JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2021, VOL. 00, NO. 0, 1–19: Theory and Methods
https://doi.org/10.1080/01621459.2021.1982723

Taylor & Francis
Taylor & Francis Group

Hamiltonian-Assisted Metropolis Sampling

Zexi Song and Zhiqiang Tan

Department of Statistics, Rutgers University, Piscataway, NJ

ABSTRACT

Various Markov chain Monte Carlo (MCMC) methods are studied to improve upon random walk Metropolis sampling, for simulation from complex distributions. Examples include Metropolis-adjusted Langevin algorithms, Hamiltonian Monte Carlo, and other algorithms related to underdamped Langevin dynamics. We propose a broad class of irreversible sampling algorithms, called Hamiltonian-assisted Metropolis sampling (HAMS), and develop two specific algorithms with appropriate tuning and preconditioning strategies. Our HAMS algorithms are designed to simultaneously achieve two distinctive properties, while using an augmented target density with a momentum as an auxiliary variable. One is generalized detailed balance, which induces an irreversible exploration of the target. The other is a rejection-free property for a Gaussian target with a prespecified variance matrix. This property allows our preconditioned algorithms to perform satisfactorily with relatively large step sizes. Furthermore, we formulate a framework of generalized Metropolis–Hastings sampling, which not only highlights our construction of HAMS at a more abstract level, but also facilitates possible further development of irreversible MCMC algorithms. We present several numerical experiments, where the proposed algorithms consistently yield superior results among existing algorithms using the same preconditioning schemes.

ARTICLE HISTORY
Received April 2020
Accepted September 2021

KEYWORDS
Auxiliary variables; Detailed balance; Hamiltonian Monte Carlo; Markov chain Monte Carlo; Metropolis-adjusted Langevin algorithms; Metropolis–Hastings sampling; Underdamped Langevin dynamics

1. Introduction

In various statistical applications, it is desired to generate observations from a probability density $\pi(x)$, referred to as the target distribution. The density function $\pi(x)$ is often defined such that an unnormalized density function $\tilde{\pi}(x) \propto \pi(x)$ can be readily evaluated, but the normalizing constant $\int \tilde{\pi}(x)\,\mathrm{d}x$ is intractable due to high-dimensional integration. A prototypical example is posterior sampling for Bayesian analysis, where the product of the likelihood and prior is an unnormalized posterior density. For such sampling tasks, a useful methodology is Markov chain Monte Carlo (MCMC), where a Markov chain is simulated such that the associated stationary distribution coincides with the target $\pi(x)$. Under ergodic conditions, observations from the Markov chain can be considered an approximate sample from $\pi(x)$. See, for example, Liu (2001) and Brooks et al. (2011).

One of the main workhorses in MCMC is Metropolis–Hastings sampling (Metropolis et al. 1953; Hastings 1970). Given the current variable x_0, the Metropolis–Hastings algorithm generates x^* from a proposal density $x^* \sim Q(x^*|x_0)$, and then accepts $x_1 = x^*$ as the next variable with probability

$$\rho(x^*|x_0) = \min\left\{1, \frac{\pi(x^*)Q(x_0|x^*)}{\pi(x_0)Q(x^*|x_0)}\right\}, \tag{1}$$

or rejects x^* and set $x_1 = x_0$, where $\pi(x^*)/\pi(x_0)$ can be evaluated as $\tilde{\pi}(x^*)/\tilde{\pi}(x_0)$ without requiring the normalizing constant. The update from x_0 to x_1 defines a Markov transition $K(x_1|x_0)$, depending on both the proposal density and the acceptance-rejection step, such that reversibility is satisfied: $\pi(x_0)K(x_1|x_0) = \pi(x_1)K(x_0|x_1)$. This condition is also called detailed balance, originally in physics. As a result, the Markov chain defined by the transition kernel K is reversible and admits $\pi(x)$ as a stationary distribution.

The Metropolis–Hastings algorithm is flexible in allowing various choices of the proposal density Q. A simple choice, known as random walk Metropolis (RWM), is to add a Gaussian noise to x_0 for generating x^*. However, RWM may perform poorly for sampling from complex distributions. To tackle this issue, various MCMC methods are developed by exploiting gradient information in the target density $\pi(x)$. A common approach is to use discretizations of physics-based continuous-time dynamics as proposal schemes, while staying within the framework of Metropolis–Hastings sampling. One group of algorithms include preconditioned Metropolis-adjusted Langevin algorithm (pMALA) (Besag 1994; Roberts and Tweedie 1996) and preconditioned Crank–Nicolson Langevin (pCNL) (Cotter et al. 2013), related to (overdamped) Langevin diffusion. Another popular algorithm is Hamiltonian Monte Carlo (HMC), which introduces a momentum variable and uses a leapfrog discretization of the deterministic Hamiltonian dynamics as the proposal scheme combined with momentum resampling (Duane et al. 1987; Neal 2011). A subtle point is that the momentum can be artificially negated at the end of leapfrog to ensure reversibility.

CONTACT Zhiqiang Tan ztan@stat.rutgers.edu Department of Statistics, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

There are also various MCMC methods, designed by simulating irreversible Markov chains which converge to the target distribution. One group of algorithms include guided Monte Carlo (GMC) (Horowitz 1991; Ottobre et al. 2016) and the underdamped Langevin sampler (UDL) (Bussi and Parrinello 2007), related to the underdamped Langevin dynamics. Another group of algorithms includes irreversible MALA (Ma et al. 2018) and nonreversible parallel tempering (Syed et al. 2019), related to lifting with a binary auxiliary variable (Gustafson 1998; Vucelja 2016). A third group of algorithms involve careful construction of nonreversible Markov updates (Suwa and Todo 2012) or continuous-time Markov processes (Ohzeki and Ichiki 2015; Duncan, Pavliotis, and Zygalakis 2017), without introducing auxiliary variables. A fourth group of algorithms includes the bouncy particle (Bouchard-Cote, Vollmer, and Doucet 2018) and Zig–Zag samplers (Bierkens, Fearnhead, and Roberts 2019), using Poisson jump processes.

The contribution of this article can be summarized as follows. First, we propose a broad class of irreversible sampling algorithms, called HAMS, and develop two specific algorithms, HAMS-A/B, with appropriate tuning and preconditioning strategies. Our HAMS algorithms use an augmented target density (corresponding to a Hamiltonian) with a momentum as an auxiliary variable. Each iteration of HAMS consists of a proposal step depending on the gradient of the Hamiltonian, and an acceptance–rejection step using an acceptance probability different from the usual formula (1). The two steps are designed to achieve generalized detailed balance and a rejection-free property for Gaussian targets discussed below. Second, we formulate a framework of generalized Metropolis–Hastings sampling, which not only highlights our construction of HAMS as a special case, but also facilitates possible further development of irreversible MCMC algorithms. Third, we present several numerical experiments, where the proposed algorithms consistently yield superior results among existing ones.

Compared with existing algorithms, there are two important properties which are *simultaneously* satisfied by our HAMS algorithms. The first is generalized detailed balance (or generalized reversibility), where the backward transition is related to the forward transition after negating the momentum. This condition is known in the study of continuous dynamics in physics (Gardiner 1997), where the acceptance–rejection step is often ignored with small step sizes, but is in general crucial for proper sampling from a target distribution (see Section 4). By generalized detailed balance, the momentum can be accepted without sign negation, which induces an irreversible exploration of the target. Second, our algorithms satisfy a rejection-free property (i.e., the proposal is always accepted) when the target distribution is standard Gaussian. By preconditioning, the rejection-free property can also be satisfied when the target distribution is Gaussian with a prespecified variance. For clarity, our algorithms are said to satisfy the Gaussian-calibrated rejection-free property. A similar motivation can be found in the construction of the (reversible) pCNL algorithm (Cotter et al. 2013). From our experiments, this property, combined with irreversibility, allows our algorithms to achieve superior results with relatively large step sizes, compared with existing algorithms using the same preconditioning schemes. We expect that the HAMS algorithms can perform well in a broad range of problems, especially when the target distribution roughly resembles standard Gaussian after preconditioning.

Notation. Assume that a target density $\pi(x)$ is defined on $\mathbb{R}^k$. The potential energy function $U(x)$ is defined such that $\pi(x) \propto \exp\{-U(x)\}$ as in physics. Denote the gradient of $U(x)$ as $\nabla U(x)$. The normal (or Gaussian) distribution with mean μ and variance V is denoted as $\mathcal{N}(\mu, V)$, and the density function as $\mathcal{N}(\cdot|\mu, V)$. Whenever possible, we treat a probability distribution and its density function interchangeably. Write $\mathbf{0}$ for a vector or matrix with all 0 entries, and I for an identity matrix of appropriate dimensions.

2. Related Methods

We describe several MCMC algorithms, related to our work, for sampling from a target distribution $\pi(x)$. Throughout, we write the current variable as x_0, a proposal as x^*, and the next variable as x_1 after the acceptance-rejection step. Denote as Σ a constant variance matrix used as an approximation to the variance of the target $\pi(x)$.

Random walk Metropolis (RWM) sampling generates a proposal x^* by directly adding a Gaussian noise to x_0 and then performs acceptance or rejection.

RWM.

- Generate $x^* = x_0 + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\epsilon > 0$ is a tunable step size.
- Set $x_1 = x^*$ with acceptance probability $\rho(x^*|x_0) = \min(1, \pi(x^*)/\pi(x_0))$ by (1),
  or set $x_1 = x_0$ with the remaining probability.

RWM does not exploit gradient information, and may be slow in exploring the target $\pi(x)$. On the other hand, RWM is operationally low-cost, without gradient evaluation.

The preconditioned Metropolis-adjusted Langevin algorithm (pMALA) generates a proposal x^* by moving along the gradient from current x_0 (Roberts and Tweedie 1996). Hence, pMALA is more directed and encourages exploration to high density regions.

pMALA.

- Generate $x^* = x_0 - \frac{\epsilon^2}{2}\Sigma\nabla U(x_0) + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\epsilon > 0$ is a step size.
- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{2}\Sigma\nabla U(x_0), \epsilon^2\Sigma)$,
  or set $x_1 = x_0$ with the remaining probability.

The preconditioned Crank–Nicolson Langevin (pCNL) algorithm is originally designed for posterior sampling with a latent Gaussian field model (Cotter et al. 2013). The target density is $\pi(x) \propto \exp\{-U(x)\} = \exp\{\ell(x)\}\mathcal{N}(x|\mathbf{0}, C)$, a product of a likelihood function and a normal prior with variance C. For easy comparison, we use a parameterization in terms of the step size ϵ and the potential gradient, $\nabla U(x) = -\nabla\ell(x) + C^{-1}x$.

pCNL.

- Sample $Z \sim \mathcal{N}(\mathbf{0}, C)$ and compute

$$x^* = \sqrt{1-\epsilon^2}x_0 + \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}C\nabla\ell(x_0) + \epsilon Z$$

$$= x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} C\nabla U(x_0) + \epsilon Z. \quad (2)$$

- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} C\nabla U(x_0), \epsilon^2 C)$, or set $x_1 = x_0$ with the remaining probability.

It is interesting to compare pMALA and pCNL. On one hand, pCNL is close to pMALA with the preconditioning matrix Σ chosen to be C, as the step size $\epsilon \to 0$ and hence $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \to \frac{\epsilon^2}{2}$ in Equation (2). On the other hand, as ϵ stays away from 0, the coefficient $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}$ associated with the potential gradient in pCNL can differ considerably from $\frac{\epsilon^2}{2}$ in pMALA. As discussed in Cotter et al. (2013), a simple advantage of pCNL is that when the likelihood gradient $\nabla\ell$ is dropped, the proposal (2) becomes $x^* = \sqrt{1-\epsilon^2}x_0 + \epsilon Z$, which is invariant and reversible with respect to the prior $\mathcal{N}(\mathbf{0}, C)$. Hence, for the target $\mathcal{N}(\mathbf{0}, C)$, the proposal x^* is always accepted in pCNL, but not in pMALA. To achieve such a rejection-free property for Gaussian targets also plays an important role in our work.

From the preceding discussion, it is natural to define a modified pMALA algorithm, by replacing the update coefficient $\frac{\epsilon^2}{2}$ with $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}$ in pMALA. Equivalently, this algorithm can also be obtained from pCNL, by replacing the prior variance C with a general preconditioning matrix Σ, which can be specified as an approximation to the variance of the target distribution $\pi(x)$, depending on both the prior and the likelihood. The modified pMALA algorithm is rejection-free (i.e., the proposal x^* is always accepted) when the target density is $\mathcal{N}(\mathbf{0}, \Sigma)$. To our knowledge, such an extension of pMALA and pCNL appears not explicitly studied before. In Section 3.5, we obtain modified pMALA (with $\Sigma = I$) as a boundary case of the proposed HAMS algorithms (before preconditioning).

Modified preconditioned Metropolis-adjusted Langevin algorithm (pMALA*).

- Generate $x^* = x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \Sigma\nabla U(x_0) + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$.
- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \Sigma\nabla U(x_0), \epsilon^2\Sigma)$, or set $x_1 = x_0$ with the remaining probability.

We also point out that modified pMALA is distinct from a related gradient-based algorithm, denoted as mGrad, in Titsias and Papaspiliopoulos (2018), which is proposed in the context of posterior sampling with the target density $\pi(x) \propto \exp\{-U(x)\} = \exp\{\ell(x)\}\mathcal{N}(x|\,\mathbf{0}, C)$. The associated proposal scheme can be written as

$$x^* = \frac{2}{\delta}\tilde{C}x_0 + \tilde{C}\nabla\ell(x_0) + Z = x_0 - \tilde{C}\nabla U(x_0) + Z,$$
$$Z \sim \mathcal{N}(\mathbf{0}, \frac{2}{\delta}\tilde{C}^2 + \tilde{C}), \quad (3)$$

where $\tilde{C} = (\frac{2}{\delta}I + C^{-1})^{-1}$ and $\nabla U(x_0) = -\nabla\ell(x_0) + C^{-1}x_0$. Compared with the pCNL proposal (2), when $\nabla\ell$ is dropped, the proposal (3) is also reversible with respect to $\mathcal{N}(\mathbf{0}, C)$, but the coefficient matrix $\frac{2}{\delta}\tilde{C}$ depends on C instead of being a multiple of identity. Both pCNL and mGrad can be seen to use only the prior variance C for preconditioning and hence in general differ from pMALA*, which explicitly allows a preconditioning matrix Σ to capture both the prior and the likelihood. See the Supplement Section I for further discussion on preconditioning and auxiliary variable derivations.

The following methods require augmenting the sample space to include a momentum variable $u \in \mathbb{R}^k$, which is assumed to be normally distributed, $u \sim \mathcal{N}(\mathbf{0}, M)$. The variance M is also called a mass matrix, and the quantity $u^{\mathrm{T}}M^{-1}u/2$ represents the kinetic energy in physics. The joint target density of (x, u) becomes

$$\pi(x, u) \propto \exp\{-H(x, u)\} = \exp\{-U(x) - \frac{1}{2}u^{\mathrm{T}}M^{-1}u\}, \quad (4)$$

where $H(x, u) = U(x) + \frac{1}{2}u^{\mathrm{T}}M^{-1}u$, called a total energy or Hamiltonian. For sampling from an augmented target distribution $\pi(x, u)$, HMC generates a proposal by first redrawing a momentum variable and then performing a series of deterministic updates, based on molecular dynamics (MD) simulations such that the Hamiltonian $H(x, u)$ is approximately preserved (Duane et al. 1987; Neal 2011).

HMC.

- Sample $u^* \sim \mathcal{N}(\mathbf{0}, M)$, reset $u_0 = u^*$, and set $x^* = x_0$.
- For i from 1 to *nleap*, repeat:
  $u^* \leftarrow u^* - \frac{\epsilon}{2}\nabla U(x^*)$, $\quad x^* \leftarrow x^* + \epsilon M^{-1}u^*$,
  $u^* \leftarrow u^* - \frac{\epsilon}{2}\nabla U(x^*)$.
- Set $(x_1, u_1) = (x^*, u^*)$ with probability $\min(1, \exp(H(x_0, u_0) - H(x^*, u^*)))$ or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability.

The steps within the for loop are called leapfrog updates, which provide an accurate discretization of the Hamiltonian dynamics, defined as a system of differential equations by Newton's laws of motion such that the Hamiltonian $H(x, u)$ is preserved over time. Although the update of u can be ignored, the acceptance–rejection step above is stated such that the update of (x, u) matches UDL and GMC later with $c = 1$, if the momentum were not resampled. For HMC, both the step size ϵ and the number of leapfrog steps *nleap* need to be tuned. For automated tuning, it seems popular to use the No-U-Turn Sampler (Hoffman and Gelman 2014). Nevertheless, HMC often requires a large number of leapfrog steps for each update of configurations, which can be computationally wasteful.

