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) algorithms and Sequential Monte Carlo (SMC) methods (a.k.a., particle filters) are well-known Monte Carlo methodologies, widely used in different fields for Bayesian 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 another advanced MCMC technique specifically designed for scenarios where the multidimensional target density can be easily factorized as multiplication of conditional densities. PMH combines jointly SMC and MCMC approaches. Both, MTM and PMH, have been widely studied and applied in literature. PMH variants have been often applied for the joint purpose of tracking dynamic variables and tuning constant parameters in a state space model. Furthermore, PMH can be also considered as an alternative particle smoothing method. In this work, we investigate connections, similarities and differences among MTM schemes and PMH methods. This study allows the design of novel efficient schemes for filtering and smoothing purposes in state space models. More specially, one of them, called *particle Multiple Try Metropolis* (P-MTM), obtains very promising results in different numerical simulations.

Keywords: Bayesian Inference; Particle Filter; Particle smoother; Markov Chain Monte Carlo (MCMC); Multiple Try Metropolis; Particle MCMC.

1. INTRODUCTION

Monte Carlo statistical methods are powerful tools for numerical inference and stochastic optimization [34, 21]. Markov Chain Monte Carlo (MCMC) [15, 19, 20, 34] and Sequential Monte Carlo (SMC) algorithms (a.k.a., particle filters) [2, 7, 10, 29] are classical Monte Carlo techniques employed in order to approximate an otherwise-incalculable (analytically) integral involving a complicated target probability density function (pdf) [21, 20] (the posterior distribution in Bayesian inference). MCMC algorithms produce a Markov chain with a stationary distribution that coincides with the target pdf, whereas SMC methods (a.k.a., particle filters) yield an approximation of the target measure by weighted samples.

The *Multiple Try Metropolis* (MTM) method of [22], [21, Chapter 5] is an advanced MCMC technique, which is extension of the well-known Metropolis-Hastings (MH) algorithm [28, 13]. In MTM, the next state of the chain is selected among a set of candidates according to some suitable weights. This enables the MTM sampler to make large step-size jumps without a lowering of the acceptance rate and thus the exploration of a larger portion of the state space is facilitated. A famous special case of MTM, well-known in molecular simulation field, is the orientational

---

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.

bias Monte Carlo technique [11]. Due to its good performance, several generalizations of the basic MTM scheme [22] can be found in literature: with correlated candidates [26, 33], more general form of the weights and different frameworks [18, 27, 31, 39], with adaptive and interacting proposal pdfs [4]. Interesting and related studies about MTM or the use of multiple auxiliary variables for building acceptance probabilities within a MH approach can be found in [3, 37, 25].

Independently from the derivation of the MTM schemes, the class of *Particle MCMC methods* (P-MCMC) has been proposed [1, 6, 30, 38] in literature. P-MCMC methods are specifically designed to solve inference problems in state space models, combining SMC and the MCMC approaches. In this work, we focus on a specific P-MCMC method, called *Particle Metropolis-Hastings* (PMH) algorithm [1, 6]. The idea behind PMH is first to approximate the target measure with a delta approximation based on weighted samples obtaining by a particle filter, and then to use this approximation as proposal pdf within MH technique. The *Particle Marginal MH* technique is a variant of the standard PMH method designed in order to estimate jointly the sequence of hidden states and the static parameters of the model [38]. Note that the standard PMH scheme can be interpreted as a *particle smoother* [9, 10, 12, 16, 17, 36] since, at each iteration of PMH, the weighted paths obtained by a particle filter are employed in order to update the previous estimation according to a suitable MH-type rule, taking into account all the received observations.[1]

The authors in [1] discuss the relationships of P-MCMC with other existing techniques. They mention and describe precisely the relationship with the so-called configurational bias Monte Carlo method [35],[21, Chapter 5]. This technique is also strictly connected to the MTM scheme. The authors also allude quickly to the MTM method [22]. However, the relationship between MTM and PMH deserves a more careful look. In this work, we show that MTM and PMH algorithms are strictly connected. PMH can be interpreted as an MTM using an independent proposal pdf which generates correlated candidates, drawn and weighted sequentially through a particle filter. In order to clarify the relationship between MTM and PMH, we recall the batch importance sampling (IS) and sequential importance sampling (SIS) methods and point out some relevant considerations about different estimators of the marginal likelihood (a.k.a, Bayesian evidence) [9]. Furthermore, we also introduce a variant of the MTM scheme with independent proposal pdf. These observations are essential in order to show the connection between MTM and PMH.

This exhaustive study allows the design of novel more efficient schemes, taking advantage of the different alternatives and analyses already provided in literature about both techniques [37, 27, 30, 38]. Hence, we propose novel possible MTM and PMH schemes. One of them, called *Particle MTM (P-MTM)*, combines the standard PMH and MTM kernels, mixing properly the main advantages of both: the sequential construction of the different tries by PMH, and the possibility given an MTM scheme of considering several candidates perturbing the previous state of the chain. P-MTM provide excellent performance as we show with numerical results. In one simulation, we test P-MTM as a particle smoother in order to make inference of a sequence of hidden states in a stochastic volatility model, obtaining very favorable results.

The paper is structured as follows. In Section 2, we recall some required concepts about importance sampling and resampling techniques. The MTM methods are described in Section 3 whereas the description of the PMH algorithms and their relationship with the MTM schemes is provided in Section 4. Novel schemes are discussed in Section 5. Section 6 is devoted to the numerical simulations and, in Section 7, we provide some conclusions.

