Majorize-Minimize Adapted Metropolis-Hastings Algorithm
Yosra Marnissi, Emilie Chouzenoux, Amel Benazza-Benyahia, Jean-Christophe Pesquet

To cite this version:
Yosra Marnissi, Emilie Chouzenoux, Amel Benazza-Benyahia, Jean-Christophe Pesquet. Majorize-Minimize Adapted Metropolis-Hastings Algorithm. 2018. <hal-01909153>

HAL Id: hal-01909153
https://hal.archives-ouvertes.fr/hal-01909153
Submitted on 30 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Majorize-Minimize Adapted Metropolis–Hastings Algorithm

Yosra Marnissi, Student Member, IEEE, Emilie Chouzenoux, Member, IEEE, Amel Benazza-Benyahia, Member, IEEE, and Jean-Christophe Pesquet, Fellow, IEEE

Abstract—The dimension and the complexity of inference problems have dramatically increased in statistical signal processing. It thus becomes mandatory to design improved proposal schemes in Metropolis-Hastings algorithms, providing large proposal transitions that are accepted with high probability. The proposal density should ideally provide an accurate approximation of the target density with a low computational cost. In this paper, we derive a novel Metropolis-Hastings proposal, inspired from Langevin dynamics, where the drift term is preconditioned by an adaptive matrix constructed through a Majorization-Minimization strategy. We propose several variants of low-complexity curvature metrics applicable to large scale problems. We demonstrate the geometric ergodicity of the resulting chain for the class of super-exponential distributions. The proposed method is shown to exhibit a good performance in two signal recovery examples.

Index Terms—MCMC methods, Langevin diffusion, Majorization-Minimization, signal recovery

I. INTRODUCTION

THE resolution of an inverse problem consists of estimating an unknown signal from measurements based on the direct model linking the target signal to the observed one. However, perfect measurements are generally not available due to the presence of some random parasite signals that make difficult the extraction of useful information. In this work, we consider the following observation model:

\[ z = D(Hx) \]  

where \( \bar{x} \in \mathbb{R}^Q \) denotes the target signal, \( z \in \mathbb{R}^N \) is the measured data, \( H \in \mathbb{R}^{N \times Q} \) is an observation matrix describing the linear acquisition model, and \( D \) expresses nonlinear degradations and measurement errors considered as noise. Such model arises in several signal processing applications (deblurring, denoising, super resolution, reconstruction, compressive sensing, inpainting) with appropriate definitions of the model are simple, the posterior law is almost always tractable in the sense that it can only be known up to a multiplicative constant and/or has a complicated form which requires massive computing resources for handling it. Regarding the difficulty in directly dealing with the posterior distribution, many methods have been proposed [6]. In this paper, we are interested in Markov chain Monte Carlo (MCMC) simulation based techniques for large scale signal processing problems.

The Bayesian framework has been widely adopted to perform the task of retrieving an estimate of the target signal given the data \( z \) and the model matrix \( H \). Bayesian modeling considers the parameters of interest as random variables rather than deterministic ones. Hence, this approach requires to specify a prior probability density \( p(x) \) that describes what is known about the sought signal before data are observed. Estimates are then computed relying on the posterior law that takes into account the prior \( p(x) \) combined with information about observations \( p(z|x) \) via Baye’s rule:

\[ p(x|z) = \frac{p(z|x)p(x)}{\int_{\mathbb{R}^Q} p(z|x')p(x')dx'}. \] (2)

A major challenge in Bayesian methods is the calculation of the posterior distribution, or more precisely, its exploration. In addition, nowadays, it is common in many fields such as medicine, astronomy and microscopy, to handle huge amounts of data with increasingly sophisticated models [5]. In these challenging settings, even if the prior and the observation model are simple, the posterior law is almost always intractable in the sense that it can only be known up to a multiplicative constant and/or has a complicated form which requires massive computing resources for handling it. Regarding the difficulty in directly dealing with the posterior distribution, many methods have been proposed [6]. In this paper, we are interested in Markov chain Monte Carlo (MCMC) simulation based techniques for large scale signal processing problems.

MCMC methods are stochastic simulation methods that allow to approximate a given target distribution such as the posterior law, by relying on Markov chain theory and Monte Carlo integration. They proceed in two main steps. First, a Markov chain is built with a given transition rule so that its stationary distribution, Monte Carlo approximation is used to infer the posterior characteristics, A famous MCMC method is the Metropolis-Hastings (MH) one. This algorithm has been firstly introduced in [11] and then has been generalized to a more statistical setting in [7]. In order to draw a sample from a target distribution \( p(x|z) \), two steps are applied alternately. First, a sample \( \tilde{x}^{(t)} \) is generated according to some proposal distribution of density \( q(.|x^{(t)}) \) that may depend on the current state \( x^{(t)} \) at each iteration \( t \) and should be easy to draw from. The proposed variable is then accepted or rejected according to the following acceptance probability:

\[ \alpha(x^{(t)}, \tilde{x}^{(t)}) = \min \left( 1, \frac{p(\tilde{x}^{(t)}|z)q(x^{(t)}|\tilde{x}^{(t)})}{p(x^{(t)}|z)q(\tilde{x}^{(t)}|x^{(t)})} \right). \] (3)
However, the performance of the MH algorithm is obviously strongly related to the choice of the proposal distribution. This issue becomes especially critical in large scale problems. In general, when selecting a proposal in MH algorithms, one should consider two issues. First, whilst MH algorithms are guaranteed to yield samples from the target distribution after some burn-in period, the number of iterations required to reach convergence may be infeasibly large. Second, the generated samples after convergence may be correlated. This correlation originates from two main sources: (i) the correlation introduced by keeping unchanged the parameter value because the newly generated one was rejected; (ii) the correlation between successive samples for non-independent proposals. A poorly mixed chain tends to generate samples that are highly correlated leading to an incomplete summary of the target distribution for large scale problems. In this respect, a large amount of works reflect the dependence structure of the target distribution to accelerate the exploration of the whole target space [17]. Intuitively, a good mixing behavior could make the Markov chain more likely to sample from. This problem is often tackled in an empirical manner. However, it is also possible to determine theoretically some variants of ULA in [18], [29], [30].

The popularity of this algorithm is mainly related to the simplicity of implementation. However, the RW usually takes too many steps to reach stability for high dimensional models. Furthermore, slow convergence together with bad mixing behavior could make the Markov chain more likely to get trapped into some regions and thus fail to explore efficiently the whole target space [17]. Intuitively, a good proposal density should take advantage of the local properties of the target distribution to accelerate the exploration of regions with high probability values. In particular, it should reflect the dependence structure of the target distribution for large scale problems. In this respect, a large amount of works has been devoted to construct proposals in MH algorithms in an attempt to meet these requirements [18]–[26]. In this work, we are interested in proposals based on the Euler discretization of the Langevin stochastic differential equation where the drift term accounts for the slope and curvature of the target law. Our main contribution is to propose a preconditioned version of the standard Metropolis Hastings adapted Langevin algorithm using an adaptive matrix based on a Majorize-Minimize strategy.

This paper is organized as follows. In Section II, we formulate the problem and we give a brief overview of the Langevin diffusion process. In Section III, we describe the new Majorize-Minimize adapted MH algorithm. In Section IV, a particular attention is paid to the convergence proof of the proposed algorithm. Section V is devoted to experimental results. Finally, some concluding remarks are drawn in Section VI.

