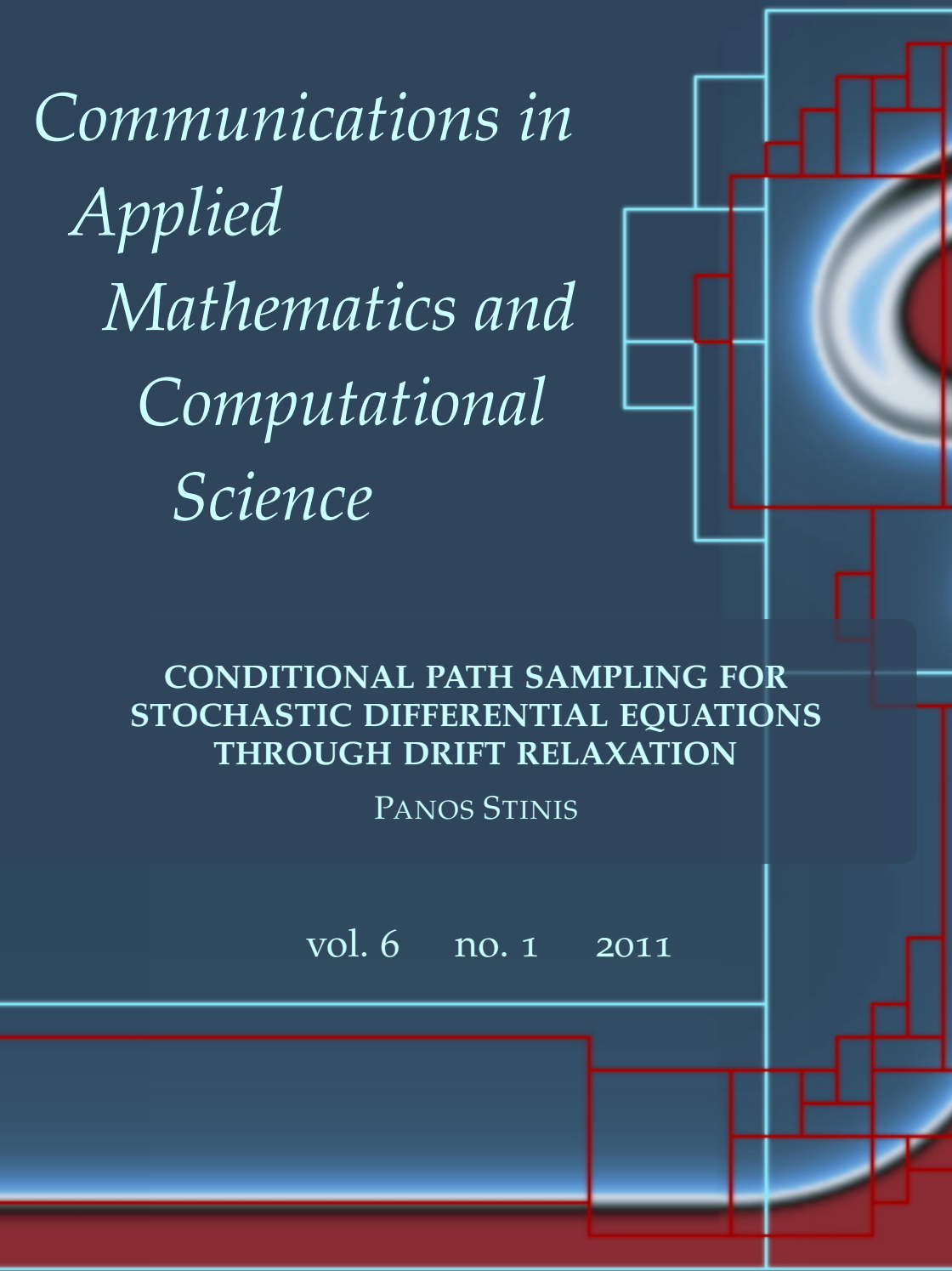
Communications in Applied Mathematics and Computational Science

CONDITIONAL PATH SAMPLING FOR STOCHASTIC DIFFERENTIAL EQUATIONS THROUGH DRIFT RELAXATION

PANOS STINIS

vol. 6 no. 1 2011



mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 6, No. 1, 2011

msp

CONDITIONAL PATH SAMPLING FOR STOCHASTIC DIFFERENTIAL EQUATIONS THROUGH DRIFT RELAXATION

PANOS STINIS

We present an algorithm for the efficient sampling of conditional paths of stochastic differential equations (SDEs). While unconditional path sampling of SDEs is straightforward, albeit expensive for high dimensional systems of SDEs, conditional path sampling can be difficult even for low dimensional systems. This is because we need to produce sample paths of the SDE that respect both the dynamics of the SDE and the initial and endpoint conditions. The dynamics of a SDE are governed by the deterministic term (drift) and the stochastic term (noise). Instead of producing conditional paths directly from the original SDE, one can consider a sequence of SDEs with modified drifts. The modified drifts should be chosen so that it is easier to produce sample paths that satisfy the initial and endpoint conditions. Also, the sequence of modified drifts should converge to the drift of the original SDE. We construct a simple Markov chain Monte Carlo algorithm that samples, in sequence, conditional paths from the modified SDEs, by taking the last sampled path at each level of the sequence as an initial condition for the sampling at the next level in the sequence. The algorithm can be thought of as a stochastic analog of deterministic homotopy methods for solving nonlinear algebraic equations or as a SDE generalization of simulated annealing. The algorithm is particularly suited for filtering/smoothing applications. We show how it can be used to improve the performance of particle filters. Numerical results for filtering of a stochastic differential equation are included.

