SYRTO

Systemic Risk Tomography
Signals, Measurements, Transmission channels and Policy Interventions

Adaptive Sticky Generalized Metropolis

Luca Martino, Roberto Casarin, Fabrizio Leisen, David Luengo

SYRTO WORKING PAPER SERIES
Working paper n. 10 | 2013

This project is funded by the European Union under the 7th Framework Programme (FP7-SSH/2007-2013) Grant Agreement n° 320270

Adaptive Sticky Generalized Metropolis

Luca Martino[†] Roberto Casarin[‡]
Fabrizio Leisen[§ ‖] David Luengo[¶]

[†]Universidad Carlos III de Madrid

[‡]University Ca' Foscari, Venice

[§]University of Kent

[¶]Universidad Politecnica de Madrid

Abstract

We introduce a new class of adaptive Metropolis algorithms called adaptive sticky algorithms for efficient general-purpose simulation from a target probability distribution. The transition of the Metropolis chain is based on a multiple-try scheme and the different proposals are generated by adaptive nonparametric distributions. Our adaptation strategy uses the interpolation of support points from the past history of the chain as in the adaptive rejection Metropolis. The algorithm efficiency is strengthened by a step that controls the evolution of the set of support points. This extra stage improves the computational cost and accelerates the convergence of the proposal distribution to the target. Despite the algorithms are presented for univariate target distributions, we show that they can be easily extended to the multivariate context by a Gibbs sampling strategy. We show the ergodicity of the proposed algorithms and illustrate their efficiency and effectiveness through some simulated examples involving target distributions with complex structures.

Keywords: Adaptive Markov chain Monte Carlo, Adaptive rejection Metropolis, Multiple-try Metropolis, Metropolis within Gibbs.

1 Introduction

Markov Chain Monte Carlo (MCMC) methods (see Liu (2004); Liang et al. (2010); Robert and Casella (2004) and references therein) are now a very important numerical tool in statistics and in many others

‖Corresponding author: Fabrizio Leisen, fabrizio.leisen@gmail.com. Other contacts: r.casarin@unive.it (Roberto Casarin); luca.martino@uc3m.es (Luca Martino).

fields, because they can generate samples from any target distribution available up to a normalizing constant. The standard MCMC techniques require the specification of a proposal distribution and produce a Markov chain that converges to the target distribution. A crucial issue in MCMC is the choice of the proposal distribution, which can heavily affect the mixing of the MCMC chain when the target distribution has a complex structure, e.g., multimodality and heavy tails. Thus, in the last decade and after the seminal paper of Haario et al. (2001), a remarkable stream of literature focuses on adaptive proposal distributions, which allow for self-tuning procedures of the MCMC algorithms, flexible movements within the sample space and reasonable acceptance rates.

Adaptive MCMC algorithms are used in many statistical applications (e.g., see Roberts and Rosenthal (2009), Craiu et al. (2009), Giordani and Kohn (2010)) and different adaptive strategies have been proposed in the literature. One of the strategies consists in updating the proposal distribution according to the past values of the chain (e.g., see (Haario et al., 2001) and Andrieu and Robert (2001)). Another strategy relies on the use of auxiliary chains, which are run in parallel and interact with the principal chain (e.g., see Jasra et al. (2007), Casarin et al. (2013)).

One of the most used class of MCMC algorithms, is the Metropolis-Hastings (MH) algorithm (see Metropolis et al. (1953) and Hastings (1970)) and its generalizations. Among the different variants of the MH, in this paper we focus on multiple-try Metropolis (MTM) (see Liu et al. (2000)), which have revealed to be efficient in different applications (e.g., see Craiu and Lemieux (2007) and So (2006)). While in the MH formulation one accepts or rejects a single proposed move, the MTM is designed so that the next state of the chain is selected among multiple proposals. The multiple-proposal setup can be used effectively to explore the sample space of the target distribution. The MTM has been further generalized with the use of antithetic and quasi-Monte Carlo sampling (Craiu and Lemieux (2007) and Bédard et al. (2012)), and the use of general weighting function in the selection step of the MTM (Martino and Read (2012) and Martino and Read (2013)).

We contribute to the adaptive MCMC literature by proposing a new class of adaptive generalized Metropolis algorithms. More specifically, we propose adaptive sticky MTM (ASMTM) which has the adaptive sticky Metropolis (ASM) as a special case. Adaptation strategies for MTM based on interacting chains have been proposed in Casarin et al. (2013). We follow here an alternative route and use the past iterations of the MTM algorithm to adapt the proposal distribution over the chain iterations. The proposal distribution is nonparametric and the construction method relies upon alternative interpolation strategies. Our adaptation mechanism also builds on and extends the adaptation mechanism in the adaptive rejection sampling (ARS) (Gilks and Wild, 1992) and in the accept/reject Metropolis (ARMS) (Gilks et al., 1995b) and its extensions (e.g., see Meyer et al. (2008),

Cai et al. (2008) and Martino et al. (2012)). We shall notice that the interpolation approach has been used also in Krzykowski and Mackowiak (2006) and Shao et al. (2013), but not in an adaptive MH framework. Our extension of the algorithms in the ARMS class is twofold. First we use the more efficient multiple-proposal transition instead of the single proposal transition kernel. Secondly we apply a random test procedure for the inclusion of new points in the support set of the proposal distribution. We discuss different testing procedures for the inclusion of new support points. They represent more efficient generalizations of the accept/reject rule of the ARMS algorithm.

Another contribution of the paper regards the converge of the proposed adaptive algorithms. Adaptive MCMC algorithms, which use previous iterations or auxiliary variables in their future transitions, violate the Markov property which provides the justification for conventional MCMC. Thus, their validity in terms of convergence to the desired target distribution, has to be demonstrated. We shall notice that convergence of adaptive MCMC is reached under various conditions (Haario et al. (2001), Atchade and Rosenthal (2005), Andrieu and Moulines (2006), Roberts and Rosenthal (2007), Saksman and Vihola (2010), Latuszynski et al. (2013), and Holden et al. (2009)). In this paper we follow the Holden et al. (2009) approach and show the ergodicity of the adaptive Metropolis chain under suitable conditions on the proposal distribution. Our interpolation approach guaranties that the adaptive proposal distributions satisfy such conditions. These results extend to adaptive MTM algorithm the previous results on adaptive MH due to Holden et al. (2009).

The structure of the paper is as follows. Section 2 introduces adaptive sticky Metropolis and discusses convergence issues. Section 3 presents different updating schemes for the proposal distributions. Section 4 discusses some practical issues for the implementation and some acceleration strategies for reducing the computational cost. Section 5 presents a multivariate extension based on a Gibbs sampling updating rule. Section 6 contains algorithm comparisons using simulated data. Section 7 contains conclusions and suggestions for further research.

2 Adaptive Generalized Metropolis

2.1 Adaptive Sticky Metropolis

Let $\pi(x)$ be a real target distribution known up the normalizing constant. Fix an initial state x_0 of the chain x_t, $t = 0, 1, 2, \ldots$, and an initial set of support points $\mathcal{S}_0 = \{s_1, \ldots, s_{m_0}\}$, with $m_0 > 0$. Assume that the current state of the chain is x_t, then the general update of the proposed Adaptive Sticky Metropolis (ASM) algorithm is described in Algorithm 1.

Algorithm 1. Adaptive Sticky Metropolis (ASM)

For $t = 1, \ldots, T$:

1. **Construction of the proposal:** Build a proposal $q_t(x|\mathcal{S}_{t-1})$ via a suitable interpolation procedure using the set of support points $\mathcal{S}_{t-1}$.

2. **MH step:**

   2.1 Draw x' from $q_t(x|\mathcal{S}_{t-1})$.

   2.2 Set $x_{t+1} = x'$ and $z = x_t$ with probability

$$\alpha(x_t, x', \mathcal{S}_{t-1}) = \min\left[1, \frac{\pi(x')q_t(x_t|\mathcal{S}_{t-1})}{\pi(x_t)q_t(x'|\mathcal{S}_{t-1})}\right],$$

   and set $x_{t+1} = x_t$ and $z = x'$, with probability $1 - \alpha(x_t, x', \mathcal{S}_{t-1})$.

3. **Test to update $\mathcal{S}_t$:** Let $\eta : \mathbb{R}^+ \to [0, 1]$ be a strictly increasing continuous function such that $\eta(0) = 0$. Then, set

$$\mathcal{S}_t = \begin{cases} \mathcal{S}_{t-1} \cup \{z\} & \text{with prob. } \eta(d_t(z)), \\ \mathcal{S}_{t-1} & \text{with prob. } 1 - \eta(d_t(z)), \end{cases}$$

   where $d_t(z)$ is a positive measure (at the iteration t) of the distance in z between the target and the proposal distributions.

The proposal distribution changes along the iterations (see step 1 of Algorithm 1) following an adaptation scheme which relies upon a suitable interpolation of a set of support points. In Section 3 we provide several interpolation methods based on a partition of the support of $\pi(x)$. The insight behind this adaptation strategy is to build a proposal that is closer and closer to the target as the number of iterations increases.

The proposal generated from the updated distribution are then used in a standard accept-reject Metropolis-Hastings (MH) step (see step 2 of the algorithm), hence the resulting algorithm is in the class of adaptive MH.

Another important feature of the proposed adaptation strategy is given by the test for updating the set of support points (see step 3). This step includes with probability η the rejected proposal from the MH step in the set of support points by applying an accept-reject rule. The ratio behind this test is to use information from the target distribution in order to include in the set only the points where the proposal is far from the target. More specifically, we set the acceptance probability η as a function of a distance $d_t(z)$. This allows to design a strategy that incorporates the point z only if distance in z between the proposal distribution and the target is large. Moreover, a suitable construction of the proposal leads to a probability of adding a new point that converges to zero. This

implies that both the total number of points in the support set and the computational cost of building the proposals are kept bounded along the iterations, provided that $\eta(0) = 0$. Different choices of η, which ensure quick convergence of the proposal to the target, are presented in Section 4.1.

Finally, it should be noted that Algorithm 1 is a special case of the adaptive sticky MTM presented in the next section (see Algorithm 2) and the proof of the validity of the algorithm follows closely the proof given in next session for the adaptive sticky MTM and, therefore, it is not given here.

2.2 Adaptive Sticky Multiple Try Metropolis

In the ASM one accepts or rejects a single proposed value. We extend the ASM by allowing for multiple-proposals in order to further improve the ability of the Metropolis chain to explore the state space. We focus on the multiple-try Metropolis (MTM) (see Liu et al. (2000) and Craiu and Lemieux (2007)) and propose an Adaptive Sticky MTM (ASMTM). The ASMTM can also be seen as a generalization of the MTM which allows for adaptive proposal distributions. Note that our adaptation strategy can be combined with MTM algorithms with different proposal distributions and with interacting MTM algorithms (see Casarin et al. (2013)) to design new adaptive algorithms. We adaptation can be also used within the multi-point algorithms (e.g., Martino and Read (2012)) as well.

At the iteration t, the ASMTM builds the proposal distribution $q_t(x|\mathcal{S}_{t-1})$ (step 1 of Algorithm 2) using the current set of support points $\mathcal{S}_{t-1}$. Let $x_t = x$ be the current value of the chain and x'_j, $j = 1, \ldots, M$, a set of i.i.d. proposals simulated from $q_t(x|\mathcal{S}_{t-1})$ (see step 2). Moreover, let $w_{jt}(x, x'_j) = \pi(x)q_t(x'_j|\mathcal{S}_{t-1})\lambda_t(x, x'_j|\mathcal{S}_{t-1})$ be the unnormalized selection weights, where $\lambda_t(x, x'|\mathcal{S}_{t-1})$ is a non-negative symmetric function in x and x'. It is worth noticing that in the adaptive MTM not only the proposal distribution changes over the iterations, but also the function λ_t may adapt following the update in the set of support points.

Algorithm 2. Adaptive Sticky Multiple Try (ASMTM)

For $t = 1, \ldots, T$:

1. **Construction of the proposal:** Build a proposal $q_t(x|\mathcal{S}_{t-1})$ via a suitable interpolation procedure using the set of support points $\mathcal{S}_{t-1}$. In Section 3 we provide several procedures that are based in a partition of the support of $\pi(x)$.

2. **MTM step:**

   2.1 Draw $x'_1, \ldots, x'_M$ from $q_t(x|\mathcal{S}_{t-1})$ and compute the weights $w_t(x'_i) = \frac{\pi(x'_i)}{q_t(x'_i|\mathcal{S}_{t-1})}$.

   2.2 Select $x' = x'_j$ among the M proposals with probability proportional to $w_t(x'_i)$, $i = 1, \ldots, M$.

   2.3 Set the auxiliary point $x^*_i = x'_i$ and $z_i = x'_i$, $i \neq j$ and $x^*_j = x_t$

   2.4 Set $x_{t+1} = x'$ and $z_j = x^*_j$ with probability

$$\alpha(x_t, x', \boldsymbol{x}'_{-j}, \mathcal{S}_{t-1}) = \min\left[1, \frac{w_t(x'_1) + \cdots + w_t(x'_M)}{w_t(x^*_1) + \cdots + w_t(x^*_M)}\right],$$

   and set $x_{t+1} = x_t$ and $z_j = x'_j$, with probability $1 - \alpha(x_t, x', \boldsymbol{x}'_{-j}, \mathcal{S}_{t-1})$.

3. **Test to update $\mathcal{S}_t$:** Let $\eta_i : \mathbb{R}^+ \to [0, 1]$, $i = 1, \ldots, M$, be strictly increasing continuous functions such that $\eta_i(0) = 0$, $\forall i$ and $\sum_{i=1}^M \eta_i \leq 1$. Then, set

$$\mathcal{S}_t = \begin{cases} \mathcal{S}_{t-1} \cup \{z_i\} & \text{with prob. } \eta_i(d_t(z_i)), i = 1, \ldots, M \\ \mathcal{S}_{t-1} & \text{with prob. } 1 - \sum_{i=1}^M \eta_i(d_t(z_i)), \end{cases}$$

   where $d_t(z)$ is a positive measure (at the iteration t) of the distance in z between the target and the proposal distributions.

Liu et al. (2000) discussed various possible specifications of the function λ_t and found in their experiments that the efficiency gain when using MTM is generally not sensitive to the choice of this function. However, in some of the experiments of Liu et al. (2000) and in quite all the simulation experiments of Casarin et al. (2013), the choice $\lambda_t(x, x'|\mathcal{S}_{t-1}) = 1/(q_t(x|\mathcal{S}_{t-1})q_t(x'|\mathcal{S}_{t-1}))$ leads to better performance of the MTM algorithms. Thus, in this work we consider this choice of λ_t and focus on $w_{jt}(x, x') = w_t(x)$, $\forall j$, where $w_t(x)$ are unnormalized importance weights

$$w_t(x) = \frac{\pi(x)}{q_t(x|\mathcal{S}_{t-1})}.$$

The importance weights are used at the step 2 of the ASMTM to select one of the proposals. The selected candidate is accepted or rejected with the generalized acceptance probability given at step 2. Finally, step 3 includes the selected proposal in the set of support points, with probability η. This

updating step can be extended to allow for more than one proposals to be included into the set of support points. The strategy leads to recycle the proposals and possibly improves the adaptation of the proposal distributions. For the sake of simplicity, in the presentation of the ASMTM algorithm, we consider the case only one proposal is added, at each iteration, to $\mathcal{S}_{t-1}$.

We show the convergence of the ASMTM algorithm by extending to the MTM the results in Holden et al. (2009) where they show the convergence for independent MH scheme with adaptive proposal avoiding the requirement of diminishing adaptation. The difference between the adaptive independent MH algorithm of Holden et al. (2009) and a standard independent MH algorithm is that the proposal distribution $q_t(x|\mathcal{S}_{t-1})$ depend on the set of support points $\mathcal{S}_{t-1}$, which can include part of the past history of the MH algorithm except for the current state of the MH chain (see Liang et al. (2010), pp. 312-315). The main difference between our adaptive independent MTM algorithm and the adaptive independent MH algorithm of Holden et al. (2009) is that the at each iteration multiple-proposals can be used in the Metropolis transition. The following theorem implies that the AMTM chain never leaves the stationary distribution $\pi(x)$ once it is reached.

Theorem 1. *The target distribution $\pi(x)$ is invariant for the adaptive independent MTM algorithm; that is, $p_t(x_t|\mathcal{S}_{t-1}) = \pi(x_t)$ implies $p_{t+1}(x_{t+1}|\mathcal{S}_t) = \pi(x_{t+1})$, where $p_t(\cdot|\mathcal{S}_{t-1})$ denotes the distribution of x_t conditional on the past samples.*

Let us assume that the proposal distribution $q_t(x|\mathcal{S}_{t-1})$ satisfies the strong Doeblin's condition

$$q_t(x|\mathcal{S}_{t-1}) \geq a_t(\mathcal{S}_{t-1})\pi(x) \tag{1}$$

for all $x \in \mathcal{X}$ and $\mathcal{S}_{t-1} \in \mathcal{X}^{t-1}$, where $\mathcal{X}$ denotes the state space, and $a_t(\mathcal{S}_{t-1}) \in (0, 1]$. This condition is satisfied in our proposal distributions discussed in the next sections. The proofs of following theorem and Theorem 1 are in Appendix A.

