Noname manuscript No.
(will be inserted by the editor)

Modified Hamiltonian Monte Carlo for Bayesian Inference

Tijana Radivojević · Elena Akhmatskaya

Received: date / Accepted: date

Abstract The Hamiltonian Monte Carlo (HMC) method has been recognized as a powerful sampling tool in computational statistics. We show that performance of HMC can be significantly improved by incorporating importance sampling and an irreversible part of the dynamics into a chain. This is achieved by replacing Hamiltonians in the Metropolis test with modified Hamiltonians, and a complete momentum update with a partial momentum refreshment. We call the resulting generalized HMC importance sampler—Mix & Match Hamiltonian Monte Carlo (MMHMC). The method is irreversible by construction and further benefits from (i) the efficient algorithms for computation of modified Hamiltonians; (ii) the implicit momentum update procedure and (iii) the multi-stage splitting integrators specially derived for the methods sampling with modified Hamiltonians. MMHMC has been implemented, tested on the popular statistical models and compared in sampling efficiency with HMC, Riemann Manifold Hamiltonian Monte Carlo, Generalized Hybrid Monte Carlo, Generalized Shadow Hybrid Monte Carlo, Metropolis Adjusted Langevin Algorithm and Random Walk Metropolis-Hastings. To make a fair comparison, we propose a metric that accounts for correlations among samples and weights, and can be readily used for all methods which generate such samples. The experiments reveal the superiority of MMHMC over popular sampling techniques, especially in solving high dimensional problems.

Keywords Bayesian inference · Markov chain Monte Carlo · Hamiltonian Monte Carlo · importance sampling · modified Hamiltonians

PACS 02.50.Ng · 02.70.-c · 07.05.Tp

T. Radivojević
Basque Center for Applied Mathematics
Mazarredo 14,
Bilbao, 48009, Spain
E-mail: tijana.radivojevic@gmail.com
Present address: Biological Systems and Engineering Division,
Lawrence Berkeley National Laboratory
Berkeley, CA 94720, USA
DOE Agile Biofoundry
5885 Hollis Street, Emeryville, CA 94608, USA

E. Akhmatskaya
Basque Center for Applied Mathematics
Mazarredo 14,
Bilbao, 48009, Spain
IKERBASQUE, Basque Foundation for Science
María Díaz de Haro 3, E-48013
Bilbao, Spain
E-mail: akhmatskaya@bcamath.org

1 Introduction

Despite the complementary nature, Hamiltonian dynamics and Metropolis Monte Carlo had never been considered jointly until the *Hybrid Monte Carlo* method was formulated in the seminal paper by Duane et al. (1987). It was originally applied to lattice field theory simulations and remained unknown for statistical applications till 1994, when R. Neal used the method in neural network models (Neal, 1994). Since then, the common name in statistical applications is *Hamiltonian Monte Carlo* (HMC). The practitioners-friendly guides to HMC were provided by Neal (2011) and Betancourt (2017), while comprehensive geometrical foundations were set by Betancourt et al. (2017). The conditions under which HMC is geometrically ergodic are also established (Livingstone et al., 2016).

Nowadays, HMC is used in a wide range of applications—from molecular simulations to statistical problems appearing in many fields, such as ecology,

cosmology, social sciences, biology, pharmacometrics, biomedicine, engineering, business. The software packages Stan (Stan Development Team, 2017) and PyMC3 (Salvatier et al., 2016) have contributed to the increased popularity of the method through the implementation of HMC based sampling in a probabilistic modeling language to help statisticians writing their models in familiar notations.

For a range of problems in computational statistics the HMC method has proved to be a successful and valuable technique. The efficient use of gradient information of the posterior distribution allows it to overcome the random walk behavior typical of the Metropolis-Hastings Monte Carlo method.

On the other hand, the performance of HMC deteriorates, in terms of acceptance rates, with respect to the system's size and step size, due to errors introduced by numerical approximations (Izaguirre and Hampton 2004). Many rejections induce high correlations between samples and reduce the efficiency of the estimator. Thus, in systems with a large number of parameters, or latent parameters, or when the observations data set is very big, efficient sampling might require a substantial number of evaluations of the posterior distribution and its gradient. This may be computationally too demanding for HMC. In order to maintain the acceptance rate for larger systems at a high level, one could decrease a step size or use a higher order integrator, but both solutions are usually impractical for complex systems.

Ideally, one would like to have a sampling method that maintains high acceptance rates, achieves fast convergence, demonstrates good sampling efficiency and requires modest computational and tuning efforts.

To achieve some of those goals, several modifications of the HMC method have been recently developed in ***computational statistics*** (see Figure 1).

It is worth of mentioning here the methods employing a *position dependent 'mass' matrix* (Girolami and Calderhead, 2011; Betancourt, 2013a; Lan et al., 2015), *adaptive* HMC (Hoffman and Gelman, 2014; Betancourt, 2013b; Wang and de Freitas, 2011; Wang et al., 2013), HMC with the *approximated gradients* (Chen et al., 2014; Strathmann et al., 2015; Zhang et al., 2017a,b,c; Zou et al., 2018), *tempered* HMC (van de Meent et al., 2014; Betancourt, 2014; Graham and Storkey, 2017; Nishimura and Dunson, 2017; Luo et al., 2017), HMC with *alternative kinetic energy* (Zhang et al., 2016; Lu et al., 2017; Livingstone et al., 2017), *problem related* HMC (Betancourt, 2011; Brubaker et al., 2012; Lan et al., 2014b; Pakman and Paninski, 2013; Lan et al., 2014a; Betancourt and Girolami, 2015; Zhang and Sutton, 2014; Zhang et al., 2012; Afshar and Domke, 2015; Nishimura et al., 2018; Dinh et al., 2017; Yi and Doshi-Velez, 2017; Kleppe, 2018), *enhanced sampling* HMC (Sohl-Dickstein and Culpepper, 2012; Sohl-Dickstein et al., 2014; Campos and Sanz-Serna, 2015; Fu et al., 2016; Nishimura and Dunson, 2015; Zhang et al., 2018; Tripuraneni et al., 2017; Levy et al., 2018), and *special cases* of HMC, such as, Metropolis Adjusted Langevin Algorithm (Kennedy, 1990).

Among the modifications introduced in ***computational physical sciences***, the most important ones are *partial momentum update* and sampling with *modified energies* (Figure 1).

The partial momentum update (in contrast to the complete momentum update in HMC) was introduced by Horowitz (1991) within Generalized guided Monte Carlo, also known as the second order Langevin Monte Carlo (L2MC). The purpose of this method was to retain more dynamical information on a simulated system.

Kennedy and Pendleton (2001) formalized this idea in the Generalized Hybrid Monte Carlo (GHMC) method. GHMC is defined as the concatenation of two steps: Molecular Dynamics Monte Carlo and Partial Momentum Update.

Applications of the GHMC method to date include mainly molecular simulations. Behavior of non-special cases of GHMC are not well studied in statistical computations, with only a few exceptions (e.g. Sohl-Dickstein 2012; Sohl-Dickstein et al. 2014).

The idea of using the modified (shadow) Hamiltonian for sampling in HMC was suggested by Izaguirre and Hampton (2004). The performance of the resulting Shadow Hybrid Monte Carlo (SHMC) is limited by the need for a finely tuned parameter introduced for controlling the difference in the true and modified Hamiltonians and for the evaluation of a non-separable modified Hamiltonian. The SHMC was modified by Sweet et al. (2009) through replacing a non-separable shadow Hamiltonian with the separable 4th order shadow Hamiltonian to result in Separable Shadow Hybrid Monte Carlo (S2HMC).

The first method to incorporate both, the partial momentum update and sampling with respect to a modified density, was introduced by Akhmatskaya and Reich (2006) and called Targeted Shadow Hybrid Monte Carlo (TSHMC). However, the Generalized Shadow Hybrid Monte Carlo (GSHMC) method formulated by Akhmatskaya and Reich (2008) appears the most efficient (Wee et al. 2008; Akhmatskaya et al. 2009, 2011; Akhmatskaya and Reich 2012) among the methods, which sample with modified Hamiltonians and are often referred to as Modified Hamiltonian Monte Carlo (MHMC) methods (Akhmatskaya et al., 2017).

The potential advantage of GSHMC compared to HMC is the enhanced sampling resulting from: (i) higher

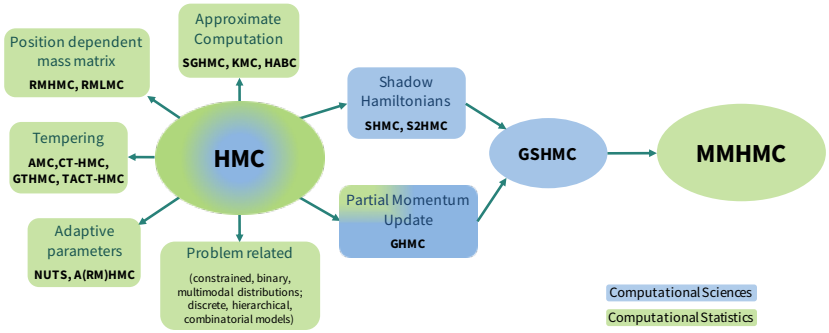

Fig. 1: Evolution and relationships between some variants of the HMC methods.

acceptance rates, achieved due to better conservation of modified Hamiltonians than Hamiltonians by symplectic integrators; (ii) an access to second-order information about the target distribution; (iii) an additional tunable parameter for improving performance; and (iv) irreversibility. The latter property of the method has never been mentioned whatsoever. Nevertheless, there is a great evidence that irreversible samplers may provide better mixing properties than their reversible counterparts do (Ottobre 2016). On the other hand, potential disadvantages of GSHMC include an extra parameter to tune and the computational overhead due to repetitive evaluations of modified Hamiltonians and a momentum update Metropolis function.

The efficiency of GSHMC method in solving statistical inference problems has never been investigated although its applicability has been recognized (Akhmatskaya and Reich 2012).

In this paper, we present the Mix & Match Hamiltonian Monte Carlo (MMHMC) method which is based on the GSHMC method but modified, enriched with the new features and adapted specially to computational statistics. The modifications of GSHMC that led to the MMHMC method include:

- a new formulation of the importance sampling distribution relying on the modified Hamiltonians for splitting integrating schemes;
- numerical integration of Hamiltonian dynamics using novel multi-stage integrators, specifically derived for improving conservation of modified Hamiltonians in the MHMC methods;
- an incorporation of momentum updates in the Metropolis test for a less frequent calculation of derivatives.

Additionally, we propose a new metric for measuring sampling efficiency of methods which generate samples that are both correlated and weighted.

We implemented MMHMC in our software package HaiCS, which also offers implementation of several other HMC based samplers as well as a range of popular statistical models.

The paper is structured as follows. We start with the summary of the Hamiltonian Monte Carlo method in Section 2.1. The MMHMC method is formulated in Section 2.2 and its essential features are reviewed in Section 2.3. The ways of tuning and measuring performance of MMHMC are discussed in Section 2.4. The expected performance of the method is analyzed in Section 2.5. The details of software implementation and testing procedure as well as the test results obtained for MMHMC and compared with various popular sampling techniques are presented in Section 3. The conclusions are summarized in Section 4.

2 Mix & Match Hamiltonian Monte Carlo (MMHMC)

Before introducing and analyzing Mix & Match Hamiltonian Monte Carlo we briefly revise the Hamiltonian Monte Carlo method.

2.1 Hamiltonian Monte Carlo: Essentials

The purpose of HMC is to sample a random variable (r. v.) $\boldsymbol{\theta} \in \mathbb{R}^D$ with the distribution $\pi(\boldsymbol{\theta})$, or to estimate integrals of the form

$$I = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{1}$$

We use the same notation π for the probability density function (p.d.f.), which can be written as

$$\pi(\boldsymbol{\theta}) = \frac{1}{Z}\exp(-U(\boldsymbol{\theta})),$$

where the variable $\boldsymbol{\theta}$ corresponds to the position vector, $U(\boldsymbol{\theta})$ to the potential function of a Hamiltonian system and Z is the normalizing constant such that $\pi(\boldsymbol{\theta})$

integrates to one. In Bayesian framework, the target distribution $\pi(\boldsymbol{\theta})$ is the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$ of unknown parameters given data $\mathbf{y} = \{y_1, \ldots, y_K\}$, K is the size of the data, and the potential function can be defined as

$$U(\boldsymbol{\theta}) = -\log L(\boldsymbol{\theta}|\mathbf{y}) - \log p(\boldsymbol{\theta}),$$

for the likelihood function $L(\boldsymbol{\theta}|\mathbf{y})$ and prior p.d.f. $p(\boldsymbol{\theta})$ of model parameters.

The auxiliary momentum variable $\mathbf{p} \in \mathbb{R}^D$, conjugate to and independent of the vector $\boldsymbol{\theta}$ is typically drawn from a normal distribution

$$\mathbf{p} \sim \mathcal{N}(0, M), \tag{2}$$

with a covariance matrix M, which is positive definite and often diagonal. The Hamiltonian function can be defined in terms of the target p.d.f. as the sum of the potential function $U(\boldsymbol{\theta})$ and the kinetic function $K(\mathbf{p})$

$$\begin{aligned} H(\boldsymbol{\theta}, \mathbf{p}) &= U(\boldsymbol{\theta}) + K(\mathbf{p}) \\ &= U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + \frac{1}{2}\log\left((2\pi)^D |M|\right). \end{aligned} \tag{3}$$

The joint p.d.f. is then

$$\begin{aligned} \pi(\boldsymbol{\theta}, \mathbf{p}) &= \frac{1}{Z}\exp(-H(\boldsymbol{\theta}, \mathbf{p})) \\ &= \frac{(2\pi)^{\frac{D}{2}}|M|}{Z}\exp(-U(\boldsymbol{\theta}))\exp(-\frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}). \end{aligned} \tag{4}$$

By simulating a Markov chain with the invariant distribution (4) and marginalizing out momentum variables, one recovers the target distribution $\pi(\boldsymbol{\theta})$. The integral (1) can then be estimated using N simulated samples as

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{\theta}^n).$$

HMC samples from $\pi(\boldsymbol{\theta}, \mathbf{p})$ by alternating a step for a momentum refreshment and a step for a joint, position and momentum, update, for each Monte Carlo iteration. In the first step, momentum is replaced by a new draw from the normal distribution (2). In the second step, a proposal for the new state $(\boldsymbol{\theta}', \mathbf{p}')$ is generated by integrating Hamiltonian dynamics

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = M^{-1}\mathbf{p}, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{5}$$

for L steps using a symplectic integrator Ψ_h with a step size h. Due to the numerical approximation of integration, Hamiltonian function, and thus the density (4), are not preserved. In order to restore this property, which ensures invariance of the target density, an accept-reject step is added through a Metropolis criterion. The acceptance probability has a simple form

$$\alpha = \min\left\{1, \exp\left(H(\boldsymbol{\theta}, \mathbf{p}) - H(\boldsymbol{\theta}', \mathbf{p}')\right)\right\},$$

which, due to the preservation of volume, does not include potentially difficult to compute Jacobians of the mapping. As in any Markov chain Monte Carlo (MCMC) method, in case of a rejection, the current state is stored as a new sample. Once next sample is obtained, momentum is replaced by a new draw, so Hamiltonians have different values for consecutive samples. This means that samples are drawn along different level sets of Hamiltonians, which actually makes HMC an efficient sampler.

For a constant matrix M, the last term in the Hamiltonian (3) is a constant that cancels out in the Metropolis test. Therefore, the Hamiltonian can be defined as

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}. \tag{6}$$

The algorithmic summary of the HMC method is given in Appendix D.

2.2 Formulation of MMHMC

As HMC, the MMHMC method aims at sampling unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ with the distribution (known up to a normalizing constant)

$$\pi(\boldsymbol{\theta}) \propto \exp(-U(\boldsymbol{\theta})).$$

However, this is achieved indirectly, as shown in Figure 2.

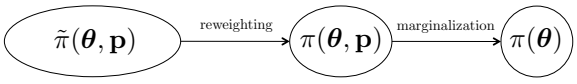

Fig. 2: MMHMC indirect sampling of the target distribution.