An important extension of the Hamiltonian dynamics is Langevin dynamics, which can be defined as a system of stochastic differential equations,

$$\mathrm{d}x_t = u_t\,\mathrm{d}t, \quad \mathrm{d}u_t = -\eta\,\mathrm{d}x_t - \nabla U(x_t)\,\mathrm{d}t + \sqrt{2\eta}\,\mathrm{d}W_t, \quad (5)$$

where $\eta > 0$ is a friction coefficient and W_t is the standard Brownian process. In the case of $\eta \to 0$, the Langevin dynamics reduces to the deterministic Hamiltonian dynamics, $\mathrm{d}x_t = u_t\,\mathrm{d}t$ and $\mathrm{d}u_t = -\nabla U(x_t)\,\mathrm{d}t$. In the high-friction limit (i.e., large η), the overdamped Langevin diffusion process is obtained: $\mathrm{d}x_t = -\eta^{-1}\nabla U(x_t)\,\mathrm{d}t + \sqrt{2\eta^{-1}}\,\mathrm{d}W_t$. Hence, Equation (5) is also called underdamped Langevin dynamics. Although Langevin dynamics has long been used in molecular simulations (e.g., van Gunsteren and Berendsen 1982), there is extensive and growing

research related to Langevin dynamics in physics and chemistry (e.g., Horowitz 1991; Scemama et al. 2006; Bussi and Parrinello 2007; Goga et al. 2012; Grønbech-Jensen and Farago 2013, 2020) and machine learning and statistics (e.g., Ottobre et al. 2016; Cheng et al. 2018; Dalalyan and Riou-Durand 2020). In particular, the Metropolized version of the algorithm in Bussi and Parrinello (2007) can be described as follows, to accommodate an acceptance-rejection step.

UDL.

- Sample $Z_1, Z_2 \sim \mathcal{N}(\mathbf{0}, M)$ independently, and compute

$$\begin{aligned} u^+ &= \sqrt{c}u_0 + \sqrt{1-c}Z_1, \\ \tilde{u} &= u^+ - \tfrac{\epsilon}{2}\nabla U(x_0), \quad x^* = x_0 + \epsilon M^{-1}\tilde{u}, \\ u^- &= \tilde{u} - \tfrac{\epsilon}{2}\nabla U(x^*), \\ u^* &= \sqrt{c}u^- + \sqrt{1-c}Z_2, \end{aligned}$$

  where $0 \le c \le 1$ is a tuning parameter and can be interpreted as $c = \mathrm{e}^{-\eta\epsilon}$.
- Set $(x_1, u_1) = (x^*, u^*)$ with probability $\min(1, \exp(H(x_0, u^+) - H(x^*, u^-)))$
  or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability.

There are several interesting features in UDL. First, the proposal scheme in UDL contains a (deterministic) leapfrog update, which is sandwiched by two random updates of the momentum. Notably, the current momentum u_0 is partially refreshed at the beginning, where the amount of "carryover" is controlled by the parameter c. At the two extremes, $c = 0$ or 1, UDL recovers pMALA or Metropolized leapfrog, respectively. When $c = 0$, the first updated momentum $u^+ = Z_1$ is independent of u_0 and the final updated momentum $u^* = Z_2$ can be ignored. In this case, UDL reduces to HMC with one leapfrog step (after redrawing the momentum) and hence is equivalent to pMALA as discussed in Neal (2011). When $c = 1$, UDL generates a proposal by one leapfrog update and then accept or reject (with u_0 flipped) based on the change in the Hamiltonian.

Second, the proposal scheme in UDL is derived in Bussi and Parrinello (2007) by a particular choice of operator splitting in discretizing the Langevin dynamics (5). Compared with other possible choices, the UDL proposal scheme is shown to satisfy a generalized formulation of detailed balance. However, as discussed later in Section 4, whether a sampling algorithm leaves a target distribution invariant also depends on how acceptance or rejection is executed. While Bussi and Parrinello (2007) only mentioned that acceptance–rejection can be performed similarly as in Scemama et al. (2006), the acceptance-rejection step above is explicitly added by our understanding. In the Supplement, we verify the validity of the UDL algorithm in leaving the target augmented density $\pi(x, u)$ invariant, using our proposed framework of generalized Metropolis–Hastings sampling.

Third, the two momentum updates are in the form of an order-1 autoregressive process, which leaves the momentum distribution invariant. As discussed in Bussi and Parrinello (2007), such updates using two independent noise vectors are exploited to achieve generalized detailed balance. In fact, it is instructive to compare UDL with a related algorithm in Horowitz (1991), which uses only one noise vector per iteration as described below. For this algorithm, invariance with respect to $\pi(x, u)$ is valid because each iteration is a composition of two steps, first $(x_0, u_0) \to (x_0, u^+)$ and then $(x_0, u^+) \to (x_1, u_1)$ by Metropolized leapfrog, and each step leaves the target $\pi(x, u)$ invariant. However, it seems difficult to show that generalized detailed balance is directly satisfied by the GMC algorithm. The discussion of GMC in Fang, Sanz-Serna, and Skeel (2014, sec. V.B.) requires splitting the initial momentum update into two updates presumably similar as in UDL, and does not specify how acceptance-rejection would be performed after the modification.

GMC.

- Sample $Z_1 \sim \mathcal{N}(\mathbf{0}, M)$, and compute

$$\begin{aligned} u^+ &= \sqrt{c}u_0 + \sqrt{1-c}Z_1, \\ \tilde{u} &= u^+ - \tfrac{\epsilon}{2}\nabla U(x_0), \quad x^* = x_0 + \epsilon M^{-1}\tilde{u}, \\ u^- &= \tilde{u} - \tfrac{\epsilon}{2}\nabla U(x^*). \end{aligned}$$

- Set $(x_1, u_1) = (x^*, u^-)$ with probability $\min(1, \exp(H(x_0, u^+) - H(x^*, u^-)))$
  or set $(x_1, u_1) = (x_0, -u^+)$ with the remaining probability.

Another interesting method is the irreversible MALA algorithm in Ma et al. (2018). Compared with our method using an augmented density with a momentum as an auxiliary variable, this method relies on a binary auxiliary variable to facilitate irreversible sampling, while using discretizations of continuous dynamics in the original variable x as proposal schemes. See Section 4 and Supplement Section III for further discussion.

3. Proposed Methods

We develop our methods in several steps. We first construct proposal schemes using gradient information, then introduce modifications to derive a class of generalized reversible algorithms HAMS, and finally study two specific algorithms, HAMS-A/B, and propose tuning and preconditioning strategies. To focus on main ideas, consider the augmented target density (4) with momentum variance $M = I$, that is,

$$\pi(x, u) \propto \exp(-H(x, u)) = \exp(-U(x) - u^{\mathrm{T}}u/2), \quad (6)$$

until Section 3.6 on preconditioning. The proposed algorithms are then placed in a more abstract framework of generalized Metropolis–Hastings sampling in Section 4.

3.1. Construction of Hamiltonian Proposals

We provide a simple, broad class of proposal distributions, which are suitable for use in standard Metropolis–Hastings sampling from an augmented density $\pi(x, u)$. These proposal schemes will be modified later for developing irreversible algorithms.

Given the current variables (x_0, u_0), a proposal (x^*, u^*) can be generated as

$$\begin{pmatrix} x^* \\ u^* \end{pmatrix} = \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A \begin{pmatrix} \nabla U(x_0) \\ u_0 \end{pmatrix} + \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, 2A - A^2), \quad (7)$$

where A is a $(2k)\times(2k)$ symmetric positive semi-definite (PSD) matrix and $Z_1, Z_2 \in \mathbb{R}^k$ are Gaussian noises independent of (x_0, u_0), with k the dimension of x and that of u. We require $\mathbf{0} \le A \le 2I$, where inequalities between matrices are in the PSD sense. This ensures that $2A - A^2$ is also symmetric PSD, although allowed to be singular. The update in (7) takes a gradient step from the current variables (x_0, u_0) and then injects Gaussian noises (Z_1, Z_2). Hence, the proposal scheme (7) is similar to that in pMALA. However, Equation (7) is applied to (x, u) jointly, instead of x alone.

The proposal scheme (7) can be derived through an auxiliary variable argument related to Titsias and Papaspiliopoulos (2018), but with at least two nontrivial differences: a momentum variable is included and an over-relaxation technique (Adler 1981; Neal 1998) is exploited to allow $I < A \le 2I$. See Supplement Section I for further discussion.

Another important motivation for the proposal scheme (7) is that Metropolis–Hastings sampling using Equation (7) becomes rejection-free, while generating correlated draws, in the canonical case where the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, that is, $U(x) = x^{\mathrm{T}}x/2$ with the gradient $\nabla U(x) = x$. In fact, the proposal scheme (7) in this case gives

$$\begin{pmatrix} x^* \\ u^* \end{pmatrix} = (I - A)\begin{pmatrix} x_0 \\ u_0 \end{pmatrix} + \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, 2A - A^2). \tag{8}$$

See Law (2014) and Titsias and Papaspiliopoulos (2018), Section 3.4, for related discussion. The update from (x_0, u_0) to (x^*, u^*) in Equation (8) can be seen to define an order-1 vector autoregressive process, VAR(1), which is invariant and reversible with respect to $\mathcal{N}(\mathbf{0}, I)$ due to symmetry of A (Osawa 1988). The invariance can be easily verified: if $(x_0, u_0) \sim \mathcal{N}(\mathbf{0}, I)$, then (x^*, u^*) is normal and the mean and variance are

$$\begin{aligned} &\mathrm{E}\{(x^{*\mathrm{T}}, u^{*\mathrm{T}})^{\mathrm{T}}\} = \mathbf{0}, \\ &\mathrm{var}\{(x^{*\mathrm{T}}, u^{*\mathrm{T}})^{\mathrm{T}}\} = (I - A)(I - A)^{\mathrm{T}} + 2A - A^2 = I. \end{aligned} \tag{9}$$

The reversibility of Equation (8) with respect to $\mathcal{N}(\mathbf{0}, I)$ implies that when the augmented density $\pi(x, u)$ is $\mathcal{N}(\mathbf{0}, I)$, Metropolis–Hastings sampling using the proposal scheme (8) is rejection-free: the proposal (x^*, u^*) is always accepted. This can also be shown by using the proposal density, $Q(x^*, u^*|x_0, u_0) = \mathcal{N}(x^*, u^*|(I - A)(x^{\mathrm{T}}, u^{\mathrm{T}})^{\mathrm{T}}, 2A - A^2)$, and directly verifying that the acceptance probability (1) with x replaced by (x, u) reduces to 1.

Our discussion focuses on the proposal scheme (7) for a Hamiltonian with momentum $u \sim \mathcal{N}(\mathbf{0}, I)$ and the VAR(1) representation (8) in the canonical case $x \sim \mathcal{N}(\mathbf{0}, I)$, related to the normal approximation (S2) in the Supplement with identity variance I in the auxiliary variable derivation. The development can be readily extended to handle general variance matrices, for a momentum distribution $u \sim \mathcal{N}(\mathbf{0}, M)$ and a normal approximation to $\pi(x)$ with variance matrix Σ. Nevertheless, as discussed in Section 3.6, it is convenient to set $M = I$ and if an approximation of $\mathrm{var}(x)$ is available, apply linear transformation to x such that the target density $\pi(x)$ can be roughly aligned with an identity variance $\Sigma = I$.

3.2. HAMS: A Class of Generalized Reversible Algorithms

In this and subsequent sections, we exploit the class of proposals (7) with general choices of matrix A, to first derive a broad class of generalized reversible algorithms HAMS and then study two specific algorithms HAMS-A/B more elaborately.

For simplicity, consider the following form of A matrix in Equation (7),

$$A = \begin{pmatrix} a_1 I & a_2 I \\ a_2 I & a_3 I \end{pmatrix}, \tag{10}$$

where each I is a $k \times k$ identity matrix and a_1, a_2, a_3 are scalar coefficients. We require $0 \le a_1, a_3 \le 2$, $a_1 a_3 \ge a_2^2$, and $(2 - a_1)(2 - a_3) \ge a_2^2$, such that $\mathbf{0} \le A \le 2I$ (in the PSD sense). Substituting this choice of A into (7) yields

$$x^* = x_0 - a_1 \nabla U(x_0) - a_2 u_0 + Z_1, \tag{11}$$

$$u^* = u_0 - a_2 \nabla U(x_0) - a_3 u_0 + Z_2, \tag{12}$$

where $(Z_1^{\mathrm{T}}, Z_2^{\mathrm{T}})^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$ as before. As discussed in Section 3.1, standard Metropolis-Hastings sampling using this proposal scheme is rejection-free, that is, (x^*, u^*) is always accepted, when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$.

Modification for generalized reversibility. We first make a modification to (11) and (12) by replacing the momentum u_0 with $-u_0$. Although a formal justification is to achieve generalized reversibility as shown in Proposition 1, we give a heuristic motivation by noticing that $a_2 u_0$ in Equation (11) and $a_2 \nabla U(x_0)$ in Equation (12) are of the same sign. In contrast, for the discretization of Hamiltonian dynamics using Euler's method:

$$x^* = x_0 + \epsilon u_0, \quad u^* = u_0 - \epsilon \nabla U(x_0),$$

the momentum u_0 and gradient $\nabla U(x_0)$ are of the opposite signs. This discrepancy can be resolved by setting $u_0 \mapsto -u_0$, for which (11) and (12) become

$$x^* = x_0 - a_1 \nabla U(x_0) + a_2 u_0 + Z_1, \tag{13}$$

$$u^* = -u_0 - a_2 \nabla U(x_0) + a_3 u_0 + Z_2, \tag{14}$$

where $(Z_1^{\mathrm{T}}, Z_2^{\mathrm{T}})^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$ as before.

The proposal (x^*, u^*) in (13) and (14) can be accepted or rejected, similarly as in standard Metropolis–Hastings sampling but using a different acceptance probability, which we derive through generalized detailed balance. Rewrite the proposal scheme (13)–(14) as

$$\tilde{Z}_1 = Z_1 - a_1 \nabla U(x_0) + a_2 u_0, \quad \tilde{Z}_2 = Z_2 - a_2 \nabla U(x_0) + a_3 u_0, \tag{15}$$

$$x^* = x_0 + \tilde{Z}_1, \quad u^* = -u_0 + \tilde{Z}_2. \tag{16}$$

Equations (15) and (16) determine a forward transition from (x_0, u_0) to (x^*, u^*), depending on noises (Z_1, Z_2). To construct a backward transition, define new noises

$$Z_1^* = \tilde{Z}_1 - a_1 \nabla U(x^*) - a_2 u^*, \quad Z_2^* = \tilde{Z}_2 - a_2 \nabla U(x^*) - a_3 u^*. \tag{17}$$

Then (17) and (16) can be equivalently rearranged to

$$\begin{aligned} -\tilde{Z}_1 &= -Z_1^* - a_1 \nabla U(x^*) + a_2(-u^*), \\ -\tilde{Z}_2 &= -Z_2^* - a_2 \nabla U(x^*) + a_3(-u^*), \end{aligned} \tag{18}$$

$$x_0 = x^* + (-\tilde{Z}_1), \quad -u_0 = u^* + (-\tilde{Z}_2). \tag{19}$$

Importantly, (18) and (19) can be seen to correspond to the *same* mapping as (15) and (16), but applied from $(x^*, -u^*)$ to $(x_0, -u_0)$ using the new noises $(-Z_1^*, -Z_2^*)$. In other words, (18)–(19) are obtained from (15) and (16) by replacing (x_0, u_0), (x^*, u^*), and (Z_1, Z_2) with $(x^*, -u^*)$, $(x_0, -u_0)$, and $(-Z_1^*, -Z_2^*)$, respectively.

