Chin. Ann. Math.
34B(1), 2013, 89–98
DOI: 10.1007/s11401-012-0757-5

Chinese Annals of Mathematics, Series B

Implicit Sampling, with Application to Data Assimilation*

Alexandre J. CHORIN[1] Matthias MORZFELD[2] Xuemin TU[3]

(*In honor of the scientific heritage of Jacques-Louis Lions*)

Abstract There are many computational tasks, in which it is necessary to sample a given probability density function (or pdf for short), i.e., to use a computer to construct a sequence of independent random vectors x_i $(i = 1, 2, \cdots)$, whose histogram converges to the given pdf. This can be difficult because the sample space can be huge, and more importantly, because the portion of the space, where the density is significant, can be very small, so that one may miss it by an ill-designed sampling scheme. Indeed, Markov-chain Monte Carlo, the most widely used sampling scheme, can be thought of as a search algorithm, where one starts at an arbitrary point and one advances step-by-step towards the high probability region of the space. This can be expensive, in particular because one is typically interested in independent samples, while the chain has a memory. The authors present an alternative, in which samples are found by solving an algebraic equation with a random right-hand side rather than by following a chain; each sample is independent of the previous samples. The construction in the context of numerical integration is explained, and then it is applied to data assimilation.

Keywords Importance sampling, Bayesian estimation, Particle filter, Implicit filter, Data assimilation
2000 MR Subject Classification 11K45, 34K60, 62M20, 65C50, 93E11

1 Implicit Sampling

Suppose that one wants to evaluate the integral

$$I = \int g(x)f(x)\mathrm{d}x,$$

where x is a vector variable, and $f(x)$ is a probability density function (or pdf for short). If the dimension of x is large, it is natural for Monte Carlo to write $I = E[g(x)]$, where $E[\cdot]$ denotes an expected value and x is a random variable whose pdf is $f(x)$, $x \sim f(x)$. The integral can

Manuscript received May 10, 2012.
[1]Department of Mathematics, University of California, Berkeley, CA, 94720, USA; Lawrence Berkeley National Laboratory, CA, 94720, USA. E-mail: chorin@math.berkeley.edu
[2]Lawrence Berkeley National Laboratory, CA, 94720, USA. E-mail: mmo@math.lbl.gov
[3]Department of Mathematics, University of Kansas, Lawrence, KS, 66045, USA. E-mail: xtu@math.ku.edu
*Project supported by the Director, Office of Science, Computational and Technology Research, U. S. Department of Energy (No. DE-AC02-05CH11231) and the National Science Foundation (Nos. DMS-0705910, OCE-0934298).

then be approximated through the law of large numbers,

$$I \approx I_n = \frac{1}{n}\sum_{j=1}^{n} g(X_j),$$

where the X_i are n independent samples of the pdf f, and the error is proportional to $n^{-\frac{1}{2}}$ (see [1–2]).

To perform this calculation, one has to find samples X_j of a given pdf f, which is often difficult. One way to proceed is to find an "importance" density f_0, whose support contains the support of f, and which is easier to sample. Write

$$I = \int g(x)\frac{f(x)}{f_0(x)} f_0(x)\mathrm{d}x = E[g(x)w(x)],$$

where

$$w(x) = \frac{f(x)}{f_0(x)}$$

is a "sampling weight" and $x \sim f_0(x)$. We can approximate this integral through the law of large numbers as above, so that

$$I_n = \frac{1}{n}\sum_{j=1}^{n} g(X_j)w(X_j)$$

converges almost surely to I as $n \to \infty$. One requirement for this to be a practical computing scheme is that the ratio $\frac{f}{f_0}$ be close to a constant, and in particular, that f_0 be large where f is large; otherwise, one wastes one's efforts on samples that contribute little to the result. However, one may not know in advance where f is large — indeed, in the application to data assimilation below, the whole purpose of the computation is to identify the set where f is large.