II. PROBLEM STATEMENT AND RELATED WORK

A. Langevin diffusion

A $Q$-dimensional Langevin diffusion is a continuous time Markov process $(x(t))_{t \in [0, +\infty[}$ with values in $\mathbb{R}^Q$ defined as the solution to the following stochastic differential equation [27]:

$$\forall t \in [0, +\infty[ \quad dx(t) = b(x(t))dt + V(x(t))dB(t), \quad (4)$$

where $x(0) \in \mathbb{R}^Q$, $(B(t))_{t \in [0, +\infty[} \in \mathbb{R}^Q$ is a Brownian motion, and for every $x \in \mathbb{R}^Q$, $V(x) \in \mathbb{R}^Q \times \mathbb{R}$ is the volatility matrix and $b(x) = (b_i(x))_{i=1}^Q$ is the drift term defined as follows:

$$\forall i \in \{1, \ldots, Q\} \quad b_i(x) = \frac{1}{2} \sum_{j=1}^Q A_{i,j}(x) \frac{\partial \log \pi(x)}{\partial x_j} + |A(x)|^{1/2} \sum_{j=1}^Q \frac{\partial}{\partial x_j} \left( A_{i,j}(x) |A(x)|^{-1/2} \right) \quad (5)$$

where $A(x) = V(x)V(x)^T = (A_{i,j}(x))_{1 \leq i, j \leq Q}$ is a symmetric positive definite matrix and $|A(x)|$ denotes its determinant. Note that the process is stationary and $\pi$ is the density of the stationary distribution of the diffusion i.e., if a state $x(t_0)$ follows the distribution of density $\pi$, all subsequent states $x(t_0 + \tau)$, $\tau > 0$ also follow the same distribution. This density is here assumed to be differentiable. Thereby, when $\pi(\cdot) = p(\cdot \mid z)$, one can construct a Langevin Markov chain whose stationary law is the target posterior distribution. In the following, this choice for $\pi$ is made.

The Langevin diffusion describes a dynamic in continuous time. However, one can still approximate this equation by discretizing time. This is done by splitting the time interval into a series of small intervals of length $\Delta t = \epsilon^2$. The smaller the value of $\epsilon$, the closer the approximation to the dynamic in continuous time. Numerous procedures have been developed for time discretization [28]. We focus here on Euler’s scheme. Then, the Langevin diffusion reads

$$\forall t \in \mathbb{N} \quad x^{(t+1)} = x^{(t)} + \epsilon^2 b(x^{(t)}) + \epsilon A^{1/2}(x^{(t)}) \omega^{(t+1)} \quad (6)$$

where $\epsilon > 0$ is the stepsize resulting from Euler’s discretization and $(\omega^{(t)})_{t \in \mathbb{N}} \in \mathbb{R}^Q$ is a realization of zero-mean white noise with covariance matrix $I_Q$. Scheme (6) is referred to as the Unadjusted Langevin Algorithm (ULA) [18]. Due to the discretization error, the Markov chain following ULA scheme may behave differently from the diffusion process resulting from (5). In particular, it may sway away from the target stationary distribution as pointed out in [18], [23]. This discrepancy can be corrected by adding a Metropolis acceptance test at each iteration to guarantee the reversibility of the chain with respect to the posterior distribution. The resulting sampler can be seen as an MH algorithm where, for every $t \in \mathbb{N}$, $g(\cdot \mid x^{(t)})$ is the density of a Gaussian distribution with mean $x^{(t)} + \epsilon^2 b(x^{(t)})$ and covariance matrix $\epsilon^2 A(x^{(t)})$. Note that convergence properties have also been obtained for some variants of ULA in [18], [29], [30].
It is worth noting that two scale parameters play an important role in (6): $\varepsilon$ determines the length of the proposed jumps, whereas the scale matrix $\mathbf{A}(\cdot)$ controls their direction. Various classes of algorithms have been developed from this diffusion model depending on the choice of this matrix. In the subsequent subsection, we will review the most popular ones.

B. Choice of the scale matrix

The standard Metropolis adjusted Langevin algorithm (MALA) is the simplest form of diffusion (6) when $\mathbf{A}(\cdot)$ equals $\mathbf{I}_Q$ [18]:

$$
(\forall t \in \mathbb{N}) \quad \mathbf{x}(t+1) = \mathbf{x}(t) + \frac{\varepsilon^2}{2} \nabla \log p(\mathbf{x}(t)|z) + \varepsilon \mathbf{w}(t+1). \tag{7}
$$

It can be proved that MALA has $p(\mathbf{x}|z)$ as its stationary distribution and is more likely to accept proposed values than a standard RW [18]. Indeed, the gradient information of the target distribution allows the chain to be guided toward regions of higher probability where most of the samples should lie and hence, it enables to achieve high acceptance rates. As a consequence, MALA explores the invariant distribution much faster than the standard RW [14], [31]. Moreover, it should be noted that a bad adjustment of $\varepsilon$ can significantly affect the convergence rate especially for high dimensional problems [26]. For this reason, many methods have focused on how to choose a suitable stepsize in order to make the asymptotic average acceptance rate bounded away from zero in high dimensions [21], [26]. Despite these improvements, when the variables of interest are strongly correlated with strongly differing variances, MALA algorithm fails to explore efficiently the target space. In fact, since the third term in the MALA update is an isotropic Brownian motion, the discretization stepsize $\varepsilon$ in such a parameter space, is generally constrained to take very small values in order to deal with the directions with smallest variances, which may result in a slow convergence of the algorithm, poor mixing of the chain and highly correlated samples [23]. The performance of MALA can be improved by introducing a scale matrix different from the identity matrix [27]. Some approaches have been proposed to accelerate the algorithm by preconditioning the proposal density with a constant scale matrix [20], according to the following scheme:

$$
(\forall t \in \mathbb{N}) \quad \mathbf{x}(t+1) = \mathbf{x}(t) + \frac{\varepsilon^2}{2} \mathbf{A} \nabla \log p(\mathbf{x}(t)|z) + \varepsilon \mathbf{A}^{1/2} \mathbf{w}(t+1), \tag{8}
$$

where $\mathbf{A} \in \mathbb{R}^{Q \times Q}$ is a constant symmetric positive definite matrix. While the stepsize $\varepsilon$ can easily be tuned with respect to the asymptotic acceptance rate, there is no clear guiding strategies for the selection of the constant matrix in the absence of some knowledge about the moments of the target density which are usually unknown. Furthermore, the use of the same preconditioning matrix in the whole algorithm may be inefficient since optimal scaling in the burn-in period may differ from that in the stationary phase [32]. Therefore, rather than employing a fixed global scale matrix in the proposal density, a position dependent matrix may be employed to take into account the local structure of the target density at each state of the Markov chain. In that respect, many algorithms [15], [22]–[25], [33] rely on adaptive procedures where $\mathbf{A}(\cdot)$ is tuned automatically according to the past behavior of the Markov chain resorting to some deterministic optimization tools. For example, when setting $\mathbf{A}(\mathbf{x})$ to the inverse of the Hessian matrix of $-\log p(\mathbf{x}|z)$ at every $\mathbf{x} \in \mathbb{R}^Q$, and assuming a locally constant curvature, the term involving the derivatives of the scale matrix in (5) reduces to zero. Consequently, the generated sample at each iteration $t \in \mathbb{N}$ reads:

$$
\mathbf{x}(t+1) = \mathbf{x}(t) + \frac{\varepsilon^2}{2} \mathbf{A}(\mathbf{x}(t)) \nabla \log p(\mathbf{x}(t)|z) + \varepsilon \mathbf{A}^{1/2}(\mathbf{x}(t)) \mathbf{w}(t+1) \tag{9}
$$

where, for every $\mathbf{x} \in \mathbb{R}^Q$, $\mathbf{A}^{-1}(\mathbf{x}) = -\nabla^2 \log p(\mathbf{x}|z)$ that is, for every $i \in \{1, \ldots, Q\}$ and, for every $j \in \{1, \ldots, Q\}$, $[\mathbf{A}^{-1}]_{i,j}(\mathbf{x}) = -\frac{\partial^2 \log p(\mathbf{x}|z)}{\partial \mathbf{x}_i \partial \mathbf{x}_j}$. Consequently, the computation of the drift term $\mathbf{b}(\cdot)$ becomes a scaled Newton step for minimizing $-\log p(\cdot|z)$. Thus, a new sample of the Newton-based MCMC is more likely drawn from a highly probable region and then more likely accepted, which can speed up the convergence of the sampling process [24], [25], [33]. Note that, in practice, this method has a high computational cost since it requires the computation of the full Hessian matrix and its inverse at each iteration. This is especially critical for large scale problems and/or when the Hessian matrix is not positive definite. One appealing solution is to replace the Hessian by a scale matrix that provides information similar to the Hessian with a lower computational cost. In particular, several methods rely on the Fisher information matrix as a preconditioning matrix in the Langevin diffusion [22], [23], and can thus be interpreted as the discretization of the MALA algorithm directly on the natural Riemannian manifold where the parameters live. In the following, we propose a new approach where the scale matrix of the Langevin diffusion is chosen according to a Majorize-Minimize strategy.

