General and Efficient Bayesian Computation through Hamiltonian Monte Carlo Extensions

by

Akihiko Nishimura

Department of Mathematics
Duke University

Date: _____

Approved:

_____
David Dunson, Supervisor

_____
Jianfeng Lu

_____
Jonathan Mattingly

_____
Sayan Mukherjee

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Mathematics
in the Graduate School of Duke University
2017

Abstract

General and Efficient Bayesian Computation through Hamiltonian Monte Carlo Extensions

by

Akihiko Nishimura

Department of Mathematics
Duke University

Date: _____

Approved:

_____
David Dunson, Supervisor

_____
Jianfeng Lu

_____
Jonathan Mattingly

_____
Sayan Mukherjee

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the Graduate School of Duke University
2017

Abstract

Hamiltonian Monte Carlo (HMC) is a state-of-the-art sampling algorithm for Bayesian computation. Popular probabilistic programming languages Stan and PyMC rely on HMC's generality and efficiency to provide automatic Bayesian inference platforms for practitioners. Despite its wide-spread use and numerous success stories, HMC has several well known pitfalls. This thesis presents extensions of HMC that overcome its two most prominent weaknesses: inability to handle discrete parameters and slow mixing on multi-modal target distributions.

Discontinuous HMC (DHMC) presented in Chapter 2 extends HMC to discontinuous target distributions – and hence to discrete parameter distributions through embedding them into continuous spaces — using an idea of event-driven Monte Carlo from the computational physics literature. DHMC is guaranteed to outperform a Metropolis-within-Gibbs algorithm since, as it turns out, the two algorithms coincide under a specific (and sub-optimal) implementation of DHMC. The theoretical justification of DHMC extends an existing theory of non-smooth Hamiltonian mechanics and of measure-valued differential inclusions.

Geometrically tempered HMC (GTHMC) presented in Chapter 3 improves HMC's performance on multi-modal target distributions. The efficiency improvement is achieved through differential geometric techniques, relating a target distribution to another distribution with less severe multi-modality. We establish a geometric theory behind Riemannian manifold HMC to motivate our geometric tempering methods.

We then develop an explicit variable stepsize reversible integrator for simulating Hamiltonian dynamics to overcome a stability issue of the usual Störmer-Verlet integrator. The integrator is of independent interest, being the first of its kind designed specifically for HMC variants.

In addition to the two extensions described above, Chapter 4 describes a *variable trajectory length* algorithm that generalizes the acceptance and rejection procedure of HMC — and in fact of any reversible dynamics based samplers — to allow for more flexible choices of trajectory lengths. The algorithm in particular enables an effective application of a variable stepsize integrator to HMC extensions, including GTHMC. The algorithm is widely applicable and provides a recipe for constructing valid dynamics based samplers beyond the known HMC variants. Chapter 5 concludes the thesis with a simple and practical algorithm to improve computational efficiencies of HMC and related algorithms over their traditional implementations.

To my family, friends, and all those who helped me along the way.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

The following symbols are used consistently across chapters.

$\lvert \cdot \rvert$	Determinant of a matrix (or an absolute value when applied to a scalar).
$\pi(\cdot)$	Probability density (or mass) function.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$\boldsymbol{\theta}$	Parameter of interest.
$\boldsymbol{p}$	Auxillary momentum variable.
d	Number of model parameters i.e. the dimension of $\boldsymbol{\theta}$.
$U(\boldsymbol{\theta})$	Potential energy function.
$K(\boldsymbol{\theta}, \boldsymbol{p})$	Kinetic energy function, often a function only of $\boldsymbol{p}$.
$H(\boldsymbol{\theta}, \boldsymbol{p})$	Total energy — a sum of the potential and kinetic energy — also referred to as the Hamiltonian.
$\boldsymbol{F}$, $\boldsymbol{F}_\epsilon$	Reversible map on a parameter space. Usually corresponds to one step of an integrator with the stepsize ϵ indicated by a subscript.
$\boldsymbol{R}$	Momentum flip operator. Denotes a more general involution operator in Chapter 4.
$\boldsymbol{\Psi}_t$	Solution operator of a differential equation at time t.
$\boldsymbol{I}$	Identity matrix.
$\boldsymbol{e}_i$	ith standard basis vector.

Terminologies

integrator	algorithm that numerically approximates the exact solution to a differential equation.
Stan	probabilistic programming language for Bayesian inference written in C++ with various interfaces including R and command line.
PyMC	probabilistic programming language for Bayesian inference written in Python with a Python interface.

Abbreviations

DHMC	Discontinuous Hamiltonian Monte Carlo
ESS	Effective sample sizes
GTHMC	Geometrically tempered Hamiltonian Monte Carlo
HMC	Hamiltonian Monte Carlo
MCMC	Markov chain Monte Carlo
RMHMC	Riemann manifold Hamiltonian Monte Carlo
VTL-CHMC	Variable trajectory length compressible Hamiltonian Monte Carlo

Acknowledgements

First, I would like to express my deepest gratitude to my advisor David Dunson for all his support and guidance. I am fortunate that he is very open-minded and willing to work with people of diverse backgrounds including myself. Working with and learning form him had an enormously positive impact on my career and led me to many wonderful opportunities I could not have imagined. Also, his constant energy and excitement with research always provided additional push to help me navigate through occasionally frustrating experiences of being a graduate student.

I am also indebted to Jonathan Mattingly for his help and guidance, especially during the early years of my Ph.D. career when I was still searching for my academic passion and career goals. Thanks also to my mentors and collaborators Jianfeng Lu, Beth Hauser, Bill Kraus, and Amilcare Porporato — I learned tremendously from them and had many valuable experiences while working together.

I am grateful to my fellow classmates and friends for making my time at Duke enjoyable and memorable. Thanks especially to Henri Roesch for being my long-time office neighbor and keeping the office always a cheerful and friendly place.

I would like to thank the departmental staffs for providing essential supports for my study. Special thanks to Andrew Schretter for always trouble-shooting computer issues promptly and to Wayne Williamson for finding a way through the labyrinth of administrative procedures — together with Director of Graduate Studies Rick Durrett — to make my summer internship possible.

