ON MULTIPLE TRY SCHEMES AND THE PARTICLE METROPOLIS-HASTINGS ALGORITHM

L. Martino**, F. Leisen****, J. Corander**

* University of Helsinki, Helsinki (Finland).
** University of Kent, Canterbury (UK).

ABSTRACT

Markov Chain Monte Carlo (MCMC) methods are well-known Monte Carlo methodologies, widely used in different fields for statistical inference and stochastic optimization. The Multiple Try Metropolis (MTM) algorithm is an extension of the standard Metropolis-Hastings (MH) algorithm in which the next state of the chain is chosen among a set of candidates, according to certain weights. The Particle MH (PMH) algorithm is other advanced MCMC technique specifically designed for scenarios where the multidimensional target density can be easily factorized as multiplication of (lower - dimensional) conditional densities. Both have been widely studied and applied in literature. In this note, we investigate similarities and differences among the MTM schemes and the PMH method. Furthermore, novel schemes are also designed.

Keywords: Markov chain Monte Carlo; Multiple Try Metropolis; Particle MCMC.

1. INTRODUCTION

Monte Carlo statistical methods are powerful tools for numerical inference and stochastic optimization [19]. Markov chain Monte Carlo (MCMC) methods are classical Monte Carlo techniques that generate samples from a target probability density function (pdf) by drawing from a simpler proposed pdf, usually to approximate an otherwise-incalculable (analytically) integral [10, 9]. MCMC algorithms produce a Markov chain with a stationary distribution that coincides with the target pdf.

The Metropolis-Hastings (MH) algorithm [15, 7] is the most famous MCMC technique. It can be applied to almost any target distribution but, in practice, the performance dependence crucially on the choice of the proposal pdf. In some cases, the Markov chain generated by the MH algorithm can remain trapped almost indefinitely in a local mode meaning that, in practice, convergence can be very slow.

The *Multiple-Try Metropolis* (MTM) method of [11], [10, Chapter 5] is an extension of the MH algorithm in which the next state of the chain is selected among a *set* of independent and identically distributed (i.i.d.) samples. This enables the MCMC sampler to make large step-size jumps without a lowering the acceptance rate; and thus MTM is can explore a larger portion of the sample space in fewer iterations. A famous special case of MTM, well-known in molecular simulation field, is the *orientational bias Monte Carlo* technique [6].

Several generalizations of the basic MTM scheme [11] can be found in literature: with correlated candidates [13, 18], more general form of the weights and different frameworks [8, 14, 16, 22], with adaptive and interacting proposal pdfs [2, 12]. Interesting and related considerations about the use of multiple auxiliary variables for building acceptance probabilities within a MH approach can be found in [21].

This work has been supported by the AoF grant 251170, and by the European Union 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861.

Independently from the MTM schemes, *Particle MCMC methods* have been proposed [1] in literature. They are specially designed to solve inference problem in state space model applications. Here, we focus on the so-called *Particle (Independent) Metropolis-Hastings* (PMH) algorithm.

The authors in [1] also provide a short description of the relationships with other existing techniques. They mention and describe precisely the relationship with the so-called *configurational bias Monte Carlo* method [20],[10, Chapter 5], technique that is also strictly connected to the MTM scheme. They also allude quickly to the MTM method [11].

The relationship between MTM and PMH deserves a more careful look. This the aim of this work. We introduce a slight variant of a MTM technique with an independent proposal density. The structure of this algorithm coincides exactly with the PMH method although the mechanism of generation of the candidates is different, as discussed in the sequel. Clarifying this strong connection allows us to design new efficient schemes.

[? ?]

2. IMPORTANCE SAMPLING

Many applications can be described a system characterized by a vector of unknown parameters, $\mathbf{x} \in \mathbb{R}^{D\times\zeta}$, and a set of observed data, $\mathbf{y} \in \mathbb{R}^{d_Y}$. In these cases, one is interested in approximating the posterior density $\bar{\pi}(\mathbf{x}|\mathbf{y})$ that, hereafter, we simply denote as $\bar{\pi}(\mathbf{x})$. More specifically, in this work, we denote the variable of interest as

$$\mathbf{x} = x_{1:D} = [x_1, x_2, \dots, x_D] \in \mathcal{D} = \mathcal{X}^D \subseteq \mathbb{R}^{D\times\zeta},$$

where $x_d \in \mathcal{X} \subseteq \mathbb{R}^\zeta$ for all $d = 1, \dots, D$. The target density is indicated as $\bar{\pi}(\mathbf{x}) = \frac{1}{Z_D}\pi(\mathbf{x})$, where

$$Z_D = \int_{\mathcal{D}} \pi(\mathbf{x}) d\mathbf{x}. \tag{1}$$

Monte Carlo techniques employ a proposal density, denoted as $q(\mathbf{x})$, with support $\mathcal{X} \subseteq \mathbb{R}^{D\times\zeta}$,[1] for generating possible candidates. Then, these candidates are filtered using some suitable procedure, in order to produce a particle approximation of $\bar{\pi}(\mathbf{x})$ and provide an estimation of Z_D.

2.1. Batch and Sequential Importance Sampling

A well-known Monte Carlo algorithm is the importance sampling (IS) method. IS provides an approximation with weighted samples of the measure of π. More specifically, N samples $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}$ are drawn from a proposal pdf $q(\mathbf{x})$ and then they are weighted as

$$w_D^{(n)} = \frac{\pi(\mathbf{x}^{(n)})}{q(\mathbf{x}^{(n)})}, \quad n = 1, \dots, N, \tag{2}$$

where the super-index n in $w_D^{(n)}$ denotes the corresponding particle and the subindex D refers to $\mathbf{x} = x_{1:D} = [x_1, ..., x_D]$. Thus, the particle approximation is

$$\widehat{\pi}_D(\mathbf{x}) = \sum_{n=1}^{N} \bar{w}_D^{(n)} \delta(\mathbf{x} - \mathbf{x}^{(n)}), \tag{3}$$

with $\bar{w}_D^{(n)} = \frac{w_D^{(n)}}{\sum_{i=1}^N w_D^{(i)}}$. An estimation of Z is given by

$$\widehat{Z}_D = \frac{1}{N} \sum_{n=1}^{N} w_D^{(n)}. \tag{4}$$

[1] For the sake of simplicity, in the observations of the rest of the work, we consider the proposal function $q(\mathbf{x})$ be normalized, i.e., $\int_{\mathcal{X}} q(\mathbf{x}) d\mathbf{x} = 1$.