More precisely, MMHMC performs HMC importance sampling on the joint state space of positions and momenta $(\boldsymbol{\theta}, \mathbf{p})$ with respect to the modified density $\tilde{\pi}$. The target distribution on the joint state space $\pi(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-H(\boldsymbol{\theta}, \mathbf{p}))$, with respect to the true Hamiltonian H, is recovered through importance reweighting and finally, the desired distribution $\pi(\boldsymbol{\theta})$ is retrieved by marginalizing momenta variables. The MMHMC algorithm consists of three major steps: (1) Hamiltonian Dynamics Monte Carlo (HDMC) step to generate the next state, (2) Partial Momentum Monte Carlo (PMMC) step to refresh a momentum for each state, and (3) importance reweighting to recover the target distribution. The essential constituents of the algorithm are explained below.

2.2.1 Importance Distribution

The importance distribution in MMHMC ought to satisfy two principal requirements. First, it should lead to more favourable values of the acceptance probability than may be achieved in the HMC algorithm. Second, the target density and the importance density have to be close to maintain a low variability among weights, essential for efficient sampling. The so called modified Hamiltonian is a promising candidate for serving these purposes.

Given Hamiltonian dynamics with Hamiltonian function H (6) and a symplectic integrator with an integration step size h for solving consequent ODE equations (5), the corresponding modified equations are guaranteed to be Hamiltonian and the modified Hamiltonian can be determined as (Hairer et al. 2006)

$$\tilde{H}_h = H + hH_2 + h^2 H_3 + \cdots . \tag{7}$$

In contrast to the Hamiltonian, the modified Hamiltonian is exactly preserved along the computed trajectory by symplectic integrators (Leimkuhler and Reich 2005). For an integrator of order m ($m \geq 2$),

$$\tilde{H}_h = H + \mathcal{O}(h^m).$$

For the k-truncation of $\tilde{H}_h$ ($k > m$) defined as

$$\tilde{H}_h^{[k]} = H + ... + h^m H_{m+1} + \cdots + h^{k-1} H_k, \tag{8}$$

one obtains

$$\tilde{H}_h^{[k]} = H + \mathcal{O}(h^k), \tag{9}$$

and hence, a symplectic method preserves the k-truncated modified Hamiltonian up to order h^k. The expectation of the increment of $\tilde{H}_h^{[k]}$ in an integration leg satisfies

$$\mathbb{E}_{\tilde{\pi}}[\Delta \tilde{H}_h^{[k]}] = \mathcal{O}\left(Dh^{2k}\right), \tag{10}$$

with D being the dimension, while for the Hamiltonian it is

$$\mathbb{E}_{\pi}[\Delta H] = \mathcal{O}\left(Dh^{2m}\right) \tag{11}$$

(Beskos et al. 2013), and therefore the MMHMC algorithm may benefit from high acceptance rates due to better conservation of $\tilde{H}^{[k]}$.

The importance canonical density in MMHMC is then chosen as

$$\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-\tilde{H}_h^{[k]}(\boldsymbol{\theta}, \mathbf{p})). \tag{12}$$

For simplicity, we drop the subscript h and superscript $[k]$ in $\tilde{H}_h^{[k]}$ assuming an arbitrary choice of a truncation order. We shall return to the issue in the discussion of the specific formulations of the modified Hamiltonians associated with particular choices of a numerical integrator.

We notice that randomization of a step size commonly applied in HMC simulations is not compatible with the proposed importance distribution (12). On the one hand, randomization of a step size implies that a general modified equation does not exist, and thus the modified Hamiltonian can be constructed locally only, hence, the importance density has to be modified accordingly. On the other hand, randomization of a step size inevitably leads to the increased variability of weights, meaning the ultimate performance degradation of the importance sampling algorithm. Therefore, in MMHMC, the priority is given to a fixed step size. The advantages of this strategy are demonstrated in Section 2.5.

2.2.2 Hamiltonian Dynamics Monte Carlo (HDMC)

At every MC iteration, a proposal $(\boldsymbol{\theta}', \mathbf{p}')$ is generated by simulating Hamiltonian dynamics (5) using a symplectic and reversible numerical integrator ψ_h with a step size h, and is accepted with the Metropolis criterion corresponding to the modified distribution (12) as

$$(\boldsymbol{\theta}^{new}, \mathbf{p}^{new}) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{with probability } \alpha \\ \mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) & \text{otherwise,} \end{cases} \tag{13}$$

where $\alpha = \min\left\{1, \exp(-\Delta \tilde{H})\right\}$ and $\mathcal{F}(\boldsymbol{\theta}, \mathbf{p})$ flips the momentum in the case of rejection, i.e. $\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}, -\mathbf{p})$, and $\Delta \tilde{H} = \tilde{H}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}(\boldsymbol{\theta}, \mathbf{p})$.

An integrator ψ_h can be chosen arbitrarily from the class of symplectic and reversible integration schemes, though computationally efficient and accurate ψ_h are highly desirable for achieving the top performance of MMHMC. While GSHMC was formulated with the leapfrog integrator in mind, in MMHMC we employ multi-stage splitting schemes, proposed by Radivojević et al. 2018. More specifically, we consider numerical schemes belonging to the two-stage

$$\psi_h = \varphi^B_{bh} \circ \varphi^A_{\frac{h}{2}} \circ \varphi^B_{(1-2b)h} \circ \varphi^A_{\frac{h}{2}} \circ \varphi^B_{bh} \tag{14}$$

and three-stage

$$\psi_h = \varphi^B_{bh} \circ \varphi^A_{ah} \circ \varphi^B_{(\frac{1}{2}-b)h} \circ \varphi^A_{(1-2a)h} \circ \varphi^B_{(\frac{1}{2}-b)h} \circ \varphi^A_{ah} \circ \varphi^B_{bh} \tag{15}$$

families of splitting methods, which can offer better conservation properties than Verlet / leapfrog (Blanes et al. 2014). Here, the exact flows φ^A_h and φ^B_h are solutions to the split systems

$$A: \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -U_{\boldsymbol{\theta}}(\boldsymbol{\theta}),$$

and

$$B: \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = M^{-1}\mathbf{p}, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = 0,$$

respectively, corresponding to the Hamiltonian (6), and a, b are parameters of an integrator ψ_h, which will be discussed later.

2.2.3 Partial Momentum Monte Carlo (PMMC)

Whereas in HMC momentum is completely reset at each MC step before numerical integration, MMHMC relies on the partial refreshment of momentum. The idea behind the partial momentum update is to suppress the random walk behaviour arising from the complete, and hence independent from the current momentum, update. The PMMC can be performed in two steps.

First, for the current momentum $\mathbf{p}$ and a noise vector $\mathbf{u} \sim \mathcal{N}(0, M)$ a proposal for the new momentum $\mathbf{p}^*$ is generated from the mapping $\mathcal{R}: (\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) \mapsto (\boldsymbol{\theta}, \mathbf{p}^*, \mathbf{u}^*)$ such that

$$\mathcal{R}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) = (\boldsymbol{\theta}, \sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u}, -\sqrt{\varphi}\mathbf{p} + \sqrt{1-\varphi}\mathbf{u}). \tag{16}$$

Here, parameter $\varphi \in (0, 1]$ controls the amount of noise introduced in every MC iteration.

Then, to secure sampling from the modified density (12), the proposal is accepted according to the extended modified distribution

$$\hat{\pi} \propto \exp(-\hat{H}), \tag{17}$$

with the extended Hamiltonian $\hat{H}$ defined as

$$\hat{H}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) = \tilde{H}(\boldsymbol{\theta}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^\intercal M^{-1}\mathbf{u}. \tag{18}$$

Therefore, a new momentum can be determined as

$$\bar{\mathbf{p}} = \begin{cases} \mathbf{p}^* = \sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u} & \text{with probability} \\ & \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\} \\ \mathbf{p} & \text{otherwise,} \end{cases} \tag{19}$$

where $\Delta\hat{H} = \hat{H}(\boldsymbol{\theta}, \mathbf{p}^*, \mathbf{u}^*) - \hat{H}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})$.

Formulated in such a way, the PMMC step introduces two extra evaluations of the modified Hamiltonian within the Metropolis test and thus a computational overhead. To reduce the overhead, we incorporated a momentum proposal in the Metropolis test and derived the computationally tractable expressions for $\Delta\hat{H}$, for the particular choices of modified Hamiltonian, which we recommend to use in MMHMC. The details are provided in Section 2.3 and Appendix C.

2.2.4 Reweighting

After N iterations of the MMHMC algorithm, reweighting is required in order to estimate the integral (1). By making use of the standard technique for importance samplers, the integral is rewritten as

$$\begin{aligned} I = \mathbb{E}_\pi[f] &= \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta}, \mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} \\ &= \int f(\boldsymbol{\theta})\frac{\pi(\boldsymbol{\theta}, \mathbf{p})}{\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p})}\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} \\ &= \int f(\boldsymbol{\theta})w(\boldsymbol{\theta}, \mathbf{p})\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} = \mathbb{E}_{\tilde{\pi}}[fw], \end{aligned}$$

where $\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p})$ is the importance distribution (12) and $w(\boldsymbol{\theta}, \mathbf{p})$ the importance weight function. Therefore, the integral can be approximated by a self-normalized estimator as

$$\hat{I} = \frac{\sum_{n=1}^N f(\boldsymbol{\theta}^n)w_n}{\sum_{n=1}^N w_n}, \tag{20}$$

$$w_n = \exp(\tilde{H}(\boldsymbol{\theta}^n, \mathbf{p}^n) - H(\boldsymbol{\theta}^n, \mathbf{p}^n)), \tag{21}$$

where $\{(\boldsymbol{\theta}^n, \mathbf{p}^n)\}_{n=1}^N$ is drawn from $\tilde{\pi}$, and w_n are the corresponding weights.

Performance of importance sampling methods strongly depends on the discrepancy between the target and importance sampling distributions, and thus on weights. Bounded weights imply a bounded variance of an estimator. The choice of the importance distribution (12) in MMHMC along with (9) guarantee that the MMHMC weights are bounded and thus the reduction in efficiency of the estimator (20), introduced due to importance sampling, is minor in the case of the MMHMC method.

2.3 Features of MMHMC

In this section we discuss in more detail the specific features of the MMHMC method. The main algorithmic differences between HMC and MMHMC are listed in Table 1 and full algorithmic summary of MMHMC is provided in Appendix D.

2.3.1 Irreversibility

Until recently, the significant attention in the literature has been paid to the theoretical analysis of reversible Markov chains rather than the study of irreversible MCMC methods. However, numerous latest theoretical and numerical results demonstrate the advantage of irreversible MCMC over reversible algorithms both in terms of variance of an estimator and rates of convergence to the target distribution (Neal 2004; Suwa

Table 1: Algorithmic differences between HMC and MMHMC.

	HMC	MMHMC
Momentum update	complete	partial
Momentum Metropolis test	✗	✓
Metropolis test	H	$\tilde{H}$
Momentum flips	✗	✓
Re-weighting	✗	✓
Reversibility	✓	✗

and Todo 2012; Ohzeki and Ichiki 2015; Bouchard-Côté et al. 2018; Ottobre 2016; Duncan et al. 2016, 2017). These well documented facts have induced a design of new algorithms which break the detailed balance condition (DBC)—a commonly used criterion to demonstrate the invariance of the chain. Some recent examples of irreversible methods based on Hamiltonian dynamics can be found in papers by Ottobre (2016); Ottobre et al. (2016); Ma et al. (2016).

The core of the MMHMC algorithm consists of two steps, PMMC and HDMC, which both leave the target distribution $\tilde{\pi}$ invariant. However, the resulting chain is not reversible.

Apart from being invariant with respect to the target distribution, the HDMC step satisfies the modified DBC. The proof for the GHMC method can be found elsewhere (e.g. Fang et al. 2014), and the only difference in the case of MMHMC is that the target distribution, and thus the acceptance probability, is defined with respect to the modified Hamiltonian.

As the PMMC step is specific only to MMHMC and GSHMC, we provide a direct proof of invariance of this step (Appendix A). Furthermore, in an analogous way to HDMC, it can be proved that PMMC satisfies the modified DBC. The key observation is that the proposal mapping $\mathcal{R}$ for momenta (16) is reversible w.r.t. the extended target $\hat{\pi}$, $\mathcal{R}^{-1} = \hat{\mathcal{F}}^{-1} \circ \mathcal{R} \circ \hat{\mathcal{F}}$, and the reversing mapping $\hat{\mathcal{F}}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) := (\boldsymbol{\theta}, \mathbf{p}, -\mathbf{u})$ is an involution.

The irreversibility of MMHMC arises from an important property—a non-symmetric composition of steps satisfying DBC does not preserve DBC. Therefore, although both steps of MMHMC do satisfy the (modified) DBC, their compositon is not symmetric and hence, the chain generated by MMHMC is not reversible by construction.

2.3.2 Numerical Integrators

The detailed discussion on efficiency of various numerical integrators in the MHMC methods can be found elsewhere (Radivojević et al. 2018). Here we review the most promising integration schemes for the MMHMC method and provide some practical recommendations.

The Verlet/leapfrog integrator, considered as the integrator of choice for MHMC methods until recently, still can be seen as a perfect option for MMHMC in sampling small sized problems, where comparatively long step sizes are allowed. For such problems, Verlet is expected to demonstrate the highest conservation of modified Hamiltonians due to its best stability among splitting integrators. For bigger dimensions and thus for smaller optimal step sizes, the multi-stage integrators (14)–(15) designed specifically for MHMC and referred to as modified splitting integrators (Radivojević et al. 2018) should provide better conservation of modified Hamiltonian than the Verlet integrator, resulting in enhanced accuracy and sampling performance of MHMC methods.

Modified splitting integrators are characterized by values of parameters a and b in (14)–(15) obtained through minimization of the (expected) modified Hamiltonian error introduced by integration. Following the ideas of McLachlan (1995) and Blanes et al. (2014) for improving HMC performance by minimizing (expected) energy error through the appropriate choice of parameters of the integrator, the modified splitting integrators have been derived (Radivojević et al., 2018) by considering either the error in the modified Hamiltonians for splitting integrators, $\tilde{H}^{[l]}$, of order $l = 4, 6$

$$\Delta = \tilde{H}^{[l]}(\Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})) - \tilde{H}^{[l]}(\boldsymbol{\theta}, \mathbf{p}),$$

to yield the integrators M-ME2 and M-ME3, or the expected values of such errors $\mathbb{E}_{\tilde{\pi}}(\Delta)$ taken with respect to the modified canonical density $\tilde{\pi}$ (12) to give rise to the integrators M-BCSS2 and M-BCSS3. Here $\Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})$ is the hL-time map of the integrator.

In Table 2 we provide important characteristics of the integrators which can be recommended for the use in modified Hamiltonian Monte Carlo methods in general, and in MMHMC in particular, for a broad range of problems and methods' parameters.

Table 2: The splitting integrators for sampling with modified Hamiltonian Monte Carlo methods using 4th order modified Hamiltonians. Stability limit $h_{\max}$ is presented in terms of the three-stage family (Radivojević et al. 2018).

Integrator	N. of stages	Coefficients	$h_{\max}$
Verlet	1	–	6.000
M-BCSS2	2	$b = 0.238016$	4.144
M-ME2	2	$b = 0.230907$	4.089
M-BCSS3	3	$a = (1-2b)/4(1-3b)$ $b = 0.144115$	4.902
M-ME3	3	$a = (1-2b)/4(1-3b)$ $b = 0.142757$	4.887

2.3.3 Modified Hamiltonians

As in any modified Hamiltonian Monte Carlo (MHMC) method, in MMHMC, the importance distribution $\tilde{\pi}$ is ultimately defined through a modified Hamiltonian associated with a particular numerical integrator. In the early MHMC methods, various implementations of modified Hamiltonians for the Verlet/leapfrog integrator have been proposed and used. The idea to employ multi-stage integration splitting schemes in MHMC methods has been explored for the first time in the context of Mix & Match Hamiltonian Monte Carlo. Nevertheless, the derived formulations of modified Hamiltonians and parameters for corresponding integration schemes can be successfully used with other MHMC methods, as it has been discussed and demonstrated by Radivojević et al. (2018). In the following, we briefly review the formulations of the modified Hamiltonian for splitting integrators (Radivojević et al., 2018), which we recommend to use along with Mix & Match Hamiltonian Monte Carlo.

Two alternative formulations of the 4th and 6th order modified Hamiltonians corresponding to the Verlet integrator and multi-stage integrators (14)–(15) with arbitrary coefficients, have been proposed (Radivojević et al., 2018).

