Particle Markov Chain Monte Carlo for Efficient Numerical Simulation

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein

Abstract Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) methods are the two most popular classes of algorithms used to sample from general high-dimensional probability distributions. The theoretical convergence of MCMC algorithms is ensured under weak assumptions, but their practical performance is notoriously unsatisfactory when the proposal distributions used to explore the space are poorly chosen and/or if highly correlated variables are updated independently. We show here how it is possible to systematically design potentially very efficient high-dimensional proposal distributions for MCMC by using SMC techniques. We demonstrate how this novel approach allows us to design effective MCMC algorithms in complex scenarios. This is illustrated by a problem of Bayesian inference for a stochastic kinetic model.

1 Introduction

Assume that we are interested in sampling from a probability distribution $\pi(\mathbf{x})$ where $\mathbf{x} = (x_1, \ldots, x_T)$ for some $T > 1$. For ease of presentation, we assume that each $x_i \in \mathcal{X}$ for some space $\mathcal{X}$. For complex problems, it is impossible to sample directly from $\pi(\mathbf{x})$.

Christophe Andrieu
Department of Mathematics, Bristol University, UK
url: http://www.stats.bris.ac.uk/~maxca

Arnaud Doucet
The Institute of Statistical Mathematics, Japan
url: http://www.cs.ubc.ca/~arnaud

Roman Holenstein
Department of Computer Science, University of British Columbia, Canada
url: http://www.cs.ubc.ca/~romanh

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_3, © Springer-Verlag Berlin Heidelberg 2009

The standard MCMC approach consists of sampling long realisations of ergodic Markov chains with invariant distribution $\pi(\mathbf{x})$. The Metropolis-Hastings (MH) algorithm is the main known generic mechanism to define such updates. It requires the choice of proposal distributions that sample possible states for the Markov chain which are either accepted or rejected. A popular application of this principle consists, for example, of repeatedly updating in turn the lower-dimensional components x_i of $\mathbf{x}$ conditional upon the remaining components $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T)$. The size reduction often allows for a better choice of local proposal distributions. Although this strategy can result in an improvement over the full updating of $\mathbf{x}$ in one block, it can still be ineffective when highly dependent components are not updated simultaneously.

SMC methods are an alternative to MCMC methods where a swarm of samples, named particles, evolves towards the distribution of interest according to a combination of importance sampling (IS) and resampling; see [6] for a collection of articles on the subject and [11, chapters 3 and 4]. Where traditional IS would try to directly produce weighted samples to approximate $\pi(\mathbf{x})$, and most likely fail for the same reason that an independent MH (IMH) algorithm would fail, SMC methods decompose the problem of sampling from $\pi(\mathbf{x})$ into a series of "simpler" sub-problems. We introduce a sequence of intermediate "bridging" probability distributions of increasing dimension $\{\pi_n(\mathbf{x}_n), n = 1, \ldots, T-1\}$ with $\mathbf{x}_n = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$, then we sequentially sample approximately from $\pi_1(\mathbf{x}_1), \pi_2(\mathbf{x}_2), \ldots, \pi_{T-1}(\mathbf{x}_{T-1})$ and $\pi_T(\mathbf{x}) = \pi(\mathbf{x})$. As is the case for MCMC algorithms this dimension reduction usually allows for the design of better proposal distributions. In this paper we present a recent addition to the Monte Carlo toolbox named Particle MCMC (PMCMC) which aims to take advantage of the differing strengths of MCMC and SMC methods.

The rest of this paper is organised as follows. In Section 2, we briefly review SMC methods and discuss some of their properties. In Section 3 we present the *particle IMH sampler*, a recently developed IMH update targeting $\pi(\mathbf{x})$ which has the capability of using SMC approximations of $\pi(\mathbf{x})$ as a proposal mechanism [1]. In Section 4, we review extensions of this basic update to the case where we are interested in sampling from $\pi(\theta, \mathbf{x})$ on $\Theta \times \mathcal{X}^T$: the *particle marginal MH sampler* and the *particle Gibbs sampler*. As shown in [1], such updates are of particular interest in the context of inference in state-space models, but their relevance is not limited to such models. Connections to previous work are discussed in Section 5. Finally in Section 6, we demonstrate the performance of the methodology in the context of inference in a stochastic kinetic model. Space constraints prevent us from detailing all the results and proofs; we refer the reader to [1] for details.

2 Sequential Monte Carlo Methods

We briefly review here the principle of SMC methods to sample from a given target $\pi(\mathbf{x})$. We first introduce an artificial sequence of bridging distributions

$\{\pi_n(\mathbf{x}_n); n = 1, \ldots, T-1\}$ of increasing dimension and define $\pi_T(\mathbf{x}_T) = \pi(\mathbf{x})$. Each distribution is assumed known up to a normalising constant, that is

$$\pi_n(\mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_n)}{Z_n},$$

where $\gamma_n : \mathcal{X}^n \to \mathbb{R}^+$ can be evaluated pointwise, but Z_n is unknown. We will use the notation Z for Z_T. An SMC algorithm also requires us to specify an importance distribution $q_1(x_1)$ on $\mathcal{X}$ in order to initialise the recursion at time 1 and a family of proposal distributions $\{q_n(x_n|\mathbf{x}_{n-1}); n = 2, \ldots, T\}$ in order to extend $\mathbf{x}_{n-1} \in \mathcal{X}^{n-1}$ by sampling $x_n \in \mathcal{X}$ conditional upon $\mathbf{x}_{n-1}$ at time instants $n = 2, \ldots, T$. Guidelines on how to best select $q_n(x_n|\mathbf{x}_{n-1})$ are well known, and the main recommendation is to use the conditional distribution $\pi_n(x_n|\mathbf{x}_{n-1})$ or an approximation [6], [11]. An SMC algorithm also involves a resampling procedure of the N particles, which relies on a family of probability distributions $\{r(\cdot|\mathbf{w}), \mathbf{w} \in [0,1]^N\}$ on $\{1, \ldots, N\}^N$. The resampling step is usually necessary as in most applications the variance of the importance weights would otherwise typically increase exponentially with n.