Introduction

The study of systems arising in different areas, from signal processing and chemical kinetics to econometrics and finance [1; 19] often requires the sampling of paths of stochastic differential equations (SDEs) subject to initial and endpoint conditions. While unconditional path sampling of SDEs is straightforward, albeit expensive for high dimensional systems of SDEs, conditional path sampling can be difficult even for low dimensional systems. This is because we need to produce sample paths of the SDE that respect both the dynamics of the SDE and the initial and endpoint

MSC2000: 60G35, 62M20, 65C05, 65C30, 93E10.

Keywords: conditional path sampling, stochastic differential equations, particle filters, homotopy methods, Monte Carlo, simulated annealing.

conditions. An analogous situation arises in ordinary differential equations, where it can be considerably more difficult to create solutions to boundary value problems than it is to construct solutions to initial value problems (see, for example, Chapter 8 in [5]). The problem of conditional path sampling of SDEs has been a subject of active research in recent years and some very interesting approaches have already been developed [2; 19; 23].

The dynamics of a SDE are governed by the deterministic term (drift) and the stochastic term (noise). Instead of producing conditional paths directly from the original SDE, one can consider a sequence of SDEs with modified drifts. The modified drifts should be chosen so that it is easier to produce sample paths that satisfy the initial and endpoint conditions. Also, the sequence of modified drifts should converge to the drift of the original SDE. We construct a simple Markov chain Monte Carlo (MCMC) algorithm that samples, in sequence, conditional paths from the modified SDEs, by taking the last sampled path at each level of the sequence as an initial condition for the sampling at the next level in the sequence. We have called the algorithm the drift relaxation algorithm.

We have used the drift relaxation algorithm to modify a popular filtering method called particle filter [6]. A particle filter is a sequential importance sampling algorithm based on the recursive (online) Bayesian updating of the values of samples (called particles) to incorporate information from noisy observations of the state of a dynamic model. While the particle filter is a very versatile method it may require a very large number of samples to approximate accurately the conditional density of the state of the model. This has led to considerable research [8; 20; 24] into how one can modify a particle filter to make it more efficient (see also [4; 3] for a different approach to particle filtering). As an application of the drift relaxation algorithm we show in Section 2 how it can be used to construct a more efficient particle filter.

The paper is organized as follows. Section 1 presents the drift relaxation algorithm for an SDE conditional path sampling problem. Section 2 shows how to use the algorithm to modify a particle filter. Section 3 contains numerical results for the application of the modified particle filter to the standard example of filtering a diffusion in a double-well potential (more elaborate examples will be presented in [15]). Finally, Section 4 discusses the results as well as current and future work.

1. Conditional path sampling and drift relaxation

Suppose that we are given a system of stochastic differential equations (SDEs)

$$dX_t = a(X_t)\,dt + \sigma(X_t)\,dB_t, \tag{1}$$

Suppose also that we want to construct, in the time interval $[0, T]$, sample paths from (1) such that the endpoints are distributed according to the densities $h(X_0)$

and $g(X_T)$ respectively. Equation (1) can be discretized in the interval $[0, T]$ by some numerical approximation scheme [11]. Suppose that we have discretized the interval $[0, T]$ using a stepsize $\Delta t = T/I$. Let $0 = T_0 < T_1 < \ldots < T_I = T$. To construct conditional paths of (1) we have to sample the density

$$h(X_{T_0}) \prod_{i=1}^{I} p(X_{T_i}|X_{T_{i-1}}) g(X_{T_I}), \tag{2}$$

where $p(X_{T_i}|X_{T_{i-1}})$ is the transition probability from $X_{T_{i-1}}$ at time T_{i-1} to X_{T_i} at time T_i. The density (2) can be sampled using MCMC assuming that the transition densities $p(X_{T_i}|X_{T_{i-1}})$ can be evaluated. However, the major issue with the MCMC sampling is whether it can be performed efficiently [23; 4]. Instead of MCMC sampling directly from the density (2) i.e., starting from an arbitrary initial path and modifying it to become a path corresponding to (2), we can aid the MCMC sampling process by providing the MCMC sampler of the density (2) with a better initial condition.

To this end, consider an SDE system with modified drift

$$dY_t = b(Y_t)\,dt + \sigma(Y_t)\,dB_t, \tag{3}$$

where $b(Y_t)$ can be suitably chosen to facilitate the conditional path sampling problem (see also comments at the end of this section).

Also, consider the collection of $L+1$ modified SDE systems

$$dY_t^l = (1-\epsilon_l) b(Y_t^l)\,dt + \epsilon_l a(Y_t^l)dt + \sigma(Y_t^l)\,dB_t,$$

where $\epsilon_l \in [0, 1]$, $l = 0, \ldots, L$, with $\epsilon_l < \epsilon_{l+1}$, $\epsilon_0 = 0$ and $\epsilon_L = 1$. Note that the zeroth level SDE corresponds to (3) while the L-th level SDE corresponds to the original SDE (1). Also, for the l-th SDE in the sequence we denote as $p_l(Y^l_{T_i}|Y^l_{T_{i-1}})$ the corresponding transition probability. With this notation, $p_L(Y^L_{T_i}|Y^L_{T_{i-1}}) = p(X_{T_i}|X_{T_{i-1}})$.