From the preceding discussion, the forward and backward transitions of the proposals in (15) and (16), and (18) and (19) can be illustrated as

$$\begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \overset{(Z_1,Z_2)}{\longrightarrow} \begin{pmatrix} x^* \\ u^* \end{pmatrix}, \qquad \begin{pmatrix} x^* \\ -u^* \end{pmatrix} \overset{-(Z_1^*,Z_2^*)}{\longrightarrow} \begin{pmatrix} x_0 \\ -u_0 \end{pmatrix}, \tag{20}$$

where the two arrows denote the *same* mapping, depending on (Z_1, Z_2) or $-(Z_1^*, Z_2^*)$. For $(Z_1^{\mathrm{T}}, Z_2^{\mathrm{T}})^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$, the proposal density from (x_0, u_0) to (x^*, u^*) is

$$Q(x^*, u^* | x_0, u_0) = \mathcal{N}(Z_1, Z_2 | \, \mathbf{0}, 2A - A^2),$$

because the Jacobian of the transformation is 1. Moreover, evaluation of the *same* proposal density from $(x^*, -u^*)$ to $(x_0, -u_0)$ gives

$$Q(x_0, -u_0 | x^*, -u^*) = \mathcal{N}(-(Z_1^*, Z_2^*) | \, \mathbf{0}, 2A - A^2),$$

because the transition from $(x^*, -u^*)$ to $(x_0, -u_0)$ is determined by the same mapping as (x_0, u_0) to (x^*, u^*), only with the noises $(-Z_1^*, -Z_2^*)$ used instead of (Z_1, Z_2).

By mimicking (and extending) the standard Metropolis–Hastings probability, we set $(x_1, u_1) = (x^*, u^*)$ with the acceptance probability

$$\rho(x^*, u^* | x_0, u_0) = \min\left(1, \frac{\pi(x^*, u^*) Q(x_0, -u_0 | x^*, -u^*)}{\pi(x_0, u_0) Q(x^*, u^* | x_0, u_0)}\right), \tag{21}$$

or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability. Then the probability (21) can be calculated as

$$\rho(x^*, u^* | x_0, u_0) = \min\left(1, \frac{\exp\left\{-H(x^*, u^*) - \frac{1}{2}\mathbf{Z}^{*\mathrm{T}}(2A - A^2)^{-1}\mathbf{Z}^*\right\}}{\exp\left\{-H(x_0, u_0) - \frac{1}{2}\mathbf{Z}^{\mathrm{T}}(2A - A^2)^{-1}\mathbf{Z}\right\}}\right), \tag{22}$$

where $\mathbf{Z} = (Z_1^{\mathrm{T}}, Z_2^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{Z}^* = (Z_1^{*\mathrm{T}}, Z_2^{*\mathrm{T}})^{\mathrm{T}}$. Note that $u_1 = u^*$ upon acceptance, but $u_1 = -u_0$ in the case of rejection. The resulting transition from (x_0, u_0) to (x_1, u_1) can be shown to satisfy generalized detailed balance.

Proposition 1. For an augmented density $\pi(x, u)$ in Equation (6), let $K_0(x_1, u_1 | x_0, u_0)$ be the transition kernel from (x_0, u_0) to (x_1, u_1), defined by the proposal scheme (15) and (16) and the acceptance probability (21). Then generalized detailed balance holds for any (x_0, x_1)

$$\pi(x_0, u_0) K_0(x_1, u_1 | x_0, u_0) = \pi(x_1, u_1) K_0(x_0, -u_0 | x_1, -u_1). \tag{23}$$

Furthermore, the augmented density $\pi(x, u)$ is a stationary distribution of the Markov chain defined by transition kernel K_0.

Condition (23), called generalized detailed balance (or generalized reversibility), differs from detailed balance (or reversibility) in standard Metropolis–Hastings sampling because the momentum variable is negated in defining the backward transition. Accordingly, the acceptance probability (21) is called a generalized Metropolis–Hastings probability. The concept of generalized detailed balance is known in connection with Fokker–Planck equations in physics (Gardiner 1997, sec. 5.3.4). The momentum is called an odd variable, for which the time-reversed variable is defined with sign negation to achieve generalized detailed balance. Such a generalized detailed balance is used in various algorithms in physics (Scemama et al. 2006; Bussi and Parrinello 2007; Fang, Sanz-Serna, and Skeel 2014), but overall seems to be underappreciated in the MCMC literature. See Section 4 for a further extension.

Modification for updating momentums. To further broaden our method, we introduce another modification to the proposal scheme (15) and (16). In fact, a potential limitation of (15) and (16), compared with the popular leapfrog scheme, is that the updated momentum u^* ignores the new gradient information $\nabla U(x^*)$. To incorporate $\nabla U(x^*)$ in updating the momentum, we revise (16) with an additional term in u^* as

$$\begin{aligned} x^* &= x_0 + \tilde{Z}_1, \\ u^* &= -u_0 + \tilde{Z}_2 + \phi(\tilde{Z}_1 + \nabla U(x_0) - \nabla U(x^*)), \end{aligned} \tag{24}$$

where ϕ is a (constant) tuning parameter, and $(\tilde{Z}_1, \tilde{Z}_2)$ remain the same as in Equation (15). Moreover, the update Equation (24) can be rearranged to

$$\begin{aligned} x_0 &= x^* + (-\tilde{Z}_1), \\ -u_0 &= u^* + (-\tilde{Z}_2) + \phi(-\tilde{Z}_1 + \nabla U(x^*) - \nabla U(x_0)). \end{aligned} \tag{25}$$

With (Z_1^*, Z_2^*) still defined as Equation (17), (15) and (24) and (18) and (25) can be seen to be determined by the *same* mapping, similarly as illustrated in Equation (20). The forward transition is from (x_0, u_0) to (x^*, u^*) depending on (Z_1, Z_2), whereas the backward transition is from $(x^*, -u^*)$ to $(x_0, -u_0)$ depending on $-(Z_1^*, Z_2^*)$. With the modified proposal (x^*, u^*), the acceptance–rejection is the same as before: set $(x_1, u_1) = (x^*, u^*)$ with probability (21) or $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability. Then generalized detailed balance remains valid for the transition from (x_0, u_0) and (x_1, u_1).

Proposition 2. For an augmented density $\pi(x, u)$ in Equation (6), let $K_\phi(x_1, u_1 | x_0, u_0)$ be the transition kernel from (x_0, u_0) to (x_1, u_1), defined by the proposal scheme (15) and (24) and the acceptance probability (21). Then generalized detailed balance holds for any (x_0, x_1):

$$\pi(x_0, u_0) K_\phi(x_1, u_1 | x_0, u_0) = \pi(x_1, -u_1) K_\phi(x_0, -u_0 | x_1, -u_1). \tag{26}$$

Furthermore, the augmented density $\pi(x, u)$ is a stationary distribution of the Markov chain defined by transition kernel K_ϕ.

General HAMS. Using the proposal scheme and acceptance probability as in Proposition 2 leads to a class of generalized

Algorithm 1: General HAMS

Initialize x_0, u_0
for $t = 0, 1, 2, ..., N_{iter}$ **do**
Sample $w \sim$ Uniform$[0, 1]$ and
$(Z_1, Z_2)^{\mathrm{T}} \sim N(\mathbf{0}, 2A - A^2)$ with $A = \begin{pmatrix} a_1 I & a_2 I \\ a_2 I & a_3 I \end{pmatrix}$
$\tilde{Z}_1 = Z_1 - a_1 \nabla U(x_t) + a_2 u_t$
$\tilde{Z}_2 = Z_2 - a_2 \nabla U(x_t) + a_3 u_t$
Propose $x^* = x_t + \tilde{Z}_1$ and
$u^* = -u_t + \tilde{Z}_2 + \phi(\tilde{Z}_1 + \nabla U(x_t) - \nabla U(x^*))$
$Z_1^* = \tilde{Z}_1 - a_1 \nabla U(x^*) - a_2 u^*$
$Z_2^* = \tilde{Z}_2 - a_2 \nabla U(x^*) - a_3 u^*$
$\rho = \exp\{H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2}\mathbf{Z}^{\mathrm{T}}(2A - A^2)^{-1}\mathbf{Z} - \frac{1}{2}\mathbf{Z}^{*\mathrm{T}}(2A - A^2)^{-1}\mathbf{Z}^*\}$
if $w < \min(1, \rho)$ **then**
$(x_{t+1}, u_{t+1}) = (x^*, u^*)$ # Accept
else
$(x_{t+1}, u_{t+1}) = (x_t, -u_t)$ # Reject

reversible MCMC algorithms, which is called Hamiltonian-assisted Metropolis sampling (HAMS) and shown in Algorithm 1.

Although the modifications of the proposal scheme from (11)–(12) to (13)–(14) and then to (15) and (24) are constructed for different purposes, the resulting HAMS algorithm preserves the rejection-free property for a standard normal target density $\pi(x)$, which is satisfied by standard Metropolis–Hastings sampling with proposal scheme (11)–(12). In fact, the second modification from (16) to (24) has no effect when $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, because in this case $\tilde{Z}_1 + \nabla U(x_0) - \nabla U(x^*) = \tilde{Z}_1 + x_0 - x^* = \mathbf{0}$. The justification for the first modification is subtler. Whether rejection-free is achieved by a sampling algorithm depends on both the proposal scheme and the associated acceptance–rejection mechanism. When $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, our HAMS algorithm is rejection-free, due to the fact the proposal scheme (13) and (14) is used in conjunction with the generalized acceptance probability (21), not the standard Metropolis–Hastings probability. We provide further discussion in Section 4, where consideration of the rejection-free property for normal targets is instrumental to a general approach for constructing generalized reversible algorithms.

Corollary 1. Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. Then the generalized acceptance probability (21) or equivalently (22) reduces to 1, and hence (x^*, u^*) from the proposal scheme (13) and (14) is always accepted under the HAMS algorithm.

The general HAMS algorithm involves four tuning parameters $\phi, a_1, a_2,$ and a_3, which need to be specified for practical implementation. In the following sections, we develop concrete versions of HAMS with a reduced number of tuning parameters.

3.3. HAMS-A and HAMS-B

The noise term (Z_1, Z_2) in HAMS is $2k$ dimensional Gaussian with variance matrix $2A - A^2$. There are related methods developed for simulating Langevin dynamics, using k dimensional noises at each time step (Grønbech-Jensen and Farago 2013, 2020). We investigate HAMS where the variance matrix $2A - A^2$ is singular and hence only a k dimensional Gaussian noise is used in each iteration. There are two possible choices: either A itself is singular or $2I - A$ is singular, corresponding to HAMS-A and HAMS-B below.

HAMS-A. First, we set A singular by taking $a_1 = a$, $a_3 = b$, and $a_2 = \sqrt{ab}$ in Equation (10). The constraint $\mathbf{0} \le A \le 2I$ reduces to $a \ge 0$, $b \ge 0$, and $a + b \le 2$. To avoid trivial cases, we also assume that $a > 0$. The noise variance becomes

$$\mathrm{var}\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = 2A - A^2 = \begin{pmatrix} a(2-a-b)I & \sqrt{ab}(2-a-b)I \\ \sqrt{ab}(2-a-b)I & b(2-a-b)I \end{pmatrix}. \quad (27)$$

As expected, this implies that Z_1 and Z_2 are proportional: $Z_2 = \sqrt{b/a}Z_1$. By definitions (15), (24), and (17), it can be easily verified that $\tilde{Z}_2 = \sqrt{b/a}\tilde{Z}_1$ and $Z_2^* = \sqrt{b/a}Z_1^*$ as well. The proportionality between Z_1^* and Z_2^* is important, because it ensures that both forward and backward transitions, illustrated in Equation (20), can be determined using a single noise vector, Z_1 or $-Z_1^*$. Hence, the proposal density from (x_0, u_0) to (x^*, u^*) is $\mathcal{N}(Z_1|\mathbf{0}, a(2-a-b)I)$ and that from $(x^*, -u^*)$ to $(x_0, -u_0)$ is $\mathcal{N}(-Z_1^*|\mathbf{0}, a(2-a-b)I)$. The acceptance probability (21) can be evaluated as Equation (28) below, while Equation (22) is not well defined.

From the preceding discussion, the HAMS algorithm can be simplified as follows, given the current variables (x_0, u_0):

$$\tilde{Z} = Z - a\nabla U(x_0) + \sqrt{ab}u_0, \quad Z \sim \mathcal{N}(\mathbf{0}, a(2-a-b)I),$$
$$x^* = x_0 + \tilde{Z},$$
$$u^* = -u_0 + \sqrt{\frac{b}{a}}\tilde{Z} + \phi(\tilde{Z} + \nabla U(x_0) - \nabla U(x^*)),$$
$$Z^* = \tilde{Z} - a\nabla U(x^*) - \sqrt{ab}u^*.$$

The proposal (x^*, u^*) is accepted with probability

$$\min\left(1, \exp\left\{H(x_0, u_0) - H(x^*, u^*) + \frac{Z^{\mathrm{T}}Z - (Z^*)^{\mathrm{T}}Z^*}{2a(2-a-b)}\right\}\right). \quad (28)$$

With the choice of ϕ derived below, this algorithm is shown as HAMS-A in Algorithm 2, after a transformation $Z = \sqrt{a(2-a-b)}\zeta$ with $\zeta \sim \mathcal{N}(\mathbf{0}, I)$.

To derive a specific choice for ϕ, we examine the situation where the target density $\pi(x)$ deviates from standard normal. As discussed in Section 3.2, the HAMS algorithm is rejection-free, that is, the acceptance probability (28) is always 1, when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. We seek a choice of ϕ such that the acceptance probability can be minimally affected by the deviation of γ from 1, when $\pi(x)$ is $\mathcal{N}(\mathbf{0}, \gamma^{-1}I)$. For simplicity, we study the behavior of the quantity inside exp() in Equation (28) as γ varies.

Lemma 1. Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, \gamma^{-1}I)$. Then the quantity inside exp() in Equation (28) can be expressed as a quadratic form,

$$H(x_0, u_0) - H(x^*, u^*) + \frac{Z^{\mathrm{T}}Z - (Z^*)^{\mathrm{T}}Z^*}{2a(2-a-b)} = (x_0^{\mathrm{T}}, u_0^{\mathrm{T}}, Z^{\mathrm{T}})G(\gamma)(x_0^{\mathrm{T}}, u_0^{\mathrm{T}}, Z^{\mathrm{T}})^{\mathrm{T}},$$

where $G(\gamma)$ is a 3×3 block matrix. For $i, j = 1, 2, 3$, the (i, j)th block of $G(\gamma)$ is of the form $g_{ij}(\gamma)I$, where $g_{ij}(\gamma)$ is a scalar, polynomial of γ, with coefficients depending on (a, b, ϕ). For any $a > 0, b \geq 0$ and $a + b \leq 2$, the coefficients of the leading terms of $g_{11}(\gamma), g_{22}(\gamma), g_{33}(\gamma)$ are simultaneously minimized in absolute values by the choice $\phi = \sqrt{ab}/(2 - a)$.

It seems remarkable that a single choice of ϕ leads to simultaneous minimization of the absolute coefficients of the leading terms of $g_{11}(\gamma), g_{22}(\gamma), g_{33}(\gamma)$. Moreover, the particular choice $\phi = \sqrt{ab}/(2-a)$ also ensures that HAMS-A reduces to leapfrog or modified pMALA in the special cases where $a + b = 2$ or $b = 0$, as discussed in Section 3.5.

HAMS-B. For a singular $2A - A^2$, another possibility is to set $2I - A$ singular. We take $a_1 = 2 - \tilde{a}, a_3 = 2 - \tilde{b}$ and $a_2 = \sqrt{\tilde{a}\tilde{b}}$ in (10), with the constraints that $\tilde{a} > 0, \tilde{b} \geq 0$ and $\tilde{a} + \tilde{b} \leq 2$. The noise variance is then

$$\operatorname{var}\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = 2A - A^2 = \begin{pmatrix} \tilde{a}(2 - \tilde{a} - \tilde{b})I & \sqrt{\tilde{a}\tilde{b}}(\tilde{a} + \tilde{b} - 2)I \\ \sqrt{\tilde{a}\tilde{b}}(\tilde{a} + \tilde{b} - 2)I & \tilde{b}(2 - \tilde{a} - \tilde{b})I \end{pmatrix}, \quad (29)$$

which implies that Z_1 and Z_2 are proportional: $Z_2 = -\sqrt{\tilde{b}/\tilde{a}}Z_1$. However, it does not in general hold that $Z_2^* = -\sqrt{\tilde{b}/\tilde{a}}Z_1^*$, except for the choice $\phi = \sqrt{\tilde{b}/\tilde{a}}$. Moreover, this choice of ϕ is the only one such that any proportionality between (Z_1^*, Z_2^*) holds. This situation is in contrast with HAMS-A, where $Z_2^* = \sqrt{b/a}Z_1^*$ automatically holds for any choice of ϕ and additional consideration is needed to derive a specific choice of ϕ.

Lemma 2. For the preceding choice of A in (10), it holds that $Z_2^* = rZ_1^*$ for a constant coefficient $r \in \mathbb{R}$ and arbitrary values (x_0, u_0, Z_1) by definitions (15), (24), and (17) if and only if $r = -\sqrt{\tilde{b}/\tilde{a}}$ and $\phi = \sqrt{\tilde{b}/\tilde{a}}$.