Theorem 2. *Assume the proposal $q_t(x|\mathcal{S}_{t-1})$ in the ASMTM algorithm satisfies the condition 1 for all t. Then*

$$||p_t - \pi||_{TV} \leq 2 \int_{\mathcal{X}^t} \prod_{j=1}^{t} (1 - a_j(\mathcal{S}_{j-1}))d\mu(\mathcal{S}_{t-1}) \tag{2}$$

The algorithm converges if the product goes to zero when $t \to \infty$.

3 Construction of sticky proposal functions

There are many alternatives available for the construction of a suitable proposal distribution in the ASM and ASMTM algorithms. In this section, we focus on certain procedures that approximate the

target distribution interpolating points that belong to the graph of the (unnormalized) target. The points are identified by evaluating the target at the support points and the set of support points change over the algorithm iterations. The name "sticky", we choose for this algorithm, highlights the ability of the adaptation schemes to generate a sequence of proposal distributions which converge to the target, allowing for a full adaptation of the proposal distribution.

The adaptation relies upon interpolation scheme which are easy to improve by adding new points to the support set and are easy to sample. We note that the resulting proposal density can be represented as a mixture of probability density functions, so that to draw from it one need to compute mixture weights, to sample from a discrete distribution in order to choose one of the mixture components and finally to be able to draw samples from the selected component.

In this paper, we will present three novel adaptation strategies for the proposal distributions. Let us assume that a set $\mathcal{S}_t = \{s_1, \ldots, s_{m_t}\}$ of m_t support points is available at the iteration $t+1$ of a Metropolis algorithm. Define a sequence, of $m_t + 1$ intervals: $\mathcal{I}_0 = (-\infty, s_1]$, $\mathcal{I}_j = (s_j, s_{j+1}]$ for $j = 1, \ldots, m_t - 1$, and $\mathcal{I}_{m_t} = (s_{m_t}, +\infty)$. In the first type of adaptation schemes, the proposal distribution is a mixture of $m_t + 1$ densities with bounded disjoint supports $\mathcal{I}_j$, $j = 0, \ldots, m_t$. An addition of a new support point, say s', can change the shape of the densities associated to the different intervals. For instance, if $s' \in \mathcal{I}_k$, then the algorithm will update the mixture components associated with $\mathcal{I}_k$, $\mathcal{I}_{k-1}$ and $\mathcal{I}_{k+1}$. This feature of the adaptation scheme has, as a special case, the construction in Gilks et al. (1995b).

The proposal distribution, in the second type of adaptation schemes, is a mixture of densities with bounded disjoint supports, like the one used in the first method, but the addition of a new support point, say s', can change only one component of the mixture. For instance, if $s' \in \mathcal{I}_k$, then the k-th density of the mixture will be improved. This proposal updating scheme is a simpler alternative to Gilks et al. (1995b). In the following sections, we discuss the three adaptation schemes and illustrate how our sticky proposal construction applies within these schemes.

3.1 Disjoint supports and proposal changes in different intervals

The first adaptation strategy relies upon interpolation for points on the graph of the target. For the sake of simplicity we describe the interpolation procedure representing the target and proposal densities in a log-domain. Hence, let us define the log-density functions

$$W_{t+1}(x) \triangleq \log[q_{t+1}(x|\mathcal{S}_t)], \quad V(x) \triangleq \log[\pi(x)]. \tag{3}$$

where $q_{t+1}(x|\mathcal{S}_t)$ is the proposal at the iteration $t+1$ of the Algorithms 1 and 2 and π is the target distribution. Let us denote as $L_{j,j+1}(x)$ the straight line passing through the points $(s_j, V(s_j))$ and $(s_{j+1}, V(s_{j+1}))$ for $j = 1, \ldots, m_t - 1$ where $s_j \in \mathcal{S}_t$. Also, set

$$L_{-1,0}(x) = L_{0,1}(x) \triangleq L_{1,2}(x), \quad \text{and} \quad L_{m_t,m_t+1}(x) = L_{m_t+1,m_t+2}(x) \triangleq L_{m_t-1,m_t}(x).$$

In Gilks et al. (1995b), $W_{t+1}(x)$ is a piecewise linear function,

$$W_{t+1}(x) = \max\left[L_{j,j+1}(x), \min\left[L_{j-1,j}(x), L_{j+1,j+2}(x)\right]\right], \tag{4}$$

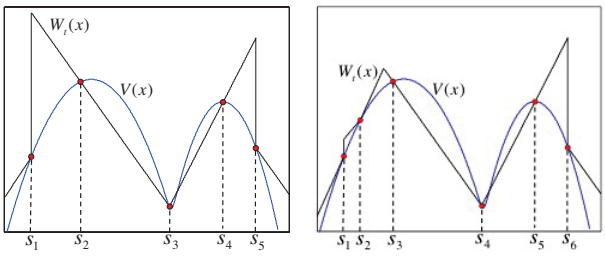
with $x \in \mathcal{I}_i$ where $\mathcal{I}_j = (s_j, s_{j+1}]$, $j = 1, \ldots, m_t - 1$ and $\mathcal{I}_0 = (-\infty, s_1]$ and $\mathcal{I}_{m_t} = (s_{m_t}, +\infty)$. The function $W_{t+1}(x)$ can be re-written as follows

$$W_t(x) = \begin{cases} L_{1,2}(x), & x \in \mathcal{I}_0; \\ \max\left\{L_{1,2}(x), L_{2,3}(x)\right\}, & x \in \mathcal{I}_1; \\ \max\left\{L_{j,j+1}(x), \min\left\{L_{j-1,j}(x), L_{j+1,j+2}(x)\right\}\right\}, & x \in \mathcal{I}_j, \quad 2 \leq j \leq m_t - 2; \\ \max\left\{L_{m_t-1,m_t}(x), L_{m_t-2,m_t-1}(x)\right\}, & x \in \mathcal{I}_{m_t-1}; \\ L_{m_t-1,m_t}(x), & x \in \mathcal{I}_{m_t}. \end{cases} \tag{5}$$

Eq.(**??**) and 5 show that the construction of the log-density in a interval $\mathcal{I}_j$ depends also on the points s_{j-1} and s_{j+2}. Therefore, an addition of a point in a interval can change the construction in the adjacent regions. For instance, let us assume $\mathcal{S}_t = \{s_1, s_2, s_3, s_4, s_5\}$. Fig. 1(a) illustrate the construction using the points in the set $\mathcal{S}_t$. Fig. 1(b) show how the construction change when a new point is added between the points s_1 and s_2 of the set $\mathcal{S}_t$ used Fig. 1(a). As illustrated in Fig. 1(b), intervals $\mathcal{I}_0 = (-\infty, s_1]$, $\mathcal{I}_1 = (s_1, s_2]$ and $\mathcal{I}_2 = (s_2, s_3]$, this construction requires to modify lines for the intervals $\mathcal{I}_0$ and $\mathcal{I}_1$ of Fig. 1(a) and to compute the intersection point between two straight lines (see interval $\mathcal{I}_2 = (s_2, s_3]$ of Fig. 1(b)), to be able to draw adequately from the corresponding proposal distribution. Note that, a similar procedure using pieces of quadratic functions in the log-domain (namely, pieces of truncated Gaussians density in the pdf domain) also has been proposed in Meyer et al. (2008).

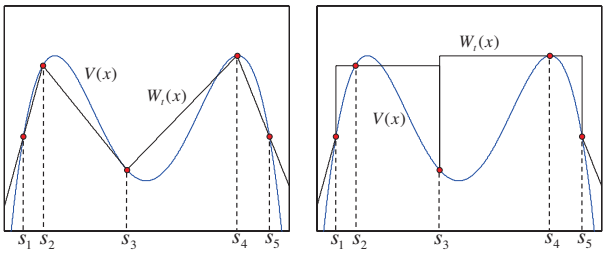
3.2 Disjoint supports and proposal changes in one interval

Gilks et al. (1995b) introduced for the ARMS algorithm the procedure to build $q_{t+1}(x|\mathcal{S}_{t+1})$, described in the previous section. The computational complexity of the procedure arises from the need to construct a proposal function above the target in more regions as possible, in order to take advantage of the rejection sampling step. We note that a simpler approach to build the proposal is to define



(a) (b)

Figure 1: Examples of piecewise linear function, $W_{t+1}(x)$, built using the procedure described in Gilks et al. (1995b) for the set $\mathcal{S}_t = \{s_1, \ldots, s_5\}$ of support points (graph (a)) and the set of support points $s_1, \ldots, s_6$ (graph (b)), obtained by adding a new point between the two points s_1 and s_2 in $\mathcal{S}_t$.



(a) (b)

Figure 2: Examples of the construction of $W_{t+1}(x)$ using the procedures described in Eq. (6) (graph (a)) and in Eq. (7) (graph (b)).

$W_{t+1}(x)$ inside the i-th interval as the straight line passing through $(s_i, V(s_i))$ and $(s_{i+1}, V(s_{i+1}))$, $L_{i,i+1}(x)$, for $1 \leq i \leq m_t - 1$, and extending the straight lines corresponding to $\mathcal{I}_1$ and $\mathcal{I}_{m_t-1}$. Formally, this can be expressed as

$$W_{t+1}(x) = \begin{cases} L_{1,2}(x), & x \in \mathcal{I}_0 = (-\infty, s_1]; \\ L_{i,i+1}(x), & x \in \mathcal{I}_i = (s_i, s_{i+1}], \quad 1 \leq i \leq m_t - 1; \\ L_{m_t-1,m_t}(x), & x \in \mathcal{I}_{m_t} = (s_{m_t}, +\infty). \end{cases} \tag{6}$$

This construction is illustrated in Fig. 2(a). Although this procedure looks similar to the one used in ARMS by Gilks et al. (1995b), it is much simpler in fact, since there is not any minimization or maximization involved, and thus it does not require the calculation of intersection points to determine when one straight line is above the other. Observe that the proposal $q_{t+1}(x|\mathcal{S}_t) = \exp\{W_{t+1}(x)\}$, with such a definition, is formed by exponential pieces (in the pdf-domain). Moreover, an even simpler procedure to construct $W_{t+1}(x)$ can be devised using a piecewise constant approximation with two

straight lines inside the first and last intervals. Mathematically, it can be expressed as

$$W_{t+1}(x) = \begin{cases} L_{1,2}(x), & x \in \mathcal{I}_0 = (-\infty, s_1]; \\ \max\{V(s_i), V(s_{i+1})\}, & x \in \mathcal{I}_i = (s_i, s_{i+1}], \quad 1 \leq i \leq m_t - 1; \\ L_{m_t-1,m_t}(x), & x \in \mathcal{I}_{m_t} = (s_{m_t}, +\infty). \end{cases} \tag{7}$$

The construction described above leads to the simplest proposal density, i.e., a collection of uniform pdfs with two exponential tails. Fig. 2(b) shows an example of the construction of the proposal using this approach. Note that we can also apply the procedure proposed for adaptive trapezoid Metropolis sampling (ATRAMS, Cai et al. (2008)) to build the proposal distribution. However, the structure of the ATRAMS algorithm Cai et al. (2008) is completely different to the ASM and ARMS-type techniques. In both cases the proposal is constructed in the domain of the target pdf, $\pi(x)$, rather than in the domain of the log-pdf, $V(x) = \log(\pi(x))$. For instance, the basic idea proposed for ATRAMS is using straight lines, $\widetilde{L}_{i,i+1}(x)$, passing through the points $(s_i, \pi(s_i))$ and $(s_{i+1}, \pi(s_{i+1}))$ for $i = 1, \ldots, m_t - 1$ and two exponential pieces, $E_0(x)$ and $E_{m_t}(x)$, for the tails:

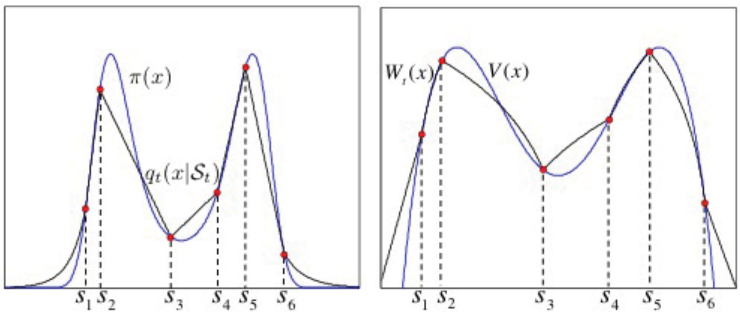
$$q_t(x|\mathcal{S}_t) \propto \begin{cases} E_0(x), & x \in \mathcal{I}_0 = (-\infty, s_1]; \\ \widetilde{L}_{i,i+1}(x), & x \in \mathcal{I}_i = (s_i, s_{i+1}], \quad i = 1, \ldots, m_t - 1; \\ E_{m_t}(x), & x \in \mathcal{I}_{m_t} = (s_{m_t}, +\infty). \end{cases} \tag{8}$$

Unlike in Cai et al. (2008), here the tails $E_0(x)$ and $E_{m_t}(x)$ do not necessarily have to be equivalent in the areas they enclose. Note that $\widetilde{L}$ denotes a straight line built directly in the domain of $\pi(x)$, whereas L denotes the linear function constructed in the log-domain. Indeed, we may follow a much simpler approach calculating two secant lines $L_{1,2}(x)$ and $L_{m_t-1,m_t}(x)$ passing through $(s_1, V(s_1))$, $(s_2, V(s_2))$, and $(s_{m_t-1}, V(s_{m_t-1}))$, $(s_{m_t}, V(s_{m_t}))$ respectively, so that the two exponential tails are defined as $E_0(x) = \exp\{L_{1,2}(x)\}$ and $E_{m_t}(x) = \exp\{L_{m_t-1,m_t}(x)\}$. Fig. 3 depicts an example of the construction of $q_t(x|\mathcal{S}_t)$ using this last procedure. Note that drawing samples from these trapezoidal pdfs inside $\mathcal{I}_i = (s_i, s_{i+1}]$ can be easily done (Cai et al., 2008; Devroye, 1986). Indeed, given $u', v' \sim \mathcal{U}([s_i, s_{i+1}])$ and $w' \sim \mathcal{U}([0, 1])$, then

$$x' = \begin{cases} \min\{u', v'\}, & w' < \frac{\pi(s_i)}{\pi(s_i)+\pi(s_{i+1})}; \\ \max\{u', v'\}, & w' \geq \frac{\pi(s_i)}{\pi(s_i)+\pi(s_{i+1})}; \end{cases} \tag{9}$$

is distributed according to a trapezoidal density defined in the interval $\mathcal{I}_i = [s_i, s_{i+1}]$.

For the approximation methods presented in Sections 3.1-3.2 it is possible to show that the proposal distributions generated by the interpolation algorithm converge to the target distribution when the number of support points goes to infinity.



(a) (b)

Figure 3: Example of the construction of the proposal density, $q_{t+1}(x|\mathcal{S}_t)$, using a procedure described in Cai et al. (2008), within the ATRAMS algorithm, in the pdf domain (graph (a)) and in the log-domain (graph (b)).

Theorem 3. *Consider a continuous bounded target density $\pi(x)$ with bounded second order derivative. Denote with $\tilde{\pi}$ the unormalized density, with $x \in \mathcal{X}$, and with $\{\tilde{q}_t(x|\mathcal{S}_{t-1})\}_{t=1}^{+\infty}$ a sequence of possibly unnormalized proposal density functions such that $\tilde{q}_t(x|\mathcal{S}_{t-1}) > 0$ for all $x \in \mathcal{X}$. Then, $\int_{\mathcal{X}} |\tilde{q}_t(x|\mathcal{S}_{t-1}) - \tilde{\pi}(x)|dx \underset{t\to\infty}{\longrightarrow} 0$*

For sake of simplicity, we denote as $\tilde{q}_t(x|\mathcal{S}_{t-1})$ and $\tilde{\pi}(x)$ the unnormalized density functions whereas $q_t(x|\mathcal{S}_{t-1})$ and $\pi(x)$ indicate the normalized densities. However, we remark that in the rest of this work we have considered $q_t(x|\mathcal{S}_{t-1})$ and $\pi(x)$ as unnormalized pdfs. Therefore, so far the interpolation (or approximation) was applied to the unnormalized target $\tilde{\pi}(x)$ to deal with the general case. Hence the proposal function $q_t(x|\mathcal{S}_{t-1})$ is unnormalized as well. Namely, we build $\tilde{q}_t(x|\mathcal{S}_{t-1})$ via interpolation using the information of $\tilde{\pi}(x)$. We denote the corresponding normalizing constants $1/c_t$ and $1/c_\pi$, respectively. As the $\tilde{q}_t$ converges to $\tilde{\pi}$ in L_1 as t goes to infinity, then the normalizing constants also convergences, i.e. c_t converge to c_π. Indeed, denoting as

$$d(f,g) = ||\tilde{\pi} - \tilde{q}_t|| = \int_{\mathcal{X}} |f(x) - g(x)|dx,$$

the L_1 distance between $f(x)$ and $g(x)$, we have the following result that is proved in Appendix A jointly with Theorem 3.

Theorem 4. *Let $q_t(x|\mathcal{S}_{t-1}) = \frac{1}{c_t}\tilde{q}_t(x|\mathcal{S}_{t-1})$ and $\pi(x) = \frac{1}{c_\pi}\tilde{\pi}(x)$, where $c_\pi = ||\tilde{\pi}|| = \int_{\mathcal{X}} \tilde{\pi}(x)dx$ and $c_t = ||\tilde{q}_t|| = \int_{\mathcal{X}} \tilde{q}_t(x|\mathcal{S}_{t-1})dx$. If $d(\tilde{q}_t, \tilde{\pi}) \underset{t\to\infty}{\longrightarrow} 0$, then $d(q_t, \pi) \underset{t\to\infty}{\longrightarrow} 0$*

Remark. The adaptation procedures presented in the previous sections build proposal distributions with exponential tails. However, the construction of the tails can be easily modified if desired by the user. It is worth to mention that it is not strictly necessary to change the construction of the tails, but there could be some benefits in handling the tails with different approaches. Specifically, we can

diminish the dependence from the initial points and also speed up the convergence of the chain when the target has heavy tails. Furthermore, in a similar fashion, the previous construction procedures can be modified in order to handle unbounded target distributions as well.

