THE FUSS ALGORITHM: A FAST UNIVERSAL SELF-TUNED SAMPLER WITHIN GIBBS

L. Martino[1]*, H. Yang*[2]*, D. Luengo*[3]*, J. Kanniainen*[2]*, J. Corander*[1]

[1] Dep. of Mathematics and Statistics, University of Helsinki, 00014 Helsinki (Finland).
[2] Dep. of Industrial Engineering, Tampere University of Technology, Tampere (Finland).
[3] Dep. of Circuits and Systems Engineering, Universidad Politécnica de Madrid, 28031 Madrid (Spain).

ABSTRACT

Bayesian inference often requires efficient numerical approximation algorithms such as sequential Monte Carlo (SMC) and Markov chain Monte Carlo (MCMC) methods. The Gibbs sampler is a well-known MCMC technique widely applied in several fields (e.g., machine learning, finance, etc.). In the application of the Gibbs sampler one needs to efficiently generate values from univariate full-conditional distributions. In this work, we present a simple, self-tuned and extremely efficient MCMC algorithm which produces virtually independent samples from univariate target densities. The proposal density used is self-tuned and tailored to the specific target, but it is not adaptive. Indeed, the proposal is adjusted during an initialization stage following a simple procedure. As a consequence, there is no "fuss" about convergence or tuning, and the execution of the algorithm is remarkably sped up. Although it can be used as a stand-alone algorithm to sample from a generic univariate distribution, the proposed approach is particularly suited for its use within a Gibbs sampler, especially when sampling from spiky multi-modal distributions. Hence, we call it FUSS (Fast Universal Self-tuned Sampler). Numerical experiments on several data sets show its good performance in terms of speed and estimation accuracy.

Keyword: Markov Chain Monte Carlo; Gibbs sampling; non-parametric MCMC.

1. INTRODUCTION

Bayesian methods and their implementations by means of sophisticated Monte Carlo techniques [18], [30] have become very popular over the last two decades. Indeed, many practical problems demand procedures for drawing from probability distributions with non-standard forms, such as Markov chain Monte Carlo (MCMC) methods [5] and particle filters [3].

MCMC techniques generate samples from a target probability density function (pdf) by drawing from a simpler proposal pdf [17, 18], generating a Markov chain. The two most widely applied MCMC approaches are the Metropolis- Hastings (MH) algorithm and the Gibbs sampler [18, 30]. The Gibbs sampling technique is extensively used in Bayesian inference [14, 31, 25], to generate samples from multivariate target densities, drawing each component of the samples from univariate full-conditional densities.

Therefore, when the multivariate target can be factorized into univariate conditional pdfs, the key point for successful application of the Gibbs sampler is the ability to draw efficiently from these univariate pdfs [14, 18, 30]. The best scenario for Gibbs sampling occurs when exact samplers for each full- conditional are available. Otherwise, sampling techniques like rejection sampling (RS) or MH-type algorithms are used *within* the Gibbs sampler to draw from the complicated full-conditionals. In the first case, samples generated from the RS algorithm are independent, but the acceptance rate can be very low. In the second case, we have an approach where an MCMC, the MH method, is applied *inside* another MCMC, that is the Gibbs samplers. Therefore, the typical problems of the *external*-MCMC (long "burn-in" period, large correlation, etc.) could raise dramatically if the *internal*-MCMC is not extremely efficient. Although the Gibbs sampler needs only one sample from each full-conditional, in this case several iterations are necessary to avoid the "burn-in" period of the *internal*-MCMC. The length of the "burn-in" period is strictly related to the correlation among the samples. Higher correlation corresponds to a slower convergence of the chain.

Thus, several automatic and self-tuning samplers, such as *adaptive rejection sampling* (ARS) [6, 9], *Adaptive Rejection Metropolis Sampling* (ARMS) [7, 8, 24], *Independent Doubly Adaptive Rejection Metropolis Sampling* (IA2RMS) [23], and *Adaptive Sticky Metropolis* [21] have been proposed. All of these methods build an adaptive sequence of proposal pdfs via some interpolation procedure given a set of support points. The proposal is updated when a new support point is incorporated, according to some statistical criterion. Although their performance can be extremely good, the results show a dependence

from the initial set of support points.Another drawback is to ensure the ergodicity especially in the applications within Gibbs sampling [8, 30].

Other related works, where a non-adaptive proposal pdf is built via interpolation procedures can be found in literature [26], [32]. Furthermore, owing to their good performance, different kind of adaptive MH methods based on an independent proposal have been also studied [12, 13].

In this work, we present a novel algorithm that uses a different strategy: start with a huge number of support points and then remove some of them after a pruning step, according to certain conditions. The resulting method is fast and extremely efficient (it yields virtually independent samples), as shown in the numerical results, even with highly multimodal and complicated targets. The dependence on the initial set of points is drastically reduced, since algorithm only requires a large interval where it is considered that the main computational effort should be concentrated. Moreover, the proposal is self-tuned, during the initialization stage, but non-adaptive afterwards. Hence, ergodicity is not an issue and the convergence of the chain to the target distribution is always guaranteed. For these reasons, we call the new method as *FUSS algorithm* ("Fast Universal Self-tuned Sampler") since, with this sampler, there is no "fuss" about convergence or tuning. Furthermore, we introduce different pruning procedures: we discuss and analyze them from the theoretical point of view and also test the performance in the numerical simulations. We design an optimal pruning scheme based on the minimization of L_1 distance between proposal and target pdf. The FUSS algorithm is particularly advisable for multimodal spiky target densities, i.e., densities with several sharp and tight modes, where virtually all of the existent MCMC techniques often fail. This kind of target pdfs often appears in several applications, as in ecological and financial inference problems (see Section 6).

The rest of the paper is divided as follows. Sections 2-3 are devoted to recalling the general framework and describing the structure of the novel technique. Details about the proposal construction and generation are given in Section 4. Then, different pruning algorithms are introduced in Section 5. Section 6 provides numerical results on several different model types. Finally, Section 7 contains some brief final remarks.

2. PROBLEM STATEMENT

Bayesian inference often requires drawing samples from complicated multivariate posterior pdfs, $\pi(\mathbf{x}|\mathbf{y})$ with $\mathbf{x} \in \mathcal{X}^D \subseteq \mathbb{R}^D$. A common approach, when direct sampling from $\pi(\mathbf{x}|\mathbf{y})$ is unfeasible, is using a Gibbs sampler [30]. At the i-th iteration, a Gibbs sampler obtains the d-th component ($d = 1, \ldots, D$) of $\mathbf{x}$, x_d, by drawing from the full conditional pdf of x_d given all the previous generated components [30, 4, 15], i.e.,

$$x_d^{(i)} \sim \bar{\pi}(x_d|\mathbf{x}_{1:d-1}^{(i)}, \mathbf{x}_{d:D}^{(i-1)}) = \bar{\pi}(x_d) \propto \pi(x_d), \tag{1}$$

with $x_d \in \mathcal{X}$, and the initial vector drawn from the prior, i.e., $\mathbf{x}^{(0)} \sim \bar{\pi}_0(\mathbf{x})$. However, even sampling from the univariate pdf in Eq. (1) can often be complicated. In these cases, a common approach is to use another Monte Carlo technique (e.g., rejection sampling (RS) or the Metropolis-Hastings (MH) algorithms) within the Gibbs sampler, drawing candidates from a simpler proposal pdf,

$$\bar{p}(x) \propto p(x) = e^{W(x)}, \quad x \in \mathbb{R}.$$

The best case is when an RS technique can be applied since it yields independent and identically distributed (i.i.d.) samples. However, the RS technique requires that $p(x) \geq \pi(x)$ for all $x \in \mathcal{X}$. In general, it is not straightforward to satisfy this inequality. For instance, the adaptive rejection sampling (ARS) technique [9] can be applied only to log-concave target pdfs. Thus, in general, the use of another MCMC method becomes almost mandatory. In this case, the performance of this approach depends strictly on the choice of $\bar{p}(x)$. For sake of simplicity, in the sequel we denote the univariate target pdf (i.e., the full-conditional proposal in Eq. (1)) as $\bar{\pi}(x)$. Our aim is designing an efficient fast sampler to draw from the univariate target pdf,

$$\bar{\pi}(x) \propto \pi(x) = e^{V(x)}, \quad x \in \mathcal{X} \subseteq \mathbb{R}, \tag{2}$$

where $\pi(x)$ is unnormalized and $V(x) = \log[\pi(x)]$ is a generic function, so that $\pi(x)$ can be multimodal, with light or heavy tails. For the sake of simplicity, here we consider bounded target density $\pi(x)$, although the FUSS method, described in the following, could be easily extended for dealing with unbounded target as well.

3. STRUCTURE OF THE ALGORITHM

The FUSS algorithm is an MCMC based on an independent proposal pdf built via a simple interpolation procedure, shown in the next section. The general structure is given in Table 1. The first 3 steps form a pre-processing procedure, applied only once,

Table 1. General structure of the FUSS algorithm.

1. **Initialization:** Choose a set of support points $\mathcal{S}_M = \{s_1, \ldots, s_M\}$, such that $s_1 < s_2 < \ldots < s_M$, and the total number of samples is K.
2. **Pruning:** Remove certain support points according to a certain criterion, providing a new set $\mathcal{S}_m$ with $m < M$.
3. **Construction:** Build adequately a proposal function $p(x|\mathcal{S}_m)$ given $\mathcal{S}_m$.
4. **MCMC algorithm:** apply K steps of an MCMC method using $p(x|\mathcal{S}_m)$ as proposal pdf and yielding a set of samples $\{x_1, \ldots, x_K\}$.

Table 2. The Metropolis-Hastings method

3.1 Set $k = 0$ and choose x_0.

3.2 Draw $x' \sim \bar{p}(x) \propto p(x|\mathcal{S}_m)$ and $u' \sim \mathcal{U}([0, 1])$.

3.4 Set $x_{k+1} = x'$ with probability

$$\alpha_{MH} = 1 \wedge \frac{\pi(x')p(x_k|\mathcal{S}_m)}{\pi(x_k)p(x'|\mathcal{S}_m)}, \tag{3}$$

otherwise, with probability $1 - \alpha_{MH}$ set $x_{k+1} = x_k$.

