A unifying view of multiple-try Metropolis and particle Metropolis-Hastings algorithms

L. Martino⋆,
⋆ Universitá degli studi di Catania, Italy.

February 8, 2026

Abstract

Markov chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) methods are cornerstone techniques for Bayesian inference and stochastic optimization. The multiple-try Metropolis (MTM) algorithm generalizes the Metropolis-Hastings (MH) scheme by selecting the next state from a set of weighted candidates, improving exploration of the state space. Particle Metropolis-Hastings (PMH) integrates MCMC and SMC ideas to efficiently tackle high-dimensional targets with sequentially factorized structures, embedding a particle filter within an MH framework. While both approaches have been extensively studied, particularly for state-space models, their relationship has not been fully explored. In this work, we examine the connections and distinctions between MTM and PMH schemes, which motivates the design of novel, highly efficient algorithms for filtering and smoothing. Among these, we introduce a particle multiple-try Metropolis (P-MTM) method, which demonstrates excellent performance across a range of numerical experiments.

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

1 Introduction

Monte Carlo methods are fundamental tools for numerical inference and stochastic optimization [38, 23]. In particular, Markov Chain Monte Carlo (MCMC) [17, 21, 22, 38] and Sequential Monte Carlo (SMC), also known as particle filtering [2, 7, 11, 32], are widely used to approximate integrals involving analytically intractable target probability density functions, typically Bayesian posterior distributions [23, 22]. MCMC methods generate samples from a Markov chain with the target distribution as its invariant measure, whereas SMC methods approximate the target distribution using a set of weighted particles.

The multiple-try Metropolis (MTM) algorithm [23, 24, 25] is an extension of the Metropolis-Hastings (MH) method [31, 15] in which the next state is selected from a set of candidate

points using suitable weights. By considering multiple proposals, MTM allows larger moves without significantly reducing the acceptance rate, thereby improving exploration of the state space. A well-known special case is the orientational bias Monte Carlo method used in molecular simulations [12]. Numerous extensions of MTM have been proposed, including schemes with correlated candidates, generalized weighting strategies, and adaptive or interacting proposals [37, 29, 20, 30, 34, 45, 4], as well as related approaches based on multiple auxiliary variables [3, 41, 28]. Furthermore, MTM methods have attracted considerable recent interest in the literature, with new developments focused on theoretical understanding, improved proposal and weighting strategies, and broader algorithmic frameworks that enhance performance and applicability [8, 13, 44, 36, 43].

The class of particle MCMC (P-MCMC) methods has been introduced to address inference in state-space models by combining Sequential Monte Carlo and MCMC techniques [1, 6, 33, 42]. In this work, we focus on the particle Metropolis-Hastings (PMH) algorithm [1, 6], which uses a particle filter to construct an approximation of the target distribution that serves as a proposal within a Metropolis-Hastings framework. A related variant, known as Particle Marginal Metropolis-Hastings, enables joint inference of latent states and static parameters [42]. The standard PMH scheme can also be interpreted as a particle smoothing method, since each iteration exploits weighted particle trajectories to update the posterior using all available observations [10, 11, 14, 18, 19, 40]. In [1], the authors discuss the connections between P-MCMC and related methods, including a detailed link to configurational bias Monte Carlo [39, 23], which is closely related to the MTM framework, and a brief mention of MTM itself [24]. However, the relationship between MTM and Particle Metropolis-Hastings (PMH) has not been fully explored.
In this work, we show that PMH can be interpreted as a specific MTM scheme with an independent proposal that generates correlated candidates sequentially via a particle filter. To clarify this connection, we revisit batch and sequential importance sampling and discuss marginal likelihood estimation [10], and we introduce an MTM variant with independent proposals. These elements are key to establishing the formal link between MTM and PMH. This unified analysis enables the design of more efficient algorithms by leveraging existing results on both MTM and PMH [41, 30, 33, 42]. Accordingly, we propose several new MTM- and PMH-based schemes. Among them, the particle MTM (P-MTM) algorithm combines PMH and MTM kernels, exploiting the sequential generation of candidates in PMH together with the multiple-try mechanism of MTM. Numerical results demonstrate the strong performance of P-MTM. In particular, when applied as a particle smoother to a stochastic volatility model, it yields highly accurate inference of the latent state sequence.
The paper is organized as follows. Section 2 introduces the main notation, and Section 3 reviews key concepts in importance sampling and resampling. MTM methods are presented in Section 4, followed by PMH algorithms and their connection to MTM in Section 5. Section 6 introduces the proposed novel schemes, Section 7 reports numerical simulations, and Section 8 concludes the paper.

2 Preliminaries and main notation

In many applications, we aim to infer an unknown parameter vector $\mathbf{x} \in \mathbb{R}^{D\times\zeta}$ from observed data $\mathbf{y} \in \mathbb{R}^{d_Y}$. The goal is typically to approximate moments of the posterior density $\bar{\pi}(\mathbf{x}|\mathbf{y})$, hereafter, we simply denote as $\bar{\pi}(\mathbf{x})$. Here, we write

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

with $x_d \in \mathcal{X} \subseteq \mathbb{R}^\zeta$ for $d = 1, \ldots, D$. The target density is

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

where Z_D is the marginal likelihood (or Bayesian evidence). In practice, we often know $\pi(\mathbf{x})$ but not Z_D, and direct sampling from $\bar{\pi}(\mathbf{x})$ is generally infeasible. Monte Carlo methods address this by introducing a simpler proposal density $q(\mathbf{x})$, supported on $\mathcal{X}$,[1] from which candidate samples are drawn. These candidates are then weighted and filtered to produce a particle approximation of $\bar{\pi}(\mathbf{x})$ and an estimate of Z_D.