4 Practical implementation

4.1 Updating of the set of support points

In this section, we focus on the update step of Algorithm 1-2 where a test is introduced for controlling the evolution of the set of support points. This step can be seen as a measure of similarity between the proposal and target distributions. It is a part of the algorithm that is extremely important since it controls the trade-off between mixing of the Metropolis chain and computational cost. Indeed, the use of large number of support points improves the performance but, at the same time, increases the computational cost. In this step a choice of two functions η and d_t is needed. The first one is a strictly increasing function with values in $[0, 1]$, and d_t is a distance between the proposal and the target distribution. For instance, following the literature on adaptive mixture proposals, one can choose logistic weights and a local absolute distance between proposal and target, which has a low computational cost. These choices corresponds to the following specification:

$$\eta(d_t(z)) = \frac{1}{1 + \exp\{-\gamma(d_t(z) - \varepsilon)\}}, \qquad d_t(z) \quad = |\pi(z) - q_t(z|\mathcal{S}_{t-1})|, \tag{10}$$

with $\gamma, \varepsilon \in (0, +\infty)$. In the experiments we will consider two special cases of this rule. The first one is for $\gamma = 1$ and $\varepsilon = 0$ (random updating) and the second one is for $\gamma \to +\infty$ and $\varepsilon \in (0, +\infty)$ (deterministic updating). In tghe deterministic updating of the set of support point, the function η takes value 0, if $d_t(z) > \varepsilon$ and 1 if $d_t(z) \leq \varepsilon$. Through the threshold parameter ε it is possible to control the number of support points. The parameter can be updated over the iterations following a deterministic rule to control the computational cost of the algorithm. We left this issue as a matter of future research.

We investigate also an alternative specification of η and d_t, which allows for recycling some of the outputs of the Metropolis steps of the Algorithm 1. From this perspective a natural choice could be

$$\eta(d_t(z)) = d_t(z)^{\beta}, \qquad d_t(z) \quad = 1 - \frac{\min\{\pi(z), q_t(z|\mathcal{S}_{t-1})\}}{\max\{\pi(z), q_t(z|\mathcal{S}_{t-1})\}}, \tag{11}$$

with $\beta \in (0, +\infty)$. When $\beta < 1$, the incorporation of new points is facilitated w.r.t. the case $\beta = 1$ whereas, with $\beta > 1$, the growth of $\mathcal{S}_t$ is made more difficult. In our experiments we set $\beta = 1$. Note

that the choice of a linear function for η produces valid weights since $d_t \in [0,1]$ As a final remark, we shall note that this choice of $\eta(d_t(z))$ resembles the probability of adding a new support point in the ARS method. Moreover, if $q_t(z|\mathcal{S}_{t-1}) \geq \pi(z)$, $\forall z \in \mathcal{D}$ and $\forall t$, then $\eta(d_t(z)) = 1 - \frac{\pi(z)}{q_t(z|\mathcal{S}_{t-1})}$, that is exactly the probability of incorporating z to the set of support points in the ARS method. The updating rules presented above for Algorithm 1 require some changes when used in a multiple proposal algorithm such as Algorithm 2. Let us consider the updating scheme in Eq. (11). Let z_i, $i = 1, \ldots, M$ be a set of proposals, then the updating step for $\mathcal{S}_{t-1}$ splits in two parts. First, a z is selected among the proposals, $z_1, \ldots, z_M$, with probability proportional to

$$\varphi_t(z_i) = \max\left\{w(z_i), \frac{1}{w(z_i)}\right\} = \frac{\max\{\pi(z_i), q_t(z_i|\mathcal{S}_{t-1})\}}{\min\{\pi(z_i), q_t(z_i|\mathcal{S}_{t-1})\}}, \tag{12}$$

$i = 1, \ldots, M$. This step selects with high probability a sample at which the proposal value is far from the target. The second step is a control step, where z is included in the set of support points with probability $d_t(z) = 1 - \frac{1}{\varphi_t(z)}$. This step is similar to the accept-reject step in the ARMS algorithm and the probability of the point to be included corresponds exactly to the probability of a proposal to be be accepted in a ARMS algorithm. It can be shown that this two-steps updating procedure corresponds to the following step of our algorithm

$$\mathcal{S}_t = \begin{cases} \mathcal{S}_{t-1} \cup \{z_i\} & \text{with prob. } \eta_i(d_t(z_i)) = \frac{\varphi_t(z_i)-1}{\sum_{j=1}^{M}\varphi_t(z_j)}, \\ \mathcal{S}_{t-1} & \text{with prob. } \frac{M}{\sum_{i=1}^{M}\varphi_i(d_t(z_i))}, \end{cases}$$

where $\varphi_t(z_i) = \frac{1}{1-d_t(z_i)}$ and $d_t(z_i) = 1 - \frac{\min\{\pi(z_i), q_t(z_i|\mathcal{S}_{t-1})\}}{\max\{\pi(z_i), q_t(z_i|\mathcal{S}_{t-1})\}}$.

4.2 Acceleration of the ASMTM

So far, we have presented the general structure of the ASMTM (Section 2) and different procedures for building the proposal distributions and updating their support set when the number of proposals is fixed. In our simulation experiments, we found that the ASMTM is sensibly more robust than the ASM to the specification of the initial set of points. The multiple proposals of the MTM transition allows to reduce the dependence problem to the initial support points and also allows for a faster convergence of the proposal distribution to the target. The superior efficiency of the MTM algorithms over the MH algorithm certainly relies upon the use of multiple proposals which improves the mixing of the transition kernel and the adaptation of the proposal distribution to the target. The price to pay for this gain of efficiency is a higher computational cost. Nevertheless we found that the improvement of the initial set of points benefit of the multiple proposals in a initial transition of the ASMTM chain

and then reduces. Thus, it is possible to design adaptation strategies for the number of tries, which reduce the computational cost. For adapting N_t one can consider a decreasing sequence of number of tries, N_t, $t = 1, 2, \ldots, T$ where T is the number of iterations of the Metropolis chain, $N_t \in 1, \ldots, M$ with M the maximum number of tries. In order to make the tuning phase not too computationally expensive, we suggest to fix M not too large. Following our numerical results, for $N_t = M \ \forall t$, we suggest to set $M = 10$. However, we let the choice and the adaptation of the number of proposals for future research.

5 ASM within Gibbs sampling

Several MCMC techniques need efficient univariate samplers in order to be applied. A well-known example is the Gibbs sampling algorithm, which generates samples from a multivariate distribution by means of sequentially sampling from the full conditionals (Robert and Casella, 2004). Other remarkable examples are hit-and-run methods (Liang et al., 2010, Chapter 3) and adaptive direction sampling (Gilks et al., 1995a, Chapter 6). In this section, we focus our attention on the Gibbs sampler, illustrating the application of the ASM (or ASMTM) within Gibbs sampling. Let $\pi(\mathbf{x})$ be a multivariate target probability density function, with $\mathbf{x} = (x_1, \ldots, x_L)' \in \mathbb{R}^L$. Let t indicate the iteration index of the Gibbs chain and j the current component of $\mathbf{x}$ being generated. Denote with $\pi_j(x_j|\mathbf{x}_{-j,t}) = \pi(x_j|x_{1,t+1}, \ldots, x_{j-1,t+1}, x_{j+1,t}, \ldots, x_{L,t})$ the $j-th$ conditional distribution of x_j given $\mathbf{x}_{-j,t} = (x_{1,t+1}, \ldots, x_{j-1,t+1}, x_{j+1,t}, \ldots, x_{L,t})$. Let $\mathbf{x}_0 = (x_{1,0}, \ldots, x_{L,0})'$ be the initial state of the Gibbs chain at $t = 0$, then at the iteration $t + 1$ the Gibbs sampler is described by the following steps:

1. Draw $x_{j,t+1} \sim \pi_{-j}(x|\mathbf{x}_{-j,t})$ for $j = 1, \ldots, L$.

2. Set $\mathbf{x}_{t+1} = (x_{1,t+1}, \ldots, x_{L,t+1})'$ and $t = t + 1$. Repeat from step 1.

In order to apply the Gibbs sampler we need to be able to draw from all the L full-conditional univariate densities, $\pi_j(x_j|\mathbf{x}_{-j,t})$ for $j = 1, \ldots, L$. Ideally, we would like to be able to sample directly from the L full-conditionals or, at least, to be able to use a rejection sampling (or even better, an adaptive rejection sampler) technique to draw independent samples. However, in general this is not the case and a MCMC technique has to be usually employed within the Gibbs sampler. Moreover, to improve the convergence of the Gibbs, several iterations, say K, of the chain are run and only the last sample is used. This call for the use of efficient MCMC method within the Gibbs sampler.

Algorithm 3. ASM-within-Gibbs

1. Start with $\mathbf{x}_0 = [x_{1,0}, \ldots, x_{L,0}]^T$ and set $t = 0$.
2. Draw $x_{j,t+1} \sim \pi_{-j}(x|\mathbf{x}_{-j,t})$, for $j = 1, \ldots, L$, using the ASM method:
   2.a Initialize the set of support points, $\mathcal{S}_{t,0}^{(j)}$, and the starting value, x_1, of the ASM chain.
   2.b For $k = 1, \ldots, K$:
      i. Build the proposal $q_{t,k}^{(j)}(x|\mathcal{S}_{t,k-1}^{(j)})$ and draw $x' \sim q_{t,k}^{(j)}(x|\mathcal{S}_{t,k-1}^{(j)})$.
      ii. Accept $x_{k+1} = x'$ with probability
      $$\alpha = \min\left[1, \frac{\pi_j(x'|\mathbf{x}_{-j,t}) q_{t,k}^{(j)}(x_k|\mathcal{S}_{t,k-1}^{(j)})}{\pi_j(x_k|\mathbf{x}_{-j,t}) q_{t,k}^{(j)}(x'|\mathcal{S}_{t,k-1}^{(j)})}\right].$$
      Otherwise, set $x_k = x_{k-1}$.
      iii. Update the set $\mathcal{S}_{t,k}^{(j)} = \mathcal{S}_{t,k-1}^{(j)} \cup \{x'\}$, or not (i.e., let $\mathcal{S}_{t,k}^{(j)} = \mathcal{S}_{t,k-1}^{(j)}$), according to a suitable control test (see Section 4.1).
   2.c Set $x_{j,t+1} = x_K$.
3. Set $\mathbf{x}_{t+1} = (x_{1,t+1}, \ldots, x_{L,t+1})'$ and $t = t + 1$. Repeat from step 1.

We propose to use ASM within the Gibbs sampler. The steps of the ASM-within-Gibbs are given in Alg. 3, where $k = 1, \ldots, K$ denotes the iteration index for the ASM algorithm. The efficiency of our ASM and ASMTM algorithm allow for choosing relatively small value of K, say between 10 and 40, in order to achieve good performance of the Gibbs sampler.

Regarding the set of support points, it should be restarted each time the target changes both for ASM and ASMTM, exactly as in the ARMS method. The initial support points are the parameters of the ARMS, ASM and ASMTM algorithms and the choice of these parameters can play a crucial role in the validity of the algorithm and in the behaviour of the MCMC chain. More specifically, the following conditions on the initial set are required (see (Gilks et al., 1997)) for the validity of the ARMS-within-Gibbs algorithm. For all $j = 1, \ldots, L$ and $t = 1, \ldots, T$:

1. Each $\mathcal{S}_{t,0}^{(j)}$ does not contain the current state $x_{j,t}$.
2. Each $\mathcal{S}_{t,0}^{(j)}$ does not depend on the previous set of the same component, i.e., $\mathcal{S}_{t-1,K}^{(j)}$.

These two conditions apply to the ASM-within-Gibbs approach. However, for instance, initializing with the same set of initial support points $\mathcal{S}_{t,0}^{(j)} = \mathcal{S}_0^{(j)}, \quad \forall t, j$, does not jeopardize the validity of the ASM-within-Gibbs algorithm. Furthermore, according to our simulation experiments reported in the following section, the ARMS is extremely sensitive to the choice of the initial set. Our ASM and

ASMTM algorithms are instead robust with respect this choice. Thus, a naive initialization strategy which use the same initial points at each Gibbs iteration can be used.

6 Simulations

6.1 Gaussian mixtures

We study the ability of different algorithms to simulate from multimodal distributions which are locally not log-concave. More specifically we assume the target distribution is the following mixture of two Gaussian distributions

$$0.5\mathcal{N}(7,1) + 0.5\mathcal{N}(-7,0.1).$$

where $\mathcal{N}(\mu,\sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. This corresponds to an example with multiple modes so separated that ordinary MCMC methods fail to visit one of the modes. We design to set of experiments in order to compare the standard ARMS and the our proposed ASM and ASMTM algorithms, combined with the proposal distributions given in Section 2. In a first set of experiments we study the performance of the algorithms for a given proposal distribution. In the second set of experiments we compare different proposal distributions for a given algorithm. In the two set of experiments we use the same function $\eta(d_t(x))$ given in Eq. (11), for the inclusion of a point in the set of support points. We apply different construction methods for the proposal distribution and indicate with:

- 1: the construction proposed by Gilks et al. (1995b) for the ARMS (see Eq (4)), which is formed by exponential pieces (see Fig. 1).
- 2: the construction with exponential pieces, or straight lines in the log-domain (see Eq. (6) and Fig. 2(a)).
- 3: the construction with uniform pieces (see Eq. (7) and Fig. 2(b)).
- 4: the construction with linear pieces in the density domain (see Eq (8) and Fig. 3).

For testing the performance of the algorithms, we run each algorithm $2,000$ times using the same starting values and the same initial set of support points, i.e. $\mathcal{S}_0 = \{-10,-8,5,10\}$. For each run we iterate $T = 5,000$ times the Metropolis chains. Thus, the results given in Tab. 1 are averages over $2,000$ experiments and refer to $T = 5,000$ iterations without removing of the initial burn-in sample.

Within each class of algorithms, ARMS, ASM and ASMTM, the proposal distributions 1 and 2 have higher mean square errors (MSE) and autocorrelation (ACF) (see also MSE and ACF

panels in Fig. 4) with respect to the proposal distributions 3 and 4. The high value of the MSE is due to the difficulty of the Metropolis chain to explore the two modes of the mixture of distributions. Given the initial support points, the proposal distributions 1 and 2 have low density regions between these points with respect constructions 3 and 4. The proposal 3 is the one, among the fours, that has the highest density value between the support points. This feature favours the exploration of the space and the addition of new points to the support set. Comparing the performance of the first and second proposal distribution we finally remark that the first distribution is overperforming the second one (see MSE panel in Fig. 4) only in the ARMS algorithm. The intuition behind this result is that the first construction approach (see Fig. 1) is specifically designed (see Gilks et al. (1995b)) to generate distributions which stay above the target, while the second adaptation procedure allows for proposal graph which are not necessarily above the graph of the target allowing for more flexibility in the exploration of the space. The better mixing properties of distributions 3 and 4 are evident from the estimated autocorrelation functions (see ACF panel in Fig. 4). The absence of full adaptation of the proposal to the target results from the lower acceptance rate given in panel ACC of Fig. 5. Proposals 3 and 4 do not differ substantially, in terms of MSE (see Table 1 and MSE panel in Fig. 4) and autocorrelation, ACF(k), at the lags $k = 1, 10, 50$ (see ACF panel in Fig. 4), while they exhibit a different number of support points (see NSP panel in Fig. 5). For all algorithms proposal distribution 3 requires a larger set of support points. This is confirmed by the number of support points at the last iteration (see m_T in Table 1). Note that in the calculation of the number of support points for the ARMS construction we include the intersection points of the interpolation lines. Independently on the choice of the proposal adaptation, ASM and ASMTM algorithms overperform, in terms of ACF and MSE, the ARMS. Also, for the proposal distributions 1 and 2, the ASM and ASMTM algorithms are able to improve the poor performances of the such proposals combined within a ARMS algorithm. Moreover, increasing the number of ASMTM proposals, from $N = 10$ to $N = 50$, one obtains a further improvement of the MSE and ACF with a increase of the computational cost of the 167%. Note that the number of iterations of the ARMS is slightly higher than the number of iterations of the ASM and ASMTM. This is due to some rejected samples in the accept/reject step of the ARMS.