3.5 If $k \leq K$, set $k = k + 1$ and repeat from step 3.2. Otherwise, stop.

Table 3. The rejection chain method

3.1 Set $k = 0$ and choose x_0.

3.2 Draw $x' \sim \bar{p}(x) \propto p(x|\mathcal{S}_m)$ and $u' \sim \mathcal{U}([0, 1])$.

3.3 If $u' \geq \frac{\pi(x')}{p(x'|\mathcal{S}_m)}$ repeat from step 3.2.

3.4 If $u' \leq \frac{\pi(x')}{p(x'|\mathcal{S}_m)}$, with probability

$$\alpha_{RC} = 1 \wedge \frac{\pi(x')\left[\pi(x_k) \wedge p(x_k|\mathcal{S}_m)\right]}{\pi(x_k)\left[\pi(x') \wedge p(x'|\mathcal{S}_m)\right]}, \tag{4}$$

set $x_{k+1} = x'$, otherwise, with probability $1 - \alpha_{RC}$ set $x_{k+1} = x_k$.

3.5 If $k \leq K$, set $k = k + 1$ and repeat from step 3.2. Otherwise, stop.

useful to obtain a good proposal density. The step 4 contains the MCMC iterations, repeated K times. In this work, we consider two possible techniques for the step 4 of FUSS. The first one is the *Metropolis-Hastings (MH) algorithm* [30], shown in Table 2. In this case, we denote the whole method as *FUSS-MH*. The second one is the *rejection chain algorithm* [33, 34]: it is shown in Table 3. In this case, firstly a rejection sampling (RS) test is performed, and a MH step is applied only when a sample is accepted, which ensures drawing from the target pdf. We denote the complete method as *FUSS-RC*.

FUSS-RC is slower than FUSS-MH since when a sample is rejected in the RS test the chain is not moved forward. On the other hand, FUSS-RC yields samples with less correlation due to application of the RS test. We will test and compare the performance in the numerical simulations. The notation $a \wedge b$ denotes the minimum between two real values, i.e., $\min[a, b]$.

3.1. Important remarks

It is important to emphasize some aspects of our approach. First of all, we remark that steps from 1 to 3 of the general FUSS algorithm in Table 1 are performed only once. The success of the FUSS algorithms lies on the speed in performing these steps

and the quality of the final proposal density. Indeed, since the shape of the proposal is tailored to $\pi(x)$, the generated samples will be virtually independent.

The initial support points in $\mathcal{S}_M$ play the role of internal "tuning" parameters of the FUSS algorithm. After the pruning step, the proposal $p(x|\mathcal{S}_m)$ is built according to the new set $\mathcal{S}_m$ with $m < M$. After that, the proposal p is kept fixed. Thus, the FUSS techniques are standard non-adaptive MCMC, avoiding any issue about the ergodicity.

The possibility of applying FUSS directly for drawing from multidimensional distributions depends on the ability to construct efficiently the proposal pdf via interpolation in dimensions higher than one (step 3 in the general structure of FUSS). In this work, we consider the application of FUSS within Gibbs sampling, drawing from univariate full-conditional pdfs. Furthemore, note also that all the operations in both algorithms, FUSS-MH and FUSS-RC, can be easily implemented in log-domain evaluating only the functions $V(x) = \log[\pi(x)]$ and $W(x) = \log[p(x|\mathcal{S}_m)]$.

If, in some case, we obtain $p(x|\mathcal{S}_m) \geq \pi(x)$ for all $x \in \mathcal{X}$, then the FUSS-RC algorithm becomes a standard rejection sampler, providing independent identically distributed (i.i.d.) samples from $\bar{\pi}(x)$. Indeed, note that in this scenario the probability α_{RC} of accepting the new state is always 1, i.e., after passing the rejection test, the movement is automatically accepted. The reason is that, in FUSS-RC after the rejection test, the proposed samples are distributed as

$$\bar{q}(x) \propto q(x) = \pi(x) \wedge p(x|\mathcal{S}_m).$$

Thus, $q(x)$ is used as proposal in the acceptance function α_{RC}. Finally, note also that $q(x)$ is closer, in terms of L_1 distance, to $\pi(x)$ than $p(x|\mathcal{S}_m)$: this is the reason why FUSS-RC produces samples with less correlation than FUSS-MH.[1]

3.2. Initialization and general strategy for FUSS

The initial set $\mathcal{S}_M$ should cover the regions of high probabilities described by the target $\pi(x)$. In general, if no prior information is available we suggest the following FUSS approach: choose a huge, dense, (uniform) initial grid of support points, $s_{i+1} - s_i = \epsilon$, i.e.,

$$\mathcal{S}_M = \{s_1, s_2 = s_1 + \epsilon, \ldots, s_M\},$$

in order to capture all the main features of the target. The grid could be thicker in the regions where the user desires to focus the computational effort. The number of support points is drastically reduced according to a certain criterion obtaining a new set $\mathcal{S}_m$ (see Section 5 for some examples) and we build a stepwise approximation of the target pdf given the pruned support points (see Section 4). The resulting proposal pdf is a self-tuned and tailored to the target but is non-adaptive, since it does not vary during the run of the chain (it is "adapted" offline). In the next section, we describe in details the construction of the proposal.

4. CONSTRUCTION OF THE PROPOSAL DENSITY

In several applications, it is useful to evaluate the target pdf in the log-domain so that, here, we consider the construction of the proposal function in the log-domain as well. Let us consider a set of support points after the pruning step (see Section 5),

$$\mathcal{S}_m = \{s_1, s_2, \ldots, s_m\} \subset \mathcal{X},$$

where $s_1 < \ldots < s_m$, and the intervals $\mathcal{I}_0 = (-\infty, s_1]$, $\mathcal{I}_j = (s_j, s_{j+1}]$ for $j = 1, ..., m-1$ and $\mathcal{I}_m = (s_m, +\infty)$. Then, let us consider

$$\bar{p}(x) \propto p(x|\mathcal{S}_m) = e^{W(x)},$$

where $W(x)$ is built as in Eq. (5) using a piecewise constant approximation, with the exception of the first and last intervals corresponding to the tails. Mathematically,

$$W(x) = w_i(x) = \max\left[V(s_i), V(s_{i+1})\right] \mathbb{I}_{\mathcal{I}_i}(x), \tag{5}$$

where $1 \leq i \leq m-1$ and

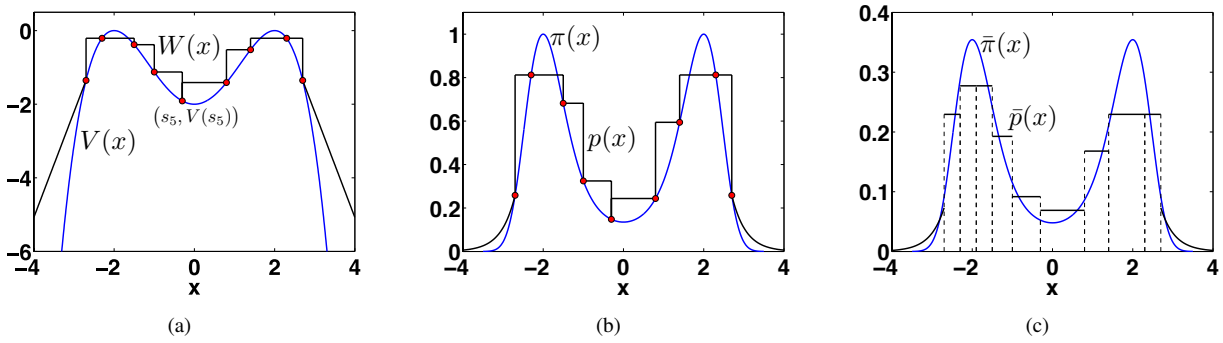
$$\mathbb{I}_{\mathcal{I}_i}(x) = \begin{cases} 1, & x \in \mathcal{I}_i = (s_i, s_{i+1}], \\ 0, & x \neq \mathcal{I}_i = (s_i, s_{i+1}]. \end{cases} \tag{6}$$

[1] In the extreme and trivial case when proposal and target are the same function, i.e. $q(x) = \pi(x)$, the correlation is zero since we are drawing directly i.i.d. samples from the target.

In the first and last interval, $\mathcal{I}_0$ and $\mathcal{I}_m$, we have

$$W(x) = w_j(x), \quad j = \{0, m\}, \quad x \in \mathcal{I}_j,$$

where $w_j(x)$ represents a generic log-tail function. For instance, choosing light tails, $w_j(x)$, $j = \{0, m\}$, are linear functions. Figures 1 and 6 provide some specific examples. For further details, see Appendix A. The choice of taking the maximum is in order to satisfy the inequality $W(x) \geq V(x)$, in as many regions as possible. If this inequality is verified for all $x \in \mathcal{X}$ then also $p(x|\mathcal{S}_m) \geq \pi(x)$, and FUSS-RC becomes a standard rejection sampler providing independent samples.



(a) (b) (c)

Fig. 1. Example of the proposal construction. **(a)** Construction procedure with $m = 9$ support points, in the log-domain. In this case, we have two light tails (two straight lines in the log-domain). **(b)** The corresponding unnormalized densities $p(x) = e^{W(x)}$ and $\pi(x) = e^{V(x)}$. **(c)** The corresponding normalized densities $\bar{p}(x) \propto p(x)$ and $\bar{\pi}(x) \propto \pi(x)$.