Finally, I would like to acknowledge National Science Foundation (NSF), Statistical and Applied Mathematical Sciences Institute (SAMSI), Health Effects Institute (HEI), International Society for Bayesian Analysis (ISBA), Section on Bayesian Statistical Science of American Statistical Association (ASA) for providing financial support for my education and professional developments.

1

Introduction

Bayesian inference is a powerful and versatile paradigm to extract information from a variety of complex data sets, providing a unified framework for inference and decision-making based on a posterior distribution obtained by Bayes' rule. The Bayesian framework has a number of theoretical advantages over alternatives while being highly intuitive at the same time (Berger, 2013). Despite its conceptual simplicity, however, one serious challenge in practice is computing a posterior distribution. Most posterior distributions, and hence corresponding posterior summaries of interest, are analytically intractable and must be approximated by computational methods.

Markov chain Monte Carlo algorithms (MCMC) are arguably the most widely used tools in applied Bayesian modeling, providing a means to generate samples from posterior distributions (Brooks et al., 2011). Given an unnormalized target distribution $\pi(\cdot)$, MCMC generates a sequence of correlated samples $\{\boldsymbol{\theta}^{(i)}\}_{i\geqslant 1}$ whose empirical measure $N^{-1}\sum_{i=1}^{N}\delta_{\boldsymbol{\theta}^{(i)}}(\cdot)$ converges to the target in the limit $N\to\infty$. Efficiency of an MCMC algorithm is determined by both its mixing rate — magnitude of auto-correlation in the generated sequence $\{\boldsymbol{\theta}^{(i)}\}_{i\geqslant 1}$ — and computational speed for generating the sequence.

One common way to quantify the mixing rate of MCMC is through *effective sample sizes* (ESS), which compare the quality of the MCMC samples $\{\boldsymbol{\theta}^{(i)}\}_{i\geqslant 1}$ to a sequence of independent samples. More precisely, when estimating $\mathbb{E}_\pi[g(\boldsymbol{\theta})]$ with an MCMC estimator $N^{-1}\sum_{i=1}^N g(\boldsymbol{\theta}^{(i)})$, the ESS of $\{\boldsymbol{\theta}^{(i)}\}_{i\geqslant 1}$ with respect to the function g is defined to be

$$\mathrm{ESS}_g\left(\{\boldsymbol{\theta}^{(i)}\}_{i=1}^N\right) = \left[\frac{\mathrm{var}\left(N^{-1}\sum_{i=1}^N g(\boldsymbol{\theta}^{(i)})\right)}{\mathrm{var}_\pi\left(g(\boldsymbol{\theta})\right)}\right]^{-1} \tag{1.1}$$

Correspondingly, an asymptotic statistical efficiency of MCMC is characterized by

$$\lim_{N\to\infty} N^{-1}\,\mathrm{ESS}_g\left(\{\boldsymbol{\theta}^{(i)}\}_{i=1}^N\right) = \left[1 + 2\sum_{k=1}^{\infty}\mathrm{corr}_{\boldsymbol{\theta}^{(1)}\sim\pi}\left(g(\boldsymbol{\theta}^{(1)}), g(\boldsymbol{\theta}^{(k+1)})\right)\right]^{-1} \tag{1.2}$$

where the correlation terms are computed at stationarity $\boldsymbol{\theta}^{(1)} \sim \pi(\cdot)$. It is clear from Equation (1.2) that auto-correlations among successive samples negatively affect the efficiency of an MCMC algorithm.

Specialized MCMC algorithms exist for restricted model classes, but most general-purpose MCMC algorithms — adaptable to a broad range of posterior distributions — are often highly inefficient and scale poorly in the number of parameters (Roberts and Rosenthal, 2001). Originally proposed by Duane et al. (1987), and more recently popularized in the statistics community through the works of Neal (1996a, 2010), Hamiltonian Monte Carlo (HMC) promises a better scalability (Neal, 2010; Beskos et al., 2013) and has enjoyed wide-ranging successes as one of the most reliable approaches in general settings (Gelman et al., 2013; Kruschke, 2014; Monnahan et al., 2016). Each iteration of HMC requires multiple gradient evaluations and hence is more computationally intensive than that of simpler algorithms such as random-walk Metropolis and Metropolis-adjusted Langevin algorithm. On the other hand, HMC is capable of generating samples with much weaker auto-correlations even in a complex

high-dimensional distribution (Betancourt, 2017). Due to its faster mixing rate, HMC typically demonstrates a clear advantage over others in overall efficiency as the number of parameters and model complexity grow (Neal, 2010; Beskos et al., 2013; Monnahan et al., 2016).

The generality and efficiency of HMC has made it not only a valuable tool for Bayesian statisticians but also for much wider applied statistics communities through development of probabilistic programming languages such as Stan and PyMC (Stan Development Team, 2016; Salvatier et al., 2016). These popular software packages rely on HMC and its variants to carry out posterior inferences automatically and efficiently. Such automatic Bayesian inference platforms have been essential catalysts in a wider adoption of Bayesian methodologies since many practitioners may understand the principles of Bayesian analysis but lack necessary expertise in computational methods (Lunn et al., 2009). Even for experts in Bayesian computation, these softwares are valuable time-savers since implementing a specialized algorithm often demands a great deal of time and effort. In fact, as data analysis typically requires multiple iterations of fitting and diagnosing different models, time and effort required to manually implement individual models often add up to a serious burden (Kucukelbir et al., 2015).

This thesis develops extensions of HMC to significantly expand its scope and improve its efficiency. These extensions retain the generality of HMC that makes it suitable for deployment in automatic Bayesian inference softwares. Given HMC's proven utility for such purposes, these extensions could have far-reaching impacts by further expanding practitioners' ability to efficiently employ Bayesian approaches in a wide range of models.