3 Importance sampling (IS)

3.1 Batch Importance Sampling

Importance sampling (IS) is a fundamental Monte Carlo method for approximating a target distribution $\pi(\mathbf{x})$ using weighted samples. Given N samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ drawn from a proposal density $q(\mathbf{x})$, the corresponding importance weights are

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

where the superscript n indexes the particle and the subscript D denotes the dimension of $\mathbf{x} = x_{1:D} = [x_1, \ldots, x_D]$. The normalized weights are defined as

$$\bar{w}_D^{(n)} = \frac{w_D^{(n)}}{\sum_{i=1}^N w_D^{(i)}}, \tag{2}$$

allowing a particle approximation of the target measure,

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

and an estimator of the normalizing constant Z_D,

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

[1] For simplicity, we assume $q(\mathbf{x})$ is normalized, $\int_{\mathcal{X}} q(\mathbf{x})d\mathbf{x} = 1$.

3.2 Sequential Importance Sampling (SIS)

In high-dimensional problems, $\mathbf{x} \in \mathcal{D} = \mathcal{X}^D \subseteq \mathbb{R}^{D\times\zeta}$, a sequential approach, called sequential importance sampling (SIS), is often preferable when can be applied. Recall that $\mathbf{x} = x_{1:D} = [x_1, ..., x_D]$, we can observe that a target probability density function (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 [35] where $\gamma_1(x_1)$ is a marginal pdf and $\gamma_d(x_d|x_{1:d-1})$ are conditional pdfs. In many applications, the target appears directly decomposed as in Eq. (5), e.g., as in state-space models. We also consider the joint probability of the partial vector $x_{1:d} = [x_1, \ldots, x_d]$ (i.e., a partial posterior),

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

where

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

and, clearly, we have $\bar{\pi}_D(x_{1:D}) = \bar{\pi}(\mathbf{x})$. Similarly to the target distribution in Eq. (5), a sequentially factorized proposal density can be designed and employed by the user, i.e.,

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

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

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

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 partial 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

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

with the convention $\widehat{Z}_0 = 1$. $\widehat{Z}_0 = 1$. A alternative derivation of the (final) estimator $\widetilde{Z}_D$ is given in Appendix A.

Remark 1. *In SIS, there exist two equivalent formulations for estimating the normalizing constants Z_d. The first is the straightforward average of the recursive weights, $\widehat{Z}_d$, given in Eq.* (11)*, while the second is the product form, $\widetilde{Z}_d$, presented in Eq.* (12)*. Both estimators yield identical results, i.e.,*

$$\widehat{Z}_d = \widetilde{Z}_d, \quad d = 1,\ldots,D,$$

but the product form $\widetilde{Z}_d$ is often preferred in practice due to its numerical stability and natural sequential interpretation.

3.3 Sequential Importance Resampling (SIR)

Sequential Importance Resampling (SIR) [23, 38] extends sequential importance sampling (SIS) by incorporating *resampling* steps [7, 9]. In SIR, the sequential construction of importance weights proceeds as in SIS, but whenever a pre-specified criterion is met [7, 9, 26] -for example, a low effective sample size- the particles are resampled. Specifically, N independent particles are drawn according to the discrete probability mass defined by $\widehat{\pi}_d(x_{1:d})$, the current particle approximation. After resampling, all particles are assigned equal weights, and the resulting set is then propagated to construct the next approximation $\widehat{\pi}_{d+1}(x_{1:d+1})$.

More specifically, let us suppose a resampling step occurs at iteration d. Then, N particles $x_{1:d}^{(j)}$ are drawn from $\widehat{\pi}_d(x_{1:d})$, and their weights are reset to the uniform value $1/N$ [7, 9], ensuring that the particle population maintains diversity while approximating the target distribution sequentially. A proper choice [27] is to define the unnormalized importance weights as

$$w_d^{(n)} = \widehat{Z}_d, \quad \forall j = 1,\ldots,N. \tag{13}$$

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 $j = 1,\ldots,N$.

Remark 2. *One reason why this is a good choice, for instance, is that defining the following weights*

$$\xi_d^{(n)}) = \begin{cases} w_d^{(n)}, & \textit{without resampling at d-th iteration,} \\ \widehat{Z}_d, & \textit{with resampling at d-th iteration.} \end{cases} \tag{14}$$

then, in any case, we properly recover the estimation of the marginal likelihood, $\frac{1}{N}\sum_{n=1}^{N}\xi_d^{(n)} = \widehat{Z}_d$*, as expected.*

Therefore, the weight recursion for SIR becomes

$$\xi_d^{(n)} = \xi_{d-1}^{(n)}\beta_d^{(n)}, \tag{15}$$

where

$$\xi_{d-1}^{(n)} = \begin{cases} \xi_{d-1}^{(n)}, \text{ without resampling at } (d-1)\text{-th iteration,} \\ \widehat{Z}_{d-1}, \text{ with resampling at } (d-1)\text{-th iteration} \end{cases} \tag{16}$$

See Appendix A for further details.

Remark 3. *With the recursive definition of the weights* $\xi_d^{(n)}$ *in Eqs.* (15)-(16)*, the two estimators*

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

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 *[27].*

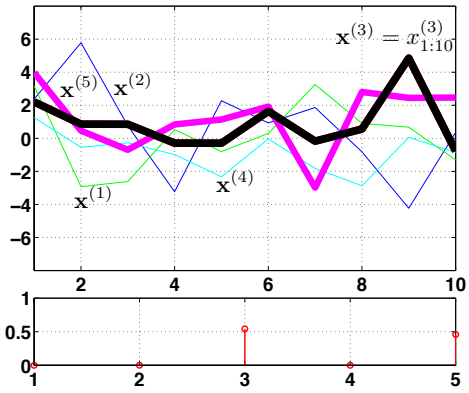
Remark 3 is necessary to to describe exhaustively the relationship between MTM and PMH algorithms. As an example, if the resampling is applied at each iteration and applying the suggested proper weighting of a resampled particle in Eqs. (15)-(16), the two possible estimators become