III. PROPOSED ALGORITHM

A. Majorize-Minimize framework

The Majorization-Minimization (MM) principle is a powerful tool for designing optimization algorithms. The idea behind the MM approach is to replace an original complicated minimization problem with successive minimizations of some well chosen surrogate functions, satisfying the so-called tangent majorant conditions [34]–[36]:

**Definition III.1. Tangent majorant.**
Let $\mathcal{F} : \mathbb{R}^Q \to \mathbb{R}$ and let $\mathbf{x}' \in \mathbb{R}^Q$. A function $f(\mathbf{x}', \cdot)$ is said to be a tangent majorant function of $\mathcal{F}$ at $\mathbf{x}'$ if

$$
\left\{ \begin{array}{ll}
P_1 : & f(\mathbf{x}', \mathbf{x}) = \mathcal{F}(\mathbf{x}'), \\
P_2 : & f(\mathbf{x}', \mathbf{x}) \geq \mathcal{F}(\mathbf{x}) \quad (\forall \mathbf{x} \in \mathbb{R}^Q). 
\end{array} \right. \tag{10}
$$

Let $\mathbf{x}^{(0)} \in \mathbb{R}^Q$ be an arbitrary initial value and let $(\mathbf{x}(t))_{t \geq 1}$ be the sequence constructed as follows

$$
(\forall t \in \mathbb{N}) \quad \mathbf{x}(t+1) = \arg\min_{\mathbf{x} \in \mathbb{R}^Q} f(\mathbf{x}(t), \mathbf{x}). \tag{11}
$$
According to (10), the scheme (11) will produce a monotonically decreasing sequence $(J(x^{(l)}))_{l\in\mathbb{N}}$ since we have
\[
J(x^{(l)}) = f(x^{(l)}, x^{(l)}) \geq f(x^{(l)}, x^{(l+1)}) \geq J(x^{(l+1)})
\]
where (a) follows from the tangency property $P_1$, (b) from the minimization step (11), and (c) from the majorization property $P_2$ (see Figure 1). Then, under mild assumptions, the sequence can be shown to converge to a stationary point of $J$ [37].

Fig. 1: MM algorithm: the new iterate $x^{(l+1)}$ is the minimizer of the tangent majorant $f(x^{(l)} \cdot )$ of $J$ at $x^{(l)}$.

The performance of MM algorithms depends crucially on the choice of the surrogate function $f$. In particular, it has to be chosen so that a minimizer of it is easy to compute. A simple choice is to adopt an MM quadratic strategy, which consists in assuming the existence, for every $x' \in \mathbb{R}^Q$, of a positive definite matrix $Q(x') \in \mathbb{R}^{Q \times Q}$ such that the following quadratic function defined, for every $x \in \mathbb{R}^Q$, by
\[
f(x', x) = J(x') + (x-x')^T \nabla J(x') + \frac{1}{2} (x-x')^T Q(x')(x-x')
\]
is a tangent majorant of $J$ at $x'$. Then, the MM optimization algorithm reduces to building a sequence $(x^{(l)})_{l\in\mathbb{N}}$ through the following preconditioned gradient scheme:
\[
(\forall l \in \mathbb{N}) \quad x^{(l+1)} = x^{(l)} - \frac{\epsilon^2}{2} Q^{-1}(x^{(l)}) \nabla J(x^{(l)})
\]
with $\epsilon \in [0, \sqrt{2}]$ is a relaxation stepsize. Note that (14) implies that inequality (b) in (12) is satisfied, by noticing that $2\epsilon^{-2} Q(x') \succeq Q(x')$, for every $x' \in \mathbb{R}^Q$ and every $\epsilon \in [0, \sqrt{2}]$.

**B. Proposed sampling algorithm**

In this work, we propose to extend the idea behind the aforementioned MM quadratic strategy to the context of stochastic samplers. More specifically, our idea is to push the proposal distribution of the MH algorithm at each iteration from the current state to a region with high density value. Unlike the RW where the proposal is centered on the current state, we propose to pick the mean of the proposal density using an MM search step of the form (14), and then to explore the space around this center according to an MM curvature matrix $Q(x^{(l)})$ that should well describe the local curvature of the target distribution. This results in a preconditioned Langevin proposal where the scale matrix $A(x^{(l)})$ in (9), equal to the inverse of the curvature matrix $Q(x^{(l)})$, is constructed according to the MM strategy. Similarly to Newton-based MCMC methods, the drift term, when assuming zero curvature changes, leads, from a current state $x^{(l)}$, to a state with a higher value of $\log p(x | z)$ since it results from an iteration of MM algorithm minimizing $J(x) = -\log p(x | z)$. Consequently, the obtained proposal reduces to a stochastically perturbed version of an MM iteration for minimizing $-\log p(x | z)$. The proposed sample is then subjected to the accept/reject rule of the MH algorithm. The resulting sampler called 3MH is described by Algorithm 1.

**Algorithm 1 Majorize-Minimize adapted Metropolis–Hastings (3MH) algorithm**

**Initialize**: $x^{(0)} \in \mathbb{R}^Q$, $\epsilon \in [0, \sqrt{2}]

1: \textbf{for } t = 0, 1, \ldots \textbf{do}
2: \quad \text{Generate } u \sim U(0, 1)
3: \quad \text{Compute } \alpha(x^{(t)}, \tilde{x}^{(t)}) = \min \left(1, \frac{p(\tilde{x}^{(t)} | z) g(\tilde{x}^{(t)} | x^{(t)})}{p(x^{(t)} | z) g(x^{(t)} | x^{(t)})} \right)
4: \quad \text{where, for every } v \in \mathbb{R}^Q,
5: \quad \quad g(\cdot | v) \propto |Q(v)|^{\frac{1}{2}} \exp \left( -\frac{1}{2\epsilon^2} \| \cdot - m(v) \|_2^2 Q(v) \right)
6: \quad \text{if } u < \alpha(x^{(t)}, \tilde{x}^{(t)}) \text{ then}
7: \quad \quad \text{Accept: } x^{(t+1)} = \tilde{x}^{(t)}
8: \quad \quad \text{else}
9: \quad \quad \text{Reject: } x^{(t+1)} = x^{(t)}
10: \textbf{end if}
11: \textbf{end for}

The metric $Q(\cdot)$ is thus the precision matrix of the Gaussian proposal distribution, the choice of which is crucial for the efficiency of the sampling algorithm. We propose to set $Q(x^{(l)})$ at each iteration $t \in \mathbb{N}$ such that (13) is a tangent majorant to the minus logarithm of the posterior density at the current state $x^{(t)}$, i.e. it should satisfy Properties $P_1$ and $P_2$ in (10). Furthermore, for practical efficiency, it must be chosen so as to provide a good approximation to the local curvature of the posterior distribution. In the following, we propose a general procedure for building such a set of suitable preconditioning matrices $\{Q(x)\}_{x \in \mathbb{R}^Q}$ under some mild conditions on the posterior distribution.

