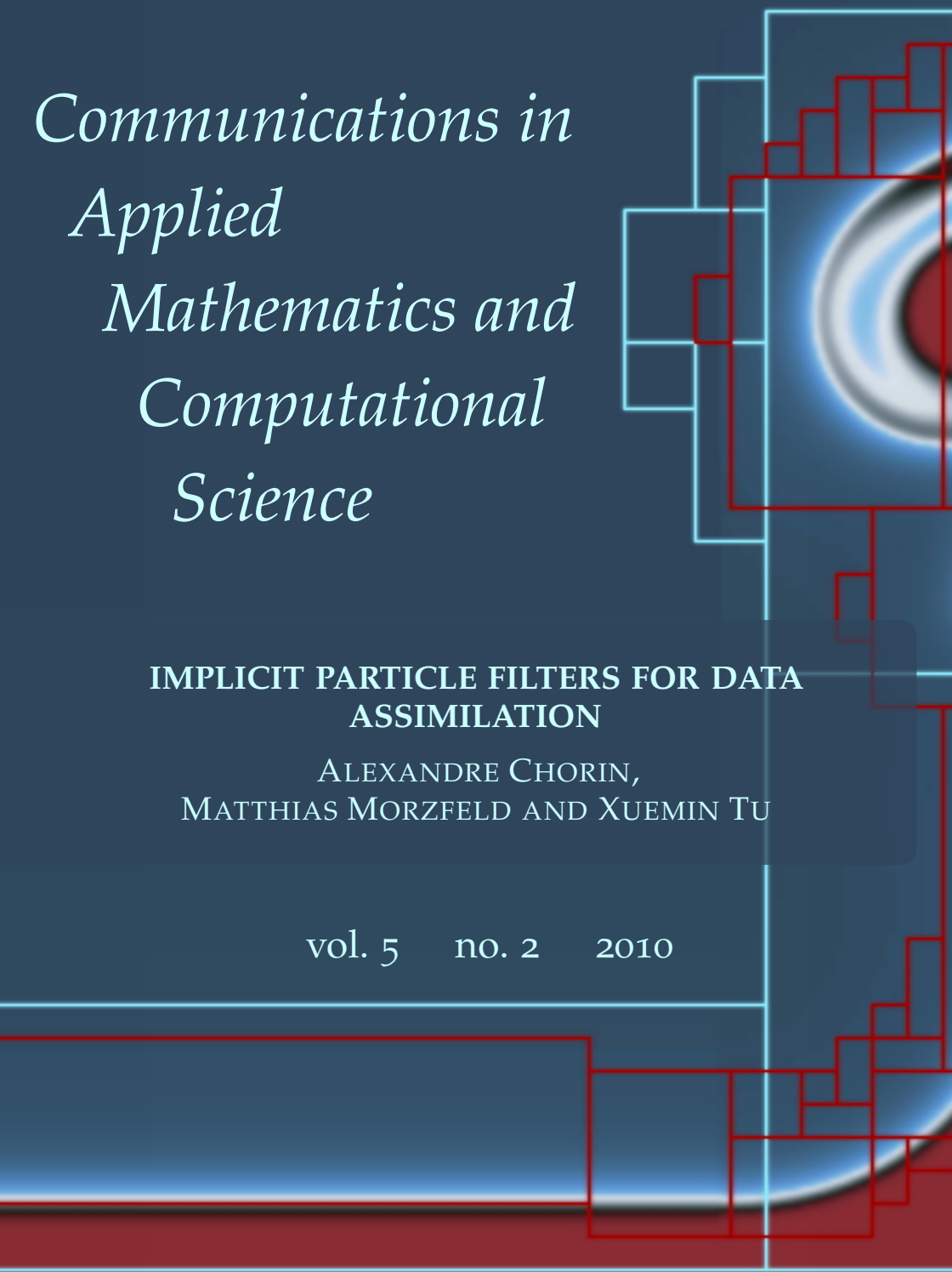
Communications in Applied Mathematics and Computational Science

IMPLICIT PARTICLE FILTERS FOR DATA ASSIMILATION

ALEXANDRE CHORIN,
MATTHIAS MORZFELD AND XUEMIN TU

vol. 5 no. 2 2010



mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 2, 2010

IMPLICIT PARTICLE FILTERS FOR DATA ASSIMILATION

ALEXANDRE CHORIN, MATTHIAS MORZFELD AND XUEMIN TU

Implicit particle filters for data assimilation update the particles by first choosing probabilities and then looking for particle locations that assume them, guiding the particles one by one to the high probability domain. We provide a detailed description of these filters, with illustrative examples, together with new, more general, methods for solving the algebraic equations and with a new algorithm for parameter identification.

1. Introduction

There are many problems in science, for example in meteorology and economics, in which the state of a system must be identified from an uncertain equation supplemented by noisy data (see, for instance, [9; 22]). A natural model of this situation consists of an Ito stochastic differential equation (SDE):

$$dx = f(x,t)\,dt + g(x,t)\,dw, \tag{1}$$

where $x = (x_1, x_2, \ldots, x_m)$ is an m-dimensional vector, f is an m-dimensional vector function, $g(x,t)$ is an m by m matrix, and w is Brownian motion which encapsulates all the uncertainty in the model. In the present paper we assume for simplicity that the matrix $g(x,t)$ is diagonal. The initial state $x(0)$ is given and may be random as well.

The SDE is supplemented by measurements b^n at times t^n, $n = 0, 1, \ldots$. The measurements are related to the state $x(t)$ by

$$b^n = h(x^n) + GW^n, \tag{2}$$

where h is a k-dimensional, generally nonlinear, vector function with $k \le m$, G is a matrix, $x^n = x(t^n)$, and W^n is a vector whose components are independent Gaussian variables of mean 0 and variance 1, independent also of the Brownian motion in (1). The independence requirements can be greatly relaxed but will be

MSC2000: 60G35, 62M20.

Keywords: implicit sampling, data assimilation, particle filter.

This work was supported in part by the Director, Office of Science, Computational and Technology Research, United States Department of Energy under Contract no. DE-AC02-05CH11231, and by the National Science Foundation under grants DMS-0705910 and OCE-0934298.

observed in the present paper. The task of a filter is to assimilate the data, that is, estimate x on the basis of both (1) and the observations (2).

If the system (1) and the function h in (2) are linear and the data are Gaussian, the solution can be found in principle via the Kalman–Bucy filter [19]. In the general case, one often estimates x as a statistic (often the mean) of a probability density function (pdf) evolving under the combined effect of (1) and (2). The initial state x^0 being known, all one has to do is evaluate sequentially the pdfs P_{n+1} of the variables x^{n+1} given the equations and the data. In a "particle" filter this is done by following "particles" (replicas of the system) whose empirical distribution at time t^n approximates P_n. One may for example [1; 9; 7; 8; 19] use the pdf P_n and (1) to generate a prior density (in the sense of Bayes) , and then use the data b^{n+1} to generate sampling weights which define a posterior density P_{n+1}. This can be very expensive because in most weighting schemes, most of the weights tend to zero fast and the number of particles needed can grow catastrophically [21; 2]; various strategies have been proposed to ameliorate this problem.