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

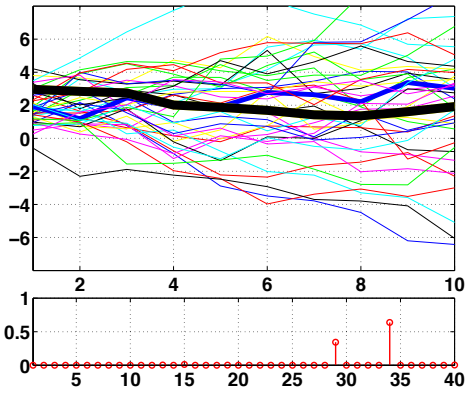
and

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

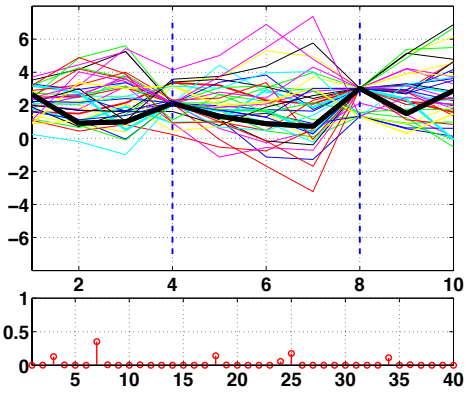
and hence coincide. Note that surprisingly, w.r.t. the estimator in Eq. (11) (for SIS, i.e., without resampling), the operations of product and sum in Eqs. (18)-(19) are inverted.
Figure 1 depicts different examples of generation of weighted samples $\mathbf{x}^{(n)}$ with or without employing resampling steps. More specifically, Figure 1 shows the components $x_1,^{(n)}\ldots, x_D^{(n)}$ of each sample, with $D = 10$. The line-width of each path is proportional to the corresponding weight $\bar{w}_n$. Below, we recall the MTM schemes and discuss a novel suitable variant in order to link MTM to PMH.



(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 $N = 40$).

Figure 1: 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 Multiple Try Metropolis (MTM) algorithms

The Multiple Try Metropolis (MTM) algorithm [24] 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 [23, 38]. We consider importance weights for facilitating the comparison with other techniques. However, different kind of weights could be applied [24, 30]. The MTM method generates a reversible Markov chain that converges to $\bar{\pi}(\mathbf{x})$ [24, 30].

4.1 MTM with an independent proposal density (I-MTM)

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 1 and 1 can be removed in the MTM scheme. Namely, one does not need to generate the auxiliary samples at step 1. Indeed, in this case, we could directly set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \ldots, N-1$. The simplified MTM algorithm (I-MTM) is given in Table 1. A graphical representation of a MTM scheme is provided in Figure 2, with $D = 1$ and $N = 2$.

Table 1: **Generic MTM algorithm.**

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 $\mathbf{x}^{(i)} \sim q(\mathbf{x}|\mathbf{x}_{k-1})$, $i = 1, \ldots, N$.

b) Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights

$$w_D^{(i)} = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)}|\mathbf{x}_{k-1})}, \quad i = 1, \ldots, N.$$

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 $\mathbf{z}^{(j)} \sim q(\mathbf{x}|\mathbf{x}^*)$, $j = 1, \ldots, N-1$, and set $\mathbf{z}^{(N)} = \mathbf{x}_{k-1}$.

d) Compute the importance weights also for the auxiliary points,

$$\rho_D^{(i)} = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)}|\mathbf{x}^*)}, \quad i = 1, \ldots, N.$$

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

4.2 Alternative version of the I-MTM method (I-MTM2)

An alternative formulation of the Independent multiple-try Metropolis (I-MTM) method can be provided, which will see to be "closer" to a particle MH scheme [25]. When the proposal density is independent of the current state, each candidate $\mathbf{x}^{(j)}$, $j = 1, \ldots, N-1$, can be directly used as an auxiliary point, i.e., we may set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, since all candidates are drawn independently from $q(\mathbf{x})$. By the same reasoning, it is also possible to use the samples generated in the previous iteration as auxiliary points, as they are likewise independent draws from the same proposal. A summary of this alternative I-MTM formulation is provided in Table 3. Note that, in this case,

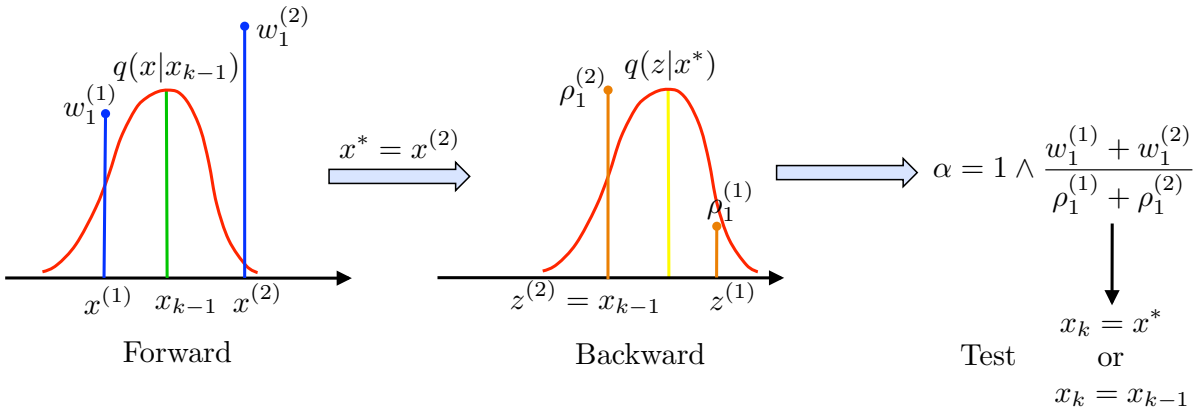


Figure 2: Sketch of a generic MTM method with $D = 1$ and $N = 2$ tries. In this example, the second candidate is selected as $x^* = x^{(2)}$. It has been selected with probability $\bar{w}_1^{(1)} = \frac{w_1^{(1)}}{w_1^{(1)}+w_1^{(2)}}$. The auxiliary points are $z^{(1)} \sim q(z|x^*)$ and $z^{(2)}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

we can write the acceptance probability α as

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

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

5 Particle Metropolis-Hastings (PMH) algorithm and its relationship with MTM

Assume that the target density admits the factorization

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

The PMH algorithm [1] is an MCMC method developed independently of the MTM framework, specifically tailored for such sequentially structured targets. A detailed description of the method is provided in Table 4. At each iteration of PMH, a particle filter is executed to generate N weighted particles approximating the target measure. One particle is then selected through a resampling step and proposed as the next state of the Markov chain. The resampled particle is then accepted or rejected according to a MH-type acceptance probability, which involves two possible estimators of the normalizing constant 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. (13) [27]. The PMH algorithm can be interpreted as a particle smoother, since the outputs of each particle filter run (that is, the N weighted paths generated via a SIR procedure) are further processed through the MH acceptance mechanism, thereby incorporating all