The main idea behind drift relaxation is that instead of sampling directly a conditional path for the SDE (1), one can sample a conditional path for the modified SDE (3) and gradually morph the path into a path of (1).

Drift relaxation algorithm:

- Sample through MCMC the density $h(Y^0_{T_0}) \prod_{i=1}^{I} p_0(Y^0_{T_i}|Y^0_{T_{i-1}}) g(Y^0_{T_I})$.
- For $l = 1, \ldots, L$ take the last sample path from the $(l-1)$-st level and use it as an initial condition for MCMC sampling of the density

  $$h(Y^l_{T_0}) \prod_{i=1}^{I} p_l(Y^l_{T_i}|Y^l_{T_{i-1}}) g(Y^l_{T_I})$$

  at the l-th level.
- Keep the last sample path at the L-th level.

We repeat here that the levels from 0 to $L-1$ are auxiliary and only serve the purpose of providing the sampler at level L with a better initial condition. The final sampling is performed at the L-th level that corresponds to the original SDE (1).

The drift relaxation algorithm is similar to simulated annealing (SA), used in equilibrium statistical mechanics [12]. However, instead of modifying a temperature as in SA, here we modify the drift of the system. Also, the idea behind drift relaxation resembles the main idea behind homotopy methods used in deterministic optimization problems [7; 10].

Note that there are two ways to utilize the drift relaxation idea. The first one is by sampling the densities for the different levels sequentially as in the algorithm presented above. The second is to consider the $L+1$ systems *in parallel*, sample simultaneously the conditional densities

$$h(Y^l_{T_0}) \prod_{i=1}^{I} p_l(Y^l_{T_i}|Y^l_{T_{i-1}}) g(Y^l_{T_I}), \quad \text{for } l = 0, \ldots, L,$$

and occasionally swap paths between levels (the swapping of paths between levels should be performed in a manner that preserves detailed balance [12]). This approach is in the spirit of parallel tempering used in Monte Carlo sampling [12]. In the current work we have applied the drift relaxation idea only in the form presented in the algorithm above.

We end this section with a brief discussion on the choice of the modified drift. For stochastic gradient flows with transitions between multiple metastable states, one can choose the modified drift as a mollified version of the original drift (see also the discussion in Section 3). This amounts to making the potential wells shallower and thus facilitates the transitions between metastable states. For general problems, one can choose to use for the modified drift a mean-field drift. This has been used successfully by the author to improve the performance of particle filters for multiple target tracking [15].

2. Application to particle filtering

We show in this section how the drift relaxation algorithm can be applied to particle filtering with the aim of bringing the samples closer to the observations.

2.1. ***Generic particle filter.*** Suppose that we are given an SDE system and that we also have access to noisy observations $Z_{T_1}, \ldots, Z_{T_K}$ of the state of the system at specified instants $T_1, \ldots, T_K$. The observations are functions of the state of the system, say given by $Z_{T_k} = G(X_{T_k}, \xi_k)$, where $\xi_k, k = 1, \ldots, K$ are mutually independent random variables. For simplicity, let us assume that the distribution of the observations admits a density $g(X_{T_k}, Z_{T_k})$, i.e., $p(Z_{T_k}|X_{T_k}) \propto g(X_{T_k}, Z_{T_k})$.

The filtering problem consists of computing estimates of the conditional expectation $E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k]$, i.e., the conditional expectation of the state of the system given the (noisy) observations. Equivalently, we are looking to compute the conditional density of the state of the system given the observations $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$. There are several ways to compute this conditional density and the associated conditional expectation but for practical applications they are rather expensive.

Particle filters fall in the category of importance sampling methods. Because computing averages with respect to the conditional density involves the sampling of the conditional density, which can be difficult, importance sampling methods proceed by sampling a reference density $q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$, which can be easily sampled and then compute the weighted sample mean

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{1}{N}\sum_{n=1}^{N} f(X_{T_k}^n)\frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)},$$

or the related estimate

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{\sum_{n=1}^{N} f(X_{T_k}^n)\frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}}{\sum_{n=1}^{N}\frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}}, \tag{4}$$

where N has been replaced by the approximation

$$N \approx \sum_{n=1}^{N}\frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}.$$

Particle filtering is a recursive implementation of the importance sampling approach. It is based on the recursion

$$p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k) \propto g(X_{T_k}, Z_{T_k})p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}), \tag{5}$$

$$\text{where } p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}) = \int p(X_{T_k}|X_{T_{k-1}})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}. \tag{6}$$

If we set

$$q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k) = p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}),$$

then from (5) we get

$$\frac{p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)} \propto g(X_{T_k}, Z_{T_k}).$$

The approximation in expression (4) becomes

$$E[f(X_{T_i})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{\sum_{n=1}^{N} f(X_{T_k}^n)g(X_{T_k}^n, Z_{T_k})}{\sum_{n=1}^{N} g(X_{T_k}^n, Z_{T_k})}. \tag{7}$$

From (7) we see that if we can construct samples from the predictive distribution $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1})$ then we can define the (normalized) weights

$$W_{T_k}^n = \frac{g(X_{T_k}^n, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^n, Z_{T_k})}$$

and use them to weigh the samples, and the weighted samples will be distributed according to the posterior distribution $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k})$.

In many applications, most samples will have a negligible weight with respect to the observation, so carrying them along does not contribute significantly to the conditional expectation estimate (this is the problem of degeneracy [12]). To create larger diversity one can resample the weights to create more copies of the samples with significant weights. The particle filter with resampling is summarized in the following algorithm, due to Gordon et al. [9].