We now propose a construction that makes it possible to find a suitable importance density under quite general conditions. Write

$$F(x) = -\log f(x),$$

and suppose for the moment that F is convex. Pick a reference variable ξ, such that (i) ξ is easy to sample, (ii) its pdf $g(\xi)$ has a maximum at $\xi = 0$, (iii) the logarithm of g is convex, (iv) it is possible to write the variable with pdf f as a function of ξ. It is often convenient to pick ξ as a unit Gaussian variable, $\xi \sim \mathcal{N}(0, I)$, where I is the identity, and $\mathcal{N}(\mu, \Sigma)$ denotes a Gaussian with mean μ and covariance matrix Σ, and we will do so here. This choice does not imply any Gaussianity assumption for the pdf f we wish to sample.

Then proceed as follows: find

$$\phi = \min F,$$

the minimum of F, and pick a sequence of independent samples $\xi \sim \mathcal{N}(0, I)$. For each one, solve the equation

$$F(X) - \phi = \frac{1}{2}\xi^{\mathrm{T}}\xi, \tag{1.1}$$

i.e., equate the logarithm of f, the pdf to be sampled, to the logarithm of the pdf of the reference variable, after subtracting ϕ, the minimum of F. Subtracting ϕ ensures that solutions exist.

Pick the solutions so that the map $\xi \to x$ is one-to-one and onto. The resulting samples X are independent, because the samples ξ of the reference density are independent. This is in contrast to Markov-chain Monte Carlo schemes, where the successive samples are dependent. Moreover, under the assumptions on ξ, most of the samples of ξ are close to the origin; the corresponding samples X are near the minimizer of F, and therefore near the mode of f. The minimization of F guides the samples of x to where the probability is high.

It is important to note that this construction can be carried out even if the pdf of x is not known explicitly, as long as one can evaluate f for each value of its argument up to a multiplicative constant. The normalization of f need not be known because a multiplicative factor in f becomes an additive factor in $F = -\log f$, and cancels out when the minimum ϕ is subtracted.

To calculate the sampling weight, note that, on one hand, equation (1.1) yields $f(x) = \mathrm{e}^{-\phi} g(\xi)$, where g is the Gaussian $\mathcal{N}(0, I)$. On the other hand, by the change of the variable theorem for integrals, the pdf of x is $\frac{g(\xi)}{J}$, where J is the Jacobian of the map $\xi \to x$. The sampling weight is therefore

$$w \propto \mathrm{e}^{-\phi} J.$$

The assumption that F is convex is too strong. Nothing changes if F is merely U-shaped, that is, a function f of a single scalar variable is U-shaped if it has a single minimum ϕ, has no local maxima or inflection points, and tends to ∞ as $|x| \to \infty$. A function of many variables is U-shaped if the intersection of its graph with every vertical plane through its minimum is U-shaped. If F is not U-shaped, the construction above can still be carried out. Often one can write F as a union of U-shaped functions with disjoint supports, and then a compound reference density directs the samples to the various pieces in turn. One can also approximate F by an approximation of its convex hull. For example, one can expand F around its minimizer $m = \operatorname{argmin} F$ (i.e., $F(m) = \phi$),

$$F = \phi + \frac{1}{2}(x - m)^{\mathrm{T}} H (x - m) + \cdots,$$

where a superscript T denotes a transpose, and H is the Hessian of F which may be left over from the minimization that produced ϕ. One defines

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

and replaces F by F_0 in equation (1.1), so that it becomes

$$(x - m)^{\mathrm{T}} H (x - m) = \xi^{\mathrm{T}} \xi,$$

where the left-hand side is now convex. This still maps the neighborhood of the maximum of g onto the neighborhood of the maximum of f. The sampling weight becomes $w \propto \mathrm{e}^{-\phi_0} J$, where $\phi_0 = F(x) - F_0(x)$.

There remains the task of solving equation (1.1) and evaluating the Jacobian J. How onerous this task is depends on the problem. Observe that equation (1.1) is a single equation while the vector x has many components, so that there are many solutions, but only one is needed. One may, for example, look for a solution in a random direction, reducing the problem of solving equation (1.1) to a scalar problem and greatly simplifying the evaluation of J. This is a "random map" implementation of implicit sampling (for details, see [3–4]).

One may worry that the minimization that produces ϕ may be expensive. However, any successful sampling in a multi-dimensional problem requires a search for high probability areas and, therefore, includes an unacknowledged maximization of a pdf. One may as well do this maximization consciously and bring to use the tools that make it efficient (see also the comparison with variational methods below).