**C. Construction of the tangent majorant**

We focus on the case when the minus-log of the target density function $J = -\log p(\cdot | z)$ can be expressed up to an
can use smoothed approximations of these functions that have
algorithms require the use of differentiable regularizations, one
the Bernoulli-Gaussian models [38], [39]. As Langevin based
that can be well described using suitable heavy tailed functions
while the remaining operators give the high frequency ones
coefficients and may be the operator that computes low frequency wavelet
Assumption III.1.
\[^{\text{Assumptions III.1(ii)-(iv) are satisfied for several commonly}}^{\text{used prior models such as the Student-t distribution,}}\]
\[^{\text{the Gaussian distribution as well as smoothed approximation of}}^{\text{asymptotically constant with a quadratic behavior near 0}}\]
\[^{\text{used to approximate the } \ell_0 \text{ pseudo-norm}}^{\text{[35], [48]–[50].}}\]
Under Assumptions III.1(i)-(iv), convex quadratic majorants of (15) can be obtained by setting (see [35]):
\[^{\text{resulting posterior distribution is proper.}}^{\text{Hereabove, } \zeta \text{ is a nonnegative constant that can be useful}}\]
to ensure the invertibility of \(Q_1(x)\) for every \(x \in \mathbb{R}^Q\).
The numerical efficiency of the proposed algorithm relies on the use of quadratic majorants that provide tight approximations of the target density but also whose curvature matrices are simple to compute. However, sampling from the proposal constructed by the MM strategy when using the curvature matrix \(Q_1(\cdot)\) given by (17) is often very difficult because of the high computational cost of each iteration and/or memory limitations. In fact, similarly to Newton MCMC samplers, the main computational cost is related to the computation of the inverse of (17) and sampling from the associated high-dimensional Gaussian distribution at each iteration. In the following, we will propose alternative choices of the curvature matrix, when matrix \(Q_1(\cdot)\) given by (17) leads to an intractable scheme. The practical efficiency of the different metric strategies will be analyzed in our experimental part.

Constant curvature matrix: We can resort to the following constant curvature matrix which can be seen as a majorant of (17) constant with respect to variable \(x\):
\[^{\text{It can be noted that in the special case when } H \text{ is circulant}}^{\text{and } \nu > 0, \text{ which is the case for example}}\]
\[^{\text{when } V \text{ is a tight frame analysis operator, then } Q_2 \text{ is easily}}^{\text{invertible in the Fourier domain. More generally, when } H \text{ and}}\]
\[^{\text{and the square root decomposition of (19) can be easily performed}}^{\text{in this basis.}}\]
Diagonal curvature matrix: We also propose the following alternative choice described in [51], which can be understood as a diagonal approximation of \(Q_1\):
\[^{\text{Assumption III.1(i) holds for a large number of data fidelity terms. This includes for example the Gaussian noise model, the}}^{\text{the Huber function which may be useful for limiting the influence}}\]
\[^{\text{of outliers present in some datasets [44], the Cauchy model}}^{\text{as well as the exact Poisson-Gaussian likelihood}}\]
\[^{\text{More examples can be found in [35]. Furthermore, Assumptions III.1(iii)-(iv) are satisfied for several commonly}}^{\text{Assumptions III.1(iii)-(iv) are satisfied for several commonly}}\]
\[^{\text{used prior models such as the Student-t distribution,}}^{\text{used prior models such as the Student-t distribution,}}\]
\[^{\text{as well smoothed approximation of } \ell_p \text{ regularization functions for } p \leq 2 \text{ and } \ell_2 - \ell_0 \text{ penalties}}^{\text{as well smoothed approximation of } \ell_p \text{ regularization functions for } p \leq 2 \text{ and } \ell_2 - \ell_0 \text{ penalties}}\]

\[^{\text{Note that, in this work, improper prior laws are allowed provided that the}}^{\text{Note that, in this work, improper prior laws are allowed provided that the}}\]
[^{\text{resulting posterior distribution is proper.}}^{\text{resulting posterior distribution is proper.}}]
and

\[(\forall i \in \{1, \ldots, P\})(\forall j \in \{1, \ldots, Q\}) \quad P_{i,j} = |V_{i,j}| \sum_{k=1}^{Q} |V_{i,k}|.\]  

(22)

IV. CONVERGENCE ANALYSIS

In this section, we establish the convergence of the proposed algorithm.

It can be first noticed that the drift term in Algorithm 1 is equivalent to

\[(\forall x \in \mathbb{R}^Q) \quad b(x) = \frac{\varepsilon^2}{2} Q^{-1}(x)D(x)\]  

(23)

where \(D(x)\) is the truncated gradient defined by

\[D(x) = \frac{d}{\max(d, \|\nabla \log p(x|z)\|)} \nabla \log p(x|z),\]

provided that the parameter \(d > 0\) tends to \(+\infty\). Similarly to [15], we will study the asymptotic behavior of the algorithm when using the modified drift term (23).

We further make the following assumptions:

Assumption IV.1. \(p(\cdot | z)\) is the density of a super-exponential distribution that is \(p(\cdot | z)\) is positive and has continuous first derivatives such that

\[\lim_{\|x\| \to +\infty} \frac{x^T \nabla \log p(x|z)}{\|x\|} = -\infty,\]  

(25)

and

\[\limsup_{\|x\| \to +\infty} \frac{x^T \nabla \log p(x|z)}{\|x\| \|\nabla \log p(x|z)\|} < 0.\]  

(26)

Assumption IV.2. For every \(x \in \mathbb{R}^Q\), the preconditioning matrix \(Q(x)\) has a bounded spectrum i.e., there exist two constants \(\nu_{\min} > 0\) and \(\nu_{\max} > 0\) independent of \(x\) such that

\[(\forall x \in \mathbb{R}^Q) \quad \nu_{\min} I_Q \preceq Q(x) \preceq \nu_{\max} I_Q.\]  

(27)

Remark IV.1. Assumption IV.2 holds for all the curvature matrices \(Q_1(\cdot)\), \(Q_2(\cdot)\) and \(Q_3(\cdot)\) proposed in Section III-C provided that \(\zeta > 0\). Furthermore, Assumption IV.2 together with (24), imply that the drift term \(b\) is bounded that is

\[(\forall x \in \mathbb{R}^Q) \quad \|b(x)\| \leq \frac{\varepsilon^2}{2} \nu_{\min} d.\]  

(28)

We now state sufficient conditions for Assumption IV.1 to be satisfied.

Proposition IV.1. Consider Model (15) with \(\Phi = \frac{1}{2} \|\cdot\|^2\) and \((\psi_s)_{1 \leq s \leq S}\) satisfying Assumptions III.1(ii)-(iv). Then, Assumption IV.1 is satisfied if one of the following properties holds:

- \(H\) is injective, for example \(H = I_Q\) which is the case for denoising problems;
- there exists \(s_0 \in \{1, \ldots, S\}\) such that
  - (i) \(\text{Ker}(H) \cap \text{Ker}(V_{s_0}) = \{0_Q\}\),
  - (ii) \(\lim_{u \to +\infty} \psi_{s_0}(u)/u > 0\).

\[\square\]

Proof. See Appendix A.

Subsequently, we can establish the geometric ergodicity of the proposed algorithm based on the results concerning RW in [52] and by adapting the proofs in [15], [18], [53]. Since the algorithm appears as a special case of the MH algorithm, the chain \((x^{(t)})_{t \in \mathbb{N}}\) constructed by the 3MH algorithm has \(p(x|z)\) as an invariant distribution. The first important step of the proof of geometric ergodicity is to compare the proposal density \(g\) to Gaussian proposals.

Proposition IV.2. Under Assumption IV.2, there exists \((k_1, k_2, \sigma_1, \sigma_2) \in \{0, +\infty\}^4\) such that for every \((x, y) \in (\mathbb{R}^Q)^2\),

\[k_1 n(y; x, \sigma_1^2 I_Q) \leq g(x) \leq k_2 n(y; x, \sigma_2^2 I_Q)\]

(29)

where \(n(\cdot; x, \sigma_I^2)\), is the density of the Gaussian distribution of mean \(x\) and covariance matrix \(\sigma_I^2 I_Q\), \(i \in \{1, 2\}\).

Proof. See Appendix B.

\[\square\]

Theorem IV.1. Under Assumptions IV.1 and IV.2, the Markov chain defined by the 3MH algorithm using the truncated gradient (24) is geometrically ergodic with stationary distribution \(p_{\mathbb{R}^Q}\).

Proof. From Algorithm 1 and Proposition IV.2, \(g\) is positive and, for every \((x, y) \in (\mathbb{R}^Q)^2\), \(g(x) > 0\). Since \(p(\cdot | z)\) is also positive and continuous, we can deduce from [52] that the chain is aperiodic and \(p(\cdot | z)\)- irreducible with unique invariant distribution \(p(\cdot | z)\).