The algorithm proceeds as follows to produce a sequence of samples $\{\mathbf{X}_n^i, i = 1, \ldots, N\}$ for $n = 1, \ldots, T$. Note that we adopt below the convention that whenever the index i is used we mean "for all $i \in \{1, \ldots, N\}$." Further on, we also use the standard convention whereby capital letters are used for random variables while lower case letters are used for their values. We also use the notation $\mathbf{W}_n = \left(W_n^1, \ldots, W_n^N\right)$ and $\mathbf{A}_n = \left(A_n^1, \ldots, A_n^N\right)$.

Sequential Monte Carlo Algorithm

$n = 1$

- Sample $\mathbf{X}_1^i \sim q_1(\cdot)$.
- Update and normalise the weights

$$w_1\left(\mathbf{X}_1^i\right) = \frac{\gamma_1(\mathbf{X}_1^i)}{q_1(\mathbf{X}_1^i)},\ W_1^i = \frac{w_1\left(\mathbf{X}_1^i\right)}{\sum_{k=1}^N w_1\left(\mathbf{X}_1^k\right)}. \tag{1}$$

For $n = 2, \ldots, T$

- Sample $\mathbf{A}_{n-1} \sim r(\cdot|\mathbf{W}_{n-1})$.
- Sample $X_n^i \sim q_n(\cdot|\mathbf{X}_{n-1}^{A_{n-1}^i})$ and set $\mathbf{X}_n^i = (\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i)$.
- Update and normalise the weights

$$w_n\left(\mathbf{X}_n^i\right) = \frac{\gamma_n\left(\mathbf{X}_n^i\right)}{\gamma_{n-1}\left(\mathbf{X}_{n-1}^{A_{n-1}^i}\right) q_n\left(X_n^i \middle| \mathbf{X}_{n-1}^{A_{n-1}^i}\right)},\ W_n^i = \frac{w_n\left(\mathbf{X}_n^i\right)}{\sum_{k=1}^N w_n\left(\mathbf{X}_n^k\right)}. \tag{2}$$

The variable A_{n-1}^i plays an important role in our formulation of SMC methods, and represents the index of the "parent" at time $n-1$ of particle $\mathbf{X}_n^i$ for $n = 2, \ldots, T$. The vector $\mathbf{A}_n$ is thus a random mapping defined on $\{1, \ldots, N\} \to \{1, \ldots, N\}^N$, and

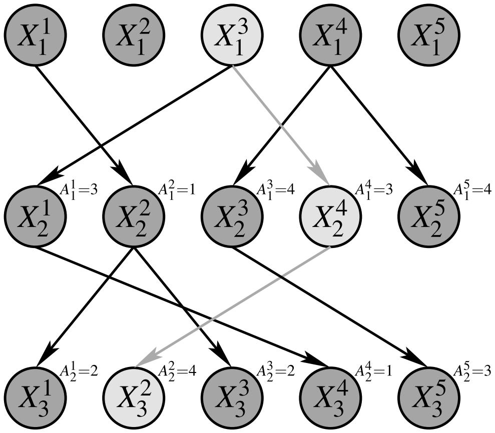


Fig. 1 Example of ancestral lineages generated by an SMC algorithm for $N = 5$ and $T = 3$. The lighter path is $X_{1:3}^2 = (X_1^3, X_2^4, X_3^2)$ and its ancestral lineage is $B_{1:3}^2 = (3,4,2)$.

the resampling procedure is thus interpreted here as being the operation by which child particles at time n choose their parent particles at time $n-1$ according to a probability $r(\cdot|\mathbf{W}_{n-1})$ dependent on the parents' weights $\mathbf{W}_{n-1}$, or "fitness." The introduction of the variables $\mathbf{A}_n$ allows us to keep track of the "genealogy" of particles and is necessary to describe precisely one of the algorithms introduced later on (see Section 4). For this purpose, for $i = 1,\ldots,N$ and $n = 1,\ldots,T$ we introduce B_n^i the index the ancestor particle of $\mathbf{X}_T^i$ at generation n had at that time. More formally for $i = 1,\ldots,N$ we define $B_T^i := i$ and for $n = T-1,\ldots,1$ we have the following backward recursive relation $B_n^i := A_n^{B_{n+1}^i}$. As a result for any $i = 1,\ldots,N$ we have the identity $\mathbf{X}_T^i = (X_1^{B_1^i}, X_2^{B_2^i},\ldots,X_{T-1}^{B_{T-1}^i}, X_T^{B_T^i})$ and $\mathbf{B}_T^i = (B_1^i, B_2^i,\ldots,B_{T-1}^i, B_T^i = i)$ is the ancestral 'lineage' of a particle. This is illustrated in Figure 1.

This SMC algorithm provides an approximation of $\pi(\mathbf{x})$ and its normalising constant Z given by

$$\hat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} W_T^i \delta_{\mathbf{X}_T^i}(\mathbf{x}) \text{ and } \hat{Z} = \prod_{n=1}^{T}\left[\frac{1}{N}\sum_{i=1}^{N} w_n\left(\mathbf{X}_n^i\right)\right]. \tag{3}$$

The validity of the algorithms presented here relies on a set of very weak assumptions. First we require the importance weight functions $w_n(\mathbf{x}_n)$ to be properly defined; *i.e.* the supports of the proposals cover the supports of the targets. Second it also relies on the following assumptions on the resampling procedure.

Let $O_n^i = \sum_{k=1}^{N} \mathbb{I}\left\{A_n^k = i\right\}$ be the number of offspring of particle i at time n. Then for any $i = 1,\ldots,N$ and $n = 1,\ldots,T$ the resampling scheme must satisfy the following unbiasedness condition

$$\mathbb{E}\left[O_n^i|\mathbf{W}_n\right] = NW_n^i. \tag{4}$$

In fact in practice, for computational efficiency, $\mathbf{O}_n = \left(O_n^1, \ldots, O_n^N\right)$ is typically drawn first (*i.e.* without explicit reference to $\mathbf{A}_n$) according to a probability distribution $s(\cdot|\mathbf{W}_n)$ such that (4) holds and the offspring then matched to their parents. For example, the simplest unbiased resampling algorithm consists of sampling $\mathbf{O}_n$ according to a multinomial distribution of parameters $(N, \mathbf{W}_n)$. More sophisticated schemes such as residual resampling [11] and stratified resampling [9] also satisfy (4). Once $\mathbf{O}_n$ has been sampled, this is followed by a deterministic allocation procedure of the child particles to the parents, which defines a new set of indices *e.g.* the O_n^1 first child particles are associated to the parent particle number 1, *i.e.* $A_n^1 = 1, \ldots, A_n^{O_n^1} = 1$, likewise for the O_n^2 following child particles and the parent particle number 2, *i.e.* $A_n^{O_n^1+1} = 2, \ldots, A_n^{O_n^1+O_n^2} = 2$ etc.