2 Filtering and Data Assimilation

There are many problems in science and engineering, where one wants to identify the state of a system from an uncertain model supplemented by a stream of noisy and incomplete data. An example of this situation is shown in Figure 1.

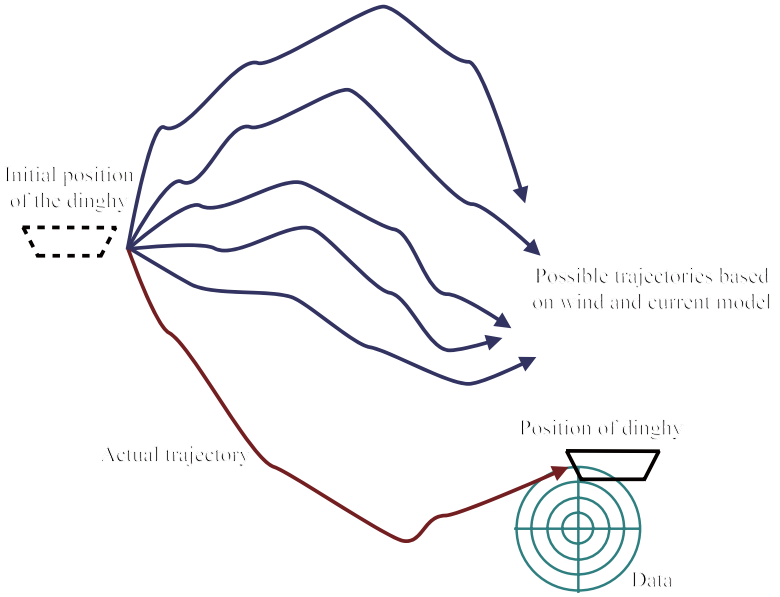

Figure 1 A dinghy in the Pacific ocean: the floating passengers can be located by combining the information from an uncertain model of the currents and winds with the information from a ham radio operator.

Imagine that a ship sank in the Pacific ocean. Its passengers are floating in a dinghy, and you are the coast guard and want to send a navy ship to the rescue. A model of the currents and winds in the ocean makes it possible to draw possible trajectories, but these are uncertain. A ham radio operator spoke to someone in the dinghy several times, but could not locate it without fault. These are the data. The most likely position of the dinghy is somewhere between the trajectories and the observations. Note that the location of the highest probability area is the unknown.

In mathematical terms, the model is often a Markov state space model (often a discretization of a stochastic differential equation (or SDE for short) (see [5])) and describes the state sequence $\{x^n; n \in N\}$, where x^n is a real, m-dimensional vector. To simplify notations, we assume here that the noise is additive, so that the model equations are

$$x^n = f^n(x^{n-1}) + v^{n-1}, \tag{2.1}$$

where f^n is an m-dimensional vector function, and $\{v^{n-1}, n \in N\}$ is a sequence of independent identical distributed (or i.i.d. for short) m-dimensional random vectors which, in many applications, are Gaussian vectors with independent components. One can think of the x^n as values of a process $x(t)$ evaluated at times $n\delta$, where δ is a fixed time increment. The probability density function of the initial state x^0 is assumed to be known.

The model is supplemented by an observation (or measurement) equation, which relates observations $\{b^n; n \in N\}$, where b^n is a real, k-dimensional vector and $k \leq m$, to the states x^n. We assume here that the observation equation is

$$b^n = h^n(x^n) + z^n, \tag{2.2}$$

where h^n is a k-dimensional, possibly nonlinear, vector function, and $\{z^n, n \in N\}$ is a k-dimensional i.i.d. process, independent of v^n. The model and the observation equations together constitute a hidden Markov state space model. To streamline notation, we denote the state and observation sequences up to time n by

$$x^{0:n} = \{x^0, \cdots, x^n\} \quad \text{and} \quad b^{1:n} = \{b^1, \cdots, b^n\},$$

respectively.