Assumption IV.1 has been introduced in [52] as a sufficient condition for the geometric ergodicity of the RW algorithm. It has been shown that under Assumption IV.1, when (29) holds, the MALA algorithm with truncated gradient (24) is geometrically ergodic [15]. Since the drift term of the proposed algorithm is bounded by a truncated gradient, the geometric ergodicity property can be deduced by a straightforward adaptation of the proof in [15] for MALA algorithm with truncated drift. Note that our convergence proof is valid for any preconditioned MALA algorithm provided that the preconditioning metric has a bounded spectrum.

\[\square\]

V. EXPERIMENTAL RESULTS

To illustrate the benefits that can be drawn from the proposed algorithm, we will focus on two applicative examples, namely 1D signal deconvolution and multicomponent image denoising.

A. Sparse signal deconvolution with a Student-t prior

Our first example focuses on the deconvolution of a seismic signal. The original signal \(x\) is a sparse vector of length \(Q = 784\) composed of a sequence of spikes called primary reflection coefficients [54], [55] as depicted in Figure 2. The non-zero coefficients give information about the travel time of seismic waves between two seismic reflectors, and the amplitude of the seismic events reflected back to the sensor [54]. We assume that the signal is degraded by a known
In most Bayesian methods, it is generally used in this form:

$$\mu$$

from their posterior joint distribution. In this work, we propose

$$\mu$$

is

Thus, the set of hyperparameters to be estimated jointly with

have only few prior information about the others parameters.

ST

that the

sparsity properties and better preserve discontinuities. Recall

$$\ell$$

of normal distribution where the hidden variable follows a

$$\nu$$

gamma distribution with both parameters equal to

$$\mu$$

is the number of degrees of freedom determining

$$\ell$$

norm and the non-convex approximation of

$$\nu$$

in an approximate manner, using inner conjugate gradient

implementation. Figure 4 shows the error between the original

$$\sigma$$

will test the performance of 3MH using the three different

curvature matrices proposed in Section III, namely

$$Q_1$$, the constant circulant matrix

$$Q_2$$, and the diagonal matrix

$$Q_3$$. In our context, these matrices are defined by

\[
(\forall x \in \mathbb{R}^Q) \quad Q_1(x) = \frac{1}{\sigma^2} H^\top H + \text{Diag}\{\omega(x)\} + \Omega_I, \quad (31)
\]

\[
Q_2 = \frac{1}{\sigma^2} H^\top H + \frac{\nu + 1}{\nu \gamma^2} \Omega_I, \quad (32)
\]

\[
(\forall x \in \mathbb{R}^Q) \quad Q_3(x) = \text{Diag} \left( \frac{1}{\sigma^2} \Omega L^\top 1_N + \omega(x) \right), \quad (33)
\]

where \(\omega(x) = (\omega_i(x))_{1 \leq i \leq Q}\) is such that

\[
(\forall i \in \{1, \ldots, Q\}) \quad \omega_i(x) = \frac{\nu + 1}{\nu \gamma^2 + (x_i - \mu)^2} \quad (34)
\]

and \(L \in \mathbb{R}^{N \times Q}\) is given by (21). Hereabove, \(\Omega \geq 0\) is a constant added to ensure the positive definiteness of the

matrix \(Q_1\). Matrix \(Q_2\) is positive definite for every \(x \in \mathbb{R}^Q\). Furthermore, \(Q_3\) is also ensured to be positive definite for all

\(x \in \mathbb{R}^Q\) provided that the observation matrix \(H\) contains no column whose elements are all equal to zero. It is worth noting

that, the posterior density satisfies the sufficient conditions in Proposition IV.1 when \(H\) is injective. Similarly to MALA

[18], the geometric ergodicity of 3MH is not theoretically guaranteed if \(H\) is not injective.

The posterior laws of the \(ST\) prior parameters do not have usual forms. Then, it is not easy to directly generate samples of

\(\mu\) and \(\gamma\). We propose therefore to estimate them using a RW algorithm whose scale parameter is tuned automatically during

the burn-in period so as to reach an acceptance probability equal to 0.33.

3) Results: The test signal is artificially degraded by a band-pass filter with finite impulse response of length 41 with a

frequency band concentrated between 10 and 40 Hz and an additive Gaussian noise of variance \(\sigma^2 = 2.5 \times 10^{-3}\) (see Figure 2). The initial signal-to-noise ratio (SNR) is

\(-4.58\) dB. We fix \(\nu = 1\) which corresponds to the special case of the Cauchy prior. Simulations are performed on an

Intel(R) Xeon(R) CPU E5-2630, @ 2.40 GHz, using a Matlab7 implementation. Figure 4 shows the error between the original

signal and the degraded one as well as the error between the original signal and the restored one using the Minimum Mean

Square Estimator (MMSE) which corresponds to a SNR equal to 8.24 dB.

We propose to compare the 3MH algorithm using the different curvatures matrices \(Q_1\), \(Q_2\), and \(Q_3\) and the standard

MALA algorithm. All tested algorithms have been run until convergence. The discretization stepsize \(\varepsilon\) is adjusted for all

these algorithms during the burn-in period to correspond to an acceptance probability between 0.3 and 0.6. Note that in order

to reduce the complexity of each iteration when using

\(Q = Q_1\), the inversion of the curvature matrix is performed in an approximate manner, using inner conjugate gradient

iterations and the generation of random variables according to the proposal is then ensured using the sampling method

from [61]. Figure 5 shows the evolution of \(J\) with respect to time. Following [15], we also compare the different methods

1) Prior and posterior distributions: In order to promote the signal sparsity, we suppose that its coefficients are independent

and identically distributed according to a Student-t (ST) distribution. Then, we can identify the following functions:

\[
\Phi = \frac{1}{2 \sigma^2} \| \cdot \|^2 \quad \text{and} \quad \Psi(x) = \frac{\nu + 1}{2} \sum_{i=1}^{Q} \log \left( \gamma^2 + \frac{(x_i - \mu)^2}{\nu} \right) \quad (30)
\]

where \(\nu > 0\) is the number of degrees of freedom determining the shape of the distribution, \(\mu\) is the position parameter, and

\(\gamma > 0\) is the scale parameter [56]. Note that the Cauchy distribution is recovered as a particular case when \(\nu = 1\).

The ST distribution is often used in image reconstruction to model the distribution of wavelet coefficients [57]. This

penalty has also been proposed in [58] as a tradeoff between the squared \(\ell_2\) norm and the non-convex approximation of

the semi-norm \(\ell_0\) presented in [59], with the aim to enforce sparsity properties and better preserve discontinuities. Recall

that the ST distribution can be written as a scale mixture of normal distribution where the hidden variable follows a

gamma distribution with both parameters equal to \(\nu/2\) [60]. In most Bayesian methods, it is generally used in this form: the unknown signal \(x\) and the hidden variable are estimated from their posterior joint distribution. In this work, we propose to directly use the expression defined in (30).

In the following, we assume that \(\nu\) is known, and that we have only few prior information about the others parameters.

Thus, the set of hyperparameters to be estimated jointly with \(x\) is \(\Theta = \{\mu, \gamma\}\). More specifically, uniform distributions are used for

\(\mu\) and \(\gamma\) defined on \([-\mu_m, \mu_M]\) and \([\gamma_m, \gamma_M]\) respectively, where \(\mu_m, \mu_M, \gamma_m\) and \(\gamma_M\) are positive constants.

