoint method to compute an approximate gradient of \mathcal{J}_s . The conditions under which a numerical minimization with an approximate gradient converges to the minimum of the cost function \mathcal{J}_s are not well understood. However the method seems to work in many applications. In fact, the use of the adjoint method makes the minimization of \mathcal{J}_s very efficient and, as a result, strong constraint 4D-Var a powerful method for nonlinear data assimilation.

The strong constraint 4D-Var cost function \mathcal{J}_s in (19) is identical to F in (18) (up to irrelevant constants), provided we use the same, and not necessarily Gaussian, prior pdf p_0 . Turning a strong constraint 4D-Var code into an implicit particle smoother (see section 3.4) thus amounts to adding a sampling and weighting step, which in turn amounts to solving the scalar equation (12), or more generally (4). Efficient methods for executing the sampling and weighting can be found in [7, 39], so that the additional computational cost of implicit particle smoothing is small. The payoff is that the implicit particle smoother approximates the conditional mean and, thus, minimizes the mean square error, whereas 4D-Var computes the conditional mode. The conditional mean is, under wide conditions, a better state estimate than the conditional mode, particularly if the conditional pdf has significant skew. Moreover, the implicit particle smoother naturally produces a quantification of the uncertainty (because it generates an empirical estimate of the conditional pdf), whereas 4D-Var codes provide at best approximate error estimates [41].

When the data are sparse in space or time, the conditional pdf often has more than one mode so that the cost function \mathcal{J}_s has multiple minima. Strong constraint 4D-Var will find one of these minima and return it as the state estimate. Important information from the other modes is lost. The implicit particle smoother on the other hand can perform well in multimodal situations (see sections 2 and 5) and, in theory, represents all modes of the

conditional pdf by its samples. In practice, there is no guarantee that the implicit particle smoother can sample all modes in all cases (because the numerical minimization may miss local minima), however the representation of a multimodal conditional pdf by the implicit particle smoother through at least some of its modes is superior to the results of a 4D-Var code, that represents the conditional pdf by only one of its modes.

4.2 Connection with weak constraint 4D-Var

Weak constraint 4D-Var (see e.g. [1, 27, 31]) relaxes the perfect model assumption made in strong constraint 4D-Var. There are several ways of doing so [48], however we only consider here the “full” weak 4D-Var problem, i.e. we choose the model state $x_{0:n_m}$ as the control vector. The weak constraint 4D-Var method then computes the most likely state trajectory given the available data $y_{1:m}$, i.e. the mode of the conditional pdf $p(x_{0:n_m} | y_{1:m})$.

The conditional mode is found by minimizing the weak constraint cost function

$$\mathcal{J}_w(x_{0:n_m}) = -2 \log p(x_{0:n_m} | y_{1:m}).$$

Specifically, for a Gaussian prior density $p(x_0) \sim \mathcal{N}(x_b, B)$, the weak constraint 4D-Var cost function is

$$\begin{aligned} \mathcal{J}_w(x_{0:n_m}) &= (x_0 - x_b)^T B^{-1} (x_0 - x_b) \\ &+ \sum_{j=0}^{n_m-1} (x_{j+1} - R_j(x_j))^T \Sigma_j^{-1} (x_{j+1} - R_j(x_j)) \\ &+ \sum_{j=1}^m (y_{jk} - h(x_{n_j}))^T Q^{-1} (y_j - h(x_{n_j})). \end{aligned} \quad (20)$$

The adjoint method is not directly applicable to finding the gradient of \mathcal{J}_w , but related approximate methods can be devised to streamline and accelerate the minimization, see e.g. [27, 51]. Although it is not yet well understood under what conditions a minimization with such approximations converges to a minimum, weak constraint 4D-Var is widely used and is known to produce good state estimates, in particular in numerical weather prediction.

Note that the cost function \mathcal{J}_w in (20) equals F in (11), the function that is minimized by the implicit particle smoother of section 3.2 (up to irrelevant constants). We can thus use a weak 4D-Var code for the implementation of an implicit particle smoother to minimize this F . Once the minimum

is found, we can obtain M samples from the conditional pdf by solving (12) repeatedly and at a low cost. The additional cost of implementing the implicit particle smoother versus a weak constraint 4D-Var method is thus not large. The implicit particle smoother however has the advantage that it can return the conditional mean as a state estimate, which, under wide conditions, is a better state estimate than the conditional mode (the result of a weak 4D-Var calculation). Moreover, the state estimate of the implicit particle smoother is equipped with a quantitative measure of its uncertainty.

Recall that the implicit particle filter of section 3.3 is an efficient *sequential* sampling method for the conditional pdf. The implicit particle filter requires at each assimilation and for each particle, the minimization of the function F^k in (14). These F^k 's are parameterized by the previous position of each particle and by the current observation. Moreover, for each particle, F^k is nearly identical to the cost function \mathcal{J}_w of weak constraint 4D-Var in (20). The differences are in the treatment of the background state. It is unnecessary to include the background state in the functions F^k because the implicit particle filter samples the prior directly, and without making a Gaussian assumption. Since the implicit particle filter is a sequential method, we set it up in section 3.3 to assimilate one observation at a time, so that the arguments of F^k are $x_{n_m+1:n_m+1}$. We can thus obtain the F^k 's from the weak constraint cost function \mathcal{J}_s in (20) by removing the background state, turning the *variables* x_0 into *parameters* $X_{n_m}^k$ (the position of the k th particle at time n_m), and running the variational assimilation over one observation only. The particle-by-particle minimizations of F^k for the implicit particle filter can thus be carried out by existing weak constraint 4D-Var codes with only minor modifications. Once the minimum of each F^k is found, the sampling can be carried out efficiently using the methods in [7, 39], so that this extension comes at an acceptable cost. Moreover, the minimization for each particle can be done in parallel so that the runtime of a parallelized implementation of the implicit particle filter is comparable to serial 4D-Var codes.