The best performance, in terms of MSE, is achieved by the ASMTM with $N = 50$ proposals, irrespectively on the choice of the proposal distribution. The lowest MSE is achieved by the ASMTM with proposal distribution 3 and 4. It is the most efficient, with a low autocorrelation level, and the one with the highest number of points in the support set (see number of points at the last iteration, m_T). However, in this example and for the ASMTM with $N = 10$, the large number of support points of constructions 3 and 4 is not affecting the computing time, which is substantially equivalent

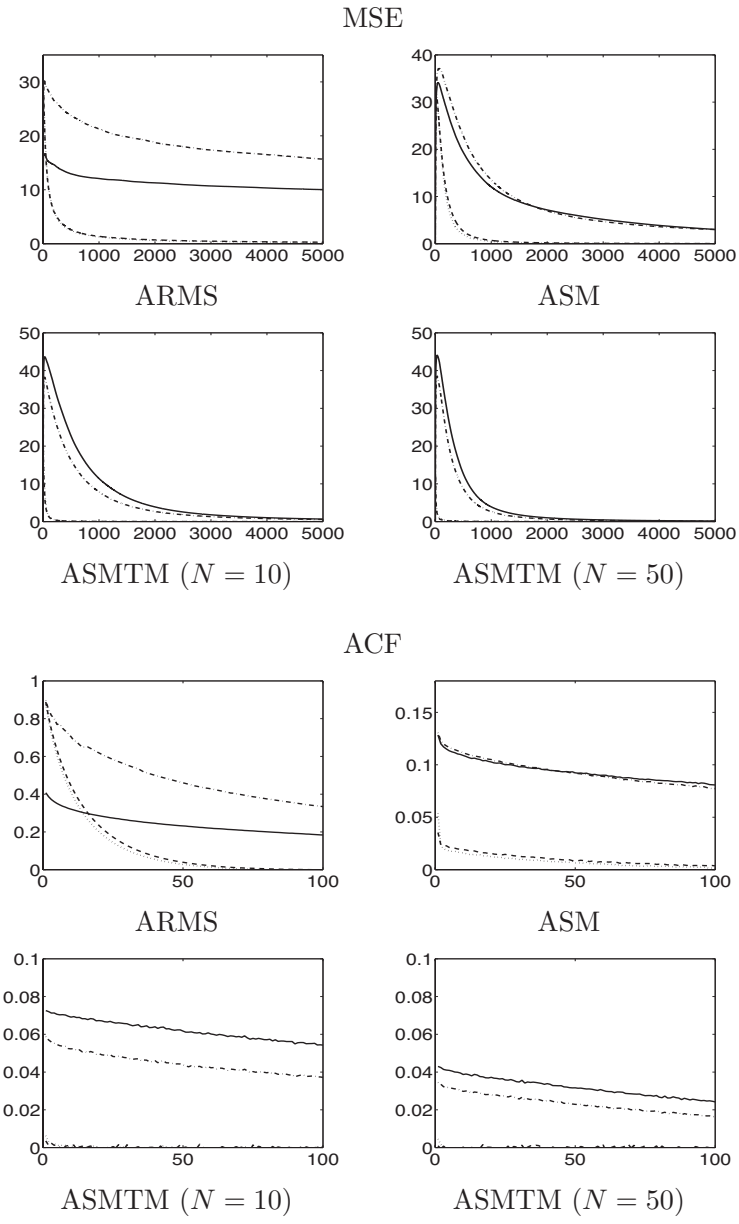


Figure 4: Mean square error (MSE) over the Metropolis chain iterations and Autocorrelation Function (ACF) at lags from 1 to 100. In each plot: construction 1 (solid line), construction 2 (dashed-dotted line), construction 3 (dotted line) and construction 4 (dashed line).

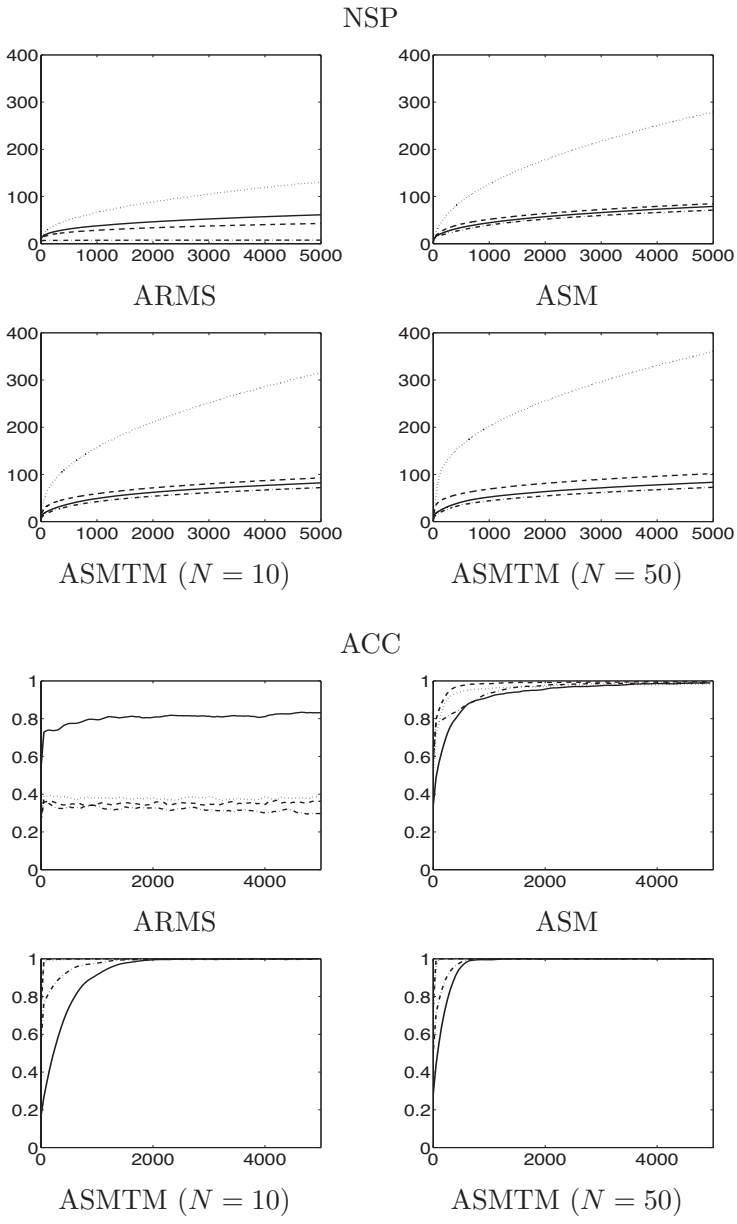


Figure 5: Number of support points (NSP) and acceptance rate (ACC) over the Metropolis chain iterations. In each plot: construction 1 (solid line), construction 2 (dashed-dotted line), construction 3 (dotted line) and construction 4 (dashed line).

to the one of the ASMTM with proposal distributions 1 and 2, while improving the adaptation of the proposal distribution to the target. The full adaptation of the ASMTM with proposal 3 and 4 is clear from the estimated acceptance rate given in panel ACC of Fig. 5. The rate converges to one after a few iterations. From our experiments the ASMTM with $N = 50$ proposals is quite efficient but usually has a higher computational cost. Adding more points to the support set increases the adaptation of the proposal to the target, thus improving the acceptance rate but also implies an increase in the computational cost for constructing the proposal distribution. In order to reduce the computing time, without loosing in efficiency, one can use the acceleration mechanisms described in Section 4 or the procedure for the inclusion of support points to reduce the number of points and the time required by the construction of the proposal distribution.

In this section, we study the effects on time and efficiency of the procedure for the inclusion of the proposal in the set of support points. We show how the parameter of the test to update the support set can be used to control the trade-off between computing time and proposal distribution efficiency. For the sake of brevity, we report the results of such a simulation study, only for the ASM algorithm using the four proposal construction methods described above in this section. We compare the random test procedure given in Eq. (11) with the deterministic test procedure given as a limiting case of Eq. (10) when $\gamma \to +\infty$. The random test procedures has no parameter to tune, while the deterministic test requires the setting of the parameter ε. This parameter allows for controlling the adaptation level and the efficiency of the proposal. The comparison is done in terms of number of support points, acceptance rate, mean square error and autocorrelation function.

In all construction methods, the deterministic test to update $\mathcal{S}_{t-1}$ is more parsimonious, in terms of number of support points, than the random test procedure (see Fig. 10 in Appendix A). The proposal construction method number 4 is the most efficient within the four methods. The efficiency can be evaluated as follows. For all values of ε, at the 5,000 iteration, the MSE and the ACF are both close to zero, while the number of support points in the deterministic test case is smaller than those in the random test case (see Fig. 10 in Appendix A). This means that the same statistical efficiency of the adaptive proposal distribution case can be achieved at a smaller computational effort with a deterministic test procedure. In Fig. 6, we provide an estimate of the relationship between number of support points (NSP in the left chart), acceptance rates (ACC in the right chart) and the parameter ε of the deterministic test for the inclusion of new points in the support set. Both the NSP and the ACC are evaluated at the 5,000-th iteration of the ASM, assuming alternatively proposal construction methods from 1 to 4. We find that the deterministic test (curved lines) has lower NSP with respect to the random test (horizontal lines). Note that the horizontal lines correspond to the values given in

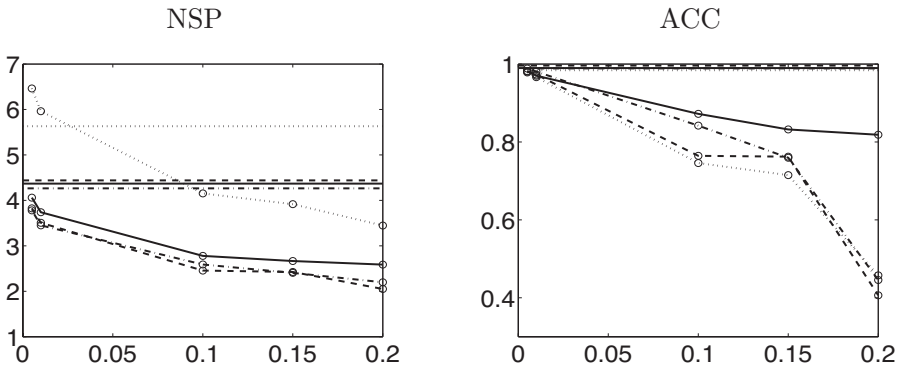


Figure 6: The logarithm of the number of support points (NSP) and the acceptance rate (ACC), at the 5,000-th iteration of the ASM algorithm, for different values of the deterministic test parameter ε and for construction 1 (solid line), construction 2 (dashed-dotted line), construction 3 (dotted line) and construction 4 (dashed line). Horizontal lines indicate the result of the ASM with a random test for inclusion of new points.

the ASM panels of Fig. 5. The deterministic test may lead to a partial adaptation of the proposal (see the acceptance rates below one) when compared to the the random test, but it allows to achieve the same level of autocorrelation and thus has the same efficiency of the random test. Moreover, increasing ε, from 0.005 to 0.2, the NSP and the ACC decreases exponentially fast. The lower bound for the NSP is $\log(4)$ and corresponds to the case of no updates of the initial set of support points ($\mathcal{S}_0 = \{-10, -8, 5, 10\}$). For all values of ε the ranking of the algorithms does not change. The results bring us to conclude that, in the ASM implementation, the construction method 3 is the less efficient in terms of number of support points while the construction method 4 is the most efficient.

6.2 Generalized Gaussian mixtures

In order to corroborate our simulation results, we compare the algorithms on a mixture model with well separated modes and with heavy tail components. More specifically we consider the following mixtures of generalized exponential power (GEP) distributions:

1) mixture of heavy- and normal-tail symmetric distributions (Mix1)

$$0.6\,\mathcal{GEP}(0, 1, 1/2, 1) + 0.4\,\mathcal{GEP}(50, 1, 2, 1),$$

2) mixture of heavy- and normal-tail asymmetric distributions (Mix2(κ)), $\kappa = 0.01, 0.1, 0.4$,

$$0.4\,\mathcal{GEP}(0, 1, 1/2, 2) + 0.6\,\mathcal{GEP}(50, 1, 1/2, \kappa),$$

where $\mathcal{GEP}(\mu, \sigma^2, \alpha, \kappa)$ denotes a GEP distribution with location, scale, shape and asymmetry parameters μ, σ, α and κ, respectively. The density of the GEP distribution is

$$\pi(x) = \frac{\alpha}{\sigma\Gamma(1/\alpha)} \frac{\kappa}{1+\kappa^2} \exp\left\{ -\frac{\kappa^\alpha}{\sigma^\alpha} \left((x-\mu)^+\right)^\alpha - \frac{1}{\sigma^\alpha \kappa^\alpha} \left((x-\mu)^-\right)^\alpha \right\}, \tag{13}$$

where $\Gamma(z)$ is the complete gamma function, x^+ is equal to x if $x \geq 0$ and 0 otherwise and x^+ is equal to $-x$ if $x \leq 0$ and 0 otherwise. The shape parameter α controls the tails of the density function and determines if it is flat or peaked. The parameter κ is an inverse scale factor (see Fernandez et al. (1995) and Fernandez and Steel (1998)) which controls the asymmetry of the distribution. When $\alpha = 2$ (and $\kappa = 1$) we have the (symmetric) Gaussian distribution, when $\alpha = 1$ (and $\kappa = 1$) we have the (symmetric) Laplacian or double exponential distribution and when $\alpha \to 1$ then we have the uniform distribution. Finally smaller value of α correspond to heavy tailed distribution and when $\alpha \to 0$ we have the Dirac mass centred at μ. The GEP distribution has the exponential power (EP) distribution as special case for $\kappa = 1$. The EP is also known as the generalized normal or generalized error distribution popularized by Box and Tiao (1964). It has been used successfully in many fields (see Kotz et al. (2001) for a review) thanks to its shape flexibility which allows for modelling deviations from the Gaussian distribution. In the recent years, mixtures of EP distributions have received increasing attention (e.g., see Elguebaly and Bouguila (2012)) as flexible models for robust data clustering.

We report the results of the ARMS-1 (Gilks et al. (1995b)), ASM-4 with random test and ASMTM-4 with random test and $N = 10$. We generate 5,000 draws from each algorithm and compute, without removing the burn-in sample, the mean, the autocorrelation function, the number of support points at the last iteration and the computing time. As in the previous section, in order to have an accurate algorithm comparison, each quantity is the result of an average over 2,000 independent runs of each algorithm. We start all algorithms with the same initial value and set of support points. We study the sensitivity of the algorithm performances to the choice of the initial support points and run three sets of experiments with $\mathcal{S}_0 = \{-1, 1, 20\}$, $\mathcal{S}_0 = \{-1, 1, 70\}$ and $\mathcal{S}_0$ with three random points drawn independently from the uniform distribution $\mathcal{U}([-70, 70])$. For comparison purposes we also report the results of the slice sampling algorithm (see Neal (2003)) implemented in MATLAB (see the statistical toolbox documentation of The-MathWorks (2013)). The results of these experiments are given in panel I of Tab. 2. As a reference for comparing the performances of the algorithms we shall recall that the true mean of the mixture given above is 20. In panel (I.a) one can see that the performance of the algorithms, in terms of estimated mean, is similar. Nevertheless, the level of autocorrelation of the ARMS-1 is higher than the one of the ASM and ASMTM. Note that ASM-4 and ASMTM-4 ($N = 10$) is more efficient and more time consuming than the ASM-4. Note however that acceleration techniques (see Section 4.2) can be applied to reduce the computational cost of the MTM transition. The best efficiency of the ASMTM-4 also affects the standard deviation (column SD) of the mean estimates. The SD is about 6 for the slice, 10 for the ARMS-1, 3 for the ASM-4 and 0.5 for the ASMTM-4. The results also show the superiority of the ASMTM-4 to the slice sampling (in the

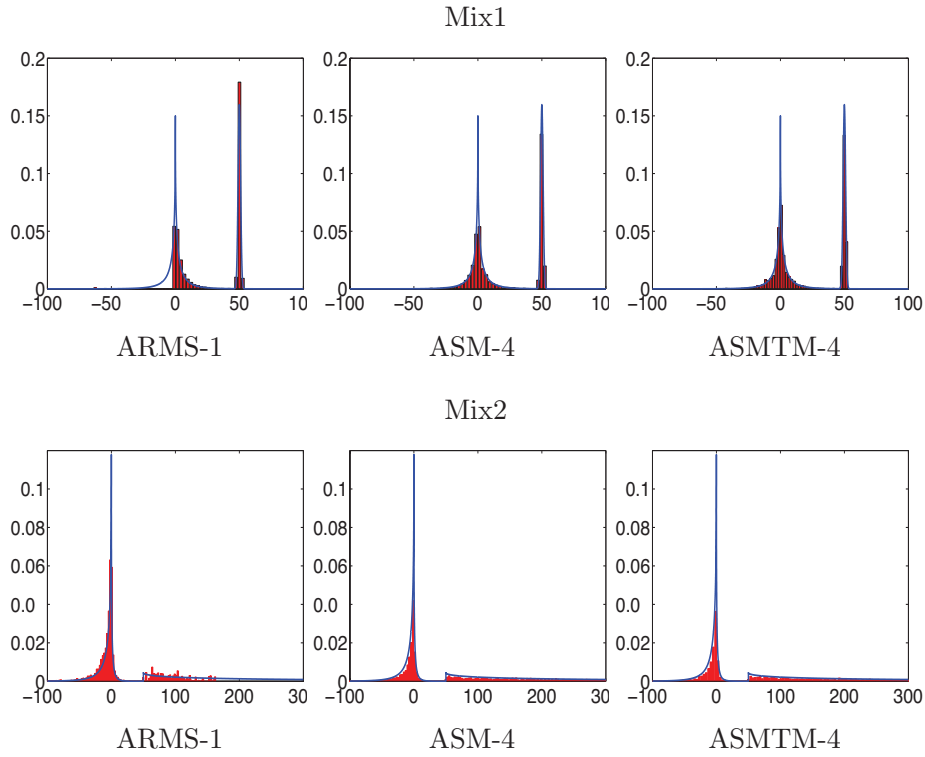


Figure 7: Density of the Mix1 and Mix2 mixtures of exponential power distributions (solid line) and histograms of 5,000 samples from the Mix1 distribution generated with ARMS-1, ASM-4 and ASMTM-4 ($N = 10$) assuming $\mathcal{S}_0 = \{-1, 1, 20\}$.

implementation available from the Matlab toolbox). From panel (I.b) one can see that the ARMS of Gilks et al. (1995b) is sensitive to the choice of the initial set of support points. A bad choice of the points lead to bad estimates of the mean and to higher ACF values. Panel II of Tab. 2 show the results for Mix2 for different values of κ (panel II.a-c) given the same initial set of support points. The results confirm the inferior mixing of the ARMS chain and the lack of convergence of the slice. Fig. 7 exhibits the densities of the Mix1 and Mix2 distributions and their histogram approximations generated in one of the experiments by the ARMS, ASM and ASMTM algorithms. A graphical inspection reveals that the ARMS-1 has difficulties in exploring the tails of the target in the case of a bad choice of the initial set of points. The randomization of the initial support points across the 2,000 experiments allow us to account in the algorithm comparison for the problem of the sensitivity to the initial set. The results in panel (I.c) are averages over experiment outcomes with different initial sets of three random support points. They show that the ASM and AMTM are more efficient than the ARMS and the slice sampler.