For problems in which analytical derivatives of the potential functions are available and inexpensive to compute, the 4th and 6th order modified Hamiltonians for splitting integrators can be calculated as

$$\begin{aligned}\tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) &= H(\boldsymbol{\theta},\mathbf{p}) + h^2 c_{21}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ &\quad + h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \qquad (22)\\ \tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) &= \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) \\ &\quad + h^4 c_{41} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ &\quad + h^4 c_{42} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ &\quad + h^4 c_{43} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ &\quad + h^4 c_{44}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}. \qquad (23)\end{aligned}$$

If the potential function is quadratic, i.e. corresponding to problems of sampling from Gaussian distributions, the 6th order modified Hamiltonian (23) simplifies to

$$\begin{aligned}\tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) &= \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) \\ &\quad + h^4 c_{43} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ &\quad + h^4 c_{44}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}. \qquad (24)\end{aligned}$$

The values of the coefficients c_{ij} in (22)–(24) for Verlet, two- and three-stage integrators are provided in Appendix B.

The alternative formulations of modified Hamiltonians address to problems with a dense Hessian matrix (and higher derivatives) and mainly rely on quantities that are available during a simulation (Radivojević et al., 2018). In this case, the 4th and 6th order modified Hamiltonians, respectively, are given as

$$\begin{aligned}\tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) &= H(\boldsymbol{\theta},\mathbf{p}) + hk_{21}\mathbf{p}^T M^{-1} P_1 \\ &\quad + h^2 k_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \qquad (25)\\ \tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) &= \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) \\ &\quad + hk_{41}\mathbf{p}^T M^{-1} P_3 + h^2 k_{42} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} P_2 \\ &\quad + h^2 k_{43} P_1^T M^{-1} P_1 \\ &\quad + h^4 k_{44} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \qquad (26)\end{aligned}$$

where the coefficients k_{ij} are provided in Appendix B. Here $P_i = \mathbf{U}^{(i)} \cdot h^i, i = 1,2,3$, and $\mathbf{U}^{(i)}$ are centered finite difference approximations of time derivatives of the gradient of the potential function (see Appendix B for further details).

We note that the expression (25) allows for computation of $\tilde{H}^{[4]}$ using quantities available from a simulation. Nevertheless, this is not the case for the resulting 6th order Hamiltonian. The last term in (26), arising from an expansion of the Poisson bracket $\{B,B,A,A,B\}$, cannot be computed using time derivatives of available quantities and requires explicit calculation of the Hessian matrix of the potential function. Only for the Verlet integrator does this term vanish and the resulting coefficients are

$$k_{21} = \frac{1}{12}, \quad k_{22} = -\frac{1}{24},$$
$$k_{41} = -\frac{1}{720}, \quad k_{42} = \frac{1}{240}, \quad k_{43} = \frac{11}{720}, \quad k_{44} = 0.$$

Finally, we remark that the presented formulations of modified Hamiltonians (22)–(24) and (25)–(26) were used to derive the computationally tractable expressions for the Metropolis function of the modified PMMC step proposed in MMHMC (see Appendix C).

2.4 Tuning and Measuring Performance of MMHMC

In this section, we first discuss the impact of the parameters of the MMHMC method on its performance. Secondly, we present the metrics for assessing the performance, which are specifically designed for the class of MHMC methods.

2.4.1 Choice of Parameters

MMHMC has five tunable parameters that affect the performance of the method—the integration step size

h, number of integration steps L, mass matrix M, noise parameter φ, and order k of the modified Hamiltonian. In principle, these parameters may be chosen arbitrarily within allowed-by-the-algorithm ranges, except for some special cases when they might affect the ergodicity of the chain (e.g. combinations leading to a value that is a multiple of the period of a mode of the system). However, the choice of parameters may have a dramatic impact on the overall performance of MMHMC, and thus tuning free parameters in order to maximize sampling efficiency and minimize computational costs is one of the most important but challenging tasks.

We notice that the first three parameters of MMHMC are the same as in HMC, and like for HMC, the optimal choice of these parameters in MMHMC is still an unresolved issue, though some recommendations and observations for both methods are available (Mackenzie 1989; Liu 2008; Neal 2011; Hoffman and Gelman 2014; Akhmatskaya and Reich 2008; Wee et al. 2008). Below we briefly discuss considerations and observations which are essential for choosing free parameters in MMHMC, while not necessarily relevant to HMC.

For example, the experiments revealed that the parameter L found to be the best for HMC is not necessarily the best for MMHMC. Actually, too long values of L may result in poorer overall efficiency of MMHMC at particular choices of φ, although the computational overhead is smaller with larger L, due to a less frequent calculation of modified Hamiltonians. In contrast, longer trajectories are needed for HMC to achieve its full potential, especially for larger dimensions. Intuitively, such a difference can be explained by the presence of a partial momentum update and high acceptance rates in MMHMC, which together, for small L, mimic as long or even longer, but more variative than in the case of large L, trajectory. On the contrary, a complete momentum update and short trajectories in HMC may initiate too frequent switches to not necessarily preferable directions.

We have to stress that the choice of a step size h critically affects the accuracy and sampling efficiency of MMHMC not only through its influence on acceptance rates (like in HMC) but also on importance weights (see Section 2.5). Indeed, the reduction in efficiency due to use of importance sampling is expected to be negligible for small values of h. The reason is a choice of the importance density $\tilde{\pi}$ in MMHMC, which stays closer to the true density π when h tends to 0. The larger values of step size may lead to a high variability in the importance weights and thus to a performance degradation. As a result, given a sampling problem, the best performance of MMHMC (often superior to the one accessible with HMC) may be achieved at step sizes smaller than the optimal ones for HMC.

Similarly to conventional HMC, the current implementation of MMHMC uses the identity mass matrix and offers different randomization schedules for a number of integration steps. In addition, a randomization of a noise parameter is provided in the algorithm. However, in contrast to HMC, in MMHMC a step size stays fixed on the reasons explained in Section 2.2.1.

The parameters φ and k are specific to MMHMC and are not used in HMC.

Noise parameter φ. Too small values of φ may reduce sampling efficiency by producing almost deterministic proposals, whereas too large φ may introduce a random walk effect or increase momenta rejection rates and thus lessen a potentially positive role of φ in tuning sampling performance.

In Figure 3, we report position and momenta acceptance rates (top) and sampling efficiency, in terms of time-normalized minimum ESS (bottom) in the problem of sampling from the 100-dimensional Gaussian distribution for different choices of the trajectory length hL and noise parameter φ. Two different schemes for treating the noise parameter φ are considered, namely (i) using a fixed value φ at every MC iteration, and (ii) choosing a random value uniformly from the interval $(0, \varphi)$.

The figure provides a good illustration of an effect of different parameters of MMHMC on the overall performance of the method. One immediately sees a positive influence of smaller values of the step size h and noise parameter φ on the sampling performance of MMHMC. The parameter L seems to play a less important role in the performance tuning. This also applies to the randomization of φ once the optimal value of φ is chosen ($\varphi = 0.1$). The situation changes when φ is far from its optimal value. In this case the randomization mitigates the effect of those unfavorable choices. We summarize the observations specific to a role of φ in the MMHMC performance below.

Position acceptance rate is not affected by φ, unless $\varphi = 1$ at which it slightly drops, whereas the acceptance rate of the PMMC step is visibly higher for smaller values of φ. Bigger values of φ, meaning more random noise introduced in momenta, might stimulate a better space exploration; however, those values lead to more frequent momenta rejections. In general, smaller values of φ result in better sampling efficiency, though this trend is more obvious for smaller trajectory lengths hL. A noticeable drop in efficiency appears for a fixed value $\varphi = 1$, however, randomization around 1 reduces the negative effect of complete momentum update.

The various numerical tests suggest that a random value from $(0, 0.5)$ drawn for every MC iteration is a safe initial guess for a good choice of the parameter φ. A

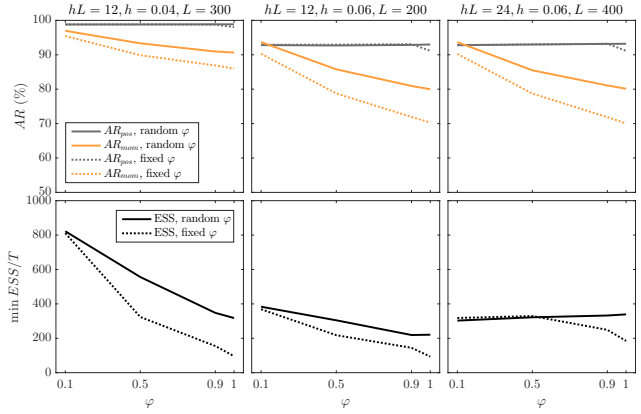

Fig. 3: Position and momenta acceptance rates (top) and time-normalized minimum ESS (bottom) obtained in sampling from the 100-dimensional Gaussian distribution using MMHMC with different choices of the trajectory length hL and noise parameter φ. For each MC iteration, the noise parameter is chosen to be either fixed (dashed line) or random, uniformly drawn from the interval $(0, \varphi)$ (solid line).

more theoretically grounded choice of a noise parameter is proposed by Akhmatskaya et al. (2017).

Finally, we note that different values of φ can be assigned to different variates—those that require longer trajectories to decorrelate could have bigger values of φ and those that do not, can use smaller values.

Eventually, an automatic choice of the above free parameters for optimal efficiency can be achieved by adapting the techniques from Wang et al. (2013) to MMHMC (Radivojević, 2016).

Order of modified Hamiltonian k. The decision on the order of modified Hamiltonian is not a problematic one. Our experiments indicate that the 4th order modified Hamiltonian combined with the multi-stage integrators performs just well. For more complex models, if the acceptance rate is low with the 4th order, the 6th order modified Hamiltonian might be needed. This comes at a higher computational cost; however, such complex models might require large values of L for which the computational overhead due to the calculation of modified Hamiltonian becomes negligible.

For more detailed discussion on the effect of free parameters on the MMHMC performance and accuracy, we refer the reader to (Radivojević, 2016).

2.4.2 Performance Metrics

To assess performance of the MMHMC method we use the following metrics:

- Acceptance rate (AR);
- Effective Sample Size (ESS) and ESS normalized by the computational time in seconds (ESS/T);
- Monte Carlo Standard Error (MCSE) and MCSE normalized by the computational time in seconds (MCSE·T);
- Efficiency Factor (EF)—relative ESS/T (MCSE·T) of MMHMC with respect to another algorithm.
- Total distance from the mean, defined as $||\boldsymbol{\theta} - \boldsymbol{\mu}|| = \sum_{d=1}^{D} |\hat{\theta}_d - \mu_d|$ for the true mean $\boldsymbol{\mu}$, and time-normalized total distance from the mean.

Effective Sample Size is a commonly used measure for sampling efficiency of an MCMC method. It indicates the number of effectively uncorrelated samples out of N collected samples and is defined as

$$ESS_{\text{MCMC}} = \frac{N}{1 + 2\sum_k \hat{\gamma}_k},$$

where $\hat{\gamma}_k$ is the k-lag sample autocorrelation (Geyer 1992).

Monte Carlo Standard Error of an estimator specifies how much error is in the estimate due to the use of a Monte Carlo method. It is related to ESS and is defined as

$$MCSE_{\text{MCMC}} = \sqrt{\frac{\hat{\sigma}^2}{ESS_{\text{MCMC}}}},$$

where $\hat{\sigma}^2$ is the sample variance.

For general importance sampling methods, high variability in the importance weights might occur if the importance density is not close enough to the target density. In this case ESS is calculated as

$$ESS_{\text{IS}} = \frac{\left(\sum_{n=1}^{N} w_n\right)^2}{\sum_{n=1}^{N} w_n^2},$$

where $w_n, n = 1, \ldots, N$ are weights associated to all samples, as first introduced by Kong et al. (1994).

For importance sampling methods such as GSHMC and MMHMC, one should use a metric for sampling efficiency that takes into account both correlations among samples and weights. To the best of our knowledge, a metric for samplers that generate correlated weighted samples has not been introduced, though the importance of such an objective criterion was discussed e.g. by Neal (2001); Gramacy et al. (2010).

Here we propose a new metric that addresses these issues and is based on calculation of ESS for MCMC and importance samplers jointly. More specifically, we first find the number of uncorrelated samples in the modified ensemble $M := ESS_{\text{MCMC}}$ using all N posterior samples collected. We estimate ESS_{MCMC} using the CODA package (Plummer et al. 2006). Then, we choose M samples out of N by thinning, i.e. at a distance of $\lceil N/M \rceil$. Finally, we calculate MCSE of the importance

sampling estimator $\hat{I} = \sum w_n f(\boldsymbol{\theta}^n) / \sum w_n$ for those M uncorrelated samples as

$$MCSE_{\text{MCMC-IS}} = \sqrt{\frac{\hat{\sigma}_w^2}{ESS_{\text{MCMC-IS}}}},$$

where $\hat{\sigma}_w^2$ is the unbiased weighted sample variance (Rimoldini 2014)

$$\hat{\sigma}_w^2 = \frac{\sum_{n=1}^M w_n}{(\sum_{n=1}^M w_n)^2 - \sum_{n=1}^M w_n^2} \sum_{n=1}^M w_n \left(f(\boldsymbol{\theta}^n) - \hat{I} \right)^2$$

and

$$ESS_{\text{MCMC-IS}} = \frac{\left(\sum_{n=1}^M w_n \right)^2}{\sum_{n=1}^M w_n^2} \tag{27}$$

is the effective sample size for samplers that generate weighted correlated samples. Note that the effective sample size depends directly on variability in the normalized importance weights.

Although in the numerical experiments through the paper for MCMC (HMC, GHMC, MALA, RMHMC) and MCMC importance sampling (GSHMC, MMHMC) methods we use the corresponding equations to compute ESS_{MCMC}, $\text{ESS}_{\text{MCMC-IS}}$ and $\text{MCSE}_{\text{MCMC}}$, $\text{MCSE}_{\text{MCMC-IS}}$, we simplify their notation to ESS and MCSE, respectively, in the remainder of the paper.

2.5 Expected Performance of MMHMC

By design, MMHMC incorporates the features and methods, known as potentially favourable for performance enhancement. Among them are irreversibility, importance sampling with modified Hamiltonians (implying high acceptance rates, bounded weights), integration of Hamiltonian dynamics using modified multi-stage splitting integrators (assuring high accuracy and acceptance rates), partial momentum refreshment (resulting in efficient sampling). On the other hand, implementation of such techniques in MMHMC introduces a computational overhead, and using importance sampling may potentially reduce the efficiency of the estimator. Contributions of those factors, positive or negative, into the overall performance of MMHMC are not equivalent, and in this section, we analyze potential performance gains and losses provoked by the most significant factors.

The main advantage of using an importance distribution defined through modified Hamiltonians comes from the fact that modified Hamiltonians are better preserved by symplectic integrators than true Hamiltonian (Leimkuhler and Reich 2005). A better conservation of modified Hamiltonians leads to a smaller error after numerical integration, which directly takes part in the Metropolis test (13) and results in higher acceptance rates. For illustration, in Figure 4 we compare the resulting numerical integration error Δ observed in the true Hamiltonian H and the 4th and 6th order modified Hamiltonians given by (22) and (24), respectively, for the 100-dimensional Gaussian problem. $\tilde{H}^{[4]}$ is significantly better conserved than H. Conservation of $\tilde{H}^{[6]}$ is even better, as expected. However in practice this must be weighted up against the computational cost of the calculation of the 6th order modified Hamiltonian (23) for general non-Gaussian problems, which includes higher order derivatives. In Section 3, we show that the combination of the computationally inexpensive 4th order modified Hamiltonians with accurate multi-stage splitting integrators makes a perfect choice in all numerical experiments, with no need for appealing to higher order expensive modified Hamiltonians.

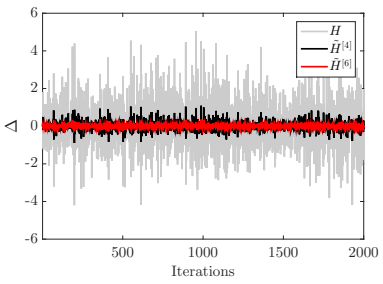

Fig. 4: Observed error in (modified) Hamiltonians after numerical integration with two-stage integrator in MMHMC sampling of a 100-dimensional Gaussian problem.

Another advantage of using modified Hamiltonians for importance sampling are bounded importance weights ensuring the efficiency of an estimator. Furthermore, avoiding randomization of a step size in MMHMC helps to maintain a low variability of importance weights. Figure 5 demonstrates the superiority of the fixed step strategy over randomization of a step size in MMHMC on the example of the D-dimensional Gaussian model.