Further on, we will impose the slightly stronger unbiasedness condition

$$r\left(A_n^i = k|\mathbf{W}_n\right) = W_n^k \ . \tag{5}$$

Note that even if (4) holds then (5) is not necessarily satisfied, for example by the standard deterministic allocation procedure, but this property can be easily enforced by the addition of a random permutation of these indices. As we shall see our indexing system makes the writing of the probability distributions underpinning our algorithms extremely simple.

Many sharp convergence results have been established for SMC methods including Lp-bounds, central limit theorems, large deviations results etc.; see [4] for a detailed overview of these results.

3 Particle Independent MH Sampler

The aim of this review is to outline how SMC approximations of $\pi(\mathbf{x})$ can be used as proposal distributions for MCMC algorithms. It is natural to suggest the use of the unconditional distribution of a particle generated by an SMC algorithm targeting $\pi(\mathbf{x})$ as a proposal distribution for an IMH algorithm targeting $\pi(\mathbf{x})$. This is likely to result in a very efficient IMH algorithm as discussed in the previous section. It is easy to sample from this unconditional distribution by running an SMC targeting $\pi(\mathbf{x})$ to obtain $\hat{\pi}(\mathbf{x})$ given in (3) and then sample from $\hat{\pi}(\mathbf{x})$. However, computing the MH acceptance ratio of such a MH update would then require us to be able to evaluate

$$q(\mathbf{x}) = \mathbb{E}\left(\hat{\pi}(\mathbf{x})\right) \ , \tag{6}$$

where the expectation is with respect to all the variables used to generate $\hat{\pi}(\mathbf{x})$: this is practically impossible. We show below how it is possible to bypass this problem. We would like to stress at this point the fact that we do not believe that the PIMH algorithm on its own is a practically relevant alternative to standard SMC approximations of $\pi(\mathbf{x})$. However its pedagogical value should become clear below while one should bear in mind that, as it is the case with standard IMH type updates, such an update can be of interest when used in conjunction with other MCMC updates. In order to illustrate the simplicity of the implementation of our approach we describe a particular instance of the methodology in order to sample from $\pi(\mathbf{x})$, where $\mathbf{x}$ is updated in one single block.

3.1 Algorithm

In order to sample from $\pi(\mathbf{x})$ the particle IMH (PIMH) sampler proceeds as follows (with the notation of Section 2, in particular (3)):

Particle Independent Metropolis-Hastings Sampler

Initialization, $m = 0$

- Run an SMC algorithm targeting $\pi(\mathbf{x})$, sample $\mathbf{X}(0) \sim \hat{\pi}(\cdot)$ and compute $\hat{Z}(0)$.

At iteration $m \geq 1$

- Run an SMC algorithm targeting $\pi(\mathbf{x})$, sample $\mathbf{X}^* \sim \hat{\pi}(\cdot)$ and compute $\hat{Z}^*$.
- With probability

$$1 \wedge \frac{\hat{Z}^*}{\hat{Z}(m-1)}, \tag{7}$$

set $\mathbf{X}(m) = \mathbf{X}^*$ and $\hat{Z}(m) = \hat{Z}^*$, otherwise set $\mathbf{X}(m) = \mathbf{X}(m-1)$ and $\hat{Z}(m) = \hat{Z}(m-1)$.

The output of the algorithm is the chain $\{\mathbf{X}(m)\}_{m \geq 0}$. Note the interesting property that the acceptance probability (7) converges to 1 as $N \to \infty$ since both $\hat{Z}^*$ and $\hat{Z}(m-1)$ are consistent estimates of the unknown normalising constant Z, under weak assumptions.

3.2 Extended Proposal and Target Distributions

We show here the surprising result that the invariant distribution of the PIMH sampler is $\pi(\mathbf{x})$ *for any* $N \geq 1$. The key to establish this result is to reformulate the PIMH as a standard IMH sampler defined on an extended state-space with a suitable invariant distribution.

Sampling from the proposal $q(\mathbf{x})$ in (6) requires sampling $\hat{\pi}(\mathbf{x})$ then drawing one particle $\mathbf{X}_T$ from $\hat{\pi}(\mathbf{x})$ by setting $\mathbf{X} = \mathbf{X}_T^K$ where $\Pr\left(K = k \middle| \hat{\pi}(\mathbf{x})\right) = W_T^k$. Denoting for $n = 1, \ldots, T$ the set of N simulated $\mathcal{X}$-valued random variables at time n as $\overline{\mathbf{X}}_n := \left(X_n^1, \ldots, X_n^N\right) \in \mathcal{X}^N$, then the joint probability distribution of all the random variables used in the proposal distribution is

$$q\left(k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) = w_T^k \; \psi\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \tag{8}$$

where w_T^k is a realization of W_T^K and

$$\psi\left(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right) := \left(\prod_{i=1}^{N} q_1\left(x_1^i\right)\right)\prod_{n=2}^{T}\left(r\left(\mathbf{a}_{n-1}|\mathbf{w}_{n-1}\right)\prod_{i=1}^{N} q_n\left(x_n^i\,|\,\mathbf{x}_{n-1}^{a_{n-1}^i}\right)\right)$$

is the distribution of all the random variables generated by the SMC sampler described in Section 2, which is defined on $\mathcal{X}^{TN}\times\{1,\ldots,N\}^{(T-1)N+1}$. We now define, on the same space, the following artificial target probability distribution