Thus, the posterior distributions of the parameters are given by

\[
p(\mu | x, z, \gamma) \propto \prod_{i=1}^{Q} \left( \gamma^2 + \frac{(x_i - \mu)^2}{\nu} \right)^{-\frac{\nu + 1}{2}} 1_{[-\mu_m, \mu_M]}(\mu),
\]

\[
p(\gamma | x, z, \mu) \propto \gamma^{Q\nu} \prod_{i=1}^{Q} \left( \gamma^2 + \frac{(x_i - \mu)^2}{\nu} \right)^{-\frac{\nu + 1}{2}} 1_{[\gamma_m, \gamma_M]}(\gamma).
\]

2) Sampling from the posterior distribution of the signal and the hyperparameters: \(\Phi\) and \(\Psi\) satisfy the properties in

Section III-C. We can thus apply the 3MH algorithm to sample from the posterior distribution of \(x\). More specifically, we
in terms of the Mean Square Jump (MSJ) at stationarity which indicates how much the Markov chain is exploring the whole target space after convergence. Note that MSJ has been estimated with an empirical average over $T = 5000$ samples $x^{(t_0+1)}, \ldots, x^{(t_0+T)}$ generated after the burn-in period as follows

$$MSJ = \left( \frac{1}{T-1} \sum_{t=1}^{T-1} \|x^{(t_0+t)} - x^{(t_0+t+1)}\|^2 \right)^{1/2}.$$  \hspace{1cm} (35)

In Table I, we show estimates of the mean square jump per second at stationarity which is defined as the ratio of the mean square jump and the computational time per iteration. We also compare the statistical efficiency of the different samplers with respect to MALA defined as the mean square jump per second for each sampler over the mean square jump per second of MALA.

| TABLE I: Mixing results for the different algorithms. First row: Estimates of the mean square jump at stationarity. Second row: Estimates of the mean square jump per second in stationarity. Third row: Efficiency relatively to MALA. |

<table>
<thead>
<tr>
<th></th>
<th>MALA</th>
<th>3MH - Q1</th>
<th>3MH - Q2</th>
<th>3MH - Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSJ</td>
<td>$1.40 \times 10^{-5}$</td>
<td>$8.14 \times 10^{-5}$</td>
<td>$1.39 \times 10^{-5}$</td>
<td>$2.32 \times 10^{-5}$</td>
</tr>
<tr>
<td>MSJ per s.</td>
<td>$3.60 \times 10^{-2}$</td>
<td>$8.65 \times 10^{-4}$</td>
<td>$1.17 \times 10^{-2}$</td>
<td>$3.89 \times 10^{-2}$</td>
</tr>
<tr>
<td>Efficiency</td>
<td>1</td>
<td>0.02</td>
<td>0.32</td>
<td>1.08</td>
</tr>
</tbody>
</table>

One can notice that the behavior of 3MH algorithm using the constant curvature matrix $Q_2$ is close to MALA in terms of convergence speed. This fact can be explained by the low dispersion of the eigenvalues of $Q_2$ in this particular example. Nevertheless, the use of the matrix $Q_1$ at each iteration becomes more expensive as the problem dimension increases which deteriorates the efficiency of the algorithm. The choice of the diagonal adaptive matrix $Q_3$ appears to outperform the other algorithms due to the low complexity that it induces at each iteration. It allows to reach stability much faster than the other algorithms while achieving mixing properties slightly better than MALA at convergence.

**B. Multispectral image denoising with a multivariate prior**

In our second example, we are interested in the denoising of a multispectral image with $R$ pixels and $B$ spectral channels corrupted with independent additive white Gaussian noises $\mathcal{N}(0, \sigma^2)$. We assume that the noise variance $\sigma^2$ is known. We denote by $\bar{y}_1, \ldots, \bar{y}_B$ the vectors that correspond to the reshaped unknown spectral images into vectors in $\mathbb{R}^R$. The objective is to recover these vectors from observed noisy vectors $\bar{z}_1, \ldots, \bar{z}_B$. Our recovery procedure will operate in the wavelet transform domain since the wavelet representations of the $B$ unknown spectral images are sparse. To this end, we choose a set of orthogonal wavelet synthesis operators $F_1, \ldots, F_B$ [62]. More precisely, for each spectral position $b \in \{1, \ldots, B\}$, $F_b$ is a linear mapping from $\mathbb{R}^K$ to $\mathbb{R}^R$ with $K \geq R$ that outputs $\tilde{y}_b$ from the vector $\bar{x}_b \in \mathbb{R}^K$ of frame coefficients:

$$\tilde{y}_b = F_b^* \bar{x}_b.$$  \hspace{1cm} (36)

Each spectral component with index $b$ is decomposed into $M$ subbands with sizes $K_m$, $m \in \{1, \ldots, M\}$ according to different orientations and resolutions. Obviously, we have $\sum_{m=1}^{M} K_m = K$ and the vector $\bar{x}_b$ is defined by

$$\bar{x}_b = \begin{pmatrix} \bar{x}_{b,1,1} & \ldots & \bar{x}_{b,1,K_1} & \ldots & \bar{x}_{b,m,1} & \ldots & \bar{x}_{b,m,K_m} & \ldots & \bar{x}_{b,M,1} & \ldots & \bar{x}_{b,M,K_M} \end{pmatrix}^\top.$$  \hspace{1cm} (37)

Thus, the problem of recovering the multispectral image can be viewed as a special case of (1) expressed by $\bar{z} = Hx + w$, where $N = Q = KB$, $z = [\bar{z}_1, \ldots, \bar{z}_B]^\top \in \mathbb{R}^N$, $x =$ Fig. 5: Convergence speed of MALA, 3MH - Q1, 3MH - Q2 and 3MH - Q3.
\[ \begin{pmatrix} \mathbf{x}_1^T, \ldots, \mathbf{x}_B^T \end{pmatrix}^T \in \mathbb{R}^Q, \; \mathbf{w} \sim \mathcal{N}(0_N, \sigma^2 \mathbf{I}_N) \text{ and the matrix } \mathbf{H} \text{ is given by} \]
\[
\mathbf{H} = \begin{pmatrix} \mathbf{F}_1^0 & 0 & \ldots & 0 \\ 0 & \mathbf{F}_2^0 & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{F}_B^0 \end{pmatrix}. \tag{38}
\]

Our objective is to build an estimate \( \hat{\mathbf{x}} \) of the frame coefficients \( \mathbf{x} \) based on the available observed frame coefficients \( \mathbf{z} \) and the transform domain matrix \( \mathbf{H} \).

1) Prior and posterior distributions: It is worth noting that the mutual similarities between the spectral images also propagate to their corresponding frame coefficients. Our idea is to capture such dependencies by resorting to a joint estimation of the frame coefficients of all the \( B \) components at a given orientation and resolution. To this end, for each subband \( m \in \{1, \ldots, M\} \), we stack the coefficients of all the \( B \) channels at the same spatial position \( k \in \{1, \ldots, K_m\} \) so as to build the vectors \( \mathbf{x}_{m,k} = (\mathbf{x}_{m,k,m})_{k \in \{1, \ldots, B\}} \in \mathbb{R}^B \). Mathematically, it is easy to show that these vectors result from a linear transform of \( \mathbf{x} \): \( \mathbf{x}_{m,k} = \mathbf{P}_{m,k} \mathbf{x} \), where \( \mathbf{P}_{m,k} \in \mathbb{R}^{B \times Q} \) denotes a sparse matrix containing \( B \) lines of an appropriate permutation matrix.