It may be interesting to compare theoretical performance of HMC and MMHMC for high-dimensional problems. As follows from analysis in Eq. (10), in order to keep acceptance rates in HMC high, an increase in system size D can be counterbalanced by a decrease of a step size h or/and increase in the order m of the symplectic integrator used ($m \geq 2$, $m = 2$ for the Verlet integrator). However, smaller step sizes mean poorer space exploration. This can be partially overcome by increasing a length of the HD trajectory but at the price of reduced computational efficiency. We recall that longer trajectories in HMC imply more frequent

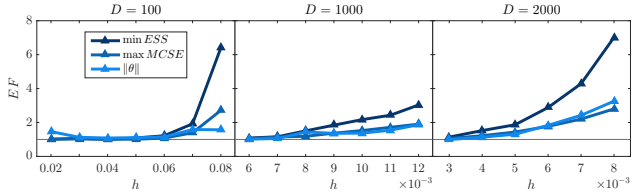

Fig. 5: Relative efficiency (EF) of MMHMC with fixed step size w.r.t. MMHMC with randomized step size in terms of minimum ESS, maximum MCSE and total distance from the mean, for a range of step size h and D-dimenasional Gaussian model.

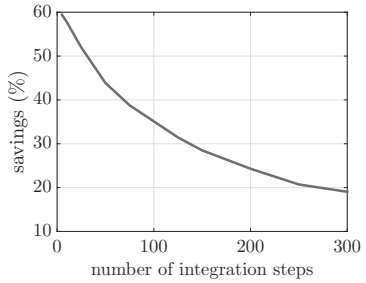

Fig. 6: Savings in computational time observed in MMHMC sampling of a model with dense Hessian matrix after replacing the original PMMC step with the newly proposed PMMC step. The 4th order modified Hamiltonian (22) with analytical derivatives was used.

time-consuming evaluations of gradients of the potential function. Using high-order symplectic numerical integrators is a possible but rather expensive way of keeping acceptance rates high as such integrators introduce a significant computational overhead.

For MMHMC (Eq. 11) the order of the modified Hamiltonian $k \geq 4$ ensures although shorter than for low-dimensional problems but longer than in HMC, step sizes for high dimensional systems. Moreover, the variability of weights, being a potential threat for MMHMC performance, is lower for smaller time steps, as follows from the definition of the truncated modified Hamiltonian (8).

There are two important reasons for using modified splitting integrators in the MMHMC method. One is their potential to achieve, for a range of step sizes, at a given computational cost, a higher accuracy than Verlet and thus higher acceptance rates and better space exploration. (We have to emphasize that for fair comparison, different integrators have to be applied with the same computational effort, rather than with the same step length; an r-stage integrator requires r gradient evaluations per time step and to be compared with Verlet has to be used with a step length correspondingly longer.)

A second possible benefit of the integrators of this class is that, due to the extra accuracy, they may avoid the need for computationally expensive, higher order modified Hamiltonians.

Numerical experiments confirm that the Verlet integrator currently used within HMC and MHMC methods can be advantageously replaced in MMHMC with modified multi-stage integrators whose implementation is essentially that of Verlet (Radivojević et al., 2018). The modified two- and three-stage integrators lead to an outstanding improvement (up to 8 times) over Verlet in terms of acceptance rate and sampling efficiency, for a range of step sizes, for high dimensional problems in which the potential function is (approximately) quadratic.

An introduction of the modified partial momentum update in MMHMC intends to reduce a computational overhead caused by the evaluation of modified Hamiltonians within the Metropolis test in the PMMC step. The proposed PMMC step is at least as efficient as the original momentum update implemented in GSHMC, whereas for specific choices of models and parameters it may demonstrate a far better computational performance that can be achieved with the original algorithm.

In Figure 6 we show the savings in computational time observed in MMHMC sampling of a model with a dense Hessian matrix after replacing the original PMMC step with the newly proposed one. The modified Hamiltonian (22) and the range of HD trajectories lengths have been considered in this case. Clearly, the new PMMC step improves the efficiency of MMHMC in sampling such models (up to 60%), especially if moderately short HD trajectories, favoured in MMHMC, are chosen.

The computational effort required for calculation of modified Hamiltonians in MMHMC is the crucial issue for the overall performance efficiency of MMHMC. In general, the higher orders modified Hamiltonians are more computationally demanding than the ones of the low orders. For models with a tridiagonal Hessian matrix, the modified Hamiltonians with analytical derivatives (22)–(24) introduce less computational overhead than those expressed in terms of numerical time derivatives (25)-(26), whereas for models with a dense Hessian matrix, the modified Hamiltonians (25)-(26) are less expensive than (22)–(24). As stated before, avoiding modified Hamiltonians of orders higher than 4 became possible with the introduction in MMHMC of accurate modified splitting integrators specifically tuned for the MHMC methods. Figure 7 shows computational overheads of MMHMC, compared to the HMC method, for models with tridiagonal and dense Hessian matrices when MMHMC uses the 4th order modified Hamiltonian with derivatives calculated analytically (22) (left

panel), and the 4th order modified Hamiltonian with numerical approximation of the time derivatives (25) (right panel). Figure 7 (left) illustrates that models with dense Hessian matrices imply non-negligible overhead. In all other cases, the overheads are minor unless the number of integration steps becomes very small.

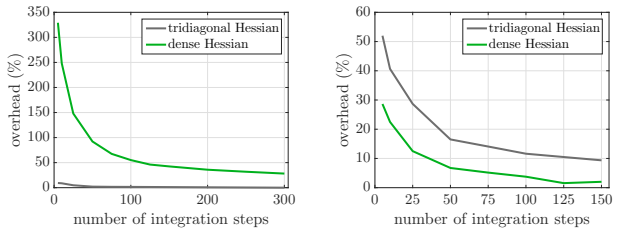

Fig. 7: Computational overhead of MMHMC compared to HMC for models with a tridiagonal and a dense Hessian matrix using the 4th order modified Hamiltonian (22) with all required derivatives calculated analytically (left), and the 4th order modified Hamiltonian (25) with numerical approximation of the time derivatives (right).

Dependence of MMHMC performance on a choice of tunable parameters is yet another factor which may deteriorate MMHMC efficiency. This is a well-known drawback common to all HMC-based methods. The advantage of vanilla HMC over other HMC methods discussed in this section comes from a fewer number of parameters to tune, due to an absence of partial momentum update in its algorithm.

In the final analysis, in Table 3 we summarize the differences between four somewhat similar methods, MMHMC, HMC, GHMC, GSHMC, in terms of how the presence or absence of various MMHMC features affects their capacity to sample efficiently.

We excluded randomization of methods' parameters from Table 3 since its impact on performance is inconsistent. While randomization of parameters normally improves performance in HMC, a randomized step size in MMHMC leads to an opposite effect, as was discussed above. Moreover, the GSHMC method has been formulated for physical applications where parameters have physical meaning and are assumed to be fixed by default.

3 Numerical Experiments

In this section we examine the performance of MMHMC on various benchmark models and compare it against other popular sampling techniques in computational statistics to answer the question of whether MMHMC emerges as a competitor to the most successful methods like HMC and RMHMC.

Table 3: Presence of performance impacting factors in HMC-based algorithms. (Bold symbols imply higher impacts)

Performance Enhancement				
	MMHMC	HMC	GHMC	GSHMC
Irreversibility	yes	no	yes	yes
Modified Hamiltonians	yes	no	no	yes
PMMC	yes	no	yes	yes
Splitting Integrators	yes	no	no	no
Performance Degradation				
Computation of High Order Derivatives	yes	no	no	**yes**
Variability of Weights	yes	no	no	yes
Ambiguous Choice of Parameters	**yes**	yes	**yes**	**yes**

3.1 Implementation

The MMHMC method has been implemented in the user-friendly in-house software package HaiCS (Hamiltonians in Computational Statistics), written in C and targeted to computers running UNIX certified operating systems.

The code is intended for statistical sampling of high dimensional and complex distributions and parameter estimation in different models through Bayesian inference using Hamiltonian Monte Carlo based methods. The currently available sampling techniques include the Metropolis algorithm, Hamiltonian Monte Carlo (HMC), Generalized Hybrid Monte Carlo (GHMC), Metropolis Adjusted Langevin Algorithm (MALA), second order Langevin Monte Carlo (L2MC), Generalized Shadow Hybrid Monte Carlo (GSHMC) and Mix & Match Hamiltonian Monte Carlo (MMHMC), the method presented in this paper. The package benefits from efficient implementation of modified Hamiltonians, the accurate multi-stage splitting integration schemes, the analysis tools compatible with CODA toolkit for MCMC diagnostics as well as an interface for implementing alternative splitting integrators and complex statistical models. The popular statistical models, such as, multivariate Gaussian distribution, Bayesian Logistic Regression and Stochastic Volatility are implemented in HaiCS.

The complete description of HaiCS package can be found in (Radivojević, 2016).

3.2 Experimental Results

We evaluate the performance of the MMHMC method and compare it with the Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Generalized Hybrid Monte Carlo (GHMC), Metropolis Adjusted Langevin Algorithm (MALA), Riemann Man-

ifold HMC (RMHMC) and Generalized Shadow Hybrid Monte Carlo (GSHMC) methods on a set of standard benchmark models used in the literature. Space exploration and/or sampling efficiency are examined on the banana-shaped distribution, multivariate Gaussian distribution, Bayesian logistic regression model, and the stochastic volatility model.

The choice of the optimal parameters of the algorithms remains an open question (Neal 2011) and not the subject of this paper. To make the comparison with other methods fair, we chose the following strategy. Since the stochastic volatility benchmark is studied well in literature, and HMC and RMHMC were tuned previously for a particular dimension of this benchmark, we took the found sets of optimal parameters as an initial guess and tuned them further. For Bayesian logistic regression and Gaussian model, especially for some data sets, such information is not available. In this case, we have located a range of reasonable parameters L, h and φ and performed the comparison for these sets.

For each MC iteration we draw the number of integration steps uniformly from $\{1, \dots, L\}$ for HMC and GHMC, and step size uniformly from $(0.8h, 1.2h)$ for HMC, GHMC and MALA methods. For GSHMC, we hold all parameters fixed as originally proposed in the method. Naturally, for r-stage integrators, a step size is set to rh and a number of integration steps to L/r. We observed that bigger values of L yield higher efficiency for HMC and GHMC for all tested step sizes, whereas for GSHMC and MMHMC this is not the case. Additionally, we tested MMHMC for a range of noise parameters φ being fixed as well as drawn uniformly from $(0, \varphi)$. Smaller values of φ tend to perform better for smaller values of the product hL and vice versa. Nevertheless, here we report only results obtained with the best φ and L among tested for each step size h. Complete experimental setup for each method and model tested is given in Appendix E. All our experiments are carried out with the identity mass matrix for HMC, GHMC, MALA, GSHMC and MMHMC.

In the results presented here, we compute ESS (MCSE) of the mean estimator for each variate, as proposed in 2.4.2, and report minimum, median, and maximum ESS (MCSE) across variates or just minimum ESS (maximum MCSE), as the most restrictive measures, calculated using the collected posterior samples. Computational time used for normalization of ESS, MCSE and efficiency comparison is measured as CPU time that each method takes to collect posterior samples. Except for the case of the banana-shaped distribution, for which we investigate a typical trajectory of a single Markov chain, all results are averaged over ten independent runs.

We examine the banana-shaped model with the Matlab code provided along with the paper by Lan et al. (2015), in which we implemented the MMHMC method. The rest of experiments are carried out with the in-house software package HaiCS, outlined in Section 3.1.

Each test model has been prepared to sampling with MMHMC, which in the first instance involved computation of derivatives of a model potential function.

3.2.1 Banana-shaped Distribution

We begin with a comparison of a space exploration achieved by MMHMC, RWMH, HMC and RMHMC in sampling of a 2-dimensional, non-linear target. The idea is to illustrate a representative mechanism of exploring a space for each tested method by generating a typical trajectory of a single Markov chain. Given data $\mathbf{y} = \{y_k\}_{k=1}^K$ we sample from the banana-shaped posterior distribution of the parameter $\boldsymbol{\theta} = (\theta_1, \theta_2)$ (Bornn and Cornebise, 2011) for which the likelihood and prior distributions are given as

$$y_k|\boldsymbol{\theta} \sim \mathcal{N}(\theta_1 + \theta_2^2, \sigma_y^2),\ k = 1, \dots, K,$$
$$\theta_1, \theta_2 \sim \mathcal{N}(0, \sigma_\theta^2),$$

respectively. Due to independency in the data and parameters, the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$ is proportional to

$$\prod_{k=1}^{K} p(y_k|\boldsymbol{\theta}) p(\theta_1) p(\theta_2).$$

Experimental setting. Data $\{y_k\}_{k=1}^K$, $K = 100$ are generated with $\theta_1 + \theta_2^2 = 1$, $\sigma_y = 2$ and $\sigma_\theta = 1$. Sampling with the MMHMC method is performed using the Verlet integrator and the modified Hamiltonian (22), a fixed number of integration steps, a step size and a noise parameter with values $L = 7, h = 1/9, \varphi = 0.5$, respectively. MMHMC is compared with RWMH, HMC and RMHMC for which simulation parameters are chosen as suggested by Lan et al. (2015).

Results. The dynamics of the four samplers is illustrated in Figure 8, in which sampling paths (lines) of the first 15 accepted proposals (dots) are shown. RWMH just has started to explore the parameter space and is still located in the low-density tail. In contrast, other methods already have visited high-density regions. As expected, RMHMC efficiently tracks a local curvature of the parameter space and is able to move along the ridge to its full extent. On the other hand, HMC and MMHMC tend to move across rather than along the ridge, with MMHMC sampling visibly broader than does HMC.

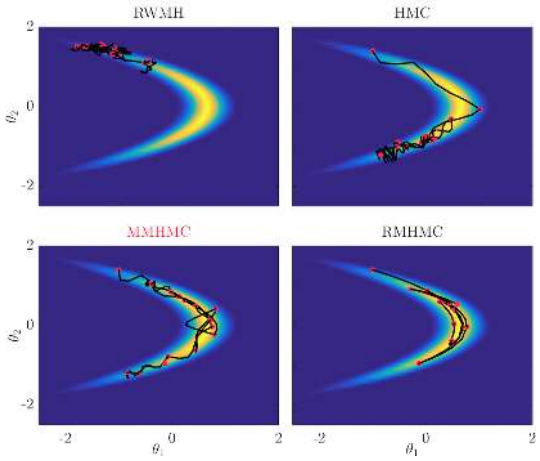

Fig. 8: The first 15 Monte Carlo iterations with sampling paths (lines) and accepted proposals (dots) in sampling from the banana-shaped distribution with Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Mix & Match HMC (MMHMC) and Riemann Manifold HMC (RMHMC).

3.2.2 Multivariate Gaussian Distribution

This benchmark has been proposed by Hoffman and Gelman (2014). The task is to sample from the D-dimensional Gaussian $\mathcal{N}(0, \Sigma)$, where the precision matrix Σ^{-1} is generated from a Wishart distribution with D degrees of freedom and the D-dimensional identity scale matrix.

Experimental setting. The tests are performed for three different dimensions, $D = 100, 1000, 2000$, using the HMC, GHMC, GSHMC and MMHMC methods. For the identity mass matrix, all four methods are invariant under rotations. Therefore, due to limited computational resources, for cases $D = 1000, 2000$ we choose the covariance matrix Σ to be diagonal with

$$\Sigma_{ii} = \sigma_i^2,$$

where σ_i^2 is the ith smallest eigenvalue of the original covariance matrix. Sampling with MMHMC is performed using the modified Hamiltonian (22), and the M-BCSS3 and M-ME3 integrators for $D = 100$ and $D = 1000, 2000$, respectively. For simplicity, we use the same formulation and implementation of the modified Hamiltonian in GSHMC as in MMHMC. However, we notice that in the original GSHMC algorithm the less efficient implementation of the modified Hamiltonian is proposed and thus the GSHMC performance in the following tests is likely overestimated. $10000, 20000, 30000$ samples are collected with each method with first 2000, 5000, 5000 being discarded as a warm-up for dimensions $D = 100, 1000, 2000$, respectively.