In high dimensional spaces $\mathcal{D}$, an equivalent sequential procedure is preferred to the previous batch approach. Recall that $\mathbf{x} = x_{1:D} = [x_1, ..., x_D]$, we can observe that a target pdf $\bar{\pi}(\mathbf{x})$ can always be expressed as

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) \quad = \quad \gamma_1(x_1) \prod_{d=2}^{D} \gamma_d(x_d | x_{1:d-1}) \tag{5}$$

using the chain rule [17] where $\gamma_1(x_1)$is a marginal pdf and $\gamma_d(x_d|x_{1:d-1})$ are conditional pdfs. We also consider the joint probability of $[x_1, \ldots, x_d]$,

$$\bar{\pi}_d(x_{1:d}) = \frac{1}{Z_d} \pi_d(x_{1:d}) \propto \pi_d(x_{1:d}) = \gamma_1(x_1) \prod_{j=2}^{d} \gamma_j(x_j | x_{1:j-1}), \tag{6}$$

so that, clearly, $\bar{\pi}_D(x_{1:D}) = \bar{\pi}(\mathbf{x})$. In many applications, the target appears directly decomposed as in Eq. (5), e.g., as in state-space models. However, in general, one needs to marginalize several times the target $\bar{\pi}(\mathbf{x})$ for obtaining analytically the conditional pdfs $\gamma_d(x_d|x_{1:d-1})$, $d = 1, \ldots, D$. Given the target in Eq. (5), we can also consider a proposal pdf decomposed in the same fashion

$$q(\mathbf{x}) = q_1(x_1) q_2(x_2|x_1) \cdots q_{D-1}(x_{D-1}|x_{1:D-2}) q_D(x_D|x_{1:D-1}).$$

In a batch IS scheme, given an n-th sample $\mathbf{x}^{(n)} = x_{1:D}^{(n)} \sim q(\mathbf{x})$, we assign the importance weight

$$w_D^{(n)} = \frac{\pi(\mathbf{x}^{(n)})}{q(\mathbf{x}^{(n)})} = \frac{\gamma_1(x_1^{(n)}) \gamma_2(x_2^{(n)}|x_1^{(n)}) \cdots \gamma_D(x_D^{(n)}|x_{1:D-1}^{(n)})}{q_1(x_1^{(n)}) q_2(x_2^{(n)}|x_1^{(n)}) \cdots q_D(x_D^{(n)}|x_{1:D-1}^{(n)})}.$$

The previous expression suggests a recursive procedure for computing the importance weights: starting with $w_1^{(n)} = \frac{\pi(x_1^{(n)})}{q(x_1^{(n)})}$ and then

$$\begin{aligned} w_d^{(n)} &= w_{d-1}^{(n)} \beta_d^{(n)}, \\ &= \prod_{j=1}^{d} \beta_j^{(n)}, \qquad d = 1, \ldots, D, \end{aligned} \tag{7}$$

where we have set

$$\beta_1^{(n)} = w_1^{(n)} \quad \text{and} \quad \beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{1:d-1}^{(n)})}{q_d(x_d^{(n)}|x_{1:d-1}^{(n)})}, \tag{8}$$

for $d = 2, \ldots, D$. Thus, given N samples $\mathbf{x}^{(1)}$,. . ., $\mathbf{x}^{(N)}$, finally we obtain the particle approximation

$$\widehat{\pi}_d(x_{1:d}) = \sum_{n=1}^{N} \bar{w}_d^{(n)} \delta(x_{1:d} - x_{1:d}^{(n)}), \quad d = 1, \ldots, D, \tag{9}$$

and an estimator of the normalizing constant Z_d is

$$\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N} \sum_{n=1}^{N} \left[\prod_{j=1}^{d} \beta_j^{(n)} \right]. \tag{10}$$

However, an alternative formulation is often used

$$\widetilde{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right], \tag{11}$$

$$= \prod_{j=1}^{d} \left[\frac{\sum_{n=1}^{N} w_j^{(n)}}{\sum_{n=1}^{N} w_{j-1}^{(n)}} \right], \tag{12}$$

$$= \prod_{j=1}^{d} \left[\frac{\widehat{Z}_j}{\widehat{Z}_{j-1}} \right] = \frac{\widehat{Z}_1}{\widehat{Z}_0} \frac{\widehat{Z}_2}{\widehat{Z}_1} \times \ldots \times \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} = \widehat{Z}_d, \tag{13}$$

where, for simplicity, we have set $\widehat{Z}_0 = 1$. A alternative derivation of the (final) estimator $\widetilde{Z}_D$ is given in Appendix A.

Remark 1. *In SIS, there are two equivalent formulations, $\widehat{Z}_d$ in Eq.* (4) *and $\widetilde{Z}_d$ in Eq.* (11) *of estimator of Z_d.*

In the rest of the work, sometimes we denote the final estimators $\widehat{Z}_D$ or $\widetilde{Z}_D$ simply as $\widehat{Z}$ or $\widehat{Z}$, as well.

2.2. Sequential Importance Resampling (SIR)

Sequential Importance Resampling (SIR) [10, 19] combines the sequential construction of the importance weights with the application of *resampling* steps [4, 5]. Namely, when some pre-established criterion is fulfilled, N independent particles are drawn according to the probability mass $\widehat{\pi}_d(x_{1:d})$. Then, the resampled particles are propagated for providing the next approximation $\widehat{\pi}_{d+1}(x_{1:d+1})$. More specifically, let us consider that a resampling step is performed at the d-th iteration. Hence, N samples $x_{1:d}^{(j)}$ are drawn from $\widehat{\pi}_d(x_{1:d})$, and then the corresponding weights are set to the same value [4, 5]. A proper choice [2] is to set the unnormalized importance weights

$$w_d^{(n)} = \widehat{Z}_d, \quad \forall j = 1, \ldots, N. \tag{14}$$

i.e., $w_d^{(1)} = w_d^{(2)} = \ldots = w_d^{(N)}$, equal for each resampled particle $x_{1:d}^{(n)}$. Hence, after a resampling step, we have that $\bar{w}_d(x_{1:d}^{(n)}) = \frac{1}{N}$, for all $j = 1, \ldots, N$. One reason why this is a good choice, for instance, is that defining the following weights

$$\xi_d^{(n)}) = \begin{cases} w_d^{(n)}, & \text{without resampling at } d\text{-th iteration,} \\ \widehat{Z}_d, & \text{with resampling at } d\text{-th iteration.} \end{cases} \tag{15}$$

then, in any case, $\frac{1}{N} \sum_{n=1}^{N} \xi_d^{(n)} = \widehat{Z}_d$, as expected. Therefore, the weight recursion for SIR becomes

$$\xi_d^{(n)} = \xi_{d-1}^{(n)} \beta_d^{(n)}, \quad \text{where} \quad \xi_{d-1}^{(n)} = \begin{cases} \xi_{d-1}^{(n)}, & \text{without res. at } (d-1)\text{-th iter.,} \\ \widehat{Z}_{d-1}, & \text{with res. at } (d-1)\text{-th iter.} \end{cases} \tag{16}$$

See Appendix A for further details.

Remark 2. *With the recursive definition of the weights $\xi_d^{(n)}$ in Eq.* (16)*, the two estimators*

$$\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} \xi_{d-1}^{(n)} \beta_d^{(n)}, \quad \widetilde{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{\xi}_{j-1}^{(n)} \beta_j^{(n)} \right] \tag{17}$$

[2] This is a proper choice, but it is not unique; see "*Concept of weighted sample*" in [10, Chapter 2] or similarly [19, Section 14.2].

where $\bar{\xi}_{j-1}^{(n)} = \frac{\xi_{j-1}^{(n)}}{\sum_{i=1}^{N} \xi_{j-1}^{(i)}}$, are both valid and equivalent estimators of Z_d. Furthermore, if the resampling is applied at each iteration, observe that they become

$$\widetilde{Z}_d = \prod_{j=1}^{d} \left[\frac{1}{N} \sum_{n=1}^{N} \beta_j^{(n)} \right], \tag{18}$$

and

$$\widehat{Z}_d = \widehat{Z}_{d-1} \left[\frac{1}{N} \sum_{n=1}^{N} \beta_d^{(n)} \right] = \prod_{j=1}^{d} \left[\frac{1}{N} \sum_{n=1}^{N} \beta_j^{(n)} \right], \tag{19}$$

and clearly coincide. Note that, w.r.t. the estimator in Eq. (10) *(for SIS, i.e., without resampling), the operations of product and sum are inverted.*

Figure 2 depicts different examples of generation of weighted samples $\mathbf{x}^{(n)}$ with or without employing resampling steps. More specifically, Figure 2 shows the components $x_1,^{(n)} \ldots, x_D^{(n)}$ of each sample, with $D = 10$.