The main benefits for the implicit particle filter are (*i*) the implicit particle filter tracks the time evolution of the conditional pdf and, thus, can compute the conditional mean, which minimizes the mean square error; (*ii*) the filter naturally produces a quantitative representation of the uncertainty (because it tracks the conditional pdf); and (*iii*) the implicit particle filter handles new observations (in time) naturally, because it is set up as a sequential method. The last point is particularly important when the data sets are large.

We argued in the previous section that the improvement of strong con-

straint 4D-Var by the implicit particle smoother is particularly pronounced if the conditional pdf has more than one mode. The arguments presented towards the end of section 4.1 also hold for the weak constraint problem and we expect the implicit particle filter and smoother to perform better than weak constraint 4D-Var in such cases.

5 Application to the Lorenz attractor

To illustrate the ideas of the previous sections, we follow [6,15,36] and apply the implicit particle filter of section 3.3 and the implicit particle smoother of section 3.4 to the Lorenz attractor [35]. We distinguish between the strong and weak constraint problem.

5.1 The strong constraint problem

The Lorenz attractor is governed the set of ordinary differential equations (ODE)

$$\frac{dx^1}{dt} = \sigma(x^2 - x^1), \quad \frac{dx^2}{dt} = x^1(\rho - x^3) - x^2, \quad \frac{dx^3}{dt} = x^1x^2 - \beta x^3, \quad (21)$$

where $\rho = 28$, $\sigma = 10$, $\beta = 8/3$ [35]. We discretize these equations using a fourth-order Runge-Kutta scheme with constant time step $\delta = 0.01$. For our purposes, and because we will consider only relatively short integration times, this approximation is sufficiently accurate.

We observe the variables x^1 and x^3 , corrupted by Gaussian noise with mean zero and covariance matrix $Q = 2I_2$ (I_m is the $m \times m$ identity matrix), every $r = 20$ model steps, i.e. every 0.2 dimensionless time units. The observation equation (9) thus becomes

$$y_k = (x^1(t_{n_k}), x^3(t_{n_k}))^T + V_k,$$

with $V_k \sim \mathcal{N}(0, Q)$. Our goal is to update the prior knowledge about the initial state x_0 , which we assume to be Gaussian, so that $p_0 \sim \mathcal{N}(x_b, B)$ with $x_b = (4.3735, 6.9590, 15.4321)^T$ and $B = 0.5I_3$, based upon 4 observations y_1, \dots, y_4 . We try to achieve this goal by using the implicit particle smoother of section 3.4.

Recall that the implicit particle smoother essentially consists of three steps: *(i)* minimize the function F in (18); *(ii)* obtain samples from the underlying conditional pdf by solving the algebraic equation (12); and *(iii)* weighing the samples using (7). As pointed out in section 4.1, the first step can be carried out using adjoint codes and that is what we did for this example.

5.1.1 Variational implementation of the implicit particle smoother

We constructed the linear tangent adjoint of the continuous time ODE's in (21) and discretized the adjoint equations using a fourth order Runge-Kutta scheme with time step $\delta = 0.01$. We use these adjoint equations to compute the gradient of the function F , which in turn is used in a BFGS method (see e.g. [18, 40]) for the minimization of F . To initialize this BFGS method, we ran a few steps of a BFGS method on the “maximum likelihood” problem (i.e. we neglect the background term in F), in which we could also use the adjoint equations for the gradient computations. The result of the BFGS iteration on the maximum-likelihood problem was used to initialize the BFGS method for the minimization of F . We found that this approach is quicker than using the BFGS method on F , initialized with the background state x_b , because, for our choice of parameters, F seems to have a rather flat region around the background state which is not the minimum. The minimization converged quickly and, on average, took only about 1 second (using non-optimized Matlab code). We observed occasionally that the minimization was trapped in very flat regions, in which case we re-started the whole process, using a sample from the prior density p_0 to initialize the minimization.

5.1.2 Implementation of the map ψ

Upon minimization of F with the adjoint method, we solve (4) to obtain samples from the conditional pdf. To solve this underdetermined equation, we need to define a reference variable ξ , as well as a map from ξ to X . Here, we follow [7] and choose a Gaussian reference variable $\xi \sim \mathcal{N}(0, I_3)$ and consider a map ψ that makes use of the quadratic expansion of F ,

$$F_0(x) = \phi + \frac{1}{2}(x - \mu)^T H(x - \mu),$$

where $\mu = \operatorname{argmin} F$ is the minimizer of F and H is an approximation of its Hessian, evaluated at the minimizer, which is available from the variational minimization using BFGS. To obtain a sample, we then solve the quadratic equation

$$F_0(x) - \phi = \frac{1}{2}\xi^T \xi, \tag{22}$$

instead of (12). This can be done efficiently using the Cholesky factor L of H :