The reminder of this chapter introduces the fundamental ideas and concepts behind HMC and its existing extensions, as well as an overview of the existing literature and our contributions. Discussions in the subsequent chapters build on the

notions and terminologies introduced in Section 1.1 and 1.2. The presentation of the HMC framework here is meant to be as concise as possible while being self-contained; for more detailed expositions, we refer the reader to Neal (2010), Betancourt (2017), and Barp et al. (2017).

1.1 General formulation of HMC and related algorithms

Given a parameter of interest $\boldsymbol{\theta}$ with (unnormalized) density $\pi_\Theta(\cdot)$, HMC and its variants introduce an auxiliary *momentum* variable $\boldsymbol{p}$ and sample from the joint distribution

$$\pi(\boldsymbol{\theta}, \boldsymbol{p}) \propto \pi_\Theta(\boldsymbol{\theta}) \pi_{P \mid \Theta}(\boldsymbol{p} \mid \boldsymbol{\theta}) \tag{1.3}$$

where $\pi_{P \mid \Theta}(\cdot \mid \boldsymbol{\theta})$ is chosen to be a symmetric distribution. Uniquely defined up to normalizing constants of $\pi_\Theta(\boldsymbol{\theta})$ and $\pi_{P \mid \Theta}(\boldsymbol{p} \mid \boldsymbol{\theta})$, the functions $U(\boldsymbol{\theta}) = -\log \pi_\Theta(\boldsymbol{\theta})$ and $K(\boldsymbol{\theta}, \boldsymbol{p}) = -\log \pi_{P \mid \Theta}(\boldsymbol{p} \mid \boldsymbol{\theta})$ are referred to as the *potential energy* and *kinetic energy* due to the physical laws that motivate HMC. The total energy $H(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + K(\boldsymbol{\theta}, \boldsymbol{p})$ is often called the *Hamiltonian*.

HMC and its variants generate a Metropolis proposal on the joint space $(\boldsymbol{\theta}, \boldsymbol{p})$ by simulating trajectories of *Hamiltonian dynamics* in which the evolution of the state $(\boldsymbol{\theta}, \boldsymbol{p})$ is governed by *Hamilton's equations*:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \nabla_{\boldsymbol{p}} H(\boldsymbol{\theta}, \boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \boldsymbol{p}) \tag{1.4}$$

Hamiltonian dynamics turns out to be a useful proposal generation mechanism because the distribution $\pi(\boldsymbol{\theta}, \boldsymbol{p})$ is invariant under the dynamics (1.4). In other words, if $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \geqslant 0}$ is the solution of (1.4) with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) \sim \pi(\cdot)$, then $(\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau)) \sim \pi(\cdot)$ for any $\tau > 0$. This allows a proposal generated by an approximate solution of Hamiltonian dynamics to be far away from the current state yet accepted with high probability. The classical theory behind (1.4) is restricted to

a smooth $H(\boldsymbol{\theta}, \boldsymbol{p})$ (and hence to a smooth density $\pi_\Theta(\boldsymbol{\theta})$), and so are the existing HMC extensions.

The solution to (1.4) is analytically intractable in general and must be approximated numerically. We will use a terminology from numerical analysis; *integrator* refers to an algorithm that numerically approximates an evolution of the exact solution to a differential equation. In order to generate a valid Metropolis (-Hastings) proposal, an integrator must retain the *reversibility* and *volume-preserving* properties of Hamiltonian dynamics. See Neal (2010) and Betancourt (2017) for more on reversible and volume-preserving integrators and their relations to HMC. Volume-preservation is actually not required under a more general framework (see Chapter 4), but is assumed here for a simpler presentation.

Algorithm 1.1 summarizes the general HMC framework, which covers a large proportion of the existing HMC extensions. The same basic formulations are shared by the HMC variants on geometrically constrained spaces as well as those on infinite dimensional spaces (Barp et al., 2017).

Algorithm 1.1 (General HMC framework)**.** Generate a Markov chain with the following transition rule $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$:

1) Sample the momentum from its conditional distribution $\boldsymbol{p} \,|\, \boldsymbol{\theta} \sim \pi_{P \,|\, \Theta}(\cdot \,|\, \boldsymbol{\theta})$.[1]

2) Using a reversible and volume-preserving integrator, approximate the solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \geqslant 0}$ of the differential equation (1.4) with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}, \boldsymbol{p})$. Use the approximate solution $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) \approx (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ for some $\tau > 0$ as a proposal.

3) Accept or reject the proposal with the Metropolis acceptance probability

$$\min \left\{ 1, \exp \left(-H(\boldsymbol{\theta}^*, \boldsymbol{p}^*) + H(\boldsymbol{\theta}, \boldsymbol{p}) \right) \right\}. \tag{1.5}$$

[1] The distribution $\pi_{P \,|\, \Theta}(\cdot \,|\, \boldsymbol{\theta})$ is still assumed to be symmetric here. The symmetry requirement can be relaxed under a more general framework discussed in Chapter 4.

With an accurate integrator, the acceptance probability (1.5) of the proposal $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ can be close to 1 because the exact solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \geqslant 0}$ of Hamiltonian dynamics satisfies the *conservation of energy property*: $H(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = H(\boldsymbol{\theta}(0), \boldsymbol{p}(0))$ for all $t \geqslant 0$.

1.2 Standard HMC with Gaussian momentum and leapfrog integrator

Here we introduce the standard version of HMC based on a Gaussian momentum $\pi_{P \mid \Theta}(\boldsymbol{p} \mid \boldsymbol{\theta}) = \pi_P(\boldsymbol{p}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{M})$ and the *leapfrog* integrator to approximate the solution of (1.4). The covariance matrix $\boldsymbol{M}$ is often referred to as a *mass matrix*. Aside from the additional automation of trajectory lengths via the *no-U-turn* algorithm of Hoffman and Gelman (2014), the default sampler of Stan and PyMC coincides with this basic version of HMC with a diagonal mass matrix. (No-U-turn algorithm only facilitates the tuning of HMC and otherwise does not improve its performance in any way (Wang et al., 2013).) Whenever we simply say "HMC," we refer to this standard version.