2. IMPORTANCE SAMPLING

In many applications, we desire to infer a vector of unknown parameters, $\mathbf{x} \in \mathbb{R}^{D\times\zeta}$, give a set of observed data, $\mathbf{y} \in \mathbb{R}^{d_Y}$. In these cases, one is interested in approximating different moments the posterior density $\bar{\pi}(\mathbf{x}|\mathbf{y})$ that,

---

[1] Let us denote $\bar{\pi}(x_{1:D}|y_{1:D})$ the posterior of the hidden states $x_{1:D}$ given the received observations $y_{1:D}$. PMH schemes can be also employed for improving the approximations of the marginal posteriors $\bar{\pi}(x_d|y_{1:D})$ with $d \leq D$, jointly with the complete posterior $\bar{\pi}(x_{1:D}|y_{1:D})$. PMH can be interpreted as a way to combine properly several independent runs of a particle filter (after obtaining all the observations $y_{1:D}$), generating a ergodic Markov chain with invariant distribution $\bar{\pi}(x_{1:D}|y_{1:D})$ [9].

hereafter, we simply denote as $\bar{\pi}(\mathbf{x})$. More specifically, in this work, we denote the variable of interest as

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

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

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

is often known as *marginal likelihood* (a.k.a., *Bayesian inference*). In many application, we are only able to evaluate $\pi(\mathbf{x})$ since Z_D is unknown. Moreover, in general, we are not able to draw random samples from $\bar{\pi}(\mathbf{x})$. Monte Carlo techniques employ a simpler proposal density, denoted as $q(\mathbf{x})$, with support $\mathcal{X} \subseteq \mathbb{R}^{D\times\zeta}$,[2] for generating possible random candidates. Then, these candidates are filtered using some suitable procedure, in order to produce a particle approximation of $\bar{\pi}(\mathbf{x})$ and also to provide an estimation of Z_D.

2.1. Batch and Sequential Importance Sampling

A well-known Monte Carlo technique 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

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

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

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

where we have denoted the normalized weights as $\bar{w}_D^{(n)} = \frac{w_D^{(n)}}{\sum_{i=1}^N w_D^{(i)}}$. An estimation of Z_D is given by

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

In high dimensional spaces ($\mathbf{x} \in \mathcal{D} = \mathcal{X}^D \subseteq \mathbb{R}^{D\times\zeta}$), an equivalent sequential procedure, called *sequential importance sampling* (SIS), 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 [32] 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 the partial vector $x_{1:d} = [x_1, \ldots, x_d]$,

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

---

[2] 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., $\int_{\mathcal{X}} q(\mathbf{x}) d\mathbf{x} = 1$.

where

$$Z_d = \int_{\mathcal{X}^d} \pi_d(x_{1:d}) dx_{1:d}, \tag{7}$$

and, clearly, we have $\bar{\pi}_D(x_{1:D}) = \bar{\pi}(\mathbf{x})$. 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

$$\begin{aligned} w_d^{(n)} &= w_{d-1}^{(n)} \beta_d^{(n)}, \\ &= \prod_{j=1}^{d} \beta_j^{(n)}, \qquad d = 1, \ldots, D, \end{aligned} \tag{8}$$

where we have set

$$\beta_1^{(n)} = w_1^{(n)} \quad \text{and} \quad \beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{1:d-1}^{(n)})}{q_d(x_d^{(n)}|x_{1:d-1}^{(n)})}, \tag{9}$$

for $d = 2, \ldots, D$. Thus, given N samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, finally we obtain the particle approximations of the sequence of pdfs $\bar{\pi}_d(x_{1:d})$ as

$$\widehat{\pi}_d(x_{1:d}) = \sum_{n=1}^{N} \bar{w}_d^{(n)} \delta(x_{1:d} - x_{1:d}^{(n)}), \quad d = 1, \ldots, D, \tag{10}$$

and an estimator of each normalizing constant Z_d is given by

$$\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N} \sum_{n=1}^{N} \left[\prod_{j=1}^{d} \beta_j^{(n)} \right]. \tag{11}$$

However, an alternative *equivalent* formulation is often used

$$\begin{aligned} \widetilde{Z}_d &= \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right], & (12) \\ &= \prod_{j=1}^{d} \left[\frac{\sum_{n=1}^{N} w_j^{(n)}}{\sum_{n=1}^{N} w_{j-1}^{(n)}} \right], & (13) \\ &= \prod_{j=1}^{d} \left[\frac{\widehat{Z}_j}{\widehat{Z}_{j-1}} \right] = \frac{\widehat{Z}_1}{\widehat{Z}_0} \frac{\widehat{Z}_2}{\widehat{Z}_1} \times \ldots \times \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} = \widehat{Z}_d, & (14) \end{aligned}$$

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.* (12) *of estimator of Z_d.*

2.2. Sequential Importance Resampling (SIR)

Sequential Importance Resampling (SIR) [21, 34] combines the sequential construction of the importance weights as in SIS with the application of *resampling* steps [7, 8]. Namely, when some pre-established criterion is fulfilled [7, 8, 23], 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 [7, 8]. A proper choice [24] is to set the unnormalized importance weights

$$w_d^{(n)} = \widehat{Z}_d, \quad \forall j = 1, \dots, N. \tag{15}$$

i.e., $w_d^{(1)} = w_d^{(2)} = \dots = w_d^{(N)}$, 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 $j = 1, \dots, N$. One reason why this is a good choice, for instance, is that defining the following weights

$$\xi_d^{(n)}) = \begin{cases} w_d^{(n)}, & \text{without resampling at } d\text{-th iteration,} \\ \widehat{Z}_d, & \text{with resampling at } d\text{-th iteration.} \end{cases} \tag{16}$$

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

$$\xi_d^{(n)} = \xi_{d-1}^{(n)} \beta_d^{(n)}, \quad \text{where} \quad \xi_{d-1}^{(n)} = \begin{cases} \xi_{d-1}^{(n)}, & \text{without res. at } (d-1)\text{-th iter.,} \\ \widehat{Z}_{d-1}, & \text{with res. at } (d-1)\text{-th iter.} \end{cases} \tag{17}$$

See Appendix A for further details.

Remark 2. *With the recursive definition of the weights* $\xi_d^{(n)}$ *in Eq.* (17)*, the two estimators*

$$\widehat{Z}_d = \frac{1}{N}\sum_{n=1}^N \xi_{d-1}^{(n)}\beta_d^{(n)}, \quad \widetilde{Z}_d = \prod_{j=1}^d \left[\sum_{n=1}^N \bar{\xi}_{j-1}^{(n)}\beta_j^{(n)}\right] \tag{18}$$

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 *[24].*

For instance, if the resampling is applied at each iteration, observe that they become

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

and

$$\widehat{Z}_d = \widehat{Z}_{d-1}\left[\frac{1}{N}\sum_{n=1}^N \beta_d^{(n)}\right] = \prod_{j=1}^d \left[\frac{1}{N}\sum_{n=1}^N \beta_j^{(n)}\right], \tag{20}$$

and clearly coincide. Note that, w.r.t. the estimator in Eq. (11) (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 $x_1,^{(n)} \dots, x_D^{(n)}$ of each sample, with $D = 10$. Remark 2 is necessary to to describe exhaustively the relationship between MTM and PMH algorithms. Below, we recall the MTM schemes and discuss a novel suitable variant in order to link MTM to PMH.