The sparsity of the frame coefficients and the spectral redundancies are captured by assuming that for each orientation and scale associated to index \( m \), the vectors \( \mathbf{x}_{m,1}, \ldots, \mathbf{x}_{m,K_m} \) correspond to \( K_m \) realizations of a random vector whose distribution is a generalized multivariate exponential power distribution (GMEP) characterized by its scale matrix \( \mathbf{\Sigma}_m \), its shape parameter \( \beta_m \) and its smoothing parameter \( \delta_m \) [63]. Consequently, the likelihood function \( \Phi \) can be defined similarly to (30). Furthermore, the minus-log of the prior density is easily derived (up to an additive constant):
\[
\Psi(\mathbf{x}) = \sum_{m=1}^{M} \sum_{k=1}^{K_m} \psi_m(\|\mathbf{\Sigma}_m^{-1/2}(\mathbf{P}_{m,k} \mathbf{x} - \mathbf{a}_m)\|) \tag{39}
\]
where, for every \( m \in \{1, \ldots, M\} \), \( \mathbf{a}_m \) is a vector of \( \mathbb{R}^B \) and for every real \( t \), \( \psi_m(t) = \frac{1}{2} (t^2 + \delta_m)^{\beta_m} \). It should be pointed out that for every \( m \in \{1, \ldots, M\} \), the shape of the GMEP is controlled by \( \beta_m \). Indeed, heavier tailed distributions than the Laplace one correspond to \( \beta_m < 0.5 \) whereas Gaussian-like ones are associated to \( \beta_m \) close to 1. In our work, for the sake of simplicity, the values of every \( \beta_m \) and \( \delta_m \) are assumed to be known. However, it has to be emphasized that these values may change from a subband to another. Very small values (\( \beta_m < 0.5 \)) are assigned at the finest scales in order to promote the frame coefficients sparsity but relatively higher values (\( 0.5 < \beta_m < 1 \)) are chosen at higher scales. Finally, a normal distribution is often retained for the approximation subband at the coarsest scale. The value of \( \delta_m \) is adjusted to a positive small value to guarantee the differentiability of \( \Psi \) given in (39). Furthermore, we need to decompose the scale matrix, for every \( m \in \{1, \ldots, M\} \) as follows:
\[
\mathbf{\Sigma}_m = \gamma_m^{-1/(2\beta_m)} \mathbf{Diag}(\mathbf{n}_m)^{-1} \mathbf{R}_m \mathbf{Diag}(\mathbf{n}_m)^{-1}, \tag{40}
\]
where \( \mathbf{R}_m \) is the normalized correlation matrix of size \( B \times B \) (with diagonal elements equal to 1 and the remaining ones correspond to the correlation factors between the coefficients), \( \mathbf{n}_m \) is a \( B \)-dimensional vector of positive elements whose sum is equal to 1 and \( \gamma_m \) is a positive real. It is worth noting that \( \gamma_m^{-1/(2\beta_m)} \mathbf{n}_m \) can be seen as the vector containing the square root of the scale parameters for all the \( B \) components in subband \( m \). For the sake of simplicity, we assume without loss of generality that the different spectral components of the transform domain matrix \( \mathbf{H} \) have the same correlation and weights in all subbands i.e., \( \mathbf{R} = \mathbf{R}_m \) and \( \mathbf{n}_m = \mathbf{n} \) for all \( m \). Moreover, \( \mathbf{n} \) and \( \mathbf{R} \) are assumed to be known. In this way, all the unknown hyperparameters form the set:
\[
\Theta = \{\gamma_1, \ldots, \gamma_M\}. \tag{41}
\]

For every \( m \in \{1, \ldots, M\} \), a gamma prior for \( \gamma_m \) is selected: \( \gamma_m \sim \mathcal{G}(a_m, b_m) \) where \( a_m > 0 \) and \( b_m > 0 \) [64]. Consequently, \( \gamma_m \) has the following posterior distribution:
\[
\begin{aligned}
&\text{p}(\gamma_m | \mathbf{x}, \mathbf{R}) \propto \gamma_m^{a_m + 1} \beta_m^{\Sigma_m^{-1}} \exp \left( -b_m \gamma_m \right) \\
&\times \exp \left( \frac{1}{2} \sum_{k=1}^{K_m} \left( \gamma_m^{\Sigma_m^{-1}} \mathbf{R}^{-1} \mathbf{Diag}(\mathbf{n}) \mathbf{P}_{m,k} \mathbf{x} \right)^2 \right). \tag{42}
\end{aligned}
\]

Our goal is to compute the posterior mean estimates of the target frame coefficients \( \mathbf{x} \) as well as of \( \Theta \) thanks to MCMC sampling algorithms.

2) Sampling from the posterior distribution of the image and hyperparameters: Samples of vectors \( \mathbf{x}_{m,k} \) can be drawn in an independent manner for every \( m \in \{1, \ldots, M\} \) and \( k \in \{1, \ldots, K_m\} \). Indeed, as the posterior law is differentiable, we propose to apply Langevin based MCMC algorithms to produce samples according to the posterior law of \( \mathbf{x}_{m,k} \). Furthermore, adding a curvature matrix that accounts for the cross-spectral dependencies can improve the sampling performance. Note that its Hessian and Fisher matrices are equal because of the Gaussianity of the fidelity term. Nevertheless, the convexity of \( \psi_m \) only holds when \( \beta_m \geq 0.5 \) and, hence there is no guarantee that the Hessian and the Fisher matrices are definite positive if \( \beta_m < 0.5 \). For every \( m \in \{1, \ldots, M\} \), \( \psi_m \) is differentiable and, the concavity on \( \mathbb{R}_+ \) of the function \( t \mapsto \psi_m(\sqrt{t}) \) is valid when \( \beta_m \leq 1 \). Consequently, we propose to employ the curvature matrices built by the MM strategy described in Section III-C. More precisely, we make use of the curvature matrix introduced in (17). Its expression for each subband \( m \) is given by
\[
\forall \mathbf{c} \in \mathbb{R}^B \quad \mathbf{Q}_m^{(m)}(\mathbf{c}) = \frac{1}{\sigma^2} \mathbf{I}_B + \mathbf{\Sigma}_m^{-1} \psi_m \left( \|\mathbf{\Sigma}_m^{1/2}(\mathbf{c} - \mathbf{a}_m)\| \right). \tag{43}
\]
Note that, the geometric ergodicity of the 3MH algorithm is fulfilled as \( \mathbf{H} \) is injective. It is also worth pointing out that in the case when a normal distribution (i.e., \( \beta_m = 1 \)) is assigned to the low frequency subband, it can be proved that the 3MH algorithm is still geometrically ergodic for a deconvolution problem (non necessarily injective \( \mathbf{H} \)) because \( (\psi_m)_{1 \leq m \leq M} \) satisfy the assumptions of Proposition IV.1. Because of the unusual form of the posterior law of \( \Theta \), sampling from (42) is carried out by an independent MH
algorithm with a gamma proposal of parameters $\tilde{a}_{\gamma_m} = a_{\gamma_m} + K_m/(2\beta_m)$, and
\[
\tilde{b}_{\gamma_m} = b_{\gamma_m} + \sum_{k}^{K_m} ||R^{-\frac{1}{2}}\text{Diag}(\mathbf{n})(P_{m,k}\mathbf{x} - a_m)||^{2\beta_m}.
\] (44)

3) Results: In our experiments, we select the Hydice\footnote{https://engineering.purdue.edu/ biehl/MultiSpec/hyperspectral.html} hyperspectral dataset containing 191 spectral components in the range [0.4,2.4]μm of the visible and infrared spectrum. From this dataset, we extract a portion of size $R = 256 \times 256$ over $B = 60$ spectral channels that we consider as our test image. Thus, the problem dimension amounts to $N = 3,932,160$. We add a zero-mean white Gaussian noise with variance $\sigma^2 = 225$ to this test image. The initial SNR is 12.15 dB. We apply to the noisy image a three-stage orthonormal wavelet decomposition using a Symlet wavelet of order 3. Hence, we have $M = 10$ and $Q = N$. Regarding the approximation coefficients ($m = M$), we retain a Gaussian prior ($\beta_M = 1$, $\delta_M = 0$). For all the remaining subbands ($m \in {1,\ldots,M-1}$), we choose $\delta_m = 10^{-4}$. Moreover, we set $\beta_m = 0.2$ for the wavelet coefficients at the lowest resolution level, $\beta_m = 0.4$ at the second level of decomposition, and $\beta_m = 0.5$ at the coarsest level of decomposition. The Gibbs sampling algorithm is run with enough iterations to reach the stability state. The empirical MMSE estimator for the original image is computed with the generated samples of the wavelet coefficients after the burn-in period. Figure 6 displays the results achieved for the various components in terms of SNR and SSIM (Structural SIMilarity [65]). It appears that our method leads to a dramatic improvement of the values of the objective metrics and the perceptual ones for all the spectral components. For example, the average increase of the SNR (resp. SSIM) values approximately amounts to 10 dB (resp. 0.3). The resulting gains tend to indicate that the MMSE estimator leads to good numerical results. This is also corroborated by a visual inspection of the recovered components. Indeed, the reduction of the noise degradation in the different components is clearly noticeable in Figure 7. Besides, small details have been enhanced in a satisfactory manner.