The goal is to estimate the sequence $x^{0:n}$, based on (2.1) and (2.2). This is known as "filtering" or "data assimilation". We compute the estimate by sequential Monte Carlo, i.e., by sampling sequentially from the conditional pdf $p(x^{0:n} \,|\, b^{1:n})$ (called the target pdf), and using these samples to approximate the conditional mean (the minimum mean square error estimator (see [2])) by the weighted sample mean. We do this by following "particles" (replicas of the system) whose empirical distribution weakly approximates the target density. For simplicity of presentation, we assume that the model equation (2.1) is synchronized with the observations (2.2), i.e., observations b^n are available at every model step (see [3] for an extension to the case where observations are sparse in time). Using Bayes' rule and the Markov property of the model, we obtain the recursion

$$p(x^{0:n+1} \,|\, b^{1:n+1}) = \frac{p(x^{0:n} \,|\, b^{1:n}) p(x^{n+1} \,|\, x^n) p(b^{n+1} \,|\, x^{n+1})}{p(b^{n+1} \,|\, b^{1:n})}. \tag{2.3}$$

At the current time $t = n + 1$, the first term in the numerator of the right-hand side of (2.3) is known from the previous steps. The denominator is common to all particles, and thus drops out in the importance sampling scheme (where the weights are normalized, so that their sum equals 1). All we have to do is sampling the right-hand side of this expression at every step and for every particle. We do that by implicit sampling, which is indifferent to all the factors on the right-hand side other than $p(x^{n+1} \,|\, x^n) p(b^{n+1} \,|\, x^{n+1})$ (see also [3–4, 6–7]). The factor $p(x^{n+1} \,|\, x^n)$ is determined by the model (2.1), while the factor $p(b^{n+1} \,|\, x^{n+1})$ represents the effect of the observation (2.2). We supplement the sampling by a resampling after each step which equalizes the weights, and gets rid of the factor $p(x^{0:n} \mid b^{1:n})$ and many of the particles with small weights (see [1, 8] for efficient resampling algorithms).

We claim that the use of implicit sampling in data assimilation makes it possible to improve on what other algorithms can do in this problem. We therefore compare the implicit sampling algorithm with other methods of data assimilation in common use.

3 Comparisons with Other Data Assimilation Algorithms

3.1 The standard Bayesian filter

Suppose that the observations are highly consistent with the SDE — for example, in Figure 1, the observations may be somewhere in the middle of the pencil of solutions to the model. There is really no need to explicitly look for the maximum of the pdf, because the model (2.1) already generates samples that are in the high probability region. Therefore, one can set $\phi = \log p(b^{n+1} \,|\, x^{n+1})$ in equation (1.1), and then solving (1.1) is simply sampling a new location determined by the SDE, to which one subsequently assigns a weight determined by the proximity of the sample X to the observation. This sampling scheme is often called the sequential importance sampling with a resampling (or SIR for short) filter. The SIR filter is widely used, and less expensive than what we propose, but may fail if the data are not close to what the model alone would predict. As the dimension of the vector x increases, the neighborhood of the observations and the pencil of solutions to the SDE occupy an ever decreasing fraction of the available space, so that with SIR, guaranteeing that at least a few samples hit the high probability area requires more and more samples (see [9–10]). In contrast, with implicit sampling, the observations affect not only the weights of the samples but also their locations. For more on the SIR, see [1, 8, 11–14].

3.2 Optimal filters

There is literature on "optimal" particle filters, defined as particle filters in which the variance of the weights of each particular particle (not the variance of all the weights) is zero (see [8, 12, 15]). In general, a filter that is "optimal" in this sense requires a precise knowledge of the normalization of the pdf to be sampled, which is not usually available (see the formulas for the pdf to be sampled, remembering that $\int f \mathrm{d}x = 1$, $\int g \mathrm{d}x = 1$, do not imply that $\int fg \mathrm{d}x = 1$.)

To see why in general the optimal filter can not be implemented without knowing the normalization constants exactly, consider first the problem of sampling a given pdf f, and carry out the following construction (in one dimension for simplicity): let $g(\xi)$ be the pdf of a reference variable ξ. Define $F = -\log f$ as before and find the region of high probability through minimization of F, i.e., compute $m = \operatorname{argmin} F$. To find a sample X, solve the differential equation $f \mathrm{d}x = g \mathrm{d}s$, or

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \frac{g}{f}$$

with the initial condition $x(0) = m$, for $s \in (0, \xi]$. This defines a map $\xi \to x(\xi)$ with $f(x) = g(\xi)J(\xi)$, where $J = \left|\frac{\mathrm{d}s}{\mathrm{d}x}\right|$. One can check that the weight is independent of the sample. This sampling scheme fails unless one knows the normalization constant with perfect accuracy, because if one multiplies f in the differential equation by a constant, the resulting samples are not distributed correctly.