4.1. Variate generation from $\bar{p}(x)$

The proposal $\bar{p}(x) \propto p(x|\mathcal{S}_m)$ is composed in general by $m + 1$ pieces, including the two tails. Note that $\bar{p}(x)$ can be seen as a finite mixture

$$\bar{p}(x) \propto p(x|\mathcal{S}_m) = \sum_{i=0}^{m} \eta_i \phi_i(x),$$

where $\sum_{i=0}^{m} \eta_i = 1$ and $\phi_i(x) \propto \exp(w_i(x))$, for all $x \in \mathcal{I}_i$ and $\phi_i(x) = 0$ for $x \notin \mathcal{I}_i$. Hence, in order to draw adequately from $\bar{p}(x)$, it is necessary:

1. Compute the area A_i below each piece, $i = 0, \ldots, m$. It can be done analytically easily for the rectangular pieces, and for the tails considered here (exponential or Pareto, for instance) as well. Then, normalize them

$$\eta_i = \frac{A_i}{\sum_{j=1}^{m} A_j}, \quad \text{for} \quad i = 0, \ldots, m.$$

2. Choose a piece, i.e., an index $j^* \in \{0, \ldots, m\}$ according to the weights η_i, $i = 0, \ldots, m$.

3. Given the index j^*, draw $x' \sim \phi_{j^*}(x) \propto \exp(w_{j^*}(x))$.

It is important to remark that the process of calculate the areas A_i and then the weights η_i is done once, before running the Markov chain. The calculation of the areas A_i of the rectangular regions is straightforward and fast.

5. PRUNING ALGORITHMS

The computational cost of drawing from the proposal pdf $\bar{p}(x)$ (and, as a consequence, the speed of the algorithm) depends on the number of used support points and on the complexity of generating each piece. Since we are considering simple uniform pieces, the speed is mainly obstructed by the total number of points. The application of the pruning step has two advantages: it speeds up the algorithm, and allows the use of a greater number of initial support points, capturing all the important features of the target. In the following, we present some possible pruning criteria, sorted for increasing level of complexity. For the sake of

Table 4. Pruning algorithm P1

1. Given $\mathcal{S}_M = \{s_1, \ldots, s_M\}$, decide the desired number m of final support points (or a rate of reduction $\frac{m}{M}$).
2. Sort $\pi(s_i)$, $i = 1, \ldots, M$, in a decreasing order
$$\pi(s_{r_1}) \geq \pi(s_{r_2}) \geq .... \geq \pi(s_{r_M}).$$
3. Return $\mathcal{S}_m = \{s_{r_1}, s_{r_2}, \ldots, s_{r_m}\}$.

Table 5. Pruning algorithm P2

1. Given $\mathcal{S}_M = \{s_1, \ldots, s_M\}$, choose a value $\delta \in (0, 1)$.
2. Find all the support points $s_{k_j} \in \mathcal{S}_M$ such that
$$\pi(s_{k_j}) \leq \delta \max_{1 \leq i \leq M} \pi(s_i). \tag{7}$$
3. Return $S_m = S_M \setminus \{s_{k_1}, \ldots, s_{k_G}\}$, where G is the number of points satisfying the inequality (7).

Table 6. Pruning algorithm P3

1. Given $\mathcal{S}_M = \{s_1, \ldots, s_M\}$. Set $\mathcal{S}^{(0)} = \mathcal{S}_M$, $r = 1$ and $L = \max_{1 \leq i \leq M} |\pi(s_{i+1}) - \pi(s_i)|$. Choose a value $\delta > 0$.
2. While $G \neq 0$:
   (a) Find all the support points $s_{k_j} \in \mathcal{S}^{(n)}$ such that
   $$\left|\pi(s_{k_j+1}) - \pi(s_{k_j})\right| \leq \delta L, \tag{8}$$
   (b) $\mathcal{S}^{(r)} = \mathcal{S}^{(r-1)} \setminus \{s_{k_1}, \ldots, s_{k_G}\}$ where $G = |\{s_{k_1}, \ldots, s_{k_G}\}|$.
   (c) Set $r = r + 1$.
3. Return $\mathcal{S}_m = \mathcal{S}^{(n)} \setminus \{s_1\}$.

simplicity, we assume a bounded target $\pi(x)$. The first two procedures P1 and P2 are shown in Tables 5 and 5. They are based on the simple idea of pruning all the points s_r with "small" value $\pi(s_r)$. They are the simplest and fastest approaches but they also present several limitations, so that P1 and P2 should be used carefully. For instance, they are not advisable for heavy tailed distributions. Another drawback of the procedures P1 and P2 is that the performance is quite sensitive to the dispersion of the target.

More refined pruning techniques can be easily provided. An example is the procedure P3 in Table 6. The underlying idea is that we can remove support points where the target is virtually flat, i.e., $|\pi(s_{i+1}) - \pi(s_i)| \approx 0$. Moreover, note that at the first iteration, if a uniform grid is used, i.e., $s_{i+1} - s_i = \epsilon$, the ratio $\frac{\pi(s_{i+1}) - \pi(s_i)}{\epsilon}$ is an estimation of the first derivative, hence a condition over $|\pi(s_{i+1}) - \pi(s_i)|$ can be consider a condition over the first derivative of π. Clearly, this procedure could be repeated until achieving the desired rate of reduction or simply iterated N times. In Table 6, the procedure is iterated until the pruning condition is no longer verified. Below, we describe the theory for providing an optimal pruning procedure.

5.1. Optimal pruning strategy

The performance of a rejection sampler or an independent Metropolis algorithm is related to the L_1 distance between the target and the proposal [30],

$$D_{p|\pi}(\mathbb{R}) = \int_{-\infty}^{\infty} |p(x|\mathcal{S}_M) - \pi(x)|dx. \tag{9}$$

With the proposal procedure considered here, we can write

$$D_{p|\pi}(\mathbb{R}) = \sum_{j=0}^{M} D_{p|\pi}(\mathcal{I}_j),$$

where $D_{p|\pi}(\mathcal{I}_j)$ denotes the local L_1 distance within the i-th interval ($0 \leq j \leq M$), i.e.

$$\begin{aligned} D_{p|\pi}(\mathcal{I}_j) &= \int_{\mathcal{I}_j} |p(x|\mathcal{S}_M) - \pi(x)|dx \\ &= \int_{s_j}^{s_{j+1}} |p(x|\mathcal{S}_M) - \pi(x)|dx. \end{aligned} \tag{10}$$

Here, for the sake of simplicity, we are considering the (non-rigorous) definition $s_0 = -\infty$ and $s_{M+1} = +\infty$, since $\mathcal{I}_0 = (-\infty, s_1]$ and $\mathcal{I}_M = [s_M, \infty)$. Moreover, recall that $s_1 < s_2 < \ldots < s_M$.

The essential consideration is the following: when a support point is removed, the distance between the target and the proposal generally will tend to increase, thus leading to a worse performance of the algorithm. Hence, an optimal criterion for pruning support points is discarding those that lead to as small increase as possible in the L_1 distance between $p(x)$ and $\pi(x)$. Since the proposal $p(x|\mathcal{S}_M)$ is a piecewise constant function with values

$$\exp(w_i) = \exp(\max\left[V(s_i), V(s_{i+1})\right]),$$

with the exception of the tails, and considering a *continuous* target pdf, it is apparent that

$$D_{p|\pi}(\mathcal{I}_j) \leq B_{j,j+1} = (s_{j+1} - s_j)|\pi(s_{j+1}) - \pi(s_j)|.$$

i.e., $B_{j,j+1}$ is an upper bound for the distance $D_{p|\pi}(\mathcal{I}_j)$. If the number m of used support points grows, then clearly $B_{j,j+1} \to D_{p|\pi}(\mathcal{I}_j)$. Thus, the value $B_{j,j+1}$ can be considered as a rough approximation of $D_{p|\pi}(\mathcal{I}_j)$. This observation is the theoretical basis of the pruning strategy detailed in Table 7, where a generic interval $[s_j, s_{j+2}]$ is considered. Depending on the approximation $B_{j,j+2}$ we decide if pruning s_{j+1} or not. Clearly, at each iteration at most $R = \left\lfloor \frac{M-1}{2} \right\rfloor$ points can be removed. The procedure is iterated until no more points are pruned, i.e., when all the upper bounds (b_r, in the notation of Table 7) are greater than the threshold δ.[2] Clearly, the algorithm could also be stopped earlier.

6. NUMERICAL EXAMPLES

In this section, we provide numerical results for different types of target distributions. First of all in Section 6.1, we test the performance of the FUSS algorithm comparing with other well-known MCMC methods, drawing from univariate (unimodal and multimodal) densities. Then, we consider the application of the FUSS technique within the Gibbs sampling for drawing from different types of multidimensional pdfs (Sections 6.2, 6.3 and 6.4).

6.1. Univariate densities

6.1.1. Unimodal target pdf: Nakami distribution

First of all we consider a Nakagami target distribution, i.e.,

$$\bar{\pi}(x) \propto \pi(x) = x^{2\beta-1} \exp\left(-\frac{\beta}{\Omega}x^2\right), \quad x > 0, \tag{11}$$

[2]Note that also in P3 and P4, as in P2, there is a maximum allowed value δ^*. For $\delta > \delta^*$, all support points are removed.

Table 7. Pruning algorithm P4

1. Choose a value $\delta > 0$. Given $\mathcal{S}_M = \{s_1, ..., s_M\}$, set $\mathcal{S}^{(0)} = \mathcal{S}_M$, $m = M$, $n = 0$ and
$$L = \max_{1 \le j \le \left\lfloor \frac{m-1}{2} \right\rfloor} (s_{2j+1} - s_{2j-1}) |\pi(s_{2j+1}) - \pi(s_{2j-1})|.$$
2. For $r = 1, \ldots, R = \left\lfloor \frac{m-1}{2} \right\rfloor$:
   (a) Compute $b_r = (s_{2r+1} - s_{2r-1}) |\pi(s_{2r+1}) - \pi(s_{2r-1})|$.
   (b) If $b_r \le \delta L$, set $\mathcal{S}^{(r)} = \mathcal{S}^{(r-1)} \setminus \{s_{2r}\}$ and $n = n + 1$.
   (c) Otherwise, if $b_r > \delta L$, set $\mathcal{S}^{(r)} = \mathcal{S}^{(r-1)}$.
3. If $n > 0$ set $n = 0$, $\mathcal{S}^{(0)} = \mathcal{S}^{(R)}$, $m = |\mathcal{S}^{(R)}|$ and repeat from step 2.
4. Otherwise, if $n = 0$, return $\mathcal{S}_m = \mathcal{S}^{(R)}$.

with $\beta \ge 0.5$ and $\Omega > 0$. The Nakagami distribution is widely used for the simulation of fading channels in wireless communications [1, 19, 29]. When β is an integer or half-integer (i.e., $\beta = \frac{n}{2}$ with $n \in \mathbb{N}$), independent samples can be directly generated through the square root of a sum of squares of n zero-mean i.i.d. Gaussian random variables. However, for generic values of β there is not direct method to sample from it. Here, our goal is to estimate the expected value of $X \sim \bar{\pi}(x)$, $\mu = E[X] = \frac{\Gamma(\beta+\frac{1}{2})}{\Gamma(\beta)} \sqrt{\frac{\Omega}{\beta}}$, and the variance, $\sigma^2 = \Omega \left(1 - \frac{1}{\beta} \left(\frac{\Gamma(\beta+\frac{1}{2})}{\Gamma(\beta)}\right)^2\right)$, with $\Omega = 1$ and $\beta = 4.6$.