When the distribution of $\boldsymbol{p}$ is independent of $\boldsymbol{\theta}$, we have $K(\boldsymbol{\theta}, \boldsymbol{p}) = K(\boldsymbol{p})$ and Hamilton's equation (1.4) becomes

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{1.6}$$

The *leapfrog* scheme approximates the evolution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) \to (\boldsymbol{\theta}(t+\epsilon), \boldsymbol{p}(t+\epsilon))$ of the dynamics (1.6) through the successive updates as follows:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \quad \boldsymbol{p} \leftarrow \boldsymbol{p} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{1.7}$$

Taking $L = L(\epsilon) = \lfloor \tau / \epsilon \rfloor$ steps of the integrator approximates the evolution $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) \to (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ and recovers it exactly in the limit $\epsilon \to 0$. For a fixed τ, the *stepsize* ϵ controls the trade-off between the accuracy and computational cost:

a smaller ϵ leads to a more accurate approximation but requires a larger number $L = \lfloor \tau/\epsilon \rceil$ of gradient evaluations. The number of numerical integration steps L is often called the *trajectory length* or *path length*.

1.3 Overview of existing literature and our contributions

1.3.1 HMC extensions to non-Euclidean spaces

HMC has been extended to non-Euclidean spaces such as differentiable manifolds (Byrne and Girolami, 2013), Hilbert spaces of functions (Beskos et al., 2011), and spaces with inequality constraints (Neal, 2010; Pakman and Paninski, 2014). While these spaces are more general than a Euclidean space, they are still continuous parameter spaces with differentiable structure. HMC extensions in such spaces are naturally motivated by a general theory of Hamiltonian dynamics in (infinite-dimensional) differentiable manifolds (Abraham and Marsden, 1978).

What has not been addressed so far is HMC's inability to handle discrete parameters, despite the fact that it has been widely considered as the main limitation of HMC in typical statistical applications (Gelman et al., 2015; Monnahan et al., 2016). *Discontinuous HMC* (DHMC) in Chapter 2 solves this outstanding problem by extending HMC to discontinuous target distributions and hence to discrete parameter distributions through embedding them into continuous spaces. The use of a surrogate smooth target distribution for discrete parameters has been proposed for very specific models (Zhang et al., 2012), but DHMC requires no such model specific constructions and handles discrete parameters in a much more general manner.

DHMC employs an independent Laplace distribution $\pi_P(\boldsymbol{p}) \propto \exp\left(-\sum_i m_i^{-1}|p_i|\right)$ for momentum since Hamiltonian dynamics corresponding to the usual Gaussian momentum is computationally problematic for discontinuous target distributions. We develop a novel integrator based on coordinate-wise integration of Hamilton's equation corresponding to a Laplace momentum. DHMC is guaranteed to outperform

a Metropolis-within-Gibbs algorithm since, as it turns out, the two algorithms coincide under a specific (and sub-optimal) implementation of DHMC. We also show that Laplace momentum based Hamiltonian dynamics for DHMC has a remarkable similarity to a *zig-zag* process — a continuous-time Markov process used to construct a state-of-the-art non-reversible sampler (Bierkens et al., 2016).

1.3.2 Sampling from multi-modality

The difficulty of sampling from multi-modal target distributions has been well recognized both in the statistics and computational physical science community (Berg and Neuhaus, 1992; Neal, 1996b; Wang and Landau, 2001; Neal, 2001; Earl and Deem, 2005; Kou et al., 2006; Liang et al., 2011). Like most other algorithms, HMC suffers from slow mixing in the presence of multi-modality in the target (Neal, 2010; Chapter 3). While the tempered transition of Neal (1996b) has been suggested as a possible way to alleviate this problem, it compromises the high acceptance probability of HMC proposals and hence HMC's efficiency in general. Moreover, little guidance exists on how to tune the degree of tempering and other tuning parameters of HMC.

Geometrically tempered HMC (GTHMC) in Chapter 3 addresses the issue of multi-modality in a more principled way without compromising the utility of HMC as a general-purpose sampler. GTHMC uses differential geometric techniques to alleviate HMC's tendency to get stuck around each mode and is motivated by a new geometric theory behind Riemann manifold HMC (RMHMC) of Girolami and Calderhead (2011). GTHMC in particular demonstrates the use of their geometric method beyond the original purpose of pre-conditioning complex target distributions. While Lan et al. (2014) also explores an application of the RMHMC framework to multi-modal target distributions, their algorithm requires tedious ad-hoc manual tunings which make the algorithm highly impractical. GTHMC does not require any such ad-hoc tricks by virtue of more precise understanding and treatment of geometry

underlying RMHMC. GTHMC can also be combined with a meta-algorithm such as parallel tempering (Earl and Deem, 2005) to further improve its ability to deal with severe multi-modality.

The particular form of a momentum distribution used by GTHMC requires an integrator that locally adapts its stepsize depending on the current parameter value. Such an integrator has never been considered in the HMC literature, so we devise a new adaptive integrator specifically designed for RMHMC applications. The use of an adaptive integrator in HMC contexts creates an additional challenge, however; local stepsize adaptation necessarily breaks the volume-preserving property of Hamiltonian dynamics. (Incidentally, this issue of volume-preservation partially explains the lack of previous research in developing adaptive integrators suitable for HMC applications.) In order to enable an effective application of adaptive integrators to HMC and related algorithms, therefore, we develop a generalized acceptance and rejection scheme in Chapter 4. The resulting algorithm *variable trajectory length compressible HMC* (VTL-CHMC) extends the framework of Fang et al. (2014) and provides one of the most general formulations of HMC-type algorithms.

1.3.3 Improving HMC efficiency in general settings

While motivated by the specific problems of a discrete parameter space and multi-modality, the techniques developed for DHMC and GTHMC push the boundary of the traditional HMC framework in many ways. As such, they open up new avenues of research and could serve as catalysts for development of HMC variants with further improvement in efficiency and application scopes. We first describe the existing research on improving HMC efficiency and then point out the ways in which this thesis provides fresh perspectives.