$$\begin{aligned}&\tilde{\pi}\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right) \qquad (9)\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)}{N^T}\frac{\psi\left(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)}{q_1(x_1^{b_1^k})\prod_{n=2}^{T} r(b_{n-1}^k|\mathbf{w}_{n-1})q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})}\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)}{N^T}\prod_{i=1,i\neq b_1^k}^{T} q_1(x_1^i)\prod_{n=1}^{T-1} r(\mathbf{a}_{n-1}^{-b_n^k}|\mathbf{w}_{n-1},b_n^k)\prod_{i=1,i\neq b_n^k}^{T} q_n(x_n^i\,|\,\mathbf{x}_{n-1}^{a_{n-1}^i})\end{aligned}$$

where we have used the notation $\mathbf{a}_{n-1}^{-b_n^k}=\mathbf{a}_{n-1}\backslash\{a_{n-1}^{b_n^k}\}$. By construction, we have $\mathbf{X}_T^K\sim\pi$ under $\widetilde{\pi}$ and it is easy to check that

$$\begin{aligned}\frac{\tilde{\pi}\left(k,\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)}{q(k,\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1})} &= \frac{1}{N^T}\frac{\pi\left(\mathbf{x}_T^k\right)}{w_T^k\; q_1(x_1^{b_1^k})\prod_{n=2}^{T} r(b_{n-1}^k|\mathbf{w}_{n-1})q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})}\\ &= \frac{1}{N^T}\frac{\pi\left(\mathbf{x}_T^k\right)}{q_1(x_1^{b_1^k})\prod\limits_{n=2}^{T} q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})\prod\limits_{n=1}^{T} w_n^{b_n^k}}\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)\prod\limits_{n=1}^{T}\left(\frac{1}{N}\sum_{m=1}^{N} w_n\left(\mathbf{x}_n^m\right)\right)}{q_1(x_1^{b_1^k})\prod\limits_{n=2}^{T} q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})\prod\limits_{n=1}^{T} w_n(\mathbf{x}_n^{b_n^k})}\\ &= \frac{\hat{Z}}{Z}.\end{aligned}$$

In the calculations above we have used (5) on the second line whereas the final result is obtained thanks to the definitions of the incremental weights (1)–(2) and of the normalising constant estimate (3). This allows us to conclude that the PIMH sampler is a standard IMH sampler of target distribution $\tilde{\pi}\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)$ and proposal distribution $q(k,\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1})$. This indeed follows by the definition of $q(k,\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1})$ and the last calculation above which explains the form of the acceptance probability of the PIMH. This IMH sampler is automatically irreducible and aperiodic as we have made the assumption that the importance weight functions $w_n\left(\mathbf{x}_n\right)$ are properly defined.

3.3 Structure of the Invariant Distribution and Alternative Algorithm

To better understand the structure of the artificial target $\tilde{\pi}$, we explain here how we would sample from it. The algorithm follows straightforwardly from (9).

- Sample uniformly on $\{1, \ldots, N\}^T$ an ancestral lineage $\mathbf{B}_T^K = \left(B_1^K, B_2^K, \ldots, B_T^K\right)$. Recall that we have $B_T^K = K$, $B_n^K := A_n^{B_{n+1}^K}$.
- Sample $\mathbf{X}_T^K = (X_1^{B_1^K}, X_2^{B_2^K}, \ldots, X_{T-1}^{B_{T-1}^K}, X_T^{B_T^K}) \sim \pi$. Obviously we cannot do this, which is why we are using MCMC in the first place.
- Sample all the remaining variables conditional upon $\left(\mathbf{X}_T^K, \mathbf{B}_T^K\right)$ according to their conditional distribution under $\tilde{\pi}$.

Sampling from this conditional distribution under $\tilde{\pi}$ can be achieved using the following conditional SMC algorithm. We recall that $\mathbf{A}_{n-1}^{-B_n^K} = \mathbf{A}_{n-1} \backslash \{A_{n-1}^{B_n^K}\}$.

Conditional Sequential Monte Carlo Algorithm

$n = 1$

- For $i \neq B_1^K$, sample $\mathbf{X}_1^i \sim q_1(\cdot)$.
- Compute $w_1\left(\mathbf{X}_1^i\right)$ and normalise the weights $W_1^i \propto w_1\left(\mathbf{X}_1^i\right)$.

For $n = 2, \ldots, T$

- Sample $\mathbf{A}_{n-1}^{-B_n^K} \sim r(\cdot | \mathbf{W}_{n-1}, A_{n-1}^{B_n^K})$.
- For $i \neq B_n^K$, sample $X_n^i \sim q_n(\cdot | \mathbf{X}_{n-1}^{A_{n-1}^i})$ and set $\mathbf{X}_n^i = (\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i)$.
- Compute $w_n\left(\mathbf{X}_n^i\right)$ and normalise the weights $W_n^i \propto w_n\left(\mathbf{X}_n^i\right)$.

In the case of multinomial resampling, denoting $\mathcal{B}(a, \mathbf{b})$ the binomial distribution of parameters $(a, \mathbf{b})$, $\mathcal{B}^+(a, \mathbf{b})$ the binomial distribution of similar parameters restricted to $\{1, \ldots, N\}$ and $\mathcal{M}(a, \mathbf{b})$ the multinomial distribution, an efficient approach to sample $\mathbf{A}_{n-1}^{-B_n^K} \sim r(\cdot | \mathbf{W}_{n-1}, A_{n-1}^{B_n^K})$ proceeds as follows.

- Sample $O_{n-1}^{B_n^K} \sim \mathcal{B}^+\left(N, W_{n-1}^{B_n^K}\right)$.
- Allocate randomly $O_{n-1}^{B_n^K} - 1$ parent indexes uniformly in $\{1, \ldots, N\} \backslash \left\{B_n^K\right\}$ and set these parents equal to B_{n-1}^K.
- For $i \neq B_n^K$ compute $\overline{W}_{n-1}^i \propto W_{n-1}^i$ with $\sum_{i=1, i \neq B_n^K}^N \overline{W}_{n-1}^i = 1$ and denote $\overline{\mathbf{W}}_{n-1}$ these $N - 1$ weights.
- Sample $\mathbf{O}_{n-1} \backslash \left\{O_{n-1}^{B_n^K}\right\} \sim \mathcal{M}\left(N - O_{n-1}^{B_n^K}, \overline{\mathbf{W}}_{n-1}\right)$.
- Allocate randomly the associated parent indexes uniformly in $\{1, \ldots, N\} \backslash \left\{\text{indexes with parents equal to } B_{n-1}^K\right\}$.