3. MULTIPLE TRY METROPOLIS (MTM) SCHEMES

The Multiple Try Metropolis (MTM) algorithm [11] is an MCMC technique, where N candidates are generated each iterations. According to some suitable weights, one candidate is chosen and accepted as new state with a certain probability α. The MTM steps with a generic proposal $q(\mathbf{x}|\mathbf{x}_{k-1})$, depending on the previous state, are summarized in Table 1. For $N = 1$, the MTM algorithm becomes the standard Metropolis-Hastings (MH) method. We consider importance weights for faciliting the comparison with the other techniques. However, different kind of weights could be applied [11, 14].We have denoted $a \wedge b = \min[a, b]$. The MTM method generates a reversible chain that converges to $\bar{\pi}(\mathbf{x})$ [11, 14].

If the proposal pdf is independent from the previous state of the chain, i.e., $q(\mathbf{x})$, the algorithm can be simplified. indeed, the steps 2c and 2d can be removed in the MTM scheme. Namely, one does not need to generate the reference samples at step 2c. Indeed, in this case, we could directly set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \ldots, N-1$. The simplified algorithm (I-MTM) is given in Table 1. A graphical representation of a MTM scheme is provided in Figure 1, with $D = 1$ and $N = 2$.

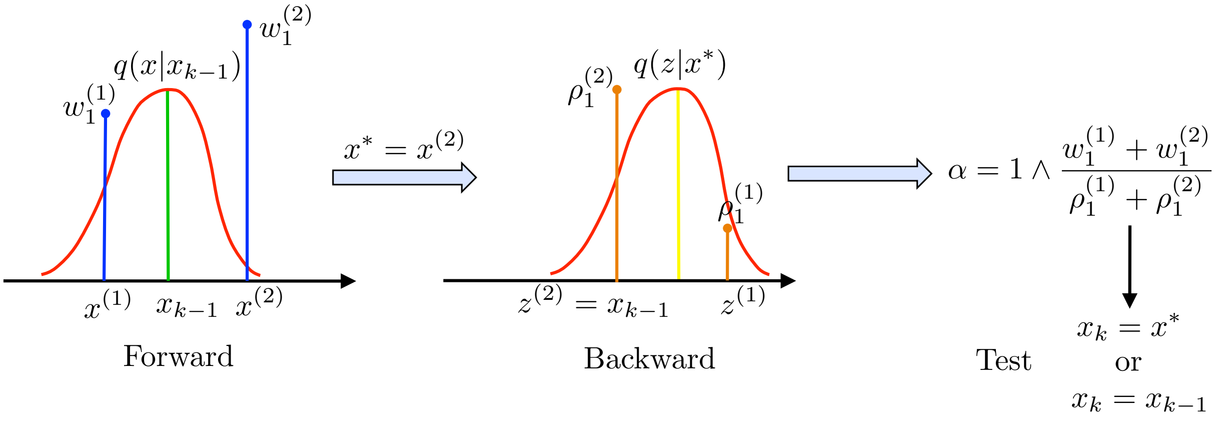


Fig. 1. Sketch of a generic MTM method with $D = 1$ and $N = 2$ tries. In this example, the second candidate is selected as $x^* = x^{(2)}$. It has been selected with probability $\bar{w}_1^{(1)} = \frac{w_1^{(1)}}{w_1^{(1)} + w_1^{(2)}}$. The reference points are $z^{(1)} \sim q(z|x^*)$ and $z^{(2)} = x_{k-1}$.

Table 1. General MTM algorithm.

```
1. Choose a initial state $\mathbf{x}_0$ and the total number of iterations $K$.
2. For $k = 1, \dots, K$:
    (a) Draw $N$ samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x}|\mathbf{x}_{k-1})$, $i = 1, \dots, N$.
    (b) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights
            $w_D^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)}|\mathbf{x}_{k-1})}, \quad i = 1, \dots, N.$
        Namely, draw a sample $\mathbf{x}^*$ from
            $\widehat{\pi}_D(\mathbf{x}) = \sum_{n=1}^{N} \bar{w}_D^{(n)} \delta(\mathbf{x} - \mathbf{x}^{(n)}).$
    (c) Draw $N-1$ "reference" samples $\mathbf{z}^{(j)} \sim q(\mathbf{x}|\mathbf{x}^*)$, $j = 1, \dots, N-1$, and set $\mathbf{z}^{(N)} = \mathbf{x}_{k-1}$.
    (d) Compute the importance weights also for the reference points,
            $\rho_D^{(i)} = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)}|\mathbf{x}^*)}, \quad i = 1, \dots, N.$
    (e) Set $\mathbf{x}_k = \mathbf{x}^*$ with probability
            $\alpha = 1 \quad \wedge \quad \frac{\sum_{i=1}^{N} w_D^{(i)}}{\sum_{i=1}^{N} \rho_D^{(i)}},$
        otherwise, with probability $1 - \alpha$, set $\mathbf{x}_k = \mathbf{x}_{k-1}$.
```

Table 2. MTM with independent proposal (I-MTM).

```
1. Choose a initial state $\mathbf{x}_0$ and the total number of iterations $K$.
2. For $k = 1, \dots, K$:
    (a) Draw $N$ samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x})$, $i = 1, \dots, N$.
    (b) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights
            $w_D^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \quad i = 1, \dots, N.$
        Moreover, we denote as $w_D^*$ and $w_{D,k-1}$ are the weights corresponding to $\mathbf{x}^*$ and $\mathbf{x}_{k-1}$, respectively.
    (c) Set $\mathbf{x}_k = \mathbf{x}^*$ with probability
            $\alpha = 1 \quad \wedge \quad \frac{\sum_{i=1}^{N} w_D^{(i)}}{\sum_{i=1}^{N} w_D^{(i)} - w^* + w_{D,k-1}} = 1 \quad \wedge \quad \frac{\sum_{i=1}^{N} w_D^{(i)}}{\sum_{i=1}^{N} \rho_D^{(i)}},   (20)$
        where the values $\rho_D^{(i)}$ denote the importance weights of $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}\} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\} \setminus \{\mathbf{x}^*\} \cup \{\mathbf{x}_{k-1}\}$.
        Otherwise, set $\mathbf{x}_k = \mathbf{x}_{k-1}$.
```

Alternative version of the I-MTM method (I-MTM2). In this work, we highlight that the I-MTM method can be designed in an alternative way. With a proposal pdf independent from the previous state, we have seen that we can set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \dots, N-1$, because each $\mathbf{x}^{(j)}$ is itself drawn from $q(\mathbf{x})$. With the same arguments, we can also use the samples generated in the previous iteration of the algorithm as reference points, since all the samples are generated independently from the same proposal pdf. Namely, the alternative version of the I-MTM is summarized

Table 3. Alternative I-MTM algorithm (I-MTM2).