$$X = \mu + L^{-T}\xi. \tag{23}$$

The Jacobian of this map is easily calculated to be the determinant of L (the product of its diagonal entries) and is constant among the particles. We account for the error we made by solving (22) rather than (12) by reweighing the samples such that

$$w^k \propto e^{-(F(X)-F_0(X))},$$

where the terms involving the ϕ 's and the Jacobian (see (7)) need not show up because they are constant for all particles, and thus drop out once the weights are scaled so that their sum equals one. This map is very efficient for this problem, because L is a easy to compute (and can be computed offline). In particular, the evaluation of (23) takes about 0.6% of the time it takes to carry out the variational minimization so that the cost of sampling is small compared to the cost of minimizing F . In this example, turning the 4D-Var code of the previous subsection into implicit particle smoothing code comes at an acceptable additional cost.

5.1.3 Numerical results

Figure 1 illustrates the data assimilation with the implicit particle smoother. On the left (time $t \leq 0.8$), we show the true state trajectory (teal), which was obtained by integrating the equations (21) starting from an initial condition which we got by sampling the prior pdf p_0 . We also show the data (red dots) with error bars that represent two standard deviations ($2\sqrt{2}$ in our case) and the mean (red dot at time 0) of the prior pdf with the same error bars. The blue lines show 30 samples from the prior pdf and the purple lines are 25 samples we obtained using the implicit particle smoother. The sample mean (obtained by using 100 particles) is not shown, but it is very close to the true state trajectory. We can observe in this figure that the implicit particle smoother generates samples within the high probability region, because all samples are compatible with the data (they are within 2 standard deviation of the data).

We can use the implicit particle smoother to make and assess a forecast (for time $t \geq 0.8$) as follows. We can approximate the pdf of the state at time 0.8 by a Gaussian whose mean and covariance matrix can be computed from the weighted samples. We can then integrate samples, say 50, from this Gaussian. The result is shown as purple lines on the right of figure 1, and we observe that the true state (teal) is well within the cloud of samples. We can also observe that the uncertainty grows dramatically for times larger than 1.4, i.e. a forecast should not be expected to be very accurate.

We further assessed the accuracy and reliability of the implicit particle smoother by running 100 twin experiments. A twin experiment amounts

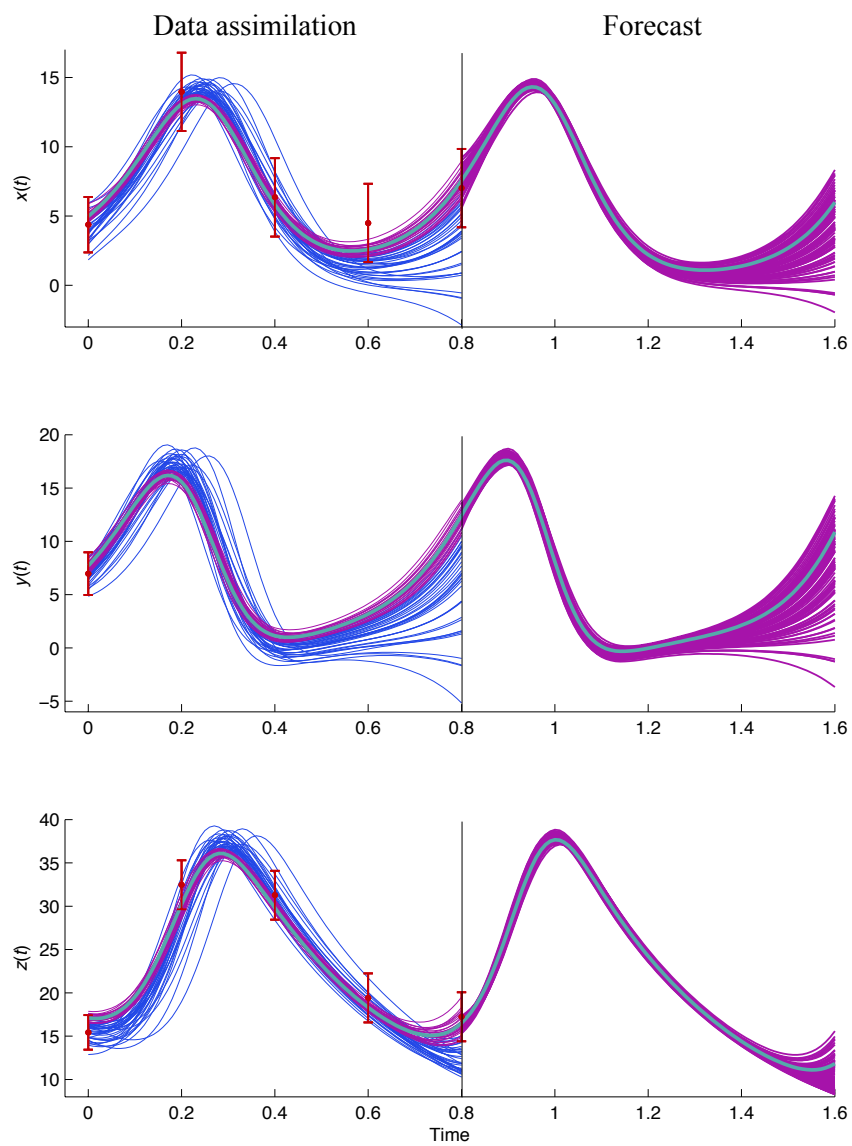


Figure 1: Illustration of data assimilation and forecasting using the implicit particle smoother. On the left (Time ≤ 0.8): 30 samples from the prior pdf (blue lines); the data and error bars (red); 25 samples obtained by the implicit particle smoother (purple); and the true state trajectory (teal). On the right (Time > 0.8): 50 samples of a Gaussian approximation of the pdf of the state at time 0.8 obtained by the implicit particle smoother (purple); and the true state trajectory (teal).