In the data assimilation problem one has to sample a different pdf for each particle, so that the application of this sampling scheme yields an "optimal filter" with a zero-variance weight for each particle, provided that one can calculate the normalization constants exactly, which can be done at an acceptable cost only in special cases. In those special cases, the resulting filter coincides with our implicit filter. The implicit filter avoids the problem of unknown

normalization constants by taking logs, converting a harmful unknown multiplicative constant in the pdf into a harmless additive constant.

3.3 The Kalman filter

If the observation function h is linear, the model (2.1) is linear, the initial data are either constant or Gaussian, and the observation noise z^n in (2.2) is Gaussian, then the pdf we are sampling is Gaussian and is entirely determined by its mean and covariance. It is easy to see that in this case a single particle suffices in the implicit filter, and that one gets the best results by setting $\xi = 0$ in the formulas above. The resulting filter is the Kalman filter (see [16–17]).

3.4 The ensemble Kalman filter

The ensemble Kalman filter (see [18]) estimates a pdf for the SDE by a Monte Carlo solution to a Fokker-Planck equation, extracts from this solution a Gaussian approximation, and then takes the data into account by an (approximate) Kalman filter step. The implicit filter on the other hand can be viewed as a Monte Carlo solution to the Zakai equation (see [19]) for the conditional probability $p(x^{0:n} \mid b^{1:n})$, doing away with the need for an expensive and approximate Kalman step.

3.5 Variational data assimilation

There is significant literature on variational data assimilation methods (see [20–25]), where one makes an estimate by maximizing some objective functions of the estimate. Clearly the computation of $\phi = \min F$ above resembles a variational estimate. One can view implicit sampling as a sampling scheme added to a variational estimate. The added cost is small, while the advantages are a better estimate (a least square estimate rather than a maximum likelihood estimate, which is particularly important when the pdf's are not symmetric), and the addition of error estimates, which come naturally with a particle filter but are hard to obtain with a variational estimate. For a thorough discussion, see [26].

4 An Example

As an example, we present a data assimilation calculation for the stochastic Kuramoto-Sivashinksy (or SKS for short) equation presented earlier in [3],

$$u_t + uu_x + u_{xx} + \nu u_{xxxx} = gW(x,t),$$

where $\nu > 0$ is the viscosity, g is a scalar, and $W(x,t)$ is a space-time white noise process. The SKS equation is a chaotic stochastic partial differential equation that has been used to model laminar flames and reaction-diffusion systems (see [27–28]), and recently, has also been used as a large dimensional test problem for data assimilation algorithms (see [29–30]).

We consider the m-dimensional Itô-Galerkin approximation of the SKS equation

$$\mathrm{d}U = (\mathcal{L}(U) + \mathcal{N}(U))\mathrm{d}t + g\mathrm{d}W_t^m,$$

where U is a finite dimensional column vector whose components are the Fourier coefficients of the solution, and W_t^m is a truncated cylindrical Brownian motion (see [31]), obtained from the projection of the noise process $W(x,t)$ onto the Fourier modes. Assuming that the initial

conditions $u(x,0)$ are odd with $\widetilde{U}_0(0)=0$ and that g is imaginary, all Fourier coefficients $U_k(t)$ are imaginary for all $t\geq 0$. Writing $U_k=\mathrm{i}\widehat{\mathrm{U}}_{\mathrm{k}}$ and subsequently dropping the hat gives

$$\mathcal{L}(U)=\mathrm{diag}(\omega_k^2-\nu\omega_k^4)U,$$
$$\{\mathcal{N}(U)\}_k=-\frac{\omega_k}{2}\sum_{k'=-m}^{m}U_{k'}U_{k-k'},$$

where $\omega_k=\frac{2\pi k}{L}$ $(k=1,\cdots,m)$, and $\{\mathcal{N}(U)\}_k$ denotes the k-th element of the vector $\mathcal{N}(U)$. We choose a period $L=16\pi$ and a viscosity $\nu=0.251$, to obtain SKS equations with 31 linearly unstable modes. This set-up is similar to the SKS equation considered in [30]. With these parameter values there is no steady state as in [29]. We choose zero initial conditions $U(0)=0$, so that the solution evolves solely due to the effects of the noise. To approximate the SKS equation, we keep $m=512$ of the Fourier coefficients and use the exponential Euler scheme (see [32]), with the time step $\delta=2^{-12}$ for time discretization (see [3] for details).