For a given target distribution, the efficiency of HMC is determined by choice of the stepsize ϵ, trajectory length L (or $\tau = \epsilon L$), and mass matrix $\boldsymbol{M}$. How to optimally

tune these parameters has been well studied and has been addressed satisfactorily for most practical purposes (Neal, 2010; Hoffman and Gelman, 2014; Beskos et al., 2013; Wang et al., 2013; Betancourt, 2013; Betancourt et al., 2014; Bou-Rabee and Sanz-Serna, 2015; Stan Development Team, 2016). Another major research effort to improve HMC's efficiency has focused on data subsampling approaches to reduce the computational cost of computing $\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta})$ (Welling and Teh, 2011; Chen et al., 2014; Ding et al., 2014; Shang et al., 2015). The speed gain through subsampling comes at the cost of a biased inference, however, and it is questionable if these approaches are reliable for inference from complex data sets commonly encountered in modern statistical applications (Betancourt, 2015). Neal (2010) and Barp et al. (2017) describe some other techniques that have been proposed to improve on standard HMC.

Beyond standard HMC, the efficiency of Algorithm 1.1 depends critically on the choice of $\pi_{P \mid \Theta}(\boldsymbol{p} \mid \boldsymbol{\theta}) \propto \exp(-K(\boldsymbol{\theta}, \boldsymbol{p}))$ as well as the computational speed and accuracy of the integrator in approximating Hamiltonian dynamics. It is of great practical interest, therefore, to find a choice of $K(\boldsymbol{\theta}, \boldsymbol{p})$ and an accompanying integrator that improve the efficiency of HMC in a general and fundamental way. There has been much less progress in this area compared to the other research areas discussed above. In fact, aside from the notable exception of Girolami and Calderhead (2011), few of the existing attempts at extending the choice of $K(\boldsymbol{\theta}, \boldsymbol{p})$ achieve a fundamental efficiency improvement in general settings.

The limited success in deployment of a more general momentum distribution seems to be due to the following reasons. In addition to theoretical challenges in characterizing behavior of the corresponding Hamiltonian dynamics, a nonstandard choice of $K(\boldsymbol{\theta}, \boldsymbol{p})$ often requires a more advanced integrator beyond the conventional ones. Such an integrator must not only preserve the structure of Hamiltonian dynamics but also strike a balance between computational costs and numerical accuracy. A

new proposal of $K(\boldsymbol{\theta}, \boldsymbol{p})$ must address all these challenges at the same time, requiring techniques from a variety of fields including numerical analysis, differential geometry, and computational physics and chemistry (Betancourt, 2017).

In addressing the prominent weaknesses of HMC in this thesis, we propose new families of $K(\boldsymbol{\theta}, \boldsymbol{p})$ as well as new classes of integrators to enable their deployments in practical problems. The DHMC chapter considers an independent Laplace momentum along with a novel coordinate-wise integrator. The potential utility of the corresponding HMC variant, beyond its original purpose to deal with discontinuous target densities, is explored theoretically in Section 2.4.4 and empirically in Section 2.E.1 with promising results.

The GTHMC chapter develops new geometric insights behind the RMHMC family of $K(\boldsymbol{\theta}, \boldsymbol{p})$ and identifies a subclass that helps RMHMC better cope with multi-modal targets. We then move on to develop not only a new class of adaptive stepsize integrators but also a generalization of the traditional HMC framework to enable effective applications of such integrators. With the traditional framework built on fixed stepsize approximations of Hamiltonian dynamics, the computational efficiency of HMC is often dominated by a small region of the parameter space where an integrator requires a very small stepsize (Neal, 2010; Stan Development Team, 2016). The VTL-CHMC framework of Chapter 4 removes such constraints, allowing a much wider range of numerical approximation schemes for Hamiltonian dynamics.

To summarize, in dealing with the two most prominent weaknesses of HMC, this thesis also tackles the related issues that so far few have successfully managed to address. With the demonstrated compelling use cases in important application areas, the ideas and techniques presented in this thesis should help future research efforts to further improve HMC in a general and fundamental way.

2

Discontinuous Hamiltonian Monte Carlo for sampling discrete parameters

2.1 Introduction

Often quoted as the main limitation of HMC is its lack of support for discrete parameters (Gelman et al., 2015; Monnahan et al., 2016). Inference problems involving discrete parameters come up in a wide range of fields, including ecology, social science, system reliability, and epidemiology (Berger et al., 2012; Schwarz and Seber, 1999; Warren and Warren, 2013; Basu and Ebrahimi, 2001; Parkin and Bray, 2009). The difficulty in extending HMC to a discrete parameter space stems from the following fact — the construction of HMC proposals relies on a numerical solution of a differential equation whose definition, under the classical theory, makes sense only for a smooth density function. The use of a surrogate continuous target distribution may be possible in some special cases (Zhang et al., 2012), but approximating a discrete parameter of a likelihood by a continuous one is generally difficult (Berger et al., 2012).

This chapter presents *discontinuous HMC* (DHMC), an extension that can effi-

ciently explore discrete parameter spaces as well as continuous ones. DHMC can also handle discontinuous posterior densities, which for example arise from models with structural change points (Chib, 1998; Wagner et al., 2002), latent thresholds (Neelon and Dunson, 2004; Nakajima and West, 2013), and pseudo-likelihoods (Bissiri et al., 2016). DHMC retains the generality that makes HMC suitable for automatic posterior inference as in Stan and PyMC. For any given target distribution, each iteration of DHMC only requires evaluations of the density and of the following quantities:

1. full conditional densities of discrete parameters (up to normalizing constants)
2. either the gradient of the log density with respect to continuous parameters or their individual full conditional densities (up to normalizing constants)

Evaluations of full conditionals can be done algorithmically and efficiently through directed acyclic graph frameworks, taking advantage of conditional independence structures (Lunn et al., 2009). Algorithmic evaluation of the gradient is also efficient (Griewank and Walther, 2008) and its implementations are widely available as open-source modules (Carpenter et al., 2015).