Our remedy is implicit sampling [4; 5]. The number of particles needed in a filter remains moderate if one can find high probability particles; to this end, implicit sampling works by first picking probabilities and then looking for particles that assume them, so that the particles are guided efficiently to the high probability region one at a time, without needing a global guess of the target density. In the present paper we provide an expository account of particle filters, separating clearly the general principles from details of implementation; we provide general solution algorithms for the resulting algebraic equations, in particular for nonconvex cases which we had not considered in our previous publications, as well as a new algorithm for parameter identification based on an implicit filter. We also provide examples, in particular of nonconvex problems.

Implicit filters are a special case of chainless sampling methods [3]; a key connection was made in [23; 24], where it was observed that in the sampling of stochastic differential equations, the marginals needed in Markov field sampling can be read off the equations and need not be estimated numerically.

2. The mathematical framework

The first thing we do is discretize the SDE (1) by a difference scheme, so that the equation becomes a discrete recurrence, and assume temporarily that the time step in the dynamics equals the fixed time δ between observations. For simplicity, in this section we assume the scheme is the Euler scheme

$$x^{n+1} = x^n + \delta f(x^n, n\delta) + V, \tag{3}$$

where V is a Gaussian of mean zero and variance $g^2(x^n, n\delta)\delta$. Higher-order schemes

are discussed in Section 4.

The conditional probability densities $P_n(x)$ at times t^n, determined by the discretized SDE (3) given the observations (2), satisfy the recurrence relation [9, p. 6]

$$P(x^{n+1}) = P(x^n)P(x^{n+1}|x^n)P(b^{n+1}|x^{n+1})/Z, \tag{4}$$

where $P(x^n) = P(x^n|b^1, b^2, \ldots, b^n)$ is the probability density at time $n\delta$ given the data up to time $n\delta$, $P(x^{n+1}|x^n)$ is the probability density of x^{n+1} given x^n as it is determined by the dynamics, $P(b^{n+1}|x^{n+1})$ is the probability of the next observation given the new position, as per the observation equation, and Z is a normalization constant.

We estimate P_{n+1} with the help of M particles, with positions X_i^n at time t^n and X_i^{n+1} at time t^{n+1} ($i = 1, \ldots, M$), which define empirical densities $\hat{P}_n$, $\hat{P}_{n+1}$ that approximate P_n, P_{n+1}. We do this by requiring that, when a particle moves from X_i^n to X_i^{n+1}, the probability of X_i^{n+1} given b^k for $k \le n+1$ be given by

$$P(X_i^{n+1}) = P(X_i^n)P(X_i^{n+1}|X_i^n)P(b^{n+1}|X_i^{n+1})/Z_0, \tag{5}$$

where the hats have been omitted as they will be from now on, $P(X_i^n)$, the probability of X_i^n given b^k for $k \le n$, is assumed given, $P(X_i^{n+1}|X_i^n)$, the probability of X_i^{n+1} given X_i^n, is determined by the discretized SDE (3), $P(b^{n+1}|X_i^{n+1})$, the probability of the observations b^{n+1} given the new positions X_i^{n+1}, is determined by the observation equaiton (2), and Z_0 is an unknown normalization constant. We shall see below that one can set $P(X_i^n) = 1$ without loss of generality.

Equation (5) defines the pdf we now need to sample for each particle. One way to do this is to pick a position X_i^{n+1} according to some prior guess of P_{n+1}, and then use weights to get the resulting pdf to agree with the true P_{n+1} (the "posterior" density); in general many of the new positions will have low probability and therefore small weights. The idea in implicit sampling is to define probabilities first, and then look for particles that assume them; this is done by choosing once and for all a fixed reference random variable, say ξ, with a given pdf, say a Gaussian $\exp(-\xi^T\xi/2)/(2\pi)^{m/2}$, which one knows how to sample, and then making X_i^{n+1} a function of ξ, a different function of each particle and each step, each function designed so that the map $\xi \to X_i^{n+1}$ connects highly probable values of ξ to highly probable values of X_i^{n+1}. To that end, write

$$P(X_i^{n+1}|X_i^n)P(b^{n+1}|X_i^{n+1}) = \exp(-F_i(X)),$$

where on the right-hand side X is a shorthand for X_i^{n+1} and all the other arguments are omitted. This defines a function F_i for each particle i and each time t^n. For each i and n, F_i is an explicitly known function of $X = X_i^{n+1}$. Then solve the equation

$$F_i(X) - \phi_i = \xi^T\xi/2, \tag{6}$$

where ξ is a sample of the fixed reference variable and ϕ_i is an additive factor needed to make the equation solvable. The need for ϕ_i becomes obvious if one considers the case of a linear observation function h in (2), so that the right side of (6) is quadratic but the left is a quadratic plus a constant. It is clear that setting $\phi_i = \min F_i$ will do the job, but it is sometimes convenient to perturb this choice by a small amount (see below). In addition, and most important, with our choice of reference variable ξ, the most likely choice of ξ is in the neighborhood of 0; if the mapping $\xi \to X$ satisfies (6), this likely choice of ξ produces an X near the minimum of F_j, hence a high probability position for the particle. We also require that for each particle, the function $X_i^{n+1} = X = X(\xi)$ defined by (6) be one-to-one so that the correct pdf is sampled, in particular, it must have distinct branches for positive values and negative values of each component of ξ. The solution of (6) is discussed in the next section. From now on we omit the index i in both F and ϕ, but it should not be forgotten that these functions vary from particle to particle and from one time step to the next.

Once the function $X = X(\xi)$ is determined, each value of $X^{n+1} = X$ (the subscript i is omitted) appears with probability $\exp(-\xi^T\xi/2)J^{-1}/(2\pi)^{m/2}$, where J is the Jacobian of the map $X = X(\xi)$, while the product $P(X^{n+1}|X^n)P(b^{n+1}|X^{n+1})$ evaluated at X^{n+1} equals $\exp(-\xi^T\xi/2)\exp(-\phi)$. The sampling weight for the particle is therefore $\exp(-\phi)J(2\pi)^{m/2}$. If the map $\xi \to X$ is smooth near $\xi = 0$, so that ϕ and J do not vary rapidly from particle to particle, and if there is an easy way to compute J (see the next section), then we have an effective way to sample P_{n+1} given P_n. It is important to note that though the functions F and ϕ vary from particle to particle, the probabilities of the various samples are expressed in terms of the fixed reference pdf, so that they can be compared with each other.