Particle filter:

(1) Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})$.

(2) *Prediction*: Generate N samples $X_{T_k}'^n$ from $p(X_{T_k}|X_{T_{k-1}})$.

(3) *Update*: Evaluate the weights

$$W_{T_k}^n = \frac{g(X_{T_k}'^n, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}'^n, Z_{T_k})}.$$

(4) *Resampling*: Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \ldots, N$ let $X_{T_k}^n = X_{T_k}'^j$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^{j} W_{T_k}^l$$

where j can range from 1 to N.

(5) Set $k = k + 1$ and proceed to Step 1.

The particle filter algorithm is easy to implement and adapt for different problems since the only part of the algorithm that depends on the specific dynamics of the problem is the prediction step. This has led to the particle filter algorithm's increased popularity [6]. However, even with the resampling step, the particle filter can still need a lot of samples in order to describe accurately the conditional density $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k})$. Snyder et al. [18] have shown how the particle filter can fail in simple high dimensional problems because one sample dominates the weight distribution. The rest of the samples are not in statistically significant regions. Even worse, as we will show in the numerical results section, there are simple examples where not even one sample is in a statistically significant region. In the

next subsection we present how drift relaxation can be used to push samples closer to statistically significant regions.

2.2. ***Particle filter with MCMC step.*** Several authors (see, e.g., [8; 24]) have suggested the use of a MCMC step after the resampling step (Step 4) in order to move samples away from statistically insignificant regions. There are many possible ways to append an MCMC step after the resampling step in order to achieve that objective. The important point is that the MCMC step must preserve the conditional density $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$.

We begin by noting that one can use the resampling step (Step 4) in the particle filter algorithm to create more copies not only of the good samples according to the observation, but also of the values (initial conditions) of the samples at the previous observation. These values are the ones who have evolved into good samples for the current observation (see more details in [24]). The motivation behind producing more copies of the pairs of initial and final conditions is to use the good initial conditions as starting points to produce statistically more significant samples according to the current observation. This process can be accomplished in two steps. First, Step 4 of the particle filter algorithm is replaced by:

Resampling. Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \ldots, N$ let $(X_{T_{k-1}}^n, X_{T_k}^n) = (X_{T_{k-1}}^{\prime j}, X_{T_k}^{\prime j})$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \le \theta^j < \sum_{l=1}^{j} W_{T_k}^l.$$

Also, with Bayes' rule [24] one can show that the posterior density $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$ is preserved if one samples from the density

$$g(X_{T_k}, Z_{T_k}) p(X_{T_k}|X_{T_{k-1}}),$$

where $X_{T_{k-1}}$ are given by the modified resampling step. This is a problem of conditional sampling for (continuous-time or discrete) stochastic systems. The important issue is to perform the necessary sampling efficiently [4; 24]. We propose to do that here using drift relaxation (see Section 1). The particle filter with MCMC step algorithm is given by:

Particle filter with MCMC step.

(1) Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})$.

(2) *Prediction*: Generate N samples $X_{T_k}^{\prime n}$ from $p(X_{T_k}|X_{T_{k-1}})$.

(3) *Update*: Evaluate the weights

$$W_{T_k}^n = \frac{g(X_{T_k}^{\prime n}, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^{\prime n}, Z_{T_k})}.$$

(4) *Resampling*: Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \ldots, N$ let $(X^n_{T_{k-1}}, X^n_{T_k}) = (X'^j_{T_{k-1}}, X'^j_{T_k})$ where

$$\sum_{l=1}^{j-1} W^l_{T_k} \leq \theta^j < \sum_{l=1}^{j} W^l_{T_k}, \quad j = 1, \ldots, N.$$

(5) *MCMC step*: For $n = 1, \ldots, N$ choose a modified drift (possibly different for each n). Construct through drift relaxation a Markov chain for $Y^n_{T_k}$ with initial value $X^n_{T_k}$ and stationary distribution

$$g(Y^n_{T_k}, Z_{T_k}) p(Y^n_{T_k} | X^n_{T_{k-1}}).$$

(6) Set $X^n_{T_k} = Y^n_{T_k}$.

(7) Set $k = k + 1$ and proceed to Step 1.

3. Numerical results

We present numerical results of the particle filter algorithm with MCMC step for the standard problem of diffusion in a double-well potential (more elaborate applications of the method will be presented elsewhere [15]). Our objective here is to show how the generic particle filter's performance can be significantly improved by incorporating the MCMC step via drift relaxation.

The problem of diffusion in a double well potential is described by the scalar SDE

$$dX_t = -4X_t(X_t^2 - 1) + \tfrac{1}{2} dB_t. \tag{8}$$

The deterministic part (drift) describes a gradient flow for the potential $U(x) = x^4 - 2x^2$, which has two minima, at $x = \pm 1$. In the notation of Section 1 we have $a(X_t) = -4X_t(X_t^2 - 1)$ and $\sigma(X_t) = \frac{1}{2}$. If the stochastic term is zero the solution wanders around one of the minima depending on the value of the initial condition. A weak stochastic term leads to rare transitions between the minima of the potential. We have chosen the coefficient $\frac{1}{2}$ to make the stochastic term rather weak. This is done because we plan to enforce the observations to alternate among the minima, and thus check if the particle filter can track these transitions.