This procedure follows directly from the fact that $\mathbf{O}_{n-1} \sim \mathcal{M}(N, \mathbf{W}_{n-1})$ so the marginal distribution of $O_{n-1}^{B_n^K}$ is $\mathcal{B}\left(N, W_{n-1}^{B_n^K}\right)$ and, conditional upon $O_{n-1}^{B_n^K}$, we have $\mathbf{O}_{n-1} \backslash \left\{O_{n-1}^{B_n^K}\right\} \sim \mathcal{M}\left(N - O_{n-1}^{B_n^K}, \overline{\mathbf{W}}_{n-1}\right)$. Finally conditional upon $O_{n-1}^{B_n^K} \geq 1$ we have $O_{n-1}^{B_n^K} \sim \mathcal{B}^{+}\left(N, W_{n-1}^{B_n^K}\right)$.

Note that an alternative to the PIMH algorithm to sample from $\pi(\mathbf{x})$ consists of alternating a conditional SMC step to update $\widehat{\pi}(\mathbf{x})$ and a step to sample $\left(\mathbf{X}_T^K, \mathbf{B}_T^K\right)$ from $\widehat{\pi}(\mathbf{x})$. For any $N \geq 1$, this algorithm admits $\pi(\mathbf{x})$ as invariant distribution as it is just a (collapsed) Gibbs sampler of invariant distribution $\widetilde{\pi}(k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1})$. Contrary to the PIMH, it is here necessary to have $N \geq 2$ to ensure irreducibility of this sampler.

3.4 Using All the Particles

The standard estimate of $\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ for M MCMC iterations is $\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{X}(m))$. A possible criticism of the PIMH is that in the implementation above we generate N particles at each iteration m of the MCMC algorithm to decide whether to accept or reject one single candidate. This might appear wasteful. However, it can be shown that the estimate

$$\frac{1}{M}\sum_{m=1}^{M}\left(\sum_{i=1}^{N} W_T^i(m)\, f(\mathbf{X}_T^i(m))\right)$$

converges also towards $\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ as $M \to \infty$ where $\{W_T^i(m), \mathbf{X}_T^i(m)\}$ corresponds to the set of normalized weights and particles used to compute $\hat{Z}(m)$. Following [8] it is also possible to propose an estimate which recycles the candidate populations of particles rejected by the PIMH; see [1] for details.

4 Particle Marginal MH Sampler and Particle Gibbs Sampler

We now consider the case where we are interested in sampling from a distribution

$$\pi(\theta, \mathbf{x}) = \frac{\gamma(\theta, \mathbf{x})}{Z}$$

with $\gamma : \Theta \times \mathcal{X}^T \to \mathbb{R}^+$ assumed known pointwise and Z a possibly unknown normalising constant, independent of $\theta \in \Theta$. For many statistical models of practical interest $\mathbf{x}$ can be high dimensional (*e.g.* a vector of latent variables of the size of a large dataset) and the conditional distribution $\pi(\mathbf{x}|\theta)$ is non-standard. We have

$$\pi(\mathbf{x}|\theta) = \frac{\gamma(\theta, \mathbf{x})}{\gamma(\theta)}, \quad \pi(\theta) = \frac{\gamma(\theta)}{Z}$$

where $\gamma(\theta) = \int_{\mathcal{X}^T} \gamma(\theta, \mathbf{x})\, d\mathbf{x}$ is typically unknown. We propose here two strategies to sample from $\pi(\theta, \mathbf{x})$. The first strategy consists of using a particle approximation of an MH algorithm updating simultaneously θ and $\mathbf{x}$. The second strategy consists of using a particle approximation of the Gibbs sampler sampling from $\pi(\mathbf{x}|\theta)$ and $\pi(\theta|\mathbf{x})$.

Both strategies will rely on the use of an SMC algorithm in order to propose approximate samples from $\pi(\mathbf{x}|\theta)$ and approximately compute its normalising constant $\gamma(\theta)$. Hence we need to consider a family of bridging distributions $\{\pi_n(\mathbf{x}_n|\theta); n = 1, \ldots, T-1\}$ where

$$\pi_n(\mathbf{x}_n|\theta) = \frac{\gamma_n(\theta, \mathbf{x}_n)}{Z_n^\theta} \tag{10}$$

and $\pi_T(\mathbf{x}_T|\theta) = \pi(\mathbf{x}|\theta)$ and a family of proposal distributions $\{q_n^\theta(x_n|\mathbf{x}_{n-1})\}$ that defines sampling of $x_n \in \mathcal{X}$ conditional upon $\mathbf{x}_{n-1} \in \mathcal{X}^{n-1}$ and θ. Note that $Z_T^\theta = \gamma(\theta)$.

4.1 Particle Marginal MH Sampler

Consider a MH algorithm of target distribution $\pi(\theta, \mathbf{x})$. Assume for the time being that sampling from $\pi(\mathbf{x}|\theta)$ for any $\theta \in \Theta$ is feasible and recall the standard decomposition $\pi(\theta, \mathbf{x}) = \pi(\theta)\pi(\mathbf{x}|\theta)$. In such situations it is natural to suggest the following form of proposal distribution for an MH update

$$q\left((\theta^*, \mathbf{x}^*)\middle|(\theta, \mathbf{x})\right) = q\left(\theta^*\middle|\theta\right)\pi\left(\mathbf{x}^*|\theta^*\right),$$

for which the proposed $\mathbf{x}^*$ is perfectly “adapted” to the proposed θ^*, and the only degree of freedom of the algorithm is $q(\theta^*|\theta)$, suggesting that the algorithm effectively targets the marginal distribution $\pi(\theta)$ as the MH acceptance ratio is given by

$$1 \wedge \frac{\pi(\theta^*, \mathbf{x}^*)}{\pi(\theta, \mathbf{x})} \frac{q((\theta, \mathbf{x})|(\theta^*, \mathbf{x}^*))}{q((\theta^*, \mathbf{x}^*)|(\theta, \mathbf{x}))} = 1 \wedge \frac{\gamma(\theta^*)}{\gamma(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}. \tag{11}$$