The weights can be eliminated by resampling. A standard resampling algorithm goes as follows [9]: let the weight of the i-th particle be W_i, $i = 1, \dots, M$. Define $A = \sum W_i$; for each of M random numbers θ_k, $k = 1, \dots, M$, drawn from the uniform distribution on [0, 1], choose a new $\widehat{X}_k^{n+1} = X_i^{n+1}$ such that $A^{-1}\sum_{j=1}^{i-1} W_j < \theta_k \le A^{-1}\sum_{j=1}^{i} W_j$, and then suppress the hat. This justifies the statement following (5) that one can set $P(X_n) = 1$.

To see what has been gained, compare our construction with the usual "Bayesian" particle filter, where one samples $P(X^{n+1}|X^n)P(b^{n+1}|X^{n+1})$ by first finding a "prior" density $Q(X^{n+1})$ (omitting all arguments other than X^{n+1}), such that the ratio $W = P(X^{n+1})/Q(X^{n+1})$ is close to a constant, and then assigning to the i-th particle the importance weight $W = W_i$ evaluated at the location of the particle. The pdf defined by the set of positions and weights is the density P_{n+1} we are looking for. An important special case is the choice $Q(X^{n+1}) = P(X^{n+1}|X^n)$; the prior is then defined by the equation of motion alone and the posterior is obtained by using the observations to weight the particles. We shall refer to this special case as

"standard importance sampling" or "standard filter". Of course, once the positions and the weights of the particles have been determined, one should resample as above.

The catch in these earlier constructions is that the prior density Q and the desired posterior can become nearly mutually singular, and the number of particles needed may become catastrophically large, especially when the number of variables m is large [2; 21]. To avoid this catch one has to make a good guess for the pdf Q, which may not be easy because Q should approximate the unknown density P_{n+1} one is looking for — this is the basic conundrum of Monte Carlo methods, in which one needs a good estimate to get a good estimate. In contrast, in implicit sampling one does a separate calculation for each sample and there is no need for prior global information. One can of course still identify the pdf defined by the positions of the particles at time t^{n+1} as a "prior" and the pdf defined by both the positions and the weights as a "posterior" density.

Note that one can recover standard importance sampling within our framework by setting $\phi = -\log P(b^{n+1}|X^{n+1})$, but this choice of course violates our rule for choosing ϕ.

Finally, implicit sampling can be viewed as an implicit Monte Carlo scheme for solving the Zakai equation [25], which describes the evolution of the unnormalized conditional distribution for an SDE conditioned by observations. This should be contrasted with the procedure in the popular ensemble Kalman filter [11], where a Gaussian approximation of the pdf defined by the SDE is extracted from a Monte Carlo solution of the corresponding Fokker–Planck equation, a Gaussian approximation is made for the pdf $P(b^{n+1}|x^{n+1})$ [17], and new particle positions are obtained by a Kalman step. Our replacement of the Fokker–Planck equation that corresponds to the SDE alone by a Zakai equation that describes the evolution of the unnormalized conditional distribution does away with the need for the approximate and expensive extraction of Gaussians and consequent Kalman step.

3. Solution of the algebraic equation that defines a new sample

We now explain how to solve (6), $F(X) - \phi = \xi^T \xi/2$, under several sets of assumptions which are met in practice. This is a well-defined, deterministic, algebraic equation for each particle. Note the great latitude that it provides in linking the ξ variables to the X variables: it is a single equation that connects $2m$ variables (the m components of ξ and the m components of X) and can be satisfied by many maps $\xi \to X$ as long as (i) they are one-to-one, (ii) they map the neighborhood of 0 into a set that contains the minimum of F, (iii) they are smooth near $\xi = 0$ so that the weights $\exp(-\phi)J$ not vary unduly from particle to particle in the target area, and (iv) they allow the Jacobian J to be calculated easily. The solution methods

presented here are far from exhaustive; further examples will be presented in the context of specific applications.

Algorithm A (presented in [4; 5]). Assume the function F is convex upwards and h is not too different from a linear function. For each particle, we set up an iteration, with iterates $X^{n+1,j}$, $j = 0, 1, \ldots$, (X^j for brevity), with $X^0 = 0$, that converge to the next position X^{n+1} of that particle. The index i that identifies the particle is omitted again. We write the equations as if the system were one-dimensional; the multidimensional case was presented in detail in [5]. First we sample the reference variable ξ. The iteration is defined when one knows how to find X^{j+1} given X^j.

Expand the observation function h in (2) around X^j:

$$h(X^{j+1}) = h(X^j) + (Dh)^j (X^{j+1} - X^j), \tag{7}$$

where $(Dh)^j$ is the derivative of h evaluated at X^j. The observation function in (2) is now approximated as a linear function of X^{j+1}, and the function F is the sum of two Gaussians in X^{j+1}. Completing a square yields a single Gaussian with a remainder ϕ, i.e., $F(X) = (x - \bar{a})^2/(2\bar{v}) + \phi(X^j)$, where the parameters ϕ, $\bar{a}$, $\bar{v}$ are functions of X^j (this is what we called in [4] a "pseudo-Gaussian"). The next iterate is now $X^{j+1} = \bar{a} + \sqrt{\bar{v}}\xi$. In the multidimensional case, when each component of the function h in (2) depends on more than one variable, finding X^{j+1} as a function of ξ may require the solution of an equation of the form $(X^{j+1})^T A X^{j+1} = \xi^T \xi/2$, where A is positive definite and symmetric. This is, as expected, a single equation for several variables, so that the solution is not unique. We may choose, as in [5], to connect ξ to X^{j+1} by performing a Choleski decomposition, $A = LL^T$, where L is lower triangular, and then solving $\sqrt{2}L^T X^{j+1} = \xi$. A different connection was presented in [4]. If the iteration converges, it converges to the exact solution of (6), with ϕ the limit of the $\phi(X^j)$. Its convergence can be accelerated by Aitken's extrapolation [13]. The Jacobian J can be evaluated either by an implicit differentiation of (6) or numerically, by perturbing ξ in (6) and solving the perturbed equation (which should not require more than a single additional iteration step). It is easy to see that this iteration, when it converges, produces a mapping $\xi \to X$ that is one to one and onto.

An important special case occurs when the observation function h is linear in X and there are observations at every step (see section 5 for the case of sparse observations). It is immaterial then whether the SDE (1) is linear. In this case the iteration converges in one step; the Jacobian J is easy to find; if in addition the function $g(x, t)$ in (1) is independent of x, then J is independent of the particle and need not be evaluated; the additive term ϕ can be written explicitly as a function of the previous position X^n of the particle and of the observation b^{n+1}. We recover an easy implementation of optimal sequential importance sampling [1; 9; 8].

