Implicit particle methods and their connection with variational data assimilation

Ethan Atkins

Department of Mathematics, University of California, Berkeley, CA, USA

Matthias Morzfeld *

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Alexandre J. Chorin

Department of Mathematics, University of California, Berkeley, CA, USA

Corresponding author address: Matthias Morzfeld, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720.

E-mail: mmo@math.lbl.gov

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 those one encounters in variational data assimilation, and we 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 additional 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 (Doucet et al. 2001; Kalnay 2003; Evensen 2006; Chorin and Hald 2006). 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 (Kalman 1960; Kalman and Bucy 1961). 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 filter (Evensen 2006, 1994), can perform poorly if these assumptions are violated (Miller et al. 1999).

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 (Talagrand and Courtier 1987; Dimet and Talagrand 1986; Tremolet 2006; Bennet et al. 1993). 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 (Doucet et al. 2001; Arulampalam et al. 2002; Gordon et al. 1993). 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, which leads to a representation of the conditional pdf by

a single and often uninformative particle (Bickel et al. 2008; Snyder et al. 2008). This effect is known as sample impoverishment, is often severe in nonlinear, large-dimensional models and, thus, 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 (Doucet et al. 2000; Johansen and Doucet 2008; van Leeuwen 2010, 2011; Carpenter et al. 1999). The problem of sample impoverishment was also considered in the context of other applications, and many promising importance sampling methods, which make use of an importance density that is informed by the data and, therefore, can delay sample impoverishment, have been invented (Cappé et al. 2008; Cornebise et al. 2008; Doucet et al. 2000; Pitt and Shephard 1999; Smídl and Hofman 2012)

The implicit particle filter (Chorin et al. 2010; Chorin and Tu 2009; Morzfeld et al. 2012) attempts to prevent sample impoverishment by focusing 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 data in one sweep, and the implicit particle filter that assimilates 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 generates samples from a hard-to-sample pdf p using an easy-to-sample pdf p_0 (Hammersley and Handscomb 1964; Kalos and Whitlock 1986; Chorin and Hald 2006; Doucet et al. 2001; Geweke 1989). In this context, the density p we want to sample, but cannot sample easily, 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. One can get a sample of x by generating a sample $X \in R^d$ (we use capital letters for realizations of random variables) from the importance density p_0 and assigning to it the weight

$$w(X) = \frac{p(X)}{p_0(X)}. \tag{1}$$

The weighted samples $\{X, w\}$ form an empirical estimate of the target pdf p. This empirical estimate approximates the target pdf weakly. That means that we can approximate the expected value, $E_p[u(x)] = \int u(x)p(x)dx$, of a function u with respect to the density p, by

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

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

The efficiency of importance sampling depends on the choice of the importance func-

tion. 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 in $p = e^{-F(x)}$ (this is natural in data assimilation, see section 3a) and, for a moment, assume that

$$F(x) = -\log p(x), \tag{3}$$

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

$$F(X) - \phi_F = G(\xi) - \phi_G, \tag{4}$$

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 of 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

$$\psi : \xi \to X. \tag{5}$$

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 $X = \psi(\xi)$

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

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

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

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 (Chorin et al. 2010; Chorin and Tu 2009; Morzfeld et al. 2012) and we will review two of these maps below.

Before construction of a map we need 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). In practice one should choose a reference density that is easy to sample and easy to minimize. For example, in (Chorin et al. 2010; Chorin and Tu 2009; Morzfeld et al. 2012), 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). It is important to realize that 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 give two examples of a map ψ to show that our construction is easy to implement.

(a) A random map. With a Gaussian reference variable, equation (4) becomes

$$F(X) - \phi_F = \frac{1}{2}\xi^T\xi, \tag{8}$$

where the superscript T denotes a transpose. We can solve this equation by looking for solutions in a given, but random, direction $\eta = \xi/(\xi^T\xi)$, i.e. we use a mapping ψ such that

$$X = \mu + \lambda\eta,$$

where $\mu = \text{argmin } F$ is the minimizer of F and λ is a scalar that depends on ξ. Substitution of the above mapping into (8) gives a scalar equation in one variable (regardless of the dimension of the state space). This equation can be readily solved and the Jacobian is also easy to calculate (see Morzfeld et al. (2012)).

(b) Quadratic approximation of F. Alternatively, one can expand F around its minimum

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

where H is the Hessian of F, evaluated at the minimizer. To obtain a sample, we then solve the quadratic equation

$$F_0(x) - \phi = \frac{1}{2}\xi^T\xi, \tag{9}$$

instead of (8). This can be done, for example, by using the Cholesky factor L of H:

$$X = \mu + L^{-T}\xi. \tag{10}$$

The expansion of F as well as the above equation to obtain samples are familiar from Laplace's method (Kass et al. 1990; Kass and Raftery 1995). However, we weigh the samples to remove the bias and to obtain samples from the target distribution (not its Gaussian approximation as in Laplace's method). For the weights, we need to compute the Jacobian of the linear mapping (10), which is the inverse of the determinant of L (the product of its diagonal entries). Thus, the Jacobian is a constant for all particles and drops out after normalization of the weights. Further, we need to account for the error we made by solving (9) rather than (8) by attaching the weight

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

to the samples. This "approximate" map is very efficient if the Hessian of F is available and was presented in Chorin et al. (2010). The construction is also related to the "importance distribution obtained by local linearization" in Doucet et al. (2000). There, the authors approximate the optimal importance function by a Gaussian centered at the mode of the optimal importance function and with a covariance matrix equal to the Hessian of this pdf. However, implicit sampling can be more efficient, because it does not make direct use of the optimal importance function, which is, in general, hard to compute.