1. Choose a initial state $\mathbf{x}_0$, the total number of iterations K and obtain an estimation $\widehat{Z}^{(0)} \approx Z$.
2. For $k = 1, \dots, K$:
   (a) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights
   $$w_D^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \quad i = 1, \dots, N.$$
   (b) Set $\mathbf{x}_k = \mathbf{x}^*$ and $\widehat{Z}^{(k)} = \widehat{Z}^* = \frac{1}{N}\sum_{i=1}^N w_D^{(i)}$ with probability
   $$\alpha = 1 \quad \wedge \quad \frac{\frac{1}{N}\sum_{i=1}^N w_D^{(i)}}{\widehat{Z}^{(k-1)}} = 1 \quad \wedge \quad \frac{\widehat{Z}_D^*}{\widehat{Z}_D^{(k-1)}}$$
   otherwise, with probability $1 - \alpha$, set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widehat{Z}^{(k)} = \widehat{Z}^{(k-1)}$.

in Table 3.

4. PARTICLE MH ALGORITHM AND ITS RELATIONSHIP WITH THE MTM SCHEMES

Consider a target density factorized as

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) = \gamma_1(x_1)\gamma_2(x_2|x_1)\cdots\gamma_D(x_D|x_{1:D-1}).$$

The Particle Metropolis Hastings (PMH) method [1] is another MCMC technique proposed independently from the MTM algorithm, specifically designed for being applied in this framework. The complete description is provided in Table 4. Both estimators $\widehat{Z}$ and $\widetilde{Z}$ can be used in PMH (although the original algorithm employed $\widetilde{Z}$), if the resampled particles are properly weighted as shown in Eq. (14). A generalization of PMH for handling both dynamic and fixed parameters, called *Particle Marginal MH* algorithm, is described in Appendix B.

4.1. Relationship between MTM and PMH

A simple look at the alternative version of the MTM technique with independent proposal (I-MTM2), introduced in Section 3, and the PMH method, shows that are strictly related. Indeed, the structure of the two algorithms coincides. The links and differences are listed below:

- The main difference lies that the candidates in PMH are generated sequentially, using a SMC procedure. If the resampling steps in the SMC are not applied them I-MTM2 and PMH are *exactly* the same algorithm, where the candidates are drawn in a *batch* setting or *sequential* way. Namely, I-MTM2 generates directly $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_D^{(i)}]$ from $q(\mathbf{x})$ whereas PMH draws sequentially each component $x_d^{(i)}$ from $q_d(x_d|x_{1:d-1}^{(i)})$.
- Hence, the resampling steps is the real difference between the generation procedures of PMH and I-MTM2. Owing to the use of the resampling, the candidates $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ proposed by PMH are not independent, differently from the MTM schemes considered in this work. Without resampling, the generated samples $\mathbf{x}^{(i)} = x_{1:D}^{(i)}$ would be independent as in I-MTM2. The generation of correlated samples can be also considered in MTM methods, as simply shown for instance in [3]), without jeopardizing the ergodicity of the chain. Thus, more precisely, PMH can be considered as an I-MTM2 scheme using correlated samples (e.g., as in [3]), and where the candidates are generated sequentially.

Table 4. **Particle Metropolis-Hastings (PMH) algorithm.**

1. Choose a initial state $\mathbf{x}_0$, the total number of iterations K and obtain an estimation $\widehat{Z}^{(0)} \approx Z$.
2. For $k = 1, \dots, K$:
    (a) Using a proposal pdf of type
    $$q(\mathbf{x}) = q_1(x_1)q_2(x_2|x_1)\cdots q_D(x_D|x_{1:D-1}),$$
    we employ SIR (see Section 2.2) for drawing with N particles and weighting properly them, $\{\mathbf{x}^{(i)}, w_D^{(i)}\}_{i=1}^N$. Namely, we obtain a particle approximation of the measure of target pdf
    $$\widehat{\pi}_D(\mathbf{x}) = \sum_{i=1}^N \bar{w}_D^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)}).$$
    Furthermore, we also obtain $\widehat{Z}^*$ in Eq. (10), or $\widetilde{Z}^*$ in Eq. (11).
    (b) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, ..., N$.
    (c) Set $\mathbf{x}_k = \mathbf{x}^*$ and $\widehat{Z}^{(k)} = \widehat{Z}^*$ with probability
    $$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}^*}{\widehat{Z}^{(k-1)}}, \tag{21}$$
    otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widehat{Z}^{(k)} = \widehat{Z}^{(k-1)}$.

For clarifying this point, in Figure 2 we show different particles weighted with IS weights (the line-width of each path is proportional to the corresponding normalized weight $\bar{w}_n$). More specifically, we represent each component of $x_d^{(n)}$, $d = 1, \dots, D = 10$ of each particle $\mathbf{x}^{(n)} = x_{1:10}^{(n)}$ with $n = 1, \dots, N \in \{5, 40\}$. The target density is a multivariate Gaussian pdf, $\bar{\pi}(\mathbf{x}) = \prod_{d=1}^{10} \mathcal{N}(x_d|2, \frac{1}{2})$, i.e., with expected value $\mu_d = 2$, for $d = 1, \dots, 10$. Figures 2(a)-(b) corresponds to the application of IS with two different proposal pdfs and without resampling. In Figure 2(a), the components $x_d^{(n)}$ are independent. In Figure 2(b), the components $x_d^{(n)}$ within each sample $\mathbf{x}^{(n)}$ are correlated, but the samples $\mathbf{x}^{(n)}$, $n = 1, \dots, N$, are still independent. In Figure 2(c) two resampling are also applied at the iterations $d = 4, 8$, generating correlation among the particles $\mathbf{x}^{(n)}$, $n = 1, \dots, N$, as well. Figure 2(c) corresponds to the sample generation in PMH.

- In their standard formulations, I-MTM2 uses the estimator $\widehat{Z}_D$ in Eq. (4) whereas PMH has been proposed using $\widetilde{Z}_D$, given in Eq. (11). However, they are equivalent formulation of an estimator of the normalizing constant Z_D.

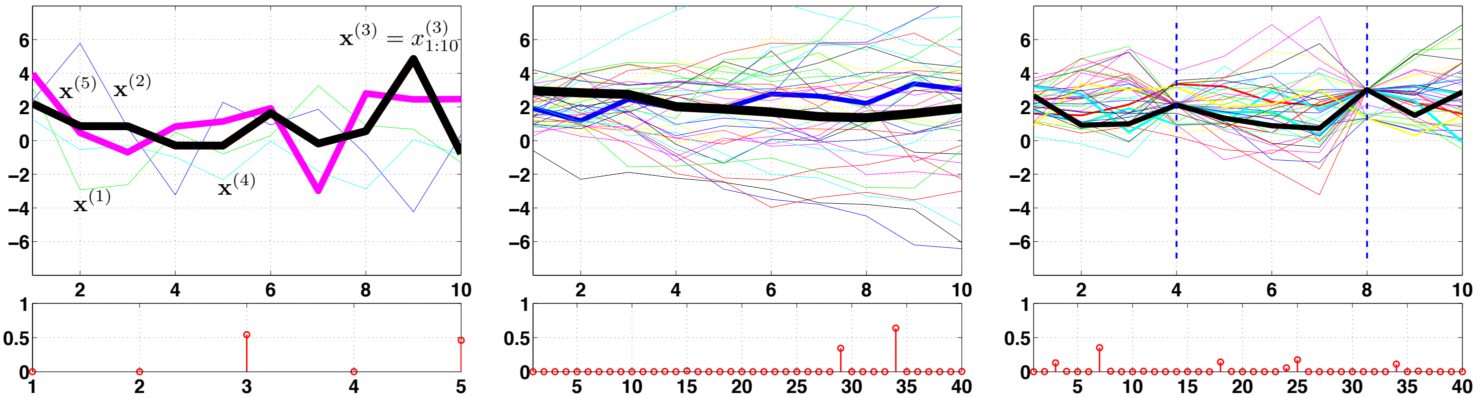
- The PMH approach is preferable in high dimension, when the target can be factorized, since the use of the resampling steps provides a better proposal generation procedure.