This iteration has been used in [5]. It may fail to converge if the function F is not convex, as happens in particular when the observation function h is highly nonlinear. In the latter case the value of ϕ it produces may also be far from the minimum of F. If h is strongly nonlinear, the next iteration is preferable.

Algorithm B. Assume the function F is U-shaped, i.e., in the scalar case, it is at least piecewise differentiable, F' 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, with $F(X) = \infty$ when $X = \pm\infty$. In the m-dimensional case, assume that F has a single minimum and that 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 (a function may be U-shaped without being convex).

Find z, the minimum of F (this is the minimum of a given real valued function, not a minimum of a possibly multimodal pdf generated by the SDE; finding this minimum is not equivalent to the difficult problem of finding a maximum likelihood estimate of the state of the system). The minimum z can be found by standard minimization algorithms.

Again we are solving the equations by finding iterates X^j that converge to X^{n+1}. In the scalar case, given a sample of the reference variable ξ, find first X^0 such that $X^0 - z$ has the sign of ξ, and then find the next iterates X^j by standard tools (e.g., by Newton iteration), modified so that the X^j are prevented from leaping over z.

In the vector case, if the observation function is diagonal — i.e., each component of the observation is a function of a single component of the solution X — then the scalar algorithm can be used component by component. In more complicated situations one can take advantage of the freedom in connecting ξ to X.

Here is an interesting example of the use of this freedom, which we present in the case of a multidimensional problem where the observation function is linear but need not be diagonal. Set $\phi = \min F$. The function $F(X) - \phi$ can now be written as $(X-a)^T A(X-a)/2$, where a is a known vector, T denotes a transpose as before, and A is a positive definite symmetric matrix. Write further $y = X - a$. Equation (6) becomes

$$y^T A y = |\xi|^2, \tag{8}$$

where $|\xi|$ is the length of the vector ξ. Make the ansatz

$$y = \lambda\eta,$$

where λ is a scalar, $\eta = \xi/|\xi|$ is a random unit vector and ξ is a sample of the reference density. Substitution into (8) yields

$$\lambda^2(\eta^T A \eta) = |\xi|^2. \tag{9}$$

It is easy to see that $E[\eta_i \eta_j] = \delta_{ij}/m$, where $E[\,\cdot\,]$ denotes an expected value, the η_i are the components of η, m is the number of variables, and δ_{ij} is the Kronecker delta, and hence

$$E[\eta^T A \eta] = \text{trace}(A)/m.$$

Replace (9) by

$$\lambda^2 \Lambda = |\xi|^2. \tag{10}$$

where $\Lambda = \text{trace}(A)/m$. This equation has the solution $\lambda = |\xi|/\sqrt{\Lambda}$, and substitution into the ansatz leads to $y_i = \xi_i/\sqrt{\Lambda}$, an easily implementable transformation with Jacobian $J = \Lambda^{-m/2}$. The difference between (9) and (10) can be compensated for by adding to ϕ the term $\lambda^2[(\eta^T A \eta) - \Lambda]$. Note that as $m \to \infty$, $(\eta^T A \eta) \to \Lambda$ provided A satisfies some minor requirements, so that when the number of variables is sufficiently large, the perturbation one has to compensate for becomes negligible. Detailed implementations and generalizations of this construction will be given elsewhere in the context of specific applications.

One can readily devise algorithms also for cases where F is not U-shaped, for example, by dividing F into monotonic pieces and sampling each of these pieces with its predetermined probability. An alternative that is usually easier is to replace the non-U-shaped function F by a suitable U-shaped function F_0 and make up for the bias by adding $F(X) - F_0(X)$ to the ϕ in the weights $\exp(-\phi)J$ so that (6) is still satisfied. One also has to make sure that the small ξ region is still mapped on the high probability region for X; see the examples below.

More generally, even in convex cases, one can often change F in (6) to make the algebraic problem easy without reducing the quality of the samples; examples will be given in the next two sections.

4. Examples

We now present examples that illustrate the algorithms we have just described. For more examples, see [4; 5]. For the sake of clarity, in this section we continue to rely on an Euler discretization of the SDE, as in (3).

We begin with a response to a comment we have often heard: "This is nice, but the construction will fail the moment you are faced with potentials with multiple wells". This is not so: the function F depends on the nature of the noise in the SDE and on the function $h = h(x)$ in the observation (2), but not on the potential. Consider, for example, a one dimensional particle moving in the potential $V(x) = 2.5(x^2 - 0.5)^2$ (see Figure 1) with the force $f(x) = -\nabla V = -10x(x^2 - 1)$ and the resulting SDE $dx = f(x)dt + \sigma dw$, where $\sigma = 0.1$ and w is Brownian motion with unit variance; with this choice of parameters the SDE has an invariant density concentrated in the neighborhoods of $x = \pm\sqrt{1/2}$. We consider linear observations $b^n = x(t^n) + W$, where W is a Gaussian variable with mean zero and variance $s = 0.025$. We

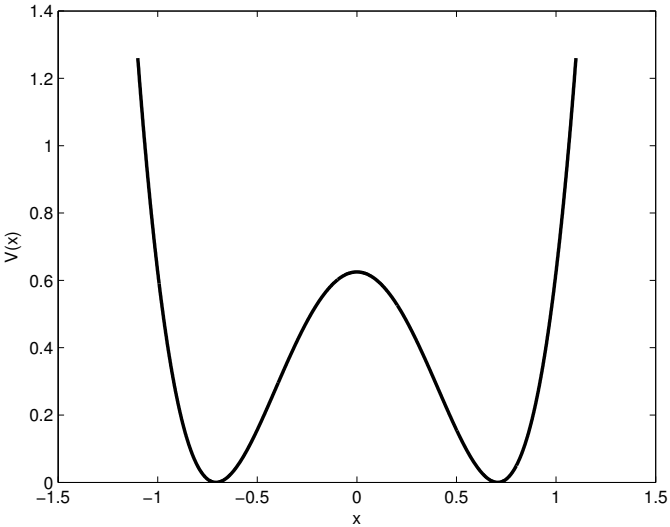


Figure 1. The potential in the first example.

approximate the SDE by an Euler scheme [16] with time step $\delta = 0.01$, and assume observations are available at all the points $n\delta$. The particles all start at $x = 0$. We produce data b^n by running a single particle and adding to its positions errors drawn from the assumed error density in (2), and then attempt to reconstruct this path with our filter. For the i-th particle located at time $n\delta$ at X_i^n the function $F(X)$ is

$$F(X) = \frac{(X - X_i^n)^2}{2\sigma\delta} + \frac{(X - b^{n+1})^2}{2s},$$

which is always convex. A completion of the square yields

$$\min F = \phi = \frac{(X_i^n - b^{n+1})^2}{2(\sigma\delta + s)};$$

the Jacobian J is independent of the particle and need not be evaluated. In Figure 2 we display a particle run used to generate data and its reconstruction by our filter with 50 particles.