In our framework, the discrete parameters of a model are first embedded into a continuous space, inducing parameters with piecewise constant densities. The key theoretical insight is that Hamiltonian dynamics with a discontinuous potential energy can be integrated analytically near its discontinuity in a way that exactly preserves the total energy. This fact was realized by Pakman and Paninski (2013) and used to sample from binary distributions through embedding them into a continuous space. Afshar and Domke (2015) explored, to a limited extent, applying the same idea in extending HMC more generally. These algorithms are instances of *event-driven Monte Carlo* in the computational physics literature, dating back to Alder and Wainwright (1959).

Another recent development in extending HMC to a more complex parameter

space is Dinh et al. (2017). Though related, their work and ours have little overlap in terms of the main contributions. Their work is more of a proof of concept to demonstrate how HMC could be extended to a parameter space involving trees. We instead focus on the issue of discrete parameters and develop techniques not only of theoretical interests but also of immediate utility, demonstrating concrete and significant improvements over existing approaches.

Several novel techniques are introduced to turn the basic idea of event-driven HMC into a general and practical sampling algorithm for discrete parameters and, more broadly, target distributions with discontinuous densities. We propose an independent Laplace distribution for the momentum variable as a more effective alternative to the usual Gaussian distribution in dealing with discontinuous target distributions. We develop an efficient integrator of Hamiltonian dynamics based on a Laplace momentum by splitting the differential operator into its coordinate-wise components. This integrator exactly preserves the Hamiltonian and leads to a type of *rejection-free* Markovian transitions (Peters and de With, 2012). As it turns out, when applying only one step of the proposed integrator, DHMC coincides with a Metropolis-within-Gibbs sampler. This fact guarantees the theoretical superiority of properly tuned DHMC. DHMC indeed outperforms a Metropolis-within-Gibbs in practice because DHMC can take advantage of the momentum information and induce large transition moves by taking multiple integration steps.

The rest of the chapter is organized as follows. We start Section 2.2 by describing our strategy for embedding discrete parameters into a continuous space in such a way that the resulting distribution can be explored efficiently by DHMC. We then develop a general technique for handling discontinuous target densities within the HMC framework based on the idea of event-driven Monte Carlo. In Section 2.3, we discuss the shortcomings of a Gaussian momentum in extending HMC to discontinuous targets and propose a Laplace momentum as a more effective alternative. Section 2.4

examines theoretical properties of DHMC with a Laplace momentum and establishes its connections to Metropolis-within-Gibbs and a zig-zag sampler. Section 2.5 presents simulation results with real data examples to demonstrate that DHMC outperforms alternative approaches.

2.2 Event-driven HMC for discrete and discontinuous distributions

While Hamilton's equation (1.4) makes no sense when $\boldsymbol{\theta}$ is discrete, this section shows how to extend the HMC framework to accommodate discrete parameters.

2.2.1 Embedding discrete parameters into continuous space

Let N denote a discrete parameter with prior distribution $\pi_N(\cdot)$ and, without loss of generality, assume that N takes positive integer values $\{1, 2, 3, \ldots\}$. For example, the inference goal may be estimation of the population size N given the observation $y \,|\, p, N \sim \text{Binom}(p, N)$ and the objective prior $\pi_N(N) \propto N^{-1}$ (Berger et al., 2012). We embed N into a continuous space by introducing a latent parameter $\widetilde{N}$ whose relationship with N is defined to be

$$N = n \quad \text{if and only if} \quad \widetilde{N} \in (a_n, a_{n+1}] \tag{2.1}$$

for an increasing sequence of real numbers $0 = a_1 \leqslant a_2 \leqslant a_3 \leqslant \ldots$. To match the prior distribution of N, the corresponding (piecewise constant) prior density of $\widetilde{N}$ is given by

$$\pi_{\widetilde{N}}(\tilde{n}) = \sum_{n \geqslant 1} \frac{\pi_N(n)}{a_{n+1} - a_n} \mathbb{1}\{a_n < \tilde{n} \leqslant a_{n+1}\} \tag{2.2}$$

where the Jacobian-like factor $(a_{n+1} - a_n)^{-1}$ adjusts for embedding into non-uniform intervals.

Although the choice $a_n = n$ for all n is an obvious one, a non-uniform embedding is useful in effectively carrying out a parameter transformation of N. For example,

a log-transform embedding $a_n = \log n$ substantially improves the mixing of DHMC when the target distribution is a heavy-tailed function of N and/or $\log N$ has weaker correlations with the rest of the parameters. Similar parameter transformations for continuous parameters are common techniques when using the standard HMC and are also applicable to DHMC. For more on the relationship between the target distribution structure and the mixing rate of HMC as well as techniques to improve the mixing, see Stan Development Team (2016); Betancourt (2017); Livingstone et al. (2016).

While this embedding strategy can be applied whether or not the values of a discrete variable have some natural ordering, embedding the discrete values in an arbitrary order likely induces a continuous distribution with multiple modes. DHMC can be applied regardless of the embedding order, but its mixing rate generally suffers from multi-modality for the same reason as HMC (Neal, 2010; Nishimura and Dunson, 2016).

2.2.2 How HMC fails on discontinuous target densities

When $U(\cdot) = -\log \pi_\Theta(\cdot)$ is smooth, the leapfrog scheme approximates the evolution $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) \to (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ of the dynamics (1.6) within an error of order $O(\epsilon^2)$. More precisely, if $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ denotes the map corresponding to $L = \lfloor \tau/\epsilon \rfloor$ leapfrog steps as defined in (1.7), then we have $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau)) + O(\epsilon^2)$ and $H(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = H(\boldsymbol{\theta}, \boldsymbol{p}) + O(\epsilon^2)$ (Neal, 2010; Leimkuhler and Reich, 2005). When $\pi_\Theta(\cdot)$ has a discontinuity, however, the leapfrog updates (1.7) completely fail to account for the instantaneous change in $\pi_\Theta(\cdot)$ and in general incur an error of order $O(1)$. The standard HMC implementation therefore comes with no guarantee of a decent acceptance rate when the parameter of interest $\boldsymbol{\theta}$ has a discontinuous density.