Results. Figure 9 compares the obtained acceptance rates (top) and corresponding time-normalized minimum ESS (bottom). While acceptance rates for HMC and GHMC drop considerably with increasing step size, especially for higher dimensions, MMHMC, in particular, and GSHMC maintain very high acceptance. For $D = 100$ acceptance rates for MMHMC and GSHMC start to drop visibly but still stay reasonably high for longest step sizes. In addition, Figure 10 presents the comparison in terms of time-normalized total distance from the mean $\|\boldsymbol{\theta}\|$ (top), and maximal MCSE (bottom) obtained with the four methods, where lower values correspond to better performance. As can be seen from the inspection of time-normalized ESS, MCSE and $\|\boldsymbol{\theta}\|$, for all tests, MMHMC outperforms in sampling efficiency all considered methods.

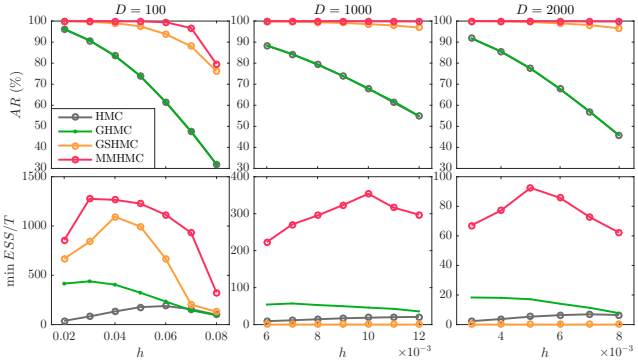

Fig. 9: D-dimensional Gaussian distribution. Acceptance rate (top) and time-normalized minimum ESS (bottom) for a range of step sizes h, obtained in sampling with Hamiltonian Monte Carlo (HMC), Generalized Hybrid Monte Carlo (GHMC), Generalized Shadow Hybrid Monte Carlo (GSHMC) and Mix & Match HMC (MMHMC).

The results on sampling efficiency are summarized in Figure 11, from which one can appreciate the amount of improvement achieved with MMHMC compared to HMC. For a range of step sizes h the efficiency factor (EF) in terms of time-normalized minimum ESS, maximum MCSE and total distance, relative with respect to HMC, is shown in such a way that values above 1 indicate superior performance of MMHMC. The improvement factor slowly increases with dimension. Depending on the choice of h, starting from at least a comparable performance (for the lowest dimension), the maximal improvement goes up to 29 times (for the highest dimension).

Finally, Figure 12 summarizes the improvements obtained with MMHMC compared to HMC in terms of the same metrics, when considering the results achieved with the best set of parameters for each method and each dimension found among the tested ones. Clearly,

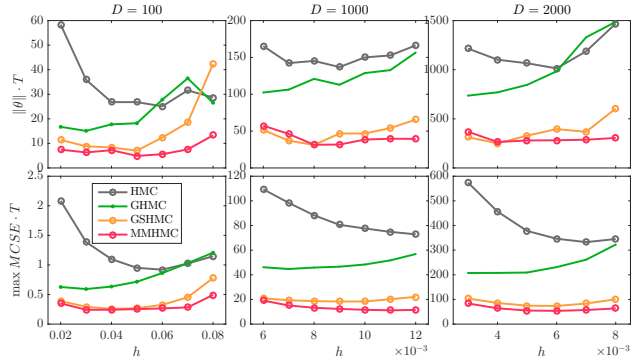

Fig. 10: D-dimensional Gaussian distribution. Time-normalized total distance from the mean (top) and maximal MCSE (bottom) for a range of step sizes h, obtained in sampling with Hamiltonian Monte Carlo (HMC), Generalized Hybrid Monte Carlo (GHMC), Generalized Shadow Hybrid Monte Carlo (GSHMC) and Mix & Match HMC (MMHMC).

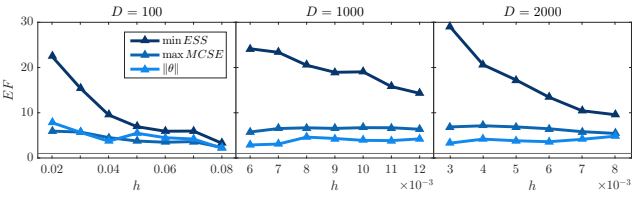

Fig. 11: D-dimensional Gaussian distribution. Relative efficiency (EF) of MMHMC w.r.t. HMC in terms of time-normalized minimum ESS, maximum MCSE and total distance from the mean, for a range of step sizes h.

the MMHMC method demonstrates superiority for all the three metrics considered, especially in terms of ESS. However, in a general case, the optimal parameters are not known a priori for either of the sampling methodologies.

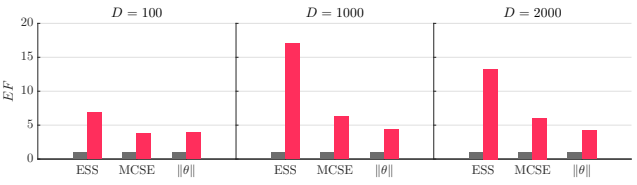

Fig. 12: D-dimensional Gaussian distribution. Relative efficiency (EF) of MMHMC w.r.t. HMC in terms of time-normalized minimum ESS, maximum MCSE and total distance from the mean, achieved using the best set of parameters for each method.

3.2.3 Bayesian Logistic Regression Model

The Bayesian logistic regression (BLR) model is used for solving binary classification problems appearing across various fields such as medical and social sciences, engineering, insurance, ecology, sports, etc.

Let consider K instances of data $\{\mathbf{x}_k, y_k\}_{k=1}^K$, where $\mathbf{x}_k$ are vectors of $D-1$ covariates and $y_k \in \{0,1\}$ are binary responses. In the BLR model, response variable $\mathbf{y} = (y_1, \ldots, y_K)$ is governed by a Bernoulli distribution with a parameter $\mathbf{p} = (p_1, \ldots, p_K)$. The unobserved probability p_k of a particular outcome is linked to the linear predictor function through the logit function, i.e.

$$\text{logit}(p_k) = \theta_0 + \theta_1 x_{1,k} + \cdots + \theta_{D-1} x_{D-1,k},$$

where $\boldsymbol{\theta} \in \mathbb{R}^D$ is the regression coefficient vector. The prior of the regression coefficient can be chosen e.g. as $\boldsymbol{\theta} \sim \mathcal{N}(0, \alpha\mathbb{I})$, with a known α.

If we construct the design matrix $X \in \mathbb{R}^{K,D}$ of input data as

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1,D-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{K1} & \cdots & x_{K,D-1} \end{bmatrix},$$

the likelihood function is given as

$$\begin{aligned} p(\mathbf{y}|X,\boldsymbol{\theta}) &= \prod_{k=1}^{K} p(y_k|X_k,\boldsymbol{\theta}) \\ &= \prod_{k=1}^{K} \left(\frac{\mathrm{e}^{X_k\boldsymbol{\theta}}}{1+\mathrm{e}^{X_k\boldsymbol{\theta}}}\right)^{y_k} \left(\frac{1}{1+\mathrm{e}^{X_k\boldsymbol{\theta}}}\right)^{1-y_k}, \end{aligned}$$

where X_k is the kth row of the matrix X. The corresponding posterior distribution over the regression coefficients is

$$\pi(\boldsymbol{\theta}|\mathbf{y},\mathbf{x}) \propto \prod_{k=1}^{K} p(y_k|X_k,\boldsymbol{\theta}) p(\boldsymbol{\theta})$$

with the prior

$$p(\boldsymbol{\theta}) \propto \exp\left\{-\frac{\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\alpha}\right\}.$$

Experimental setting. We use four different real data sets available from the University of California Irvine Machine Learning Repository Lichman (2013). The data set characteristics, such as names, numbers of regression parameters (D) and observations (K) are summarized in Table 4.

By following a common procedure, we normalize input data such that each covariate has zero mean and standard deviation of one. For each data set, a diffuse Gaussian prior is imposed by setting $\alpha = 100$.

For the German and Sonar data sets, $N = 5000$ posterior samples were generated after discarding the first 1000 samples as a warm-up, while for the bigger data sets (Musk and Secom) twice as much samples were

Table 4: Data sets used for the BLR model with corresponding numbers of regression parameters (D) and numbers of observations (K).

Data set	D	K
German	25	1000
Sonar	61	208
Musk	167	476
Secom	444	1567

collected. Apart from the comparison of MMHMC with HMC over the range of data sets, we also tested it against MALA on German data set. We do not investigate the performance of RMHMC since, as it was stated by Girolami and Calderhead (2011), RMHMC does not outperform HMC for dimensions as high as for the German data set ($D = 25$), which in our case is the data set of the smallest dimension.

In these experiments, MMHMC is used with the modified Hamiltonian (25) and the Verlet integrator.

Results. Acceptance rate (top), time-normalized minimum ESS (middle) and maximum MCSE (bottom) across variates obtained for BLR are presented in Figures 13 and 14. For all data sets, acceptance rate is the highest for MMHMC, as expected. For the smallest data set, while MALA exhibits visibly poor performance, both HMC and MMHMC demonstrate high and comparable efficiency. The trend changes for HMC method with increasing size of a problem. The superiority of MMHMC over HMC becomes more noticeable when a bigger data set is considered, resulting in the performance improvement by a factor of over 3 for the Secom data set ($D = 444$).

Figure 15 summarizes results on efficiency in terms of relative improvement of MMHMC compared to HMC, measured in terms of time-normalized minimum ESS and maximum MCSE across variates, obtained using the best set of simulation parameters among the tested ones for each method. Based on these results we can conclude that for the BLR model and tested data sets, MMHMC demonstrates improvement over HMC of up to 2.5 times.

3.2.4 Stochastic Volatility Model

Stochastic volatility (SV) models are a useful tool for modeling time-varying volatility with significant potential for applications (e.g. risk management/risk prediction, pricing of financial derivatives).

We consider the standard SV model defined with the latent, log-volatilities following autoregressive AR(1) process. The model, as described by Kim et al. (1998),

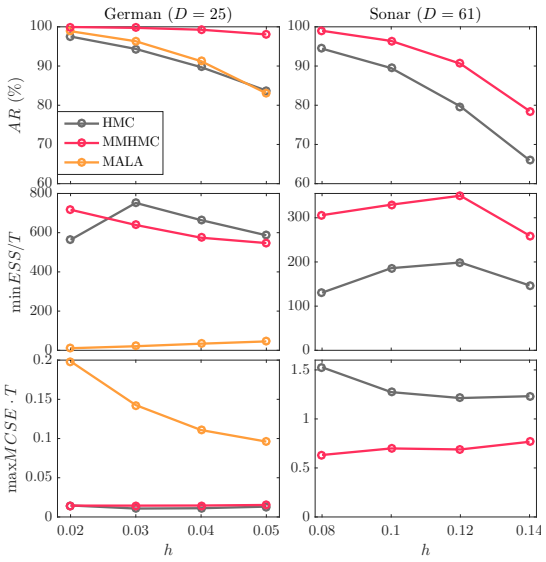

Fig. 13: Bayesian logistic regression. Acceptance rate (top), time-normalized minimum ESS (middle) and maximum MCSE (bottom) across variates obtained using Hamiltonian Monte Carlo (HMC), Mix & Match HMC (MMHMC) and Metropolis Adjusted Langevin Algorithm (MALA), for a range of step sizes h, for the German and Sonar data sets.

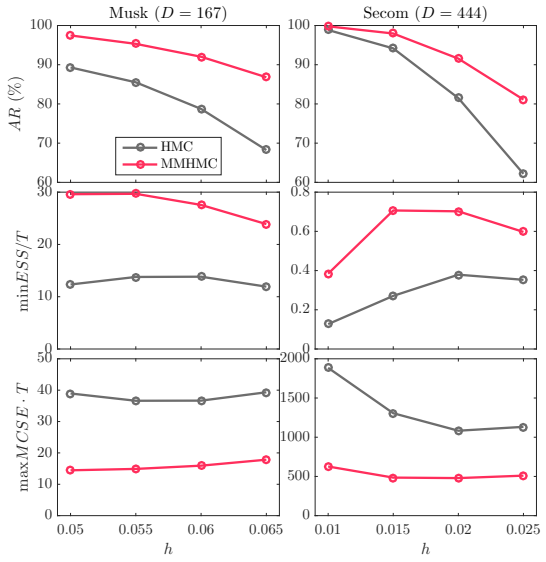

Fig. 14: Bayesian logistic regression. Acceptance rate (top), time-normalized minimum ESS (middle) and maximum MCSE (bottom) across variates obtained using HMC and MMHMC, for a range of step sizes h, for the Musk and Secom data sets.

takes the following form

$$y_t = \beta \exp(x_t/2)\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1)$$
$$x_t = \phi x_{t-1} + \sigma\eta_t, \quad \eta_t \sim \mathcal{N}(0,1)$$
$$x_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right),$$

where y_t are observed data of mean corrected log-returns, equidistantly spaced in time for $t = 1, \ldots, T$, and x_t

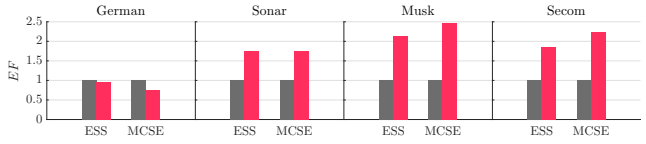

Fig. 15: Bayesian logistic regression. Relative efficiency (EF) of MMHMC w.r.t. HMC in terms of time-normalized minimum ESS and maximum MCSE across variates achieved using the best set of simulation parameters for each method.

are latent variables of log-volatility assumed to follow a stationary process. This assumption leads to the constraint $|\phi| < 1$. The error terms ϵ_t and η_t are serially and mutually uncorrelated white noise sequences with the standard normal distribution. The parameter β of the model can be interpreted as the modal instantaneous volatility, ϕ as the persistence in the volatility and σ as the volatility of the log-volatility, leading to the second constraint $\sigma > 0$.

Let denote the vector of model parameters as $\boldsymbol{\theta} = (\beta, \sigma, \phi)$. Its priors are chosen as $p(\beta) \propto 1/\beta, \sigma^2 \sim$ Scale-inv-$\chi^2(10, 0.05), (\phi+1)/2 \sim \text{Beta}(20, 1.5)$, leading to

$$p(\beta) \propto \frac{1}{\beta}$$
$$p(\sigma) \propto \sigma^{-11} \exp\{-1/4\sigma^2\}$$
$$p(\phi) \propto (\phi+1)^{19} (1-\phi)^{\frac{1}{2}} .$$

Instead of sampling jointly model parameters and latent volatilities from $\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, we follow a common procedure of cycling through the two full conditional distributions $\pi(\boldsymbol{\theta}|\mathbf{y}, \mathbf{x})$ and $\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ (see e.g. Jacquier et al. 1994; Chen et al. 2000; Liu 2008).

Since HMC methods sample real valued parameters, we handle the constraints $\sigma^2 > 0$ and $-1 \leq \phi \leq 1$ by making use of the transformation $\mathcal{T} : \boldsymbol{\theta} \to \bar{\boldsymbol{\theta}}$ to the real line, defined as

$$\bar{\boldsymbol{\theta}} = \mathcal{T}(\boldsymbol{\theta}) = (\beta, \ln(\sigma), \operatorname{artanh}(\phi)) = (\beta, \gamma, \alpha)$$

with the Jacobian

$$\mathcal{J}_{\mathcal{T}} = \begin{bmatrix} \frac{\mathrm{d}\beta}{\mathrm{d}\beta} & 0 & 0 \\ 0 & \frac{\mathrm{d}\gamma}{\mathrm{d}\sigma} & 0 \\ 0 & 0 & \frac{\mathrm{d}\alpha}{\mathrm{d}\phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma^{-1} & 0 \\ 0 & 0 & (1-\phi^2)^{-1} \end{bmatrix},$$

which accounts for the change of variables within the Hamiltonian dynamics and Metropolis test.

Experimental setting. We examine sampling of the standard SV model on simulated data with values $\beta = 0.65, \sigma = 0.15, \phi = 0.98$, for $T = 2000, 5000, 10000$ time points. This results in three experiments of dimensions $D = 2003, 5003, 10003$, which include three model parameters and T latent volatility variables to sample. We run 10000 iterations as a warm-up and generate 100000 posterior samples collecting every 5th sample. We compare MMHMC with HMC, and for $D = 2003$ we additionally run the RMHMC and GSHMC methods. The simulation parameters of the four methods are summarized in Appendix E. The results presented in this section for MMHMC are obtained with the M-ME3 and M-ME2 integrators for $D = 2003$ and $D = 5003, 10003$, respectively, and the modified Hamiltonian (22). As proposed in the original paper, we run GSHMC with modified Hamiltonians calculated using numerical derivatives. However, we notice that the original implementation of derivatives in GSHMC is less efficient than the one in HaiCS and thus the GSHMC performance in the following comparison is likely overestimated.