The SDE (8) is discretized by the Euler–Maruyama [11] scheme with step size $\Delta t = 10^{-2}$. The initial condition is set to -1 and there is a total of 10 observations at $T_k = k, k = 1, \ldots, 10$. The observations are given by $Z_{T_k} = X_{T_k} + \xi_k$, where $\xi_k \sim N(0, 0.01)$ for $k = 1, \ldots, 10$. Note that we have chosen a rather small variance for the observation noise, which in turn makes the filtering problem more difficult. For this choice of observation noise, the observation density (also called likelihood) is given by

$$g(X_{T_k}, Z_{T_k}) \propto \exp\left[-\frac{(Z_{T_k} - X_{T_k})^2}{2 \times 0.01} \right]. \tag{9}$$

The observations alternate between 1 and -1. In particular, for $k = 1, \ldots, 10$ we have $Z_{T_k} = -1$ if k is odd and $Z_{T_k} = 1$ if k is even. Given that the stochastic term is rather weak, such frequent transitions between the two potential minima are rare.

In order to apply the MCMC step with drift relaxation we need to define the modified drift $b(Y_t)$ for the process Y_t given by

$$dY_t = b(Y_t) + \tfrac{1}{2} dB_t. \tag{10}$$

The modified drift can be the same for all the samples or different for each sample. Since the difficulty in tracking the observations comes from the inability of the original SDE (8) to make frequent transitions between the two minima of the double well, an intuitively appealing choice for $b(Y_t)$ is $b(Y_t) = -\alpha 4 Y_t (Y_t^2 - 1)$, where $0 < \alpha < 1$. This drift corresponds to the potential $W(y) = \alpha(y^4 - 2y^2)$. The potential $W(y)$ has its minima also located at $y = \pm 1$. However, the value of the potential at the minima is $-\alpha$ instead of -1 for the potential $U(x)$. This means that the wells corresponding to the minima of $W(y)$ are shallower than the wells corresponding to the minima of $U(x)$. This makes the transitions between the two wells for the process Y_t more frequent than for the original process X_t. For the numerical experiments we have chosen $\alpha = 0.1$.

The sequence of modified SDEs for the drift relaxation algorithm with L levels is given by

$$dY_t^l = (1 - \epsilon_l) b(Y_t^l) dt + \epsilon_l a(Y_t^l) dt + \tfrac{1}{2} dB_t, \tag{11}$$

where $\epsilon_l \in [0, 1]$, $l = 0, \ldots, L$, with $\epsilon_l < \epsilon_{l+1}$, $\epsilon_0 = 0$ and $\epsilon_L = 1$. For our numerical experiments we chose $L = 10$ and $\epsilon_l = l/10$.

Recall that the density we want to sample during the MCMC step is given by

$$g(X_{T_k}, Z_{T_k}) p(X_{T_k} | X_{T_{k-1}}),$$

where $p(X_{T_k} | X_{T_{k-1}})$ is the transition probability between $X_{T_{k-1}}$ and X_{T_k}. For many applications, sampling directly from $p(X_{T_k} | X_{T_{k-1}})$ may be impossible. Thus, one needs to resort to some numerical approximation scheme to approximate the path between $X_{T_{k-1}}$ and X_{T_k} by a discretized path. However (see [24] for details), even the evaluation of the discretized path's density may not be efficient. Instead, by using the fact that each Brownian path in (8) gives rise to a unique path for X_t [17], we can replace the sampling of $g(X_{T_k}, Z_{T_k}) p(X_{T_k} | X_{T_{k-1}})$ by sampling from the density

$$\exp\left[-\frac{(Z_T - X_T^n(\{\Delta B_i^n\}_{i=0}^{I-1}))^2}{2 \times 0.01} \right] \prod_{i=0}^{I-1} \exp\left[-\frac{(\Delta B_i^n)^2}{2\,\Delta t} \right] =$$

$$\exp\left[-\left(\frac{(Z_T - X_T^n(\{\Delta B_i^n\}_{i=0}^{I-1}))^2}{2 \times 0.01} + \sum_{i=0}^{I-1} \frac{(\Delta B_i^n)^2}{2\,\Delta t} \right) \right], \tag{12}$$

where $\{\Delta B_i^n\}_{i=0}^{I-1}$ are the Brownian increments of the discretized path connecting $X_{T_{k-1}}$ and X_{T_k}. Also, note that the final point X_{T_k} has now become a function of the entire Brownian path $\{\Delta B_i^n\}_{i=0}^{I-1}$. For the numerical experiments we have chosen

$$\Delta t = \frac{T_k - T_{k-1}}{I} = 10^{-2},$$

which, since $T_k - T_{k-1} = 1$, gives $I = 100$.

We use drift relaxation to produce samples from the density (12). The Markov chain at each level of the drift relaxation algorithm is constructed using hybrid Monte Carlo (HMC) [12]. At the l-th level, we can discretize (11), say with the Euler–Maruyama scheme, and the points on the path will be given by

$$Y_{i\Delta t}^{l,n} = Y_{(i-1)\Delta t}^{l,n} + (1-\epsilon_l) b(Y_{(i-1)\Delta t}^{l,n})\Delta t + \epsilon_l a(Y_{(i-1)\Delta t}^{l,n})\Delta t + \tfrac{1}{2}\Delta B_{i-1}^{l,n},$$

for $i = 1, \ldots, I$. We can use more sophisticated schemes than the Euler–Maruyama scheme for the discretization of the simplified SDE (10) at the cost of making the expression for the density more complicated.