We apply the FUSS methods using different pruning procedures P2, P3, P4 with different values of the threshold parameter δ. We use an initial set $\mathcal{S}_M = \{0.01, 0.02, 0.03, \ldots, 10^3\}$ with $M = 10^5$ points. We also test a standard MH technique [30] with a random walk proposal
$$\bar{p}(x_k|x_{k-1}) \propto \exp\left\{\frac{-(x_k - x_{k-1})^2}{2\sigma_p^2}\right\},$$
with different values of σ_p. Furthermore, we consider another well-known methodology, the *slice sampling* techniques [30, Chapter 8]. For as fair as possible comparison of the required computational time, we use for both the corresponding Matlab functions directly provided by MathWorks (`mhsample.m` and `slicesample.m`).

For all these techniques, we choose $x_0 \in \mathcal{U}[0, 10]$, $K = 5000$, and we consider all the generated samples without removing any burn-in period. We have performed $3 \cdot 10^4$ independent runs and the results are shown in Tables 8-9. These tables provide the *Mean Square Errors* (MSE) in the estimation of μ and σ^2, the linear correlation among the samples at lag-1, $\rho(1)$, the acceptance rate ($0 \le$AR≤ 1) in the rejection sampling (RS) step, the number of points m after the pruning and the computational time. The time values are normalized w.r.t. the time required by the MH method. Since only FUSS-RC has an RS step, in the other algorithms AR is set to 1 since no proposed samples are discarded in a RS test.[3] This means that the total number of iterations of FUSS-RC is greater than $K = 5000$ (depending on the acceptance rate), whereas for the other methods is exactly $K = 5000$.

We can see that FUSS algorithms always outperform the standard MH and slice techniques. In all cases, the FUSS algorithms are faster, with the exception of FUSS-RC with $\delta = 0.9$ which is slightly slower than MH. Moreover, both FUSS-MH and FUSS-RC are able to reach virtually the performance of an *exact sampler* in the estimation of μ drawing independent samples, that is
$$\text{MSE}(\mu) \ge \frac{\sigma^2}{K} = 1.0560\ 10^{-5}.$$
Clearly, FUSS-MH is always faster than FUSS-RC since the lack of rejection sampling test. On the other hand, for the same reason FUSS-RC provides better results. Note that, in spite of the greater number of iterations owing to the rejected samples, FUSS-RC is faster than the standard MH and the slice sampling. The time spent in FUSS-MH always increases with m, whereas in FUSS-RC the computational cost can also decrease when m grows due to an improvement in the acceptance rate.

The pruning procedures P3 and P4 perform clearly better than P2. The best one is P4 as expected.[4] This is owing to the fact that P4 selects the final support points in a more desirable manner, which can be seen comparing the time values in FUSS-RC.

[3] We remark that we are not referring to the acceptance probability α in the FUSS-MH and MH methods. The AR value is the averaged number of samples accepted in the rejection sampling step of FUSS-RC.

[4] The benefit of using P4 can be seen more clearly in the next example, where a multimodal target pdf and a smaller number of iterations K are considered.

		FUSS-MH				FUSS-RC			
Pruning		$\boldsymbol{\delta}$**= 0.9**	$\boldsymbol{\delta}$**= 0.5**	$\boldsymbol{\delta}$**= 0.3**	$\boldsymbol{\delta}$**= 0.01**	$\boldsymbol{\delta}$**= 0.9**	$\boldsymbol{\delta}$**= 0.5**	$\boldsymbol{\delta}$**= 0.3**	$\boldsymbol{\delta}$**= 0.01**
P2	MSE(μ)	3.13 10^{-5}	2.15 10^{-5}	1.68 10^{-5}	1.10 10^{-5}	1.12 10^{-5}	1.09 10^{-5}	1.06 10^{-5}	1.05 10^{-5}
	MSE(σ^2)	3.94 10^{-6}	2.25 10^{-6}	1.65 10^{-6}	1.05 10^{-6}	1.21 10^{-6}	1.13 10^{-6}	1.12 10^{-6}	1.12 10^{-6}
	$\rho(1)$	0.4811	0.3288	0.2142	0.0087	-8.77 10^{-4}	-2.26 10^{-4}	-1.30 10^{-4}	-1.05 10^{-4}
	AR	1	1	1	1	0.3926	0.6354	0.7677	0.9779
	m	23	56	73	139	23	56	73	139
	Time	0.6683	0.6742	0.6781	0.6875	1.3476	0.8263	0.7927	0.7328
P3	MSE(μ)	1.11 10^{-5}	1.09 10^{-5}	1.06 10^{-5}	1.05 10^{-5}	1.10 10^{-5}	1.07 10^{-5}	1.05 10^{-5}	1.05 10^{-5}
	MSE(σ^2)	1.20 10^{-6}	1.16 10^{-6}	1.15 10^{-6}	1.13 10^{-6}	1.14 10^{-6}	1.11 10^{-6}	1.11 10^{-5}	1.10 10^{-6}
	$\rho(1)$	0.0200	0.0124	0.0077	0.0053	-2.35 10^{-4}	-2.50 10^{-4}	-2.80 10^{-4}	-1.85 10^{-4}
	AR	1	1	1	1	0.9570	0.9630	0.9787	0.9830
	m	50	88	106	166	50	88	106	166
	Time	0.6712	0.6816	0.6859	0.7019	0.7676	0.7408	0.7424	0.7426
P4	MSE(μ)	1.10 10^{-5}	1.09 10^{-5}	1.06 10^{-5}	1.05 10^{-5}	1.10 10^{-5}	1.07 10^{-5}	1.05 10^{-5}	1.05 10^{-5}
	MSE(σ^2)	1.19 10^{-6}	1.14 10^{-6}	1.12 10^{-6}	1.10 10^{-6}	1.13 10^{-6}	1.10 10^{-6}	1.09 10^{-6}	1.08 10^{-6}
	$\rho(1)$	0.0133	0.0096	0.0076	0.0053	1.27 10^{-4}	-2.61 10^{-4}	-2.45 10^{-4}	-2.62 10^{-4}
	AR	1	1	1	1	0.9666	0.9769	0.9800	0.9832
	m	71	109	121	177	71	109	121	177
	Time	0.6849	0.6937	0.6957	0.7105	0.7339	0.7397	0.7380	0.7502

Table 8. Results of FUSS methods with different pruning procedures, $K = 5000$ and the Nakagami target with $\beta = 4.6$ and $\Omega = 1$.

		$\boldsymbol{\sigma_p}$**= 0.2**	$\boldsymbol{\sigma_p}$**= 0.5**	$\boldsymbol{\sigma_p}$**= 0.8**	$\boldsymbol{\sigma_p}$**= 1**	$\boldsymbol{\sigma_p}$**= 2**	$\boldsymbol{\sigma_p}$**= 3**	$\boldsymbol{\sigma_p}$**= 4**
MH	MSE(μ)	0.0021	3.95 10^{-4}	1.98 10^{-4}	1.52 10^{-4}	1.43 10^{-4}	1.90 10^{-4}	2.52 10^{-4}
	MSE(σ^2)	0.0513	0.0091	0.0039	0.0027	9.20 10^{-4}	6.18 10^{-4}	5.69 10^{-4}
	$\rho(1)$	0.8935	0.7495	0.7433	0.7611	0.8389	0.8808	0.9043
	AR	1	1	1	1	1	1	1
	Time	1	1	1	1	1	1	1
Slice sampling								
MSE(μ)= 1.24 10^{-5}		MSE(σ^2)= 2.27 10^{-5}		$\rho(1)$= 0.0229		AR=1	Time= 2.5037	

Table 9. Results of the standard MH and slice sampling methods, $K = 5000$ and the Nakagami target with $\beta = 4.6$ and $\Omega = 1$.

Indeed, FUSS-RC-P4 is always faster than FUSS-RC-P3, even if more points are used. The reason is that the points are better located so that the L_1 distance between target and proposal is smaller and the acceptance rate greater. It can be noted observing also that FUSS-RC-P4 with $\delta = 0.9$ uses only $m = 71$ points obtaining AR= 0.9666, whereas FUSS-RC-P3 with $\delta = 0.5$ uses $m = 88$ achieving AR= 0.9630. Namely, FUSS-RC-P3 even with more points can have an AR smaller than FUSS-RC-P4. Thus, FUSS-RC-P4 is faster than FUSS-RC-P3.

6.1.2. Multimodal target pdf: mixture of Gaussians

Now we test the FUSS approach drawing from a multimodal target density. More specifically, we consider a mixture of 4 Gaussian pdfs,

$$\bar{\pi}(x) = \sum_{i=1}^{4} \mathcal{N}(x; \mu_i, \sigma_i^2),$$

where $\mu_1 = -7, \sigma_1 = 0.1, \mu_2 = 0, \sigma_2 = 1, \mu_3 = 8, \sigma_3 = 0.2$, and $\mu_4 = 15, \sigma_4 = 0.1$. This choice leads to 4 modes as shown in Figure 2(a). Three of them are notably tight, so that this target becomes even more challenging. We consider first only $K = 200$ as total number of iterations of the chain, taking into account all the generated samples to make the estimation (without removing any burn-in period). We again test the performance of FUSS algorithm, the standard MH method with a Gaussian random walk proposal pdf and the slice sampling, as in the previous example. For the sake of simplicity, we have only considered the FUSS-MH, in this example.[5] For all the different algorithms, the initial state x_0 of the chain is chosen uniformly in $[-10, 20]$, i.e., $x_0 \in \mathcal{U}([-10, 20])$. As in the previous example, we have used the Matlab functions directly provided by MathWorks (`mhsample.m` and `slicesample.m`). For FUSS-MH, we use an initial set $\mathcal{S}_M = \{-10^3, -10^3+0.01, \ldots, 10^3-0.01, 10^3\}$ then $M = 2 \cdot 10^5 + 1$ points.