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

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

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

$$\begin{aligned} \alpha &= 1 \quad \wedge \quad \frac{\sum_{i=1}^N w_D^{(i)}}{\sum_{i=1}^N w_D^{(i)} - w^* + w_{D,k-1}} \\ &= 1 \quad \wedge \quad \frac{\sum_{i=1}^N w_D^{(i)}}{\sum_{i=1}^N \rho_D^{(i)}}, \end{aligned} \tag{20}$$

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

available observations into the updated posterior approximation. An extension of PMH, designed to jointly handle dynamic latent states and static model parameters, is known as particle marginal Metropolis-Hastings (PMMH) algorithm, and is described in Appendix B.

5.1 Relationship between MTM and PMH

A comparison between the alternative I-MTM2 scheme, introduced in Section 4, and thePMH algorithm reveals a close relationship between the two methods. In fact, the overall structure of the algorithms is strikingly similar. The main connections and distinctions can be summarized as follows:

- The primary distinction between I-MTM2 and PMH lies in how the candidates are generated. In PMH, candidates are produced sequentially through a sequential Monte Carlo (SMC) procedure, i.e., by a particle filter, whereas I-MTM2 generates all components of each candidate simultaneously in a batch manner. If resampling steps are omitted in the SMC procedure, the two algorithms become *identical*: the only difference is whether the

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 $\widehat{Z}^{(k)} = \widehat{Z}^{(k-1)}$.

candidates are constructed sequentially or all at once. Concretely, I-MTM2 draws each candidate $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_D^{(i)}]$ directly from a proposal density $q(\mathbf{x})$ defined over the full space $\mathbf{x} \in \mathcal{D}$, while PMH draws each component $x_d^{(i)}$ sequentially from the conditional proposal $q_d(x_d|x_{1:d-1}^{(i)})$. Figure 4 provides a schematic illustration of this relationship.

- The key distinction between PMH and I-MTM2 lies in the use of resampling during candidate generation. In PMH, resampling introduces dependencies among the proposed candidates $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, whereas in the MTM schemes considered here, candidates are generated independently. If resampling is omitted, the PMH samples $\mathbf{x}^{(i)} = x_{1:D}^{(i)}$ become independent, just as in I-MTM2. While correlated candidate generation can also be incorporated into MTM methods without jeopardizing the ergodicity of the chain [5], PMH naturally produces such correlations through its sequential resampling steps. Consequently, PMH can be interpreted as an I-MTM2 algorithm in which candidates are both sequentially generated and correlated, effectively combining the features of sequential construction and dependent proposals.
  To illustrate this point, Figure 1 displays particles weighted according to their importance sampling (IS) weights, with the line width of each path proportional to the corresponding normalized weight $\bar{w}_n$. Each particle is represented as a sequence of its components $x_d^{(n)}$, $d = 1, \ldots, D = 10$, for $n = 1, \ldots, N$, with $N \in \{5, 40\}$. The target distribution is a 10-dimensional multivariate Gaussian, $\bar{\pi}(\mathbf{x}) = \prod_{d=1}^{10} \mathcal{N}(x_d|2, 1/2)$, with mean $\mu_d = 2$ for all $d = 1, \ldots, 10$. Figures 1(a) and (b) correspond to IS or sequential IS (SIS) without resampling, using two different proposal densities. In Figure 1(a), the components

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 3.3) 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
   $$\alpha = 1 \quad \wedge \quad \frac{\widehat{Z}^*}{\widehat{Z}^{(k-1)}}, \tag{21}$$
   otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widehat{Z}^{(k)} = \widehat{Z}^{(k-1)}$.

$x_d^{(n)}$ are independent both across d and n. In Figure 1(b), the components within each particle $\mathbf{x}^{(n)}$ are correlated, while the particles themselves remain independent. Figure 1(c) shows the effect of applying two resampling steps at iterations $d = 4$ and $d = 8$, which introduces correlation among the particles $\mathbf{x}^{(n)}$, $n = 1, \ldots, N$, in addition to the intra-particle correlations. This scenario corresponds to the candidate generation process in the PMH algorithm.

- In their standard formulations, I-MTM2 employs the estimator $\widehat{Z}_D$ from Eq. (4), whereas PMH is typically presented using $\widetilde{Z}_D$, defined in Eq. (12). Despite this difference, both estimators are equivalent representations of the normalizing constant Z_D (see Remark 3), provided that resampled particles are properly weighted [27].

- If the the target can be factorized as in Eq. (5), the PMH algorithm can be applied and is particularly advantageous in high-dimensional settings, because the incorporation of resampling steps improves the efficiency of candidate generation and avoid the sample degenaracy.

6 Novel schemes

The considerations discussed above also motivate the design of novel PMH schemes. For example, one can introduce an alternative acceptance probability,

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

We denote by var-PMH the variant of PMH that employs this probability α in place of the standard acceptance probability given in Eq. (21). In other words, var-PMH follows the same procedure as the PMH algorithm in Table 4, with Eq. (21) replaced by Eq. (22). Within a sequential framework, the structure of var-PMH is equivalent to the I-MTM method described in Table 2, in the same way that the standard PMH in Table 4 corresponds to I-MTM2 in Table 3.