Further constructions of suitable mappings ψ are presented in (Chorin et al. 2010); we note that generating samples is "easy," (numerically inexpensive) compared to finding the minimum F.

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 $F(X) \to \infty$ as $|X| \to \infty$; 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 (Chorin et al. 2010). 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 Chorin and Tu (2009); Chorin et al. (2010); Morzfeld et al. (2012) and highlights the variational aspects of the implicit particle methods.

a. 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

$$x_{j+1} = R_j(x_j) + G_j(x_j)Z_j, \tag{11}$$

where $j = 0, 1, 2, \ldots$ can be thought of as discrete time; the state, x_j, is a d-dimensional real vector, R_j is a d-dimensional vector function, G_j is a real $d \times d$ matrix 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 (Kloeden and Platen 1999), and we assume the Z_j's to be Gaussian with mean zero and covariance S 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 Morzfeld and Chorin (2012).

The data

$$y_k = h(x_{n_k}) + V_k, \tag{12}$$

indexed by $k = 1, 2, \ldots,$ 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 $x_0, \ldots, x_k$; we refer to a vector y_k as an "observation" (in geophysical papers, y_k is also often called an observation vector).

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

$$p(x_{0:n_m}|y_{1:m}) = p(x_0)\frac{\prod_{j=1}^{n_m} p(x_j|x_{j-1}) \prod_{j=1}^{m} p(y_j|x_{n_j})}{p(y_{1:m})}. \tag{13}$$

Since we know $p(x_0)$, and can read $p(x_j|x_{j-1})$ and $p(y_j|x_j)$ from equations (11) and (12), we know this pdf up to the normalization constant $p(y_{1:m})$, which is hard to compute.

b. The implicit particle smoother

To assimilate the observations, we can apply implicit sampling to the conditional pdf in (13). Since an importance sampling scheme that uses the observations to estimate past and current states is often called a particle smoother (Doucet et al. 2001), we will call this method the implicit particle smoother.

The target pdf is the conditional pdf in (13), 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 (12) is Gaussian with mean zero and covariance matrix Q, then this F is

$$\begin{aligned} F(x_{0:n_m}) = &-\log(p(x_0)) \\ &+ \frac{1}{2}\sum_{j=0}^{n_m-1}(x_{j+1} - R_j(x_j))^T\Sigma_j^{-1}(x_{j+1} - R_j(x_j)) \\ &+ \frac{1}{2}\sum_{j=1}^{m}(y_j - h(x_{n_j}))^T Q^{-1}(y_j - h(x_{n_j})) + C, \end{aligned} \tag{14}$$

where $\Sigma_j = G_j(x_j)^T S G_j(x_j)$, 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. Conn et al. (2000); Fletcher (1987); Nocedal and Wright (2006)) and choose a Gaussian reference variable $\xi \sim \mathcal{N}(0, I)$. In this case the algebraic equation (4) becomes (8), which we solve with a suitable mapping ψ (see Chorin et al. (2010); Chorin and Tu (2009); Morzfeld et al. (2012)) 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 $p(x_{0:n_m})$. We can use this approximation to compute a state estimate, for example, the weighted sample average. The weighted sample average approximates the conditional mean $E(x_{0:n_m}|y_{1:m})$, which, under wide conditions, is the minimum mean squared error estimate of the state (see e.g. Chorin and Hald (2006)).

c. 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 Doucet et al. (2001))

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

Given a set of M weighted samples $\{X^k_{0:n_m}, w^k\}$ (particles), $k = 1, \ldots, M$, that form an empirical estimate of the conditional pdf $p(x_{0:n_m}|y_{1:m})$ at time n_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

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

with updated weights

$$\hat{w}^k = w^k\frac{p(X^k_{n_m+1:n_{m+1}}|X^k_{n_m})p(y_{m+1}|X^k_{n_{m+1}})}{p_0(X^k_{n_m+1:n_{m+1}})}. \tag{15}$$

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 $p(X^k_{n_i+1:n_{i+1}}|X^k_{n_i})p(y_{i+1}|X^k_{n_{i+1}})$ 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

$$F^k(x_{n_m+1:n_{m+1}}) = -\log(p(x_{n_m+1:n_{m+1}}|X^k_{n_m})p(y_{m+1}|x_{n_{m+1}})). \tag{16}$$

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

$$\begin{aligned} F^k(x_{n_m+1:n_{m+1}}) = &\frac{1}{2}(x_{n_m+1} - R_{n_m}(X^k_{n_m}))^T\Sigma^{-1}_{n_m}(x_{n_m+1} - R_{n_m}(X^k_{n_m})) \\ &+ \frac{1}{2}\sum_{j=n_m+1}^{n_{m+1}-1}(x_{j+1} - R_j(x_j))^T\Sigma^{-1}_j(x_{j+1} - R_j(x_j)) \\ &+ \frac{1}{2}(y_{m+1} - h(x_{n_{m+1}}))^TQ^{-1}(y_{m+1} - h(x_{n_{m+1}})) + C \end{aligned}$$

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. Conn et al. (2000); Fletcher (1987); Nocedal and Wright (2006)). We then pick a Gaussian reference variable $\xi \sim \mathcal{N}(0, I)$ and obtain M samples, $X^k_{n_m+1:n_{m+1}}$, by solving the M equations

$$F^k(X^k_{n_m+1:n_{m+1}}) - \phi^k = \frac{1}{2}\xi^T\xi, \tag{17}$$

with a suitable mapping ψ (see Chorin et al. (2010); Chorin and Tu (2009); Morzfeld et al. (2012)). The update equation for the weights can be obtained by combining (7) with (15):

$$\hat{w}^k = w^k e^{-\phi^k} \left| J(X^k_{n_m+1:n_{m+1}}) \right| \tag{18}$$

where J is the Jacobian of ψ. We append the M samples $X^k_{n_m+1:n_{m+1}}$ to the M particles we already had, and replace their weight with the updated weight from (18). 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}. We can use this approximation to compute the weighted sample average as an approximation conditional mean as explained above.