This algorithm is appealing since the difficult problem of sampling from $\pi(\theta, \mathbf{x})$ is reduced to that of sampling from $\pi(\theta)$ which is typically defined on a much smaller space and for which the design of proposal density is usually easier. Unfortunately, as discussed earlier, sampling exactly from $\pi(\mathbf{x}|\theta)$ is rarely feasible and $\gamma(\theta)$ is rarely known analytically, preventing the use of the above “idealized” Marginal MH (MMH) algorithm. It is natural to propose a Particle MMH (PMMH) algorithm which is a particle approximation of this “ideal” MMH algorithm using an SMC approximation of both samples from $\pi(\mathbf{x}|\theta)$ and of its normalising constant $\gamma(\theta)$. The PMMH algorithm proceeds as follows.

Particle Marginal Metropolis-Hastings Sampler

Initialization, $m = 0$

- Set randomly $\theta(0)$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta(0))$, sample $\mathbf{X}(0) \sim \widehat{\pi}(\cdot|\theta(0))$ and compute $\widehat{\gamma}(\theta(0))$.

At iteration $m \geq 1$

- Sample $\theta^* \sim q(\cdot|\theta(m-1))$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta^*)$, sample $\mathbf{X}^* \sim \widehat{\pi}(\cdot|\theta^*)$ and compute $\widehat{\gamma}(\theta^*)$.
- With probability
$$1 \wedge \frac{\widehat{\gamma}(\theta^*)}{\widehat{\gamma}(\theta(m-1))} \frac{q(\theta(m-1)|\theta^*)}{q(\theta^*|\theta(m-1))} \tag{12}$$
set $\theta(m) = \theta^*$, $\mathbf{X}(m) = \mathbf{X}^*$, $\widehat{\gamma}(\theta(m)) = \widehat{\gamma}(\theta^*)$, otherwise set $\theta(m) = \theta(m-1)$, $\mathbf{X}(m) = \mathbf{X}(m-1)$, $\widehat{\gamma}(\theta(m)) = \widehat{\gamma}(\theta(m-1))$.

Under very weak assumptions, the acceptance ratio (12) converges to (11) as $N \to \infty$. However more remarkably it can be established, using a reasoning very similar to that used for the PIMH algorithm, that this algorithm admits $\pi(\theta, \mathbf{x})$ as invariant distribution *for any* $N \geq 1$.

4.2 Particle Gibbs Sampler

A popular alternative to the MH algorithm to sample from $\pi(\theta, \mathbf{x})$ consists of using the Gibbs sampler. Numerous implementations rely on the fact that sampling from the conditional distribution $\pi(\theta|\mathbf{x})$ is feasible and thus the potentially tedious design of a proposal for θ can be bypassed. We will assume that this is the case here. Sampling from $\pi(\mathbf{x}|\theta)$ is typically impossible so we propose the following particle approximation.

Particle Gibbs Sampler

Initialization, $m = 0$

- Set randomly $\theta(0)$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta(0))$, sample $\mathbf{X}(0) \sim \hat{\pi}(\cdot|\theta(0))$ and denote $\mathbf{B}(0)$ its ancestral lineage.

At iteration $m \geq 1$

- Sample $\theta(m) \sim \pi(\cdot|\mathbf{X}(m-1))$.
- Run a conditional SMC algorithm for $\theta(m)$ consistent with $\mathbf{X}(m-1), \mathbf{B}(m-1)$, sample $\mathbf{X}(m) \sim \hat{\pi}(\cdot|\theta(m))$ and denote $\mathbf{B}(m)$ its ancestral lineage.

Under very weak assumptions, the interesting feature of this algorithm is that it admits $\pi(\theta, \mathbf{x})$ as invariant distribution *for any* $N \geq 1$. Contrary to the PIMH and the

PMMH algorithms, it is however necessary to have $N \geq 2$ to ensure irreducibility of the Particle Gibbs (PG) sampler.

5 Extensions and Discussion

For ease of presentation, we have limited our description to one of the simplest SMC algorithms. However numerous more sophisticated algorithms have been proposed in the literature over the past fifteen years to improve on such basic schemes. In particular, in many applications of SMC, the resampling step is only performed when the accuracy of the estimator is poor. Practically, this is assessed by looking at the variability of the weights using the so-called Effective Sample Size (ESS) criterion [11, pp. 35–36] given at time n by

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^i \right)^2 \right)^{-1}.$$

Its interpretation is that inference based on the N weighted samples is approximately equivalent to inference based on ESS perfect samples from the target. The ESS takes values between 1 and N and we resample only when it is below a threshold N_T otherwise we set $W_n^i \propto W_{n-1}^i \; w_n \left(\mathbf{X}_n^i \right)$. We refer to this procedure as dynamic resampling. All the strategies presented in the previous sections can also be applied in this context. The PIMH and PMMH can be implemented in the dynamic resampling context without any modification. However, the PG is more difficult to implement as the conditional SMC step requires simulating a set of $N-1$ particles not only consistent with a "frozen" path but also consistent with the resampling times of the SMC method used to generate the "frozen" path [1].