Results. Figures 16 and 17 provide efficiency in terms of time-normalized ESS and MCSE relative to HMC for experiments with $D = 2003$ and $D = 5003, 10003$, respectively. Acceptance rates (shown in inset figures) are rather high for all methods. However, there is no clear connection between obtained acceptance rates and ESS/MCSE. Results for $D = 2003$ demonstrate that RMHMC, GSHMC and MMHMC, outperform HMC in terms of time-normalized ESS for β and latent variables. However, all tested methods sample σ and ϕ comparably. For all sampled parameters, MMHMC shows comparable or superior performance to RMHMC.

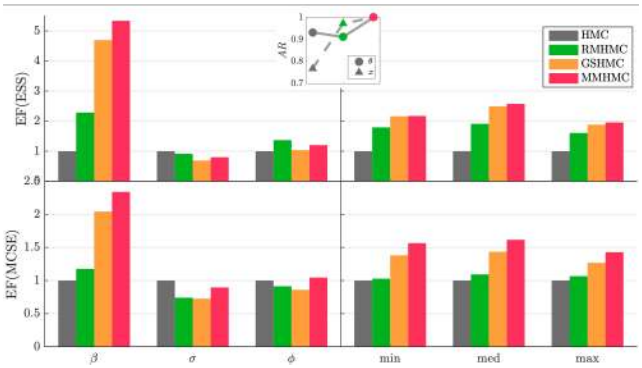

Fig. 16: Stochastic volatility. Sampling efficiency of RMHMC and MMHMC relative to HMC in terms of time-normalized ESS (top) and MCSE (bottom) for SV model parameters (left) and latent variables (right) and corresponding acceptance rates (inset) for dimension $D = 2003$.

We recall here that in contrast to the RMHMC method, HMC and MMHMC use the identity mass matrix. One way to improve the performance of these methods compared to RMHMC would be to define the mass matrix from an estimate of global covariances in

the warm-up phase and use it for obtaining the posterior samples.

We do not have an access to the optimal parameters for RMHMC for the dimensions higher than $D = 2003$. For $D = 5003, 10003$ we compare only MMHMC and HMC and observe that the superiority of MMHMC for sampling of model parameters and latent variables is maintained for higher dimensions.

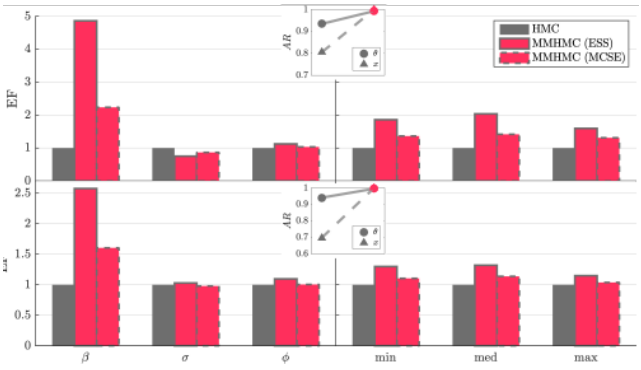

Fig. 17: Stochastic volatility. Sampling efficiency of MMHMC relative to HMC in terms of time-normalized ESS and MCSE for SV model parameters (left) and latent variables (right) and corresponding acceptance rates (inset) for dimensions $D = 5003$ (top) and $D = 10003$ (bottom).

4 Conclusions

We developed the irreversible MCMC method for enhanced statistical sampling, which offers higher sampling efficiency than the state-of-the-art MCMC method, Hamiltonian Monte Carlo. Our new approach, called Mix & Match HMC (MMHMC) arose as an extension of Generalized Shadow Hybrid Monte Carlo (GSHMC), earlier proposed for molecular simulation, published, patented and successfully tested on complex physical systems (Akhmatskaya and Reich 2008; Wee et al. 2008; Akhmatskaya et al. 2009, 2011; Escribano et al. 2017; Bonilla et al. 2018; García Daza et al. 2019). The MMHMC introduces a number of modifications in GSHMC needed for efficient sampling in computational statistics. It can be viewed as a generalized HMC importance sampler—momentum is updated in a general form and sampling is performed with respect to an importance distribution that is defined through modified Hamiltonian. To the best of our knowledge, this is the first time that the method sampling with modified Hamiltonians has been implemented and applied to Bayesian inference problems in computational statistics.

Being a method that generates both correlated and weighted samples, MMHMC requires a metric for sampling efficiency different from the one commonly used for MCMC. Here we suggested such a metric suitable for MCMC importance sampling based methods.

The method has been carefully tested and compared with the traditional and advanced sampling techniques such as Random Walk Metropolis-Hastings, Metropolis Adjusted Langevin Algorithm, Hamiltonian Monte Carlo, Riemann Manifold Hamiltonian Monte Carlo, Generalized Hybrid Monte Carlo and Generalized Shadow Hybrid Monte Carlo.

When compared to HMC, RWMH, MALA, GHMC and GSHMC, the MMHMC method demonstrates superior performance, in terms of higher acceptance rate, bigger time-normalized ESS and smaller MCSE, for a range of applications, range of dimensions and choice of parameters of the methods. The improvements are bigger for high-dimensional problems—for the multivariate Gaussian problem MMHMC demonstrated an improvement over HMC of up to 29 times. When comparing only for the best set of parameters among the tested ones for each method, MMHMC shows around 17 times better performance than HMC for the Gaussian problem and around 2.5 times improvement for the BLR model.

MMHMC and RMHMC demonstrate comparable, with a slight advantage of MMHMC, performance for the tested SV model. However, in contrast to the original RMHMC, MMHMC does not rely on higher order derivatives or inverse of the metric, and thus requires less implementation and computational effort. This issue becomes particularly important for high-dimensional problems with dense Hessian matrix. In addition, choices of integrators for RMHMC are limited due to the use of non-separable Hamiltonians, whereas MMHMC is well compatible with advanced splitting integration schemes.

Acknowledgements The authors would like to thank the financial support from MTM2016-76329-R (AEI/FEDER, EU) funded by the Spanish Ministry of Economy and Competitiveness (MINECO) and from Basque Government - ELKARTEK Programme, grant KK-2018/00054. This work has been possible thanks to the support of the computing infrastructure of the i2BASQUE academic network, and the technical and human support provided by IZO-SGI SGIker of UPV/EHU and European funding (ERDF and ESF). This research is also supported by the Basque Government through the BERC 2018-2021 program and by MINECO: BCAM Severo Ochoa accreditation SEV-2017-0718.

This work was also part of the Agile BioFoundry (http://agilebiofoundry.org) supported by the U.S. Department of Energy, Energy Efficiency and Renewable Energy, Bioenergy Technologies Office, through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

A Invariance of the PMMC step

The Partial Momentum Monte Carlo step of the MMHMC method leaves the importance target distribution $\tilde{\pi}$ (Eq. 12) invariant if for a transition kernel $T(\cdot|\cdot)$ the following condition is satisfied

$$\tilde{\pi}(\boldsymbol{\theta}', \mathbf{p}') = \int \tilde{\pi}(\boldsymbol{\theta}, \mathbf{p}) T\left((\boldsymbol{\theta}', \mathbf{p}')|(\boldsymbol{\theta}, \mathbf{p})\right) \mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p}$$

for all $n = 1, \ldots, N$.

The PMMC step is sampling on a space augmented with a noise vector $\mathbf{u} \sim \mathcal{N}(0, M)$ with the extended density $\hat{\pi}$ (defined in Eq. 17), for which

$$\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p}) = \int \hat{\pi}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) \mathrm{d}\mathbf{u}.$$

Therefore, we want to show that

$$\hat{\pi}(\boldsymbol{\theta}', \mathbf{p}', \mathbf{u}') = \int \hat{\pi}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) T\left((\boldsymbol{\theta}', \mathbf{p}', \mathbf{u}')|(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\right) \mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p}\mathrm{d}\mathbf{u}, \tag{28}$$

for the transition kernel defined as

$$\begin{aligned} T\left((\boldsymbol{\theta}', \mathbf{p}', \mathbf{u}')|(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\right) = \mathcal{P} \cdot \delta\left((\boldsymbol{\theta}', \mathbf{p}', \mathbf{u}') - \mathcal{R}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\right) \\ + (1 - \mathcal{P}) \cdot \delta\left((\boldsymbol{\theta}', \mathbf{p}', \mathbf{u}') - \hat{\mathcal{F}}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\right), \end{aligned}$$

where $\mathcal{P} = \min\left\{1, \hat{\pi}(\mathcal{R}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}))/\hat{\pi}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\right\}$ is the Metropolis probability, δ is the Delta function, $\mathcal{R}$ is the proposal function (Eq. 16) and $\hat{\mathcal{F}}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) = (\boldsymbol{\theta}, \mathbf{p}, -\mathbf{u})$ is the flipping function. Note that the map $\mathcal{R}$ is volume preserving, hence, the Metropolis probability $\mathcal{P}$ does not inlcude the Jacobian factor. For the sake of clarity, we denote $\mathbf{x} = (\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})$ and write the right-hand side of the expression (28) as

$$\begin{aligned} \int T(\mathbf{x}'|\mathbf{x})\hat{\pi}(\mathbf{x})\mathrm{d}\mathbf{x} = & \underbrace{\int \min\left\{\hat{\pi}(\mathbf{x}), \hat{\pi}(\mathcal{R}(\mathbf{x}))\right\} \cdot \delta\left(\mathbf{x}' - \mathcal{R}(\mathbf{x})\right) \mathrm{d}\mathbf{x}}_{\text{1st term}} \\ & \underbrace{+ \int \hat{\pi}(\mathbf{x}) \cdot \delta\left(\mathbf{x}' - \hat{\mathcal{F}}(\mathbf{x})\right) \mathrm{d}\mathbf{x}}_{\text{2nd term}} \\ & \underbrace{- \int \min\left\{\hat{\pi}(\mathbf{x}), \hat{\pi}(\mathcal{R}(\mathbf{x}))\right\} \cdot \delta\left(\mathbf{x}' - \hat{\mathcal{F}}(\mathbf{x})\right) \mathrm{d}\mathbf{x}}_{\text{3rd term}}. \end{aligned}$$

Applying change of variables $\mathbf{x} = \hat{\mathcal{F}} \circ \mathcal{R}(\bar{\mathbf{x}})$, which is volume preserving, to the 1st term in the sum, omitting the bars, and using the fact that $\hat{\mathcal{F}} = \mathcal{R} \circ \hat{\mathcal{F}} \circ \mathcal{R}$, one obtains

$$\text{1st term} = \int \min\left\{\hat{\pi}(\hat{\mathcal{F}} \circ \mathcal{R}(\mathbf{x})), \hat{\pi}(\hat{\mathcal{F}}(\mathbf{x}))\right\} \cdot \delta\left(\mathbf{x}' - \hat{\mathcal{F}}(\mathbf{x})\right) \mathrm{d}\mathbf{x}.$$

Since $\hat{\pi} \circ \hat{\mathcal{F}} = \hat{\pi}$, the 1st and 3rd terms cancel out. Employing change of variables $\mathbf{x} = \hat{\mathcal{F}}(\bar{\mathbf{x}})$ to the 2nd term and again omitting the bars, leads to

$$\text{2nd term} = \int \hat{\pi}(\mathbf{x}) \cdot \delta\left(\mathbf{x}' - \mathbf{x}\right) \mathrm{d}\mathbf{x} = \hat{\pi}(\mathbf{x}'),$$

which proves the equality (28).

B Modified Hamiltonians for Splitting Integrators

The coefficients for the two-stage integrator family (14) and modified Hamiltonians (22)–(24) are the following

$$\begin{aligned} c_{21} =& \frac{1}{24}\Big(6b - 1\Big) \\ c_{22} =& \frac{1}{12}\Big(6b^2 - 6b + 1\Big) \\ c_{41} =& \frac{1}{5760}\Big(7 - 30b\Big) \\ c_{42} =& \frac{1}{240}\Big(-10b^2 + 15b - 3\Big) \\ c_{43} =& \frac{1}{120}\Big(-30b^3 + 35b^2 - 15b + 2\Big) \\ c_{44} =& \frac{1}{240}(20b^2 - 1). \end{aligned} \tag{29}$$

Using (29) one can also obtain the modified Hamiltonian for the Verlet integrator, since two steps of Verlet integration are equivalent to one step of the two-stage integrator with $b = 1/4$. The coefficients are therefore

$$\begin{aligned} c_{21} =& \frac{1}{12}, \quad c_{22} = -\frac{1}{24} \\ c_{41} =& -\frac{1}{720}, \quad c_{42} = \frac{1}{120}, \quad c_{43} = -\frac{1}{240}, \quad c_{44} = \frac{1}{60}. \end{aligned}$$

For three-stage integrators (15) (a two-parameter family) the coefficients are

$$\begin{aligned} c_{21} =& \frac{1}{12}\Big(1 - 6a(1-a)(1-2b)\Big) \\ c_{22} =& \frac{1}{24}\Big(6a(1-2b)^2 - 1\Big) \\ c_{41} =& \frac{1}{720}\Big(1 + 2(a-1)a(8 + 31(a-1)a)(1-2b) - 4b\Big) \\ c_{42} =& \frac{1}{240}\Big(6a^3(1-2b)^2 - a^2(19 - 116b + 36b^2 + 240b^3) \\ & + a(27 - 208b + 308b^2) - 48b^2 + 48b - 7\Big) \\ c_{43} =& \frac{1}{180}\Big(1 + 15a(1-2b)(-1 + 2a(2 - 3b + a(4b-2)))\Big) \\ c_{44} =& \frac{1}{240}\Big(-1 + 20a(1-2b)(b + a(1 + 6(b-1)b))\Big). \end{aligned}$$

The coefficients for the modified Hamiltonians (25)–(26) are calculated as

$$k_{21} = c_{21}, \quad k_{22} = c_{22},$$

$k_{41} = c_{41}, \quad k_{42} = 3c_{41} + c_{42},$
$k_{43} = c_{41} + c_{44}, \quad k_{44} = 3c_{41} + c_{42} + c_{43}.$

For the 4th order modified Hamiltonian (25) we use the second order centered finite difference approximations of time derivatives of the gradient of the potential function

$$\mathbf{U}^{(1)} = \frac{\mathbf{U}(t_{n+1}) - \mathbf{U}(t_{n-1})}{2\varepsilon}, \tag{30}$$

with $\varepsilon = h$ for the Verlet, $\varepsilon = h/2$ for two-stage and $\varepsilon = ah$ for three-stage integrators with a being the integrator's coefficient advancing position variables. The 6th order modified Hamiltonian (26), here considered only for the Verlet and two-stage integrators, is calculated using fourth order approximation for the first derivative and second order approximations for the second and third derivatives

$$\mathbf{U}^{(1)} = \frac{\mathbf{U}(t_{n-2}) - 8\mathbf{U}(t_{n-1}) + 8\mathbf{U}(t_{n+1}) - \mathbf{U}(t_{n+2})}{12\varepsilon}$$
$$\mathbf{U}^{(2)} = \frac{\mathbf{U}(t_{n-1}) - 2\mathbf{U}(t_n) + \mathbf{U}(t_{n+1})}{\varepsilon^2}$$
$$\mathbf{U}^{(3)} = \frac{-\mathbf{U}(t_{n-2}) + 2\mathbf{U}(t_{n-1}) - 2\mathbf{U}(t_{n+1}) + \mathbf{U}(t_{n+2})}{2\varepsilon^3},$$

where ε depends on the integrator as before. The interpolating polynomial in terms of the gradient of the potential function $\mathbf{U}(t_i) = U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^i),\ i = n-k, \ldots, n, \ldots, n+k, n \in \{0, L\}$ is constructed from a numerical trajectory $\{U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^i\}_{i=-k}^{L+k}$ where $k = 1$ and $k = 2$ for the 4th and 6th order modified Hamiltonians, respectively.