6.3 Makeham's and Gompertz's distributions

We consider an example where simulation from the target is challenging due to the potential absence of log-concavity, the presence of skewness and heavy tails in the density. We apply our simulation

algorithms to one of the most known distribution in actuarial mathematics, that is the Makeham's distribution, which is used for modelling the future lifetime of individuals (see Bowers et al. (1986)). In many applications to life insurance, the analytical calculation of the expected value of transform of the Makeham's random variable is difficult and numerical integration techniques are applied. The numerical computation can be even more burdensome for higher moments or tail probabilities. This issues call for the use of Monte Carlo simulation techniques.

Let X be the random age at death for a new born life and $T(x)$ the future lifetime of an individual with life age x. Then the survival function of the individual is

$$P(T(x) > z) = \frac{P(X > x + z)}{P(X > z)}.$$

It can be shown (see Bowers et al. (1986)) that under the Makeham's mortality law the density of the future life time $T(x)$ is

$$\pi(z) = \exp\left(-Az - \frac{BC^x}{\log(C)}(C^z - 1)\right)(A + BC^{x+z})\mathbb{I}_{[0,+\infty)}(z), \tag{14}$$

with parameters $A > -B$, $B > 0$, $C \geq 1$ and $x \geq 0$. This distribution has another well known distribution, i.e. the Gompertz's distribution, as a special case for $A = 0$. Note that while the Gompertz's distribution is log-concave the Makehm's one may be not log-concave. If $A \leq 0$ then it is log-concave, if $A > 0$, which is the case in many actuarial applications, then the Makeham's distribution is not log-concave. A simulation algorithm based on the the Gilks et al. (1995b) ARMS has been proposed in Scollnik (1995). In this paper some example of pricing of the life contingent functions defining annuities or insurances are considered. We compare the ARMS with our ASM and AMTM algorithms on three pricing examples (see Scollnik (1995)), that are: the expected future life time, a life insurance with benefit payable at the moment of death and a continuous whole life annuity. In the first example we approximate the distribution of the residual life time of an individual with age $x = 50$, $T(50)$, assuming the following parameter setting of the Makeham's distribution $A = 0.001, B = 0.0000070848535, C = 1.1194379$. In the second and third example we approximate the distribution of the random present value of the benefit $Z = \exp\{-\delta T(50)\}$ and of the annuities

$$Y = \int_0^{T(50)} \exp\{-\delta s\} ds,$$

respectively. In the comparison we set the interest rate $\delta = \log(1.025)$ and consider ARMS-1, ASM-4 with random test, ASMTM-4 with random test and $N = 10$. We generate 5,000 draws from each

Metropolis algorithms and compute, without removing the burn-in sample, mean, standard deviation, skewness, kurtosis and 95% quantile of the distribution of interest. In order to have an accurate algorithm comparison, each quantity is the result of an average over 2,000 independent runs of each Metropolis algorithm. We shall notice that the estimates given in Scollnik (1995) are based on 250,000 iterations of the ARMS-1. The choose instead 5,000 iterations to show the higher efficiency of our ASM-4 and ASMTM-4 algorithms with respect to the ARMS-1. We start all algorithms with the same initial value and set of support points. We run two set of experiments with two different initial sets, that are $\mathcal{S}_0 = \{20, 40, 60\}$ and $\mathcal{S}_0 = \{0, 20, 40, 60\}$, with the aim to compare the Gilks et al. (1995b) ARMS-1 algorithm and our ASM and ASMTM algorithm also in terms of sensitivity to the initial set of points.

The results of the first set are given in panel (a) of Tab. 3 while the results of the second set are summarized in panel (b) of the same table. As a reference for comparing the performance of the algorithms we consider the results of a deterministic integration algorithm reported in Scollnik (1995). The panel (a) shows the difficulty of the ARMS-1 algorithm to provide good estimate of the quantities of interest. The positive skewness and large kurtosis values for the random variable $T(50)$ indicate that the ARMS-1 algorithm is not approximating well the tails of the distribution of $T(50)$. As an example we exhibit in Fig. 11 the histograms generated by each algorithm in one of the 2,000 experiments for the three variables $T(50)$, Z and Y. One can see in the upper-left chart of the figure that the ARMS-1 is not generating value in the left tail of the distribution. A similar problem occurs for the right tail in the mid-left and bottom-left chart of the same figure. Last three lines of panel (a) indicates that the ARMS-1 has a higher autocorrelation at the lags 1, 10 and 50 than the ASM-4 and ASMTM-4. This result confirms the better mixing of our adaptive algorithms.

In the second set of experiments (see panel (b) of Tab. 3) the initial set $\mathcal{S}_0 = \{0, 20, 40, 60\}$ includes the left bound of the support of the target distribution, which is defined on the $[0, +\infty)$ interval. In these experiments, the ARMS-1 gives better results than in the first set. The efficiency is comparable to the one of the ASM-4 and ARMS-4. These results confirm the dependence problem of the ARMS-1 on the initial set of support points. We conclude that, in our experiments, our ASM-4 and ASMTM-4 algorithms overperform the ARMS-1 irrespectively on the initial set of support points, which confirms the superior mixing of these algorithms, already proved in the experiments of the previous section.

6.4 Stochastic volatility

The accept/reject Metropolis algorithm is usually employed within a Gibbs algorithm as a general simulation method for full conditional distributions which are not easy to simulate from. A class of models where this usually happens is stochastic volatility (SV) models (e.g., see Jacquier et al. (1994), Jacquier et al. (2004) and Geweke (1994)). In this example we consider the univariate SV model with leverage due to Jacquier et al. (2004)

$$y_t = \sqrt{h_t}\varepsilon_t, \tag{15}$$

$$\log h_t = \alpha + \delta \log h_{t-1} + \eta_t, \tag{16}$$

with

$$\begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \overset{i.i.d.}{\sim} \mathcal{N}_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \psi \\ \psi & \psi^2 + \omega \end{pmatrix} \right).$$

See Jacquier et al. (2004) for prior specification on the intercept, α, persistence, ϕ, and volatility, ψ and ω, parameters. The full conditional distribution of h_t is known up to a normalizing constant, i.e.

$$p(h_t|h_{t-1}, h_{t+1}) \propto h_t^{\left(-\frac{3}{2} - \frac{\delta\psi y_{t+1}}{\omega \exp\{h_{t+1}/2\}}\right)} \exp\left\{ -\frac{y_t^2}{2h_t}\left(1 + \frac{\psi^2}{\omega}\right)\right\} \cdot \tag{17}$$
$$\cdot \exp\left\{ -(1+\delta^2)\frac{(\log h_t - \mu_t)^2}{2\omega} + \frac{\psi y_t (\log h_t - \alpha - \delta \log h_{t-1})}{\omega\sqrt{h_t}} \right\}.$$

where $\mu_t = (\alpha(1-\delta) + \delta(\log h_{t+1} + \log h_{t-1}))/(1+\delta^2)$. In the experiments we set $y_t = 0.001$, $h_{t+1} = h_{t-1} = \alpha$, $\alpha = -0.356$, $\delta = 0.95$, $\psi = -0.15$ and $\omega = 0.043$, which is in line with the empirical results of Jacquier et al. (2004). We compare our ASM and ASMTM algorithms, with the Gilks et al. (1995b)'s ARMS and one of the MH algorithms used in Jacquier et al. (2004), which is an independent MH with an inverse gamma proposal distribution. Thus the proposal of the MH is $h_t \sim \mathcal{IG}(\phi_t, \theta_t)$, with parameters

$$\phi_t = \frac{\psi\delta y_{t+1}}{\omega\sqrt{h_{t+1}}} - 0.5 + \frac{1 - 2\exp\{\omega/(1+\delta^2)\}}{1 - \exp\{\omega/(1+\delta^2)\}} - 1,$$

and

$$\theta_t = \frac{y_t^2}{2}\left(1 + \frac{\psi^2}{\omega}\right) + \left(\frac{1 - 2\exp\{\omega/(1+\delta^2)\}}{1 - \exp\{\omega/(1+\delta^2)\}} - 1\right) \exp\left(\mu_t + 0.5\omega/(1+\delta^2)\right).$$

The adaptive proposal distributions of our ASM and ASMTM are obtained with the construction method give in Eq (8) (see also Fig. 3). Both ASM and ASMTM use a random test for inclusion of the points in the set and support points. The initial support points are the same across the algorithms and the experiments, that is $\mathcal{S}_0 = \{0.0001, 0.001, 0.005, 1\}$ in the first set of experiments (Tab. 4,

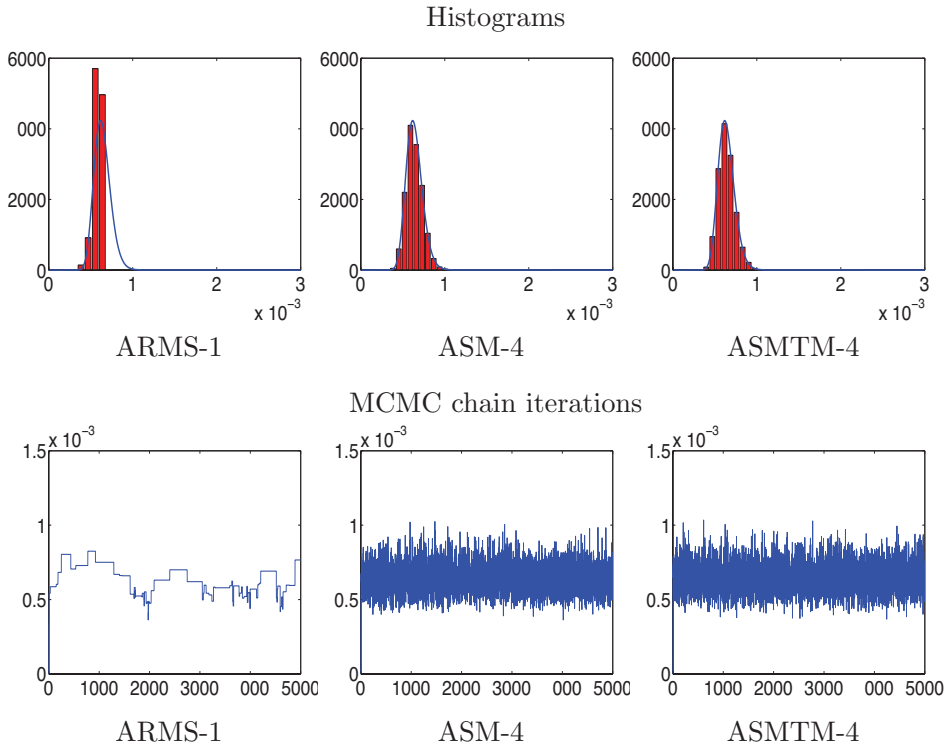


Figure 8: Results of 5,000 samples generated with ARMS-1, ASM-4 and ASMTM-4 ($N = 10$) for the log-volatility full conditional distribution assuming $\mathcal{S}_0 = \{0.0001, 0.0003, 0.005, 1\}$. Top: log-density of the target (solid line) and MCMC empirical log-density (histogram). Bottom: output of the MCMC iterations.

panel (a)), and $\mathcal{S}_0 = \{0.0001, 0.0003, 0.005, 1\}$ in the second set (Tab. 4, panel (b)). The results in Tab. 4 show that our ASM and ASMTM chains exhibit a lower autocorrelation at the first lag with respect to the MH chain. Nevertheless after looking at the mean and at the ACF at the other lags one could conclude in favour of a substantial equivalence of the algorithms in terms of efficiency. Thus, we shall stress that our approach to the design of the Metropolis proposal distribution is for general simulation purposes, since it does not require the intervention of the researcher and provides an efficient automatic adaptation of the proposal to the target. The MH considered here uses instead a proposal which has been specifically designed by the researcher for the SV model (see Jacquier et al. (2004)). Another result, that we found also in the previous examples, is the sensitivity of the ARMS-1 (Gilks et al. (1995b)) to the choice of the initial set of support points. The choice of the initial set can affect negatively the mixing of the ARMS-1 chain and its ability to visit the domain of the distribution (see ACF in the panel (b)). The bad mixing of the ARMS chain is also confirmed by the raw output of the chain iterations (see bottom charts of Fig. 8). The histograms in Fig. 8 also show that the ARMS proposal is not able to generate candidates in the high probability density region and the rejected points are not useful for improving the proposal distribution. The mixing of the ASM and ASMTM

chains is better and, as we found in all our experiments, the ASM and ASMTM algorithms are less sensitive than the ARMS to the choice of the initial support points.

7 Conclusions

We propose new adaptive sticky MTM algorithms (ASMTM) for all-purposes stochastic simulation. Different interpolation strategies for the construction of the adaptive nonparametric distributions are discussed. We have been able to prove the ergodicity of the ASMTM algorithm, thus extending previous results in the literature and using conditions which are automatically satisfied by our proposal distributions. Our simulation experiments show the best efficiency of the proposed ASMTM algorithms over traditional adaptive rejection Metropolis (ARMS). We found that the performance of the ARMS depend crucially on the choice of the initial support points, whereas our ASMTM is robust with respect to this choice. Moreover, the multiple-mode and heavy-tail target examples show that the ASMTM, as opposed to the ARMS, is efficient in exploring the sample space. The simulation experiments show that the proposal construction methods with uniform pieces and the one with linear pieces in the density domain are the most efficient. The role of the control step for the inclusion of new support points has been investigated. We found that this step is quite effective for controlling the computational cost and the efficiency of the ASMTM when a large number of proposals is used.

8 Acknowledgment

This work has been partly financed by the Spanish government, through the DEIPRO project (TEC2009-14504-C02-01), the CONSOLIDER-INGENIO 2010 Program (Project CSD2008-00010). Roberto Casarin's research is supported by the Italian Ministry of Education, University and Research (MIUR), through PRIN 2010-11 grant and by the European Union, Seventh Framework Programme FP7/2007-2013 under grant agreement SYRTO-SSH-2012-320270. Fabrizio Leisen's research is partially supported by grant ECO2011-25706 of the Spanish Ministry of Science and Innovation. The authors would also like to thank Joaquín Míguez (Universidad Carlos III de Madrid) for many useful comments on a preliminary version of this paper.

References

Andrieu, C. and Moulines, E. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, **16** 1462–1505.

Andrieu, C. and Robert, C. (2001). Controlled MCMC for optimal sampling. Tech. Rep. 0125, Cahiers de Mathématiques du Ceremade; Université Paris-Dauphine.

Atchade, Y. and Rosenthal, J. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, **11** 815–828.

Bédard, M., Douc, R. and Moulines, E. (2012). Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, **122** 758–786.

Bowers, N., Gerber, H., Hickman, J., Jones, D. and Nesbitt, C. (1986). *Acturial Mathematics*. Itasca, Ill: Society of Actuaries.

Box, G. E. P. and Tiao, G. C. (1964). A note on criterion robustness and inference robustness. *Biometrika*, **51** 169–173.

Cai, B., Meyer, R. and Perron, F. (2008). Metropolis-Hastings algorithms with adaptive proposals. *Statistics and Computing*, **18** 421–433.

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

Craiu, R., Rosenthal, J. and Yang, C. (2009). Learn from thy neighbor: Parallel-chain adaptive MCMC. *Journal of the American Statistical Association*, **488** 1454–1466.

Craiu, R. V. and Lemieux, C. (2007). Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing*, **17** 109–120.

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer.

Elguebaly, T. and Bouguila, N. (2012). Generalized Gaussian mixture models as a nonparametric Bayesian approach for clustering using class-specific visual features. *J. Vis. Commun. Immage R.*, **23** 1199–1212.

Fernandez, C., Osiewalski, J. and Steel, M. F. J. (1995). Modelling and inference with v-distributions. *Journal of the American Statistical Association*, **90** 1331–1340.

Fernandez, C. and Steel, M. F. J. (1998). On Bayesian modelling of fat tails and skewness. *Journal of the American Statistical Association*, **93** 359371.

Geweke, J. (1994). Comment on Bayesian analysis of stochastic volatility. *Journal of Business and Economics Statistics*, **12 (4)** 371–417.

Gilks, W., Richardson, S. and Spiegelhalter, D. (1995a). *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*. Taylor & Francis, Inc., UK.

Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995b). Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics*, **44** 455–472.

Gilks, W. R., Neal, R., Best, N. G. and Tan, K. K. C. (1997). Corrigidum: Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics*, **46** 541–542.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, **41** 337–348.