4.2. Novel PMH algorithms

The previous considerations allow us to design novel PMH schemes. For instance, we can easily suggest an alternative proper acceptance probability function,

$$\alpha = 1 \quad \wedge \quad \frac{N\widehat{Z}^*}{N\widehat{Z}^* - w_D^* + w_{D,k-1}}. \tag{22}$$

We denote as PMH-2 the PMH technique which uses the probability α above, instead of the probability α in Eq. (21). Namely, PMH-2 is identical with the PMH method in Table 4, replacing Eq. (21) with Eq. (22). The PMH-2



(a) Batch-IS or SIS with $q_d(x_d) = \mathcal{N}(x_d|0, \sqrt{2})$ and $N = 5$.

(b) Batch-IS or SIS with $q_d(x_d|x_{d-1})$ and $N = 40$.

(c) SIR using $q_d(x_d|x_{d-1})$ and resampling at the iterations $d = 4, 8$ (with $N = 40$).

Fig. 2. Examples of application of the IS technique. We consider as target density a multivariate Gaussian pdf, $\bar{\pi}(\mathbf{x}) = \prod_{d=1}^{10} \mathcal{N}(x_d|2, \frac{1}{2})$. In each figure, every component of different particles are represented, so that each particle $\mathbf{x}^{(i)}$ seems to form a *path*. The normalized weights $\bar{w}_n$ corresponding to each figure are also shown. The line-width of each path is proportional to the corresponding weight $\bar{w}_n$. The particle corresponding to the greatest weight is always depicted in black. **(a)** Batch IS or SIS with $N = 5$ particles and $q(\mathbf{x}) = \prod_{d=1}^{d} \mathcal{N}(x_d|0, \sqrt{2})$. **(b)** Batch IS or SIS with $N = 40$ particles and $q(\mathbf{x}) = \mathcal{N}(x_1|2, 1) \prod_{d=2}^{d} \mathcal{N}(x_d|x_{d-1}, 1)$. **(c)** SIR with $N = 40$ particles and $q(\mathbf{x}) = \mathcal{N}(x_1|2, 1) \prod_{d=2}^{d} \mathcal{N}(x_d|x_{d-1}, 1)$ and resampling steps at the iterations $d = 4, 8$.

structure is equivalent (within a sequential framework) to I-MTM of Table 2, in the same fashion as PMH in Table 4 is equivalent to I-MTM2 of Table 3.

Moreover, we can also extend the standard PMH method employing a state-dependent proposal pdf (dependent from the previous state), instead of an independent proposal (namely, independent from the previous state) as in Table 4. This novel scheme, denoted as SD-PMH, is outlined in Table 5 where a SIR. In this case, the generation of a backward path is required at step 2c. Hence, in SD-PMH, we have this additional computation cost. However, the generated backward paths could be also recycled for estimating the hidden states (nevertheless, this requires and deserves more specific analysis). The validity of SD-PMH is ensured since it corresponds to the MTM scheme in Table 1. In SD-PMH, the approximation $\widehat{\pi}_D$ is provided considering with correlated samples due to the resampling, unlike in MTM. However, it does not jeopardize the ergodicity (as shown, e.g., in [3]). Furthermore, we consider the use of resampling steps only at certain $0 \leq R \leq K$ pre-established iterations, $d_1, \ldots, d_R$. If $R = 0$, no resampling it is applied so that we obtain a standard MTM scheme. If $R = K$, the resampling is applied at each iteration, so that we have a bootstrap filter for generating the samples [**?**]. Figure 3 shows a sketch of the different schemes discussed in this work. The MTM schemes are given on the left side, whereas the corresponding PMH approaches are provided on the right. The boxes with dashed contours represent the novel schemes introduced in this work.

Random Walk PMH. As an example of SD-PMH scheme, we can consider a PMH employing with a random walk proposal pdf (RW-PMH). For simplicity, first we consider

$$q_d(s_d|s_{1:d-1}, x_{1:d,k-1}) = q_d(s_d|s_{d-1}, x_{d,k-1}),$$

so that the complete proposal is $q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1}) \prod_{d=2}^{D} q_d(s_d|s_{d-1}, x_{d,k-1})$. Secondly, we can set

$$q_d(s_d|s_{d-1}, x_{d,t-1}) = \frac{1}{2}\mathcal{E}_1\left(s_d|s_{d-1}, \Sigma_1\right) + \frac{1}{2}\mathcal{E}_2\left(s_d|x_{d,k-1}, \Sigma_2\right) \tag{24}$$

where $\mathcal{E}_i\left(x|\mu_i, \Sigma_i\right) : \mathcal{X} \to \mathbb{R}$, $i = 1, 2$, represent generic densities with expected values $\mu_1 = s_{d-1}$, $\mu_2 = x_{d,k-1}$,

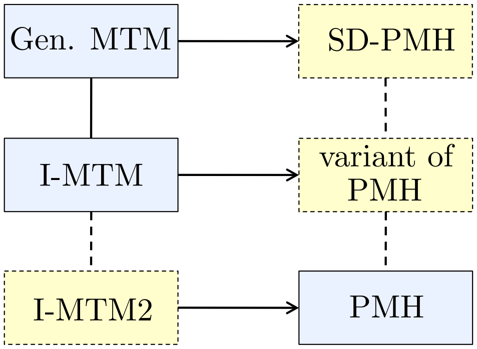


Fig. 3. Graphical representation of the MTM methods and the corresponding PMH schemes. The boxes with dashed contours contain the novel schemes presented in this work.

Table 5. **State Dependent PMH (SD-PMH)**

1. Choose a initial state $\mathbf{x}_0$, the total number of iterations K.
2. For $k = 1, \dots, K$:
   - (a) Using a proposal pdf of type
   $$q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1})q_2(s_2|s_1, x_{1:2,k-1}) \cdots q_D(s_D|s_{1:D-1}, x_{1:D,k-1}), \qquad (23)$$
   we employ SIR (see Section 2.2) for drawing with N particles, $\mathbf{x}^{(i)}$, and weighting properly them, $\{\mathbf{x}^{(i)}, w_D^{(i)}\}_{i=1}^N$. The resampling steps are applied at R fixed and pre-established iterations ($0 \leq R \leq K$),
   $$d_1 < d_2 < \dots < d_R.$$
   Thus, we obtain a particle approximation of the measure of target pdf
   $$\widehat{\pi}_D(\mathbf{x}) = \sum_{i=1}^N \bar{w}_D^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)}).$$
   Furthermore, we also obtain $\widehat{Z}_X$ in Eq. (10) or $\widetilde{Z}_X$ as in Eq. (11).
   - (b) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, ..., N$.
   - (c) Draw $N-1$ particles $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N-1)}$ via SIR using $q(\mathbf{z}|\mathbf{x}^*)$ as in Eq. (23), applying resampling at the same iterations, $d_1 < d_2 < \dots < d_R$, used in the generation of $\mathbf{x}^{(i)}$'s. Moreover, set $\mathbf{z}_N = \mathbf{x}^*$.
   - (d) Compute
   $$\widehat{Z}_Z = \frac{1}{N} \sum_{i=1}^N \rho_D^{(i)}.$$
   where $\rho_D^{(i)} = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)}|\mathbf{x}^*)}$, $i = 1, \dots, N$.
   - (e) Set $\mathbf{x}_k = \mathbf{x}^*$ with probability
   $$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}_X}{\widehat{Z}_Z},$$
   otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$.

and covariance matrices Σ_1, Σ_2, respectively. Clearly, this is only one possibility and several alternatives could be explored.