To maintain the forward and backward transitions, illustrated in Equation (20), using a single noise vector, we take the only feasible choice $\phi = \sqrt{\tilde{b}/\tilde{a}}$. Then the HAMS algorithm can be simplified as follows, given the current variables (x_0, u_0):

$$\tilde{Z} = Z - (2 - \tilde{a})\nabla U(x_0) + \sqrt{\tilde{a}\tilde{b}}u_0, \quad Z \sim \mathcal{N}(0, \tilde{a}(2 - \tilde{a} - \tilde{b})I),$$

$$x^* = x_0 + \tilde{Z}, \quad u^* = u_0 + \sqrt{\frac{\tilde{b}}{\tilde{a}}}(\nabla U(x_0) + \nabla U(x^*)),$$

$$Z^* = \tilde{Z} - (2 - \tilde{a})\nabla U(x^*) - \sqrt{\tilde{a}\tilde{b}}u^*.$$

Similarly as discussed for HAMS-A, the acceptance probability (21) can be evaluated as (28). To facilitate comparison with HAMS-A, we use a reparameterization, $a = 2 - \tilde{a}$ and $b = \tilde{a}\tilde{b}/(2 - \tilde{a})$, or equivalently $\tilde{a} = 2 - a$ and $\tilde{b} = ab/(2 - a)$. The transformation is one-to-one between $\{(a, b) : a > 0, b > 0, a+b \leq 2\}$ and $\{(\tilde{a}, \tilde{b}) : \tilde{a} > 0, \tilde{b} > 0, \tilde{a}+\tilde{b} \leq 2\}$. The resulting algorithm is shown as HAMS-B in Algorithm 2. By the (a, b) parameterization, the two algorithms, HAMS-A and HAMS-B, agree in the expressions for x^*.

Algorithm 2: HAMS-A/HAMS-B

Initialize x_0, u_0
for $t = 0, 1, 2, ..., N_{iter}$ **do**
  Sample $w \sim$ Uniform$[0, 1]$ and $\zeta \sim \mathcal{N}(\mathbf{0}, I)$
  Propose
  $x^* = x_t - a\nabla U(x_t) + \sqrt{ab}u_t + \sqrt{a(2 - a - b)}\zeta$
  if *HAMS-A* **then**
    Propose $u^* = \left(\frac{2b}{2-a} - 1\right) u_t - \frac{\sqrt{ab}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}\zeta$
    $\zeta^* = \left(1 - \frac{2b}{2-a}\right) \zeta - \frac{\sqrt{a(2-a-b)}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}u_t$
  if *HAMS-B* **then**
    Propose $u^* = u_t - \frac{\sqrt{ab}}{2-a}(\nabla U(x_t) + \nabla U(x^*))$
    $\zeta^* = \zeta - \frac{\sqrt{a(2-a-b)}}{2-a}(\nabla U(x_t) + \nabla U(x^*))$
  $\rho = \exp\left\{H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2}\zeta^{\mathrm{T}}\zeta - \frac{1}{2}(\zeta^*)^{\mathrm{T}}\zeta^*\right\}$
  if $w < \min(1, \rho)$ **then**
    $(x_{t+1}, u_{t+1}) = (x^*, u^*)$ # Accept
  else
    $(x_{t+1}, u_{t+1}) = (x_t, -u_t)$ # Reject

3.4. Default Choices of Carryover

While the (a, b) parameterization arises naturally in our development above, the (ϵ, c) parameterization used in existing algorithms (see Section 2) has a desirable interpretation, with ϵ corresponding to a step size and c the amount of carryover momentum. By matching leapfrog and modified pMALA in special cases (see Section 3.5), our HAMS algorithms can be translated into an (ϵ, c) parameterization with the following formulae:

$$a = \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}} = 1 - \sqrt{1 - \epsilon^2}, \quad b = c(2 - a), \quad 0 \leq \epsilon, c \leq 1. \quad (30)$$

Because a is expressed as a function of ϵ only, and b given a is a function of c only, we also refer to a as a step size and b as a carryover.

So far, the number of tuning parameters is reduced from four in general HAMS (Algorithm 1) to two in HAMS-A/B (Algorithm 2). To facilitate applications, we seek to further reduce tuning complexity by studying the lag-1 auto-covariance matrix for a HAMS chain in stationarity when the target density $\pi(x)$ is standard normal.

Lemma 3. Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, and $(x_0, u_0) \sim \mathcal{N}(\mathbf{0}, I)$. Given step size a, the maximum modulus of the eigenvalues of the lag-1 auto-covariance matrix $\operatorname{cov}((x_0, u_0), (x_1, u_1))$ is minimized by the following choice of b:

$$\text{HAMS-A: } b = (\sqrt{2} - \sqrt{a})^2, \quad \text{HAMS-B: } b = \frac{a(2 - a)}{(\sqrt{2} + \sqrt{2 - a})^2}. \quad (31)$$

For convenience, the formulae (31) can be used as the default choices of carryover b, given step size a. On the other hand, such

choices are derived under an idealized setting, where the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. For the default tuning to be effective, we often need to first apply transformations to bring $\pi(x)$ closer to $\mathcal{N}(\mathbf{0}, I)$, which will be discussed in Section 3.6. If such a transformation is not available for various reasons, then it is preferable to tune both a and b instead of using the default values in (31).

Our strategy in identifying choices of a, b, ϕ involve various considerations, including Lemmas 1 and 3 based on normal targets. These ideas share certain similarity with the integration schemes for HMC proposed in Blanes, Casas, and Sanz-Serna (2014), where integration coefficients are optimized in terms of a criterion related to acceptance rates under the standard normal target. A notable difference, however, is that the HAMS algorithm is, by construction, rejection-free (with acceptance rates equal to 1) under standard normal.

3.5. Special Cases of HAMS-A/B

Recall that the constraints on the step size and carryover are $a \geq 0, b \geq 0, a+b \leq 2$. In the following, we examine three boundary cases.

The first case is when $a + b = 2$ (or equivalently $c = 1$). For both HAMS-A and HAMS-B, the updates become deterministic from (x_0, u_0) to (x^*, u^*) . To help understanding, we introduce an intermediate variable $\tilde{u}$. Then the updates can be written as

$$\begin{aligned}
&\zeta \sim \mathcal{N}(\mathbf{0}, I), \quad \zeta^* = -\zeta \text{ (HAMS-A)}, \quad \zeta^* = \zeta \text{ (HAMS-B)},\\
&\tilde{u} = u_0 - \sqrt{\frac{a}{2-a}}\nabla U(x_0) = u_0 - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\nabla U(x_0),\\
&x^* = x_0 + \sqrt{a(2-a)}\tilde{u}_0 = x_0 + \epsilon\tilde{u},\\
&u^* = \tilde{u} - \sqrt{\frac{a}{2-a}}\nabla U(x^*) = \tilde{u} - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\nabla U(x^*),
\end{aligned}$$

where the Metropolis ratio is $\rho = \exp(H(x_0, u_0) - H(x^*, u^*))$. The above is similar to the leapfrog discretization of the Hamiltonian dynamics but with step size $\epsilon/(1 + \sqrt{1-\epsilon^2})$ instead of $\epsilon/2$ for momentum updates. The proposal (x^*, u^*) can be accepted or rejected (with u_0 flipped) based on the change in the Hamiltonian from the update. Incidentally, this modified leapfrog scheme can be used in place of the usual leapfrog scheme in HMC or UDL such that the Gaussian-calibrated rejection-free property is satisfied.

The second case is when $b = 0$ (or equivalently $c = 0$). We introduce another intermediate variable $\tilde{\zeta}$ to the updates. Then HAMS-A and HAMS-B reduce to

$$\begin{aligned}
&\zeta \sim \mathcal{N}(\mathbf{0}, I), \quad u^* = -u_0 \text{ (HAMS-A)}, \quad u^* = u_0 \text{ (HAMS-B)},\\
&\tilde{\zeta} = \zeta - \sqrt{\frac{a}{2-a}}\nabla U(x_0) = \zeta - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\nabla U(x_0),\\
&x^* = x_0 + \sqrt{a(2-a)}\tilde{\zeta} = x_0 + \epsilon\tilde{\zeta},\\
&\zeta^* = \tilde{\zeta} - \sqrt{\frac{a}{2-a}}\nabla U(x^*) = \tilde{\zeta} - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\nabla U(x^*),
\end{aligned}$$

where the Metropolis ratio is $\rho = \exp(U(x_0) - U(x^*) + \frac{1}{2}\zeta^{\mathrm{T}}\zeta - \frac{1}{2}(\zeta^*)^{\mathrm{T}}\zeta^*)$. Hence, u_0 remains unchanged in HAMS-B, and is negated in HAMS-A, although the update of u_0 is irrelevant in this case. The update of x_0 to x^* and acceptance-rejection coincide with modified pMALA in Section 2, which differs from ordinary pMALA because the step size $\epsilon^2/(1 + \sqrt{1-\epsilon^2})$ is associated with $\nabla U(x_0)$ for updating x_0, instead of $\epsilon^2/2$.

The third case is when $a = 0$ (or equivalently $\epsilon = 0$). This case is not interesting because x remains constant. Our discussion is for completeness. When $a = 0$, HAMS-B sets all variables constant: $x^* = x_0, u^* = u_0$, and $\zeta^* = \zeta$. HAMS-A gives the updates

$$\begin{aligned}
&\zeta \sim \mathcal{N}(\mathbf{0}, I), \qquad x^* = x_0,\\
&u^* = (b-1)u_0 + \sqrt{b(2-b)}\zeta,\\
&\zeta^* = (1-b)\zeta + \sqrt{b(2-b)}u_0.
\end{aligned}$$

In this case, the Metropolis ratio is always 1. Hence, HAMS-A can be viewed as an autoregressive process on u while x remains constant.

Finally, we note that our HAMS-A/B algorithms differ from UDL (Bussi and Parrinello 2007), which uses two noise vectors per iteration, although UDL also recovers leapfrog and pMALA in the extreme cases of $c = 1$ and $c = 0$, respectively.

3.6. Preconditioning

As commonly recognized in MCMC literatures, if there is information about the variance structure of the target density, then the performance of MCMC samplers can be improved by applying a linear transformation, that is, preconditioning. Suppose that Σ is an approximation to $\mathrm{Var}(x)$, or M is an approximation to $\mathrm{Var}^{-1}(x)$. Then RWM and pMALA involve preconditioning using the approximate variance Σ on x, whereas HMC and UDL involve preconditioning using M as the momentum variance. These two approaches are conceptually equivalent, as discussed in the context of HMC by Neal (2011), although one can be more preferable than the other in computational implementations.

We use the first approach of preconditioning: applying a linear transformation to the original variable x while keeping the momentum $u \sim \mathcal{N}(\mathbf{0}, I)$. Let L be the lower triangular matrix obtained from the Cholesky decomposition $M = LL^{\mathrm{T}}$. The transformed variable is $\tilde{x} = L^{\mathrm{T}}x$. If x is approximately $\mathcal{N}(0, M^{-1})$, then $\tilde{x}$ is approximately $\mathcal{N}(\mathbf{0}, I)$. Application of HAMS-A/B in Algorithm 2 to the transformed variable $\tilde{x}$ leads to HAMS-A/B algorithms with preconditioning, which are shown in Algorithm 3. The gradient of the potential after the transformation, denoted as $\nabla U(\tilde{x})$, is $L^{-1}\nabla U(x)$.

Our Algorithm 3 is carefully formulated, such that transforming x and keeping $u \sim \mathcal{N}(\mathbf{0}, I)$ improves computational efficiency, compared with using the original variable x and $u \sim \mathcal{N}(\mathbf{0}, M)$. See the Supplement Section IV.8 for details of simplification. Excluding the evaluation of $U(x)$ and $\nabla U(x)$, Algorithm 3 involves 2 matrix-by-vector multiplications per iteration, $(L^{\mathrm{T}})^{-1}\tilde{x}^*$ and $L^{-1}\nabla U(x^*)$. Moreover, computation of the Metropolis ratio ρ is also optimized, requiring only 1 inner product instead of 4 as in Algorithm 2. In contrast, UDL as described in Section 2 needs 5 matrix-by-vector multiplications per iteration: 2 for sampling from $\mathcal{N}(\mathbf{0}, M)$, 1 for computing x^*, and 2 in the Metropolis ratio. In the simulation studies, we implement UDL with reduced runtime in a similar way as

Algorithm 3: HAMS-A/HAMS-B (with preconditioning)

Initialize $x_0, u_0, \tilde{x}_0 = L^{\mathrm{T}} x_0$ and $\nabla U(\tilde{x}_0) = L^{-1}\nabla U(x_0)$.
for $t = 0, 1, 2, ..., N_{iter}$ **do**
  Sample $w \sim \text{Uniform}[0, 1]$ and $\zeta \sim \mathcal{N}(\mathbf{0}, I)$
  $\xi = \sqrt{ab}u_t + \sqrt{a(2-a-b)}\zeta, \quad \tilde{x}^* = \tilde{x}_t - a\nabla U(\tilde{x}_t) + \xi$
  Propose $x^* = (L^{\mathrm{T}})^{-1}\tilde{x}^*$
  $\nabla U(\tilde{x}^*) = L^{-1}\nabla U(x^*), \quad \tilde{\xi} = \nabla U(\tilde{x}^*) + \nabla U(\tilde{x}_t)$
  $\rho = \exp\left\{U(x_t) - U(x^*) + \frac{1}{2-a}(\tilde{\xi})^{\mathrm{T}}(\xi - \frac{a}{2}\tilde{\xi})\right\}$
  if $w < \min(1, \rho)$ **then**
    $x_{t+1} = x^*, \quad \tilde{x}_{t+1} = \tilde{x}^*, \quad \nabla U(\tilde{x}_{t+1}) = \nabla U(\tilde{x}^*)$
    # Accept
    if *HAMS-A* **then**
      $u_{t+1} = \left(\frac{2b}{2-a} - 1\right) u_t + \frac{2\sqrt{b(2-a-b)}}{2-a}\zeta - \frac{\sqrt{ab}}{2-a}\tilde{\xi}$
    if *HAMS-B* **then**
      $u_{t+1} = u_t - \frac{\sqrt{ab}}{2-a}\tilde{\xi}$
  else
    $x_{t+1} = x_t, u_{t+1} = -u_t, \tilde{x}_{t+1} = \tilde{x}_t, \nabla U(\tilde{x}_{t+1}) = \nabla U(\tilde{x}_t)$ # Reject

Algorithm 3, in order to make fair comparisons with HAMS-A/B.

4. Generalized Metropolis–Hastings Sampling

Our development in Section 3 presents a concrete class of generalized reversible algorithms, HAMS, using an augmented target density originated from a Hamiltonian in physics. In this section, we discuss a flexible framework of generalized Metropolis–Hastings sampling for a target distribution satisfying an invariance property. This framework not only accommodates and sheds light on our construction of HAMS at a more abstract level, but also facilitates possible further development of irreversible MCMC algorithms.

Importance of rejection. Before describing our generalization, it is instructive to discuss a fictitious generalization of Metropolis–Hastings sampling, which satisfies a reversibility-like condition upon acceptance of a proposal, but in general fails to leave a target density invariant due to improperness incurred when a proposal is rejected.

Let $\pi(y)$ be a prespecified probability density function on a space $\mathcal{Y}$. By abuse of notation, we allow that $\pi(y)$ be directly a target density $\pi(x)$ in the context of Section 1 or an augmented target density $\pi(x, u)$ with auxiliary variables u. Consider an MCMC algorithm with the following transition kernel given a current value y_0.

A fictitious generalization of Metropolis–Hastings sampling.

- Sample y^* from a (forward) proposal density $Q(\cdot|y_0)$;
- Set $y_1 = y^*$ with the acceptance probability

$$\tilde{\rho}(y^*|y_0) = \min\left(1, \frac{\pi(y^*)Q_b(y_0|y^*)}{\pi(y_0)Q(y^*|y_0)}\right),$$

  or set $y_1 = y_0$ with the remaining probability, where $Q_b(\cdot|y^*)$ is a backward proposal density.