C Modified PMMC Step

In the modified PMMC step proposed for MMHMC, a partial momentum update is integrated into the modified Metropolis test, i.e. it is implicitly present in the algorithm. This reduces the frequency of derivative calculations in the Metropolis function. To implement this idea, one should recall that the momentum update probability

$$\mathcal{P} = \min\left\{1, \frac{\exp\big(-\hat{H}(\boldsymbol{\theta}, \mathbf{p}^*, \mathbf{u}^*)\big)}{\exp\big(-\hat{H}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u})\big)}\right\} \tag{31}$$

depends on the error in the extended Hamiltonian (18). Let us first consider the 4th order modified Hamiltonian (22) with analytical derivatives of the potential function. It is easy to show that the difference in the extended Hamiltonian (18) between a current state and a state with partially updated momentum is

$$\begin{aligned}\Delta\hat{H} &= U(\boldsymbol{\theta}) + \frac{1}{2}(\mathbf{p}^*)^T M^{-1}\mathbf{p}^* \\ &+ h^2 c_{21}(\mathbf{p}^*)^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}^* \\ &+ h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^* \\ &- U(\boldsymbol{\theta}) - \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} - h^2 c_{21}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ &- h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) - \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u} \\ &= h^2 c_{21}\Big(\varphi A + 2\sqrt{\varphi(1-\varphi)}B\Big)\end{aligned} \tag{32}$$

with

$$\begin{aligned}A &= (\mathbf{u} - \mathbf{p})^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})(\mathbf{u} + \mathbf{p}) \\ B &= \mathbf{u}^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p}.\end{aligned} \tag{33}$$

For the 6th order modified Hamiltonian (24) for Gaussian problems, the error in the extended Hamiltonian (18) can be calculated in a similar manner

$$\begin{aligned}\Delta\hat{H} &= h^2 c_{21}\Big(\varphi(A - B) + 2\sqrt{\varphi(1-\varphi)}C\Big) \\ &+ h^4 c_{44}\Big(\varphi(D - E) + 2\sqrt{\varphi(1-\varphi)}F\Big),\end{aligned} \tag{34}$$

with

$$\begin{aligned}A &= \mathbf{u}^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{u} \\ B &= \mathbf{p}^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p} \\ C &= \mathbf{u}^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p} \\ D &= (U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{u})^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{u} \\ E &= (U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p})^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p} \\ F &= (U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{u})^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{p}.\end{aligned}$$

Therefore, if the modified Hamiltonians (22)–(24) with analytical derivatives are used, a new momentum can be determined as

$$\bar{\mathbf{p}} = \begin{cases}\sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u} & \text{with probability} \\ & \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\} \\ \mathbf{p} & \text{otherwise,}\end{cases} \tag{35}$$

where $\mathbf{u} \sim \mathcal{N}(0, M)$ is the noise vector, $\varphi \in (0, 1]$ and $\Delta\hat{H}$ is defined as in (32) or (34).

Consequently, for models with no hierarchical structure, there is no need to calculate gradients within the PMMC step, second derivatives can be taken from the previous Metropolis test within the HDMC step, and there is no need to generate $\mathbf{u}^*$.

If the modified Hamiltonians are calculated using numerical time derivatives of the gradient of the potential function, for the Verlet, two- and three-stage integrators as in (25)–(26), the difference in the 4th order extended Hamiltonian becomes

$$\Delta\hat{H} = hk_{21}\Big((\mathbf{p}^*)^T P_1^* - \mathbf{p}^T P_1\Big), \tag{36}$$

whereas for the 6th order extended Hamiltonian it is

$$\begin{aligned}\Delta\hat{H} &= hk_{21}\Big((\mathbf{p}^*)^T P_1^* - \mathbf{p}^T P_1\Big) \\ &+ hk_{41}\Big((\mathbf{p}^*)^T P_3^* - \mathbf{p}^T P_3\Big) \\ &+ h^2 k_{42}\Big(U_{\mathbf{x}}^T P_2^* - U_{\mathbf{x}}^T P_2\Big) \\ &+ h^2 k_{43}\Big((P_1^*)^T P_1^* - P_1^T P_1\Big).\end{aligned}$$

Here P_1^*, P_2^*, P_3^*, are the first, second and third order scaled time derivatives of the gradient, respectively (see Section 2.3.3), calculated from the trajectory with updated momentum $\mathbf{p}^*$. The computational gain of the new PMMC step, in this case, results from skipping a calculation of the terms multiplying k_{22} in (25) and k_{44} in (26). It has to be admitted that the term multiplying k_{22} in (25) is of negligible cost, and thus the gain from using the new momentum update is not as significant as in the case of modified Hamiltonians with analytical derivatives. On the contrary, the saving in computation arising from the absence of the term multiplying k_{44} in the 6th order modified Hamiltonian (26), is essential.

In summary, in the case of the 6th order modified Hamiltonian, with derivatives calculated either analytically or numerically, the proposed momentum refreshment enhances computational performance of MMHMC. This also applies to the

cases when the 4th order modified Hamiltonian with analytical derivatives is used. In this situation, however, if the Hessian matrix of the potential function is dense, instead of using the modified Hamiltonian with analytical derivatives, we recommend using numerical derivatives, for which the saving is negligible. On the other hand, if the computation of the Hessian matrix is not very costly (e.g. being block-diagonal, sparse, close to constant), it might be more efficient to use analytical derivatives, for which the new formulation of the Metropolis test leads to computational saving.

D Algorithmic Summary

Algorithm 1 Hamiltonian Monte Carlo

Input: N: number of Monte Carlo samples
h: step size
L: number of integration steps
M: mass matrix
$\Psi_{h,L}$: numerical integrator
Initialize $\boldsymbol{\theta}^0$
for $n = 1, \dots, N$ **do**
  $\boldsymbol{\theta} = \boldsymbol{\theta}^{n-1}$
  Draw momentum from Gaussian distribution: $\mathbf{p} \sim \mathcal{N}(0, M)$
  Generate a proposal by integrating Hamiltonian dynamics: $(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})$
  Set $\boldsymbol{\theta}^n = \boldsymbol{\theta}'$ with probability $\alpha = \min\{1, \exp(H(\boldsymbol{\theta}, \mathbf{p}) - H(\boldsymbol{\theta}', \mathbf{p}'))\}$, otherwise set $\boldsymbol{\theta}^n = \boldsymbol{\theta}$
  Discard momentum $\mathbf{p}'$
end for

We provide two alternative algorithms for the MMHMC method. One (Algorithm 2) uses the modified Hamiltonians defined through analytical derivatives of the potential function and is recommended for the problems with sparse Hessian matrices. The other algorithm (Algorithm 3) relies on the modified Hamiltonians expressed through numerical time derivatives of the gradient of the potential function. This algorithm, although including additional integration step, is beneficial for cases where higher order derivatives are computationally demanding.

E Experimental setup

Algorithm 2 MMHMC using Hessian of the potential function

Input: N: number of Monte Carlo samples
h: step size
$p(L)$: number-of-integration-steps randomization policy
$p(\varphi)$: noise-parameter randomization policy
M: mass matrix
r: number of stages in the numerical integrator ($r = 1, 2, 3$)
$\Psi_{h,L}$: symplectic r-stage numerical integrator
Initialize $(\boldsymbol{\theta}^0, \mathbf{p}^0)$
Calculate Hessian $U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}^0)$
for $n = 1, \dots, N$ **do**
  Draw $L_n \sim p(L), \varphi_n \sim p(\varphi)$
  $(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}^{n-1}, \mathbf{p}^{n-1})$
  PMMC step
  Draw noise $\mathbf{u} \sim \mathcal{N}(0, M)$ and update momenta

$$\bar{\mathbf{p}} = \begin{cases} \sqrt{1-\varphi_n}\mathbf{p} + \sqrt{\varphi_n}\mathbf{u} & \text{with probability} \\ & \mathcal{P} = \min\left\{1, \exp\left(-\Delta\hat{H}\right)\right\} \\ \mathbf{p} & \text{otherwise} \end{cases}$$

  $\Delta\hat{H}$ defined in Eqs. (32)–(33)
  Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})$ defined in Eq. (22)
  HDMC step
  Generate a proposal by integrating Hamiltonian dynamics with step size h over L_n steps

$$(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h,L_n}(\boldsymbol{\theta}, \bar{\mathbf{p}})$$

  Calculate Hessian $U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}')$ and modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}')$
  Metropolis test

$$(\boldsymbol{\theta}^n, \mathbf{p}^n) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{accept with probability} \\ & \alpha = \min\left\{1, \exp\left(-\Delta\tilde{H}\right)\right\} \\ (\boldsymbol{\theta}, -\mathbf{p}) & \text{reject otherwise} \end{cases}$$

  $\Delta\tilde{H} = \tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})$
end for
Calculate weights $w_n, n = 1, \dots, N$ (Eq. 21) and estimate integral (1) as

$$\hat{I} = \frac{\sum_{n=1}^{N} f(\boldsymbol{\theta}^n) w_n}{\sum_{n=1}^{N} w_n}$$

References

Afshar, H.M., Domke, J.: Reflection, Refraction, and Hamiltonian Monte Carlo. In: Advances in Neural Information Processing Systems (NIPS) (2015)

Akhmatskaya, E., Fernández-Pendás, M., Radivojević, T., Sanz-Serna, J.M.: Adaptive Splitting Integrators for Enhancing Sampling Efficiency of Modified Hamiltonian Monte Carlo Methods in Molecular Simulation. Langmuir **33**(42), 11530–11542 (2017). doi:10.1021/acs.langmuir.7b01372

Akhmatskaya, E., Nobes, R., Reich, S.: Method, apparatus and computer program for molecular simulation. US patent (granted) (2011)

Akhmatskaya, E., Reich, S.: The Targeted Shadowing Hybrid Monte Carlo (TSHMC) Method. In: New Algorithms for Macromolecular Simulation, Lecture Notes in Computational Science and Engineering, vol. 49, pp. 141–153. Springer-Verlag, Berlin (2006)

Akhmatskaya, E., Reich, S.: GSHMC: An efficient method for molecular simulation. Journal of Computational Physics **227**(10), 4934–4954 (2008). doi: http://dx.doi.org/10.1002/andp.19053221004

Akhmatskaya, E., Reich, S.: New Hybrid Monte Carlo Methods for Efficient Sampling: from Physics to Biology and Statistics. Progress in Nuclear Science and Technology **2**, 447–462 (2012)

Akhmatskaya, E., Reich, S., Nobes, R.: Method, apparatus and computer program for molecular simulation. GB patent (published) (2009)

Table 5: Parameter values used for the multivariate Gaussian model experiments. The Verlet integrator was employed in HMC, GHMC and GSHMC methods, whereas for MMHMC the M-BCSS3 integrator was used for $D = 100$ and M-ME3 for $D = 1000, 2000$. For HMC and GHMC step size is drawn from $U(0.8h, 1.2h)$. For HMC, GHMC and MMHMC trajectory length is drawn from $U\{1, \ldots, L\}$. For GHMC, MMHMC noise parameter is drawn from $U(0, \varphi)$. For GSHMC all parameters are fixed.

D	Method		Parameter value						
100		h	0.02	0.03	0.04	0.05	0.06	0.07	0.08
	HMC	L	500	500	500	500	500	500	400
	GHMC	L	500	500	500	500	500	500	400
		φ	0.1	0.1	0.1	0.1	0.9	0.9	0.9
	GSHMC	L	150	100	100	100	100	100	100
		φ	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	MMHMC	h	0.06	0.09	0.12	0.15	0.18	0.21	0.24
		L	100	67	67	67	67	67	67
		φ	0.1	0.1	0.1	0.1	0.1	0.1	0.1
1000		h	0.006	0.007	0.008	0.009	0.01	0.011	0.012
	HMC	L	5000	5000	5000	5000	5000	5000	5000
	GHMC	L	5000	5000	5000	5000	5000	5000	5000
		φ	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	GSHMC	L	2000	1500	1000	1000	1000	1000	1000
		φ	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	MMHMC	h	0.018	0.021	0.024	0.027	0.03	0.033	0.036
		L	1333	1000	667	667	667	667	667
		φ	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2000		h	0.003	0.004	0.005	0.006	0.007	0.008	
	HMC	L	10000	10000	10000	10000	10000	10000	
	GHMC	L	10000	10000	10000	10000	10000	10000	
		φ	0.1	0.1	0.1	0.1	0.1	0.1	
	GSHMC	L	3000	2000	2000	2000	2000	2000	
		φ	0.1	0.1	0.1	0.1	0.1	0.1	
	MMHMC	h	0.009	0.012	0.015	0.018	0.021	0.024	
		L	2000	1333	1333	1333	1333	1333	
		φ	0.1	0.1	0.1	0.1	0.1	0.1	

Table 6: Parameter values used for the Bayesian logistic regression model experiments. The Verlet integrator was employed on all methods. For HMC and MALA step size is drawn from $U(0.8h, 1.2h)$. For HMC trajectory length is drawn from $U\{1, \ldots, L\}$.

D	Method	h	0.02	0.03	0.04	0.05
German	HMC	L	25	25	25	25
	MALA	L	1	1	1	1
	MMHMC	L	$U\{1, \ldots, 25\}$	$U\{1, \ldots, 25\}$	$U\{1, \ldots, 25\}$	$U\{1, \ldots, 25\}$
		φ	$U(0, 0.5)$	$U(0, 0.5)$	$U(0, 0.9)$	$U(0, 0.9)$
Sonar		h	0.08	0.1	0.12	0.14
	HMC	L	200	200	200	200
	MMHMC	L	50	50	50	50
		φ	0.25	0.5	0.5	0.5
Musk		h	0.05	0.055	0.06	0.065
	HMC	L	400	400	400	400
	MMHMC	L	100	100	100	100
		φ	0.25	0.25	0.25	0.25
Secom		h	0.01	0.015	0.02	0.025
	HMC	L	900	900	900	900
	MMHMC	L	150	150	150	150
		φ	0.25	0.25	0.25	0.25

Table 7: Parameter values used for the Stochastic Volatility model experiments. For HMC step size is drawn from $U(0.8h, 1.2h)$ and trajectory length from $U\{1, \ldots, L\}$. For MMHMC noise parameter is drawn from $U(0, \varphi)$. For all other cases, parameters are fixed.