This figure is included for completeness but both of these paths are random, their difference varies from realization to realization, and may be large or small by accident. To get a quantitative estimate of the performance of the filter, we repeated this calculation 10^4 times and computed the mean and the variance of the difference Δ between the run that generated the data and its reconstruction at time $t = 1$; see Table 1. This table shows that the filter is unbiased and that the variance of Δ is comparable to the variance of the error in the observations $s = 0.025$. Even with one single particle (and therefore no resampling) the results are still acceptable.

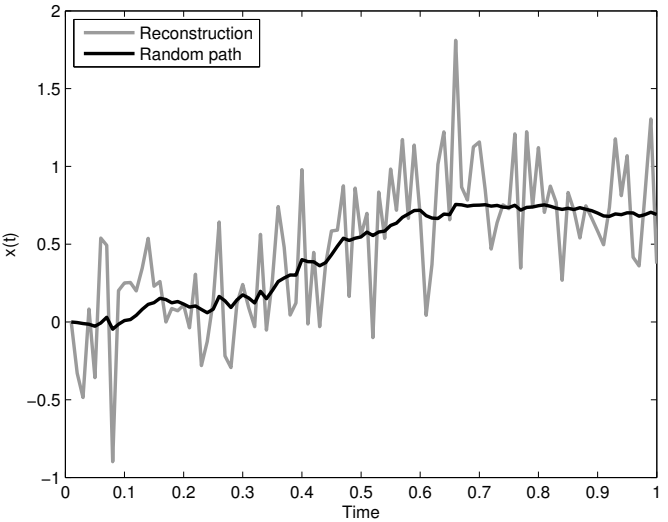


Figure 2. A random path (black) and its reconstruction by our filter (gray).

We now discuss the relation between the posterior we wish to sample and the prior in several special cases, including nonconvex situations. We want to produce samples of the pdf $P(x) = \exp(-F(x))/Z$, where Z is a normalization constant and

$$F(x) = \frac{x^2}{2\sigma} + \frac{(h(x)-b)^2}{2s}. \tag{11}$$

Here $h(x)$ is a given function of x as in (2), and σ, s, b are given parameters. This can be viewed as a first step in time for a filtering problem where all the particles start from the same point so that $\exp(-F(x))/Z = P_1$, or as an analysis of the sampling for one particular particle in a general filtering problem, or as an instance of the more general problem of sampling a given pdf when the important events may be rare. In standard Bayesian sampling one samples the variable with pdf

$$\frac{\exp(-x^2/(2\sigma))}{\sqrt{2\pi\sigma}}$$

M	mean	variance	M	mean	variance
100	−0.0001	0.021	10	0.0001	0.024
50	−0.0001	0.022	5	−0.0001	0.027
20	−0.0001	0.023	1	−0.0001	0.038

Table 1. Mean and variance of the discrepancy between the observed path and the reconstructed path in Example 1 as a function of the number of particles M, with $s = 0.025$.

and then one attaches to the sample at x the weight $\exp(-(h(x)-b)^2/(2s))$; in an implicit sampler one finds a sample x by solving $F(x)-\phi=\xi^2/2$ for a suitable ϕ and ξ and attaching to the sample the weight $\exp(-\phi)J$. For given σ, s, the problem becomes more challenging as $|b|$ increases.

In both the standard and the implicit filters one can view the empirical pdf generated by the unweighted samples as a "prior" and the one generated by the weighted samples as the "posterior". The difficulty with standard filters is that the prior and posterior densities may approach being mutually singular, so it is of interest to estimate the Radon–Nikodým derivative of one of these with respect to the other. If that derivative is a constant, we have achieved perfect importance sampling, as every neighborhood in the sample space is visited with a frequency proportional to its density. We estimate the Radon–Nikodým derivative of the prior with respect to the posterior as follows. In this simple problem one can evaluate the probability of any interval with respect to the posterior we wish to sample by quadratures. We divide the interval $[0, 1]$ into K pieces of equal lengths $1/K$, then find numerically points $Y_1, Y_2, \ldots, Y_{K-1}$, with $Y_K=+\infty$, such that the posterior probability of the interval $[-\infty, Y_k]$ is k/K for $k=1, 2, \ldots, K$. We then find $L=10^5$ samples of the prior and plot of a histogram of the frequencies with which these samples fall into the posterior equal probability intervals (Y_{k-1}, Y_k). The more this histogram departs from being a constant independent of k, the more samples are needed to calculate the statistics of the posterior.

If $h(x)$ is linear, the weights in the implicit filter are all equal and the histogram is constant for all values of b. This remains true for all values of b, i.e., however far the observation b is from what one may expect from the SDE alone. This is not the case with a standard Bayesian filter, where some parts of the sample space that have nonzero probability are visited very rarely.

In Table 2 we list the histogram of frequencies for a linear observation function $h(x)=x$ and $b=2$ in a standard Bayesian filter, with $K=10$. We used 10^4 samples; the fluctuations in the implicit case measure only the accuracy with which

k	standard	implicit	k	standard	implicit
1	0.987	0.099	6	0.003	0.099
2	0.006	0.108	7	0.001	0.101
3	0.002	0.097	8	0.001	0.101
4	0.001	0.099	9	0.000	0.102
5	0.004	0.101	10	0.000	0.093

Table 2. Histogram of the Radon–Nikodým derivative of the prior with respect to the posterior, standard Bayesian filter versus the implicit filter, 10000 particles, $b=2$, $\sigma=s=0.1$, $h(x)=x$.

b	exact	standard	implicit
0	0	−0.05	0.02
0.5	0.25	0.10	0.27
1	0.5	0.18	0.51
1.5	0.75	0.23	0.76
2	1	0.26	1.01

Table 3. Comparison of the estimates of the means, implicit vs. standard filter, 30 particles, together with the exact results, linear case, as explained in the text.

the histogram is computed with this number of samples. As a consequence, estimates obtained with the implicit filter are much more reliable than the ones obtained with the standard Bayesian filter.

In Table 3 we list the estimates of the mean position of the linear problem as a function of b, with 30 particles, $\sigma = s = 0.1$, for the standard Bayesian and the implicit filters, compared with the exact result. The standard deviations are not displayed; they are all near 0.01.

The results in this one-dimensional problem mirror the situation with the example of Bickel et al. [2; 21], designed to display the breakdown of the standard Bayesian filter when the number of dimension is large; what happens there is that one particle hogs almost the whole weight, so that the number of particles needed grows catastrophically; in contrast, the implicit filter assigns equal weights to all the particles in any number of dimensions, so that the number of particles needed is independent of dimension; see also [5].