The PIMH algorithm presented in Section 3 is related to the Configurational-Biased Monte Carlo (CBMC) method which is a very popular method in molecular simulation used to sample long proteins [7]. Similarly to the PIMH sampler, the CBMC algorithm samples N particles and uses resampling steps. However, the resampling step used by the CBMC algorithm is such that a single particle survives, to which a new set of N offspring is then attached. Using our notation, this means that the CBMC algorithm corresponds to the case where $A_n^i = A_n^j$ for all $i, j = 1, \ldots, N$ and $A_n^1 \sim r(\cdot|\mathbf{W}_n)$ *i.e.* at any time n, all the children share the same and unique parent particle. The problem with this approach is that it is somewhat too greedy and that if a "wrong" decision is taken too prematurely then the proposal will be most likely rejected. It can be shown that the acceptance probability of the CBMC algorithm does not converge to 1 for $T > 1$ as $N \to \infty$ contrary to that of the PIMH algorithm. It has been more recently proposed in [3] to improve the CBMC algorithm by propagating forward several particles simultaneously in the spirit of the PIMH algorithm. However, contrary to us, the authors in [3] propose to kill or multiply particles by comparing their weights $w_n \left(\mathbf{X}_n^i \right)$ with respect to some prespecified lower and upper thresholds; *i.e.* the particles are not interacting and their number is a random variable. In simulations, they found that the performance of this algorithm was very sensitive to the values of these thresholds. Our approach has the great advantage of bypassing the delicate choice of such thresholds. In statistics, a variation of the CBMC algorithm known as the Multiple-Try Method (MTM) has been introduced in the specific case where $T = 1$ in [10]. The key of our methodology is to build efficient proposals using sequential and interacting mechanisms for cases where $T \gg 1$: the sequential structure might be natural for some models (e.g. state-space models) but can also be induced in other scenarios in order to take advantage of the potential improvement brought in by the interacting mechanism [5]. In this respect, both methods do not apply to the same class of problems.

6 Application to Markov Jump Processes

We consider here a discretely observed stochastic kinetic Lotka-Volterra (LV) model. This model is often used to describe biochemical networks which exhibit auto-regulatory behaviour; see [12] for a thorough description of these models and their applications to system biology. Having access to noisy biochemical data, our objective is to perform Bayesian inference for the kinetic rate constants of the LV models

The LV model describes the evolution of two species X_t^1 (prey) and X_t^2 (predator) which are continuous-time non-negative integer-valued processes. In a small time interval $(t, t+dt]$, there are three possible transitions for the Markov Jump Process (MJP) $X_t = \left(X_t^1, X_t^2\right)$

$$\begin{aligned}
&\Pr\left(X_{t+dt}^1 = x_t^1 + 1, X_{t+dt}^2 = x_t^2 \,\middle|\, x_t^1, x_t^2\right) = \alpha\, x_t^1 dt + o(dt),\\
&\Pr\left(X_{t+dt}^1 = x_t^1 - 1, X_{t+dt}^2 = x_t^2 + 1 \,\middle|\, x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2 dt + o(dt),\\
&\Pr\left(X_{t+dt}^1 = x_t^1, X_{t+dt}^2 = x_t^2 - 1 \,\middle|\, x_t^1, x_t^2\right) = \gamma\, x_t^2 dt + o(dt),
\end{aligned}$$

corresponding respectively to prey reproduction, predator reproduction and prey death, and predator death. We assume that we only have access to a noisy estimate of the number of preys $Y_n = X_{n\Delta}^1 + W_n$ with $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right)$. We are interested here in making inferences about the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ which are assumed to be a priori distributed as

$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5)$$

where $\mathcal{G}$ is the Gamma distribution [12, pp. 188–189]. The initial populations X_0^1, X_0^2 are assumed to be uniformly distributed in the interval $\{20, 21, \ldots, 80\}$.

We are interested in the posterior distribution $p(\mathbf{x}_T, \theta | \mathbf{y}_T)$ where $\mathbf{x}_T = \left(x_0, x_{2\Delta}, \ldots, x_{(T-1)\Delta}\right)$ and $\mathbf{y}_T = (y_0, y_1, \ldots, y_{T-1})$. This inference problem has already been addressed in [2]. In this paper, the authors propose a sophisticated reversible jump MCMC algorithm and a block updating strategy to sample from $p(\mathbf{x}_T, \theta | \mathbf{y}_T)$. The reversible jump MCMC is used to sample the continuous-time process X_t (and its unknown number of transitions) in the interval $[0,(T-1)\Delta]$ whereas the block updating strategy attempts to update X_t for $t \in [(k-1)\Delta, k\Delta]$ using a sensible proposal. The authors note that both "algorithms suffered significant mixing problems". We use here the PMMH algorithm with $\pi_n(\mathbf{x}_n|\theta) = p(\mathbf{x}_n|\mathbf{y}_n,\theta)$. For the SMC proposals, we simply use the prior of X_t from which it is easy to sample using Gillespie's algorithm [12, pp. 188–189]. For the parameters, we use a Gaussian random walk proposal whose parameters were estimated in a short preliminary run. We could have alternatively used an adaptive MCMC strategy. We generated $T = 50$ observations by simulating the MJP using Gillespie's algorithm with parameters $\alpha = 2$, $\beta = 0.05$, $\gamma = 1.5$, $\Delta = 0.2$, $\sigma^2 = 4$ and $X_0^1 = X_0^2 = 40$; see Figure 2. We ran the algorithms for 100,000 iterations with a burn-in of 20,000. For $N = 1000$, the average acceptance rate of the PMMH sampler was 36%. The results are displayed in Figure 3.

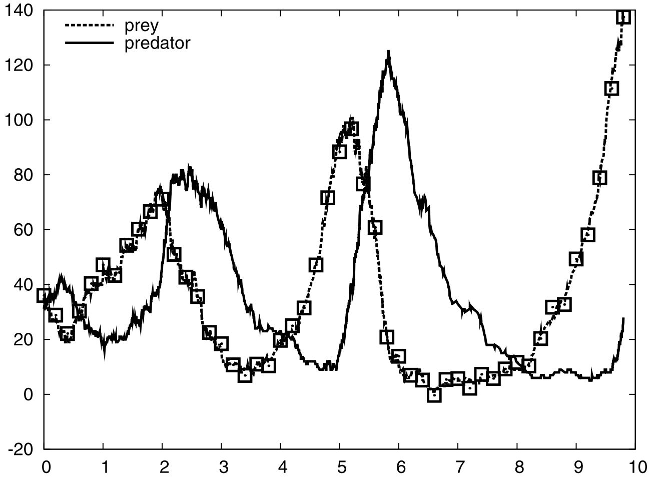


Fig. 2 Lotka-Volterra data. The number of prey X_t^1 and predators X_t^2 are shown in dotted and solid lines, respectively. The squares indicate the observations Y_n.