The weights are now removed by "resampling," a process in which particles with a low weight are replaced by particles with a larger weight. There is an extensive literature on resampling algorithms (see e.g. Doucet et al. (2001); Liu and Chen (1995); Moral et al. (2012); Smith and Gelfand (1992)). We use algorithm 2 in (Arulampalam et al. 2002), which can be implemented in $O(M)$ operations (M being the number of particles). The performance and efficiency of the overall sequential Monte Carlo method depends on the choice of the resampling algorithm. However, our goal here is to discuss how to reduce sample impoverishment by judiciously choosing the importance function. A discussion of how resampling comes into play is deferred to other papers.

Note that the term $\exp(-\phi^k)$ in (18) induces additional variability into the weights when compared to the implicit particle smoother in section 2b, 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, $X^k_{n_m}$, at time n_m (see equation (16)). If the particles at time n_m are in the high-probability region, and if this high-probability region has a (sharp) peak, then the functions F^k are all "similar," and the minima ϕ^k of these functions should not vary too much from particle to particle. In this case, the variance induced by the exponential term can be expected to be small. The numerical experiments in section 5, as well as those in Chorin et al. (2010); Morzfeld et al. (2012) confirm this statement, however a rigorous analysis of the variance of the weights of the implicit particle filter has not been reported.

Finally we want to compare our construction with the particle method in van Leeuwen (2010). The idea there is to construct an importance function that focuses the particles on the high-probability region by use of a nudging term (i.e. by changing the underlying dynamics in (11)). In order to achieve the focusing effect, a significant amount of tuning is required. The implicit particle filter searches for the high-probability regions using numerical minimizations and, therefore, seems to be more methodical and more straightforward to implement.

d. 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 equation (11), 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, $k = 1, \ldots, m$.

The implicit particle smoother in section 3b can be easily adapted to this situation by applying implicit sampling to the conditional pdf $p(x_0|y_{1:m})$. Note, however, that the implicit

smoother for a perfect model does not estimate the state at times $t > 0$, because the future states are determined by the model; the implicit smoother in section 3b however estimates the full state trajectory because the model is stochastic.

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 (12). The pdf $p(x_0)$ is called the prior density and is often chosen to be Gaussian. However, the conditional pdf is generally not Gaussian, because h can be nonlinear and the x_{n_j}'s are nonlinear functions of x_0 (see (11)).

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

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

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

$$F(x_0) = -\log\left(p(x_0)\right) + \sum_{j=1}^{m} (h(x_{n_j}) - y_j)^T Q^{-1} (h(x_{n_j}) - y_j) + C, \tag{19}$$

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 (8) repeatedly for M realizations of ξ. We can solve this scalar equation efficiently using e.g. random maps as in Morzfeld et al. (2012), or one of the methods in Chorin et al. (2010). 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 can be modified to assimilate data sequentially, i.e. assimilate $k < m$ observations in one computation. 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 (Kalnay et al. 2007). 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 (Fournier et al. 2010), however the details, as well as numerical tests for sequential implicit sampling for this problem are deferred to 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 $p(x_{0:n_m}|y_{1:m})$. Data assimilation schemes that combine the ensemble Kalman filter (EnKF) in with variational methods are a current research topic (see e.g. (Liu et al. 2008; Buehner 2005; Hunt et al. 2004; Fertig et al. 2007)). The idea is to use the Monte Carlo simulations of the EnKF to update the covariance matrices required for the variational calculations. Here, we 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.

a. Connection with strong constraint 4D-Var

Strong constraint 4D-Var (see e.g. Dimet and Talagrand (1986); Rabier and Courtier (1992); Talagrand and Courtier (1987); Talagrand (1997); Courtier (1997); Courtier et al. (1994)), 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 (11), by minimization of a suitable cost function. 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

$$\mathcal{J}_s(x_0) = (x_0 - x_b)^T B^{-1}(x_0 - x_b) + \sum_{j=1}^{m}(h(x_{n_j}) - y_j)^T Q^{-1}(h(x_{n_j}) - y_j), \tag{20}$$

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. Talagrand and Courtier (1987)). 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 (20) is identical to F in (19) (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 3d) thus amounts to adding a sampling and weighting step, which in turn amounts to solving the scalar equation (8), or more generally (4). Efficient methods for executing the sampling and weighting can be found in (Chorin et al. 2010; Morzfeld et al. 2012), so that the additional computational cost of implicit particle smoothing is small. For example, if the Hessian of F is available, then the approximate map (b) in section 2 amounts to a matrix vector multiplication (and this matrix can be sparse). If the Hessian is not available, one can use the random map (a) of section 2. In this case, one can use Newton's method to solve (8) for which a few adjoint calculations are required (one for each step of the Newton method).

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,

which, in general, is a biased state estimate. Moreover, the implicit particle smoother naturally produces a quantification of the uncertainty, because it generates an empirical estimate of the conditional pdf. The implicit particle smoother therefore can easily deal with skew or multimodal posterior pdfs, whereas 4D-Var codes typically provide error estimates based on a Gaussian approximation of the posterior pdf (Rabier and Courtier 1992).