We now turn to nonlinear and nonconvex examples. Let the observation function h be strongly nonlinear: $h(x) = x^3$. With $\sigma = s = 0.1$, the pdf (11) becomes non-U-shaped for $|b| \geq 0.77$. In Figure 3 we display the function F for $b = 1$ (the solid curve). To use the algorithms above we need a substitute function F_0 that is U-shaped; we also display in Figure 3 (the broken line) the function F_0 we used; the recipe here is to link a point above the local minimum on the left to the absolute minimum on the right by a straight line. It is important to make F_0 and F have the same minimum. Many other constructions are possible (see in particular the next section). As described above, we solve

$$F_0(x) - \phi = \xi^T \xi/2$$

with $\phi = \min F_0$, and once x has been determined, add the difference $F(x) - F_0(x)$ to ϕ in the weight $\exp(-\phi)J$. This construction does not introduce any bias. The function F_0 constructed in this way is U-shaped but need not be convex, so that one

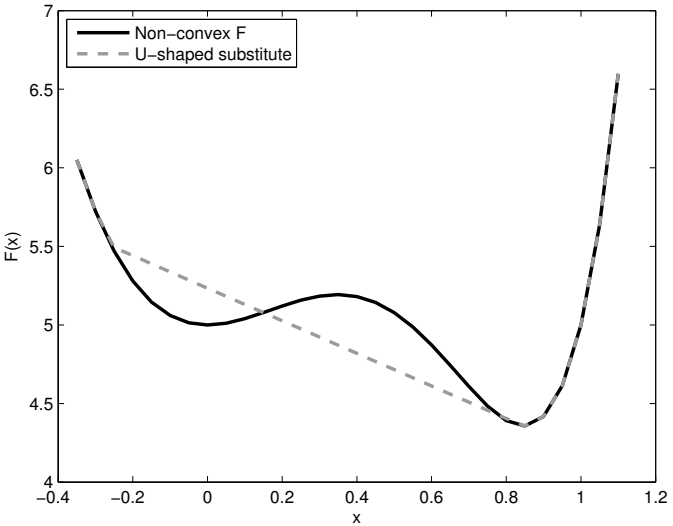


Figure 3. A nonconvex function F (solid line) and a U-shaped substitute (broken line).

needs Algorithm B above. In Table 4 we compare the Radon–Nikodým derivatives of the prior with respect to the posterior for the resulting implicit sampling and for standard Bayesian sampling with $\sigma = s = 0.1$, $b = 1.5$.

The histogram for the implicit filter is no longer perfectly balanced. The asymmetry in the histogram reflects that of F_0 and can be eliminated by biasing ξ, but there is no reason to do so; there is enough importance sampling without this extra step.

k	standard	explicit
1	0.9948	0.0899
2	0.0028	0.0537
3	0.0011	0.0502
4	0.0004	0.0563
5	0.0003	0.0696
6	0.0002	0.1860
7	0.0001	0.1107
8	0.0001	0.1194
9	0.0001	0.1196
10	0	0.1446

Table 4. Radon–Nikodým derivatives of the prior with respect to the posterior, $h(x) = x^3$, $\sigma = s = 0.1$, $b = 1.5$, 10000 samples, F_0 as in the text.

b	exact	standard	implicit
0.	0.	-0.00 ± 0.01	-0.00 ± 0.01
0.5	0.109	0.109 ± 0.01	0.109 ± 0.01
1.0	0.442	0.394 ± 0.04	0.451 ± 0.02
1.5	0.995	0.775 ± 0.09	0.995 ± 0.01
2.0	1.18	0.875 ± 0.05	1.18 ± 0.01
2.5	1.30	0.895 ± 0.02	1.29 ± 0.02

Table 5. Comparison of the estimates of the means, implicit vs. standard filter, 1000 particles, together with the exact result, when $h(x) = x^3$, as explained in the text.

In Table 5 we display the estimates of the means of the density for the two filters with 1000 particles for various values of b, compared with the exact results (the number of particles is relatively large because with $h(x) = x^3$ and our parameter choices the variance of the conditional density is significant, and this number of particles is needed for meaningful comparisons of either algorithm with the exact result).

As mentioned in the previous section, there are alternatives to the replacement of F by F_0; the point is that for each particle the function F is an explicitly known nonrandom function, and this fact can be used in multiple ways.

5. Sparse observations and higher-order difference approximations

We now discuss what happens when the observations are sparse, so that there are data only every $r > 1$ time steps, and how to sample when the difference approximation is more elaborate than the Euler scheme used so far. Along the way, we suggest additional ways to solve the algebraic equations.

Consider again the discrete SDE (3), with observations available only at times $r\delta$, $r > 1$. To simplify the presentation, let $g(x, t) = 1$ and assume the equation is scalar. Write the scheme in the form $x^{n+1} = q(x^n) + \delta V$, where V is a Gaussian with mean zero and variance one. We have data at the points $r\delta$, $r > 1$, where $b^{n+r} = h(x^{n+r}) + \sqrt{s}W$ and W is a Gaussian of mean zero and variance one and s is a constant. The probability of the particle path $(X_i^{n+1}, X_i^{n+1}, \dots, X_i^{n+r})$ (from now on we will suppress the index i) is

$$P(X^{n+1}, \dots, X^{n+r}) = \exp(-F(X^{n+1}, \dots, X^{n+r}))/Z, \tag{12}$$

where Z is a normalization constant and

$$F(X^{n+1}, \dots, X^{n+r}) = \frac{(h(X^{n+r}) - b^{n+r})^2}{2s} + \frac{1}{2\delta}\sum_{i=1}^{r}(X^{n+i} - q(X^{n+i-1}))^2.$$

The task at hand is to solve

$$F(X^{n+1},\dots,X^{n+r}) - \min F(X^{n+1},\dots,X^{n+r}) = \xi^T\xi/2, \tag{13}$$

where ξ is a sample of a r-dimensional Gaussian reference variable. This can be done by the methods presented above, but we use this opportunity to present some variants.

First, we find the minimum of F. If F is convex, this can be easily done by Newton's method (note that the matrices one gets are sparse). If F is not convex, one can try the following device: add to F the quantity αG, where $\alpha > 0$ is a parameter and G is the convex function

$$G = (X^{n+r} - b^{n+r})^2 + \sum_{j=i+1}^{j=i+r} (X^j - X^{j-1})^2.$$

Then minimize $F + \alpha G$ for a suitable sequence $\alpha_n \to 0$. (This device was inspired by the rubber band construction of computational chemistry [12]. More generally, it is useful to note the resemblance of the problem to the study of rare transitions in computational chemistry [18; 10]). A minimization by a Newton's method also yields the Hessian H of F at the minimizer z of F.