Table 1. Generic 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 $N-1$ auxiliary 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 auxiliary 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 $\mathbf{x}_k = \mathbf{x}_{k-1}$.

3. MULTIPLE TRY METROPOLIS (MTM) ALGORITHMS

The Multiple Try Metropolis (MTM) algorithm [22] is an advanced 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 suitable 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 where we have denoted $a \wedge b = \min[a, b]$. For $N = 1$, the MTM algorithm becomes the standard Metropolis-Hastings (MH) method [21, 34]. We consider importance weights for facilitating the comparison with other techniques. However, different kind of weights could be applied [22, 27]. The MTM method generates a reversible Markov chain that converges to $\bar{\pi}(\mathbf{x})$ [22, 27].

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 auxiliary samples at step 2c. Indeed, in this case, we could directly set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \dots, N-1$. The simplified MTM algorithm (I-MTM) is given in Table 1. A graphical representation of a MTM scheme is provided in Figure 1, with $D = 1$ and $N = 2$.

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 argument, we can also use the samples generated in the previous iteration of the algorithm as auxiliary points, since all the samples are generated independently from the same proposal pdf. Namely, the alternative suitable version of the I-MTM is summarized in Table 3. Note that, in this case, we can write the acceptance probability α as

$$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}_D^*}{\widehat{Z}_D^{(k-1)}}$$

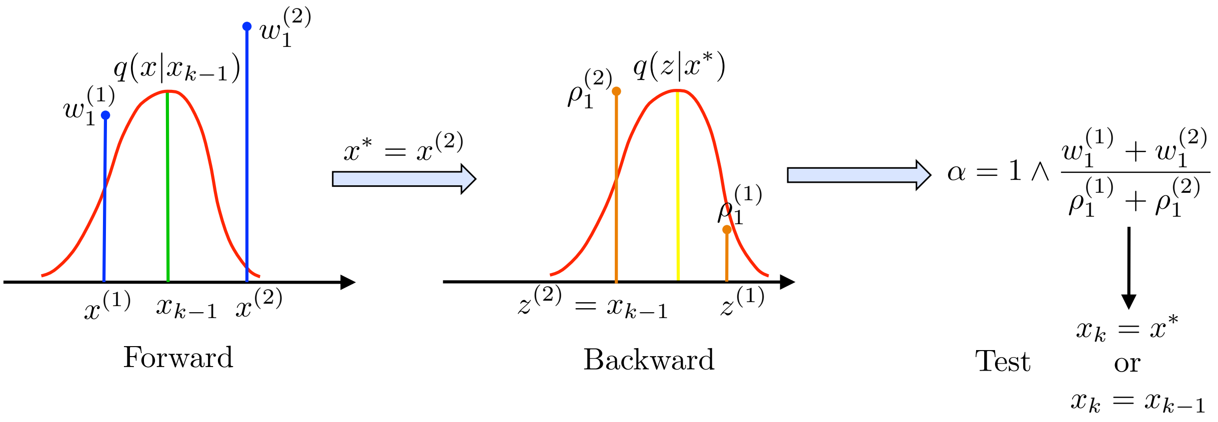


Fig. 1. Sketch of a generic MTM method with $D = 1$ and $N = 2$ tries. In this example, the second candidate is selected as $x^* = x^{(2)}$. It has been selected with probability $\bar{w}_1^{(1)} = \frac{w_1^{(1)}}{w_1^{(1)}+w_1^{(2)}}$. The auxiliary points are $z^{(1)} \sim q(z|x^*)$ and $z^{(2)} =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

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 $k = 1, \ldots, K$:
   (a) Draw N samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x})$, $i = 1, \ldots, N$.
   (b) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights

$$w_D^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \qquad i = 1, \ldots, N.$$

   Moreover, we denote as w_D^* and $w_{D,k-1}$, the weights corresponding to $\mathbf{x}^*$ and $\mathbf{x}_{k-1}$, respectively.
   (c) Set $\mathbf{x}_k = \mathbf{x}^*$ with probability

$$\alpha = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w_D^{(i)}}{\sum_{i=1}^N w_D^{(i)} - w^* + w_{D,k-1}} = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w_D^{(i)}}{\sum_{i=1}^N \rho_D^{(i)}}, \tag{21}$$

   where the values $\rho_D^{(i)}$ denote the importance weights of $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}\} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \setminus \{\mathbf{x}^*\} \cup \{\mathbf{x}_{k-1}\}$. Otherwise, set $\mathbf{x}_k = \mathbf{x}_{k-1}$.

where $\widehat{Z}_D^*$ and $\widehat{Z}_D^{(k-1)}$ are both estimators of Z_D.

4. PARTICLE METROPOLIS-HASTINGS ALGORITHM AND ITS RELATIONSHIP WITH MTM

Consider that we are able to factorize the target density 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. At each iteration, a particle filter is run in order to provide a particle approximation by N weighted samples

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

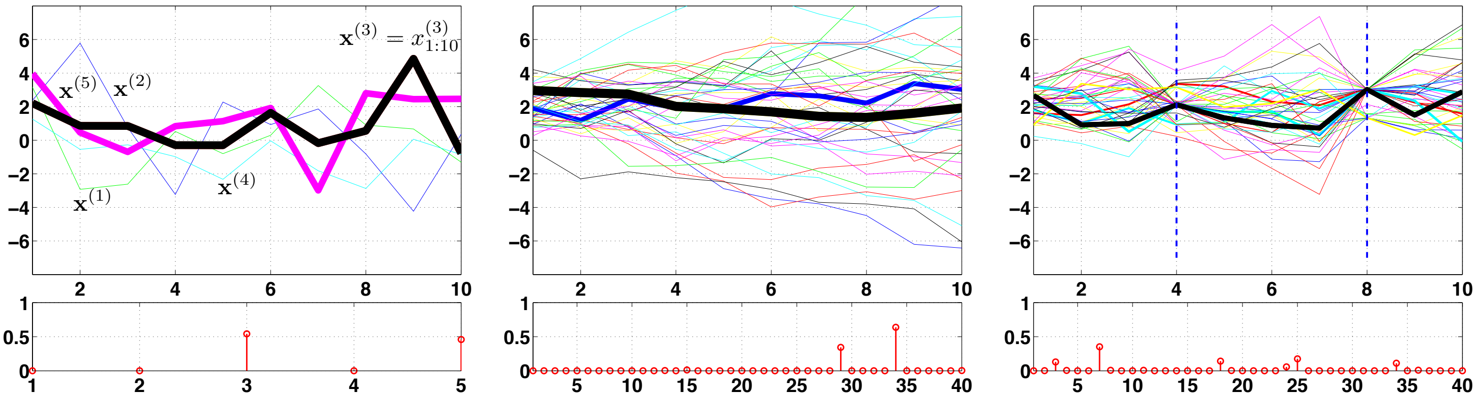
1. Choose a initial state $\mathbf{x}_0$, the total number of iterations K and obtain an estimation $\widehat{Z}^{(0)} \approx Z$.
2. For $k = 1, \ldots, K$:
    (a) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights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