[5] As shown in the previous numerical example, FUSS-RC provides in general better results wasting slightly more computational time.

All the results are averaged over $3 \cdot 10^4$ runs. The MSEs in the estimation of the mean (μ) and variance (σ^2) of the target are shown in Table 10 for FUSS-MH, whereas the results of the standard MH are provided in Table 11, using $K = 200$ samples for both algorithms. We vary the values of the internal parameters as δ in FUSS-MH and σ_p in MH. We also provide the linear correlation at lag-1 (denoted as $\rho(1)$), the number of points after pruning (m) and the spent time normalized w.r.t. the time spent by MH using $K = 200$. Note that, in this case, the theoretical bound for the MSE(μ) $\geq \frac{\sigma^2}{K}$ is $\frac{\sigma^2}{K} = \frac{68.765}{200} = 0.3438$, achievable via Monte Carlo using independent samples. Furthermore, in Table 12 we give the MSE obtained using MH and slice sampling, increasing the total number K of iterations of the Markov chain. Figures 2(b)-(c) depict respectively the log-MSE and log-time of MH and slice sampling as function of K. Observing the results, it is apparent that FUSS methodology speeds up the convergence of the Markov chain. Indeed:

- FUSS-MH always outperforms MH and slice sampling, providing an improvement at least of 41% in the MSE, and $\approx 98\%$ with the P4 pruning procedure.

- Table 11 also shows the importance of using an adaptive or self-tuned method: the results vary considerably with the choice of σ_p and, even in the best case, the MSE is much higher than in the worst case of FUSS-MH. This is owing to FUSS-MH is able to adapt the entire shape of the proposal pdf according to the target.

- FUSS-MH using the pruning P4 virtually reaches the theoretical bound for the MSE, i.e., 0.3438, obtainable using independent samples. This remarkable result is also achieved with the different choices of the threshold δ.

- Observing Table 12 and Figures 2(b)-(c), we can see that in order to obtain the same performance of FUSS-MH using a standard MH method, we need to use between $K = 15000$ and $K = 20000$ iterations, spending at least $+50\%$ more of computational time, comparing with FUSS-MH-P4 with $\delta = 0.9$. With the slice sampling we need to use between $K = 15000$ and $K = 20000$ iterations as well, using at least $+125\%$ more of computational time.

		FUSS-MH			
Pruning		$\boldsymbol{\delta = 0.9}$	$\boldsymbol{\delta = 0.5}$	$\boldsymbol{\delta = 0.3}$	$\boldsymbol{\delta = 0.01}$
P2	MSE(μ)	11.38	7.33	4.30	0.4680
	MSE(σ^2)	409.02	288.35	211.78	19.52
	$\rho(1)$	0.9142	0.8917	0.8419	0.1468
	m	18	46	103	662
	Time	1.03	1.04	1.06	1.17
P3	MSE(μ)	1.7683	1.1368	0.8686	0.3679
	MSE(σ^2)	78.96	49.82	35.92	15.3513
	$\rho(1)$	0.6693	0.5284	0.4224	0.0296
	m	61	99	135	497
	Time	1.20	1.27	1.31	1.35
P4	MSE(μ)	0.3786	0.3662	0.3638	0.3526
	MSE(σ^2)	15.53	15.31	15.10	14.53
	$\rho(1)$	0.0446	0.0306	0.0247	0.0093
	m	145	195	223	605
	Time	1.23	1.26	1.28	1.40

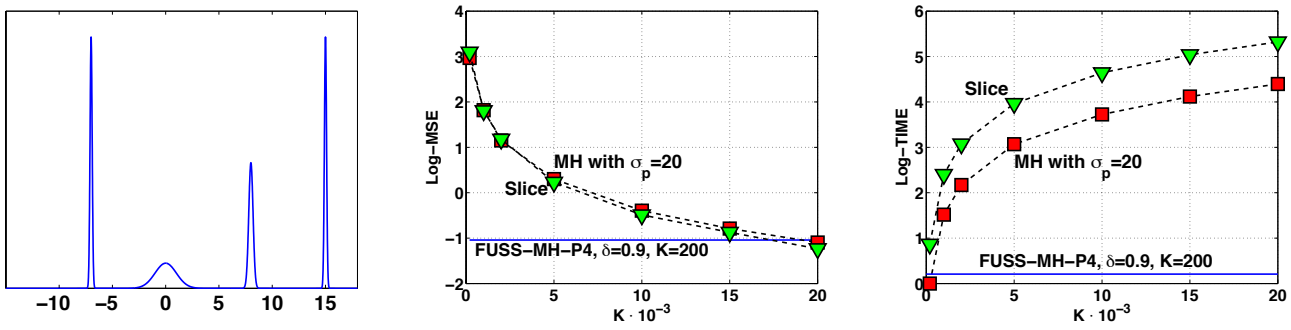
Table 10. Results of FUSS-MH with different pruning procedures and $K = 200$, drawing from the mixture of Gaussians target.

		$\boldsymbol{\sigma_p = 2}$	$\boldsymbol{\sigma_p = 8}$	$\boldsymbol{\sigma_p = 14}$	$\boldsymbol{\sigma_p = 20}$	$\boldsymbol{\sigma_p = 25}$	$\boldsymbol{\sigma_p = 30}$
MH	MSE(μ)	34.92	24.64	19.62	19.46	19.80	20.57
	MSE(σ^2)	$4.43\ 10^3$	$1.36\ 10^3$	$1.13\ 10^3$	$1.09\ 10^3$	$1.37\ 10^3$	$1.51\ 10^3$
	$\rho(1)$	0.7481	0.9465	0.9425	0.9424	0.9445	0.9459
	Time	1	1	1	1	1	1

Table 11. Results of the standard MH method with $K = 200$, drawing from the mixture of Gaussians target. The best results are obtained with $\sigma_p = 20$, but the MSE is always greater than using FUSS-MH.

		$\mathbf{K = 200}$	$\mathbf{K = 10^3}$	$\mathbf{K = 2 \cdot 10^3}$	$\mathbf{K = 5 \cdot 10^3}$	$\mathbf{K = 10^4}$	$\mathbf{K = 1.5 \cdot 10^4}$	$\mathbf{K = 2 \cdot 10^4}$
MH $\boldsymbol{\sigma_p}$**= 20**	MSE(μ)	19.46	6.16	3.16	1.35	0.6735	0.4540	0.3341
	MSE(σ^2)	1.09 10^3	263.36	123.37	44.69	22.48	14.45	10.79
	Time	1	4.55	8.75	21.51	41.48	61.44	81.06
Slice	MSE(μ)	22.13	6.08	3.25	1.26	0.6136	0.4177	0.2911
	MSE(σ^2)	1.10 10^3	171.13	68.91	23.98	10.48	7.40	5.85
	Time	2.37	11.04	21.62	52.66	103.78	154.38	204.57

Table 12. Results of the slice sampling and the standard MH method with $\sigma_p = 20$, varying the total number K of iterations of the chain, drawing from the mixture of Gaussians target.



(a) (b) (c)

Fig. 2. **(a)** The multimodal target pdf $\pi(x)$. **(b)** Log-MSE as function of $\frac{K}{10^3}$ for the slice (triangles) and the MH (squares) methods, with $\sigma_p = 20$. The solid line shows the log-MSE, $\log(0.3526) = -1.0424$, of FUSS-MH-P4 with $\delta = 0.01$, achieved using only $K = 200$. **(c)** The spent log-time as function of K for the slice (triangles) and the MH (squares) methods. The solid line corresponds to FUSS-MH-P4 with $\delta = 0.01$ and $K = 200$.

6.2. FUSS within Gibbs: a toy example

In order to show how important it is to use an adequate MCMC technique *within* a Gibbs sampler, in this section we provide a simple example using different methods for drawing from the full-conditional pdfs: the results obtained by the Gibbs sampler can vary notably, depending on the technique used "within Gibbs". Let us consider the bivariate target density $\bar{\pi}(x_1, x_2) \propto \pi(x_1, x_2)$ where

$$\pi(x_1, x_2) = \exp\left(-\frac{(x^2 - A + By)^2}{4} - \frac{x^2}{2\sigma_1^2} - \frac{y^2}{2\sigma_2^2}\right),$$

with $A = 16$, $B = 10^{-2}$, and $\sigma_1^2 = \sigma_2^2 = \frac{10^4}{2}$. Densities with this analytic form have often been used in the literature [22, 10] to evaluate the performance of different Monte Carlo algorithms. We apply a Gibbs sampler to draw from $\bar{\pi}(x_1, x_2)$. To generate from the full conditional pdfs, we use FUSS-MH-P4 with $\delta = 0.9$, starting with a uniform grid $\mathcal{S}_M = \{-10^4, -10^4 + 0.01, \ldots, 10^4 - 0.01, 10^4\}$, i.e., $M = 2 \cdot 10^5 + 1$. In order to compare the performance, we also apply the standard ARMS method [7]: this technique uses an interpolation procedure to build the proposal using a set of support points, similarly to FUSS. However, unlike FUSS, the ARMS method starts with few support points and adds new ones adaptively. We choose for ARMS the initial set $\{-10, -6, -4.3, -0.01, 3.2, 3.8, 4.3, 7, 10\}$. Finally, we also consider a standard MH algorithm with a random walk proposal

$$\bar{p}(x_t | x_{t-1}) \propto \exp\left(-\frac{(x_t - x_{t-1})^2}{2\sigma_p^2}\right),$$

with $\sigma_p \in \{1, 2, 10\}$ (initial state $x_0 \in \mathcal{U}([-5, 5])$). We use $N_G = 2000$ iterations of the Gibbs sampler, using all the samples to estimate four statistics which involve the first four moments of the target: mean, variance, skewness and kurtosis. In each iteration, we draw K samples from each full-conditionals, take the last one and continue in the Gibbs cycle.