When the data are sparse in space or time, the conditional pdf can have 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 can be superior to the results of a 4D-Var code that represents the conditional pdf by only one of its modes.

b. Connection with weak constraint 4D-Var

Weak constraint 4D-Var (see e.g. Bennet et al. (1993); Kalnay (2003); Kurapov et al. (2007)) relaxes the perfect model assumption made in strong constraint 4D-Var. There are several ways of doing so (Tremolet 2006), 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

$$\begin{aligned}\mathcal{J}_w(x_{0:n_m}) &= (x_0 - x_b)^T B^{-1}(x_0 - x_b) \\ &+ \sum_{j=0}^{n_m-1} (x_{j+1} - R_j(x_j))^T \Sigma_j^{-1}(x_{j+1} - R_j(x_j)) \\ &+ \sum_{j=1}^{m} (y_{jk} - h(x_{n_j}))^T Q^{-1}(y_j - h(x_{n_j})). \end{aligned} \tag{21}$$

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. Kalnay (2003); Zupanski (1997)).

Note that the cost function $\mathcal{J}_w$ in (21) equals F in (14), the function that is minimized by the implicit particle smoother of section 3b (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 (8) repeatedly. The cost of solving these equations is not large, compared to the computational cost of minimizing the cost functions, as was explained in section 4a. Thus, the additional cost for implementing the implicit particle smoother versus a weak constraint 4D-Var method is not large. The implicit particle smoother has the advantage that it can compute the conditional mean, which can be a better state estimate than the conditional mode (the result of a weak 4D-Var calculation), because the conditional mean minimizes the mean square error, and is unbiased, whereas the conditional mode is a biased state estimate. 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 3c 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 (16). 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 (21). 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 3c 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 (21) by removing the background state, turning the *variables* x_0 into *parameters* $X^k_{n_m}$ (the position of the kth 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 (Chorin et al. 2010; Morzfeld et al. 2012). As was explained above, the additional cost of generating the samples is small compared to finding the minimum of the F^k's. Moreover, the minimization for each particle is very easy to parallelize so that the the implicit particle filter can make use of modern computer architectures with multiple processors.

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 constraint 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 4a 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 (Miller et al. 1999; Evensen 1997; Chorin and Krause 2004) and apply the implicit particle filter of section 3c and the implicit particle smoother of section 3d to the Lorenz attractor (Lorenz 1963). We distinguish between the strong and weak constraint problem. The goal is to demonstrate the implementation of the implicit particle methods based on 4D-Var codes, and to show the benefits one can expect from turning a 4D-Var code into an implicit filter. However, the conclusions one can draw from this (simple) example about more realistic models in numerical weather prediction (where the models can have millions of state variables) are somewhat limited.

a. The strong constraint problem

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

$$\frac{dx^1}{dt} = \sigma(x^2 - x^1), \quad \frac{dx^2}{dt} = x^1(\rho - x^3) - x^2, \quad \frac{dx^3}{dt} = x^1 x^2 - \beta x^3, \tag{22}$$

where $\rho = 28$, $\sigma = 10$, $\beta = 8/3$ (see Lorenz (1963) who used the symbols σ, r and b). We discretize these equations using a fourth-order Runge-Kutta scheme with constant time step $\delta = 0.01$. We observe that the errors of this discretization have converged (in the time-step) for the short integration times we consider, so that we expect that our numerical solution is a good approximation of the true solution of the Lorenz '63 equations.

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 (12) thus becomes

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

with $V_k \sim \mathcal{N}(0, 2I_2)$. 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 $y_1, \ldots, y_4$. We try to achieve this goal by using the implicit particle smoother of section 3d.

Recall that the implicit particle smoother essentially consists of three steps: (i) minimize the function F in (19); (ii) obtain samples from the underlying conditional pdf by solving the algebraic equation (8); and (iii) weight the samples using (7). As pointed out in section 4a, the first step can be carried out using adjoint codes and that is what we did for this example.

1) Variational implementation of the implicit particle smoother

We constructed the linear tangent adjoint of the continuous time ODE's in (22) 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. Nocedal and Wright (2006); Fletcher (1987)) for the minimization of F. We can use the adjoint of the continuous equations here because our discretization is accurate enough to do so (in other applications however it may be necessary to compute the adjoint of the discrete-time equations).

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. Typically the minimization converged after a few steps. We observed occasionally that the minimization was trapped in very flat regions, in which case we re-started the whole process, using a fresh sample from the prior density p_0 to initialize the minimization.

2) Implementation of the map ψ

To generate samples, we follow Chorin et al. (2010) and choose the approximate map that makes use of a Gaussian reference variable and the quadratic expansion of F in (9) (see section 2). The Hessian of F in (9) is hard to compute, but, instead, we can use the approximate Hessian, which is available from the variational minimization using BFGS. To obtain a sample, we thus solve the quadratic equation (9), where H is the approximate Hessian of F, evaluated at the minimizer. 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 (9) rather than (8) by attaching to each sample the weight (11). This map is very efficient for this problem, because L is 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 general, the implementation of this approximate map requires a one-time calculation of the Cholesky factor of the approximate Hessian of F; for each sample it requires a matrix-vector multiplication of a triangular matrix. If the (approximate) Hessian is not available (or the cost of storing it is too large), the random map approach in Morzfeld et al. (2012) can be used because it can be implemented without using second derivatives of F (see also section 2 and Morzfeld and Chorin (2012)).

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 (22) 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 represent 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, because it practically coincides with 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 (most of them are within 2 standard deviations of the data). The samples from the prior (blue) are often not compatible with the data (they are too far away from the data points) and, therefore, are unlikely with respect to the posterior density. The computations spent on generating these samples is essentially wasted (which is why this method is computationally less effective than implicit sampling).