We can define a potential $V_{\epsilon_l}(\{\Delta B_i^{l,n}\}_{i=0}^{I-1})$ for the variables $\{\Delta B_i^{l,n}\}_{i=0}^{I-1}$. The potential is given by

$$V_{\epsilon_l}\big(\{\{\Delta B_i^{l,n}\}_{i=0}^{I-1}\big) = \frac{(Z_T - Y_{I\Delta t}^{l,n}(\{\Delta B_i^{l,n}\}_{i=0}^{I-1}))^2}{2\times 0.01} + \sum_{i=0}^{I-1}\frac{(\Delta B_i^{l,n})^2}{2\,\Delta t},$$

and the density to be sampled can be written as

$$\exp\big[-V_{\epsilon_l}\big(\{\Delta B_i^{l,n}\}_{i=0}^{I-1}\big)\big].$$

The subscript ϵ_l is to denote the dependence of the potential on the drift relaxation parameter ϵ_l. In HMC one considers the variables on which the potential depends as the position variables of a Hamiltonian system. In our case we have I position variables so we can define a I-dimensional position vector $\{q_i\}_{i=1}^I$. The next step is to augment the position variables vector by a vector of associated momenta $\{p_i\}_{i=1}^I$. Together they form a Hamiltonian system with Hamiltonian given by

$$H_{\epsilon_l}(\{q_i\}_{i=1}^I, \{p_i\}_{i=1}^I) = V_{\epsilon_l}(\{q_i\}_{i=1}^I) + \frac{p^T p}{2},$$

where $p = (p_1, \ldots, p_I)$ is the vector of momenta. Thus, the momenta variables are Gaussian distributed random variables with mean zero and variance 1. The equations of motion for this Hamiltonian system are given by Hamilton's equations

$$\frac{dq_i}{d\tau} = \frac{\partial H_{\epsilon_l}}{\partial p_i} \quad \text{and} \quad \frac{dp_i}{d\tau} = -\frac{\partial H_{\epsilon_l}}{\partial q_i}, \quad \text{for } i = 1, \ldots, I. \tag{13}$$

HMC proceeds by assigning initial conditions to the momentum variables (through sampling from $\exp(-p^T p/2)$), evolving the Hamiltonian system in fictitious time

τ for a given number of steps of size $\delta\tau$ and then using the solution of the system to perform a Metropolis accept/reject step (more details in [12]). After the Metropolis step, the values of the momenta are discarded. The most popular method for solving the Hamiltonian system, which is the one we also used, is the Verlet leapfrog scheme. In our numerical implementation, we did not attempt to optimize the performance of the HMC algorithm. For the sampling at each level of the drift relaxation process we used 10 Metropolis accept/reject steps and 1 HMC step of size $\delta\tau = 10^{-2}$ to construct a trial path. A detailed study of the drift relaxation/HMC algorithm for conditional path sampling problems outside of the context of particle filtering will be presented in a future publication.

For the chosen values of the parameters for the drift relaxation and HMC steps, the particle filter with MCMC step is about 500 times more expensive per sample (particle) than the generic particle filter. However, we show that this increase in cost per sample is worthwhile. Figure 1 compares the performance of the particle filter with MCMC step with 10 samples and the generic particle filter with 5000 samples. It is obvious that the particle filter with MCMC step follows accurately all the transitions between the two minima of the double-well. On the other hand, the generic particle filter captures accurately only every other observation. It fails to perform the transitions between the two minima of the double-well.

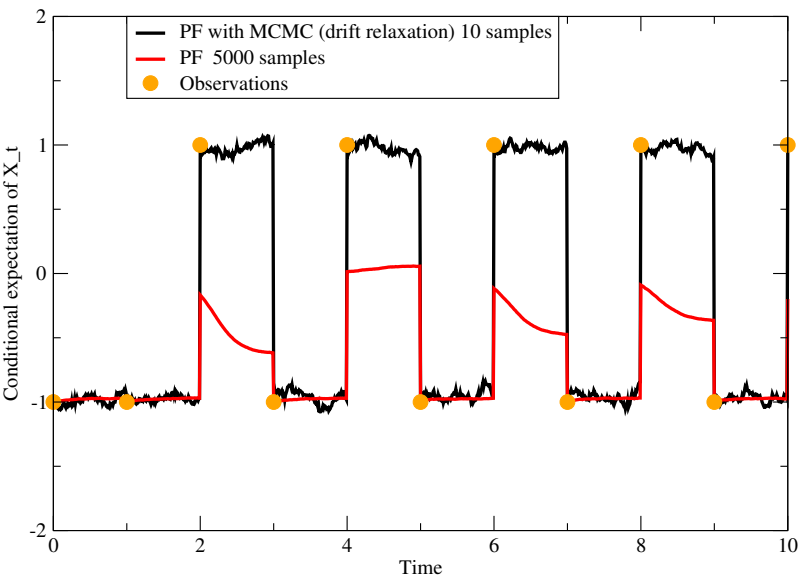


Figure 1. Comparison of the conditional expectation of X_t as computed by the generic particle filter and the particle filter with MCMC step.

Since the particle filter with MCMC step uses only 10 samples the conditional expectation estimate of the hidden signal is not as smooth as the estimate of the generic particle filter, which uses 5000 samples. The generic particle filter needs about 10^5 samples to capture accurately the transitions between the two minima. However, from the 10^5 samples, only 2 or 3 dominate the observation weight distribution at each transition, thus making the use of the generic particle filter very inefficient.