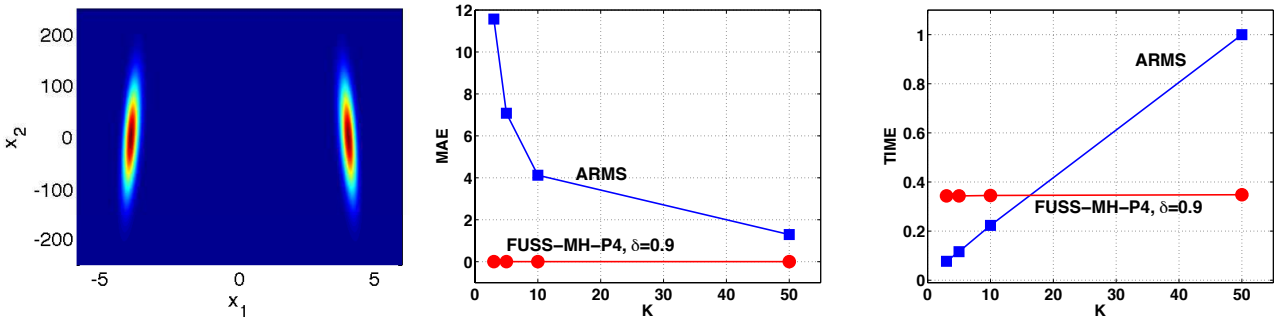
Table 6.2 provides the numerical results (averaged over 1000 runs) for the *Mean Absolute Error* (MAE) and the time required by the Gibbs sampler. The time is normalized by considering 1 to be the time elapsed by using ARMS within Gibbs

and $K = 50$. Figure 3(a) illustrates the target $\bar{\pi}(\mathbf{x})$ whereas Figures 3(b)-(c) depict the MAE and the spent time of ARMS and FUSS-MH-P4 as function of K.

Technique		K	MAE Mean	MAE Variance	MAE Skewness	MAE Kurtosis	Time
FUSS-MH-P4 $\delta = 0.9$		3	0.0735	0.0365	0.0369	0.0022	0.343
		5	0.0735	0.0361	0.0367	0.0022	0.343
		10	0.0724	0.0354	0.0365	0.0021	0.345
		50	0.0721	0.0355	0.0364	0.0021	0.348
ARMS		3	3.408	11.580	3.384	11.572	0.077
		5	3.151	9.839	2.650	7.079	0.116
		10	2.798	7.665	2.024	4.124	0.223
		50	1.918	3.407	1.134	1.292	1.000
MH	$\sigma_p = 1$	100	3.509	12.308	3.671	13.666	0.540
	$\sigma_p = 2$	100	1.756	3.077	0.9782	0.9633	0.540
	$\sigma_p = 10$	100	0.0756	0.0376	0.0368	0.0025	0.540
	$\sigma_p = 1$	1000	3.508	12.302	3.665	13.624	3.229
	$\sigma_p = 2$	1000	1.601	2.560	0.8741	0.7691	3.229
	$\sigma_p = 10$	1000	0.0743	0.0360	0.0363	0.0021	3.229

Table 13. MAE in the estimation of four statistics (first component) of $\bar{\pi}(x_1, x_2)$ and normalized simulation time. All the techniques are used within a Gibbs sampler ($N_G = 2000$ total iterations), performing K iterations for each full-conditionals.

Observing Table 6.2 and Fig. 3(b), we notice that FUSS-MH-P4 outperforms ARMS for all values of K in the estimations of the four central moments: FUSS-MH-P4 achieves always MAEs close to zero. As shown also in Fig. 3(c), due to the pruning procedure, FUSS-MH-P4 is a slightly slower than ARMS for $K = 3, 5, 10$. However, the computational time FUSS-MH-P4 remains virtually constant with K whereas in ARMS it increases with K, since ARMS adds new support points for improving the proposal pdf, hence becoming more costly. Regarding the use of the MH algorithm within Gibbs, the results depend largely on the choice of the variance of the proposal, σ_p^2, showing the needed of adaptive or self-tuned MCMC strategies. Indeed, for an inadequate scale parameter (e.g., $\sigma_p = 1$ or $\sigma_p = 2$), even the use of $K = 1000$ provides bad results. On the other hand, when a good σ_p is selected (i.e., $\sigma_p = 10$), MH with $K = 100$ and $K = 1000$ provides virtually the same performance of FUSS-MH-P4 but obviously with increased computational cost.



(a) (b) (c)

Fig. 3. **(a)** Contour plot of the target pdf $\bar{\pi}(x_1, x_2)$. **(b)** MAE in estimation of the kurtosis (first component) as function of K, using ARMS-within-Gibbs (squares) and FUSS-within-Gibbs (circles). **(c)** Normalized spent time as function of K for ARMS (squares) and FUSS (circles) within the Gibbs sampler.

6.3. Parameter estimation in a chaotic system

The FUSS algorithms are also able to overcome computational issues in certain statistical frameworks where other, even sophisticated, MCMC techniques seem to fail [27, 28]. In order to show this capability, we consider the example of estimating fixed parameters of a chaotic system, which is consider a challenging problem in literature [11, 27, 28]. This systems are often utilized for modelling the evolution of population sizes, in ecology for instance [27]. We consider a logistic map [2] perturbed

by noise,

$$z_{t+1} = R\left[\, z_t \left(1 - \frac{z_t}{\Omega}\right)\right] \exp(\epsilon_t), \quad \epsilon_t \sim \mathcal{N}(0.\lambda^2), \tag{12}$$

$R > 0$, $\Omega > 0$, $z_1 \in (0, 1)$, where R and Ω are unknown, whereas a sequence $z_{1:T} = [z_1, \ldots, z_T]$ is observed. For the sake of simplicity, we consider λ known. Therefore, the likelihood function is given by

$$p(z_{1:T}|R, \Omega) = \prod_{t=1}^{T-1} p(z_{t+1}|z_t, R, \Omega),$$

where, defining $g(z_t, R, \Omega) = R\left[\, z_t \left(1 - \frac{z_t}{\Omega}\right)\right]$, we have

$$p(z_{t+1}|z_t, K, \Omega) \propto \left|\frac{g(z_t, R, \Omega)}{z_{t+1}}\right| \exp\left(-\frac{\log\left(\frac{z_{t+1}}{g(z_t, R, \Omega)}\right)^2}{2\lambda^2}\right),$$

if $g(z_t, R, \Omega) > 0$, otherwise $p(z_{t+1}|z_t, K, \Omega) = 0$, if $g(z_t, R, \Omega) \leq 0$. Considering uniform priors over R, $p(r) \sim \mathcal{U}([0, 10^4])$, and Ω, $p(\omega) \sim \mathcal{U}([0, 10^4])$), we desire to compute efficiently the mean of the bivariate posterior pdf $p(K, \Omega|z_{1:T}) \propto p(z_{1:T}|R, \Omega)$, as estimation of the parameters.

In the experiments, we set $R = 3.7$, $\Omega = 0.4$ and $T = 20$. Furthermore, we take into account different values of λ of the same order of magnitude as considered in [27]. We apply FUSS-MH-P2 with $\delta = 10^{-3}$, $K = 10$, within a Gibbs sampler with only $N_G = 50$ iterations ($S_M = \{10^{-4}, 2 \cdot 10^{-4}, \ldots, 20\}$).[6] We also consider a MH with random walk proposal, $\bar{p}(x_k|x_{k-1}) \propto \exp\left(\frac{-(x_k - x_{k-1})^2}{2\sigma_p^2}\right)$, with two different values of $\sigma_p \in \{1, 2\}$. The initial states of the chains are chosen randomly $\mathcal{U}([1, 5])$ and $\mathcal{U}([0.38, 1.5])$ for sampling from the full conditionals $p(R|\Omega, z_{1:T})$ and $p(\Omega|R, z_{1:T})$, respectively. In order to test the performance, we also compute approximately the true value of the mean (of the corresponding posterior pdf) via an expensive deterministic numerical integration, and so we are able to calculate the MSE obtained by FUSS-MH-P2 and MH. The results, averaged 1000 over independent runs, are shown in Table 14. We can see clearly that, FUSS-MH-P2 achieves a negligible MSE in the estimation of the expected value of the different posterior distributions, corresponding to different values of λ. Comparing with the MSE of the MH, it is apparent the benefit of building a proposal tailored to the full-conditionals as in FUSS. Figures 4(a)-(b) provide an example of conditional log-pdfs, whereas Figure 4(c) shows a "sharp" conditional density, corresponding to Figure 4(b). This pdf resembles a delta function: even using sophisticated adaptive techniques, it is difficult to recognize the mode of this kind of target pdf.

		$\lambda = 0.001$	$\lambda = 0.005$	$\lambda = 0.01$	$\lambda = 0.05$	$\lambda = 0.08$	$\lambda = 0.10$
FUSS-MH-P2	MSE(R)	0.0071	0.0089	0.0093	0.0138	0.0150	0.0778
	MSE(Ω)	5.01 10^{-5}	6.15 10^{-5}	6.15 10^{-5}	5.26 10^{-5}	7.33 10^{-5}	1.78 10^{-4}
MH ($\boldsymbol{\sigma_p}$= 1)	MSE(R)	0.6830	0.7264	0.7067	1.1631	1.3298	1.3293
	MSE(Ω)	0.0373	0.0402	0.0423	0.0399	0.0471	0.0440
MH ($\boldsymbol{\sigma_p}$= 2)	MSE(R)	1.3566	1.4906	1.4247	2.0015	2.3042	2.2401
	MSE(Ω)	0.0897	0.1117	0.1041	0.0989	0.1089	0.1125

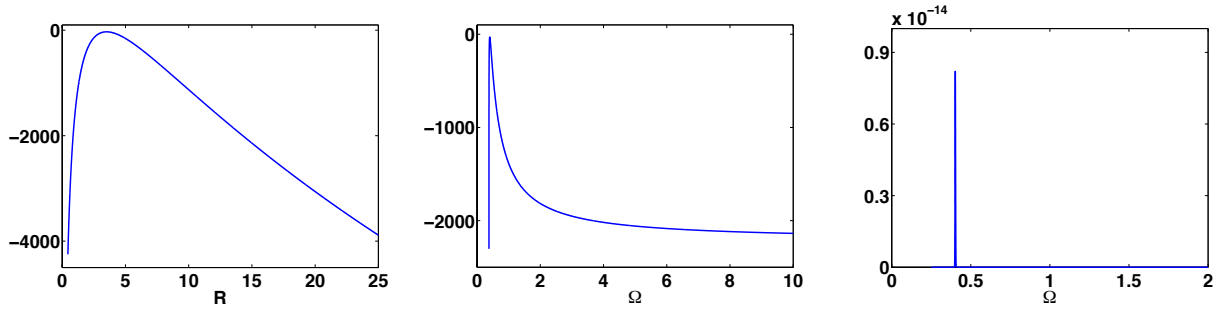
Table 14. MSEs in estimation of R and Ω using FUSS-MH-P2 and MH inside a Gibbs sampler, with $\delta = 10^{-3}$, $K = 10$ and $N_G = 50$. The observed sequence $z_{1:T}$ is generated with $R = 3.7$, $\Omega = 0.4$, $T = 20$ and different values of λ.