Let $\tilde{K}(y_1|y_0)$ be the (forward) transition kernel from y_0 to y_1 for the sampling scheme above. Then for any $y_1 \neq y_0$ (i.e., a proposal is accepted, $y_1 = y^*$), it can be easily shown that $\tilde{K}(y_1|y_0) = Q(y_1|y_0)\tilde{\rho}(y_1|y_0)$ and, by a symmetry argument,

$$\pi(y_0)\tilde{K}(y_1|y_0) = \pi(y_1)\tilde{K}_b(y_0|y_1), \tag{32}$$

where $\tilde{K}_b(y_0|y_1) = Q_b(y_0|y_1)\tilde{\rho}(y_0|y_1)$. If (32) were satisfied for $y_1 = y_0$ as well (i.e., a proposal is rejected), then integrating (32) over y_0 would indicate $\int \pi(y_0)\tilde{K}(y_1|y_0)\,dy_0 = \pi(y_1)$, that is, the transition kernel $\tilde{K}$ leaves $\pi(\cdot)$ invariant. Standard Metropolis–Hastings sampling corresponds to choosing $Q_b = Q$, in which case (32) holds trivially for $y_1 = y_0$ as well as for $y_1 \neq y_0$, Such a condition (32) with $\tilde{K}_b = \tilde{K}$ is known as detailed balance or reversibility. For $Q_b \neq Q$, however, (32) may not hold for $y_1 = y_0$, in spite of the fact that (32) is satisfied for $y_1 \neq y_0$. Therefore, the preceding sampling scheme in general fails to leave $\pi(\cdot)$ invariant, for the complication caused by rejection of a proposal.

Our discussion above uses an heuristic interpretation of the transition kernel $\tilde{K}$ in the case of rejection of a proposal. The issue is also reflected in the difficulty to obtain a more rigorous justification similar as in Tierney (1994). See (Ma et al. 2018, sec. 3.3), for a related discussion on a naive approach for constructing irreversible samplers.

Generalized Metropolis–Hastings sampling. As motivated by our construction of HAMS algorithms, we propose generalized Metropolis–Hastings sampling provided that a target density $\pi(y)$ is invariant under an orthogonal transformation. Let J be an orthogonal matrix J such that $\pi(J^{-1}y) = \pi(y)$ for $y \in \mathcal{Y}$. By the change of variables with $|\det(J)| = 1$, this is equivalent to requiring that for any set $C \subset \mathcal{Y}$,

$$\int_{J(C)} \pi(y)\,dy = \int_C \pi(y)\,dy. \tag{33}$$

where $J(C) = \{Jy : y \in C\} \subset \mathcal{Y}$. Consider a sampling algorithm defined by the following transition kernel given a current value y_0.

Generalized Metropolis–Hastings sampling (GMH).

- Sample y^* from a (forward) proposal density $Q(\cdot|y_0)$.
- Set $y_1 = y^*$ with the acceptance probability

$$\rho(y^*|y_0) = \min\left(1, \frac{\pi(J^{-1}y^*)Q(Jy_0|J^{-1}y^*)}{\pi(y_0)Q(y^*|y_0)}\right), \tag{34}$$

  or set $y_1 = Jy_0$ with the remaining probability.

Condition (33) is trivially satisfied for $J = I$ (the identity matrix), in which case the preceding algorithm reduces to standard Metropolis–Hastings sampling.

There are two notable differences compared with the fictitious generalization earlier. First, the backward proposal density is explicitly defined as $Q(Jy_0|J^{-1}y^*)$. It is helpful to think of the proposal density $Q(y^*|y_0)$ as being induced by a stochastic mapping, $y^* = \mathcal{M}(y_0; Z)$ for a noise Z. Then $Q(Jy_0|J^{-1}y^*)$ corresponds to the density of Jy_0 given $J^{-1}y^*$ under the same mapping, $Jy_0 = \mathcal{M}(J^{-1}y^*; Z^*)$, but with a new noise Z^* considered to be identically distributed as Z. See, for example, (36) and

(37) below. Hence, the forward and backward transitions of the proposals can be illustrated, similarly to Equation (20), as

$$y_0 \xrightarrow{Z} y^*, \quad J^{-1}y^* \xrightarrow{Z^*} Jy_0,$$

where the two arrows denote the same mapping, depending on Z or Z^*. Second, the next variable y_1 is defined as Jy_0 instead of y_0, in the case of rejection. The generalization can be shown to be valid in leaving the target distribution $\pi(y)$ invariant.

Proposition 3. Suppose that invariance (33) is satisfied. Let $K(y_1|y_0)$ be the (forward) transition kernel from y_0 to y_1 for generalized Metropolis–Hastings sampling. Then generalized detailed balance holds for any (y_0, y_1):

$$\pi(y_0)K(y_1|y_0) = \pi(y_1)K(Jy_0|J^{-1}y_1), \tag{35}$$

Moreover, the target density $\pi(y)$ is a stationary density of the Markov chain defined by the transition kernel $K(y_1|y_0)$.

To connect with HAMS, generalized Metropolis–Hastings sampling is discussed above in terms of continuous variables. However, our framework can be broadened to accommodate both continuous and discrete variables, by allowing Jy to be an orthogonal-like mapping, for example, flipping a binary variable from one value to the other. In the Supplement Section III, we show that the irreversible jump sampler (I-Jump) in Ma et al. (2018) can be obtained as a special case of generalized Metropolis–Hastings sampling with a symmetric, binary auxiliary variable. Hence, our HAMS algorithm differs from I-Jump in using a momentum as an auxiliary variable, and exploiting symmetry of mean-zero normal distributions.

Generalized gradient-guided Metropolis sampling. The framework of generalized Metropolis–Hastings sampling allows a flexible specification of the proposal density Q. Our HAMS algorithms use a proposal scheme which takes a gradient step and then adds Gaussian noises. Using a similar update scheme, Equation (36), in generalized Metropolis–Hastings sampling leads to a class of gradient-guided sampling algorithms. Similarly as in Section 3.1, let $\mathbf{0} \le A \le 2I$ be a symmetric matrix in the order on PSD matrices. For a target $\pi(y)$, a potential function $U(y)$ is defined such that $\pi(y) \propto \exp\{-U(y)\}$. This potential $U(y)$ can be the augmented potential $U(x) + u^{\mathrm{T}}u/2$ in Section 3.

Generalized gradient-guided Metropolis sampling (G2MS).

- Generate y^* as

$$y^* = y_0 - B\nabla U(y_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, 2A - A^2), \tag{36}$$

  where $B = I - (I - A)J$ and $2A - A^2 = B + B^{\mathrm{T}} - BB^{\mathrm{T}}$. Compute Z^* by

$$Jy_0 = J^{-1}y^* - B\nabla U(J^{-1}y^*) + Z^*, \tag{37}$$

  obtained by replacing (y_0, y^*) with $(J^{-1}y^*, Jy_0)$ and Z with Z^* in Equation (36).
- Set $y_1 = y^*$ with the acceptance probability (34), simplified as

$$\rho(y^*|y_0) = \min\left(1, \frac{\pi(y^*)\mathcal{N}(Z^*|\mathbf{0}, 2A - A^2)}{\pi(y_0)\mathcal{N}(Z|\mathbf{0}, 2A - A^2)}\right), \tag{38}$$

  or set $y_1 = Jy_0$ with the remaining probability.

Corollary 2. Suppose that invariance (33) is satisfied. The conclusions of Proposition 3 hold with transition kernel K defined by generalized gradient-guided Metropolis sampling.

In addition to exploiting gradient information, the G2MS algorithm is carefully designed to achieve the rejection-free property when the target density $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$, which satisfies invariance (33) for any orthogonal matrix J. In this case, $U(y) = y^{\mathrm{T}}y/2$ with gradient $\nabla U(y) = y$, and hence the proposal scheme (36) becomes

$$y^* = (I - A)Jy_0 + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, 2A - A^2). \tag{39}$$

The update from y_0 to y^* defines a VAR(1) process, which admits $\mathcal{N}(\mathbf{0}, I)$ as a stationary distribution, that is, if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$ then $y^* \sim \mathcal{N}(\mathbf{0}, I)$, by similar calculation as in Equation (9). However, stationarity of Equation (39) with respect to $\mathcal{N}(\mathbf{0}, I)$ does not automatically imply rejection-free. In fact, because $(I - A)J$ may be asymmetric, the VAR(1) process in Equation (39) is in general irreversible. Standard Metropolis–Hastings sampling using the proposal scheme (39) is not rejection-free when $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$. Otherwise, the resulting Markov chain is irreversible, which contradicts reversibility of standard Metropolis–Hastings sampling. Nevertheless, the G2MS algorithm achieves rejection-free when $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$, due to the combination of the proposal scheme (39) with the generalized acceptance probability (38). In other words, the backward proposal density induced from Equation (37) agrees with the conditional density of y_0 given y^* if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$ and y^* is generated by Equation (39). See the proof for details.

Corollary 3. Suppose that the target density $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$. Then the generalized acceptance probability (38) reduces to 1, and hence y^* from the proposal scheme (36) is always accepted under the G2MS algorithm.

From the preceding discussion, the G2MS algorithm can be seen as being extended from a VAR(1) process in the form Equation (39). For completeness, we remark that the form of Equation (39) depending on A and J is universal. In fact, consider a general VAR(1) process

$$y^* = (I - \tilde{B})y_0 + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, \tilde{B} + \tilde{B}^{\mathrm{T}} - \tilde{B}\tilde{B}^{\mathrm{T}}), \tag{40}$$

where $\tilde{B}$ is a possibly asymmetric matrix such that $\tilde{B} + \tilde{B}^{\mathrm{T}} - \tilde{B}\tilde{B}^{\mathrm{T}}$ is (symmetric and) PSD. Let $I - \tilde{B} = O_1\Lambda O_2$ be a singular value decomposition, where O_1 and O_2 are orthogonal matrices, Λ is a diagonal matrix containing the singular values of $I - \tilde{B}$. Then $I - \tilde{B}$ can be written as

$$I - \tilde{B} = (O_1\Lambda O_1^{\mathrm{T}})(O_1O_2) = (I - \tilde{A})\tilde{J},$$

where $\tilde{A} = I - O_1\Lambda O_1^{\mathrm{T}}$ is symmetric and $\tilde{J} = O_1O_2$ is orthogonal. Moreover, the noise variance becomes $\tilde{B} + \tilde{B}^{\mathrm{T}} - \tilde{B}\tilde{B}^{\mathrm{T}} = I - (1 - \tilde{B})(I - \tilde{B})^{\mathrm{T}} = I - (I - \tilde{A})^2 = 2\tilde{A} - \tilde{A}^2$. Therefore, the VAR(1) process (40) can be put in the form Equation (39).

Back to HAMS. The invariance (33) can be satisfied by an augmented target density defined with auxiliary variables. In fact, our HAMS algorithms can be recovered as special cases of generalized Metropolis–Hastings sampling, with $\pi(y) = \pi(x, u)$ in Equation (6) and J a block-diagonal matrix with

$(I, -I)$ on the diagonal. The invariance (33) is satisfied due to evenness of mean-zero normal distributions. The HAMS algorithm studied in Proposition 1 is a special case of G2MS with matrix A in Equation (10). The HAMS algorithm in Proposition 2 is not contained in G2MS due to a modification with $\phi \neq 0$, but can still be treated in the framework of generalized Metropolis–Hastings sampling, with the forward and backward proposal schemes discussed in Section 3.2. The general discussion here broadens our understanding of HAMS algorithms and opens doors for further development.

5. Simulation Studies

We report simulation studies comparing HAMS-A/B with RWM, pMALA, pMALA*, HMC, UDL, GMC, and mGrad (see Section 2). We include RWM as a performance baseline. The simulations include a multivariate normal distribution, a multilevel logistic (MLogit) regression model, a stochastic volatility (SV) model and a log-Gaussian Cox model. For space limitation, the normal and SV experiments, mGrad results in the other two experiments, and plots for GMC and pMALA* are deferred to the Supplement.

For the latent-variable models, all algorithms except mGrad are implemented using the same preconditioning schemes. Two sets of experiments are conducted. First, for sampling latent variables with fixed parameters, constant preconditioning matrices are derived by taking the model expectation of the Hessian in the SV and Cox experiments (Girolami and Calderhead 2011), or evaluating the Hessian at fixed latent variables in the MLogit experiments. Second, for Gibbs sampling which alternately samples latent variables and parameters, a two-stage procedure is applied: first no preconditioning to obtain rough parameter estimates, and then sampling using fixed preconditioning matrices (evaluated at the rough parameters) for latent variables and parameters, respectively. Further details are provided in the subsequent sections and the Supplement. Our preconditioned HMC implementation is equivalent in the first set of experiments to Riemann manifold HMC (RMHMC) in Girolami and Calderhead (2011), but differs in the second set of experiments from RMHMC, where the preconditioning matrices are continuously updated in the two blocks of Gibbs sampling, hence with even greater computational cost.

For ease of comparison and tuning, we use the (ϵ, c) parameterization for HAMS-A and HAMS-B, which is equivalent to the (a, b) parameterization by Equation (30). For the stochastic volatility and log-Gaussian Cox experiments, we fix the number of leapfrog steps for HMC similarly as in Girolami and Calderhead (2011): $nleap = 50$ in sampling latent variables or $nleap = 6$ in sampling parameters. For the multilevel logistic experiments, we fix $nleap = 20$ in sampling latent variables and $nleap = 7$ in sampling parameters, after trial runs using $nleap$ in $\{10, 20, \ldots, 50\}$ and $\{5, 6, 7, 8\}$ respectively. When preconditioning is applied, the c values for HAMS-A/B as well as UDL and GMC are determined in terms of ϵ, by translating the default choices of b given a in Equation (31). Without preconditioning, the c values are specified by the following consideration. Recall that the first momentum update of UDL is $u^+ = \sqrt{c}u_0 + \sqrt{1-c}Z_1$ in the form of an AR(1) process. With the noise $Z_1 \sim \mathcal{N}(0, 1)$, the lag-$h$ auto-covariance for AR(1) is $\gamma(h) = c^{h/2}$. Heuristically, to mimic refreshing momentums after leapfrog steps in HMC, we choose $c = \gamma(h)^{2/h}$ with $h = nleap$ and a small value, 0.001, for $\gamma(h)$. For example, we set $c = 0.76$ or 0.1 corresponding to $nleap = 50$ or 6 and $c = 0.5$ or $c = 0.14$ corresponding to $nleap = 20$ or 7.

For tuning, we adjust step size ϵ during a burn-in period to achieve reasonable acceptance rates: 25–45% for RWM, 50–60% for mGrad, and 70–80% for all other methods. See the Section V.4 (supplementary material) for details. Samples are then collected after the burn-in.

To evaluate MCMC samples, a useful metric is the effective sample size, $\text{ESS} = n/\{1 + 2\sum_{k=1}^{\infty}\rho(k)\}$, where n is the total number of draws and $\rho(k)$ is the lag-k correlation. We report two estimators of ESS that are both suitable for irreversible Markov chains obtained by HAMS-A/B as well as UDL and GMC. The first one is Bartlett's (1948) window estimator as in Ma et al. (2018):

$$\text{ESS}_1 = \frac{n}{1 + 2\sum_{k=1}^{K}\left(1 - \frac{k}{K}\right)\rho(k)}, \tag{41}$$

where the cut-off value K is a large number (taken to be 3000 in our results). For repeated simulations, the ESS_1 estimates are averaged from individual chains. The second one (labelled as ESS_2) is defined from the within and between variances from multiple chains in repeated simulations. Suppose that we have m Markov chains each with n draws, denoted as $\{x_{ij} : i = 1, \ldots, n, j = 1, \ldots, m\}$. Then ESS can be estimated by

$$\text{ESS}_2 = n\frac{W}{B}, \quad W = \frac{1}{m(n-1)}\sum_{i,j}(x_{ij} - \bar{x}_{\cdot j})^2,$$
$$B = \frac{n}{m-1}\sum_{j}(\bar{x}_{\cdot j} - \bar{x})^2, \tag{42}$$

where $\bar{x}_{\cdot j} = n^{-1}\sum_{i=1}^{n} x_{ij}$ and $\bar{x} = m^{-1}\sum_{j=1}^{m}\bar{x}_{\cdot j}$. In fact, B/n is an estimator of the variance of the average of n draws, whereas W is an estimator of the marginal variance of x. The estimator ESS_2 is based on directly measuring consistency between repeated simulations, whereas ESS_1 is determined solely by auto-correlations within individual simulations. Both ESS estimators are computed from each coordinate for a multi-dimensional distribution. As in Girolami and Calderhead (2011), we report the minimum ESS over all coordinates, adjusted by runtime, as a measure of computational efficiency.