to generating a “true” initial condition by sampling the prior pdf p_0 , integrating this initial condition forward in time and collecting observations by perturbing the true state trajectory with appropriate noise. The data are passed to the implicit particle smoother, which then produces an approximation to the conditional mean, which in turn is the minimum mean square error estimate of the initial condition. We then compute the Euclidean norm of the difference between the true initial condition and its approximation by the implicit particle smoother. The mean and standard deviation of this error norm, scaled by the mean of the norm of the true initial conditions, indicate the errors one should expect in each run.

We compare the implicit particle smoother to the variational data assimilation scheme (4D-Var) which is implemented as part of the implicit particle smoother. In order to check that our implementation of the implicit particle smoother is free of errors, we compare its errors to those obtained with a Bayesian bootstrap method [12]. The Bayesian bootstrap method is an importance sampling method that uses the prior pdf p_0 as the importance function, i.e. we obtain samples from the prior pdf and then assign a weight based on the observations to each sample. The conditional mean can be approximated by the weighted sample mean and, for a large number of particles, this method converges to the true conditional mean. We observed that this method has converged for 1000 particles. With this number of particles, the Bayesian bootstrap method is about 4 times slower than our variational implementation of the implicit particle smoother. The results of 100 twin experiments are shown in table 1

4D-Var	Implicit particle smoother (100 particles)	Bayesian bootstrap (1000 particles)
0.060 / 0.025	0.043 / 0.018	0.042 / 0.017

Table 1: Errors (mean / standard deviation) in the reconstruction of the true initial condition for three data assimilation techniques.

We observe that all three data assimilation methods do their job, since all three yield a small error and a small error variance. The Bayesian bootstrap method and the implicit particle smoother converge to the same errors (both approximate the conditional mean), however, it is clear that the implicit particle smoother is more efficient since it is 4 times faster than the Bayesian bootstrap method. The implicit particle smoother improved the estimate of the variational method through sampling, i.e. by computing the

conditional mean instead of the conditional mode, at an acceptable additional computational cost. Moreover, the implicit particle smoother delivers a quantitative measure of the uncertainty of its state estimate, which can be used to propagate the uncertainty forward in time and to assess the uncertainty of forecasts (see figure 1).

We conclude that the implicit particle smoother is efficient and reliable in its variational implementation and that the additional computational cost, when compared to a variational method, is acceptable in view of its clear advantages (smaller errors and state estimates that are equipped with measures of their uncertainty).

5.2 The weak constraint problem

We now consider a weak constraint problem and use a stochastic version of the Lorenz attractor

$$\begin{aligned}\frac{dx^1}{dt} &= \sigma(x^2 - x^1) + g dW^1, \\ \frac{dx^2}{dt} &= x_1(\rho - x^3) - x^2 + g dW^2 \\ \frac{dx^3}{dt} &= x_1 x^2 - \beta x^3 + g dW^3,\end{aligned}$$

where W^1, W^2 and W^3 are independent Brownian motions and where σ, ρ and β are as in section 5.1 and $g = 1/\sqrt{2}$. We discretize these stochastic differential equations (SDE) using the Euler-Maruyama scheme with constant time step $\delta = 10^{-3}$, so that the discretization is a good approximation to the continuous equations [30]. With this choice the function $R_j(x_j)$ for the discrete recurrence (8) becomes

$$R(x_j) = x_j + f(x_j)\delta,$$

where $x_j = (x_j^1, x_j^2, x_j^3)^T$ and

$$f(x_j) = (\sigma(x_j^2 - x_j^1), x_j^1(\rho - x_j^3) - x_j^2, x_j^1 x_j^2 - \beta x_j^3)^T,$$

and $Z_k \sim \mathcal{N}(0, \delta/2)$.

The observations are all three state variables, collected at times $t_{n_k} = k \cdot r \cdot \delta$, perturbed by Gaussian noise with mean zero and covariance matrix $Q = 2I_3$. The data assimilation problem is particularly hard when the time between observations is greater than the characteristic time scale at

which transitions are made between the two attractors, which, for our choice of parameters is about $T = 1/2$ [36]. We consider two cases: (a) $r = 400$, i.e. the gap between observations is 0.4 dimensionless time units and smaller than the characteristic time scale 0.5; and (b) $r = 800$, i.e. the gap between observations is 0.4 dimensionless time units and larger than the characteristic time scale. In both cases we assimilate the data sequentially using the implicit particle filter of section 3.3.