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 for times $t \geq 1.4$. We could, of course, also integrate the particles (i.e. the initial conditions) up to the desired forecast time say $t = 1.4$. However, the point here is to indicate that the Gaussian approximation we obtained for the state at time $t = 0.8$ is compatible with the true state trajectory for times $t \geq 0.8$. This indicates that this Gaussian approximation can be used as a prior pdf to assimilate data collected at $t \geq 0.8$.

We further assessed the accuracy and reliability of the implicit particle smoother by running 10,000 twin experiments. A twin experiment amounts 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 than 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 we obtained with a Bayesian bootstrap method (Doucet et al. 2001). 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. The results of 10,000 twin experiments are shown in table 1

We observe that the Bayesian bootstrap method and the implicit particle smoother give the same errors. Since both methods approximate the conditional mean, we can conclude that our implementation of the implicit particle smoother is correct. Moreover, 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 a relatively small additional computational cost (0.6%). Moreover, the implicit particle smoother delivers a quantitative measure of the uncertainty of the 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 it its variational implementation.

b. The weak constraint problem

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

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

where W^1, W^2 and W^3 are independent Brownian motions and where σ, ρ and β are as in section 5a and $g = 1/\sqrt{2}$. We discretize these stochastic differential equations (SDE) using the Euler-Maruyama scheme with constant time step $\delta = 10^{-3}$ (Kloeden and Platen 1999). With this choice the function $R_j(x_j)$ for the discrete recurrence (11) becomes

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

where $x_j = \left(x_j^1, x_j^3, x_j^3\right)^T$ and

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

and $Z_k \sim \mathcal{N}(0, \delta/2I_3)$.

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 $T = 0.5$ (Miller et al. 1999). 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; and (b) $r = 800$, i.e. the gap between observations is 0.8 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 3c.

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 (16). 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. Conn et al. (2000)). This corresponds to an "ideal" implementation of weak constraint 4D-Var, for which the control variable is the full state trajectory (Tremolet 2006).

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 $\overline{x}_m = x_{n_m+1:n_{m+1}}$ 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 $\overline{x}^r_j = \overline{x}_j - r(j/r)$ for each $j = 0, 1, 2 \ldots, r$. This procedure rotates the model path $\overline{x}_j$ towards the observation.

We refine this seed with a multi-grid technique, which is conceptually similar to the multi-grid finite difference method (Fedorenko 1961) and multi-grid Monte Carlo (Goodman and Sokal 1989) (see also Chorin (2008)). 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 $\tilde{\delta}$ is similar to the minimizer of an F^k with a model and time step $\delta < \tilde{\delta}$.

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 a coarse grid minimization is rejected if the minimum is above a threshold ϕ_c, and we restart the minimization with a new (unrefined) seed $\overline{x}_m$.

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).

644 2) Implementation of the map ψ

645 Upon minimization of the F^k's, we solve (17) for each particle to obtain samples from
646 the conditional pdf. To solve this underdetermined equation, we use the same approach
647 as in section 5a, i.e. we replace F^k in (17) by its quadratic approximation and solve a
648 quadratic equation. This approach is very efficient for this problem, because we can solve
649 the quadratic equation using the Cholesky factor, L, of the Hessian of F^k, which is available
650 from the trust-region minimization (the variational part of the implicit particle filter). The
651 Jacobian of this map is easily calculated to be the determinant of L (the product of its
652 diagonal entries). We observed that, for this example, generating a sample using this map
653 takes about 1/10,000 of the time it takes to carry out the minimization; in general, obtaining
654 a sample requires a Cholesky factorization of H (which we already have from the Newton
655 minimization), followed by a matrix-vector multiplication (where the matrix is triangular).
656 The cost of sampling is thus small compared to the cost of minimizing F^k, i.e. turning a
657 weak 4D-Var code into implicit sampling code comes at a relatively small additional cost.
658 Again, we account for the error we make by replacing F^k by its quadratic approximation
659 through the weights, which become

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

660 Note that the factors with ϕ^k and the Jacobian of the map ($\det L^{-1}$) must appear in the
661 weights because the functions F^k are different for each particle and, thus, can have different
662 minima and different Hessians. In other problems, these Hessians may not be available (or
663 too large to store). In these cases, the random map approach can be implemented without
664 using second derivatives of the F^k's (see section 2 and (Morzfeld et al. 2012; Morzfeld and
665 Chorin 2012)).

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 (Gordon et al. 1993; Liu et al. 2001, 1998). 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.

4) Numerical results

We test the efficiency and accuracy of the implicit particle filter by running twin experiments, as we did in section 5a. 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 3c); (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 (Gordon et al. 1993;

Doucet et al. 2001); 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 cannot 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 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 (because we essentially observe the same errors for the implicit particle filter with 10 and 20 particles).

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 $e = x^F_{0:n} - x_{0:n}$, where $x_{0:N}$ is the true state trajectory and $x^F_{0:N}$ 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 cannot achieve the accuracy of the particle filters, especially if the 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 (in this example), due to the particle-by-particle minimization. The additional computations of turning the variational method into an implicit particle filter however makes it possible to track all relevant modes.

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 (Doucet et al. 2001) and we compare the normalized effective sample size of the implicit particle filter and the SIR filter. The results are shown in table 3.

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. While the computational cost of an SIR filter with about 500 particles is comparable to the implicit filter with 10 particles (each boosted with 50 particles), the ensemble produced by the implicit particle filter is of higher quality, as is indicated by a significantly larger effective sample size.

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.

Finally, we wish to mention that we ran numerical experiments with an EnKF, using the Matlab implementation available at *www.enkf.nersc.co*. We experimented in both the weak and strong constraint problem set ups and came to the conclusion that our time-gap between observations is too large for the EnKF to give accurate state estimates between the observations. These observations are in line with the results reported in the detailed comparative study of Kalnay et al. (2007).