5.1. Multilevel Logistic Regression

Consider a multilevel logistic regression model (Gelman and Hill 2007, chap. 14), where the outcome is the political support according to polling data before the 1988 U.S. presidential election, defined as $y_i = 1$ or 0 for Republican or Democratic supporters, $i = 1, ..., 2015$. The outcome is associated with sex (male or female), race (black or other), age (4 categories), education (4 categories), states (49 categories), and the previous voting records as

$$P(y_i = 1) = \text{expit}(\beta_1 + \beta_2\,\text{female}_i + \beta_3\,\text{black}_i + \beta_4\,\text{black}_i \times \text{female}_i + \beta_5\,\text{previous}_i + x^{\text{age}}_{k[i]} + x^{\text{edu}}_{l[i]} + x^{\text{age}\times\text{edu}}_{k[i]\times l[i]} + x^{\text{state}}_{j[i]} + x^{\text{region}}_{m[i]}),$$

Table 1. Runtime and ESS comparison for sampling latent variables in the multilevel logistic regression.

Method	Time (s)	ESS_1 (min, median, max)	$\frac{\text{minESS}_1}{\text{Time}}$	ESS_2 (min, median, max)	$\frac{\text{minESS}_2}{\text{Time}}$
HAMS-A	21.2	(13034, 17315, 22773)	614.8	(2750, 5242, 9950)	129.7
HAMS-B	20.3	(43659, 57046, 71749)	2149.1	(11639, 17191, 29344)	573.1
UDL	20.1	(1815, 2557, 3336)	90.3	(541, 796, 1416)	26.9
GMC	20.1	(2553, 3378, 4223)	127.2	(671, 985, 1945)	33.4
HMC	257.8	(10085, 15846, 32386)	39.1	(28, 255, 1068)	0.1
pMALA	19.4	(1337, 1838, 2254)	69.0	(342, 523, 970)	17.6
pMALA*	18.9	(11457, 14795, 19089)	605.5	(2870, 4735, 7852)	151.7
RWM	6.9	(11, 22, 36)	1.5	(0.3, 1.1, 1.9)	0.04

NOTE: Results are averaged over 50 repetitions.

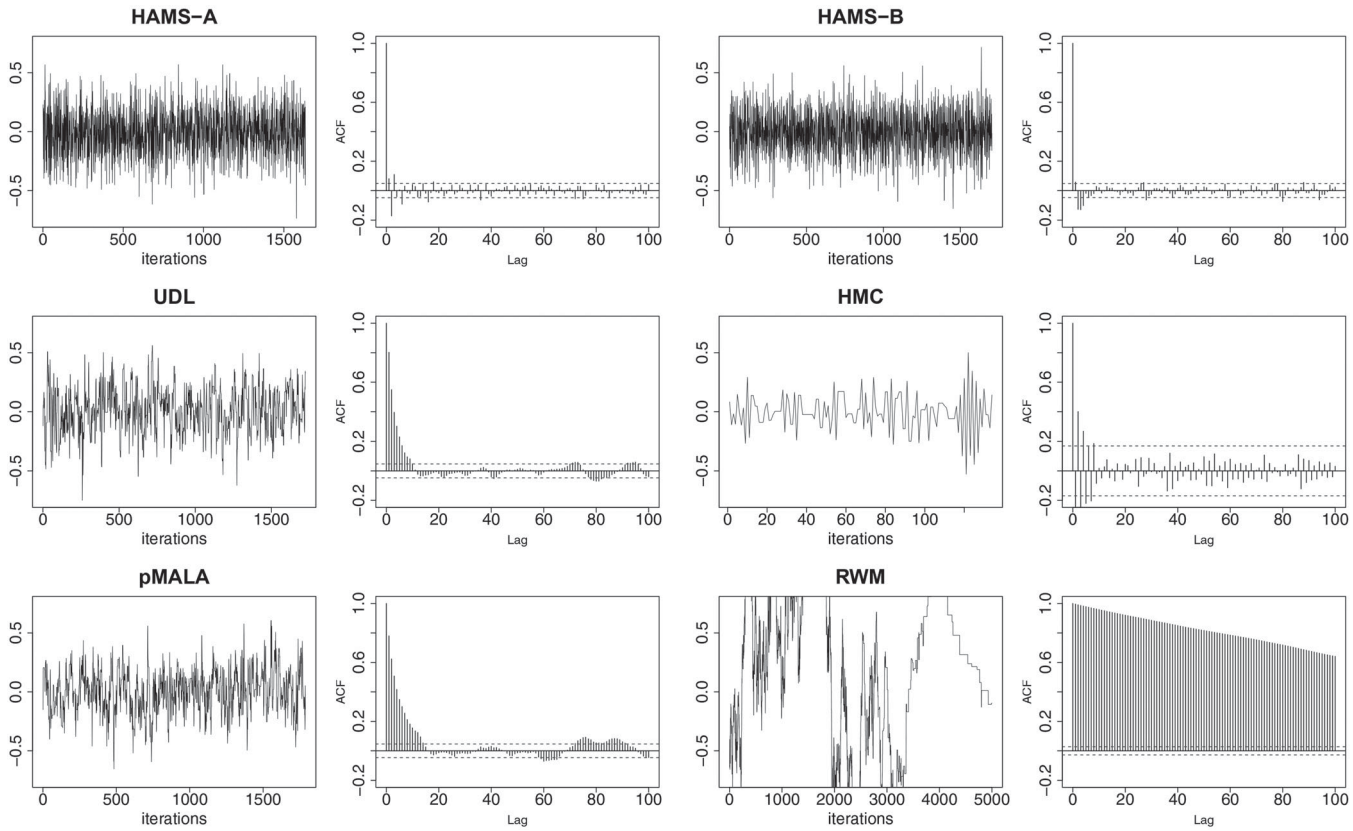


Figure 1. Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the multilevel logistic regression.

where the multilevel coefficients or latent variables (78 in total) are independently Gaussian:

$$
\begin{aligned}
x_k^{\text{age}} &\sim \mathcal{N}(0, \sigma_1^2), \quad k = 1, ..., 4, \\
x_l^{\text{edu}} &\sim \mathcal{N}(0, \sigma_2^2), \quad l = 1, ..., 4, \\
x_{k,l}^{\text{age}\times\text{edu}} &\sim \mathcal{N}(0, \sigma_3^2), \quad k, l = 1, ..., 4, \\
x_j^{\text{state}} &\sim \mathcal{N}(0, \sigma_4^2), \quad j = 1, ..., 49, \\
x_m^{\text{region}} &\sim \mathcal{N}(0, \sigma_5^2), \quad m = 1, ..., 5.
\end{aligned}
$$

The parameters of interest are $\theta = (\beta^{\mathrm{T}}, \sigma^{\mathrm{T}})^{\mathrm{T}}$, with $\beta = (\beta_1, \ldots, \beta_5)^{\mathrm{T}}$ and $\sigma = (\sigma_1, \ldots, \sigma_5)^{\mathrm{T}}$. Denote as $\mathbf{y}$ the outcome vector $(y_1, \ldots, y_{2015})^{\mathrm{T}}$ and as $\mathbf{x}$ the vector of 78 latent variables. Two sets of experiments are conducted. First, we fix parameter values and sample latent variables from $p(\mathbf{x}|\mathbf{y}, \theta)$. Then we perform Bayesian analysis and sample both latent variables and parameters from $p(\mathbf{x}, \theta|\mathbf{y})$. See Supplement Section V.1 for expressions of gradients and preconditioning matrices used.

For the first experiment, we only sample latent variables and fix parameters at the estimates obtained from the R package `lme4`, $\beta = (-3.49, -1.64, -0.09, -0.17, 7.02)^{\mathrm{T}}$ and $\sigma = (0.002, 0.105, 0.150, 0.197, 0.174)^{\mathrm{T}}$. For preconditioning, we approximate the inverse variance $\text{var}^{-1}(\mathbf{x})$ by the Hessian $M = -\nabla^2 \log p(\mathbf{x}|\mathbf{y}, \theta)$ evaluated at $\mathbf{x} = \mathbf{0}$ for HAMS-A/B, UDL, GMC and HMC and use the approximate variance $\Sigma = M^{-1}$ for pMALA, pMALA* and RWM. As mentioned earlier, we use *nleap* $= 20$ for HMC and choose c given ϵ by (30) and (31) for HAMS-A/B and UDL. All algorithms are run for 5000 burn-in iterations, and then samples are collected from 5000 iterations. The simulation process is repeated for 50 times.

Table 1 shows the runtime and ESS comparison. Clearly, HAMS-B has the best performance in terms of time-adjusted minimum ESSs, followed by HAMS-A and pMALA*. The superior performances of HAMS-A/B and pMALA* can be attributed to the fact that larger step sizes are used by these three methods than others, while maintaining high acceptance

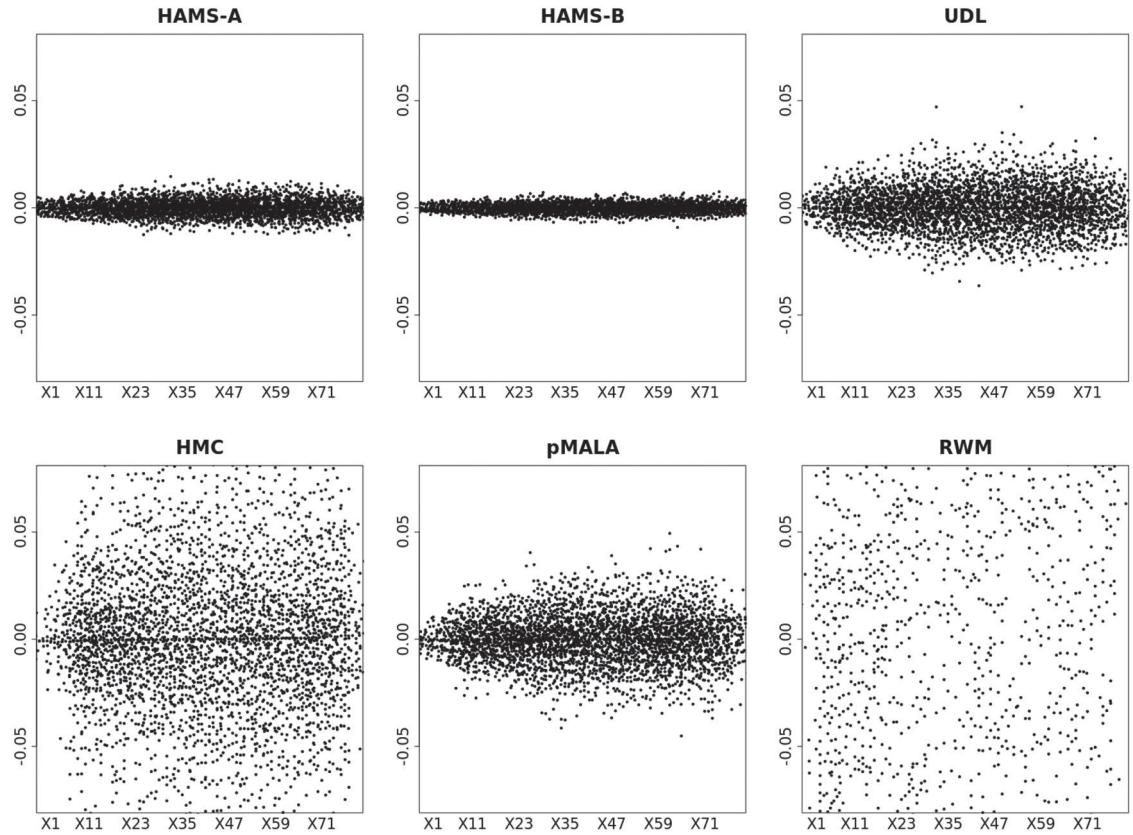


Figure 2. Time-adjusted and centered plots of sample means of all latent variables over 50 repetitions for sampling latent variables in the multilevel logistic regression.

Table 2. Comparison of posterior sampling in the multilevel logistic regression.

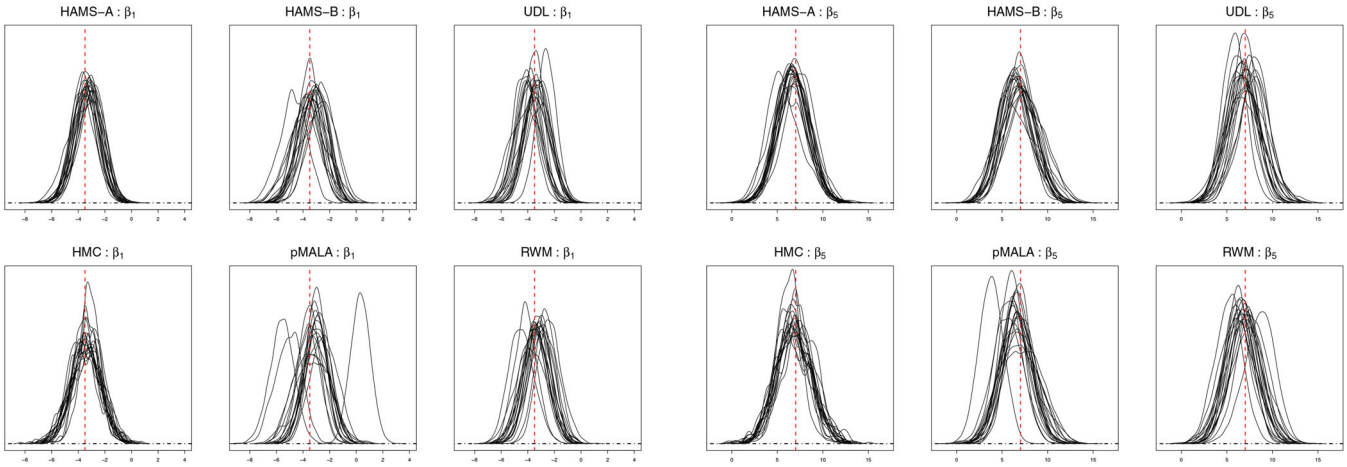
Method	Time (s)	Sample Mean				$\frac{\text{minESS}_1}{\text{Time}}$	$\frac{\text{minESS}_2}{\text{Time}}$
		β_1 (sd)	β_5 (sd)	σ_4 (sd)	σ_5 (sd)	(β, σ)	(β, σ)
HAMS-A	43.9	−3.38 (0.315)	6.77 (0.487)	0.22 (0.023)	0.39 (0.079)	(1.885, 0.461)	(0.222, 0.093)
HAMS-B	43.8	−3.37 (0.522)	6.81 (0.382)	0.23 (0.023)	0.47 (0.345)	(1.711, 0.422)	(0.092, 0.038)
UDL	43.9	−3.47 (0.437)	6.83 (0.630)	0.21 (0.055)	0.42 (0.278)	(1.649, 0.713)	(0.113, 0.029)
GMC	43.8	−3.31 (0.591)	6.88 (0.838)	0.21 (0.063)	0.44 (0.387)	(1.773, 0.462)	(0.070, 0.027)
HMC	404.8	−3.42 (0.118)	6.86 (0.092)	0.22 (0.014)	0.44 (0.037)	(1.844, 0.297)	(0.233, 0.087)
pMALA	44.4	−3.21 (1.151)	6.55 (0.805)	0.22 (0.041)	0.65 (0.629)	(1.421, 0.366)	(0.017, 0.026)
pMALA*	44.3	−3.46 (0.377)	6.96 (0.621)	0.23 (0.042)	0.42 (0.167)	(0.749, 0.240)	(0.173, 0.070)
RWM	22.7	−3.33 (0.489)	6.76 (0.743)	0.22 (0.055)	0.49 (0.292)	(1.632 0.537)	(0.168, 0.047)

NOTE: Standard deviations of sample means are in parentheses. Results are averaged over 20 repetitions.

rates. See the supplementary material, Figure S4, where the step sizes for these three methods increase to almost 1, and the acceptance rates are still over 90%. A possible explanation for these differences is that HAMS-A/B and pMALA* (but not pMALA) satisfy the Gaussian-calibrated rejection-free property and hence is more capable of achieving reasonable acceptance rates with relatively large step sizes when the target density is roughly standard Gaussian after preconditioning.