Finally, we compare the performance of the particle filter with MCMC step and drift relaxation to a particle filter with MCMC step without drift relaxation. This comparison is made to examine whether the drift relaxation algorithm offers any advantage over direct sampling of the conditional density (12). The particle filter with MCMC step without drift relaxation involved 110 Metropolis accept/reject steps and 1 HMC step of size $\delta\tau = 10^{-2}$ to construct a trial path for each observation. This makes the computational complexity the same as for the particle filter with drift relaxation.

From Figure 2 one can see that there is an advantage in the use of drift relaxation as far as the conditional expectation estimate is concerned. For the majority of the observations, the MCMC step with drift relaxation gives superior results to the particle filter with MCMC step without drift relaxation. In particular, the particle filter without drift relaxation is not as effective in bringing the samples close to the

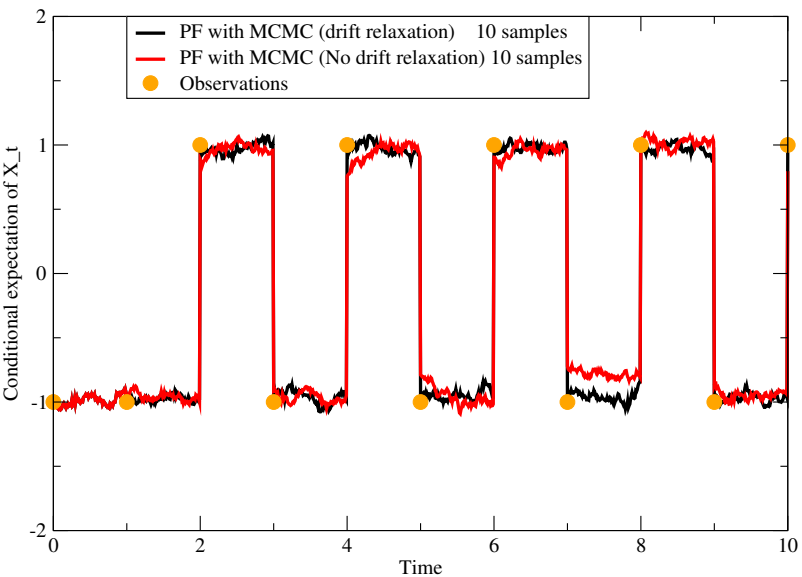


Figure 2. Comparison of the conditional expectation of X_t as computed by the particle filter with MCMC step and drift relaxation and the particle filter with MCMC step without drift relaxation.

observations as the particle filter with drift relations. The mathematical reason for the better performance of the MCMC step with drift relaxation is that the shallower modified potential allows the density of the observation $g(X_{T_k}, Z_{T_k})$ to alter faster the Brownian increments that give rise to the path between the two wells (this is straightforward to see by examination of Hamilton's equations (13) for the HMC sampler). Indeed, for the particle filter with drift relaxation, about 70% of the samples have already crossed from one well to the other after the zeroth level MCMC sampling. Numerical experiments with different choices in the number of drift relaxation levels and/or number of Metropolis accept/reject steps in HMC support the trend shown in Figure 2.

In Figure 3 we plot the error estimate of the conditional expectation estimate of X_t as computed by the particle filter with MCMC step with and without drift relaxation. The error is a measure of the tightness of the distribution of the sample values around the mean. It is obvious that the use of drift relaxation leads to a tighter distribution of the samples around the mean.

In order for the particle filter with MCMC step without drift relaxation to obtain an estimate comparable to the one of the filter with drift relaxation shown in Figure 2, one needs to use about 1000 Metropolis accept/reject steps. This is about 10 times more expensive than the particle filter with drift relaxation. This corroborates the

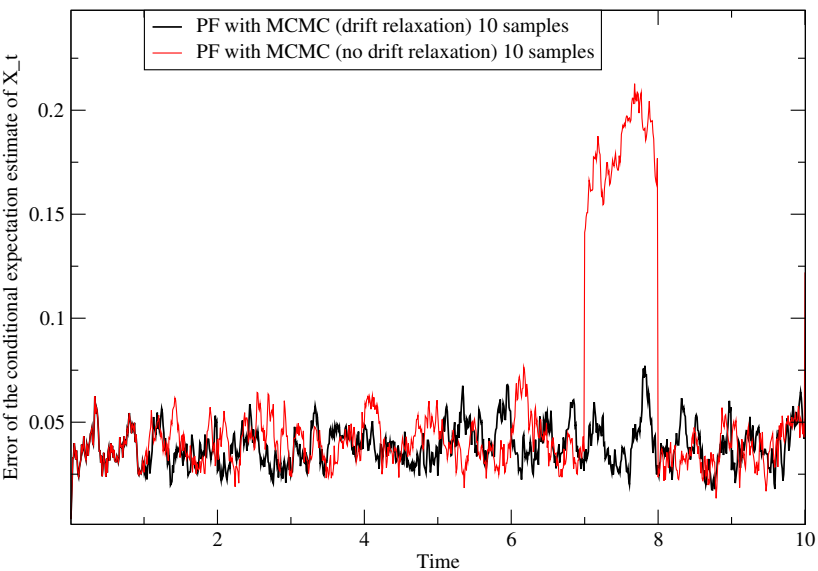


Figure 3. Comparison of the error estimate of the conditional expectation estimate of X_t as computed by the particle filter with MCMC step and drift relaxation and the particle filter with MCMC step without drift relaxation.

conclusion reached in [24], that the use of an MCMC step on its own is not enough to make for a more efficient particle filter. In particular, one has to use an *efficient* algorithm for implementing the MCMC step. A more thorough comparison between the MCMC step with and without drift relaxation will be published elsewhere.