6.4. Jump diffusion model for stock market index

We consider the inference problem in a model which is combination of the stochastic volatility model and infinite-activity Lévy jumps as in [16]. Specifically, suppose Y_t is the daily log-return of some stock market index, i.e. $Y_t = \log(S_t)$, where S_t is the daily index value, and let v_t be the instantaneous variance of return, $t \in \mathbb{N}$. We assume that the joint dynamics of the index return and variance can be described by the following stochastic equations,

$$\begin{aligned} Y_{t+1} - Y_t &= \left(r - \frac{1}{2}v_t + \phi_J(-i) + \eta_s v_t\right)\Delta + \\ &\quad + \sqrt{v_t \Delta}\epsilon_{t+1,1} + J_{t+1}, \\ v_{t+1} - v_t &= k(\theta - v_t)\Delta + \sigma_v \sqrt{v_t \Delta}\epsilon_{t+1,2}, \end{aligned} \tag{13}$$

[6] We use the simplest pruning procedure (P2) in order to show that, even in this case, FUSS provides good performance.



(a) (b) (c)

Fig. 4. **(a)-(b)** Examples of (unormalized) conditional log-pdfs with $\lambda = 0.1$ and considering $T = 20$ observations. **(a)** Fixing $\Omega = 4$. **(b)** Fixing $R = 0.7$. **(c)** The (unormalized) conditional pdf corresponding to Figure (b). Even advanced and adaptive MCMC techniques often fail for drawing from this kind of sharp densities.

where r is the risk-free interest rate, $\eta_s v_t$ is the risk premium, and Δ is the time interval of the discretization on the daily basis (we set $\Delta = \frac{1}{252}$). In the variance process, k measures the speed of mean reverting, θ is the long-term mean of variance v_t, and σ_v is the volatility of variance. $\epsilon_{t,1}$ and $\epsilon_{t,2}$ are two noise perturbations distributed according a bivariate Gaussian pdf, specifically,

$$(\epsilon_{t,1}, \epsilon_{t,2}) \sim \mathcal{N}([0,0]^\top, \Sigma),$$

where $\Sigma = [1,\ \rho;\ \rho,\ 1]^\top$. We also consider a Lévy jump J_t in the process of Y_t, and $\phi_J(-i)$ is its jump compensator, defined as $E^Q[e^{iuJ_t}] = e^{-t\phi_J(u)}$ where Q is the so-called risk-neutral probability measure. In our model setting, we employ the Variance Gamma process proposed by [20] as J_t, and for simplicity we assume the jump-relevant parameters γ, ν, and σ remain unchanged under Q, i.e.,

$$\begin{aligned} J_t &= \gamma G_t(1,\nu) + \sigma W_{G_t(1,\nu)}, \\ \phi_J(-i) &= \frac{\log(1 - \gamma\nu - \frac{1}{2}\nu\sigma^2)}{\nu}, \end{aligned} \tag{14}$$

where G_t is an independent Gamma process that we consider known, and $W_{G_t(1,\nu)}$ is a Brownian motion subordinated by G_t. In the market we observe the daily index return Y_t, $t = 1, \ldots, T$. To calibrate the model, we need to filter all the latent variables v_t, J_t, and the fixed parameters $\Theta = \{k, \theta, \rho, \sigma_v, \eta_s, \gamma, \nu, \sigma\}$. In particular, since all the latent variables are varied from day to day, for example $\mathbf{v} = [v_1, \ldots, v_T]^\top$ with $T = 252$ (in one year), the estimation of this model is very challenging owing to the high dimension.

We apply a Gibbs sampler to draw from the complete posterior distribution. The full-conditional pdfs of several latent variables and parameters have a standard form as Gaussian, inverse Gamma and Gamma densities so that direct sampling methods are available. However, the conditional pdfs corresponding to v_t, $t = 1, ..., 252$, and the parameter ν, have a complex analytic form, thus we apply FUSS-RC-P4 for drawing from them. We also apply a standard MH algorithm, with a Gaussian random walk proposal $\bar{p}(x_t|x_{t-1}) \propto \exp\left(-\frac{(x_k - x_{k-1})^2}{2\sigma_p^2}\right)$, to compare the performance. We set $x_0 = 0.02$ (regarding v) or $x_0 = 0.5$ (regarding ν), $K = 3$ for both methods. We set $\sigma_p = 0.01$ for MH. We chose these values of x_0 and σ_p after several attempts, in order to provide the best results of the MH method. Furthermore, for FUSS-RC-P4, we chose $s_1 = 0.0001$, $s_M = 6$, $\Delta_{s_i} = s_{i+1} - s_i = 0.0001$, $\delta = 0.01$. In this example, we change the number of iterations of the Gibbs sampler, $N_g = \{10, 100, 500, 5 \cdot 10^3\}$. For the MH method, we also consider $N_g = 10^4$.

We simulate artificial data by the system using as true parameters $k = 4$, $\theta = 0.02$, $\sigma_v = 0.3$, $\rho = -0.8$, $\eta_s = 0.4$, $\nu = 0.1$, $\gamma = 0.1$, $\sigma = 0.1$, $Y_0 = 7$, $v_0 = 0.02$, and $r = 0.02$ (we recall that the jump component G_t is assumed known). Then we compute the MSEs in the estimation of ν and the entire vector $\mathbf{v}$.

The results, averaged over 100 independent runs, are provided in Table 15. FUSS-RC-P4 always outperforms the standard MH method. We can also observe that FUSS-RC-P4 with only $N_g = 10$ provides a smaller MSE($\mathbf{v}$) than the MH method with $N_g = 10^4$.

Figure 5, which compares the filtered paths by FUSS-RC-P4 (dashed line) and MH (dotted-dashed line) with the simulated true path (solid line) of v_t, demonstrates that MH can yield quite poor results if there is no much prior information on v_t. In particular, given that it is a flat curve for initial variance with $v_t = 0.02$ for all $t = 1 \ldots 252$, MH did not achieve the true

Technique	N_g	MSE(v)	MSE(ν)
FUSS-RC-P4	10	$3.989\ 10^{-5}$	$1.460\ 10^{-2}$
	100	$2.542\ 10^{-5}$	$1.642\ 10^{-3}$
	500	$1.798\ 10^{-5}$	$3.783\ 10^{-4}$
	5000	$1.138\ 10^{-5}$	$9.119\ 10^{-5}$
MH	10	$3.330\ 10^{-4}$	$1.289\ 10^{-1}$
	100	$3.392\ 10^{-4}$	$3.093\ 10^{-2}$
	500	$3.289\ 10^{-4}$	$1.264\ 10^{-2}$
	5000	$2.754\ 10^{-4}$	$7.514\ 10^{-3}$
	10^5	$7.660\ 10^{-5}$	$1.368\ 10^{-3}$

Table 15. MSEs obtained by FUSS-RC-P4 and MH methods for different number of iterations N_g of the Gibbs sampler.

variance dynamics being sensitive to initial flat values whereas FUSS-RC-P4 captured the actual variance dynamics relatively well.

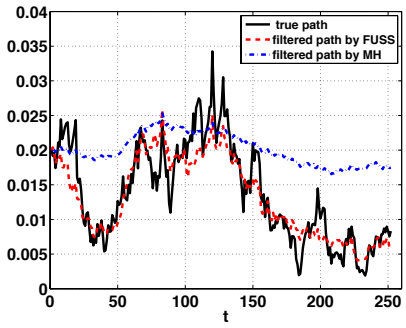


Fig. 5. Filtering of variance path v_t, $t = 1, \ldots, 252$, by FUSS-RC-P4 and $N_g = 5000$ and MH with $N_g = 10^5$. The path is the average of the filtered paths in 100 different experiments.

7. CONCLUSIONS

In this work, we have introduced a novel, extremely efficient, MCMC sampler for drawing from univariate densities. We have tested the new technique in several scenarios, with different target distributions. The performance achieved by the new methodology are close to an exact sampler, yielding virtually i.i.d. samples. The novel technique outperforms other well-known MCMC methods such as the Metropolis-Hastings (MH) method and the slice sampling in terms of accuracy and speed. Furthermore, the dependence of the initial parameters is also drastically reduced with respect to other MCMC techniques. Numerical simulations also show a clear benefit of using the FUSS schemes within a Gibbs sampler. Finally, although the proposal is tailored to the target and the performance are remarkable, the new technique is *not* an adaptive MCMC, hence there are not issues with the ergodicity.

8. ACKNOWLEDGEMENTS

This work has been supported by the ERC grant 239784 and AoF grant 251170.

References

[1] N. C. Beaulieu and C. Cheng. Efficient Nakagami-m fading channel simulation. *IEEE Transactions on Vehicular Technology*, 54(2):413–424, 2005.

[2] A. Boyarsky and P. Góra. *Law of Chaos*. Birkhöuser, Boston (USA), 1997.

[3] 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.

[4] A. Doucet and X. Wang. Monte Carlo methods for signal processing. *IEEE Signal Processing Magazine*, 22(6):152–170, Nov. 2005.