5. NUMERICAL SIMULATIONS

5.1. Comparison among schemes with independent proposal pdfs

In order to the the different techniques, we consider a multidimensional Gaussian target density,

$$\bar{\pi}(\mathbf{x}) = \bar{\pi}(x_1, \ldots, x_D) = \prod_{d=1}^{D} \mathcal{N}(x_d|\mu_d, \sigma^2), \tag{25}$$

with $\mathbf{x} = x_{1:D} \in \mathbb{R}^D$, $D = 10$, with $\mu_{1:3} = 2$, $\mu_{4:7} = 4$, $\mu_{8:10} = -1$, and $\sigma = \frac{1}{2}$. We apply I-MTM, I-MTM2, PMH and Variant-PMH for estimating the vector $\mu_{1:10}$. In each method, we employ Gaussian proposal pdf

$$q(x_d|x_{d-1}) = \mathcal{N}(x_d|x_{d-1}, \sigma_p^2),$$

with $\sigma_p = 2$, for the sequential construction of the N particles. For PMH and Variant-PMH, we consider to perform resampling at each iteration (in I-MTM and I-MTM2, no resampling is applied).

We test the techniques considering different value of number of particles N and number of iterations of the chain K. We compute the MSE in estimating the vector $\mu_{1:10}$, averaging over 500 independent simulations. The starting particles, $d = 1$, are chosen randomly $x_1^{(i)} \sim \mathcal{N}(x; -2, 4)$, for $i = 1, \ldots, N$, at each run and for each method. Figures 4(a)-(b) show the MSE as function of number of iterations K in semilog scale, keeping fixed the number of tries $N = 3$. Figure 4(a) reports the results of the MTM schemes whereas Figure 4(b) reports the results of the PMH schemes. Figure 4(c) depicts the MSE in the estimation of $\mu_{1:10}$ of function of N, for the PMH methods. These results show that the use of an acceptance probability of type in Eq. (20)-(22) provide smaller MSE. This is more evident for small number of candidates N. Namely, the use of acceptance probability in Eq. (22) within a PMH is preferable since provides better performance. When N grows, the performance of both PMH methods becomes similar, since the acceptance probability approaches 1, in both cases. The MSE vanishes to zero when N increases, as expected, confirming the validity of the novel schemes. The results also shows that performing resampling at each iteration is not optimal and that a smaller rate of resampling steps could improve the performance [**?**]. Figure 4(d) depicts 35 different states $\mathbf{x}_k = x_{1:10,k}$ at different iteration indices k, obtained with var-PMH ($N = 1000$ and $K = 1000$) and the values $\mu_{1:10}$ shown in dashed line.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

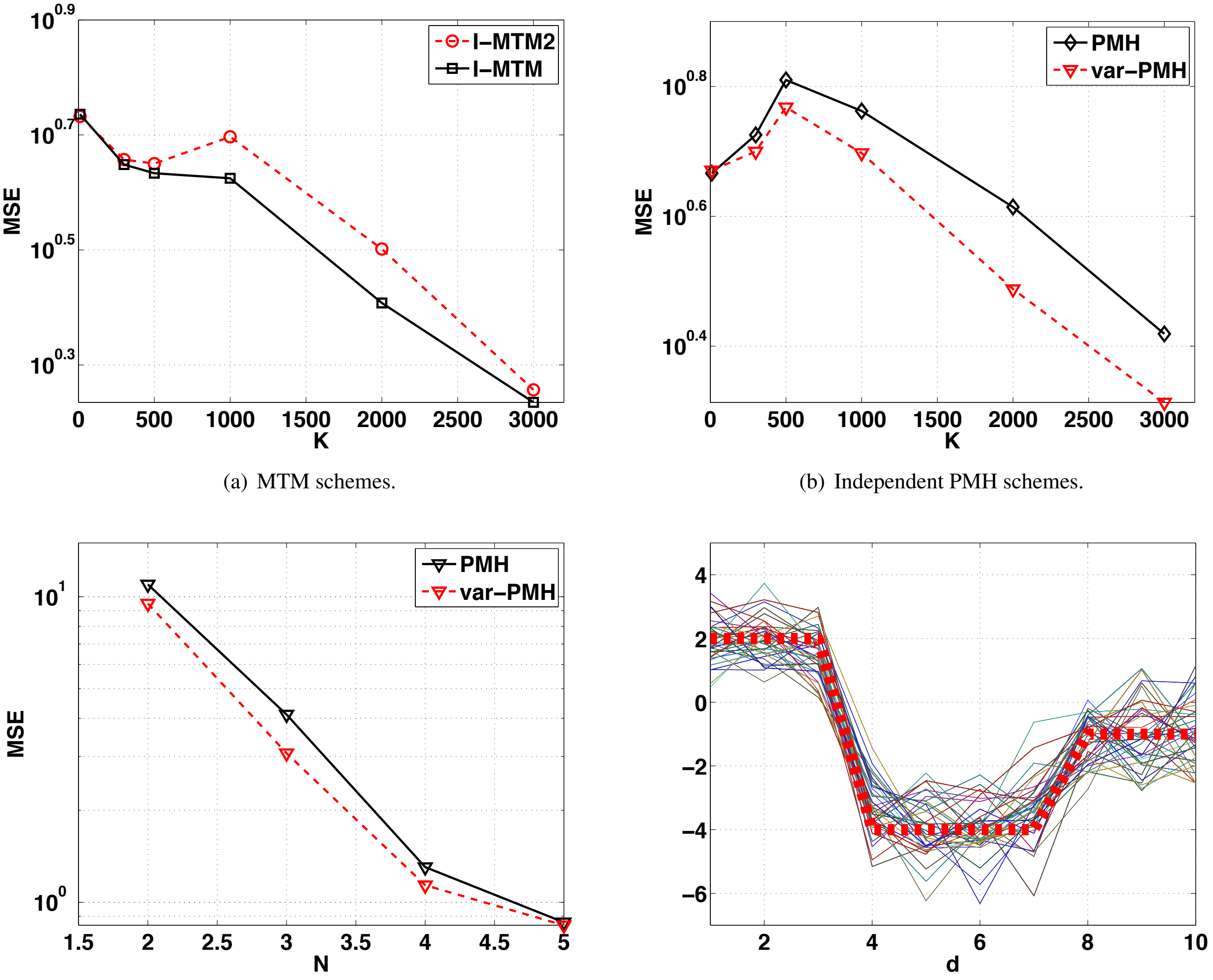
[2] R. Casarin, R. Craiu, and F. Leisen. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 23(2):185–200, 2013.

[3] R. V. Craiu and C. Lemieux. Acceleration of the Multiple-Try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing*, 17(2):109–120, 2007.

[4] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, September 2003.

[5] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York (USA), 2001.

[6] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*. Academic Press, San Diego, 1996.



(a) MTM schemes.

(b) Independent PMH schemes.

(c) Independent PMH schemes.

(d) var-PMH.