4. Discussion

We have presented an algorithm for conditional path sampling of SDEs. The proposed algorithm is based on drift relaxation, which allows to sample conditional paths from a modified drift equation. The conditional paths of the modified drift equation are then morphed into conditional paths of the original equation. We have called this process of gradually enforcing the drift of the original equation drift relaxation. The algorithm has been used to create a modified particle filter for SDEs. We have shown that the modified particle filter's performance is significantly better than the performance of a generic particle filter.

In the current work, we have examined the application of drift relaxation to the filtering problem of diffusion in a double-well potential, a standard example in the filtering literature. The same algorithm can be applied to the problem of tracking a single target. A problem of great practical interest is that of tracking not only one but multiple moving targets [13; 16; 21; 22]. The multitarget tracking problem is much more difficult than the single-target problem due to the combinatorial explosion of the number of possible target-observation association arrangements. In this context, the accurate tracking of each target becomes crucial. Suppose that only one of the targets is of interest and the rest act as decoys [14]. The inability to track each potential target accurately can lead to ambiguity about the targets' movement if the observations for different targets are close. We have already applied the drift relaxation modified particle filter to multitarget tracking problems with very encouraging results that will appear elsewhere [15].

Acknowledgements

I am grateful to Profs. V. Maroulas and J. Weare for many helpful discussions and comments. I would also like to thank for its hospitality the Institute for Mathematics and its Applications (IMA) at the University of Minnesota where the current work was completed.

References

[1] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, *Transition path sampling: throwing ropes over rough mountain passes in the dark*, Ann. Rev. Phys. Chem. **53** (2002), 291–318.

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

[3] A. J. Chorin, M. Morzfeld, and X. Tu, *Implicit filters for data assimilation*, Comm. Appl. Math. Comp. Sc. **5** (2010), no. 2, 221–240. Zbl 05833566

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

[5] P. Deuflhard and F. Bornemann, *Scientific computing with ordinary differential equations*, Texts in Applied Mathematics, no. 42, Springer, New York, 2002. MR 2003e:65001 Zbl 1001.65071

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

[7] D. Dunlavy and D. O'Leary, *Homotopy optimization methods for global optimization*, technical report SAND2005-7495, Sandia National Labs, 2005.

[8] W. R. Gilks and C. Berzuini, *Following a moving target: Monte Carlo inference for dynamic Bayesian models*, J. R. Stat. Soc. Ser. B Stat. Methodol. **63** (2001), no. 1, 127–146. MR 2001m:62038 Zbl 0976.62021

[9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, *Novel approach to nonlinear/non-Gaussian bayesian state estimation*, Proc. Inst. Elect. Eng. F **140** (1993), 107–113.

[10] C. Hillermeier, *Nonlinear multiobjective optimization: A generalized homotopy approach*, International Series of Numerical Math., no. 135, Birkhäuser, Basel, 2001. MR 2002b:90002 Zbl 0966.90069

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

[12] J. S. Liu, *Monte Carlo strategies in scientific computing*, Springer Series in Statistics, no. 16, Springer, New York, 2001. MR 2002i:65006 Zbl 0991.65001

[13] R. P. S. Mahler, *Statistical multisource-multitarget information fusion*, Artech House, London, 2007. Zbl 1126.68080

[14] R. P. S. Mahler and V. Maroulas, *Tracking spawning objects*, preprint, 2010.

[15] V. Maroulas and P. Stinis, *A drift relaxation Monte Carlo approach to particle filtering for multi-target tracking*, preprint, 2010. arXiv 1006.3100v3

[16] W. Ng, J. F. Li, S. J. Godsill, and J. Vermaak, *A hybrid approach for online joint detection and tracking for multiple targets*, 2005, pp. 2126–2141.

[17] B. Oksendal, *Stochastic differential equations: An introduction with applications*, 6th ed., Springer, 2005.

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

[19] A. M. Stuart, J. Voss, and P. Wiberg, *Conditional path sampling of SDEs and the Langevin MCMC method*, Commun. Math. Sci. **2** (2004), no. 4, 685–697. MR 2119934

[20] P. J. Van Leeuwen, *Nonlinear data assimilation in geosciences: an extremely efficient particle filter*, Q. J. R. Meteorol. Soc. **136** (2010), 1991–1996.

[21] J. Vermaak, S. Godsill, and P. Perez, *Monte Carlo filtering for multi-target tracking and data association*, IEEE Trans. Aero. Elect. Sys. **41** (2005), no. 1, 309–332.

[22] B.-N. Vo, S. Singh, and A. Doucet, *Sequential Monte Carlo methods for multi-target filtering with random finite sets*, IEEE Trans. Aero. Elect. Sys. **41** (2005), no. 4, 1224–1245.

[23] J. Weare, *Efficient Monte Carlo sampling by parallel marginalization*, Proc. Natl. Acad. Sci. USA **104** (2007), no. 31, 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

Received August 30, 2010. Revised February 23, 2011.

PANOS STINIS: stinis@math.umn.edu
School of Mathematics, University of Minnesota, 206 Church St SE, Minneapolis, MN 55455, United States
http://www.math.umn.edu/~stinis/

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 2011 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://msp.org/
A NON-PROFIT CORPORATION
Typeset in LaTeX

Communications in Applied Mathematics and Computational Science

vol. 6 no. 1 2011