Define $F_0 = \phi + (1/2)(X-z)^T H(X-z)$, where $\phi = \min F = F(z)$ and X is the vector $(X^{n+1},\dots,X^{n+r})$. Solve the equation $F_0(X) - \phi = \xi^T\xi/2$ and obtain X. This is a linear problem and Choleski construction works fine and also yields the Jacobian J. Use as weight for the resulting sample X the quantity $\exp(-\phi_0)J$, with $\phi_0 = \phi + F(X) - F_0(x)$ so that (13) is still satisfied and there is no bias. This is still a high-probability sample, because the neighborhood of $\xi = 0$ is still mapped on the neighborhood of the minimum of F.

As an example, consider the SDE $dx = \cos(5x)dt + \sigma dW$, with $\sigma = 0.1$, discretized by Euler's method with time step 0.01; the observations $b^n = x^n + \eta$ are available every 20 steps (a time interval of 0.2) and η is a Gaussian variable of mean zero and variance 10^{-3}. The data are generated by running the equation once and observing its path. We used 4 particles. In Figure 4 we display the run that generated the data path and the reconstruction; the data are used in the reconstruction only when $t = 0.2$ and $t = 0.4$. Observe that between data the discrepancy can be quite significant, as is indeed unavoidable.

This last example should make it plain what one should do when one uses a higher-order discretization of the SDE. For example, suppose one is integrating the SDE $dx = f(x)dt + dW$ using the second-order Klauder–Petersen scheme [15]:

$$x^{n+1,*} = x^n + \delta f(x^n) + \eta_1, \tag{14}$$

$$x^{n+1} = x^n + (\delta/2)\left(f(x^n) + f(x^{n+1,*})\right) + \eta_2, \tag{15}$$

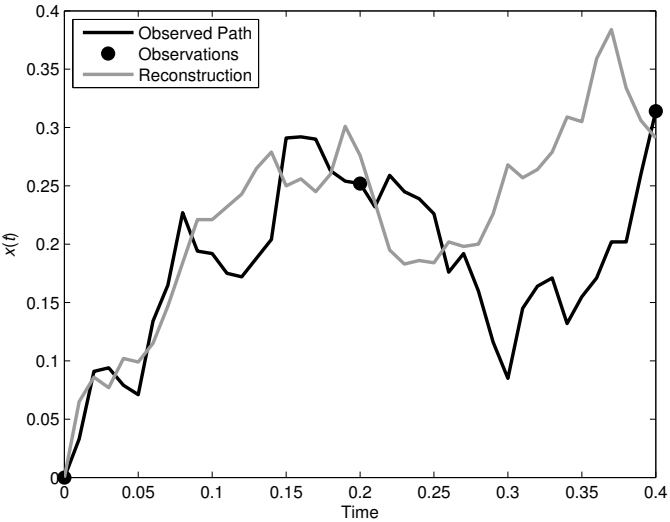


Figure 4. Reconstruction with sparse data.

where η_1, η_2 are Gaussians with mean zero and variance δ. Observations $b^n = h(x^n) + \eta_3$, where η_3 has mean zero and variance s, are assumed available at every step. The probability of the pair $(x^{n+1,*}, x^{n+1})$ is $\exp(-F)$, with

$$F = \frac{(x^{n+1,*} - x^n - \delta f(x^n))^2}{2\delta} + \frac{\left(x^{n+1} - x^n - \frac{\delta}{2}(f(x^n) + f(x^{n+1,*}))\right)^2}{2\delta} + \frac{\left(h(x^{n+1}) - b\right)^2}{2s}.$$

All one has to do then is solve $F - \min F = \xi^T \xi/2$ for a sample ξ of a two-dimensional Gaussian reference variable, along the lines suggested above.

6. Parameter identification

One important application of particle filters is to parameter identification, where the SDE contains an unknown parameter and the data are used to find this parameter's value. One of the standard ways of doing this [9; 14] is system augmentation: one adds to the SDE the equation $d\sigma = 0$ for the unknown parameter σ, one offers σ a gamut of possible values, and one relies on the resampling process that eliminates the values that do not fit the data. With the implicit filter this procedure fails, because the particles are not eliminated fast enough. One alternative is finding the unknown parameter σ by stochastic approximation. Specifically, find a statistic T of the output of the filter which is a function of σ, such that the expected value $E[T]$ vanishes when σ has the right value σ^*, and then solve the equation $E[T] = E[T(\sigma)] = 0$

by the Robbins–Monro iteration [20]:

$$\sigma_{n+1} = \sigma_n - \alpha_n T(\sigma_n), \tag{16}$$

which converges when the coefficients α_n are such that $\sum \alpha_n \to \infty$ while $\sum \alpha_n^2$ remains bounded (for example, $\alpha_n = 1/n^q$ with $0.5 < q \leq 1$). Related ideas can be found, for example, in [6].

As a concrete example, consider the SDE $dx = dW$, where W is Brownian motion with variance σ, discretized with time steps δ, with observations $b^n = x^n + \eta$, where η is a Gaussian with mean zero and variance s. Data are generated by running the SDE once with the true value σ^* of σ, adding the appropriate noise, and registering the result at time $n\delta$ as b^n for $n = 1, 2, \ldots, N$. For the functional T we chose

$$T(\sigma) = C \frac{\sum(\Delta_i \Delta_{i-1})}{\sqrt{\left(\sum \Delta_i^2\right)\left(\sum \Delta_{i-1}^2\right)}}, \tag{17}$$

where the summations are over i between 2 and N, Δ_i is the estimate of the increment of x in the i-th step and C is a scaling constant. Clearly if the σ used in the filtering equals σ^* then by construction the successive values of Δ_i are independent and $E[T] = 0$. We picked the parameters $N = 100$, $\sigma = 10^{-2}$, $s = 10^{-4}$, $\delta = 0.01$ (so that the increment of W in one step has variance 10^{-4}).

Our algorithm is as follows: We make a guess σ_1, run the filter for N steps, evaluate T, and make a new guess for σ using (16) and $\alpha_1 = 1$, rerun the filter, etc., with the α_n, the coefficient in (16) at the n-th step, equal to $1/n$. The scaling factor in (17) was found by trial and error: if it is too large the iteration becomes unstable, if it is too small the convergence is slow; we settled on $C = 4$.

This algorithm requires that the filter be run without resampling, because resampling introduces correlations between successive values of the Δ_i and bias the values of T. In a long run, in particular in a strongly nonlinear setting, one may need resampling for the filter to stay on track, and this can be done by segmentation: divide the run of the filter into segments of some moderate length L, perform the summations in the definition of T over that segment, then go back and run that segment with resampling, then proceed to the next segment, etc.