The above issue is not unique to the leapfrog scheme. Most integrators are designed for differential equations with smooth time derivatives and do not account

for discontinuities. If the discontinuity boundaries of its potential energy can be detected, however, Hamiltonian dynamics can be integrated so as to account for the instantaneous change in $\pi_\Theta(\cdot)$. We now describe how to make sense of Hamiltonian dynamics corresponding to a discontinuous potential energy $U(\boldsymbol{\theta}) = -\log \pi_\Theta(\boldsymbol{\theta})$ whose discontinuity set forms a piecewise smooth manifold.

2.2.3 Event-driven approach at discontinuity

While a discontinuous function does not have a derivative in a classical sense, the gradient $\nabla U(\boldsymbol{\theta})$ can be defined through a notion of *distributional derivatives* and the corresponding Hamilton's equations (1.4) can be interpreted as a *measure-valued differential inclusion* (Stewart, 2000). A solution of a (measure-valued) differential inclusion problem is in general not unique unlike that of a smooth ordinary differential equation. To find the solution that preserves the critical properties of Hamiltonian dynamics, therefore we rely on a so-called *selection principle* (Ambrosio, 2008) and construct a solution with desirable properties as follows.

Define a sequence of smooth approximations $U_\delta(\boldsymbol{\theta})$ of $U(\boldsymbol{\theta})$ for $\delta > 0$ through the convolution $U_\delta = U * \phi_\delta$ with $\phi_\delta(\boldsymbol{\theta}) = \delta^{-d}\phi(\delta^{-1}\boldsymbol{\theta})$ for a compactly supported smooth function $\phi \geqslant 0$ such that $\int \phi = 1$. Here the integer d denotes the dimension of $\boldsymbol{\theta}$. Now let $(\boldsymbol{\theta}_\delta(t), \boldsymbol{p}_\delta(t))$ be the solution of Hamilton's equations with the potential energy U_δ. It can then be shown that the pointwise limit $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \lim_{\delta \to 0}(\boldsymbol{\theta}_\delta(t), \boldsymbol{p}_\delta(t))$ exists for almost every initial condition and we define the dynamics corresponding to $U(\boldsymbol{\theta})$ as this limit. This construction in particular provides a rigorous mathematical foundation for the special cases of discontinuous Hamiltonian dynamics derived by Pakman and Paninski (2013) and Afshar and Domke (2015) through physical intuitions.

The behavior of the limiting dynamics near the discontinuity is deduced as follows. Suppose that the trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ hits the discontinuity at an event time t_e

and let t_e^- and t_e^+ denote infinitesimal moments before and after that. Since the discontinuity set of $U(\boldsymbol{\theta})$ was assumed to be piecewise smooth, at a discontinuity point $\boldsymbol{\theta}$ we have

$$\lim_{\delta \to 0} \nabla_{\boldsymbol{\theta}} U_\delta(\boldsymbol{\theta}) / \|\nabla_{\boldsymbol{\theta}} U_\delta(\boldsymbol{\theta})\| = \boldsymbol{\nu}(\boldsymbol{\theta}) \tag{2.3}$$

where $\boldsymbol{\nu}(\boldsymbol{\theta})$ denotes a unit vector orthonormal to the discontinuity boundary pointing in the direction of higher potential energy.[1] The relations (2.3) and $\mathrm{d}\boldsymbol{p}_\delta/\mathrm{d}t = -\nabla_{\boldsymbol{\theta}} U_\delta$ imply that the only change in $\boldsymbol{p}(t)$ upon encountering the discontinuity occurs in the direction of $\boldsymbol{\nu}_e = \boldsymbol{\nu}(\boldsymbol{\theta}(t_e))$ i.e.

$$\boldsymbol{p}(t_e^+) = \boldsymbol{p}(t_e^-) - \gamma\, \boldsymbol{\nu}_e \tag{2.4}$$

for some $\gamma > 0$. There are two possible types of change in $\boldsymbol{p}$ depending on the potential energy difference ΔU_e at the discontinuity, which we formally define as

$$\Delta U_e = \lim_{\epsilon \to 0^+} U\left(\boldsymbol{\theta}(t_e) + \epsilon \boldsymbol{p}(t_e^-)\right) - U\left(\boldsymbol{\theta}(t_e^-)\right) \tag{2.5}$$

When the momentum does not provide enough kinetic energy to overcome the potential energy increase ΔU_e, the trajectory bounces back against the plane orthogonal to $\boldsymbol{\nu}_e$. Otherwise, the trajectory moves through the discontinuity by transferring kinetic energy to potential energy. Either way, the magnitude of an instantaneous change γ can be determined via the energy conservation law:

$$K(\boldsymbol{p}(t_e^+)) - K(\boldsymbol{p}(t_e^-)) = U(\boldsymbol{\theta}(t_e^-)) - U(\boldsymbol{\theta}(t_e^+)) \tag{2.6}$$

Figure 2.1 — which is explained in more details in Section 2.3 — provides a visual illustration of the trajectory behavior at a discontinuity.

Pakman and Paninski (2013) and Afshar and Domke (2015) propose an event-driven integrator for a discontinuous Hamiltonian dynamics with a Gaussian momentum $K(\boldsymbol{p}) = \|\boldsymbol{p}\|^2/2$. The implementation of this integrator within our context is

[1] More precisely, the statement holds provided that $\boldsymbol{\theta}$ does not belong to an intersection of multiple discontinuity boundaries and has a well-defined orthonormal vector at $\boldsymbol{\theta}$.

described in Appendix 2.C. In Section 2.3, we describe a major shortcoming of approaches based on a Gaussian momentum and develop a novel event-driven integrator based an alternative momentum distribution.