State Dependent PMH (SD-PMH). The standard PMH algorithm can be further extended by employing a *state-dependent* proposal density (i.e., a proposal that depends on the previous state) rather than the independent proposal used in Table 4. This extended scheme, referred to as *SD-PMH*, is outlined in Table 5. In SD-PMH, generating a backward path is required at step 5, introducing additional computational cost. However, these backward paths could potentially be reused to improve the estimation of hidden states, although a detailed analysis of this application is beyond the scope of this work.
The validity of SD-PMH is easily guaranteed since it exactly corresponds to an MTM scheme (see Table 1). In SD-PMH, the particle approximation $\widehat{\pi}_D$ involves correlated samples due to resampling, unlike in standard MTM. Nevertheless, this correlation does not compromise ergodicity, as demonstrated in [5]. Resampling steps can be applied selectively at $0 \leq R \leq K$ pre-determined iterations $d_1, \ldots, d_R$. If $R = 0$, no resampling is applied, and the algorithm reduces to a standard MTM scheme with sequential candidate generation. Conversely, if $R = K$, resampling occurs at every iteration, corresponding to a bootstrap filter for particle generation [7, 10]. Figure 3(a) provides a schematic overview of the schemes discussed in this work, with MTM variants shown on the left and their corresponding PMH approaches on the right; boxes with dashed contours indicate the novel schemes introduced here. As an example of a state-dependent proposal in SD-PMH, one 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 factorizes as

$$q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1}) \prod_{d=2}^{D} q_d(s_d|s_{d-1}, x_{d,k-1}).$$

However, selecting and properly tuning the components $q_d(s_d|s_{d-1}, x_{d,k-1})$ is nontrivial and requires careful consideration.

Particle Multiple Try Metropolis (P-MTM) algorithm. A simple yet robust scheme

can be constructed by alternating between a standard PMH kernel (or var-PMH), denoted by $K_{PMH}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, and an MTM kernel with a random-walk proposal density, denoted by $K_{MTM}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. This alternating strategy defines the *particle Multiple-Try Metropolis* (P-MTM) algorithm, which proceeds according to 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 combined in such a way that, since each individually preserves the target distribution $\bar{\pi}$, their composition also leaves $\bar{\pi}$ invariant, as shown in Appendix C. The P-MTM algorithm leverages the strengths of both approaches: it sequentially selects N particles component by component using resampling steps, as in PMH, while simultaneously allowing the generation of N candidates that take into account the previous state of the chain, as in MTM. Figure 3(b) provides a graphical illustration of the P-MTM scheme, and Figure 4 presents a schematic of the two complementary candidate generation strategies employed within P-MTM.

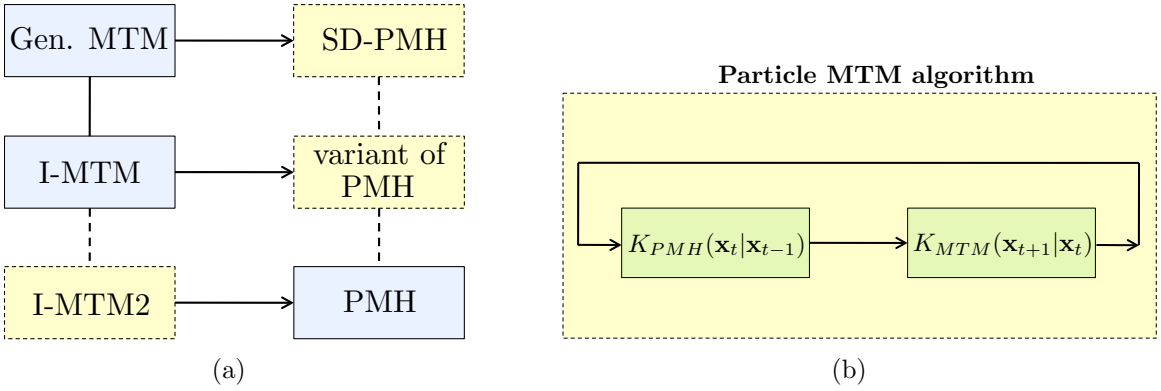
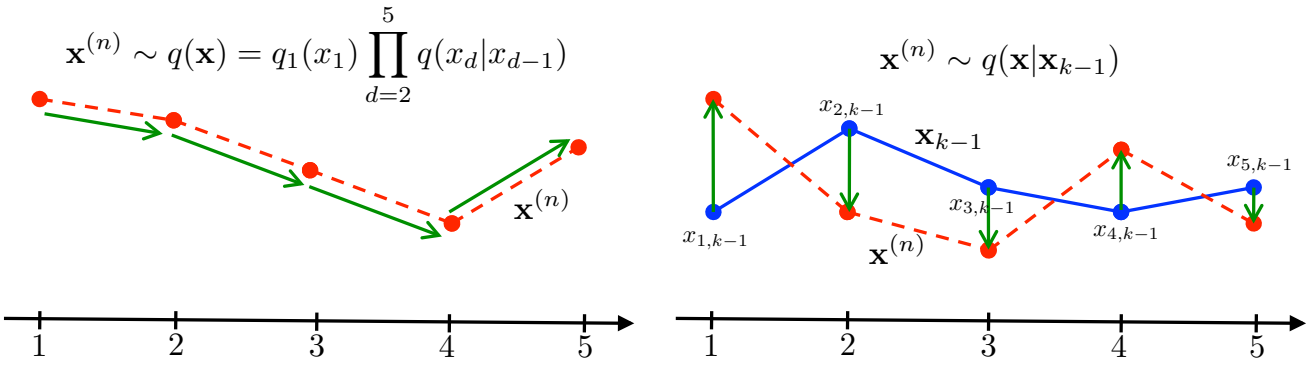


Figure 3: **(a)** Graphical representation of the MTM methods and the corresponding PMH schemes. The boxes with dashed contours contain the novel schemes presented in this work. **(b)** Graphical sketch of the P-MTM algorithm.



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

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

Figure 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).

7 Numerical Simulations

7.1 Comparison among different particle schemes

To evaluate and compare the different techniques, we consider a 10-dimensional Gaussian target density

$$\bar{\pi}(\mathbf{x}) = \bar{\pi}(x_1, \ldots, x_D) = \prod_{d=1}^{D} \mathcal{N}(x_d \mid \mu_d, \sigma^2), \tag{23}$$

with $\mathbf{x} = x_{1:D} \in \mathbb{R}^D$, $D = 10$, means $\mu_{1:3} = 2$, $\mu_{4:7} = 4$, $\mu_{8:10} = -1$, and standard deviation $\sigma = 1/2$. The methods tested include I-MTM, I-MTM2, PMH, var-PMH (using the acceptance probability in Eq. (22)), and P-MTM for estimating the vector $\mu_{1:10}$.