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 $k = 1, \ldots, K$:
    (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. (11), or $\widetilde{Z}^*$ in Eq. (12).
    (b) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, ..., N$.
    (c) Set $\mathbf{x}_k = \mathbf{x}^*$ and $\widehat{Z}^{(k)} = \widehat{Z}^*$ with probability
    $$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}^*}{\widehat{Z}^{(k-1)}}, \tag{22}$$
    otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widehat{Z}^{(k)} = \widehat{Z}^{(k-1)}$.

of the measure of the target. Then, a sample among the N weighted particles is chosen by one resampling steps. This selected sample is then accepted or rejected as next state of the chain according to an MH-type acceptance probability, which involves two estimators of Z_D. Both estimators $\widehat{Z}$ and $\widetilde{Z}$ can be used in PMH (although the original algorithm is described with the use of $\widetilde{Z}$ [1]), if the resampled particles are properly weighted as shown in Eq. (15) [24]. A generalization of PMH for handling both dynamic and fixed parameters (as hidden states and parameters of the model, respectively), called *Particle Marginal MH* algorithm, is described in Appendix B.

Smoothing. PMH is a particle smoother since the outputs of different run of a particle filter (i.e., at each run, N weighted paths obtained with a SIR procedure) are further processed through the MH acceptance functions, taking into account all the received observations.



(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 $\mathbf{x}^{(i)}$ forms 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 $d = 4, 8$.

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 connections and differences are listed below:

- The main difference lies that the candidates in PMH are generated sequentially, using a SMC procedure (i.e., by a particle filter). 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 $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_D^{(i)}]$ from a pdf $q(\mathbf{x})$ in the space $\mathbf{x} \in \mathcal{D}$ whereas PMH draws sequentially each component $x_d^{(i)}$ of $\mathbf{x}$ from $q_d(x_d|x_{1:d-1}^{(i)})$ (see also Figure 4, for further clarifications).

- The use of resampling steps is the main difference between the generation procedures of PMH and I-MTM2. Owing to the use of the resampling, the candidates $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ 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 [5], 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 [5]), and where the candidates are generated sequentially.

  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 $x_d^{(n)}$, $d = 1, \ldots, D = 10$ of each particle $\mathbf{x}^{(n)} = x_{1:10}^{(n)}$ with $n = 1, \ldots, N \in \{5, 40\}$. 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 $d = 1, \ldots, 10$. 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 $\mathbf{x}^{(n)}$, $n = 1, \ldots, N$, are still independent. In Figure 2(c) two resampling are also applied at the iterations $d = 4, 8$, generating correlation among the particles $\mathbf{x}^{(n)}$, $n = 1, \ldots, N$, as well. Figure 2(c) corresponds to the sample generation in PMH.

- In the correspoding 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. (12). However, they are equivalent formulation of an estimator of the normalizing constant Z_D (see Remark 2), if the resampled particles are proper weighted [24].
- The PMH approach is preferable in high dimension, when the target can be factorized as in Eq. (5), since the use of the resampling steps can provide a better proposal generation procedure.

5. NOVEL SCHEMES

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

$$\alpha = 1 \quad \wedge \quad \frac{N\widehat{Z}^*}{N\widehat{Z}^* - w_D^* + w_{D,k-1}}. \tag{23}$$

We denote as var-PMH the variant of the PMH technique which uses the probability α above, instead of the probability α in Eq. (22). Namely, var-PMH is identical with the PMH method in Table 4, replacing Eq. (22) with Eq. (23). The var-PMH 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.

State Dependent PMH (SD-PMH). 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. 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 [5]). 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 with a sequential generation of the tries. If $R = K$, the resampling is applied at each iteration, so that we have a bootstrap filter for generating the samples [7, 9]. Figure 3(a) 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. As an example of proposal in SD-PMH scheme, we can consider

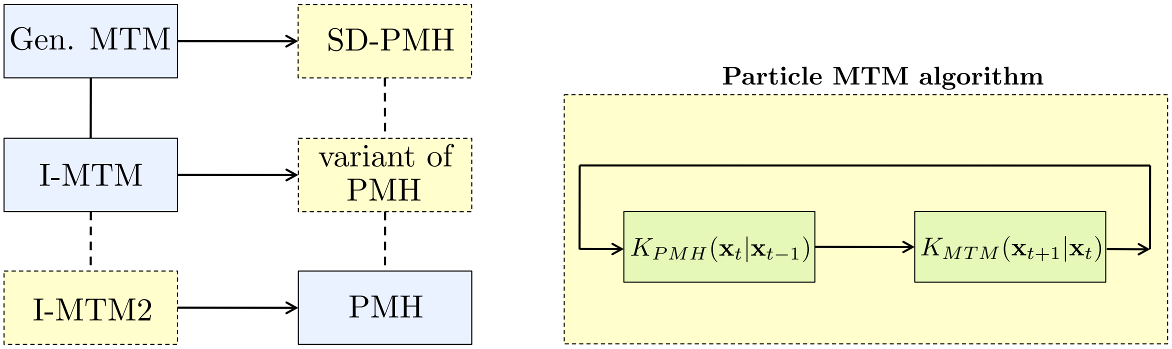
$$q_d(s_d|s_{1:d-1}, x_{1:d,k-1}) = q_d(s_d|s_{d-1}, x_{d,k-1}),$$

so that the complete proposal is $q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1}) \prod_{d=2}^{D} q_d(s_d|s_{d-1}, x_{d,k-1})$. However, it is not straightforward to choose and tune properly the components $q_d(s_d|s_{d-1}, x_{d,k-1})$.