As benchmarking, we compare the performance of the Gibbs sampler when the posterior law of the wavelet coefficients is explored using either RW, MALA, or 3MH algorithms. Simulations were performed on an Intel(R) Xeon(R) CPU E5-2630, @ 2.40 GHz, using a Matlab\footnote{https://engineering.purdue.edu/ biehl/MultiSpec/hyperspectral.html} implementation. Figure 8 illustrates the evolution of the scale parameter $\gamma_m$ in the horizontal subband at the first level of decomposition with respect to the computational time employing the underlying algorithms. Table II provides their related mixing results in terms of mean square jump per second in stationarity. It can be noticed that the stationary state is reached by our novel algorithm much faster than by the RW and MALA algorithms. More precisely, about 500 seconds are enough for the 3MH algorithm to reach stability which is fourfold less than the time required by MALA algorithm. Furthermore, the RW algorithm appears as the slowest algorithm since it needs about 10,000 seconds to converge. Moreover, it is worth emphasizing that the 3MH algorithm presents the best mixing properties at stability in terms of MSJ. Thus, our proposed method also is the most efficient choice after reaching stability compared with MALA and RW even though the computational load of a single iteration of the 3MH algorithm is around twice higher than that of the RW.

Another appealing property of the proposed Gibbs sampler concerns its straightforward extension to the case of a deconvolution problem corresponding to the case of $H = DF^*$, the matrix $D$ being a blurring operator. This extension can be realized by inserting an additional step in the Gibbs algorithm.
We deduce that

\[ \| \nabla J(x) \| \leq \| H \| \| Hx - z \| + \sum_{s=1}^{S} \| V_s \| \psi_s(\| V_s x - c_s \|) \]

where the second inequality stems from Assumption III.1(iv). It follows from (45) that

\[ x^T \nabla J(x) = \| Hx \|^2 + \sum_{s=1}^{S} \| V_s x \|^2 \frac{\psi_s(\| V_s x - c_s \|)}{\| V_s x - c_s \|} + h(x) \]

where

\[ h(x) = -x^T \left( H^T z + \sum_{s=1}^{S} V_s^T c_s \frac{\psi_s(\| V_s x - c_s \|)}{\| V_s x - c_s \|} \right). \]

Assume that \( H \) is injective. According to (48) and using Assumption III.1(iv), we obtain

\[ \frac{x^T \nabla J(x)}{\| x \|} \geq \| Hx \|^2 + h(x) \]

\[ \| x \| (\| Hx \| + \| z \|) + \sum_{s=1}^{S} \omega_s \| V_s \| (\| V_s x \| + \| c_s \|) \]

\[ \geq \frac{\| Hx \|^2}{\| H \|^2 + \| V_s \|^2} (1 + o(1)) + o(1). \]

Thus, (26) also holds, and so does Assumption IV.1.

Let us now consider the case when \( H \) is not injective and our second set of assumptions applies. According to (48), we have

\[ \frac{x^T \nabla J(x)}{\| x \|} \geq \| Hx \|^2 + \alpha_{s_0} \| V_{s_0} x \|^2 + O(1). \]

According to Assumption III.1(iii), \( u \mapsto \psi_{s_0}(u) / u \) is decreasing on \( [0, +\infty) \) and, by using Assumption (ii), we deduce that there exists \( \alpha_{s_0} > 0 \) such that \((\forall u \in [0, +\infty)) \psi_{s_0}(u) / u \geq \alpha_{s_0} \). Then, (52) yields

\[ \frac{x^T \nabla J(x)}{\| x \|} \geq \| Hx \|^2 + \alpha_{s_0} \| V_{s_0} x \|^2 + O(1). \]

It then follows from Assumption (i) that (25) is satisfied. Moreover, using (46) and (47), we have

\[ \frac{x^T \nabla J(x)}{\| x \|} \geq \| Hx \|^2 + \alpha_{s_0} \| V_{s_0} x \|^2 + O(1). \]

TABLE II: Results for the different proposed algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSJ</th>
<th>MSJ per s.</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>RW</td>
<td>5.95</td>
<td>9.15</td>
<td>1</td>
</tr>
<tr>
<td>MALA</td>
<td>10.65</td>
<td>9.95</td>
<td>1.08</td>
</tr>
<tr>
<td>3MH</td>
<td>15.52</td>
<td>12.31</td>
<td>1.34</td>
</tr>
</tbody>
</table>

to draw samples of auxiliary variables [66].

VI. CONCLUSION

In this work, we have proposed a new MCMC algorithm that can be considered as a scaled MALA where the scale matrix is adapted at each iteration by following an MM strategy. We have shown that the geometric ergodicity property of the standard Langevin MH algorithms is maintained by introducing this scale matrix for the class of super-exponential distributions. We have then applied this algorithm to compute the MMSE estimator in signal and multicomponent image recovery problems. Experimental results emphasize the good performance of this new MCMC method compared to the standard MALA algorithm.

APPENDIX A

PROOF OF PROPOSITION IV.1

Let \( x \in \mathbb{R}^Q \setminus \{0\} \). According to Assumption (ii), we have

\[ \nabla J(x) = H^T (Hx - z) + \sum_{s=1}^{S} V_s^T (V_s x - c_s) \frac{\psi_s(\| V_s x - c_s \|)}{\| V_s x - c_s \|} \]

We deduce that

\[ \| \nabla J(x) \| \leq \| H \| \| Hx - z \| + \sum_{s=1}^{S} \| V_s \| \psi_s(\| V_s x - c_s \|) \]

\[ \leq \| H \| (\| Hx \| + \| z \|) + \sum_{s=1}^{S} \omega_s \| V_s \| (\| V_s x \| + \| c_s \|), \]

\[ \text{The spectral norm is employed for matrices.} \]

First row: Estimates of the mean square jump in stationarity. Second row: Estimates of the mean square jump per second in stationarity. Third row: Relative efficiency compared to RW.
Similarly, we have
\[
\|y - \mu(x)\|^2 \leq 2 \left( \|y - x\|^2 + \frac{4}{4} \nu_{\min}^{-2} d^2 \right).
\] (67)

It follows from (62), (65) and (67) that
\[
\nu_{\min} \frac{\|y - x\|^2}{4} = \frac{\|y - \mu(x)\|^2}{4 \nu_{\max}} \leq \frac{1}{2} \nu_{\min} \|y - x\|^2 + \frac{\nu_{\max} d^2}{4 \nu_{\min}} - \frac{\nu_{\min} d^2}{4 \nu_{\max}}.
\] (68)

Then, by using (61), (63) and (68), the lower bound in (29) holds for
\[
k_1 = \left( \frac{\nu_{\min}}{2 \nu_{\max}} \right)^{\frac{Q}{2}} \exp \left( -\frac{\varepsilon^2 \nu_{\max} d^2}{8 \nu_{\min}} \right), \quad \sigma_1^2 = \frac{\varepsilon^2}{2 \nu_{\max}}.
\]

and
\[
\nu_{\min} \|y - \mu(x)\|^2 \leq \|y - \mu(x)\|^2_{Q(x)} \leq \nu_{\max} \|y - \mu(x)\|^2,\quad \text{and}
\] (62)

and
\[
\nu_{\min} \|y - \mu(x)\|^2 \leq \|y - \mu(x)\|^2_{Q(x)} \leq \nu_{\max} \|y - \mu(x)\|^2,\quad \text{and}
\] (63)

On the other hand, by using (28) and the triangle inequality, we have
\[
\|y - x\| \leq \|y - \mu(x)\| + \|\mu(x) - x\|,
\] (64)

By using Jensen’s inequality, it follows that
\[
\|y - x\|^2 \leq 2 \left( \|y - \mu(x)\|^2 + \frac{4}{4} \nu_{\min}^{-2} d^2 \right).
\] (65)

Similarly, we have
\[
\|y - \mu(x)\| \leq \|y - x\| + \|\mu(x) - x\|,
\] (66)

REFERENCES