For all methods, sequential Gaussian proposals are used for candidate generation:

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

Within P-MTM, the random-walk MTM component uses

$$q_{rw}(\mathbf{x} \mid \mathbf{x}_{k-1}) = \prod_{d=1}^{D} \mathcal{N}(x_d \mid x_{d,k-1}, \sigma_{rw}^2), \qquad \sigma_{rw} = 1,$$

where $\mathbf{x}_{k-1} = [x_{1,k-1}, \ldots, x_{D,k-1}]$ denotes the previous state of the chain. For all PMH-based schemes, resampling is applied at every iteration, whereas no resampling is applied in I-MTM and I-MTM2. We investigate the effect of varying both the number of particles N and the number of iterations K. The mean squared error (MSE) in estimating $\mu_{1:10}$ is computed over 500 independent runs. The initial particles $x_1^{(i)}$ for each run and method are sampled from $\mathcal{N}(-2, 4)$, $i = 1, \ldots, N$.

Figures 5(a) and (b) show the MSE as a function of the number of iterations K (semilog scale) for a fixed number of candidates $N = 3$. Figure 5(a) displays the MTM schemes, while Figure 5(b) shows the PMH-based methods. Figure 5(c) depicts the MSE of PMH methods as a function of N.

The results indicate that using the acceptance probability of type Eq. (22) leads to lower MSE, particularly for small N, confirming that var-PMH outperforms standard PMH under these conditions. As N increases, the performance of PMH and var-PMH converges, since the acceptance probability approaches 1 in both cases. The P-MTM algorithm exhibits excellent performance, outperforming all other schemes, with the MSE approaching zero as N grows, confirming the effectiveness of the proposed approach. The results also suggest that applying resampling at every iteration is not always optimal; selectively applying resampling at a subset of iterations can improve performance [7, 10]. Figure 5(d) illustrates 35 different states $\mathbf{x}_k = x_{1:10,k}$ at various iterations k, obtained with var-PMH ($N = 1000$, $K = 1000$), along with the true values $\mu_{1:10}$ shown as dashed lines.

7.2 Inference in a stochastic volatility model

We consider a stochastic volatility (SV) model in which the hidden state $x_d \in \mathbb{R}$ at time d follows an AR(1) process and represents the log-volatility of a financial time series [16]. The model is defined as

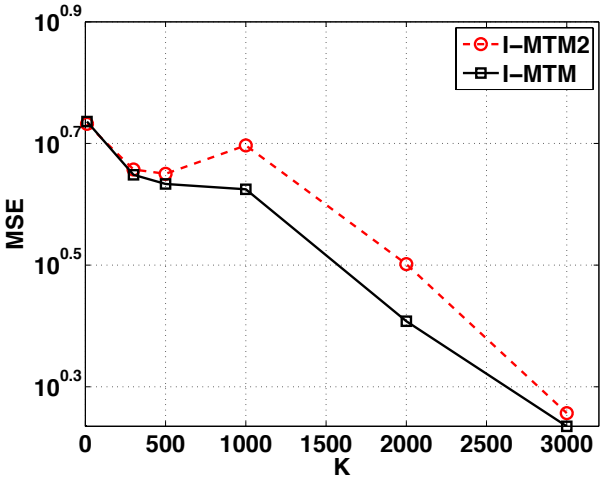
$$\begin{cases} x_d = \alpha x_{d-1} + u_d, \\ y_d = \exp\left(\frac{x_d}{2}\right) v_d, \end{cases} \qquad d = 1, \dots, D, \tag{24}$$

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

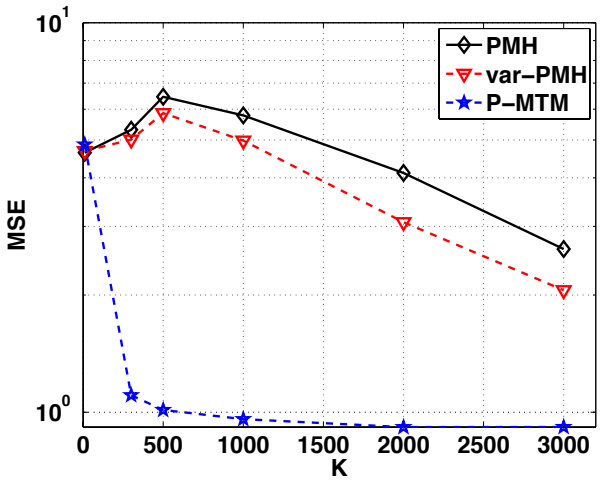
Given a sequence of observations $y_{1:D} \in \mathbb{R}^D$, our goal is to infer the hidden states $x_{1:D}$ by analyzing the joint posterior $\bar{\pi}(x_{1:D} \mid y_{1:D})$. Classical particle filtering (PF) provides a Monte Carlo approximation of this posterior, and by extension, of the marginal posteriors $\bar{\pi}(x_d \mid y_{1:D})$, which are used for smoothing. However, for early time points ($d \ll D$), standard PF often performs poorly: the marginal posterior $\bar{\pi}(x_d \mid y_{1:d})$ plays a privileged role in filtering since it is better characterized than other marginals, as noted in [10]. PMH schemes can improve the approximation of both the complete posterior $\bar{\pi}(x_{1:D} \mid y_{1:D})$ and the marginal posteriors $\bar{\pi}(x_d \mid y_{1:D})$. Conceptually, PMH can be viewed as combining multiple independent runs of a particle filterafter observing all $y_{1:D}$to generate an ergodic Markov chain whose invariant distribution is $\bar{\pi}(x_{1:D} \mid y_{1:D})$.