Particle Multiple Try Metropolis (P-MTM) algorithm. A simpler and robust scheme consists in performing alternatively a standard PMH kernel (or var-PMH), denoted as $K_{PMH}(\mathbf{x}_t|\mathbf{x}_{t-1})$ and an MTM kernel with a random walk proposal pdf, denoted as $K_{MTM}(\mathbf{x}_t|\mathbf{x}_{t-1})$. Namely, Particle Multiple Try Metropolis (P-MTM) algorithm is formed by the following steps:

1. Choose an initial state $\mathbf{x}_0$, and set $t = 1$.
2. While $t < T$:
    (a) Generate a new state using one step of PMH, i.e., $\mathbf{x}_t \sim K_{PMH}(\mathbf{x}|\mathbf{x}_{t-1})$, generating sequentially with an independent proposal pdf $q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{d-1})$, and set $t = t + 1$.
    (b) Generate a new state using one step of MTM, $\mathbf{x}_t \sim K_{MTM}(\mathbf{x}|\mathbf{x}_{t-1})$, using a random walk proposal pdf, i.e., $q(\mathbf{x}|\mathbf{x}_{t-1})$, in order to draw the N candidates, and set $t = t + 1$.

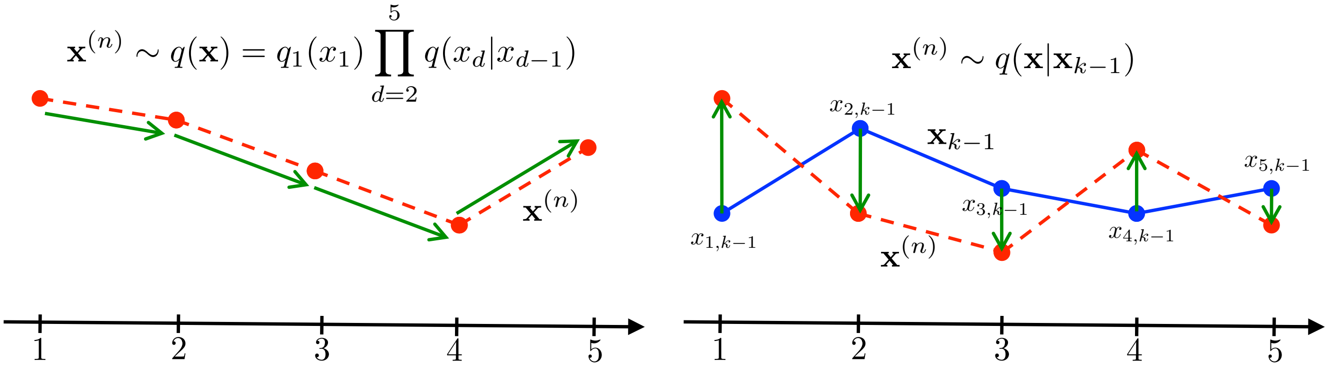
The two kernels are linked together and, since each kernel leaves invariant the target pdf $\bar{\pi}$, it is to show that the product of two valid kernels also leaves invariant $\bar{\pi}$ (see Appendix C). P-MTM exploits the advantage of selecting N particles component by component using resampling steps (as in PMH) and, jointly, the possibility of proposing N candidates taking in account the previous state of the chain. Figure 3(b) provides a graphical representation of P-MTM and Figure 4 gives a sketch of the two generation approaches jointly used in P-MTM.



(a)

(b)

Fig. 3. **(a)** Graphical representation of the MTM methods and the corresponding PMH schemes. The boxes with dashed contours contain the novel schemes presented in this work. **(b)** Graphical sketch of the P-MTM algorithm.



(a) Sequential generation of a candidate $\mathbf{x}^{(n)}$.

(b) Generation with a random walk proposal of $\mathbf{x}^{(n)}$.

Fig. 4. Examples of generation of one candidate $\mathbf{x}^{(n)}$ (with $D = 5$) **(a)** with a sequential approach as in PMH, without considering resampling steps; **(b)** from a random walk proposal $q(\mathbf{x}|\mathbf{x}_{k-1})$ which takes into account the previous state of the chain $\mathbf{x}_{k-1}$, for instance, $q(\mathbf{x}|\mathbf{x}_{k-1}) = \prod_{d=1}^{5} q_d(x_d|x_{d,k-1})$ (if each component is drawn independent from the others).

6. NUMERICAL SIMULATIONS

6.1. Comparison among different particle schemes

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

$$\bar{\pi}(\mathbf{x}) = \bar{\pi}(x_1, \dots, x_D) = \prod_{d=1}^{D} \mathcal{N}(x_d|\mu_d, \sigma^2), \tag{24}$$

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, Var-PMH (using the acceptance probability in Eq. (23)) and P-MTM for estimating the vector $\mu_{1:10}$. In each method, we employ Gaussian-piece proposal pdf for the sequential proposal construction of the N candidates, i.e.

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

setting $\sigma_p = 2$. For the random walk MTM (RW-MTM) part within P-MTM, we consider a Gaussian proposal $q_{rw}(\mathbf{x}|\mathbf{x}_{k-1}) = \prod_{d=1}^{D} \mathcal{N}(x_d|x_{d,k-1}, \sigma_{rw}^2)$ with $\sigma_{rw} = 1$ and $\mathbf{x}_{k-1} = [x_{1,k-1}, \dots, x_{D,k-1}]$ is the previous state of the chain. For all the PMH schemes, we consider to perform resampling at each iteration (in I-MTM and I-MTM2, clearly 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 $i = 1, \dots, N$, at each run and for each method. Figures 5(a)-(b) show the MSE as function of number of iterations K in semilog scale, keeping fixed the number of tries $N = 3$. Figure 5(a) reports the results of the MTM schemes whereas Figure 5(b) reports the results of the PMH schemes. Figure 5(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. (21)-(23) provide smaller MSE. This is more evident for small number of candidates N. Namely, the use of acceptance probability in Eq. (23) within a PMH is preferable since provides better performance. When N grows, the performance of PMH and var-PMH methods becomes similar, since the acceptance probability approaches 1, in both cases. P-MTM provides excellent results, outperforming the other schemes. 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 [7, 9]. Figure 5(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.