Fig. 4. **(a)-(b)** MSE versus number of iterations K of the chain in semilog scale, fixing the number of particles $N = 3$. **(a)** I-MTM (solid line) and I-MTM2 (dashed line). **(b)** PMH (solid line) and var-PMH (dashed line) using the acceptance probability in Eq. (22). **(c)** MSE versus N of the chain in semilog scale for PMH (solid line) and variant of PMH (dashed line). **(d)** Different states $\mathbf{x}_k = x_{1:10,k}$ at different iteration indices k, obtained with var-PMH ($N = 1000$ and $K = 1000$). The values $\mu_{1:10}$ are shown in dashed line ($\mu_{1:3} = 2$, $\mu_{4:7} = 4$ and $\mu_{8:10} = -1$).

[7] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[8] G. Kobayashi and H. Kozumi. Generalized multiple-point metropolis algorithms for approximate bayesian computation. *Journal of Statistical Computation and Simulation (DOI:10.1080/00949655.2013.836652)*, 2013.

[9] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, England, 2010.

[10] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[11] J. S. Liu, F. Liang, and W. H. Wong. The Multiple-Try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, March 2000.

[12] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive sticky generalized Metropolis. *arXiv:1308.3779*, 2013.

[13] L. Martino, V. P. Del Olmo, and J. Read. A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453, 2012.

[14] L. Martino and J. Read. On the flexibility of the design of multiple try metropolis schemes. *Computational Statistics (DOI 10.1007/s00180-013-0429-2)*, 2013.

[15] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[16] S. Pandolfi, F. Bartolucci, and N. Friel. A generalized multiple-try version of the reversible jump algorithm. *Computational Statistics* & *Data Analysis (available online)*, 2013.

[17] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Series in Electrical Engineering, 1984.

[18] Z. S. Qin and J. S. Liu. Multi-Point Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 172:827–840, 2001.

[19] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[20] J. I. Siepmann and D. Frenkel. Configurational bias Monte Carlo: a new sampling scheme for flexible chains. *Molecular Physics*, 75(1):59–70, 1992.

[21] G. Storvik. On the flexibility of Metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics*, 38(2):342–358, February 2011.

[22] Y. Zhang and W. Zhang. Improved generic acceptance function for multi-point Metropolis algorithms. *2nd International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT-2012)*, 2012.

A. ALTERNATIVE FORMULATION OF THE ESTIMATOR OF Z

In SIS approach, there are two possible formulations of the estimators of Z, the first one $\widehat{Z}$ in Eqs. (4)-(10) and the second one $\widetilde{Z}$ given in Eq. (11). This alternative formulation can be also derived as follows. Consider the following integrals,

$$Z_d = \int_{\mathcal{X}^d} \pi_d(x_{1:d}) dx_{1:d} \approx \widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)}, \tag{26}$$

and

$$\begin{aligned}
\int_{\mathcal{X}^d} \gamma_d(x_d|x_{1:d-1}) \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} &= \int_{\mathcal{X}^d} \frac{\pi_d(x_{1:d})}{\pi_{d-1}(x_{1:d-1})} \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d}, &(27)\\
&= \frac{Z_d}{Z_{d-1}}. &(28)
\end{aligned}$$

Clearly, we can write

$$\begin{aligned}
\int_{\mathcal{X}^d} \gamma_d(x_d|x_{1:d-1}) \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} &= \int_{\mathcal{X}^d} \frac{\gamma_d(x_d|x_{1:d-1})}{q_d(x_d|x_{1:d-1})} q_d(x_d|x_{1:d-1}) \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d},\\
&= \int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d}, &(29)
\end{aligned}$$

where we have set $\beta_d(x_d|x_{1:d-1}) = \frac{\gamma_d(x_d|x_{1:d-1})}{q_d(x_d|x_{1:d-1})}$. Replacing $\bar{\pi}_{d-1}(x_{1:d-1})$ with $\widehat{\pi}_{d-1}(x_{1:d-1})$ given in Eq. (9),

$$\begin{aligned}
&\int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \widehat{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} =\\
&= \sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \delta(x_{1:d-1} - x_{1:d-1}^{(n)}) dx_{1:d},\\
&= \sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \int_{\mathcal{X}} \beta_d(x_d|x_{1:d-1}^{(n)}) q_d(x_d|x_{1:d-1}^{(n)}) dx_d.
\end{aligned}$$

Hence, using again Monte Carlo for approximating each integral within the sum, i.e., given N samples $x_d^{(n)} \sim q_d(x_d|x_{1:d-1}^{(n)})$, $n = 1, \ldots, N$ (one sample for each different $q_d(\cdot|x_{1:d-1}^{(n)})$), and denoting $\beta_d^{(n)} = \beta_d(x_d^{(n)}|x_{1:d-1}^{(n)})$, we obtain

$$\begin{aligned}
\int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \widehat{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} &= \sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \beta_d^{(n)}, &(30)\\
&= \frac{1}{\sum_{i=1}^{N} w_{d-1}^{(i)}} \sum_{n=1}^{N} w_{d-1}^{(n)} \beta_d^{(n)},\\
&= \frac{1}{\sum_{i=1}^{N} w_{d-1}^{(i)}} \sum_{n=1}^{N} w_d^{(n)},\\
&= \frac{\frac{1}{N} \sum_{n=1}^{N} w_d^{(n)}}{\frac{1}{N} \sum_{i=1}^{N} w_{d-1}^{(i)}} = \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} \approx \frac{Z_d}{Z_{d-1}}, &(31)
\end{aligned}$$

where we have used $\bar{w}_{d-1}^{(n)} = \frac{w_{d-1}^{(n)}}{\sum_{i=1}^{N} w_{d-1}^{(i)}}$, the recursive expression of the weights, $w_d^{(n)} = w_{d-1}^{(n)}\beta_d^{(n)}$, and $\widehat{Z}_d$ is the estimator in Eq. (26). Finally, we can obtain, setting $\hat{Z}_0 = 1$,

$$\widetilde{Z} = \prod_{d=1}^{D} \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} = \widehat{Z}_1 \frac{\widehat{Z}_2}{\widehat{Z}_1} \cdots \frac{\widehat{Z}_{D-1}}{\widehat{Z}_{D-2}} \frac{\widehat{Z}_D}{\widehat{Z}_{D-1}} = \prod_{d=1}^{D} \left[\sum_{i=1}^{N} \bar{w}_{d-1}(x_{1:d-1}^{(i)}) \beta_d(x_d^{(i)}|x_{1:d-1}^{(i)}) \right] \approx Z, \tag{32}$$

that is exactly the estimator in Eq. (11).

A.1. Application of resampling

Consider to approximate the integral in Eq. (30) via importance sampling assuming in this case to draw N samples, $x_{1:d}^{(1)}$,. . ., $x_{1:d}^{(N)}$, from the pdf $q_d(x_d|x_{1:d-1})\widehat{\pi}_{d-1}(x_{1:d-1})$, so that we can write

$$\int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \widehat{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} \approx \frac{1}{N} \sum_{n=1}^{N} \beta_d^{(n)} \approx \frac{Z_d}{Z_{d-1}}, \tag{33}$$

where we remark $x_{1:d}^{(n)} \sim q_d(x_d|x_{1:d-1})\widehat{\pi}_{d-1}(x_{1:d-1})$ for $n = 1, \ldots, N$.