D	Method	Integrator	$h_{\boldsymbol{\theta}}$	$h_{\mathbf{x}}$	$L_{\boldsymbol{\theta}}$	$L_{\mathbf{x}}$	$\varphi_{\boldsymbol{\theta}}$	$\varphi_{\mathbf{x}}$
2003	HMC	Verlet	0.01	0.03	6	76		
	RMHMC	Verlet	0.5	0.1	6	50		
	GSHMC	Verlet	0.008	0.023	3	38	0.25	0.4
	MMHMC	M-ME3	0.024	0.069	2	25	0.5	0.8
5003	HMC	Verlet	0.006	0.02	6	76		
	MMHMC	M-ME2	0.012	0.032	3	38	0.5	0.5
10003	HMC	Verlet	0.004	0.02	6	76		
	MMHMC	M-ME2	0.008	0.022	3	38	0.8	0.8

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J., Stuart, A.: Optimal tuning of the hybrid Monte Carlo algorithm. Bernoulli **19**, 1501–1534 (2013)

Betancourt, M.: Nested Sampling with Constrained Hamiltonian Monte Carlo. In: AIP Conference Proceedings, vol. 1305, pp. 165–172 (2011). doi:10.1063/1.3573613

Betancourt, M.: A general metric for Riemannian manifold Hamiltonian Monte Carlo. In: Geometric Science of Information, pp. 327–334. Springer (2013a)

Betancourt, M.: Generalizing the No-U-Turn Sampler to Riemannian Manifolds. arXiv:1304.1920v1 (2013b)

Betancourt, M.: Adiabatic Monte Carlo. arXiv:1405.3489v4 (2014)

Betancourt, M.: A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv:1701.02434v1 (2017)

Betancourt, M., Byrne, S., Livingstone, S., Girolami, M.: The Geometric Foundations of Hamiltonian Monte Carlo. Bernoulli **23**(4A), 2257–2298 (2017). doi:10.3150/16-bej810

Betancourt, M., Girolami, M.: Hamiltonian Monte Carlo for hierarchical models. Current Trends in Bayesian Methodology with Applications **79** (2015)

Blanes, S., Casas, F., Sanz-Serna, J.M.: Numerical integrators for the Hybrid Monte Carlo method. SIAM Journal on Scientific Computing **36**(4), A1556–A1580 (2014)

Bonilla, M.R., Lozano, A., Escribano, B., Carrasco, J., Akhmatskaya, E.: Revealing the mechanism of sodium diffusion in naxfepo4 using an improved force field. Journal of Physical Chemistry C **122**(15), 8065–8075 (2018). doi: 10.1021/acs.jpcc.8b00230

Bornn, L., Cornebise, J.: Discussion on the paper by Girolami and Calderhead. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73**(2), 123–214 (2011)

Bouchard-Côté, A., Vollmer, S.J., Doucet, A.: The Bouncy Particle Sampler: A Non-Reversible Rejection-Free Markov Chain Monte Carlo Method. Journal of the American Statistical Association **113**(522), 855–867 (2018). doi: 10.1080/01621459.2017.1294075

Brubaker, M.A., Salzmann, M., Urtasun, R.: A Family of MCMC Methods on Implicitly Defined Manifolds. In: International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 161–172 (2012)

Campos, C.M., Sanz-Serna, J.M.: Extra chance generalized hybrid Monte Carlo. Journal of Computational Physics **281**, 365–374 (2015)

Chen, L., Qin, Z., Liu, J.S.: Exploring Hybrid Monte Carlo in Bayesian Computation. ISBA 2000, Proceedings (2000)

Chen, T., Fox, E.B., Guestrin, C.: Stochastic Gradient Hamiltonian Monte Carlo. In: Proceedings of the 31st International Conference on Machine Learning, Beijing, China (2014)

Dinh, V., Bilge, A., Zhang, C., IV, F.A.M.: Probabilistic Path Hamiltonian Monte Carlo. In: Proceedings of the 34th International Conference on Machine Learning (2017)

Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Physics Letters B **195**(2), 216–222 (1987)

Duncan, A.B., Leliévre, T., Pavliotis, G.A.: Variance Reduction Using Nonreversible Langevin Samplers. Journal of Statistical Physics **163**(3), 457–491 (2016)

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

Escribano, B., Lozano, A., Radivojević, T., Fernández-Pendás, M., Carrasco, J., Akhmatskaya, E.: Enhancing sampling in atomistic simulations of solid state materials for batteries: a focus on olivine NaFePO4. Theor. Chem. Acc. **136**(43) (2017). doi:10.1007/s00214-017-2064-4

Fang, Y., Sanz-Serna, J.M., Skeel, R.D.: Compressible generalized hybrid Monte Carlo. The Journal of Chemical Physics **140**(17), 174108 (2014)

Fu, T., Luo, L., Zhang, Z.: Quasi-Newton Hamiltonian Monte Carlo. In: Proceedings of Uncertainty in Artificial Intelligence, pp. 212–221 (2016)

García Daza, F., Bonilla, M.R., Llordés, A., Carrasco, J., Akhmatskaya, E.: Atomistic insight into ion transport and conductivity in ga/al-substituted li7la3zr2o12 solid electrolytes. ACS Appl. Mater. Interfaces **11**, 753–765 (2019). doi:10.1021/acsami.8b17217

Geyer, C.J.: Practical Markov Chain Monte Carlo. Statistical Science, **7**(4), 473–483 (1992)

Girolami, M., Calderhead, B.: Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73**(2), 123–214 (2011)

Graham, M.M., Storkey, A.J.: Continuously tempered Hamiltonian Monte Carlo. In: Proceedings of Uncertainty in Artificial Intelligence (2017)

Gramacy, R., Samworth, R., King, R.: Importance tempering. Statistics and Computing **20**(1), 1–7 (2010)

Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration. Springer-Verlag, 2 ed. (2006)

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

Horowitz, A.M.: A generalized guided Monte Carlo algorithm. Physics Letters B **268**, 247–252 (1991)

Algorithm 3 MMHMC using numerical derivatives of the gradient of the potential

1: **Input:** N: number of Monte Carlo samples
h: step size
$p(L)$: number-of-integration-steps randomization policy
$p(\varphi)$: noise-parameter randomization policy
M: mass matrix
r: number of stages in the numerical integrator ($r = 1, 2, 3$)
$\Psi_{h,L}$: symplectic r-stage numerical integrator
2: Initialize $(\boldsymbol{\theta}^0, \mathbf{p}^0)$
3: Integrate one stage (i.e. one gradient calculation) backward, $\Psi_{h,-1}(\boldsymbol{\theta}^0, \mathbf{p}^0)$, and forward, $\Psi_{h,1}(\boldsymbol{\theta}^0, \mathbf{p}^0)$
4: Calculate scaled time derivative of the gradient P_1 using Eq. (30)
5: **for** $n = 1, \ldots, N$ **do**
6: Draw $L_n \sim p(L), \varphi_n \sim p(\varphi)$
7: $(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}^{n-1}, \mathbf{p}^{n-1})$
PMMC step
8: Draw noise $\mathbf{u} \sim \mathcal{N}(0, M)$ and propose momenta

$$\mathbf{p}^* = \sqrt{1-\varphi_n}\mathbf{p} + \sqrt{\varphi_n}\mathbf{u}$$

9: Integrate one stage backward, $\Psi_{h,-1}(\boldsymbol{\theta}, \mathbf{p}^*)$, and forward, $\Psi_{h,1}(\boldsymbol{\theta}, \mathbf{p}^*)$
10: Calculate the resulting scaled time derivative of the gradient P_1^*
11: Update momenta

$$\bar{\mathbf{p}} = \begin{cases} \mathbf{p}^* & \text{with probability } \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\}, \\ & \Delta\hat{H} \text{ defined in Eq. (36)} \\ \mathbf{p} & \text{otherwise} \end{cases}$$

12: Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})$ defined in Eq. (25)
HDMC step
13: Integrate Hamiltonian dynamics with step size h over L_n^+ steps and assign a proposal {$^+$ stands for an additional forward integration}

$$(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h,L_n}(\boldsymbol{\theta}, \bar{\mathbf{p}})$$

14: Calculate the resulting scaled time derivative of the gradient P_1'
15: Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}')$
16: Metropolis test {as in Algorithm 2, line 11}
17: **end for**
18: Compute weights and estimate integral (1) {as in Algorithm 2, line 14}

Izaguirre, J.A., Hampton, S.S.: Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules. Journal of Computational Physics **200**, 581–604 (2004)

Jacquier, E., Polson, N.G., Rossi, P.E.: Bayesian analysis of stochastic volatility models. Journal of Business & Economic Statistics **12**(4) (1994)

Kennedy, A.D.: The theory of hybrid stochastic algorithms. In: Probabilistic Methods in Quantum Field Theory and Quantum Gravity, pp. 209–223. Springer (1990)

Kennedy, A.D., Pendleton, B.: Cost of the Generalised Hybrid Monte Carlo Algorithm for Free Field Theory. Nuclear Physics B **607**, 456–510 (2001). doi:10.1016/S0550-3213(01)00129-8

Kim, S., Shephard, N., Chib, S.: Stochastic volatility: Likelihood inference and comparison with arch models. Review of Economic Studies **65**, 361–393 (1998)

Kleppe, T.S.: Dynamically rescaled Hamiltonian Monte Carlo for Bayesian Hierarchical Models. arXiv:1806.02068v1 (2018)

Kong, A., Liu, J.S., Wong, W.H.: Sequential Imputations and Bayesian Missing Data Problems. Journal of the American Statistical Association **89**(425), 278–288 (1994)

Lan, S., Stathopoulos, V., Shahbaba, B., Girolami, M.: Lagrangian Dynamical Monte Carlo. Journal of Computational and Graphical Statistics **24**(2) (2015)

Lan, S., Streets, J., Shahbaba, B.: Wormhole Hamiltonian Monte Carlo. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014a)

Lan, S., Zhou, B., Shahbaba, B.: Spherical Hamiltonian Monte Carlo for Constrained Target Distributions. In: Proceedings of the 31st International Conference on Machine Learning, pp. 629–637 (2014b)

Leimkuhler, B., Reich, S.: Simulating Hamiltonian Dynamics. Cambridge University Press, Cambridge, UK (2005)

Levy, D., Hoffman, M.D., Sohl-Dickstein, J.: Generalizing Hamiltonian Monte Carlo with Neural Networks. In: 6th International Conference on Learning Representations (2018)

Lichman, M.: UCI Machine Learning Repository (2013). URL `http://archive.ics.uci.edu/ml`

Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics, Springer, New York (2008)

Livingstone, S., Betancourt, M., Byrne, S., Girolami, M.: On the geometric ergodicity of Hamiltonian Monte Carlo. arXiv:1601.08057v1 (2016)

Livingstone, S., Faulkner, M.F., Roberts, G.O.: Kinetic energy choice in Hamiltonian/hybrid Monte Carlo. arXiv:1706.02649v2 (2017)

Lu, X., Perrone, V., Hasenclever, L., Teh, Y.W., Vollmer, S.J.: Relativistic Monte Carlo. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2017)

Luo, R., Yang, Y., Wang, J., Liu, Y.: Thermostat-assisted Continuous-tempered Hamiltonian Monte Carlo for Multimodal Posterior Sampling. In: NIPS Advances in Approximate Bayesian Inference Workshop (2017)

Ma, Y.A., Fox, E.B., Chen, T., Wu, L.: A Unifying Framework for Devising Efficient and Irreversible MCMC Samplers. arXiv:1608.05973v3 (2016)

Mackenzie, P.B.: An improved hybrid Monte Carlo method. Physics Letters B **226**, 369–371 (1989)

McLachlan, R.I.: On the numerical integration of ordinary differential equations by symmetric composition methods. SIAM Journal on Scientific Computing **16**(1), 151–168 (1995)

Neal, R.M.: Bayesian Learning for Neural Networks. Ph.D. thesis, Dept. of Computer Science, University of Toronto (1994)

Neal, R.M.: Annealed Importance Sampling. Statistics and Computing **11**, 125–139 (2001)

Neal, R.M.: Improving Asymptotic Variance of MCMC Estimators: Non-reversible Chains are Better. Tech. Rep. 0406, Department of Statistics, University of Toronto (2004)

Neal, R.M.: MCMC using Hamiltonian dynamics. In: Handbook of Markov Chain Monte Carlo, vol. 2, pp. 113–162. Chapman & Hall / CRC Press (2011)

Nishimura, A., Dunson, D., Lu, J.: Discontinuous Hamiltonian Monte Carlo for models with discrete parameters and discontinuous likelihoods. arXiv:1705.08510v2 (2018)

Nishimura, A., Dunson, D.B.: Recycling intermediate steps to improve Hamiltonian Monte Carlo. arXiv:1511.06925v1

(2015)
Nishimura, A., Dunson, D.B.: Geometrically Tempered Hamiltonian Monte Carlo. arXiv:1604.00872v2 (2017)
Ohzeki, M., Ichiki, A.: Mathematical understanding of detailed balance condition violation and its application to Langevin dynamics. Journal of Physics: Conference Series **638**(012003) (2015). doi:10.1088/1742-6596/638/1/012003
Ottobre, M.: Markov Chain Monte Carlo and Irreversibility. Reports on Mathematical Physics **77**(3) (2016)
Ottobre, M., Pillai, N.S., Pinski, F.J., Stuart, A.M.: A function space HMC algorithm with second order Langevin diffusion limit. Bernoulli **22**(1), 60–106 (2016)
Pakman, A., Paninski, L.: Auxiliary-variable Exact Hamiltonian Monte Carlo Samplers for Binary Distributions. In: Advances in Neural Information Processing Systems (NIPS), pp. 2490–2498 (2013)
Plummer, M., Best, N., Cowles, K., Vines, K.: CODA: Convergence Diagnosis and Output Analysis for MCMC. R News **6**(1), 7–11 (2006)
Radivojević, T.: Enhancing Sampling in Computational Statistics Using Modified Hamiltonians. Ph.D. thesis, UPV-EHU (2016)
Radivojević, T., Fernández-Pendás, M., Sanz-Serna, J.M., Akhmatskaya, E.: Multi-stage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods. Journal of Computational Physics **373**, 900–916 (2018). doi:10.1016/j.jcp.2018.07.023
Rimoldini, L.: Weighted skewness and kurtosis unbiased by sample size and Gaussian uncertainties. Astronomy and Computing **5**, 1–8 (2014)
Salvatier, J., Wiecki, T.V., Fonnesbeck, C.: Probabilistic programming in python using pymc3. PeerJ Computer Science **2**, e55 (2016)
Sohl-Dickstein, J.: Hamiltonian Monte Carlo with Reduced Momentum Flips. arXiv:1205.1939v1 (2012)
Sohl-Dickstein, J., Culpepper, B.J.: Hamiltonian Annealed Importance Sampling for partition function estimation. arXiv:1205.1925 (2012)
Sohl-Dickstein, J., Mudigonda, M., Deweese, M.: Hamiltonian Monte Carlo Without Detailed Balance. In: Proceedings of the 31st International Conference on Machine Learning, pp. 719–726 (2014)
Stan Development Team: Stan Modeling Language User's Guide and Reference Manual, version 2.17.0 ed. (2017)
Strathmann, H., Sejdinovic, D., Livingstone, S., Szabo, Z., Gretton, A.: Gradient-free Hamiltonian Monte Carlo with efficient kernel exponential families. In: Advances in Neural Information Processing Systems (NIPS), pp. 955–963 (2015)
Suwa, H., Todo, S.: General construction of irreversible kernel in Markov Chain Monte Carlo. arXiv:1207.0258 (2012)
Sweet, C.R., Hampton, S.S., Skeel, R.D., Izaguirre, J.A.: A separable shadow Hamiltonian hybrid Monte Carlo method. The Journal of Chemical Physics **131**, 174106 (2009). doi: doi:10.1063/1.3253687
Tripuraneni, N., Rowland, M., Ghahramani, Z., Turner, R.: Magnetic Hamiltonian Monte Carlo. In: Proceedings of the 34th International Conference on Machine Learning (2017). URL `arXiv:1607.02738v2`
van de Meent, J.W., Paige, B., Wood, F.: Tempering by subsampling. arXiv:1401.7145v1 (2014)
Wang, Z., de Freitas, N.: Predictive adaptation of hybrid Monte Carlo with Bayesian parametric bandits. In: NIPS Deep Learning and Unsupervised Feature Learning Workshop (2011)
Wang, Z., Mohamed, S., de Freitas, N.: Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers. In: Proceedings of the 30th International Conference on Machine Learning, pp. 1462–1470 (2013)
Wee, C.L., Sansom, M.S., Reich, S., Akhmatskaya, E.: Improved sampling for simulations of interfacial membrane proteins: Application of generalized shadow hybrid Monte Carlo to a peptide toxin/bilayer system. The Journal of Physical Chemistry B **112**(18), 5710–5717 (2008)
Yi, K., Doshi-Velez, F.: Roll-back Hamiltonian Monte Carlo. In: 31st Conference on Neural Information Processing Systems (NIPS) (2017)
Zhang, C., Shahbaba, B., Zhao, H.: Hamiltonian Monte Carlo Acceleration Using Surrogate Functions with Random Bases. Statistics and Computing **27**(6), 1473–1490 (2017a)
Zhang, C., Shahbaba, B., Zhao, H.: Precomputing Strategy for Hamiltonian Monte Carlo Method Based on Regularity in Parameter Space. Computational Statistics **32**(1), 253–279 (2017b)
Zhang, C., Shahbaba, B., Zhao, H.: Variational Hamiltonian Monte Carlo via Score Matching. Bayesian Analysis **13**(2), 485–506 (2018)
Zhang, Y., Chen, C., Gan, Z., Henao, R., Carin, L.: Stochastic Gradient Monomial Gamma Sampler. In: Proceedings of the 34th International Conference on Machine Learning (2017c)
Zhang, Y., Ghahramani, Z., Storkey, A.J., Sutton, C.A.: Continuous relaxations for discrete Hamiltonian Monte Carlo. In: Advances in Neural Information Processing Systems (NIPS), pp. 3194–3202 (2012)
Zhang, Y., Sutton, C.: Semi-separable Hamiltonian Monte Carlo for inference in Bayesian hierarchical models. In: Advances in Neural Information Processing Systems (NIPS), pp. 10–18 (2014)
Zhang, Y., Wang, X., Chen, C., Fan, K., Carin, L.: Towards Unifying Hamiltonian Monte Carlo and Slice Sampling. In: Advances in Neural Information Processing Systems (NIPS), pp. 1749–1757 (2016)
Zou, D., Xu, P., Gu, Q.: Stochastic Variance-Reduced Hamilton Monte Carlo Methods. In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp. 6028–6037 (2018)