[5] W. J. Fitzgerald. Markov chain Monte Carlo methods with applications to signal processing. *Signal Processing*, 81(1):3–18, January 2001.

[6] W. R. Gilks. Derivative-free Adaptive Rejection Sampling for Gibbs Sampling. *Bayesian Statistics*, 4:641–649, 1992.

[7] W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 44(4):455–472, 1995.

[8] W. R. Gilks, R.M. Neal, N. G. Best, and K. K. C. Tan. Corrigidum: Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, 46(4):541–542, 1997.

[9] W. R. Gilks and P. Wild. Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics*, 41(2):337–348, 1992.

[10] H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14:375–395, 1999.

[11] F. Hartig and C. F. Dormann. Does model-free forecasting really outperform the true model? *Proceedings of the National Academy of Sciences (PNAS)*, 110(42):E3975, 2013.

[12] L. Holden, R. Hauge, and M. Holden. Adaptive independent Metropolis-Hastings. *The Annals of Applied Probability*, 19(1):395–413, 2009.

[13] J. M. Keith, D. P. Kroese, and G. Y. Sofronov. Adaptive independence samplers. *Statistics and Computing*, 18(4):409–420, 2008.

[14] K. R. Koch. Gibbs sampler by sampling-importance-resampling. *Journal of Geodesy*, 81(9):581–591, 2007.

[15] J. Kotecha and Petar M. Djurić. Gibbs sampling approach for generation of truncated multivariate Gaussian random variables. *Proceedings of Acoustics, Speech, and Signal Processing, (ICASSP)*, 1999.

[16] H. Li, M. Wells, and C. Yu. A Bayesian analysis of return dynamics with Lévy jumps. *Review of Financial Studies*, 21(5):2345–2378, 2008.

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

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

[19] D. Luengo and L. Martino. Almost rejectionless sampling from Nakagami-m distributions ($m \geq 1$). *IET Electronics Letters*, 48(24):1559–1561, 2012.

[20] D. Madan, P. Carr, and E. Chang. The variance gamma process and option pricing. *European finance review*, 2(1):79–105, 1998.

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

[22] L. Martino and J. Read. On the flexibility of the design of multiple try Metropolis schemes. *arXiv:1201.0646 (submitted to Computational Statistics)*, January 2012.

[23] L. Martino, J. Read, and D. luengo. Independent doubly adaptive rejection Metropolis sampling. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[24] R. Meyer, B. Cai, and F. Perron. Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. *Computational Statistics and Data Analysis*, 52(7):3408–3423, March 2008.

[25] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[26] S. Olver and A. Townsend. Fast inverse transform sampling in one and two dimensions. *arXiv:1307.1223*, 2013.

[27] C. T. Perretti, S. B. Munch, and G. Sugihara. Model-free forecasting outperforms the correct mechanistic model for simulated and experimental data. *Proceedings of the National Academy of Sciences (PNAS)*, 110(13):5253–5257, 2013.

[28] C. T. Perretti, S. B. Munch, and G. Sugihara. Reply to hartig and dormann: The true model myth. *Proceedings of the National Academy of Sciences (PNAS)*, 110(42):E3976–E3977, 2013.

[29] J. G. Proakis. *Digital Communications*. McGraw-Hill, Singapore, 1995.

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

[31] J. K. O. Ruanaidh and W. J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing*. Springer, 1996.

[32] W. Shao, G. Guo, F. Meng, and S. Jia. An efficient proposal distribution for Metropolis-Hastings using a b-splines technique. *Computational Statistics and Data Analysis*, 53:465–478, 2013.

[33] L. Tierney. Exploring posterior distributions using Markov Chains. *Computer Science and Statistics: Proceedings of IEEE 23rd Symp. Interface*, pages 563–570, 1991.

[34] L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

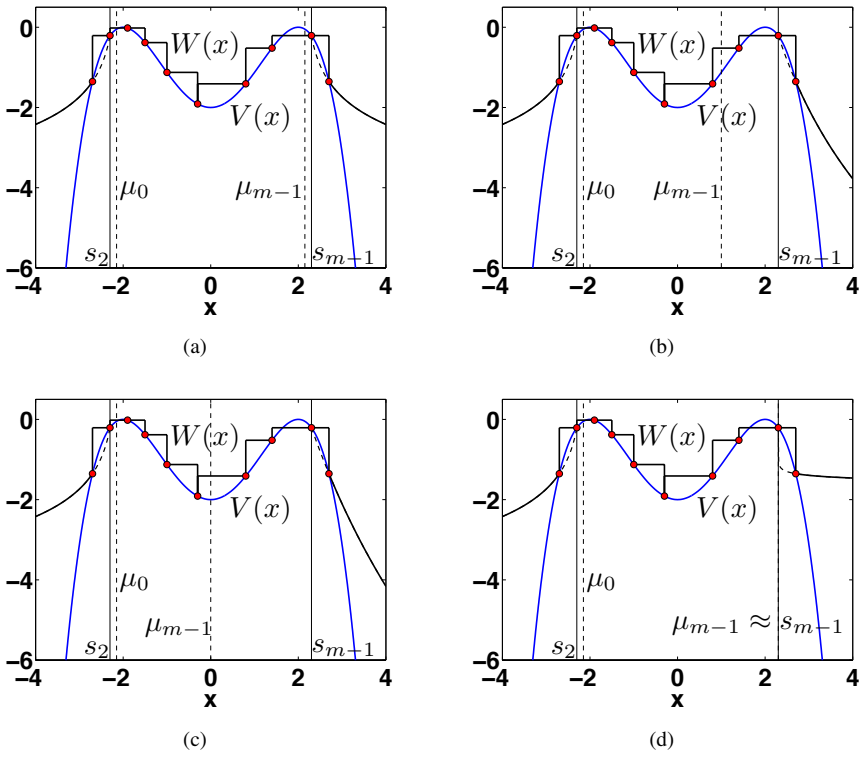


Fig. 6. Example of construction of heavy tails in the log-domain of the pdf. **(a)** Example with a specific choice of the parameters $\mu_0 > s_2$ and $\mu_{m-1} < s_{m-1}$ ($m = 10$ in figure). **(b)**-**(c)** Alternative right log-tail constructions, decreasing the parameter μ_{m-1}. For $\mu_{m-1} \to -\infty$, the log-tail tends to be a straight line passing through $(s_{m-1}, V(s_{m-1}))$, $(s_m, V(s_m))$. **(d)** Construction of the right log-tail with $\mu_{m-1} \approx s_{m-1}$. In this case, $\mu_{m-1} \to s_{m-1}$, it tends to be a constant line.

A. TAILS

The choice of the tails is important for two reasons: (a) to accelerate the convergence of the chain to the target (in the case, for instance, of heavy-tailed target distributions) and (b) to increase the robustness of the method w.r.t. the initial choice of the set

$\mathcal{S}_M$. Indeed, often the construction of tails with a bigger area below them can reduce the dependence on a specific choice of the initial support points. When it is impossible, a good choice is to build tails such that $p(x) \geq \pi(x)$ for $x \in \mathcal{I}_0$ and $x \in \mathcal{I}_m$. This is always possible when the target pdf has light tails (i.e., convex tails in the log-domain) as we have shown in Figure 1 and explained below.

A.1. Light Tails

In order to use light tails, we consider two exponential pieces

$$\begin{aligned} p(x|\mathcal{S}_m) &= e^{h_0 x + b_0}, \ \ \forall x \in \mathcal{I}_0, \\ p(x|\mathcal{S}_m) &= e^{h_m x + b_m}, \ \ \forall x \in \mathcal{I}_m. \end{aligned} \tag{15}$$

The linear function, $w_0(x) = h_0 x + b_0$ is the straight line passing through the points $(s_1, V(s_1))$ and $(s_2, V(s_2))$, whereas the second linear function $w_m(x) = h_m x + b_m$ is the straight line passing through $(s_{m-1}, V(s_{m-1}))$ and the last point $(s_m, V(s_m))$. Note that we can easily compute analytically the area below each piece, and we can also easily draw from each exponential tail by inversion method [30].

A.2. Heavy Tails

For heavy tail constructions, there are also several possibilities. For instance, here we propose to use Pareto pieces with the analytic form

$$\begin{aligned} p(x|\mathcal{S}_m) &= e^{\rho_0} \frac{1}{|x - \mu_0|^{\gamma_0}}, \ \ \forall x \in \mathcal{I}_0, \\ p(x|\mathcal{S}_m) &= e^{\rho_m} \frac{1}{|x - \mu_m|^{\gamma_m}}, \ \ \forall x \in \mathcal{I}_m, \end{aligned} \tag{16}$$

directly in the pdf domain with $\gamma_j > 1$, $j \in \{0, m\}$. In the log-domain, this is equivalent to

$$\begin{aligned} w_0(x) &= \rho_0 - \gamma_0 \log(|x - \mu_0|), \quad \text{for} \quad x \in \mathcal{I}_0, \\ w_m(x) &= \rho_m - \gamma_m \log(|x - \mu_m|), \quad \text{for} \quad x \in \mathcal{I}_m, \end{aligned}$$

Fixed the parameters μ_j, $j \in \{0, m\}$, the remaining ρ_j and γ_j are set in order to satisfying the passing conditions through the points $(s_1, V(s_1))$, $(s_2, V(s_2))$, and through the points $(s_{m-1}, V(s_{m-1}))$, $(s_m, V(s_m))$, respectively. The parameters μ_j can be arbitrarily chosen by the user, fulfilling the inequalities

$$\mu_0 > s_2, \quad \mu_m < s_{m-1}.$$

Values of μ_j such that $\mu_0 \approx s_2$ and $\mu_m \approx s_{m-1}$ yield small values of γ_j (close to 1) and, as a consequence, fatter tails. Greater differences $|\mu_0 - s_2|$ and $|\mu_m - s_{m-1}|$ yield $\gamma_j \to +\infty$, i.e., lighter tails. In the limit case, they coincide with the exponential construction presented above. Different examples of construction are illustrated in Figure 6. As in the previous case, we can compute analytically the integral of $p(x|\mathcal{S}_m)$ in $\mathcal{I}_0$ and $\mathcal{I}_m$. We can also easily draw from each Pareto tail by inversion method [30].