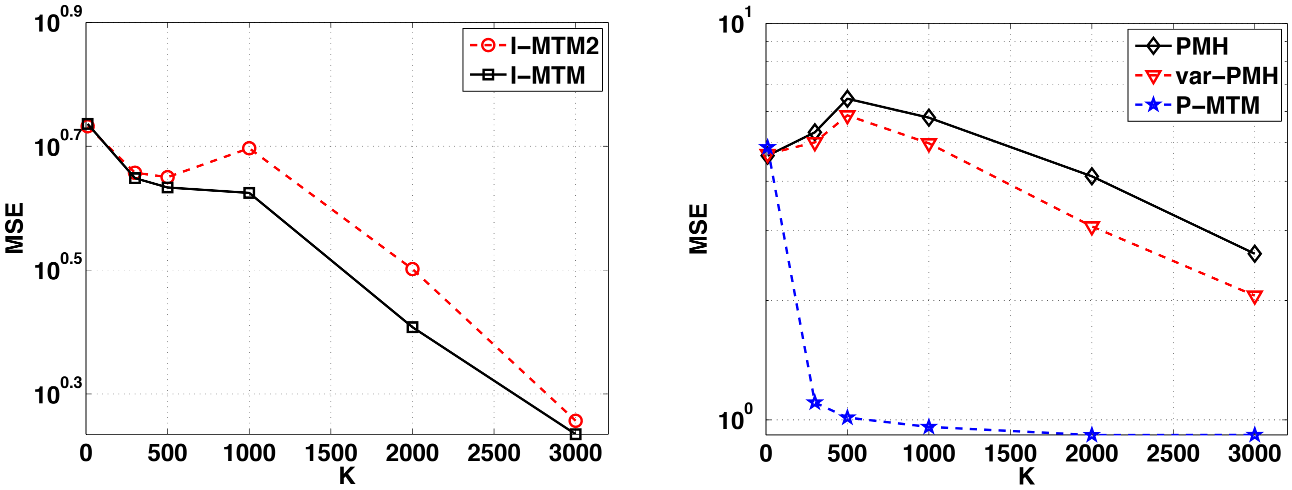
6.2. Inference in a stochastic volatility model

We consider a stochastic volatility model where the hidden state $x_d \in \mathbb{R}$ at the d-th iteration follows an AR(1) process and represents the log-volatility [14] of a financial time series at time $t \in \mathbb{N}$. The state space model is given by

$$\begin{cases} x_d = \alpha x_{d-1} + u_d, \\ y_d = \exp\left(\frac{x_d}{2}\right) v_d, \end{cases} \qquad d = 1, \dots, D. \tag{25}$$

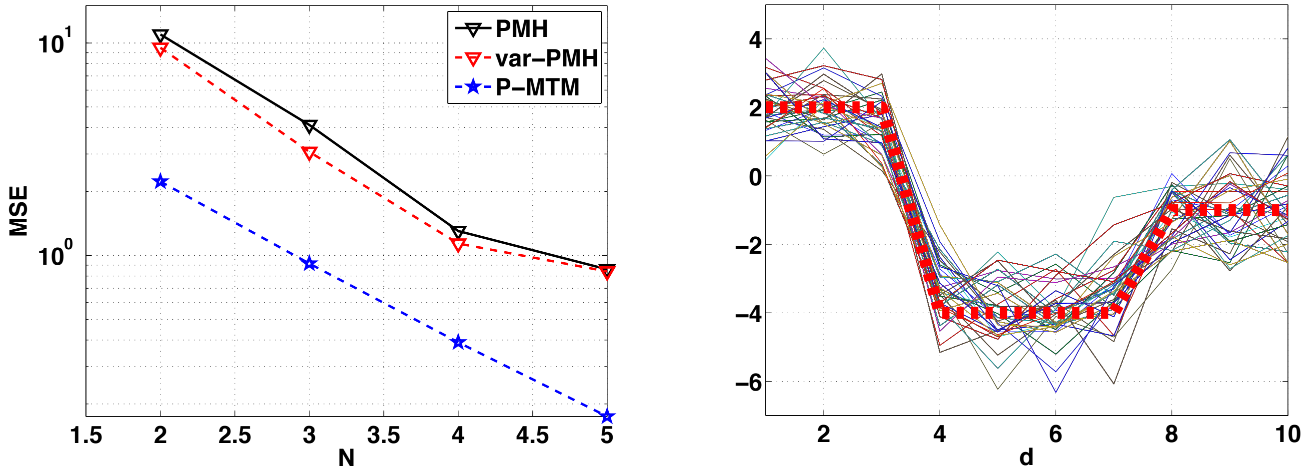
where $\alpha = 0.9$ is the AR parameter, and u_d and v_d denote independent zero-mean Gaussian random variables with variances $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.5$, respectively. Note that v_d is a multiplicative noise.

We observe a sequence $y_{1:D} \in \mathbb{R}^D$, and we desire to infer the hidden states $x_{1:D}$, analyzing the joint posterior $\bar{\pi}(x_{1:D}|y_{1:D})$. A classical particle filtering (PF) approach provides a particle approximation of the posterior $\bar{\pi}(x_{1:D}|y_{1:D})$ and, due to the Monte Carlo approach, also of the marginal posteriors $\bar{\pi}(x_d|y_{1:D})$ with $d \leq D$ (used for *smoothing*). Unfortunately, when $d << D$, the standard PF strategy in general fails: the marginal



(a) MTM schemes.

(b) PMH schemes.



(c) PMH schemes.

(d) Different States of var-PMH.