We are solving the SKS equations in Fourier variables, but we choose to observe in a physical space (as may be physically reasonable). Specifically, we observe the solution $u(x,t)$ at $\frac{m}{2}$ equidistant locations and at every model step through the nonlinear observation operator $h(x)=x+x^3$. The minimization of F_j was done by using Newton's method (see [33–34]), initialized by a model run without noise. To obtain samples, we solve the algebraic equation (1.1), which is easy when the functions F_j are nearly diagonal, i.e., when the linearizations around a current state are nearly diagonal matrices. This requires in particular that the variables that are observed coincide with the variables that are evolved by the dynamics. Observing in the physical space while computing in the Fourier space creates the opposite situation, in which each observation is related to the variables one computes by a dense matrix. This problem was overcome by using the random map algorithm, presented in [3], for solving (1.1).

To test the resulting filter, we generated data by running the model, and then compared the results obtained by the filter with these data. This procedure is called a "twin experiment" and we define, for each twin experiment, the error at time t^n as

$$e^n=\|U^n_{\mathrm{ref}}-U^n_F\|,$$

where the norm is the Euclidean norm, U^n_{ref} denotes the set of Fourier coefficients of the reference run, and U^n_F denotes the reconstruction by the filter, both at the fixed time t^n. The error statistics of 500 twin experiments are shown in Figure 2.

We observe from Figure 2 that the implicit particle filter produces accurate state estimates (small errors and small error variances) with a small number of particles. The SIR filter on the other hand requires thousands of particles to achieve a similar accuracy, and therefore, is impractical for filtering the SKS equation.

5 Conclusions

We have presented an importance sampling procedure, in which the importance density is defined implicitly through a mapping guided by a minimization rather than given by an explicit formula. This makes it possible to sample effectively a variety of pdfs that are otherwise difficult to work with. In particular, in the data assimilation problem, implicit sampling makes it possible to incorporate the information in the data into the sampling process, so that the target density

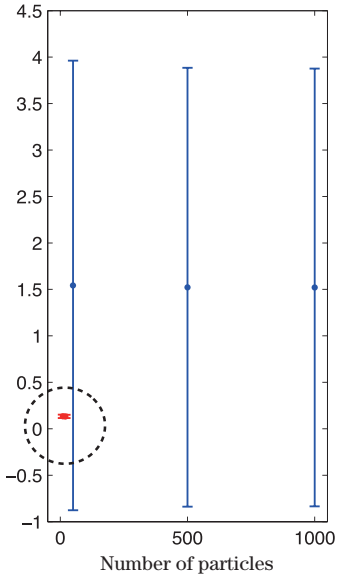

Figure 2 Filtering results for the SKS equation: the error statistics are shown as a function of the number of particles for the SIR filter (blue) and the implicit particle filter (red). The error bars represent the mean of the errors and mean of the standard deviations of the errors.

is sampled efficiently. We are confident that this construction will find wide applicability in the sciences.

References

[1] Doucet, A., de Freitas, N. and Gordon, N., Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.

[2] Chorin, A. J. and Hald, O. H. Stochastic Tools in Mathematics and Science, 2nd edition, Springer-Verlag, New York, 2009.

[3] Morzfeld, M., Tu, X., Atkins, E. and Chorin, A. J., A random map implementation of implicit filters, *J. Comput. Phys.*, **231**, 2012, 2049–2066.

[4] Morzfeld, M. and Chorin, A. J., Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation, *Nonlin. Processes Geophys.*, **19**, 2012, 365–382.

[5] Kloeden, P. E. and Platen, E., Numerical Solution of Stochastic Differential Equations, 3rd edition, Springer-Verlag, New York, 1999.