5.2.1 Variational implementation of the implicit particle filter

The main computational challenge of the implicit particle filter is to find the minima of the F^k 's in (14). We explained in section (4) that these F^k 's are related to the weak constraint 4D-Var cost function and that 4D-Var codes can be used to carry out the required minimizations. The various weak 4D-Var codes differ mainly in the extent to which approximate techniques, such as linearizations or Gaussian assumptions, are used. We decided not to favor any particular approximate version of weak constraint 4D-Var and, for that reason, computed the first and second derivatives of F^k analytically and used a trust-region method for the minimizations (see e.g. [42]). This corresponds to an “ideal” implementation of weak constraint 4D-Var, for which the control variable is the full state trajectory [48].

The trust-region approach requires a Cholesky decomposition of the Hessian of F^k at each iteration of the minimization algorithm. Since this Hessian is banded (with band width 6), the cost of one iteration is $O(3m)$, where m is the number of model steps between observations. The number of model steps between observations increases quickly as the (non-dimensional) time between observations increases, because we chose a small time step δ to ensure accuracy of the discretization of the SDE's. Because the cost of each iteration is relatively large for large gaps between observations, it is worthwhile to invest into generating “good seeds” to initialize the trust-region iteration, so that it converges quickly.

We generated a seed as follows: for each time window between observations, we first obtain $\bar{x}_m = x_{n_m+1:n_m+1}$ by integrating the stochastic differential equation. We then calculate the “residual vector” $r = x_{n_m+1} - y_{m+1}$ and perturb the model path using $\bar{x}_j^r = \bar{x}_j - r(j/r)$ for each $j = 0, 1, 2, \dots, r$. This procedure rotates the model path \bar{x}_j towards the observation.

We refine this seed with a multi-grid technique, which is conceptually similar to the multi-grid finite difference method [17] and multi-grid Monte Carlo [21] (see also [4]). The idea is to first perform a cheap minimization on a coarse grid, i.e. with a larger time step, and then use the result of this

minimization, interpolated onto the fine grid, as the seed for the minimization on the finer grid. The reason why we can use this multi-grid approach here is that the conditional pdf depends on the model (it is proportional to the product of the pdf for the model and the pdf for the observations), which in turn represents an approximation to an SDE. The conditional pdf we obtain with a model and time step say $\hat{\delta} < \delta$ should thus be somewhat similar to the conditional pdf we obtain with a time step $\delta < \hat{\delta}$. Since F^k is minus the logarithm of the conditional density, we expect that the minimizer of F^k with a model with time step $\hat{\delta}$ is similar to the minimizer of an F^k with a model and time step $\delta < \hat{\delta}$.

In addition to speeding up the minimization, the multi-grid approach proved effective to identify local minima of F^k . We observed in our experiments that the global minimum of F^k was rarely larger than 10, independent of the time step or even the gap between observation times. Local minima were observed to be as large as 200. This observation can be used to identify local minima of F^k : the result of coarse grid minimization is rejected if the minimum is above a threshold ϕ_c , and we restart the minimization with a new (unrefined) seed \bar{x}_m .

To test our minimization algorithm (the weak 4D-Var code), we compare its output to the output of a trust-region method that uses “the truth” as its seed, i.e. we generate a reference state trajectory by integrating the SDE’s, collect observations from this state trajectory and run our 4D-Var code as well as a trust-region method that is initialized with the true state trajectory. This should give us an idea of how accurate our 4D-Var code is, because the true state trajectory typically lies only a few Newton steps away from a relevant mode of the conditional pdf. We find that our multi-grid scheme finds the same minimum as seeding the minimization with the truth 100% of the time for gaps between observations that are less than 1.5 dimensionless time units (1500 model steps).

5.2.2 Implementation of the map ψ

Upon minimization of the F^k ’s, we solve (16) for each particle to obtain samples from the conditional pdf. To solve this underdetermined equation, we use the same approach as in section 5.1, i.e. we replace F^k in (16) by its quadratic approximation and solve a quadratic equation. This approach is very efficient for this problem, because we can solve the quadratic equation using the Cholesky factor, L , of the Hessian of F^k , which is available from the trust-region minimization (the variational part of the implicit particle filter). The Jacobian of this map is easily calculated to be the determinant

of L (the product of its diagonal entries). Generating a sample using this map takes about 1/10000 of the time it takes to carry out the minimization. The cost of sampling is thus small compared to the cost of minimizing F^k , i.e. turning a weak 4D-Var code into implicit sampling code comes at a low additional cost. Again, we account for the error we make by replacing F^k by its quadratic approximation through the weights, which become

$$\hat{w}^k = w^k e^{-\phi^k} e^{-\left(F(X_{n_m:n_{m+1}}^k) - F_0(X_{n_m:n_{m+1}}^k)\right)} \det L^{-1}.$$

Note that the factors with ϕ^k and the Jacobian of the map ($\det L^{-1}$) must appear in the weights because the functions F^k are different for each particle and, thus, can have different minima and different Hessians.

5.2.3 Monte Carlo variance reduction

We can improve the performance of the implicit particle filter by using standard Monte Carlo variance reduction techniques such as prior boosting, rejection control or partial rejection control [22, 33, 34]. These methods rely on generating an expanded ensemble of particles from which only a subset will be promoted to the next assimilation window. It is important to realize that the expanded ensemble of particles does not require additional minimizations, because the new “intermediate” particles share their F^k ’s with their “parent” particles (for which the minimization has already been carried out).

In particular, we can generate $m > 1$ “intermediate” particles for each of the M particles by using (23) repeatedly. We thus obtain mM samples of the conditional pdf, essentially at the cost of M samples (since using (23) is cheap compared to the minimization of F^k). This “prior boosting” technique proved effective at increasing sample diversity in our numerical experiments.