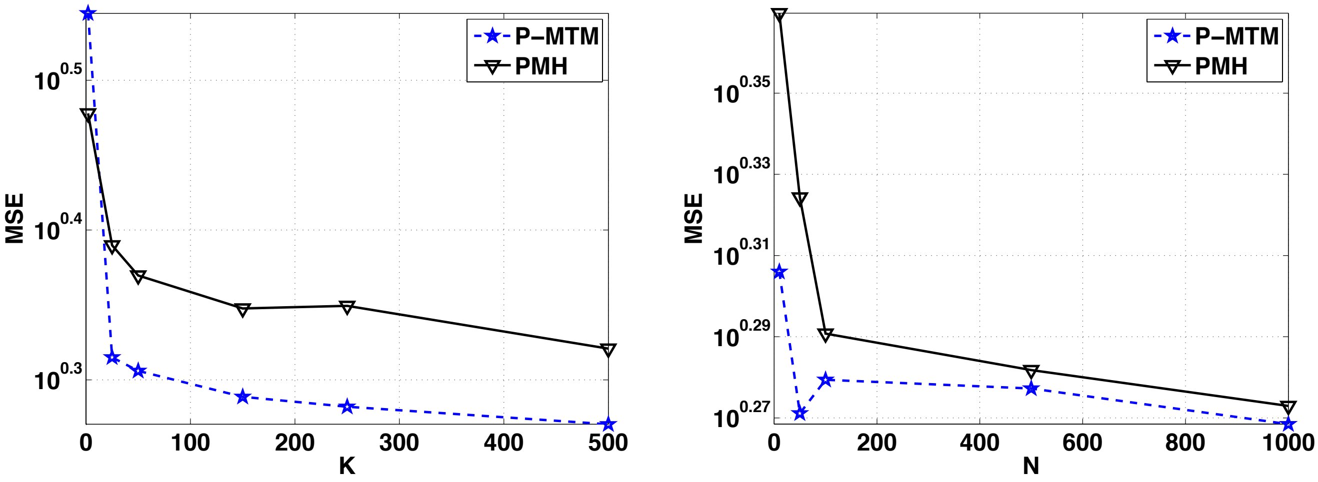
Fig. 5. **(a)**-**(b)** MSE versus number of iterations K of the chain in semilog scale, fixing the number of particles $N = 3$. **(a)** I-MTM (solid line) and I-MTM2 (dashed line). **(b)** PMH (solid line), var-PMH (triangles and dashed line) and P-MTM (stars and dashed line). **(c)** MSE versus N of the chain in semilog scale for PMH (solid line), var-PMH (triangles and dashed line) and P-MTM (stars and 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 $\mu_{8:10} = -1$).

distribution $\bar{\pi}(x_d|y_{1:d})$ (used for *filtering*) "occupies a privileged role within the particle filter framework as it is better characterized than any of the other marginal posterior distributions", as pointed out in [9]. PMH schemes can be also employed for improving the approximations of the marginal posteriors $\bar{\pi}(x_d|y_{1:D})$ with $d \leq D$ as well as of the complete posterior $\bar{\pi}(x_{1:D}|y_{1:D})$. Indeed, PMH can be interpreted as a way to combine properly several independent runs of a particle filter (after obtaining all the observations $y_{1:D}$), generating a ergodic Markov chain with invariant distribution $\bar{\pi}(x_{1:D}|y_{1:D})$.

We test the standard PMH technique [1] and the Particle MTM (P-MTM) method described in Section 5. In both cases, for the sequential proposal part, we employ a standard bootstrap particle filter (i.e., using the transition probability $p(x_d|x_{d-1})$ given by the test model as proposal and with resampling at each iteration). For the standard MTM block in P-MTM we use

$$q(\mathbf{x}|\mathbf{x}_{k-1}) = \prod_{d=1}^{D} \mathcal{N}(x_d|x_{d,k-1}, \sigma_p^2),$$

as proposal pdf, with $\sigma_p = 0.5$. We have averaged the results over 500 independent runs. At each run, we generate a different of observations $y_{1:D}$ from the model, setting $D = 100$. Clearly, both schemes are compared with the same number of evaluations of the posteriors (i.e., for instance fixing N, PMH performs K iterations of the chain, and P-MTM is formed by $\frac{K}{2}$ of PMH steps and $\frac{K}{2}$ of MTM steps, as described in Section 5). First, we consider $N = 10$ and varies the number of total iterations of the algorithms, K from $K = 2$ to $K = 500$. The results are shown in Fig. 6(a). Then, we fix $K = 50$ and test different values of N, from $N = 10$ to $N = 1000$. The results are shown in Fig. 6(b). We can observe that always P-MTM outperforms PMH obtaining smaller values of MSE (with the exception of one value with a very small number of iterations, $K = 2$, in Fig. 6(a)).



(a) MSE versus K.

(b) MSE versus N.

Fig. 6. **(a)** MSE versus number of iterations K of the generated chain in semi-log scale, fixing the number of particles ($N = 10$). **(b)** MSE versus number of particles N in semi-log scale, fixing the number of iterations of the algorithms ($K = 50$).

7. CONCLUSIONS

In this work, we have highlighted the strong connection between MTM and PMH algorithms. Specifically, PMH can be interpreted as an MTM using correlated candidates, drawn and weighted sequentially through a particle filter.

We have employed this consideration in order to introduce novel MTM and PMH schemes, mixing properly both approaches. For instance, the Particle MTM (P-MTM) algorithm combines the standard PMH and MTM kernels, using the sequential construction of the different tries obtained by PMH, and the possibility of perturbing the previous state of the chain using a MTM with random-walk proposal density. P-MTM is a very efficient technique for *filtering* and *smoothing* in state space model. We have tested P-MTM, jointly with other techniques, in different numerical simulations obtaining excellent results. As future line, we plan to design the marginal version of P-MTM for the joint purpose of inferring dynamic and static parameters in a state space framework.

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] M. S. Arulumpalam, S. Maskell, N. Gordon, and T. Klapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions Signal Processing*, 50(2):174–188, February 2002.

[3] M. Bédard, R. Douc, and E. Mouline. Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, 122:758–786, 2012.

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

[5] 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.

[6] J. Dahlin, F. Lindsten, and T. B. Schn. Particle Metropolis Hastings using Langevin dynamics. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6308–6312, May 2013.

[7] 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.

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

[9] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *technical report*, 2008.

[10] W. Fong, S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing*, 50(2):438–448, February 2002.

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

[12] S. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004.

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

[14] E. Jacquier, N. G. Polson, and P. E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, 12(4):371–389, October 1994.

[15] L. Jing and P. Vadakkepat. Interacting MCMC particle filter for tracking maneuvering target. *Digital Signal Processing*, 20:561–574, August 2010.

[16] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state-space models. *J. Comput. Graph. Statist.*, 1:1–25, 1996.

[17] G. Kitagawa and S. Sato. Monte Carlo smoothing and self-organising state-space model. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 9, pages 177–196. Springer, 2001.

[18] G. Kobayashi and H. Kozumi. Generalized multiple-point Metropolis algorithms for approximate bayesian computation. *Journal of Statistical Computation and Simulation*, 85(4), 2015.