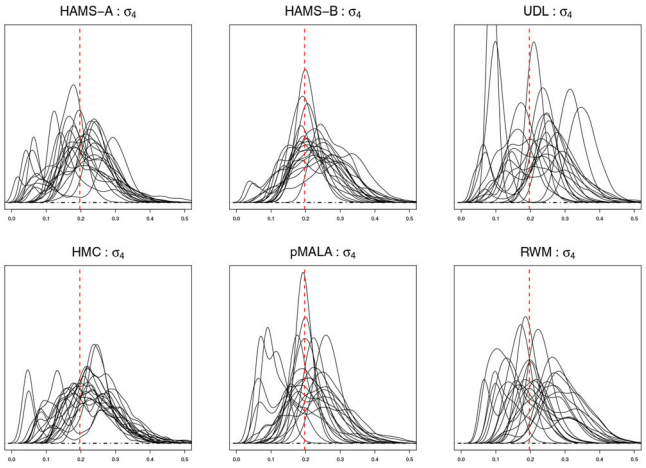
An interesting phenomenon about the ESSs from HAMS-A/B, pMALA* as well as HMC is that an ESS value estimated by both (41) and (42) can exceed the number of draws, 5000, due to negative auto-correlations. Figure 1 shows trace plots of one latent variable and auto-correlation function (ACF) plots from an individual run. The plots for each method are adjusted for runtime after burn-in: we keep the number of draws inversely proportional to the runtime, with RWM keeping all 5000 draws as the baseline. All time-adjusted plots are produced similarly in this and subsequent sections. From the trace plots, HAMS-A and HAMS-B appear to mix better than other methods. Moreover, the ACFs of HAMS-A and HAMS-B decay faster to 0 compared with other methods, while exhibiting negative auto-correlations. For HAMS-B, its large ESS values might be explained by the negative auto-correlations at several consecutive small lags.

Figure 2 shows the time-adjusted plots of the sample means of all latent variables for each method over 50 repeated runs. For each latent variable located on the x-axis, the 50 sample means are plotted along the y-axis. The plots are centered at the corresponding averages, and narrower spreads indicate that a method is more consistent across repeated simulations. Clearly,

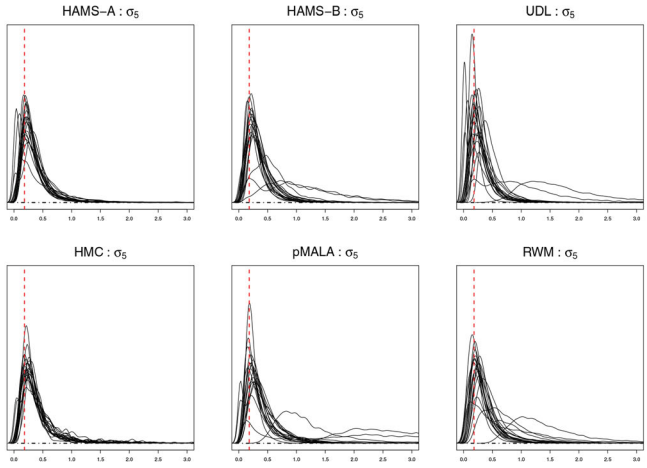


(a) Densities of β_1

(b) Densities of β_5



(c) Densities of σ_4



(d) Densities of σ_5

Figure 3. Time-adjusted posterior density plots of parameters (20 repetitions overlaid) in the multilevel logistic regression. The estimates from `lme4` are marked by vertical lines.

Table 3. Runtime and ESS comparison for sampling latent variables in the log-Gaussian Cox model.

Method	Time (s)	ESS_1 (min, median, max)	$\frac{\text{minESS}_1}{\text{Time}}$	ESS_2 (min, median, max)	$\frac{\text{minESS}_2}{\text{Time}}$
HAMS-A	2013.5	(1015, 1530, 3950)	0.50	(207, 464, 1084)	0.10
HAMS-B	1998.5	(629, 953, 1931)	0.31	(143, 290, 1005)	0.07
UDL	1997.8	(361, 576, 1187)	0.18	(87, 172, 563)	0.04
GMC	1999.1	(397, 625, 1465)	0.20	(98, 185, 532)	0.05
HMC	44425.1	(1011, 7381, 12824)	0.02	(29, 330, 3567)	0.001
pMALA	2862.4	(246, 382, 797)	0.09	(55, 113, 263)	0.02
pMALA*	2873.0	(611, 903, 1955)	0.21	(145, 272, 696)	0.05
RWM	1217.6	(7, 11, 22)	0.01	(0.1, 0.3, 0.7)	0.0001

NOTE: Results are averaged over 50 repetitions.

HAMS-B and HAMS-A are the most consistent, followed by UDL and pMALA. Much more variability is associated with HMC and RWM.

In the second experiment, we perform Bayesian analysis and sample both latent variables and parameters from the posterior $p(\mathbf{x}, \theta|\mathbf{y})$. As in Gelman and Hill (2007), the priors are $\pi(\beta_j) \sim \mathcal{N}(0, 100^2)$ and $\pi(\sigma_j) \propto 1, j = 1, ..., 5$, independently. Under these priors, β and σ are conditionally independent given $(\mathbf{y}, \mathbf{x})$, and each σ_j^2 follows an inverse Gamma distribution (Gamerman 1997). Hence, we employ the following Gibbs sampling scheme: sample $x \sim p(\mathbf{x}|\mathbf{y}, \beta, \sigma^2)$ and $\beta \sim p(\beta|\mathbf{y}, \mathbf{x})$ using MCMC and directly sample $\sigma^2 \sim p(\sigma^2|\mathbf{y}, \mathbf{x})$ as inverse Gamma. In the first experiment, the preconditioning matrix for latent variables are computed only once because the parameters are fixed. In the current experiment, to avoid reevaluating the preconditioning matrix every Gibbs iteration, we first run each algorithm without any preconditioning to obtain crude estimates of the parameters, and then fix the preconditioning matrix evaluated

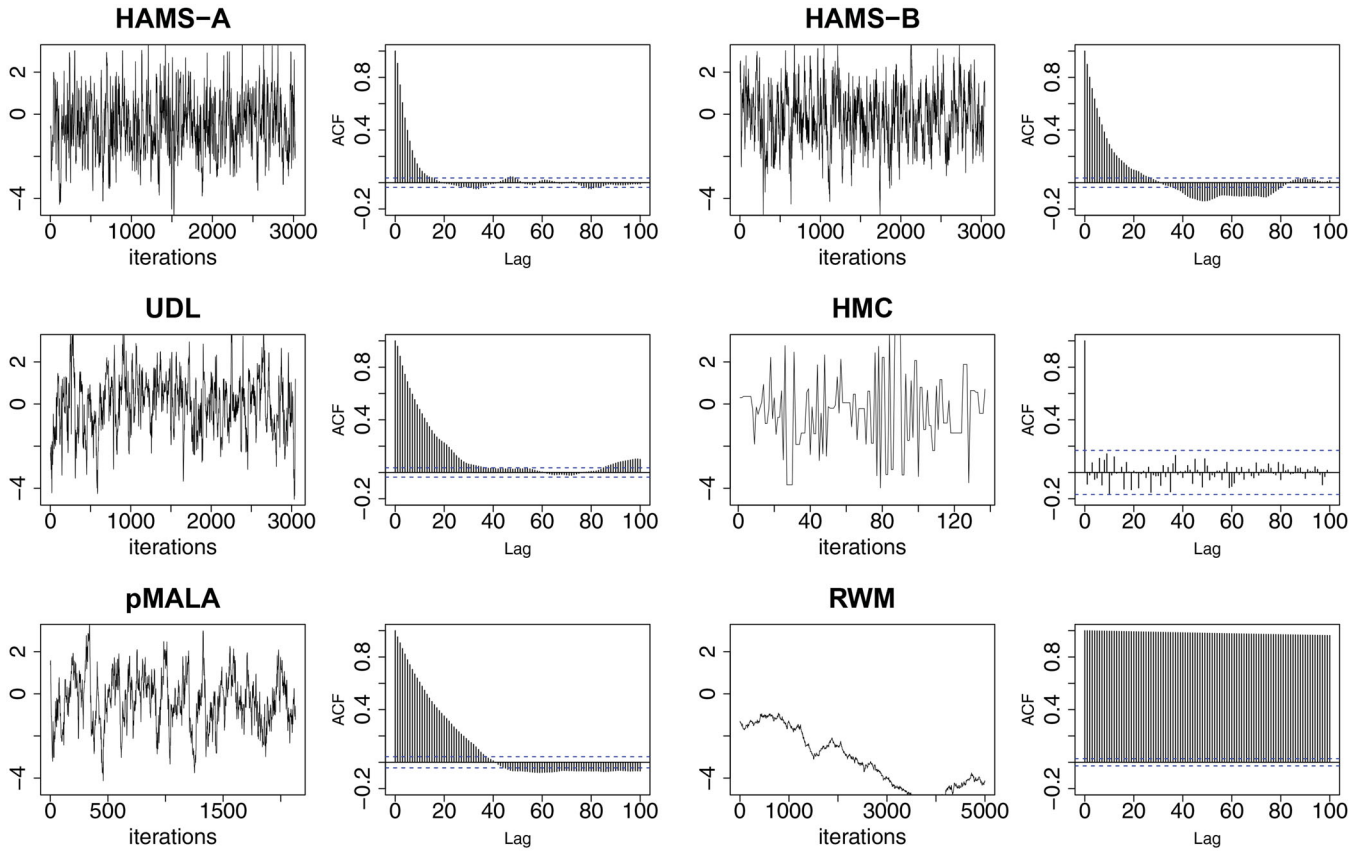


Figure 4. Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the log-Gaussian Cox model.

at the estimates. Each stage includes a burn-in period before sample collection. For HMC, the numbers of leapfrog steps are 20 for latent variables and 7 for parameters. The initial values of parameters are dispersed over the intervals $\beta_j \in [-3, 3]$ and $\sigma_j^2 \in [0.1, 2]$ for $j = 1, ..., 5$. For all methods, 10,000 draws are collected after 10,000 iterations, which includes a stage without preconditioning and a burn-in period with preconditioning. The simulation process is repeated for 20 times.

Posterior sampling results are shown in Table 2 for several selected parameters (see the Supplement for full results). Time adjusted ESSs are reported for β and σ separately, because of their different ranges. Overall, HAMS-A has the best performance in terms of time adjusted ESSs. For posteriors means, all methods except pMALA produce comparable averages, whereas pMALA deviates from the others in β_1, β_2 and σ_5, which along with its large standard deviations are caused by inconsistency during repeated runs. However, HAMS-A produces smaller standard deviations than other methods except HMC, which yields the smallest standard deviations with the same sample size but at the cost of about 10 times longer runtime. The relatively high ESS_1 for σ from RWM should be interpreted with caution. The posterior means from RWM exhibit large variations, notably in β_5 and σ_1, indicating inconsistency among repeated runs (see supplementary material, Table S2).

Figure 3 shows time-adjusted density plots for the selected parameters (see the supplement for full results). The number of draws used is inversely proportional to the run time, with RMW keeping all 10,000 draws. Each plot shows estimated densities from 20 repeated runs overlaid together. Clearly, for β_1 and β_5, HAMS-A yields the most consistent density curves followed by HAMS-B, UDL and HMC, then pMALA and RWM, which sometimes produce outlying curves. For σ_4, curves of HAMS-B are the most consistent, whereas for σ_5, curves of HAMS-A and HMC are the most consistent. HMC density curves are wiggly after adjusted for runtime, which demonstrates its high computational cost.

5.2. Log-Gaussian Cox Model

Consider a log-Gaussian Cox model, where the latent variables $\mathbf{x} = (x_{ij})_{i,j=1,...,m}$ are associated with an $m \times m$ grid (Christensen, Roberts, and Rosenthal 2005; Girolami and Calderhead 2011). Assume that x_{ij}'s are normal with means 0 and a covariance function $C[(i,j),(i',j')] = \sigma^2 \exp(-\sqrt{(i-i')^2 + (j-j')^2}/(m\beta))$. By abuse of notation, we denote $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, C)$, of dimension $n = m^2$. The observations $(y_{ij})_{i,j=1,...,m}$ are independently Poisson, where the mean of y_{ij} is $\lambda_{ij} = n^{-1} \exp(x_{ij} + \mu)$, with μ treated as known. Hence, the unknown parameters are $\theta = (\sigma^2, \beta)^{\mathrm{T}}$. Given a prior $\pi(\theta)$, the posterior density is

$$p(\mathbf{x}, \theta | \mathbf{y}) \propto \pi(\theta) |\det(C)|^{-1/2} \exp\left\{-\frac{1}{2} x^{\mathrm{T}} C^{-1} x\right\} \times \exp\left\{\sum_{i,j} (y_{ij}(x_{ij} + \mu) - \lambda_{ij})\right\}. \quad (43)$$

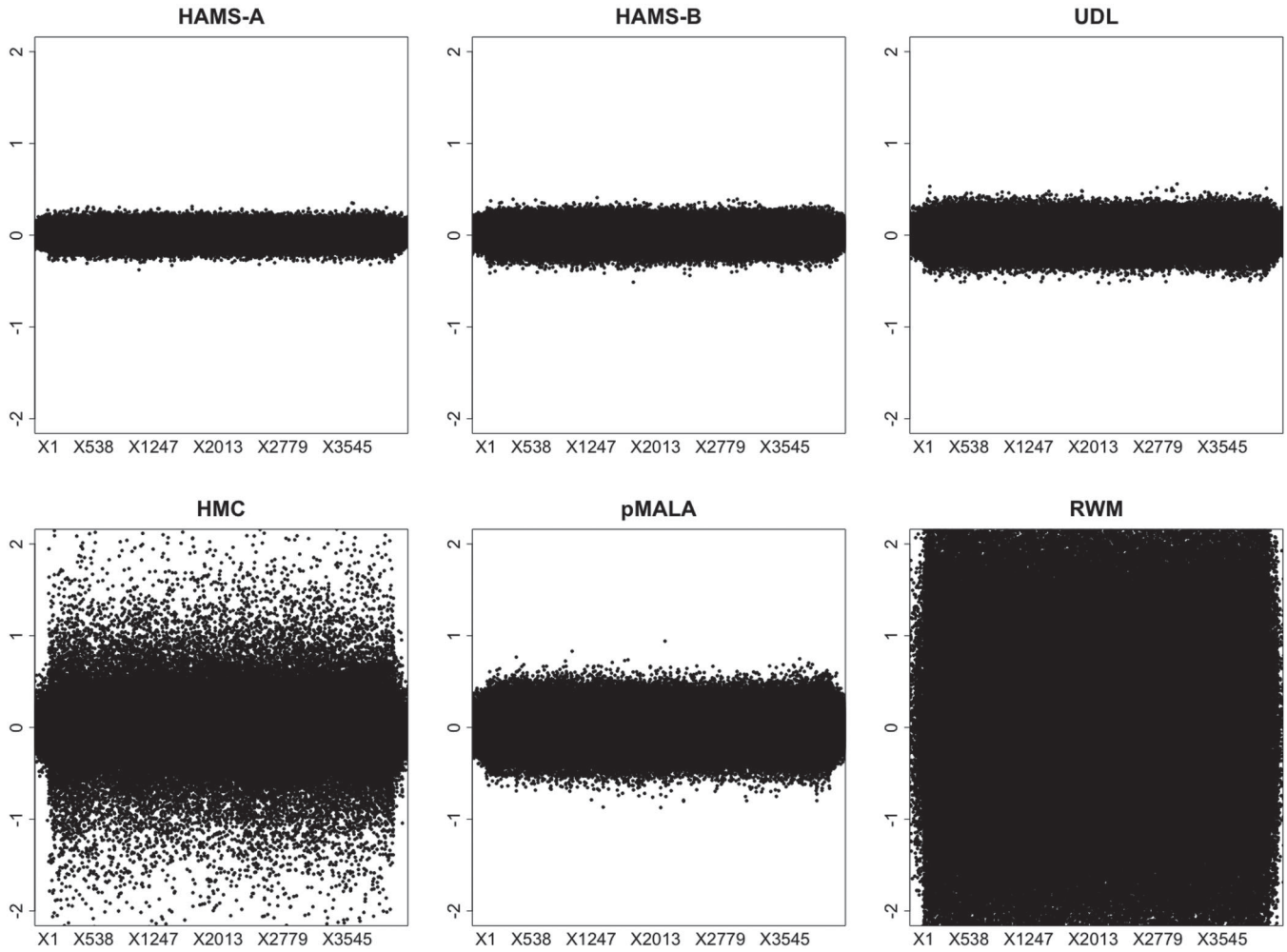


Figure 5. Time-adjusted and centered plots of sample means of all latent variables for sampling latent variables in the log-Gaussian Cox model.

As in Section 5.1, we conduct two sets of experiments: one is sampling latent variables with fixed parameters, and the other is sampling both parameters and latent variables. See Supplement Section V.3 for details of associated calculations.