The first question is, how well is it possible in principle to reconstruct an unknown value of σ from N observations; this issue was already discussed in [4]. Given 100 samples of a Gaussian variable of mean 0 and variance σ, the variance reconstructed from the observations is a random variable of mean σ and variance $0.16 \cdot \sigma$; 100 observations do not contain enough information to reconstruct σ perfectly. A good way to estimate the best result that can be achieved is to run the algorithm with the guess σ_1 equal to the exact value σ^* with which the data were generated. When

Iteration	0	1	2	3	4	5	6	7	8	9
new estimate σ/σ^*	10	0.819	0.943	1.02	1.05	1.08	1.10	1.13	1.15	1.16

Table 6. Convergence of the parameter identification algorithm.

this was done, the estimate of σ was $1.27\sigma^*$. This result indicates the order of magnitude of the accuracy that can be achieved.

In Table 6 we display the result of our algorithm, run with 50 particles and starting value $\sigma_1 = 10\sigma^*$. Each iteration requires running the filter once.

7. Conclusions

We have presented the implicit filter for data assimilation, together with several algorithms for the solution of its algebraic equations, including cases with nonconvex functions F, as well as an algorithm for parameter identification. The key idea in implicit sampling is to solve an algebraic equation of the form

$$F(X) - \phi = \xi^T \xi/2$$

for every particle, where the function F is explicitly known, X is the new position of the particle, ϕ is an additive factor, and ξ is a sample of a fixed reference pdf; F varies from particle to particle and step to step. This construction makes it possible to guide the particles to the high-probability area one by one under a wide variety of circumstances. It is important to note that the equation that links ξ to X is underdetermined and its solution can be adapted for each particular problem. The effectiveness of implicit sampling depends on one's ability to design maps $\xi \to X$ that satisfy the criteria above and are computationally efficient. The design of such maps is problem dependent and we will present examples in the context of specific applications.

Acknowledgements

We would like to thank Prof. Jonathan Weare for asking penetrating questions and for making very useful suggestions, Prof. Robert Miller for his advice and encouragement, and Mr. G. Zehavi for performing some of the preliminary computations.

References

[1] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online nonlinear/nongaussian Bayesian tracking*, IEEE Trans. Sig. Proc. **50** (2002), 174–188.

[2] P. Bickel, B. Li, and T. Bengtsson, *Sharp failure rates for the bootstrap particle filter in high dimensions*, Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh (B. Clarke and S. Ghosal, eds.), IMA Collections, no. 3, Inst. Math. Statist., Beachwood, OH, 2008, pp. 318–329. MR 2010c:93107

[3] A. J. Chorin, *Monte Carlo without chains*, Commun. Appl. Math. Comput. Sci. **3** (2008), 77–93. MR 2425547 Zbl 1165.65302

[4] A. J. Chorin and X. Tu, *Implicit sampling for particle filters*, Proc. Nat. Acad. Sc. USA **106** (2009), 17249–17254.

[5] ———, *An iterative implementation of the implicit nonlinear filter*, 2010, submitted to *Math. Mod. Num. Anal.*

[6] D. P. Dee, *On-line estimation of error covariance parameters for atmospheric data assimilation*, Mon. Wea. Rev. **123** (1995), 1128–1145.

[7] P. Del Moral, *Feynman–Kac formulae: Genealogical and interacting particle systems with applications*, Springer, New York, 2004. MR 2005f:60003 Zbl 1130.60003

[8] A. Doucet, S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Stat. Comp. **10** (2000), 197–208.

[9] A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential Monte Carlo methods in practice*, Springer, New York, 2001. MR 2003h:65007

[10] W. E, *Principles of multiscale modeling*, Cambridge University Press, New York, To appear.

[11] G. Evensen, *Data assimilation: the Ensemble Kalman Filter*, 2nd ed., Springer, Berlin, 2009. MR 2555209 Zbl 1157.86001

[12] R. Gillian and K. Wilson, *Shadowing, real events, and rubber bands. a variational Verlet algorithm for molecular dynamics*, J. Chem. Phys. **97** (1992), 1757–1772.

[13] E. Isaacson and H. B. Keller, *Analysis of numerical methods*, John Wiley & Sons, New York, 1966. MR 34 #924 Zbl 0168.13101

[14] G. Kitagawa, *A self-organizing state-space model*, J. Am. Stat. Ass. **93** (1998), 1203–1215.

[15] J. R. Klauder and W. P. Petersen, *Numerical integration of multiplicative-noise stochastic differential equations*, SIAM J. Numer. Anal. **22** (1985), no. 6, 1153–1166. MR 87a:34064 Zbl 0583.65098

[16] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, Appl. Math., no. 23, Springer, Berlin, 1992. MR 94b:60069 Zbl 0752.60043

[17] I. N. M. Jardak and M. Zupanski, *Comparison of ensemble data assimilation for the shallow water equations model in the presence of nonlinear observation operators*, J. Geophys. Res. (2010), in press.

[18] P. Metzner, C. Schuette, and E. Vanden-Eijnden, *Illustration of transition path theory on a collection of simple examples*, J. Chem. Phys. **125** (2006), 084110.

[19] R. Miller, E. Carter, and S. Blue, *Data assimilation into nonlinear stochastic systems*, Tellus **51A** (1999), 167–194.

[20] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics **22** (1951), 400–407. MR 13,144j Zbl 0054.05901

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

[22] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numer. **19** (2010), 451–559. MR 2652785

[23] J. Weare, *Efficient Monte Carlo sampling by parallel marginalization*, Proc. Nat. Acad. Sc. USA **104** (2007), 12657–12662.

[24] J. Weare, *Particle filtering with path sampling and an application to a bimodal ocean current model*, J. Comput. Phys. **228** (2009), no. 12, 4312–4331. MR 2010g:86010 Zbl 1165.76045

[25] M. Zakai, *On the optimal filtering of diffusion processes*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete **11** (1969), 230–243. MR 39 #3883 Zbl 0164.19201

Received May 24, 2010.

ALEXANDRE CHORIN: chorin@math.berkeley.edu
Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, United States
http://math.berkeley.edu/∼chorin

MATTHIAS MORZFELD: mmo@berkeley.edu
Department of Mechanical Engineering, University of California, Berkeley, CA 94720, United States

XUEMIN TU: xuemin@math.berkeley.edu
Department of Mathematics, University of Kansas, 1460 Jayhawk Boulevard, Lawrence, KS 66045, United States
http://math.ku.edu/∼xtu

Communications in Applied Mathematics and Computational Science

pjm.math.berkeley.edu/camcos

See inside back cover or pjm.math.berkeley.edu/camcos for submission instructions.

The subscription price for 2010 is US $70/year for the electronic version, and $100/year for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840, USA.

Communications in Applied Mathematics and Computational Science, at Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW™ from Mathematical Sciences Publishers.

PUBLISHED BY
mathematical sciences publishers
http://www.mathscipub.org
A NON-PROFIT CORPORATION
Typeset in LaTeX

Communications in Applied Mathematics and Computational Science

vol. 5 no. 2 2010