[19] J. R. Larocque and P. Reilly. Reversible jump MCMC for joint detection and estimation of sources in colored noise. *IEEE Transactions on Signal Processing*, 50(2), February 1998.

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

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

[22] 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.

[23] L. Martino, V. Elvira, and F. Louzada. Alternative Effective Sample Size measures for Importance Sampling. *IEEE Workshop on Statistical Signal Processing (SSP)*, June 2016.

[24] L. Martino, V. Elvira, and F. Louzada. Weighting a resampled particle in Sequential Monte Carlo. *IEEE Workshop on Statistical Signal Processing (SSP)*, June 2016.

[25] L. Martino and F. Louzada. Issues in the Multiple Try Metropolis mixing. *Computational Statistics (to appear)*, pages 1–14, 2016.

[26] 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.

[27] L. Martino and J. Read. On the flexibility of the design of Multiple Try Metropolis schemes. *Computational Statistics*, 28(6), 2797-2823 2013.

[28] 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.

[29] J. Míguez, T. Ghirmai, M. F. Bugallo, and P. M. Djurić. A sequential Monte Carlo technique for blind synchronization and detection in frequency-flat Rayleigh fading wireless channels. *Signal Processing*, 84(11):2081–2096, November 2004.

[30] I. Nevat, G. W. Peters, and J. Yuan. Channel tracking in relay systems via particle MCMC. In *IEEE Vehicular Technology Conference*, pages 1–5, Sept 2011.

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

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

[33] 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.

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

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

[36] P. Stavropoulos and D. M. Titterington. Improved particle filters and smoothing. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 14, pages 295–317. Springer, 2001.

[37] 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.

[38] T. Vu, B. N. Vo, and R. Evans. A particle marginal Metropolis-Hastings multi-target tracker. *IEEE Transactions on Signal Processing*, 62(15):3953–3964, Aug 2014.

[39] 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*, 2012.

A. ALTERNATIVE FORMULATION OF THE ESTIMATOR OF Z

In SIS approach, there are two possible equivalent formulations of the estimators of Z, the first one $\widehat{Z}$ in Eqs. (4)-(11) and the second one $\widetilde{Z}$ given in Eq. (12). 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. (10),

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

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. (12).

A.1. Application of resampling

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

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

Moreover using Eq. (27), we have $\frac{1}{N}\sum_{n=1}^{N} \beta_d^{(n)} \approx \frac{Z_d}{Z_{d-1}}$.

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

The *Particle Marginal Metropolis-Hastings* (PM-MH) algorithm is a simple 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 $\theta \in \boldsymbol{\Theta}$ 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 $d = 2, \dots, D$. 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 it into 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. $\mathbf{x}$.

C. INVARIANT DENSITY OF P-MTM

Let us consider two MCMC kernels, $K_{PMH}(\mathbf{y}|\mathbf{x})$ and $K_{MTM}(\mathbf{z}|\mathbf{y})$ with $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{D} \in \mathbb{R}^{d_x}$, corresponding to the PMH and MTM steps in the P-MTM scheme, respectively. We assume $\bar{\pi}(\cdot)$ is the invariant density of both chains. The two kernels have been designed such that

$$\begin{aligned} \int_{\mathcal{D}} K_{PMH}(\mathbf{y}|\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} &= \bar{\pi}(\mathbf{y}), \\ \int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y})\bar{\pi}(\mathbf{y})d\mathbf{y} &= \bar{\pi}(\mathbf{z}). \end{aligned}$$

In P-MTM, the kernels K_{PMH}, K_{MTM} are used sequentially. Namely, first a sample is drawn from $\mathbf{y}' \sim K_{PMH}(\mathbf{y}|\mathbf{x})$ and then $\mathbf{z}' \sim K_{MTM}(\mathbf{z}|\mathbf{y}')$. The complete transition probability from $\mathbf{z}$ to $\mathbf{x}$ is given by

$$K_{PMTM}(\mathbf{z}|\mathbf{x}) = \int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) K_{PMH}(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \tag{44}$$

The target $\bar{\pi}$ is also invariant w.r.t. $K_{PMTM}(\mathbf{z}|\mathbf{x})$ [34, 21, 20]. Indeed, we can write

$$\begin{aligned} \int_{\mathcal{D}} K_{PMTM}(\mathbf{z}|\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} &= \\ &= \int_{\mathcal{D}} \left[\int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) K_{PMH}(\mathbf{y}|\mathbf{x}) d\mathbf{y}\right] \bar{\pi}(\mathbf{x}) d\mathbf{x}, \\ &= \int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) \left[\int_{\mathcal{D}} K_{PMH}(\mathbf{y}|\mathbf{x}) \bar{\pi}(\mathbf{x}) d\mathbf{x}\right] d\mathbf{y}, \\ &= \int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) \bar{\pi}(\mathbf{y}) d\mathbf{y}, \\ &= \bar{\pi}(\mathbf{z}), \end{aligned} \tag{45}$$

which is precisely the definition of invariant pdf of $K_{PMTM}(\mathbf{z}|\mathbf{x})$. Clearly, we can invert the order of the application of the kernels, i.e., using first K_{MTM} and then K_{PMH}. Thus, since the two are connected sequentially, also the intermediate steps are distributed as $\bar{\pi}(\mathbf{z})$, after a burn-in period.

Table 5. State Dependent PMH (SD-PMH)

1. Choose a initial state $\mathbf{x}_0$, the total number of iterations K.
2. For $k = 1, \ldots, K$:
   - (a) Using a proposal pdf of type

     $$q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1})q_2(s_2|s_1, x_{1:2,k-1}) \cdots q_D(s_D|s_{1:D-1}, x_{1:D,k-1}), \qquad (46)$$

     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 ($0 \leq R \leq K$),

     $$d_1 < d_2 < \ldots < d_R.$$

     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. (11) or $\widetilde{Z}_X$ as in Eq. (12).
   - (b) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, ..., N$.
   - (c) Draw $N-1$ particles $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N-1)}$ via SIR using $q(\mathbf{z}|\mathbf{x}^*)$ as in Eq. (46), applying resampling at the same iterations, $d_1 < d_2 < \ldots < d_R$, 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 $\rho_D^{(i)} = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)}|\mathbf{x}^*)}$, $i = 1, \ldots, N$.
   - (e) Set $\mathbf{x}_k = \mathbf{x}^*$ with probability

     $$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}_X}{\widehat{Z}_Z},$$

     otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$.