6. Conclusions

The implicit particle filter was introduced in (Chorin et al. 2010; Chorin and Tu 2009; Morzfeld et al. 2012) 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 Chorin et al. (2010); Morzfeld et al. (2012)). 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 error quantifications that often rely on Gaussian approximations.

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 10% 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.

Acknowledgments.

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

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp, 2002: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, **50 (2)**, 174 –188.

Bennet, A. F., L. M. Leslie, C. R. Hagelberg, and P. E. Powers, 1993: A cyclone prediction using a barotropic model initialized by a general inverse method. *Monthly Weather Review*, **121**, 1714–1728.

Bickel, P., T. Bengtsson, and J. Anderson, 2008: 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.

Buehner, M., 2005: Ensemble-derived stationary and flow dependent background error covariances: evaluation in a quasi-operaion nwp setting. *Quarterly Journal of the Royal Meteorological Society*, **131**, 1013–1043.

Cappé, O., R. Douc, A. Guillin, J. Marin, and C. Robert, 2008: Adaptive importance sampling in general mixture classes. *Statistics and Computing*, **18 (4)**, 447–459.

Carpenter, J., P. Clifford, and P. Fearnhead, 1999: Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings* -, **146 (1)**, 2 –7.

Chorin, A. J., 2008: Monte Carlo without chains. *Communications in Applied Mathematics and Computational Science*, **3**, 77–93.

Chorin, A. J. and O. H. Hald, 2006: *Stochastic Tools in Mathematics and Science*. 1st ed., Springer.

Chorin, A. J. and P. Krause, 2004: Dimensional reduction for a Bayesian filter. *Proceedings of the National Academy of Sciences*, **101 (42)**, 15 013–15 017.

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

Chorin, A. J. and X. Tu, 2009: Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, **106 (41)**, 17 249–17 254.

Conn, A. R., N. I. M. Gould, and P. L. Toint, 2000: *Trust-region methods*. 1st ed., Society for Industrial and Applied Mathematics.

Cornebise, J., E. Moulines, and J. Olsson, 2008: Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, **18 (4)**, 461–480.

Courtier, P., 1997: Dual formulation of four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **123**, 2449–2461.

Courtier, P., J. Thepaut, and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, **120**, 1367–1387.

Dimet, F. X. L. and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, **38A (2)**, 97–110.

Doucet, A., N. de Freitas, and N. Gordon, (Eds.) , 2001: *Sequential Monte Carlo Methods in Practice*. Springer.

Doucet, A., S. Godsill, and C. Andrieu, 2000: On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10**, 197–208.

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

Evensen, G., 1997: Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, **125**, 1342–1354.

Evensen, G., 2006: *Data assimilation: the ensemble Kalman filter*. Springer.

Fedorenko, R. P., 1961: A relaxation method for solving elliptic difference equations. *USSR Computational Mathematics and Mathematical Physics*, **1**.

Fertig, E. J., J. Harlim, and B. R. Hunt, 2007: A comparative study of 4D-Var and a 4D ensemble Kalman filter: perfect model simulations with Lorenz-96. *Tellus*, **59 (A)**, 96–100.

Fletcher, R., 1987: *Practical Methods of Optimization*. 2d ed., Wiley.

Fournier, A., et al., 2010: An introduction to data assimilation and predictability in geomagnetism. *Space Science Reviews*, **155 (1-4)**, 247–291.

Geweke, J., 1989: Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, **24**, 1317 – 1399.

Goodman, J. and A. D. Sokal, 1989: Multigrid Monte Carlo method. Conceptual foundations. *Phys. Rev. D*, **40**, 2035–2071.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith, 1993: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, **140 (2)**, 107 –113.

Hammersley, J. M. and D. Handscomb, 1964: *Monte Carlo Methods*, Vol. 1. 1st ed., Methuen young books.

Hunt, B. R., et al., 2004: Four-dimensional ensemble Kalman filtering. *Tellus*, **56 (A)**, 273–277.

Johansen, A. M. and A. Doucet, 2008: A note on auxiliary particle filters. *Statistics & Probability Letters*, **78 (12)**, 1498 – 1504.

Kalman, R. E., 1960: A new approach to linear filtering and prediction theory. *Transactions of the ASME–Journal of Basic Engineering*, **82 (Series D)**, 35–48.

Kalman, R. E. and R. S. Bucy, 1961: New results in linear filtering and prediction theory. *Transactions of the ASME–Journal of Basic Engineering*, **83 (Series D)**, 95–108.

Kalnay, E., 2003: *Atmospheric modeling, data assimilation and predictabilty*. Cambridge University Press.

Kalnay, E., H. Li, T. Miyoshi, S. C. Yang, and J. Ballabrera-Poy, 2007: 4-D-Var or ensemble Kalman filter. *Tellus*, **59A**, 758–773.

Kalos, M. H. and P. A. Whitlock, 1986: *Monte Carlo Methods*, Vol. 1. 1st ed., John Wiley & Sons.

Kass, R. and A. Raftery, 1995: Bayes factors. *Journal of the American Statistical Association*, **90**, 773–795.

Kass, R., L. Tierny, and J. Kadane, 1990: The validity of posterior expansions based on Laplace's method. *Bayesian and Likelihood methods in Statistics and Econometrics*, **7**, 473–488.

Kloeden, P. E. and E. Platen, 1999: *Numerical Solution of Stochastic Differential Equations*. 3d ed., Springer.