We evaluate the standard PMH algorithm [1] and the particle P-MT) method introduced in Section 6. For the sequential proposal, both methods employ a bootstrap particle filter, using the model transition $p(x_d \mid x_{d-1})$ as the proposal and performing resampling at every iteration. Within the MTM component of P-MTM, we use a random-walk Gaussian proposal

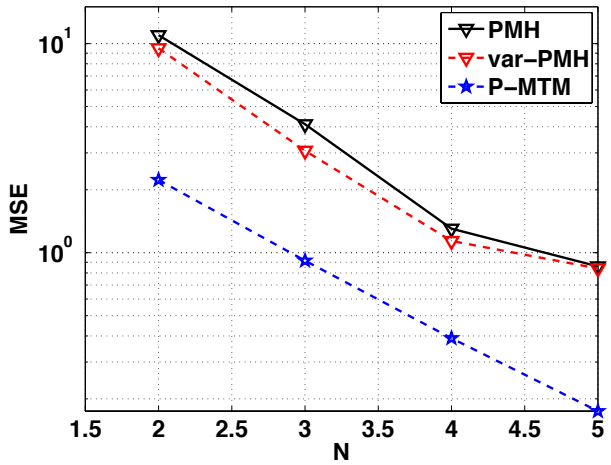
$$q(\mathbf{x} \mid \mathbf{x}_{k-1}) = \prod_{d=1}^{D} \mathcal{N}(x_d \mid x_{d,k-1}, \sigma_p^2),$$



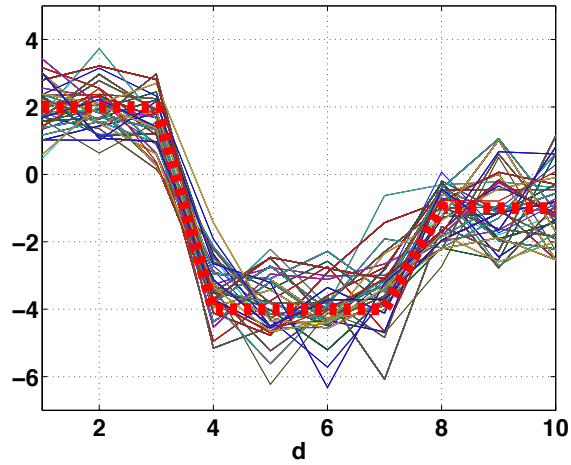
(a) MTM schemes.



(b) PMH schemes.



(c) PMH schemes.

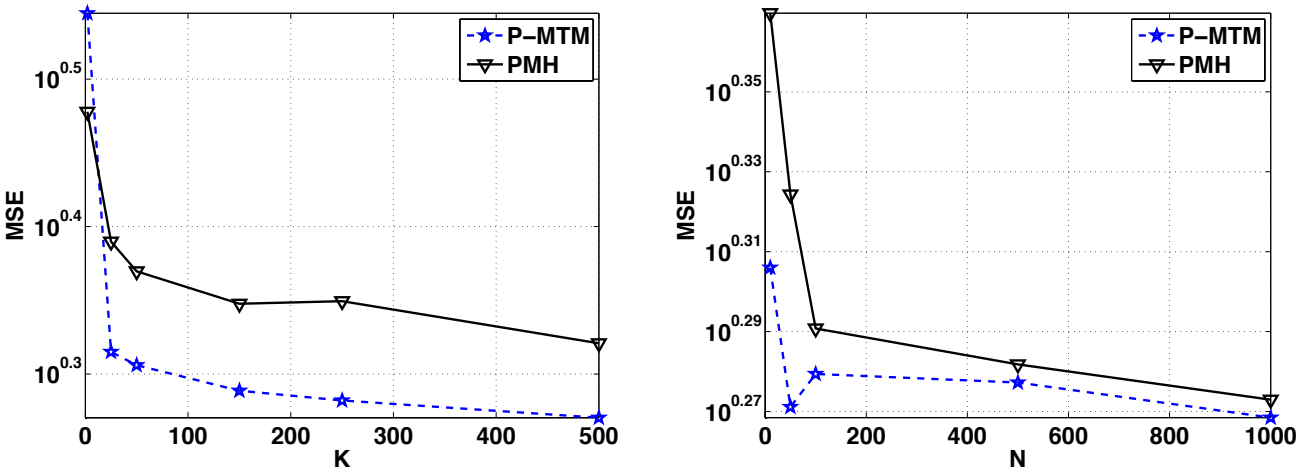


(d) Different States of var-PMH.

Figure 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 $\mu_{8:10} = -1$).

with $\sigma_p = 0.5$. Results are averaged over 500 independent runs, with a new sequence of observations $y_{1:D}$ generated in each run, setting $D = 100$. Both schemes are compared using the same total number of posterior evaluations: for a given N, PMH runs K iterations, while

P-MTM alternates $\frac{K}{2}$ PMH steps and $\frac{K}{2}$ MTM steps (see Section 6). First, we fix $N = 10$ and vary the total number of iterations K from 2 to 500, with results shown in Fig. 6(a). Next, we fix $K = 50$ and test various values of N from 10 to 1000, shown in Fig. 6(b). In all cases, P-MTM consistently outperforms standard PMH, achieving lower mean squared error (MSE), except for a very small number of iterations ($K = 2$) where PMH performs slightly better. These results confirm the improved efficiency and accuracy of the proposed P-MTM approach in smoothing applications.



(a) MSE versus K.

(b) MSE versus N.

Figure 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 ($K = 50$).

8 Conclusions

In this work, we have emphasized the strong connection between MTM and PMH algorithms. In particular, PMH can be viewed as an MTM scheme that employs correlated candidates, generated and weighted sequentially through a particle filter. Leveraging this insight, we have proposed novel MTM and PMH variants that effectively combine the strengths of both approaches. Notably, the particle Multiple-Try Metropolis (P-MTM) algorithm integrates the sequential candidate construction of PMH with the ability of MTM to perturb the previous state of the chain via a random-walk proposal. P-MTM has proven to be a highly efficient method for both filtering and smoothing in state-space models. Through extensive numerical experiments, P-MTM consistently outperformed alternative techniques, demonstrating its robustness and accuracy. As a future direction, we plan to develop a marginal version of P-MTM for jointly inferring dynamic and static parameters within state-space frameworks.