5.2.4 Numerical results

We test the efficiency and accuracy of the implicit particle filter by running twin experiments, as we did in section 5.1. Each twin experiment amounts to generating a reference solution up to time 4, also called “the truth,” using the Euler-Maruyama discretization of the stochastic Lorenz attractor, and collecting observations at times $t_{n_k} = k \cdot r \cdot \delta$. We consider two cases: (a) data is collected every $r = 400$ model steps (the gap between observations is smaller than the characteristic time scale of the Lorenz attractor); and (b) data is collected every $r = 800$ model steps (the gap between observations is larger than the characteristic time scale of the Lorenz attractor).

In each case, the data are passed to three data assimilation algorithms: (i) the implicit particle filter in its sequential form (see section 3.3); (ii) the Bayesian bootstrap filter with resampling (also sometimes known as the standard SIR filter), which uses the pdf $p(x_{n_m+1:n_{m+1}}|x_{n_m})$ as its importance function [12, 22]; and (iii) an implementation of weak constraint 4D-Var, which uses the same (nonlinear) multi-grid trust-region method as the implicit particle filter to carry out the minimizations. The weak 4D-Var code also assimilates the observations sequentially. The output of the two filters is an approximation of the conditional mean, and the output of the weak constraint 4D-Var code is an approximation of the conditional mode.

In figure 2 we plot the results of one twin experiment, where we assimilate sequentially 5 observations, with $r = 800$ model steps (0.8 dimensionless time units) between observations (case (b)). We observe that, with 20 particles, the SIR filter loses track of the true state trajectory after a relatively short time. The reason is that none of the samples is sufficiently close to the observations, i.e. we observe the typical effect of sample impoverishment. The weak constraint 4D-Var code can not follow the true state trajectory, because, starting at time 2.4, it is trapped in a local minimum. The implicit particle filter with 20 particles, each boosted with 50 intermediate particles (see section 5.2.3) can follow the true state trajectory at all times. The reason why the implicit particle filter is not “stuck” in a local minimum (as is 4D-Var) is that it is able to track the various modes of the conditional pdf, since the minimization is performed particle by particle. In this example, about 10 particles appear sufficient to track all relevant modes.

We perform 100 such twin experiments, because a single twin experiment is not very informative (it is a random event). For each one we compute the errors $e = x_{0:n}^F - x_{0:n}$, where $x_{0:N}$ is the true state trajectory and $x_{0:N}^F$ is the output of the data assimilation (implicit particle filter, SIR filter, or 4D-Var). The mean and standard deviation of the Euclidean norm of these errors indicates the errors one can expect for each method and in each run. The results are shown in table 2, where we scaled the errors and their standard deviations by the mean of the Euclidean norm of the true state trajectory.

We observe from table 2, that the implicit particle filter as well as the standard SIR filters can provide accurate approximations of the true state in both cases (since all errors are relatively small), provided that the number of particles is large enough. What is important to realize here is that the implicit particle filter can achieve a similar accuracy, but with a significantly lower number of particles than the standard SIR filter. The weak 4D-Var method can not achieve the accuracy of the particle filters, especially if the

Case (a): $r = 400$ model steps between observations			
Number of particles	4D-Var	IPF	SIR
-	0.086 / 0.063	-	-
10	-	0.042 / 0.012	0.15 / 0.16
20	-	0.040 / 0.013	0.092 / 0.10
100	-	-	0.048 / 0.050
1000	-	-	0.038 / 0.013
5000	-	-	0.037 / 0.037

Case (b): $r = 800$ model steps between observations			
Number of particles	4D-Var	IPF	SIR
-	0.13 / 0.15	-	-
10	-	0.074 / 0.070	0.18 / 0.17
20	-	0.074 / 0.080	0.14 / 0.15
100	-	-	0.077 / 0.082
1000	-	-	0.065 / 0.056
5000	-	-	0.064 / 0.064

Table 2: Errors (mean / standard deviation) of three data assimilation techniques. 4D-Var: an ideal implementation of weak constraint 4D-Var. IPF: the implicit particle filter (each particle has 50 intermediate particles). SIR: the Bayesian bootstrap filter.