For latent variables sampling, we take $m = 64$ and generate $n = 64^2 = 4096$ observations using the parameter values $\sigma^2 = 1.91$, $\beta = 1/33$, and $\mu = \log(126) - 0.5(1.91)$ as in Girolami and Calderhead (2011). From Equation (43), the gradient of the negative log-likelihood is $\nabla U(\mathbf{x}) = n^{-1}\exp(\mathbf{x}+\mu)+C^{-1}\mathbf{x}-\mathbf{y}$. The expected Hessian is $\mathrm{E}[\nabla^2 U(\mathbf{x})] = D+C^{-1}$, taken with respect to the prior of $\mathbf{x}$, where D is a diagonal matrix with diagonal elements $n^{-1}\exp(\mu+\frac{1}{2}\sigma^2)$. Hence, for preconditioning, we set $\Sigma^{-1} = M = D + C^{-1}$. The number of leapfrog steps is 50 for HMC. For all methods, 5000 draws are collected after a burn-in of 5000. The simulation process is repeated for 50 times.

Table 3 summarizes runtime and ESSs. Clearly, HAMS-A has the best performance in terms of time-adjusted minimum ESSs, followed by HAMS-B. Figure 4 shows time-adjusted trace plots of one latent variable and corresponding ACF plots from an individual run. From both plots, HAMS-A and HAMS-B appear to mix better than the other methods. Figure 5, similarly as Figure 2, shows the time-adjusted and centered plots of sample means for each method over 50 repetitions. The spreads corroborate the ESS results: HAMS-A and HAMS-B are more consistent than the remaining methods over repeated simulations.

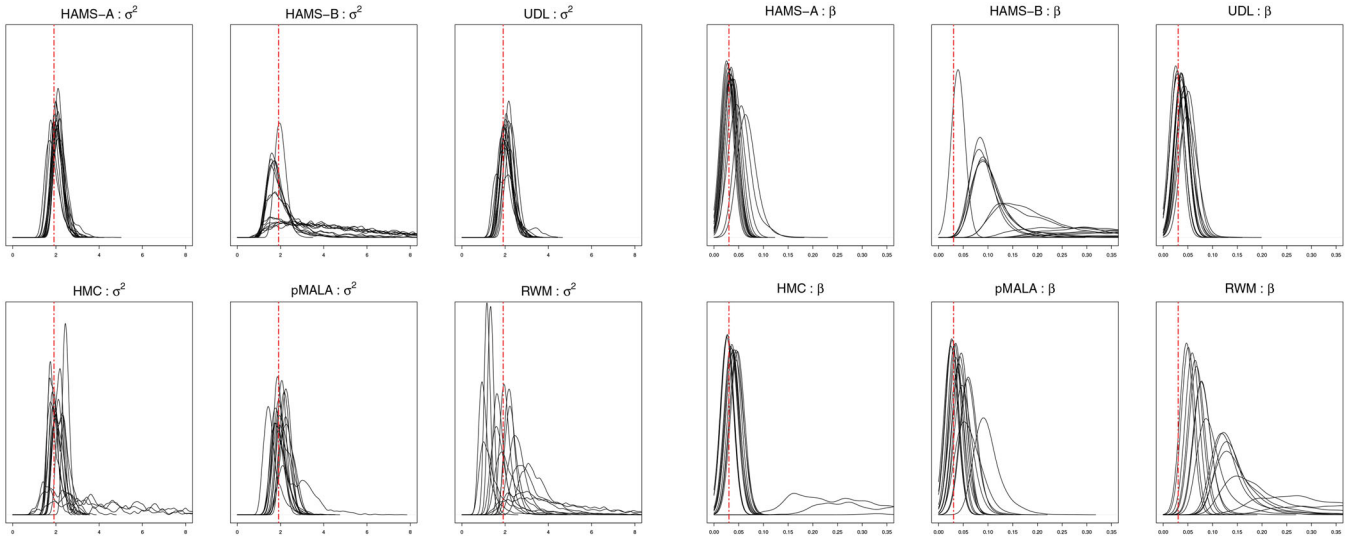
Similarly as in Section 5.1, the superior performances of HAMS-A/B are related to the Gaussian-calibrated rejection-free property, which facilitates use of relatively large step sizes while reasonable acceptance rates are obtained. See Figure S24 (supplementary material).

Our final experiment is sampling both latent variables and parameters for Bayesian analysis. The grid size remains 64×64 and the dimension of latent variables is $n = 4096$. We still simulate observations $\mathbf{y}$ using the ground truth $\sigma^2 = 1.91$, $\beta = 1/33$, and $\mu = \log(126) - 0.5(1.91)$. The priors are $\sigma^2, \beta \sim \mathrm{Gamma}(2, 0.5)$, independently. Then, we perform Gibbs sampling, alternating between $p(\mathbf{x}|\mathbf{y}, \theta)$ and $p(\theta|\mathbf{y}, \mathbf{x})$, after log transformations of (σ^2, β). HMC takes 50 leapfrog steps for latent variables and 6 for parameters. For each method, 5000 draws are collected after 9000 iterations, which include a stage without preconditioning and a burn-in period with preconditioning. The simulation process is repeated for 15 times using dispersed starting values for the parameters $\sigma^2 \in [1, 3]$ and $\beta \in [0.01, 0.2]$. Compared with the first experiment, the computational cost substantially increases here, mainly because the $n \times n$ matrix C needs to be inverted numerically whenever

Table 4. Comparison of posterior sampling in log-Gaussian Cox model.

Method	Time (1000 s)	Sample mean		ESS_1	$\frac{\text{ESS}_1}{\text{Time}}$	ESS_2	$\frac{\text{ESS}_2}{\text{Time}}$
		σ^2 (sd)	β (sd)	(σ^2, β)	(σ^2, β)	(σ^2, β)	(σ^2, β)
HAMS-A	161.1	2.08 (0.086)	0.04 (0.010)	(178, 24)	(1.105, 0.147)	(10.9, 1.0)	(0.068, 0.006)
HAMS-B	161.0	3.26 (1.309)	0.57 (0.560)	(554, 468)	(3.441, 2.908)	(2.4, 0.7)	(0.015, 0.005)
UDL	161.0	2.11 (0.109)	0.04 (0.006)	(109, 31)	(0.677, 0.190)	(7.7, 2.0)	(0.048, 0.012)
GMC	160.8	2.18 (0.112)	0.03 (0.007)	(91, 31)	(0.566, 0.194)	(8.0, 1.2)	(0.050, 0.007)
HMC	1366.9	2.45 (0.850)	0.21 (0.436)	(342, 375)	(0.250, 0.274)	(1.9, 0.4)	(0.001, 0.0003)
pMALA	162.5	2.08 (0.207)	0.04 (0.019)	(75, 17)	(0.462, 0.103)	(2.8, 0.4)	(0.017, 0.003)
pMALA*	162.4	1.97 (0.092)	0.05 (0.029)	(116, 32)	(0.714, 0.195)	(19.0, 0.7)	(0.117, 0.004)
RWM	82.5	2.42 (1.074)	0.16 (0.158)	(304, 279)	(3.685, 3.377)	(1.1, 0.6)	(0.013, 0.007)

NOTE: The runtimes reported are in 10^3 seconds. Standard deviations of sample means are in parentheses. Results are averaged over 15 repetitions.



(a) Densities of σ^2 (b) Densities of β

Figure 6. Time-adjusted posterior density plots (15 repetitions overlaid) in log-Gaussian Cox model. The true parameter values are marked by vertical lines.

the density (43) and its gradient are evaluated at new parameters in Gibbs sampling. The experiment with 15 repeated runs took 16 days on high-performance computing (HPC) clusters (120 Intel Haswell cores with 120 gigabytes of memory) at Rutgers University.

Table 4 summarizes the results of posterior sampling. For both parameters, HAMS-B, HMC, and RWM not only show large variations but their average posterior means deviate far away from the data-generating values. The relatively high ESS_1 values from HAMS-B and RWM should be interpreted with caution, because ESS_1 is determined from the auto-correlations in each run separately and cannot capture the inconsistency between repeated runs. Moreover, pMALA and pMALA* show large variations in the posterior means for σ^2 and β, respectively. Hence, only HAMS-A, UDL, and GMC give posterior means with reasonable averages and standard deviations. Among them, in terms of σ^2, HAMS-A yields the highest ESSs and has the smallest standard deviation. In terms of β, UDL, and GMC are comparable to each other, and both lead HAMS-A slightly. Figure 6 shows time-adjusted overlaid density plots for the parameters. The curves from HAMS-A and UDL are among the most consistent, whereas there are noticeably outlying curves from HAMS-B, HMC and RWM. This comparison is in agreement with that from Table 4.

6. Conclusion

We propose a broad class of HAMS algorithms and develop two specific algorithms, HAMS-A/B. From our numerical experiments, HAMS-A/B demonstrate consistent and sometimes substantial advantages over existing methods. The performance of HAMS-A is consistently among the best for sampling latent variables only or posterior sampling of parameters and latent variables, whereas that of HAMS-B varies for posterior sampling but remains among the best for sampling latent variables only. In addition, alternative HAMS algorithms can be derived by using two noise vectors per iteration. It is interesting to further study these algorithms, together with other algorithms related to underdamped Langevin dynamics. The improvement from HAMS may depend on the preconditioning schemes used, and further comparison is desired in settings where preconditioning is more difficult than in our experiments. Finally, our framework of generalized Metropolis–Hastings can also be exploited to develop other possible irreversible sampling algorithms.

Acknowledgments

The authors acknowledge the Office of Advanced Research Computing at Rutgers University for providing access to computing resources for the numerical studies reported here. The authors also thank two referees for helpful comments.

Supplementary Material

Appendices: (I) Auxiliary variable derivation, (II) Validity of UDL, (III) Generalized Metropolis-Hastings sampling, (IV) Proofs, (V) Details for simulation studies, (VI) Additional simulation results. (pdf).

Computer codes: R and Python codes for simulation studies in Section 5 and Supplement Section VI. (zip).

References

Adler, S. L. (1981), "Over-Relaxation Method for the Monte Carlo Evaluation of the Partition Function for Multiquadratic Actions," *Physical Review D*, 23, 2901–2904. [5]

Bartlett, M. S. (1948), "Smoothing Periodograms From Time-Series With Continuous Spectra," *Nature*, 161, 686–687. [12]

Besag, J. E. (1994), "Comments on "Representations of knowledge in complex systems" by U. Grenander and M.I. Miller," *Journal of the Royal Statistical Society*, Series B, 56, 591–592. [1]

Bierkens, J., Fearnhead, P., and Roberts, G. (2019), "The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data," *Annals of Statistics*, 47, 1288–1320. [2]

Blanes, S., Casas, F., and Sanz-Serna, J. M. (2014), "Numerical Integrators for the Hybrid Monte Carlo Method," *SIAM Journal of Scientific Computing*, 36, A1556–A1580. [9]

Bouchard-Cote, A., Vollmer, S. J., and Doucet, A. (2018), "The Bouncy Particle Sampler: A Nonreversible Rejection-Free Markov Chain Monte Carlo Method," *Journal of the American Statistical Association*, 113, 855–867. [2]

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011), *Handbook of Markov Chain Monte Carlo*, Boca Raton: CRC Press. [1]

Bussi, G., and Parrinello, M. (2007), "Accurate Sampling Using Langevin Dynamics," *Physical Review E*, 75, 056707. [2,4,6,9]

Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. (2018), "Underdamped Langevin MCMC: A Non-Asymptotic Analysis," in *Proceedings of the 31st Conference On Learning Theory*, Vol. 75, pp. 300–323. [4]

Christensen, O. F., Roberts, G. O., and Rosenthal, J. S. (2005), "Scaling Limits for the Transient Phase of Local Metropolis–Hastings Algorithms," *Journal of the Royal Statistical Society*, Series B, 67, 253–268. [16]

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013), "MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster," *Statistical Science*, 28, 424–446. [1,2,3]

Dalalyan, A. S., and Riou-Durand, L. (2020), "On Sampling From a Log-Concave Density Using Kinetic Langevin Diffusions," *Bernoulli*, 26, 1956–1988. [4]

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987), "Hybrid Monte Carlo," *Physics Letters B*, 195, 216–222. [1,3]

Duncan, A. B., Pavliotis, G. A., and Zygalakis, K. C. (2017), "Nonreversible Langevin Samplers: Splitting Schemes, Analysis and Implementation," arXiv:1701.04247. [2]

Fang, Y., Sanz-Serna, J. M., and Skeel, R. D. (2014), "Compressible Generalized Hybrid Monte Carlo," *Journal of Chemical Physics*, 140, 174108. [4,6]

Gamerman, D. (1997), "Sampling From the Posterior Distribution in Generalized Linear Mixed Models," *Statistics and Computing*, 7, 57–68. [15]

Gardiner, C. (1997), *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, New York: Springer. [2,6]

Gelman, A., and Hill, J. (2007), *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge: Cambridge University Press. [12,15]

Girolami, M., and Calderhead, B. (2011), "Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods," *Journal of the Royal Statistical Society*, Series B, 73, 123–214. [12,16,17]

Goga, N., Rzepiela, A. J., de Vries, A. H., Marrink, S. J., and Berendsen, H. J. C. (2012), "Efficient Algorithms for Langevin and DPD Dynamics," *Journal of Chemical Theory and Computation*, 8, 3637–3649. [4]

Grønbech-Jensen, N., and Farago, O. (2013), "A Simple and Effective Verlet-Type Algorithm for Simulating Langevin Dynamics," *Molecular Physics*, 111, 983–991. [4,7]

——— (2020), "Defining Velocities for Accurate Kinetic Statistics in the Grønbech-Jensen Farago Thermostat," *Physical Review E*, 101, 022123. [4,7]

Gustafson, P. (1998), "A Guided Walk Metropolis Algorithm," *Statistics and Computing*, 8, 357–364. [2]

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109. [1]

Hoffman, M. D., and Gelman, A. (2014), "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, 15, 1593–1623. [3]

Horowitz, A. M. (1991), "A Generalized Guided Monte Carlo Algorithm," *Physics Letters B*, 268, 247–252. [2,4]

Law, K. J. H. (2014), "Proposals Which Speed Up Function-Space MCMC," *Journal of Computational and Applied Statistics*, 262, 127–138. [5]

Liu, J. (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer. [1]

Ma, Y.-A., Fox, E., Chen, T., and Wu, L. (2018), "Irreversible Samplers From Jump and Continuous Markov Processes," *Statistics and Computing*, 29, 177–202. [2,4,10,11,12]

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1092. [1]

Neal, R. M. (1998), "Suppressing Random Walks in Markov Chain Monte Carlo Using Ordered Overrelaxation," in *Learning in Graphical Models*, ed. M. I. Jordan. pp. 205–228. The Netherlands: Springer. [5]

Neal, R. M. (2011), "MCMC Using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*, Chapter 5, Boca Raton: CRC Press. [1,3,4,9]

Ohzeki, M., and Ichiki, A. (2015), "Mathematical Understanding of Detailed Balance Condition Violation and Its Application to Langevin Dynamics," *Journal of Physics: Conference Series*, 638, 012003. [2]

Osawa, H. (1988), "Reversibility of First-Order Autoregressive Processes," *Stochastic Processes and their Applications*, 28, 61–69. [5]

Ottobre, M., Pillai, N. S., Pinski, F. J., and Stuart, A. M. (2016), "A Function Space HMC Algorithm With Second Order Langevin Diffusion Limit," *Bernoulli*, 22, 60–106. [2,4]

Roberts, G. O., and Tweedie, R. L. (1996), "Exponential Convergence of Langevin Distributions and Their Discrete Approximations," *Bernoulli*, 2, 341–363. [1,2]

Scemama, A., Lelièvre, T., Stoltz, G., Cancès, E., and Caffarel, M. (2006), "An Efficient Sampling Algorithm for Variational Monte Carlo," *Journal of Chemical Physics*, 125, 114105. [4,6]

Suwa, H., and Todo, S. (2012), "General Construction of Irreversible Kernel in Markov Chain Monte Carlo," arXiv:1207.0258. [2]

Syed, S., Bouchard-Cote, A., Deligiannidis, G., and Doucet, A. (2019), "Non-Reversible Parallel Tempering: A Scalable Highly Parallel MCMC Scheme," arXiv:1905.02939. [2]

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *Annals of Statistics*, 22, 1701–1728. [10]

Titsias, M. K., and Papaspiliopoulos, O. (2018), "Auxiliary Gradient-Based Sampling Algorithms," *Journal of the Royal Statistical Society*, Series B, 80, 749–767. [3,5]

van Gunsteren, W., and Berendsen, H. (1982), "Algorithms for Brownian Dynamics," *Molecular Physics*, 45, 637–647. [3]

Vucelja, M. (2016), "Lifting — A Nonreversible Markov Chain Monte Carlo Algorithm," *American Journal of Physics*, 84, 958–968. [2]