Kurapov, A., G. D. Egbert, J. S. Allen, and R. N. Miller, 2007: Representer- based variational data assimilation in a nonlinear model of nearshore circulation. *Journal of Geophysical Research*, **112**, C11 019.

Liu, C., Q. Xiao, and B. Wang, 2008: An ensemble-based four-dimensional variational data assimilation scheme. Part I: technical formulation and preliminary test. *Monthly Weather Review*, **136**, 3363–3373.

Liu, J. S. and R. Chen, 1995: Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, **90 (430)**, pp. 567–576.

Liu, J. S., R. Chen, and T. Logvinenko, 2001: A theoretical framework for sequential importance sampling with resampling. *Sequential Monte Carlo Methods in practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., Springer, chap. 11.

Liu, J. S., R. Chen, and W. H. Wong, 1998: Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, **93 (443)**, pp. 1022–1031.

Lorenz, E. N., 1963: Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, **20**, 130–148.

Miller, R. N., E. F. Carter, and S. T. Blue, 1999: Data assimilation into nonlinear stochastic models. *Tellus A*, **51 (2)**, 167–194.

Moral, P. D., A. Doucet, and A. Jasra, 2012: On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, **18 (1)**, 252–278.

Morzfeld, M. and A. J. Chorin, 2012: Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation. *Nonlinear Processes in Geophysics*, **19**, 365–382.

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

Nocedal, J. and S. T. Wright, 2006: *Numerical Optimization*. 2d ed., Springer.

Pitt, M. and N. Shephard, 1999: Filtering via simulation: auxiliary particle fitlers. *Journal of the American Statistical Association*, **94 (446)**, 590–599.

Rabier, F. and P. Courtier, 1992: Four-dimensional assimilation in the presence of baroclinic instability. *Quarterly Journal of the Royal Meteorological Society*, **118 (506)**, 649–672.

Smídl, V. and R. Hofman, 2012: Application of sequential Monte Carlo estimation for early phase of radiation accident. *Technical Report*, **UTIA**.

Smith, A. F. M. and A. E. Gelfand, 1992: Bayesian statistics without tears: a sampling-resampling perspective. *The American Statistician*, **46 (2)**, pp. 84–88.

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

Talagrand, O., 1997: Assimilation of observations, an introduction. *Journal of the Meteorological Society of Japan*, **75 (1)**, 191–209.

Talagrand, O. and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quarterly Journal of the Royal Meteorological Society*, **113 (478)**, 1311–1328.

Tremolet, Y., 2006: Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, **132 (621)**.

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

van Leeuwen, P. J., 2011: Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers & Fluids*, **46 (1)**, 52 – 58.

Zupanski, D., 1997: A general weak constraint applicable to operational 4DVAR data assimilation systems. *Monthly Weather Review*, **125**, 2274–2292.

List of Tables

4D-Var	Implicit particle smoother	Bayesian bootstrap
	(100 particles)	(1000 particles)
0.063 / 0.027	0.047 / 0.023	0.046 / 0.022

TABLE 1. Errors (mean / standard deviation) in the reconstruction of the initial condition for three data assimilation techniques for the strong constraint problem.

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
250	-	-	0.039 / 0.0098
500	-	-	0.038 / 0.0089
1000	-	-	0.038 / 0.013
5000	-	-	0.037 / 0.0087

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
250	-	-	0.066 / 0.055
500	-	-	0.063 / 0.054
1000	-	-	0.065 / 0.056
5000	-	-	0.064 / 0.056

TABLE 2. Errors (mean / standard deviation) of three data assimilation techniques for the weak constraint problem. 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.

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 %
250	-	48.3 %
500	-	48.4 %
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 %
250	-	34.3 %
500	-	34.0 %
1000	-	33.0%
5000	-	33.0%

TABLE 3. Normalized effective sample size of the implicit particle filter and the standard SIR filter for the weak constraint problem.