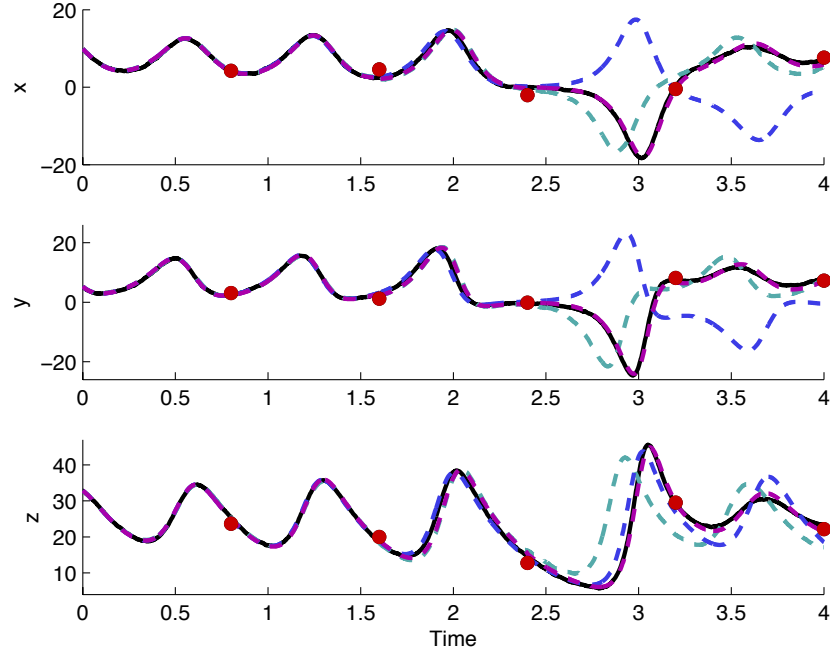


Figure 2: Reconstructions of a reference path (solid-black) from a set of 5 observations (red dots) by three data assimilation methods. Dashed-teal: reconstruction by the standard SIR filter with 20 particles. Dashed-blue: reconstruction by weak constraint 4D-Var. Dashed-purple: reconstruction by the implicit particle filter with 10 particles, each with 50 intermediate particles.

gap between observations is larger (case *(b)*). The reason is that the method is trapped in local minima, i.e. 4D-Var is unable to track more than one mode. The implicit particle filter on the other hand is able to track all relevant modes (at least in this example), due to the particle-by-particle minimization. The cost is that the implicit particle filter performs a 4D-Var calculation for each particle, i.e. the computational cost is roughly the cost of the 4D-Var method times the number of particles, because sampling is cheap compared to the minimization. The pay-off of being able to track all relevant modes however outweighs this cost in this example. Moreover, the implicit particle filter is very easy to parallelize (the particles only communicate in

the resampling step), and a parallel implementation of the implicit particle filter is comparable in computation time to a (serial) implementation of weak 4D-Var.

To further assess the quality of the implicit particle filter, we compute the normalized effective sample size

$$\frac{M_{\text{eff}}}{M} = \frac{(\sum_{k=1}^M w_k)^2}{M \sum_{k=1}^M w_k^2},$$

where M is the number of particles, for each twin experiment at the last data assimilation cycle. The normalized effective sample size indicates the percentage of particles that contribute meaningfully to the approximation of the conditional pdf [12] and we compare the normalized effective sample size of the implicit particle filter and the SIR filter. The results are shown in table 3.

Case (a): $r = 400$ model steps between observations		
Number of particles	IPF	SIR
10	95.0%	50.4 %
20	94.5 %	49.2 %
100	-	49.0 %
1000	-	48.5%
5000	-	48.6%

Case (b): $r = 800$ model steps between observations		
Number of particles	IPF	SIR
10	84.8%	37.9 %
20	84.1 %	34.7 %
100	-	32.6 %
1000	-	33.0%
5000	-	33.0%

Table 3: Normalized effective sample size of the implicit particle filter and the standard SIR filter

We observe that, with a relatively short time between observations (case (a)), about 50% of the particles of the standard SIR filter are contributing meaningfully to the ensemble averages. The situation is more dramatic for a larger gap between observations (case (b)), where we observe effective sample sizes of about 35%. The normalized effective sample size of the

implicit particle filter is about 95% for small gaps, and about 84% for larger gaps. In summary, we conclude that the implicit particle filter performs accurately and reliably on our test problems and yields accurate results (with uncertainty quantifications) at a reasonable computational cost.

6 Conclusions

The implicit particle filter was introduced in [7, 8, 39] as a sequential Monte Carlo method for data assimilation. In the present paper, we derived the implicit particle filter in a more general set up and presented extensions to implicit particle smoothing and to data assimilation for perfect models.

We explored the connection of these implicit particle methods with variational data assimilation and showed that existing variational codes can be used for efficient implementation of implicit particle methods. In particular, we showed that variational codes can carry out the minimizations required by implicit particle methods. Turning a variational code into an implicit particle method then amounts to solving an underdetermined scalar equation; methods to solve these equations efficiently can be found in our earlier work (e.g. in [7, 39]). The additional cost of implicit particle methods is thus small, and the payoff is that one can obtain the minimum mean square error estimate of the state along with a quantitative measure of its uncertainty, whereas variational codes produce biased state estimates with at best approximate error quantifications.

We have demonstrated the applicability and efficiency of the implicit particle methods by applying them to the Lorenz attractor. We considered the strong constraint data assimilation problem (estimation of initial conditions for a perfect model) as well as the weak constraint problem (estimation of the state trajectory of an uncertain model) and, in both cases discussed the details of the variational aspects of the filter. In the strong constraint problem, we found that the implicit particle filter can improve the variational estimate significantly by turning the conditional mode into the conditional mean (the minimum mean square error estimator). Moreover, the implicit particle smoother produced quantitative measures of the uncertainty which were useful in assessing the uncertainty in forecasts. In the weak constraint problem, we found that the implicit particle filter requires about 1% of the particles of a standard SIR filter, and that it performs better than weak constraint 4D-Var because it can track all relevant modes of the conditional pdf. In every case we considered, the cost of solving the implicit equations to generate samples was small compared to the cost of the minimizations,

i.e. to the cost the implicit particle filter shares with variational data assimilation.

Acknowledgements

We would like to thank our collaborators at Oregon State University, Professors Robert Miller and Yvette Spitz, and Dr. Brad Weir, for helpful discussion and comments. This work was supported in part by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and by the National Science Foundation under grants DMS-0705910 and OCE-0934298.