Giordani, P. and Kohn, R. (2010). Adaptive independent Metropolis-Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics*, **19** 243–259.

Haario, H., Saksman, E. and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, **7** 223–242.

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

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

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

Jacquier, E., Polson, N. G. and Rossi, P. E. (2004). Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. *Journal of Econometrics*, **122** 185–212.

Jasra, A., Stephens, D. and Holmes, C. (2007). On population-based simulation for static inference. *Statistics and Computing*, **17** 263–279.

Kotz, S., Kozubowoski, T. J. and Podgórski, K. (2001). *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering and Finance*. Birkhöuser.

Krzykowski, G. and Mackowiak, W. (2006). Metropolis Hastings simulation method with spline proposal kernel. An Isaac Newton Institute Workshop, URL `http://www.newton.ac.uk/programmes/SCB/Poster1/mackowiak.html`.

Latuszynski, K., Roberts, G. and Rosenthal, J. (2013). Adaptive Gibbs samplers and related MCMC methods. *Annals of Applied Probability*, **23** 66–98.

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

Liu, J. S. (2004). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag.

Liu, J. S., Liang, F. and Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, **95** 121–134.

Martino, L. and Read, J. (2012). A multi-point Metropolis scheme with generic weight functions. *Statistics and Probability Letters*, **82** 1445–1453.

Martino, L. and Read, J. (2013). On the flexibility of the design of multiple try Metropolis schemes. *Computational Statistics, forthcoming*.

Martino, L., Read, J. and Luengo, D. (2012). Improved adaptive rejection Metropolis sampling algorithms. *arXiv:1205.5494v4*.

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

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

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, **31** 705–767.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, **44** 458–475.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, **18** 349–367.

Saksman, E. and Vihola, M. (2010). On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. *Annals of Applied Probability*, **20** 2178–2203.

Scollnik, D. P. (1995). Simulating random variables from Makeham's distribution and from others with exact or nearly log-concave densities. *Transactionso of Society of Actuaries*, **47** 409–454.

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

So, M. (2006). Bayesian analysis of nonlinear and non-Gaussian state space models via multiple-try sampling methods. *Statistics and Computing*, **16** 125–141.

The-MathWorks (2013). *MATLAB - The Language of Technical Computing, Version R2013a*. The MathWorks, Inc., Natick, Massachusetts.

A Proofs

Proof of Theorem 1: Let ρ be the state appended to the history $\mathcal{S}_{t-1}$. Without loss of generality, suppose that $\eta_j = 1$ where j is the index sampled at the selection step, then $\mathcal{S}_t = \rho \cup \mathcal{S}_{t-1}$ with $\rho = z_j$. Moreover, let $f_t(\mathcal{S}_t)$ be the joint distribution of the history $\mathcal{S}_t$ and let $q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) = \prod_{i\neq j} q_t(x'_i|\mathcal{S}_{t-1})$ where $\boldsymbol{x}'_{-j} = (x'_1, \ldots, x^{'}_{j-1}, x^{'}_{j+1}, \ldots, x_M)$. Following Liu et al. (2000), Theorem 1 and Casarin et al. (2013) Theorem 1, the actual transition probability of the MTM step in our ASMTM writes as follows:

$$
\begin{aligned}
A(\rho, x_{t+1}) &= \int_{\mathcal{J}} h_{Mt}(dJ) \int_{\mathcal{X}^M} q_{t,-J}(\boldsymbol{x}'_{-J}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-J} q_t(x'_J|\mathcal{S}_{t-1}) \\
&\int_{\mathcal{X}^{M+1}} \delta_{\rho}(dx^*_J)\delta_{x_{t+1}}(dx'_J) \prod_{k\neq J} \delta_{x'_k}(dx^*_k) \min\left[1, \frac{\sum_{i\neq j} w_t(x'_i) + w_t(x'_J)}{\sum_{i\neq j} w_t(x^*_i) + w_t(x^*_J)}\right] \\
&= \sum_{j=1}^{M} \int_{\mathcal{X}^{M-1}} q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-j} \alpha(\rho, x_{t+1}, \boldsymbol{x}'_{-j}, \mathcal{S}_{t-1}) \frac{w_t(x_{t+1}) q_t(x_{t+1}|\mathcal{S}_{t-1})}{\sum_{k\neq j} w_t(x'_k) + w_t(x_{t+1})},
\end{aligned}
$$

where $\mathcal{J} = \{1, \ldots, M\}$ and $h_{Mt}(dJ) = \sum_{j=1}^{M} \frac{w_t(x'_j)}{\sum_{k=1}^{M} w_t(x'_k)} \delta_j(dJ)$ is the empirical measure generated by the selection step. We show that the chain with this transition probability never leaves the stationary π distribution once it is reached:

$$
\begin{aligned}
p_{t+1}(x_{t+1}|\mathcal{S}_t) f_t(\mathcal{S}_t) = f_{t-1}(\mathcal{S}_{t-1}) \sum_{j=1}^{M} \Bigg\{ &\int_{\mathcal{X}^{M-1}} \pi(\rho) q_t(x_{t+1}|\mathcal{S}_{t-1}) \frac{w_t(x_{t+1})}{\sum_{i\neq j} w_t(x'_i) + w_t(x_{t+1})} \times \\
&\times \alpha(\rho, x_{t+1}, \boldsymbol{x}'_{-j}, \mathcal{S}_{t-1}) q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-j} + \int_{\mathcal{X}^{M-1}} \pi(x_{t+1}) q_t(\rho|\mathcal{S}_{t-1}) \frac{w_t(\rho)}{\sum_{i\neq j} w_t(x'_i) + w_t(\rho)} \times \\
&\times [1 - \alpha(x_{t+1}, \rho, \boldsymbol{x}'_{-j}, \mathcal{S}_{t-1})] q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-j} \Bigg\}
\end{aligned}
$$

$$= f_{t-1}(\mathcal{S}_{t-1}) \sum_{j=1}^{M} \left\{ \int_{\mathcal{X}^{M-1}} \pi(x_{t+1}) q_t(\rho|\mathcal{S}_{t-1}) \frac{w_t(\rho)}{\sum_{i\neq j} w_t(x'_i) + w_t(\rho)} \times q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-j} \right\}$$

$$= \pi(x_{t+1}) f_{t-1}(\mathcal{S}_{t-1}) q_t(\rho|\mathcal{S}_{t-1}) g_t(\rho|\mathcal{S}_{t-1}),$$

where

$$g_t(\rho|\mathcal{S}_{t-1}) = M \int_{\mathcal{X}^{M-1}} \frac{w_t(\rho)}{\sum_{i\neq j} w_t(x'_i) + w_t(\rho)} q_{t,-j}(\boldsymbol{x}'_{-j}|\mathcal{S}_{t-1}) d\boldsymbol{x}'_{-j},$$

and this concludes the proof. □

Proof of Theorem 2: Let x_t be the current value of the chain at the iteration t and x' the j-th proposal accepted if $u_t < \alpha_j(x_t, x'_j, \mathbf{x}'_{-j}, \mathcal{S}_{t-1})$, where u_t is a uniform number on the $[0,1]$ interval. The acceptance probability $\alpha_j(x_t, x'_j, \mathbf{x}'_{-j}, \mathcal{S}_{t-1})$ satisfies

$$\min\left\{1, \frac{\sum_{k\neq j} \frac{\pi(x'_k)}{q_t(x'_k|\mathcal{S}_{t-1})} + \frac{\pi(x'_j)}{q_t(x'_j|\mathcal{S}_{t-1})}}{\sum_{k\neq j} \frac{\pi(x'_k)}{q_t(x'_k|\mathcal{S}_{t-1})} + \frac{\pi(x_t)}{q_t(x_t|\mathcal{S}_{t-1})}}\right\} > \min\left\{1, \frac{a_t(\mathcal{S}_{t-1})}{M}\left(\sum_{k\neq j} \frac{\pi(x'_k)}{q_t(x'_k|\mathcal{S}_{t-1})} + \frac{\pi(x'_j)}{q_t(x'_j|\mathcal{S}_{t-1})}\right)\right\}$$

$$= \min\left\{1, \frac{\pi(x'_j)}{q_t(x'_j|\mathcal{S}_{t-1})} \tilde{a}_t(\mathcal{S}_{t-1}, \mathbf{x}')\right\},$$

where

$$\tilde{a}_t(\mathcal{S}_{t-1}, \mathbf{x}') = \frac{a_t(\mathcal{S}_{t-1})}{M} \frac{\sum_{k\neq j} \frac{\pi(x'_k)}{q_t(x'_k|\mathcal{S}_{t-1})} + \frac{\pi(x'_j)}{q_t(x'_j|\mathcal{S}_{t-1})}}{\frac{\pi(x'_j)}{q_t(x'_j|\mathcal{S}_{t-1})}}.$$

Then let A_t be the condition that $u_t q_t(x'|\mathcal{S}_{t-1})/\pi(x') \leq \tilde{a}_t(\mathcal{S}_{t-1}, \mathbf{x}')$. Then the conditional distribution of x' given $\mathcal{S}_{t-1}$, x_t and A_t is proportional to

$$\sum_{j=1}^{M} \int_{\mathcal{X}^{M-1}} \frac{w_t(x')}{\sum_{i\neq j} w_t(x'_i) + w_t(x')} P(A_t|\mathcal{S}_{t-1}, x', \mathbf{x}_{-j}, x_t) q_{t,-j}(\mathbf{x}'_{-j}|\mathcal{S}_{t-1}) d\mathbf{x}'_{-j} q_t(x'|\mathcal{S}_{t-1}) =$$

$$= \sum_{j=1}^{M} \int_{\mathcal{X}^{M-1}} \frac{w_t(x')}{\sum_{i\neq j} w_t(x'_i) + w_t(x')} P\left(u_t \leq \frac{\pi(x')}{q_t(x'|\mathcal{S}_{t-1})} \tilde{a}_t(\mathcal{S}_{t-1}, \mathbf{x}')\right) \times q_t(x'|\mathcal{S}_{t-1}) q_{t,-j}(\mathbf{x}'_{-j}|\mathcal{S}_{t-1}) d\mathbf{x}'_{-j}$$

$$= \sum_{j=1}^{M} \int_{\mathcal{X}^{M-1}} \frac{a_t(\mathcal{S}_{t-1})}{M} \pi(x') q_{t,-j}(\mathbf{x}'_{-j}|\mathcal{S}_{t-1}) d\mathbf{x}'_{-j} = a_t(\mathcal{S}_{t-1}) \pi(x').$$

Following Holden et al. (2009) we define

$$I_{t+1} = \begin{cases} 0 & \text{with probability } 1 - a_{t+1}(\mathcal{S}_t) \text{ if } I_t = 0, \\ 1 & \text{otherwise,} \end{cases}$$

for $t \geq 1$, with $I_0 = 0$, and the probability not to be in the stationary after j step is $P(I_t = 0|\mathcal{S}_{t-1}) = b_t(\mathcal{S}_{t-1})$ where $b_t(\mathcal{S}_{t-1}) = \prod_{j=1}^{t}(1 - a_j(\mathcal{S}_{j-1}))$. Then conditional distribution of x_{t+1} can be written

as $p_t(x|\mathcal{S}_t) = \pi(x)(1 - b_t(\mathcal{S}_{t-1})) + v_t(x|\mathcal{S}_t)b_t(\mathcal{S}_t)$, where v_t is a probability distribution. Then the total variation distance between the limiting distribution and the marginal distribution of x_{t+1}, that is

$$||p_t - \pi||_{TV} = \int_{\mathcal{X}} |p_t(x) - \pi(x)| \, d\mu(x)$$

can be bounded as follows

$$\begin{aligned}
||p_t - \pi||_{TV} &= \int_{\mathcal{X}} \left| \int_{\mathcal{X}^t} p_t(x|\mathcal{S}_t) p_t(\mathcal{S}_t) d\mu(\mathcal{S}_t) - \pi(x) \right| d\mu(x) \\
&= \int_{\mathcal{X}} \left| \int_{\mathcal{X}^t} (v_t(x|\mathcal{S}_t) - \pi(x)) b_t(\mathcal{S}_{t-1}) p_t(\mathcal{S}_t) d\mu(\mathcal{S}_t) \right| d\mu(x) \\
&\leq \int_{\mathcal{X}^t} \int_{\mathcal{X}} |v_t(x|\mathcal{S}_t) - \pi(x)| \, d\mu(x) b_t(\mathcal{S}_{t-1}) p_t(\mathcal{S}_t) d\mu(\mathcal{S}_t) \leq 2 \int_{\mathcal{X}^t} b_t(\mathcal{S}_{t-1}) p_t(\mathcal{S}_t) d\mu(\mathcal{S}_t).
\end{aligned}$$

Thanks to this bound, the probability to jump in the stationary within t steps, can be made arbitrarily close to one. □

Proof of Theorem 3: Let us consider a set of support points, $\mathcal{S}_{t-1} = \{s_1, \ldots, s_{m_{t-1}}\}$, with $s_1 < \ldots < s_{m_{t-1}}$, at time step t. Note that, by using any of the procedures described in Section 3, the corresponding proposal density function, $\tilde{q}_t(x|\mathcal{S}_{t-1})$, is a bounded function, since $\pi(x)$ is bounded. Moreover, since $\int_{\mathcal{X}} \tilde{\pi}(x)dx < +\infty$ and $\int_{\mathcal{X}} \tilde{q}_t(x|\mathcal{S}_{t-1})dx < +\infty$, then the L_1-distance between $\tilde{q}_t(x|\mathcal{S}_{t-1})$ and $\tilde{\pi}(x)$ is bounded for any t, i.e., $\int_{\mathcal{X}} |\tilde{q}_t(x|\mathcal{S}_{t-1}) - \tilde{\pi}(x)|dx < +\infty$. Let us consider the finite interval $\mathcal{I} = [s_1, \ s_{m_t}]$, then all the interpolation methods proposed in Section 3 to build $q_t(x|\mathcal{S}_{t-1})$ can be represented as a Taylor approximation of the order zero or one inside each interval. Hence, the discrepancy between $\tilde{q}_t(x|\mathcal{S}_{t-1})$ and $\tilde{\pi}(x)$ over $\mathcal{I}$ can be bounded as follows

$$\int_{\mathcal{I}} |\tilde{q}_t(x|\mathcal{S}_{t-1}) - \tilde{\pi}(x)|dx \leq \sum_{i=1}^{m_{t-1}-1} \int_{\mathcal{I}_i} |\tilde{q}_t(x|\mathcal{S}_{t-1}) - \tilde{\pi}(x)|dx = \sum_{i=1}^{m_{t-1}-1} \int_{\mathcal{I}_i} |r_\ell^{(i)}(x)|dx, \tag{18}$$

where $r_\ell^{(i)}(x)$ is the remainder associated to the ℓ-th order (with $\ell \in \{0, 1\}$ in our case) polynomial approximation of $\pi(x)$ inside the interval $\mathcal{I}_i$, as given by Taylor's theorem. Let us recall that the Lagrange form of this remainder is $r_\ell^{(i)}(x) = \frac{(x-s_i)^{\ell+1}}{(\ell+1)!} \frac{d^{\ell+1}\tilde{\pi}(x)}{dx^{\ell+1}}\Big|_{x=\xi}$, for a value $\xi \in [s_i, x]$. Moreover, since $x \in \mathcal{I}_i = [s_i, s_{i+1}]$, it is straightforward to show that

$$|r_\ell^{(i)}(x)| \leq \frac{(s_{i+1} - s_i)^{\ell+1}}{(\ell+1)!} C_\ell^{(i)}, \tag{19}$$

where $C_\ell^{(i)} = \max_{x \in \mathcal{I}_i} |\tilde{\pi}^{\ell+1)}(x)|$, and $\tilde{\pi}^{\ell+1)}(x)$ denotes the $(\ell+1)$-th derivative of $\tilde{\pi}(x)$, i.e.,

$\tilde{\pi}^{\ell+1)}(x) = \frac{d^{\ell+1}\tilde{\pi}(x)}{dx^{\ell+1}}$. Hence, replacing (19) in (18), we obtain

$$\sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} |r_\ell^{(i)}(x)|dx \leq \sum_{i=1}^{m_t-1} \frac{(s_{i+1}-s_i)^{\ell+2}}{(\ell+2)!} C_\ell^{(i)}. \tag{20}$$