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] R. Doig and L. Wang. A unified framework for multiple?try Metropolis algorithms. *arXiv:2503.11583*, 2025.

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

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

[11] 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.

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

[13] P. Gagnon, F. Maire, and G. Zanella. Improving multiple?try Metropolis with local balancing. *Journal of Machine Learning Research*, 24:1–59, 2023.

[14] 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.

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

[16] 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.

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

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

[19] 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.

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

[21] 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.

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

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

[24] 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.

[25] L. Martino. A review of multiple try MCMC algorithms for signal processing. *Digital Signal Processing*, 75:134–152, 2018.

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

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

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

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.

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

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

[36] F. Pozza and G. Zanella. On the fundamental limitations of multi-proposal Markov chain Monte Carlo algorithms. *arXiv:2410.23174*, 2024.

[37] 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.

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

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

[40] 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.

[41] 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.

[42] 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.

[43] Z. Wang and J. Yang. Stereographic multi?try Metropolis algorithms for heavy?tailed sampling. *arXiv:2505.12487*, 2025.

[44] X. Yang and J. S. Liu. Convergence rate of multiple?try Metropolis independent sampler. *Statistics and Computing*, 33:79, 2023.

[45] 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,

$$Z_d = \int_{\mathcal{X}^d} \pi_d(x_{1:d}) dx_{1:d} \approx \widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)}, \tag{25}$$

and

$$\begin{aligned} & \int_{\mathcal{X}^d} \gamma_d(x_d|x_{1:d-1}) \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} = \\ = & \int_{\mathcal{X}^d} \frac{\pi_d(x_{1:d})}{\pi_{d-1}(x_{1:d-1})} \bar{\pi}_{d-1}(x_{1:d-1}) dx_{1:d}, \\ = & \frac{Z_d}{Z_{d-1}}. \end{aligned} \tag{26}$$

Clearly, we can write

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

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),

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

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

$$\int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \widehat{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} = \tag{27}$$

$$\begin{aligned}
&= \sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \beta_d^{(n)}, \\
&= \frac{1}{\sum_{i=1}^{N} w_{d-1}^{(i)}} \sum_{n=1}^{N} w_{d-1}^{(n)} \beta_d^{(n)}, \\
&= \frac{1}{\sum_{i=1}^{N} w_{d-1}^{(i)}} \sum_{n=1}^{N} w_d^{(n)}, \\
&= \frac{\frac{1}{N}\sum_{n=1}^{N} w_d^{(n)}}{\frac{1}{N}\sum_{i=1}^{N} w_{d-1}^{(i)}} = \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} \approx \frac{Z_d}{Z_{d-1}},
\end{aligned} \tag{28}$$

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. (25). Finally, we can obtain, setting $\hat{Z}_0 = 1$,

$$\begin{aligned}
\widetilde{Z} &= \prod_{d=1}^{D} \frac{\widehat{Z}_d}{\widehat{Z}_{d-1}} = \widehat{Z}_1 \frac{\widehat{Z}_2}{\widehat{Z}_1} \cdots \frac{\widehat{Z}_{D-1}}{\widehat{Z}_{D-2}} \frac{\widehat{Z}_D}{\widehat{Z}_{D-1}}, \\
&= \prod_{d=1}^{D} \left[\sum_{i=1}^{N} \bar{w}_{d-1}(x_{1:d-1}^{(i)}) \beta_d(x_d^{(i)}|x_{1:d-1}^{(i)}) \right] \approx Z,
\end{aligned} \tag{29}$$

that is exactly the estimator in Eq. (12).

A.1 Application of resampling

Let us consider to approximate the integral in Eq. (27) 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

$$\begin{aligned}
&\int_{\mathcal{X}^d} \beta_d(x_d|x_{1:d-1}) q_d(x_d|x_{1:d-1}) \widehat{\pi}_{d-1}(x_{1:d-1}) dx_{1:d} \approx \\
\approx\ & \frac{1}{N} \sum_{n=1}^{N} \beta_d^{(n)}.
\end{aligned} \tag{30}$$

Moreover using Eq. (26), 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

$$\begin{cases} q_d(x_d|x_{d-1},\theta), \\ \ell_d(y_d|x_d,\theta) \end{cases} \tag{31}$$

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 3, 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

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

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 $k = 1,\ldots,K$ 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

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

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

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

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 $k = 1,\ldots,K$:
   (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

$$K_{PMTM}(\mathbf{z}|\mathbf{x}) = \int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) K_{PMH}(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \tag{32}$$

The target $\bar{\pi}$ is also invariant w.r.t. $K_{PMTM}(\mathbf{z}|\mathbf{x})$ [38, 23, 22]. Indeed, we can write

$$\begin{aligned}
&\int_{\mathcal{D}} K_{PMTM}(\mathbf{z}|\mathbf{x}) \bar{\pi}(\mathbf{x}) d\mathbf{x} = \\
= &\int_{\mathcal{D}} \left[\int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) K_{PMH}(\mathbf{y}|\mathbf{x}) d\mathbf{y} \right] \bar{\pi}(\mathbf{x}) d\mathbf{x}, \\
= &\int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) \left[\int_{\mathcal{D}} K_{PMH}(\mathbf{y}|\mathbf{x}) \bar{\pi}(\mathbf{x}) d\mathbf{x} \right] d\mathbf{y}, \\
= &\int_{\mathcal{D}} K_{MTM}(\mathbf{z}|\mathbf{y}) \bar{\pi}(\mathbf{y}) d\mathbf{y}, \\
= &\bar{\pi}(\mathbf{z}),
\end{aligned} \tag{33}$$

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

$$q(\mathbf{s}|\mathbf{x}_{k-1}) = q_1(s_1|x_{1,k-1}) \prod_{d=1}^{D} q_d(s_d|s_{1:d-1}, x_{1:d,k-1}), \tag{34}$$

we employ SIR (see Section 3.3) 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. (34), 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 $\mathbf{x}_k = \mathbf{x}_{k-1}$.