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

[7] Chorin, A. J., Morzfeld, M. and Tu, X., Implicit particle filters for data assimilation, *Commun. Appl. Math. Comput. Sci.*, **5**(2), 2010, 221–240.

[8] Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T., A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. Signal Process*, **10**, 2002, 197–208.

[9] Bickel, P., Li, B. and Bengtsson, T., Sharp failure rates for the bootstrap particle filter in high dimensions, Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh, 2008, 318–329.

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

[11] Gordon, N. J., Salmon, D. J. and Smith, A. F. M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEEE Proceedings F on Radar and Signal Processing*, **140**, 1993, 107–113.

[12] Doucet, A., Godsill, S. and Andrieu, C., On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, **50**, 2000, 174–188.

[13] Del Moral, P., Feynman-Kac Formulae, Springer-Verlag, New York, 2004.

[14] Del Moral, P., Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems, *Annals of Applied Probability*, **8**(2), 1998, 438–495.

[15] Zaritskii, V. S. and Shimelevich, L. I., Monte Carlo technique in problems of optimal data processing, *Automation and Remote Control*, **12**, 1975, 95–103.

[16] Kalman, R. E., A new approach to linear filtering and prediction theory, *Trans. ASME, Ser. D*, **82**, 1960, 35–48.

[17] Kalman, R. E. and Bucy, R. S., New results in linear filtering and prediction theory, *Trans. ASME, Ser. D*, **83**, 1961, 95–108.

[18] Evensen, G., Data Assimilation, Springer-Verlag, New York, 2007.

[19] Zakai, M., On the optimal filtering of diffusion processes, *Zeit. Wahrsch.*, **11**, 1969, 230–243.

[20] Talagrand, O. and Courtier, P., Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory, *Q. J. R. Meteorol. Soc.*, **113**, 1987, 1311–1328.

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

[22] Courtier, P., Thepaut, J. N. and Hollingsworth, A., A strategy for operational implementation of 4D-var, using an incremental appoach, *Q. J. R. Meteorol. Soc.*, **120**, 1994, 1367–1387.

[23] Courtier, P., Dual formulation of four-dimensional variational assimilation, *Q. J. R. Meteorol. Soc.*, **123**, 1997, 2449–2461.

[24] Talagrand, O., Assimilation of observations, an introduction, *J. R. Meteorol. Soc. of Japan*, **75**(1), 1997, 191–209.

[25] Tremolet, Y., Accounting for an imperfect model in 4D-var, *Q. J. R. Meteorol. Soc.*, **621**(132), 2006, 2483–2504.

[26] Atkins, E., Morzfeld, M. and Chorin, A. J., Implicit particle methods and their connection to variational data assimilation, *Mon. Weather Rev.*, in press.

[27] Kuramoto, Y. and Tsuzuki, T., On the formation of dissipative structures in reaction-diffusion systems, *Progr. Theoret. Phys.*, **54**, 1975, 687–699.

[28] Sivashinsky, G., Nonlinear analysis of hydrodynamic instability in laminar flames, Part I, Derivation of basic equations, *Acta Astronaut.*, **4**, 1977, 1177–1206.

[29] Chorin, A. J. and Krause, P., Dimensional reduction for a Bayesian filter, *PNAS*, **101**, 2004, 15013–15017.

[30] Jardak, M., Navon, I. M. and Zupanski, M., Comparison of sequential data assimilation methods for the Kuramoto-Sivashinsky equation, *Int. J. Numer. Methods Fluids*, **62**, 2009, 374–402.

[31] Lord, G. J. and Rougemont, J., A numerical scheme for stochastic PDEs with Gevrey regularity, *IMA Journal of Numerical Analysis*, **24**, 2004, 587–604.

[32] Jentzen, A. and Kloeden, P. E., Overcoming the order barrier in the numerical approximation of stochastic partial differential equations with additive space-time noise, *Proc. R. Soc. A*, **465**, 2009, 649–667.

[33] Fletcher, R., Practical Methods of Optimization, Wiley, New York, 1987.

[34] Nocedal, J. and Wright, S. T., Numerical Optimization, 2nd edition, Springer-Verlag, New York, 2006.