Now, let us assume that a new point, $s' \in \mathcal{I}_k = [s_k, s_{k+1}]$ for $1 \leq k \leq m_t - 1$, is added at the next iteration. In this case, the construction of the proposal density changes only on the interval $\mathcal{I}_k$. Assume that $\mathcal{I}_k$ is split into $\mathcal{I}^{(1)} = [s_k, s']$ and $\mathcal{I}^{(2)} = [s', s_{k+1}]$, i.e., $\mathcal{I}_k = \mathcal{I}^{(1)} \cup \mathcal{I}^{(2)}$, then $\max_{x\in\mathcal{I}^{(j)}} |\tilde{\pi}^{\ell+1)}(x)| \leq \max_{x\in\mathcal{I}_k} |\tilde{\pi}^{\ell+1)}(x)|$ with $j \in \{1, 2\}$, and $(s' - s_k)^{\ell+2} + (s_{k+1} - s')^{\ell+2} < (s_{i+1} - s_i)^{\ell+2}$, for any $\ell \geq 0$, since $A^{\ell+2} + B^{\ell+2} < (A + B)^{\ell+2}$ for any $A, B > 0$ thanks to Newton's binomial theorem, and we have $A = s' - s_k > 0$ and $B = s_{k+1} - s' > 0$. Hence, the bound in Eq. (20) always decreases when a new support point is incorporated and we can finally ensure that $\lim_{t\to+\infty} \sum_{i=1}^{m_t-1} \int_{\mathcal{I}_i} |r_\ell^{(i)}(x)|dx = 0$, since support points become arbitrarily close as $t \to \infty$ (i.e., $s_{i+1} - s_i \to 0$), and thus the bound in the right hand side of (20) tends to zero as $t \to \infty$. Hence, we can guarantee that $\int_{\mathcal{I}} |\tilde{q}_t(x|\mathcal{S}_{t-1}) - \tilde{\pi}(x)|dx \to 0$ for $t \to +\infty$. Note that we cannot guarantee a monotonic decrease of the distance between $\tilde{q}_t(x|\mathcal{S}_{t-1})$ and $\tilde{\pi}(x)$ inside $\mathcal{I}$, since adding a new support point might occasionally lead to an increase in the discrepancy. However, we can guarantee that the upper bound on this distance decreases monotonically, thus ensuring that $\tilde{q}_t(x|\mathcal{S}_{t-1}) \to \tilde{\pi}(x)$ as $t \to \infty$, i.e., adding support points will eventually take us arbitrarily close to $\tilde{\pi}(x)$. Finally, w.r.t. the tails, note that the distance between $\tilde{q}_t$ and π remains bounded even for heavy tailed distributions. Furthermore, the interval $\mathcal{I}$ will become greater as $t \to +\infty$, since there is always a non-null probability of adding new support points inside the tails. Therefore, the probability mass associated to the tails decreases monotonically as $t \to \infty$. Hence, even though the distance between the target and the proposal may again increase occasionally due the introduction of a new support point in the tails, we can guarantee such a distance goes to zero as t goes to infinity. □

Proof of Theorem 4: Let us denote $\tilde{D}_t = d(\tilde{q}_t, \tilde{\pi})$ and $D_t = d(q_t, \pi)$. We can use an extended triangle inequality of type $d(A, E) \leq d(A, B) + d(B, C) + d(C, E)$, using the points $A = q_t$, $B = \frac{1}{c_t}\tilde{q}_t$, $C = \frac{1}{c_\pi}\tilde{q}_t$ and $E = \pi$, i.e.,

$$\begin{aligned} D_t = d(q_t, \pi) &\leq d\left(q_t, \frac{1}{c_t}\tilde{q}_t\right) + d\left(\frac{1}{c_t}\tilde{q}_t, \frac{1}{c_\pi}\tilde{q}_t\right) + d\left(\frac{1}{c_\pi}\tilde{q}_t, \pi\right) \\ &\leq d\left(q_t, \frac{1}{c_t}\tilde{q}_t\right) + d\left(\frac{1}{c_t}\tilde{q}_t, \frac{1}{c_\pi}\tilde{q}_t\right) + d\left(\frac{1}{c_\pi}\tilde{q}_t, \frac{1}{c_\pi}\tilde{\pi}\right) \leq 0 + \left|\frac{1}{c_t}c_t - \frac{1}{c_\pi}c_t\right| + \frac{1}{c_\pi}\tilde{D}_t, \end{aligned}$$

Hence, setting $C_t = \left|1 - \frac{c_t}{c_\pi}\right|$ we can finally write $C_t + \frac{1}{c_\pi}\tilde{D}_t \geq D_t$. Since $D_t \geq 0$, if $\lim_{t\to\infty} C_t = 0$

and $\lim_{t\to\infty} \tilde{D}_t = 0$ then $\lim_{t\to\infty} D_t = 0$ as well. Therefore, now we just need to prove $c_t \to c_\pi$ when $\lim_{t\to\infty} \tilde{D}_t = 0$. Clearly, $|\tilde{\pi}(x) - \tilde{q}_t(x|\mathcal{S}_{t-1})| \geq |\tilde{\pi}(x)| - |\tilde{q}_t(x|\mathcal{S}_{t-1})| = \tilde{\pi}(x) - \tilde{q}_t(x|\mathcal{S}_{t-1})$ since $\tilde{\pi}(x), \tilde{q}_t(x|\mathcal{S}_{t-1}) \geq 0$. The equality is given if $\tilde{\pi}(x) \geq \tilde{q}_t(x|\mathcal{S}_{t-1})$, so that $|\tilde{\pi}(x) - \tilde{q}_t(x|\mathcal{S}_{t-1})| = |\tilde{\pi}(x)| - |\tilde{q}_t(x|\mathcal{S}_{t-1})|$. Moreover, using again the triangle inequality, we can also write

$$||\tilde{\pi}|| = ||(\tilde{\pi} - \tilde{q}_t) + \tilde{q}_t|| \leq ||\tilde{\pi} - \tilde{q}_t|| + ||\tilde{q}_t|| \Rightarrow ||\tilde{\pi}|| - ||\tilde{q}_t|| \leq ||\tilde{\pi} - \tilde{q}_t||,$$

and in a similar fashion $-(||\tilde{\pi}|| - ||\tilde{q}_t||) \leq ||\tilde{\pi} - \tilde{q}_t||$. Combining the two previous inequalities, we obtain $||\tilde{\pi} - \tilde{q}_t|| \geq \Big|||\tilde{\pi}|| - ||\tilde{q}_t||\Big|$. Since $\tilde{D}_t = d(\tilde{\pi}, \tilde{q}) = ||\tilde{\pi} - \tilde{q}_t||$ and $c_\pi = ||\tilde{\pi}||$, $c_t = ||\tilde{q}_t||$, we can finally rewrite this expression as $\tilde{D}_t \geq |c_\pi - c_t|$. The expression above is also called *reverse triangle inequality*. Then, if $\lim_{t\to\infty} \tilde{D}_t = 0$, we also have $\lim_{t\to\infty} |c_\pi - c_t| = 0$, i.e., $c_t \to c_\pi$ for $t \to \infty$ and $C_t = \left|1 - \frac{c_t}{c_\pi}\right| \to 0$. □

B Limitations of the ARMS

In the ARMS algorithm, when a sample x' is rejected by the RS test (this can only happen when $q_t(x') > \pi(x')$), this point x' is added to the set $\mathcal{S}_t$ to update the proposal q_{t+1}. On the other hand, when a sample is initially accepted by the RS test (it could happen with $q_{t+1}(x') > \pi(x')$ and always happens if $q_{t+1}(x') \leq \pi(x')$), the ARMS method uses the MH acceptance rule to determine whether the new state is finally accepted or not. However, the proposal the proposal is never updated in this case. Its performance depends on the following two issues:

a) $W_{t+1}(x)$ should be constructed in such a way that $W_t(x) \geq V(x)$ for most intervals, and covering as much of the domain $\mathcal{D}$ as possible. In this case, the adaptive procedure of the ARMS method allows the proposal to improve almost everywhere. Indeed, in the extreme (positive) case that $q_{t+1}(x|\mathcal{S}_t) \geq \pi(x)\ \forall x \in \mathcal{D}$ and $\forall t \in \mathbb{N}$, the ARMS technique is reduced to the standard ARS algorithm (using the construction in Eq. (4)).

b) The addition of a support point within an interval must entail an improvement of the proposal pdf inside other neighbouring intervals when building $W_{t+1}(x)$. This allows that the proposal pdf can be improved even inside regions where $q_{t+1}(x|\mathcal{S}_t) < \pi(x)$. For instance, in the procedure described in Eq. (4), when a support point is added inside $\mathcal{I}_j$, the proposal pdf also changes in the intervals $\mathcal{I}_{j-1}$ and $\mathcal{I}_{j+1}$. Consequently, the drawback of not adding support points within the intervals where $q_{t+1}(x|\mathcal{S}_t) < \pi(x)$ is reduced, but may not completely eliminated, as we show below.

Therefore, the convergence of the proposal $q_{t+1}(x|\mathcal{S}_t)$ to the target pdf $\pi(x)$ cannot be guaranteed regardless of the construction used for $W_t(x)$, except for the special case where $W_t(x) \geq V(x)\ \forall x \in \mathcal{D}$ and $\forall t \in \mathbb{N}$, and the ARMS method becomes the standard ARS algorithm. This is owing to this fundamental structural limitation, caused by not adding support points inside regions where $q_{t+1}(x|\mathcal{S}_t) < \pi(x)$ at some time t. For instance, it is possible that inside some region $\mathcal{C} \subset \mathcal{D}$, where $q_{t+1}(x|\mathcal{S}_t) < \pi(x)$, we obtain a sequence of proposals $q_{t+1+\tau}(x) = q_{t+1}(x|\mathcal{S}_t)$ for an arbitrarily large value of τ. Furthermore, we could have an even more critical situation, where $q_{t+1+\tau}(x) = q_{t+1}(x|\mathcal{S}_t)$ $\forall x \in \mathcal{C}$ and $\forall \tau \in \mathbb{N}$, i.e., the proposal pdf does not change within an interval $\mathcal{C} \subset \mathcal{D}$.

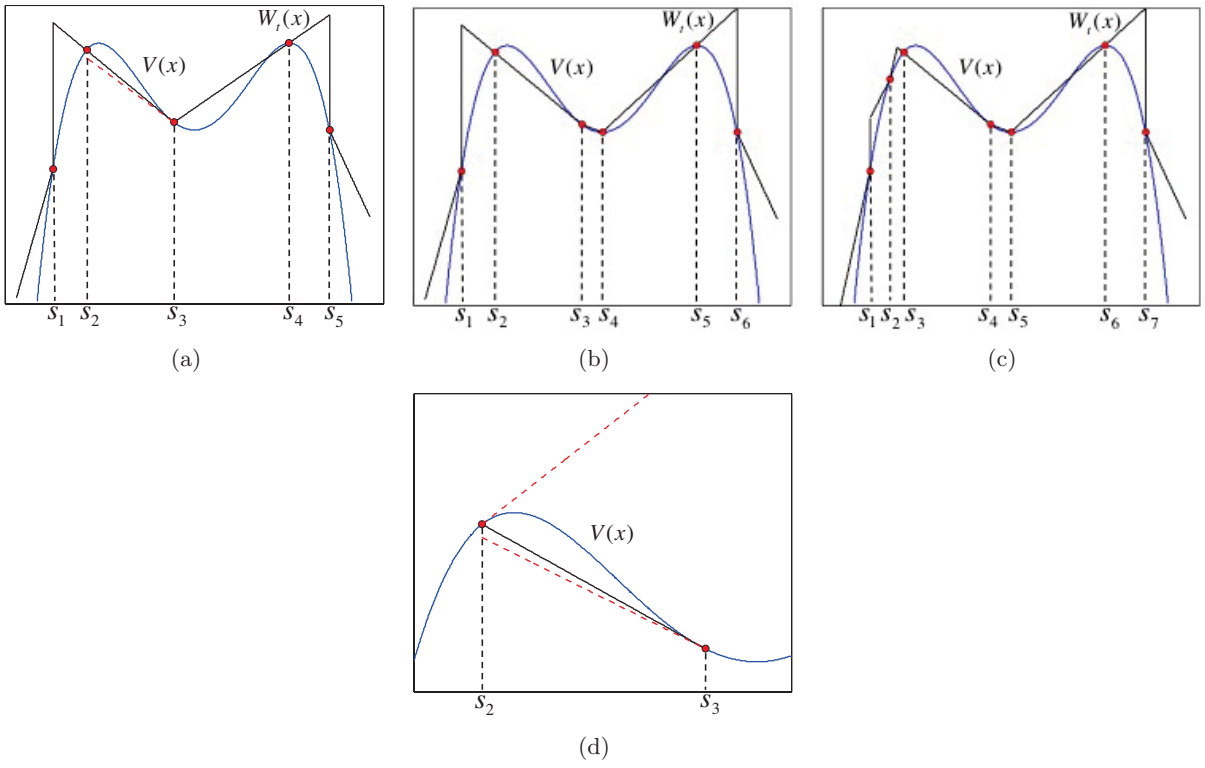


Figure 9: Example of a critical structural limitation in the adaptive procedure of ARMS. **(a)** Construction of $W_t(x)$ with 5 support points. Within $\mathcal{I}_2 = (s_2, s_3]$ we have $W_t(x) < V(x)$. **(b)-(c)** Adding new support points inside the contiguous intervals the construction of $W_t(x)$ does not vary within $\mathcal{I}_2$ ($\mathcal{I}_3$ in Figure (c)). **(d)** The secant line $L_{2,3}(x)$ passing through $(s_2, V(s_2))$ and $(s_3, V(s_3))$, and the two tangent lines to $V(x)$ at s_2 and s_3, respectively.

These limitations of the ARMS adaptation scheme can be illustrated with a simple graphical example. Consider a multi-modal target density, $\pi(x) = \exp(V(x))$, with $V(x)$ as shown in Figure 9(a). We build $W_t(x)$ using 5 support points and the procedure in Eq. (4). Note that we have $W_t(x) < V(x)$ for all x in the interval $\mathcal{I}_2 = (s_2, s_3]$, as shown in Figure 9(a), where the dashed line depicts the tangent line to $V(x)$ at s_3. From Eq. (4), the construction of $W_t(x)$ within this interval is $W_t(x) = \max\left\{L_{2,3}(x), \min\left\{L_{1,2}(x), L_{3,4}(x)\right\}\right\}$. From Figure 9(a), we see that

$\min\{L_{1,2}(x), L_{3,4}(x)\} = L_{3,4}(x)$ and $\max\{L_{2,3}(x), L_{3,4}(x)\} = L_{2,3}(x)$ $\forall x \in \mathcal{I}_2 = (s_2, s_3]$. Therefore, $W_t(x) = L_{2,3}(x)$ inside this interval, and this situation does not change when new support points are added inside the contiguous intervals. Figures 9(b) and 9(c) show that we can incorporate new support points, s_4 in Figure 9(b) and s_2 in Figure 9(c), arbitrarily close to the interval $\mathcal{I}_3$, $\mathcal{I}_2$ in Figures 9(a)-(b), without altering the construction of $W_t(x)$ within this interval. Indeed, consider now the limit case where two points are incorporated arbitrarily close to s_2 and s_3. In this extreme situation, the secant lines of the adjacent intervals become tangent lines, as shown in Figure 9(d), and the minimum between the two tangent lines is represented by the straight line tangent to s_3. Moreover, this tangent line stays always below the secant line, $L_{2,3}(x)$, passing through $(s_2, V(s_2))$ and $(s_3, V(s_3))$, meaning that $W_t(x) = L_{2,3}(x)$ even in this case.

C Tables and additional figures

Alg.	MSE	ACF(1)	ACF(10)	ACF(50)	m_T	Time	EI
ARMS-1	10.0395	0.4076	0.3250	0.2328	118.1912	1.0000	5057.833
ARMS-2	15.6756	0.8955	0.7210	0.4639	7.6126	0.1195	5003.612
ARMS-3	0.2398	0.8753	0.4410	0.0296	131.3360	0.3589	5127.336
ARMS-4	0.2874	0.8882	0.4758	0.0418	42.8872	0.2291	5038.887
ASM-1	3.0277	0.1284	0.1099	0.0934	152.6301	1.2274	5000
ASM-2	2.9952	0.1306	0.1125	0.0929	71.1478	0.2757	5000
ASM-3	0.0290	0.0535	0.0165	0.0077	279.6570	0.6494	5000
ASM-4	0.0354	0.0354	0.0195	0.0086	84.8742	0.3297	5000
ASMTM-1 ($N = 10$)	0.6720	0.0726	0.0696	0.0624	159.0060	2.3547	5000
ASMTM-1 ($N = 50$)	0.1666	0.0430	0.0395	0.0316	160.7579	6.4518	5000
ASMTM-2 ($N = 10$)	0.5632	0.0588	0.0525	0.0443	72.1628	1.1291	5000
ASMTM-2 ($N = 50$)	0.1156	0.0345	0.0303	0.0231	72.5270	4.3802	5000
ASMTM-3 ($N = 10$)	0.0105	0.0045	0.0001	0.0001	315.7808	2.6022	5000
ASMTM-3 ($N = 50$)	0.0099	0.0063	0.0001	0.0001	360.7323	10.5935	5000
ASMTM-4 ($N = 10$)	0.0108	0.0036	0.0011	0.0014	92.6660	1.8618	5000
ASMTM-4 ($N = 50$)	0.0098	0.0001	0.0001	0.0001	101.7775	7.2475	5000

Table 1: For each algorithm (Alg.), the table shows in different columns, the mean square error (MSE), the autocorrelation function (ACF(k)) at different lags, $k = 1, 10, 50$, the number of support points at the last iteration (m_T), the ratio between the algorithm and the ARMS-1 computing times (Time), and the effective number of iterations (EI). The class of ASMTM algorithms have been analyzed for two different choices of number of proposals, i.e. $N = 10$ and $N = 50$.