B. PARTICLE MARGINAL METROPOLIS-HASTINGS (PM-MH) ALGORITHM AND ALTERNATIVES

The *Particle Marginal Metropolis-Hastings* (PM-MH) algorithm is an extension of the PMH method for the combined sampling of dynamic and fixed unknown parameters, denoted as $\mathbf{x}$ and θ, respectively. Let us consider the following state space model

$$\begin{cases} q_d(x_d|x_{d-1}, \theta), \\ \ell_d(y_d|x_d, \theta) \end{cases} \tag{34}$$

where q_d represents a transition probability, and ℓ_d is the likelihood function. The parameter $\theta \in \mathbf{\Theta}$ is considered also unknown so that the inference problem consists in inferring $(x_{1:D}, \theta)$ given the sequence of received measurements $y_{1:D}$. With respect to the notation used in Section 2.1, we have $\gamma_1(x_1|\theta) = \ell_1(y_1|x_1,\theta)q_1(x_1|\theta)$, and

$$\gamma_d(x_d|x_{1:d-1}, \theta) = \ell_d(y_d|x_d, \theta) q_d(x_d|x_{d-1}, \theta),$$

with $d = 2, \ldots, D$. Hence, considering also a prior $p(\theta)$ over θ, and $\mathbf{x} = x_{1:D}$, $\mathbf{y} = y_{1:D}$, the complete target is

$$\begin{aligned} \bar{\pi}(\mathbf{x}, \theta|\mathbf{y}) &= \bar{\pi}(\mathbf{x}|\mathbf{y}, \theta) p(\theta|\mathbf{y}), && (35) \\ &= \bar{\pi}(\mathbf{x}|\mathbf{y}, \theta) \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}, && (36) \\ &= \bar{\pi}(\mathbf{x}, \mathbf{y}|\theta) \frac{p(\theta)}{p(\mathbf{y})}, && (37) \\ &= \left[\ell_1(y_1|x_1, \theta) q_1(x_1|\theta) \prod_{d=2}^{D} \ell_d(y_d|x_d, \theta) q_d(x_d|x_{d-1}, \theta) \right] \frac{p(\theta)}{p(\mathbf{y})}. && (38) \end{aligned}$$

We can evaluate $\bar{\pi}(\mathbf{x}, \mathbf{y}|\theta) \propto \bar{\pi}(\mathbf{x}|\mathbf{y}, \theta)$, it is not an issue using a self-normalized IS approach for approximating $\bar{\pi}(\mathbf{x}|\mathbf{y}, \theta)$. However, we cannot evaluate $p(\theta|\mathbf{y})$, $p(\mathbf{y}|\theta)$ and $p(\mathbf{y})$. Let us consider to apply a standard MH method for sampling from $\bar{\pi}(\mathbf{x}, \theta|\mathbf{y})$. We assume possible to draw samples $[\mathbf{x}, \theta]$ as proposal pdf

$$q(\theta^*, \mathbf{x}^*|\theta_{k-1}) = q_\theta(\theta^*|\theta_{k-1}) \bar{\pi}(\mathbf{x}^*|\mathbf{y}, \theta^*),$$

where $k = 1, \dots, K$ is the iteration of the chain and $\bar{\pi}(\mathbf{x}|\mathbf{y}, \theta)$ is the posterior of $\mathbf{x}$. Assume hypothetically that it is possible to draw from $q(\theta_k, \mathbf{x}_k|\theta_{k-1})$, we obtain the following acceptance probability

$$\begin{aligned} \alpha &= 1 \wedge \frac{\bar{\pi}(\mathbf{x}^*, \theta^*|\mathbf{y})q(\theta_{k-1}, \mathbf{x}_{k-1}|\theta^*)}{\bar{\pi}(\mathbf{x}_{k-1}, \theta_{k-1}|\mathbf{y})q(\theta^*, \mathbf{x}^*|\theta_{k-1})}, && (39) \\ &= 1 \wedge \frac{\bar{\pi}(\mathbf{x}^*, \theta^*|\mathbf{y})q_\theta(\theta_{k-1}|\theta^*)\bar{\pi}(\mathbf{x}_{k-1}|\mathbf{y}, \theta_{k-1})}{\bar{\pi}(\mathbf{x}_{k-1}, \theta_{k-1}|\mathbf{y})q_\theta(\theta^*|\theta_{k-1})\bar{\pi}(\mathbf{x}^*|\mathbf{y}, \theta^*)}. && (40) \end{aligned}$$

Then, since $\bar{\pi}(\mathbf{x}, \theta|\mathbf{y}) = \bar{\pi}(\mathbf{x}|\mathbf{y}, \theta)p(\theta|\mathbf{y})$, we can replace in the expression above

$$\begin{aligned} \alpha &= 1 \wedge \frac{\bar{\pi}(\mathbf{x}^*|\mathbf{y}, \theta^*)p(\theta^*|\mathbf{y})q_\theta(\theta_{k-1}|\theta^*)\bar{\pi}(\mathbf{x}_{k-1}|\mathbf{y}, \theta_{k-1})}{\bar{\pi}(\mathbf{x}_{k-1}|\mathbf{y}, \theta_{k-1})p(\theta_{k-1}|\mathbf{y})q_\theta(\theta^*|\theta_{k-1})\bar{\pi}(\mathbf{x}^*|\mathbf{y}, \theta^*)}, && (41) \\ &= 1 \wedge \frac{p(\theta^*|\mathbf{y})q_\theta(\theta_{k-1}|\theta^*)}{p(\theta_{k-1}|\mathbf{y})q_\theta(\theta^*|\theta_{k-1})}, && (42) \\ &= 1 \wedge \frac{p(\mathbf{y}|\theta^*)p(\theta^*)q_\theta(\theta_{k-1}|\theta^*)}{p(\mathbf{y}|\theta_{k-1})p(\theta_{k-1})q_\theta(\theta^*|\theta_{k-1})}, && (43) \end{aligned}$$

The problem is that, in general, we are not able to evaluate the likelihood function

$$Z(\theta) = p(\mathbf{y}|\theta) = \int_{\mathcal{D}} \bar{\pi}(\mathbf{x}, \mathbf{y}|\theta)d\mathbf{x}.$$

However, we can approximate $Z(\theta)$ via importance sampling. Thus, the idea is to use the approximate proposal pdf

$$\widehat{q}(\theta^*, \mathbf{x}^*|\theta_{k-1}) = q_\theta(\theta^*|\theta_{k-1})\widehat{\pi}(\mathbf{x}^*|\mathbf{y}, \theta^*),$$

where $\widehat{\pi}$ is a particle approximation of $\bar{\pi}$ obtained by SIR and, at the same, we get the estimation $\widehat{Z}(\theta^*)$. Therefore, the PM-MH algorithm can be summarized as following:

1. For $k = 1, \dots, K$:
   (a) Draw $\theta^* \sim q_\theta(\theta|\theta_{k-1})$ and then $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x}|\mathbf{y}, \theta^*)$ via SIR.
   (b) Set $[\theta_k, \mathbf{x}_k] = [\theta^*, \mathbf{x}^*]$ with probability

$$\alpha = 1 \wedge \frac{\hat{Z}(\theta^*)p(\theta^*)q_\theta(\theta_{k-1}|\theta^*)}{\hat{Z}(\theta_{k-1})p(\theta_{k-1})q_\theta(\theta^*|\theta_{k-1})}$$

   otherwise set $[\theta_k, \mathbf{x}_k] = [\theta_{k-1}, \mathbf{x}_{k-1}]$.

Given the observations provided in this work, PM-MH can be seen as a combination of a MH method w.r.t. θ and a MTM-type method w.r.t. $\mathbf{x}$.