List of Figures

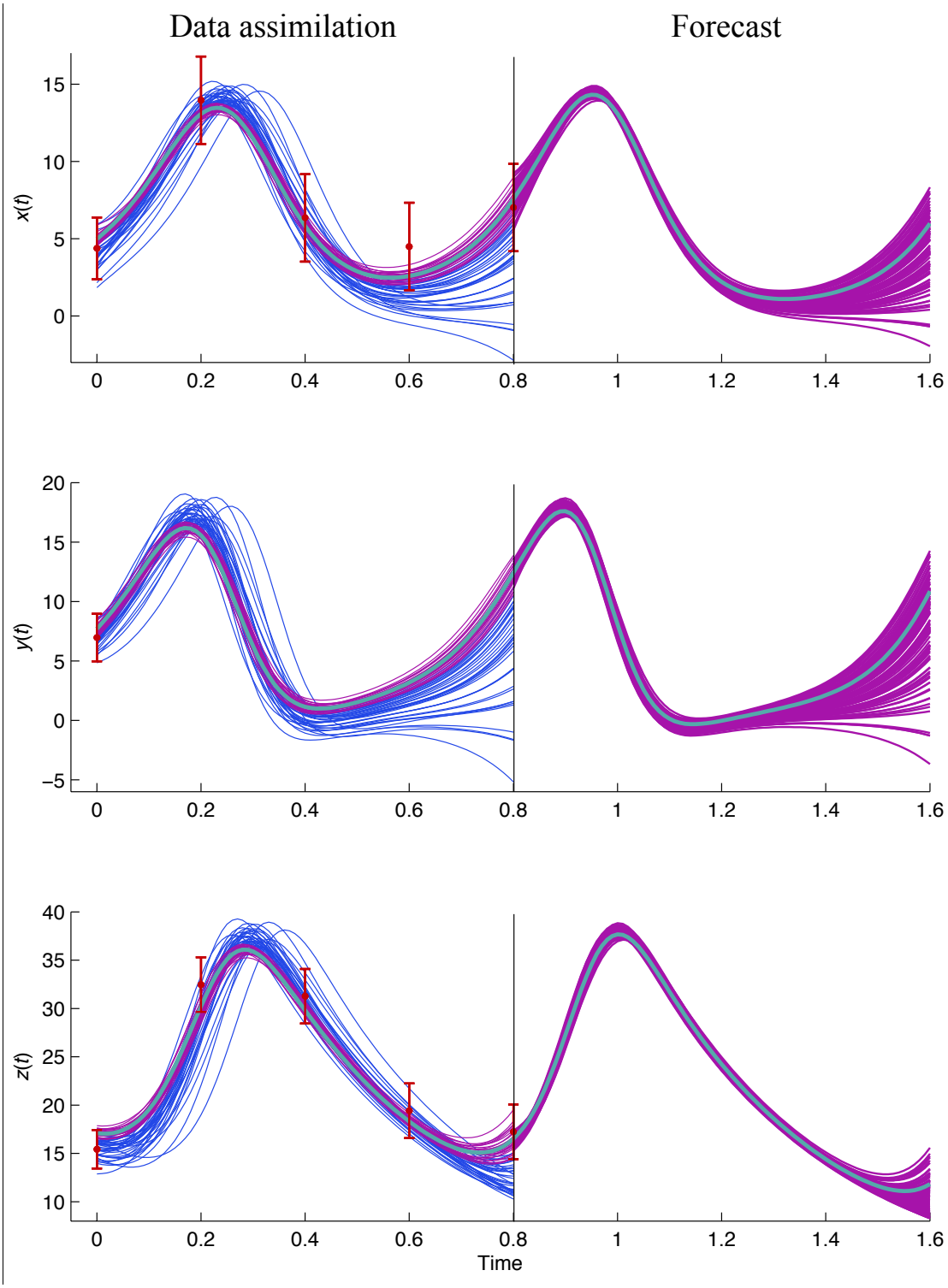

FIG. 1. Illustration of data assimilation and forecasting using the implicit particle smoother for the strong constraint problem. On the left (Time $\leq$ 0.8): 30 samples from the prior pdf (blue lines); the data and error bars (red); 25 samples obtained by the implicit particle smoother (purple); and 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).

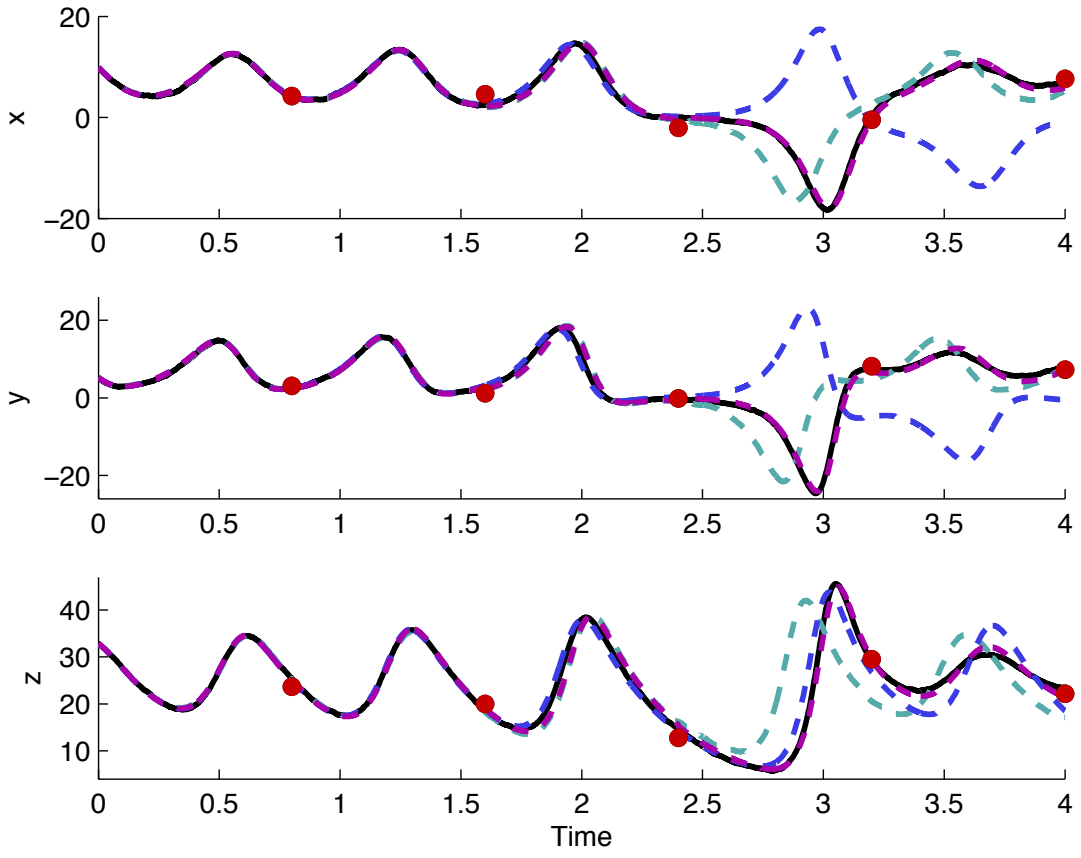

FIG. 2. Reconstructions of a reference path (solid-black) from a set of 5 observations (red dots) by three data assimilation methods for the weak constraint problem. Dashed-teal: reconstruction by the standard SIR filter with 20 particles. Dashed-blue: reconstruction by weak constraint 4D-Var. Dashed-purple: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