References

- [1] A. F. Bennet, L. M. Leslie, C. R. Hagelberg, and P. E. Powers. A cyclone prediction using a barotropic model initialized by a general inverse method. *Monthly Weather Review*, 121:1714–1728, 1993.
- [2] P. Bickel, T. Bengtsson, and J. Anderson. Sharp failure rates for the bootstrap particle filter in high dimensions. *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, 3:318–329, 2008.
- [3] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings -*, 146(1):2–7, feb 1999.
- [4] A. Chorin. Monte carlo without chains. *Communications in Applied Mathematics and Computational Science*, 3:77–93, 2008.
- [5] A. J. Chorin and O. H. Hald. *Stochastic Tools in Mathematics and Science*. Springer, first edition, 2006.
- [6] A. J. Chorin and P. Krause. Dimensional reduction for a Bayesian filter. *Proceedings of the National Academy of Sciences*, 101(42):15013–15017, 2004.
- [7] A. J. Chorin, M. Morzfeld, and X. Tu. Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Science*, 5(2):221–240, 2010.

- [8] A. J. Chorin and X. Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254, 2009.
- [9] P. Courtier. Dual formulation of four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 123:2449–2461, 1997.
- [10] P. Courtier, J.N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120:1367–1387, 1994.
- [11] F. X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110, 1986.
- [12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in practice*. Springer, 2001.
- [13] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.
- [14] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.
- [15] G. Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, 125:1342–1354, 1997.
- [16] G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2006.
- [17] R. P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Computational Mathematics and Mathematical Physics*, 1, 1961.
- [18] R. Fletcher. *Practical Methods of Optimization*. Wiley, second edition, 1987.
- [19] A. Fournier, G. Hulot, D. Jault, W. Kuang, A. Tangborn, N. Gillet, E. Canet, J. Aubert, and F. Lhuillier. An introduction to data assimilation and predictability in geomagnetism. *Space Science Reviews*, 155(1-4):247–291, 2010.

- [20] J. Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 24:1317 – 1399, 1989.
- [21] J. Goodman and A. D. Sokal. Multigrid monte carlo method. conceptual foundations. *Phys. Rev. D*, 40:2035–2071, Sep 1989.
- [22] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, apr 1993.
- [23] J. M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*, volume 1. Methuen young books, 1 edition, 1964.
- [24] A. M. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498 – 1504, 2008.
- [25] R. E. Kalman. A new approach to linear filtering and prediction theory. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–48, 1960.
- [26] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME–Journal of Basic Engineering*, 83(Series D):95–108, 1961.
- [27] E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003.
- [28] E. Kalnay, H. Li, T. Miyoshi, S. C. Yang, and J. Ballabrera-Poy. 4-d-var or ensemble kalman filter. *Tellus*, 59A:758–773, 2007.
- [29] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*, volume 1. John Wiley & Sons, 1 edition, 1986.
- [30] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 3 edition, 1999.
- [31] A. Kurapov, G. D. Egbert, J. S. Allen, and R. N. Miller. Representer-based variational data assimilation in a nonlinear model of nearshore circulation. *Journal of Geophysical Research*, 112:C11019, 2007.
- [32] J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):pp. 567–576, 1995.

- [33] J. S. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling with resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in practice*, chapter 11. Springer, 2001.
- [34] J. S. Liu, R. Chen, and W. H. Wong. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):pp. 1022–1031, 1998.
- [35] E. N. Lorenz. Deterministic Nonperiodic Flow. *Journal of Atmospheric Sciences*, 20:130–148, March 1963.
- [36] R. N. Miller, E. F. Carter, and S. T. Blue. Data assimilation into nonlinear stochastic models. *Tellus A*, 51(2):167–194, 1999.
- [37] P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling strategies for sequential monte carlo methods. *Bernoulli*, 18(1):252–278, 2012.
- [38] M. Morzfeld and A. J. Chorin. Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation. *Nonlinear Processes in Geophysics*, page submitted for publication, 2012.
- [39] M. Morzfeld, X. Tu, E. Atkins, and A. J. Chorin. A random map implementation of implicit filters. *Journal of Computational Physics*, 231(0):-, 2011.
- [40] J. Nocedal and S. T. Wright. *Numerical Optimization*. Springer, second edition, 2006.
- [41] F. Rabier and P. Courtier. Four-dimensional assimilation in the presence of baroclinic instability. *Quarterly Journal of the Royal Meteorological Society*, 118(506):649–672, 1992.
- [42] A. R.Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, first edition, 2000.
- [43] M. S.Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174 –188, feb 2002.
- [44] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):pp. 84–88, 1992.

- [45] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, Dec 2008.
- [46] O. Talagrand. Assimilation of observations, an introduction. *Journal of the Meteorological Society of Japan*, 75(1):191–209, 1997.
- [47] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.
- [48] Y. Tremolet. Accounting for an imperfect model in 4d-var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006.
- [49] P. J. van Leeuwen. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999, 2010.
- [50] P. J. van Leeuwen. Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers & Fluids*, 46(1):52 – 58, 2011.
- [51] D. Zupanski. A general weak constraint applicable to operational 4DVAR data assimilation systems. *Monthly Weather Review*, 125:2274–2292, 1997.