In Figure 4, we display the autocorrelation function (ACF) for the parameters (α,β) for various N. We can see that $N = 500$ is sufficient in this case for obtaining good performance and that increasing N does not improve the performance of the PMMH algorithm.

7 Conclusion

We have presented a new class of MCMC algorithms which rely on proposal distributions built using SMC methods. One of the major advantages of this approach is

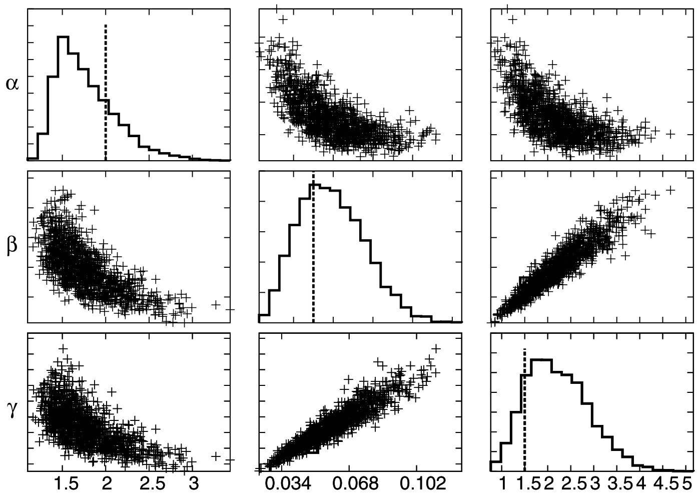
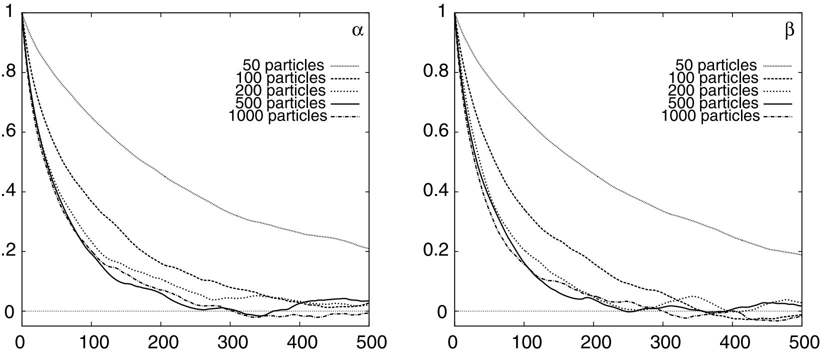


Fig. 3 Histograms and scatter plots of the sampled parameters. The straight lines on histograms represent the true values of the parameters.



(a) ACF for α

(b) ACF for β

Fig. 4 Autocorrelation of the parameter α (left) and β (right) for the PMMH sampler for various numbers N of particles.

that it systematically builds high-dimensional proposal distributions whilst requiring the practitioner to design only low-dimensional proposal distributions. It offers the possibility to simultaneously update large vectors of dependent random variables. The lower the variance of the SMC estimates of the normalising constants, the better the performance of these algorithms. This strategy is computationally expensive but to some extent unavoidable and useful in complex scenarios for which standard proposals are likely to fail.

We believe that many problems in statistics where SMC methods have already been used could benefit from PMCMC methods. We have already successfully used this methodology to fit complex continuous-time Lévy-driven stochastic volatility models and Dirichlet process mixtures [1]. Note that in the former case proposing samples from the prior distribution is the only known approach, which can lead to poor results when using standard MCMC algorithms. The CBMC method, to which our approach is related, is a very popular method in computational chemistry and physics which has been widely used for molecular and polymer simulation [7], and PMCMC algorithms might also prove useful in these areas.

References

1. Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society Series B, to appear.
2. Boys, R.J., Wilkinson, D.J., Kirkwood, T.B.L.: Bayesian inference for a discretely observed stochastic kinetic model. Statistics and Computing **18**, 125–135 (2008)
3. Combe, N., Vlugt, T.J.H., Wolde, P.R., Frenkel, D.: Dynamic pruned-enriched Rosenbluth method. Molecular Physics **101**, 1675–1682 (2003)
4. Del Moral, P.: Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer-Verlag, New York (2004)
5. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. Journal of the Royal Statistical Society Series B **68**, 411–436 (2006)
6. Doucet, A., Freitas, de J.F.G., Gordon, N.J (eds.): Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
7. Frenkel, D., Smit, B.: Understanding Molecular Simulation. 2nd edition, Academic Press, Orlando (2002)
8. Frenkel, D.: Waste-recycling Monte Carlo. In *Computer simulations in condensed matter: from materials to chemical biology*, Lecture Notes in Physics 703, Springer Berlin, 127–138 (2006)
9. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics **5**, 1–25 (1996)
10. Liu, J.S., Liang, F., Wong, W.H.: The use of multiple-try method and local optimization in Metropolis sampling. Journal of the American Statistical Association **95**, 121–134 (2000)
11. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer Verlag, New York (2001)
12. Wilkinson, D.J.: Stochastic Modelling for Systems Biology. Chapman & Hall/CRC Press, Boca Raton (2006)

Particle Markov chain Monte Carlo for Efficient Simulation - Erratum

The procedure described pages 52 and 53 after the conditional SMC algorithm to sample efficiently in the multinomial case in incorrect. Denoting $\mathcal{M}\left(a,\mathbf{b}\right)$ the multinomial distribution, the correct approach to sample $\mathbf{A}_{n-1}^{-B_n^K} \sim r(\cdot|\mathbf{W}_{n-1}, A_{n-1}^{B_n^K})$ proceeds as follows.

- Sample $\mathbf{O}_{n-1} \sim \mathcal{M}\left(N-1, \mathbf{W}_{n-1}\right)$ then set $O_{n-1}^{B_{n-1}^K} = O_{n-1}^{B_{n-1}^K} + 1$.
- Sample the indices of the $N-1$ 'free' offspring uniformly on the set $\{1,...,N\} \setminus \left\{B_{n-1}^K\right\}$.