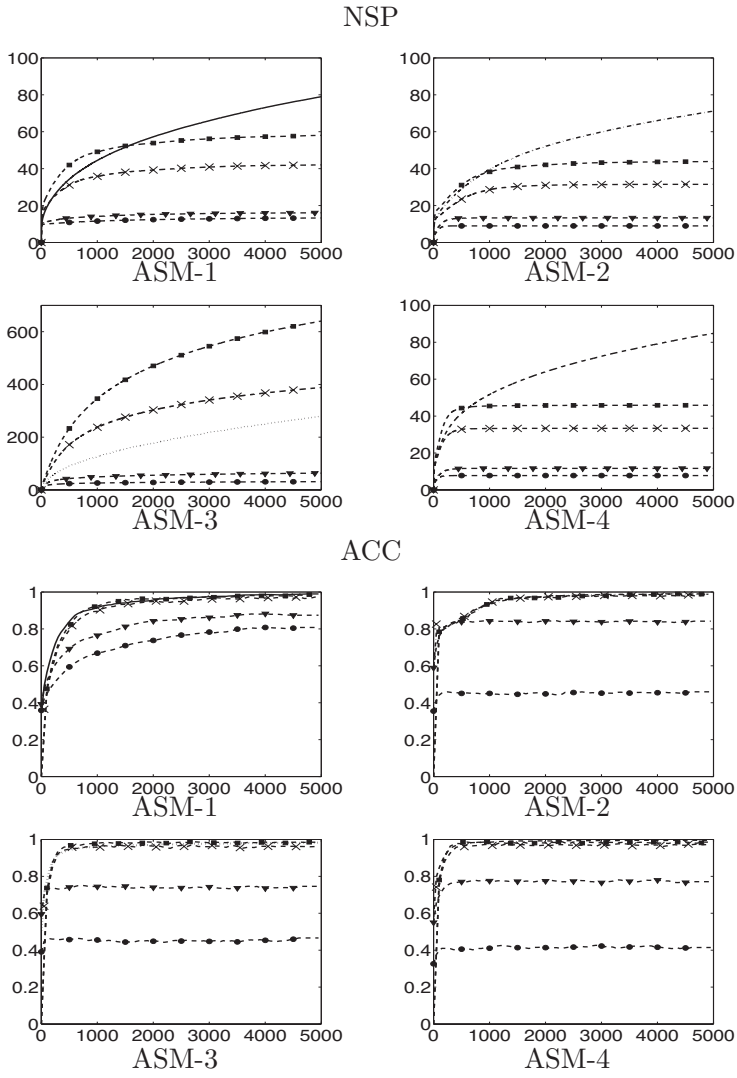


Figure 10: Number of support points (NSP) and acceptance rate (ACC) over the ASM chain iterations for different constructions. In each plot the results of the ASM with random test (line without symbol) is compared with the results of a deterministic test with $\varepsilon = 0.005$ (square), $\varepsilon = 0.01$ (cross), $\varepsilon = 0.1$ (triangle) and $\varepsilon = 0.2$ (circle).

Panel I

(a) $\mathcal{S}_0 = \{-1, 1, 20\}$ and $T = 5000$									
Alg.	Mean	SD	ACF(1)	ACF(10)	ACF(50)	m_T	$\underline{c}_T$	$\bar{c}_T$	Time
Slice	19.5039	6.0238	0.8759	0.8230	0.6244	-	0.9934	1.0089	0.8257
ARMS-1	20.2803	10.3538	0.8593	0.8112	0.6848	42.5690	0.7417	1.7490	1.0000
ASM-4	19.1416	2.7723	0.1182	0.0951	0.0785	112.7360	0.9933	1.0085	0.4679
ASMTM-4	19.9408	0.4342	0.0108	0.0054	0.0006	117.0665	0.9938	1.0083	3.0578
(b) $\mathcal{S}_0 = \{-1, 1, 70\}$ and $T = 5000$									
Alg.	Mean	SD	ACF(1)	ACF(10)	ACF(50)	m_T	$\underline{c}_T$	$\bar{c}_T$	Time
Slice	19.5039	6.0238	0.8759	0.8230	0.6244	-	0.9934	1.0089	1.0687
ARMS-1	0.3292	10.5940	0.2650	0.1597	0.1266	30.2585	0.4330	4.0560	1.0000
ASM-4	19.1196	2.4849	0.1081	0.0846	0.0689	111.9015	0.9934	1.0084	0.6174
ASMTM-4	19.9120	0.4483	0.0126	0.0066	0.0014	115.6575	0.9938	1.0083	3.9497
(c) $\mathcal{S}_0$ with points drawn from $\mathcal{U}([-70, 70])$ ($m_0 = 3$) and $T = 5000$									
Alg.	Mean	SD	ACF(1)	ACF(10)	ACF(50)	m_T	$\underline{c}_T$	$\bar{c}_T$	Time
Slice	19.5039	6.0238	0.8759	0.8230	0.6244	-	0.9934	1.0089	1.7232
ARMS-1	19.6855	4.9838	0.8802	0.7644	0.4837	12.1335	0.9223	1.0927	1.0000
ASM-4	18.7981	3.0211	0.1281	0.1072	0.0908	118.5636	0.9925	1.0085	0.9731
ASMTM-4	19.9276	0.4945	0.0145	0.0087	0.0033	128.1860	0.9937	1.0084	8.2425

Panel II

(a) $\kappa = 0.1$, $\mathcal{S}_0 = \{-1, 1, 20\}$ and $T = 5000$									
Alg.	Mean	SD	ACF(1)	ACF(10)	ACF(50)	m_T	$\underline{c}_T$	$\bar{c}_T$	Time
Slice	53.0797	14.5540	0.6835	0.3562	0.2666	-	0.9563	1.3878	0.6670
ARMS-1	61.1859	3.4341	0.0625	0.0219	0.0079	59.3500	0.7651	1.6203	1.0000
ASM-4	61.9253	1.6282	0.0283	0.0015	0.0005	121.1240	0.9575	1.1836	0.4344
ASMTM-4	61.9885	1.3193	0.0014	0.0003	0.0001	127.6415	0.9582	1.1966	2.5667
(b) $\kappa = 0.4$, $\mathcal{S}_0 = \{-1, 1, 20\}$ and $T = 5000$									
Slice	33.4459	4.6767	0.6933	0.5131	0.2230	-	0.9895	1.0113	0.4557
ARMS-1	33.9293	1.0835	0.1451	0.0375	0.0047	57.7728	0.9622	1.0394	1.0000
ASM-4	33.8768	0.7482	0.0247	0.0013	0.0007	131.7785	0.9896	1.0112	0.5212
ASMTM-4	33.9096	0.5660	0.0028	0.0003	-0.0002	137.9935	0.9897	1.0111	2.7921
(c) $\kappa = 0.01$, $\mathcal{S}_0 = \{-1, 1, 20\}$ and $T = 5000$									
Slice	-	-	-	-	-	-	-	-	-
ARMS-1	272.4381	45.4137	0.3977	0.2410	0.1403	55.3359	0.5753	7.6209	1.0000
ASM-4	384.9001	14.7383	0.0622	0.0051	-0.0002	114.2115	0.7663	3.1612	0.5537
ASMTM-4	385.5778	11.2497	0.0101	0.0005	0.0001	119.6660	0.7816	3.2146	2.8134

Table 2: Results of the slice sampler (Slice), ARMS with construction 1 (ARMS-1), ASM with construction 4 (ASM-4) and ASMTM with construction 4 (ASMTM-4), with $N = 10$ proposals, for the target Mix1 (panel I) and Mix2 (panel II), for different initial set of suppport points (panels (I.a), (I.b) and (I.c)) and for different values of the parameter κ (panels (II.a), (II.b) and (II.c)). Each row of a panel, shows in different columns, the mean (Mean), the mean estimate standard deviation (SD), the autocorrelation function (ACF(k)) at different lags, $k = 1, 10, 50$, the number of support points at the last iteration (m_T), the estimates of the normalizing constant ($\underline{c}_T$, $\bar{c}_T$) at the last iteration, and the ratio between the algorithm and the ARMS-1 computing times (Time).

	(a)				(b)			
Quantity of Interest	DI	ARMS-1	ASM-4	ASMTM-4	DI	ARMS-1	ASM-4	ASMTM-4
$T(50)$								
$\mathbb{E}(T(50))$	30.8112	29.8411	30.7904	30.7968	30.8112	30.8123	30.7921	30.7995
	-	(8.1242)	(0.1501)	(0.1498)	-	(0.1475)	(0.1482)	(0.1518)
$\mathbb{V}(T(50))$	108.8711	66.3152	109.342	109.1356	108.8711	108.961	109.1785	108.9996
$Sk(T(50))$	-0.6091	1.0629	-0.6138	-0.6102	-0.6091	-0.6104	-0.6112	-0.6102
$Ku(T(50))$	2.9668	73.7041	2.9772	2.9702	2.9668	2.9718	2.9718	2.9693
$Q_{0.95}(T(50))$ -	45.3989	41.6797	45.3917	45.3974	45.3989	45.4032	45.3947	45.3896
ACF(1)	-	0.2388	0.0108	0.0023	-	-0.0005	0.0069	0.0000
ACF(10)	-	0.1701	0.0000	0.0000	-	-0.0003	0.0007	0.0000
ACF(50)	-	0.0892	0.0000	0.0000	-	0.0000	0.0000	0.0000
Z								
$\mathbb{E}(Z)$	0.4838	0.4997	0.4842	0.4841	0.4838	0.4839	0.4842	0.4840
	-	(0.1268)	(0.0020)	(0.0020)	-	(0.0019)	(0.0019)	(0.0020)
$\mathbb{V}(Z)$	0.0185	0.0092	0.0187	0.0186	0.0185	0.0185	0.0186	0.0186
$Sk(Z)$	1.2600	-0.4848	1.2682	1.2626	1.26	1.2629	1.264	1.2619
$Ku(Z)$	4.5066	68.7498	4.5376	4.518	4.5066	4.5211	4.5227	4.5155
$Q_{0.95}(Z)$	0.77004	0.6689	0.7714	0.7707	0.77004	0.77	0.7709	0.7706
ACF(1)	-	0.2834	0.0143	0.0036	-	-0.0005	0.0069	0.0000
ACF(10)	-	0.2007	0.0000	0.0000	-	-0.0003	0.0007	0.0000
ACF(50)	-	0.1057	0.0000	0.0000	-	0.0000	0.0000	0.0000
Y								
$\mathbb{E}(Y)$	20.9016	20.2615	20.8881	20.8928	20.9016	20.9014	20.8902	20.8951
	-	(7.3888)	(0.0318)	(0.0319)	-	(0.0786)	(0.0785)	(0.0800)
$\mathbb{V}(Y)$	30.36526	15.023	30.5973	30.4875	30.36526	30.4107	30.5147	30.4377
$Sk(Y)$	-1.2600	0.4848	-1.2682	-1.2626	-1.26	-1.2629	-1.264	-1.2619
$Ku(Y)$	4.5066	68.7498	4.5376	4.518	4.5066	4.5211	4.5227	4.5155
$Q_{0.95}(Y)$	27.29774	25.1935	27.2952	27.2971	27.29774	27.299	27.2962	27.2945
ACF(1)	-	0.2834	0.0143	0.0036	-	-0.0005	0.0069	0.0000
ACF(10)	-	0.2007	0.0000	0.0000	-	-0.0003	0.0007	0.0000
ACF(50)	-	0.1057	0.0000	0.0000	-	0.0000	0.0000	0.0000

Table 3: Results for deterministic integration (DI) and stochastic integration with 5,000 iterations of the ARMS with construction 1 (ARMS-1), ASM with construction 4 (ASM-4) and ASMTM with construction 4 (ASMTM-4), with $N = 10$ proposals. In all algorithms the initial set of support points is $\mathcal{S}_0 = \{20, 40, 60\}$ (panel a) and $\mathcal{S}_0 = \{0, 20, 40, 60\}$ (panel b). The Monte Carlo standard errors, of the sample mean given above, are reported in parenthesis. The autocorrelation at the k-lag ($ACF(k)$), $k = 1, 10, 50$, is given in the last three rows.

(a) $\mathcal{S}_0 = \{0.0001, 0.001, 0.005, 1\}$ and $T = 5000$						
Alg.	Mean	ACF(1)	ACF(10)	ACF(50)	m_T	EI
MH	6.3885 10^{-4} (1.4710 10^{-6})	0.0566	-8.5215 10^{-5}	-2.3288 10^{-4}	-	5000
ARMS-1	6.3887 10^{-4} (1.3652 10^{-6})	0.0011	-5.6182 10^{-4}	8.0548 10^{-5}	51.1515	5047.15
ASM-4	6.3939 10^{-4} (1.4635 10^{-6})	0.0286	-6.5102 10^{-5}	-1.2774 10^{-4}	55.7780	5000
ASMTM-4 $N = 10$	6.3886 10^{-4} (1.4071 10^{-6})	-0.0013	-3.0372 10^{-4}	-6.4483 10^{-4}	61.2585	5000
(b) $\mathcal{S}_0 = \{0.0001, 0.0003, 0.005, 1\}$ and $T = 5000$						
Alg.	Mean	ACF(1)	ACF(10)	ACF(50)	m_T	EI
MH	6.3885 10^{-4} (1.4710 10^{-6})	0.0566	-8.5215 10^{-5}	-2.3288 10^{-4}	-	5000
ARMS-1	6.3153 10^{-4} (3.3260 10^{-5})	0.9865	0.9289	0.7461	14.0170	5010.01
ASM-4	6.3920 10^{-4} (1.4404 10^{-6})	0.0177	1.7499 10^{-5}	2.0126 10^{-4}	55.6900	5000
ASMTM-4 $N = 10$	6.3883 10^{-4} (1.4120 10^{-6})	-4.5178 10^{-4}	-5.0299 10^{-4}	-5.2821 10^{-4}	60.5280	5000

Table 4: Results for different initial set of support points (panels (a) and (b)) and different algorithms: MH, ARMS, ASM and ASMTM. In columns: the mean (Mean), the autocorrelation function (ACF(k)) at different lags, $k = 1, 10, 50$, the number of support points at the last iteration (m_T), and the effective number of iterations (EI). In parenthesis the standard deviation of the estimated mean.

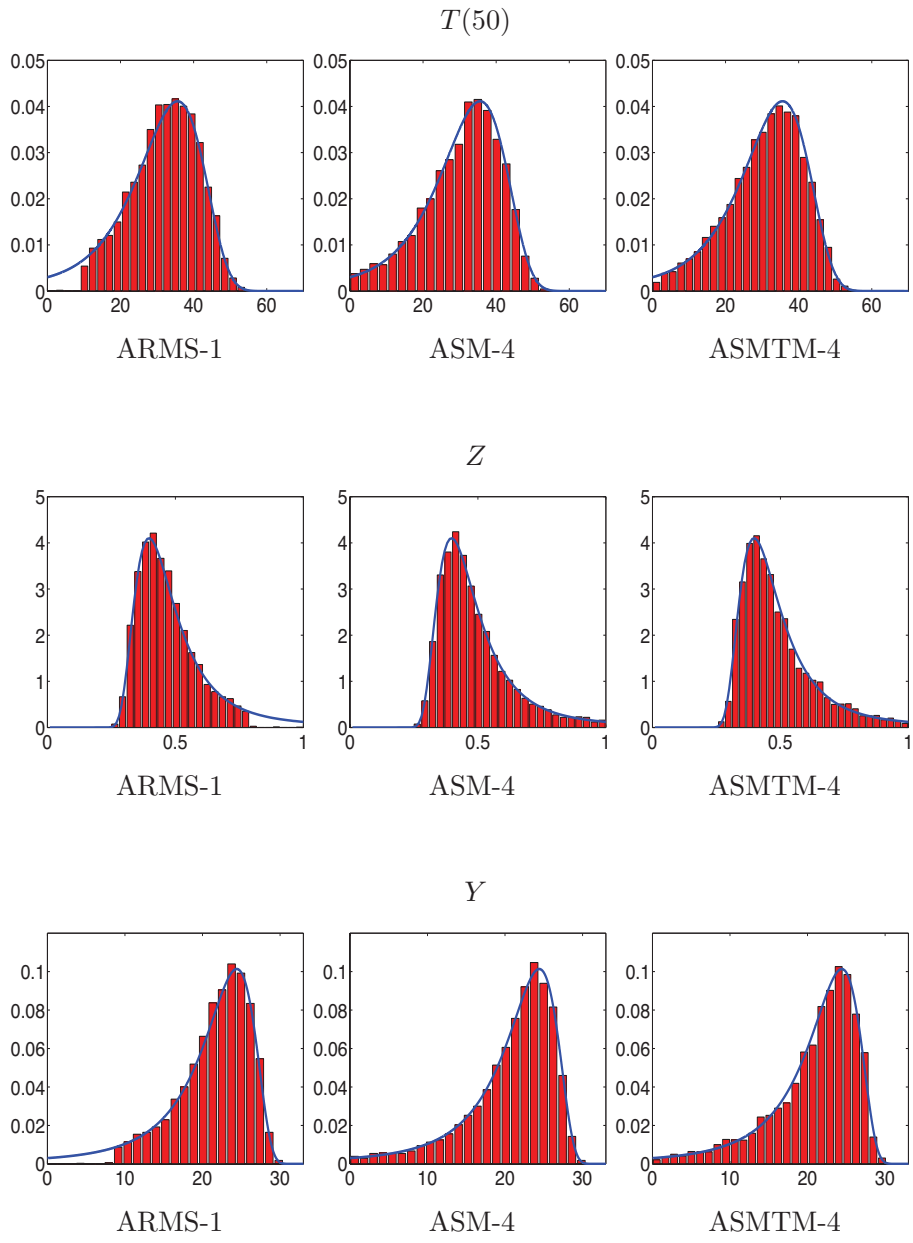


Figure 11: Makeham's density of parameters $A = 0.001$, $B = 0.0000070848535$, $C = 1.1194379$ (solid line) and histograms of 5,000 samples from the Makeham's distribution generated with ARMS-1, ASM-4 and ASMTM-4 ($N = 10$).