2.3 Discontinuous HMC with Laplace momentum

2.3.1 Issue with Gaussian momentum

Most of the existing variations of HMC assume a (conditionally) Gaussian distribution on the momentum variable. It is a natural choice in the original molecular dynamics applications (Duane et al., 1987) and is arguably the most intuitive in terms of the underlying physics (Neal, 2010; Nishimura and Dunson, 2016). Correspondingly, the existing event-driven HMC algorithms use a Gaussian momentum (Pakman and Paninski, 2013; Afshar and Domke, 2015).

For sampling discrete parameters, however, discontinuous Hamiltonian dynamics based on a Gaussian momentum have a serious shortcoming. In order to approximate the dynamics accurately, an integrator must detect every single discontinuity encountered by a trajectory and then compute the potential energy difference each time (Appendix 2.C). To see why this is a serious problem, consider a discrete parameter $N \in \mathbb{Z}^+$ with a substantial posterior uncertainty, say $\mathrm{Var}(N \,|\, \mathrm{data}) \approx 1000^2$. We can then expect that a Metropolis move such as $N \to N + 1000$ would be accepted with a moderate probability, which costs us a single likelihood evaluation. On the other hand, if we were to sample a continuously embedded counter part $\widetilde{N}$ of N using DHMC with the Gaussian momentum based integrator of Algorithm 2.3, a transition of the corresponding magnitude would require *1000 likelihood evaluations*. Such a high computational cost for otherwise simple parameter updates makes the algorithm practically useless.

2.3.2 Hamiltonian dynamics based on Laplace momentum

The above example illustrates that, for discontinuous Hamiltonian dynamics to be of practical value in sampling discrete parameters, we need to be able to devise a reversible integrator that can jump through multiple discontinuities in a small number of target density evaluations while approximately preserving the total energy. It is not at all obvious if such an integrator exists for the dynamics based on a Gaussian momentum.

As we will show below, there is a simple solution to this problem if we instead employ an independent Laplace distribution $\pi_P(\boldsymbol{p}) \propto \prod_i \exp(-m_i^{-1}|p_i|)$. The use of a multivariate Laplace momentum $\pi_P(\boldsymbol{p}) \propto \exp(-\|\boldsymbol{p}\|)$ was previously considered in Zhang et al. (2016). To the best of our knowledge, however, our work is the first of its kind to consider the independent Laplace momentum and propose a practical integrator for the corresponding Hamiltonian dynamics.

Hamilton's equation under the independent Laplace momentum is given by

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \boldsymbol{m}^{-1} \odot \operatorname{sign}(\boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{2.7}$$

where $\odot$ denotes an element-wise multiplication. A unique characteristic of the dynamics (2.7) is that the time derivative of $\boldsymbol{\theta}(t)$ depends only on the sign of p_i's and not on their magnitudes. In particular, if we know that $p_i(t)$'s do not change their signs on the time interval $t \in [\tau, \tau + \epsilon]$, then we also know that

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \epsilon\, \boldsymbol{m}^{-1} \odot \operatorname{sign}\left(\boldsymbol{p}(\tau)\right) \tag{2.8}$$

irrespective of the intermediate values $U(\boldsymbol{\theta}(t))$ along the trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ for $t \in [\tau, \tau + \epsilon]$. Our integrator's ability to jump through multiple discontinuities of $U(\boldsymbol{\theta})$ in single target density evaluation depends critically on this property of the dynamics. While the value of $\boldsymbol{p}(\tau + \epsilon)$ is dependent on the intermediate values $U(\boldsymbol{\theta}(t))$, solving

for this dependence is greatly simplified by splitting the differential operator of (2.7) into its coordinate-wise components as will be shown in the next section.

2.3.3 Integrator for Laplace momentum via operator splitting

Operator splitting is a technique to approximate the solution of a differential equation by decomposing it into components each of which can be solved more easily (McLachlan and Quispel, 2002). Hamiltonian splitting methods more commonly found in the HMC literature are special cases (Neal, 2010). A convenient splitting scheme for (2.7) can be devised by considering the equation for each coordinate (θ_i, p_i) while the other parameters $(\boldsymbol{\theta}_{-i}, \boldsymbol{p}_{-i})$ are fixed:

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = m_i^{-1}\,\mathrm{sign}(p_i), \quad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\partial_{\theta_i} U(\boldsymbol{\theta}), \quad \frac{\mathrm{d}\boldsymbol{\theta}_{-i}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_{-i}}{\mathrm{d}t} = 0 \tag{2.9}$$

There are two possible behaviors for the solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ of (2.9) for $t \in [\tau, \tau + \epsilon]$, depending on the amount of the initial momentum $p_i(\tau)$. Let $\boldsymbol{\theta}^*(t)$ denote a potential path of $\boldsymbol{\theta}(t)$:

$$\theta_i^*(t) = \theta_i(\tau) + (t - \tau) m_i^{-1} \mathrm{sign}(p_i(\tau)), \quad \boldsymbol{\theta}_{-i}^*(t) = \boldsymbol{\theta}_{-i}(\tau) \tag{2.10}$$

In case the initial momentum is large enough that $m_i^{-1}|p_i(\tau)| > U\big(\boldsymbol{\theta}^*(t)\big) - U(\boldsymbol{\theta}(\tau))$ for all $t \in [\tau, \tau + \epsilon]$, we have

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}^*(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \epsilon\, m_i^{-1} \mathrm{sign}(p_i(\tau)) \boldsymbol{e}_i \tag{2.11}$$

where $\boldsymbol{e}_i$ denotes the standard ith basis vector. Otherwise, the momentum p_i flips (i.e. $p_i \leftarrow -p_i$) and the trajectory $\theta_i(t)$ reverses its course at the event time t_e given by

$$t_e = \inf \left\{ t \in [\tau, \tau + \epsilon] : U\big(\boldsymbol{\theta}^*(t)\big) - U(\boldsymbol{\theta}(\tau)) > K(\boldsymbol{p}(\tau)) \right\} \tag{2.12}$$

See Figure 2.1 for a visual illustration of the trajectory $\boldsymbol{\theta}(t)$. By emulating the qualitative behavior of the solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$, we obtain an efficient integrator of

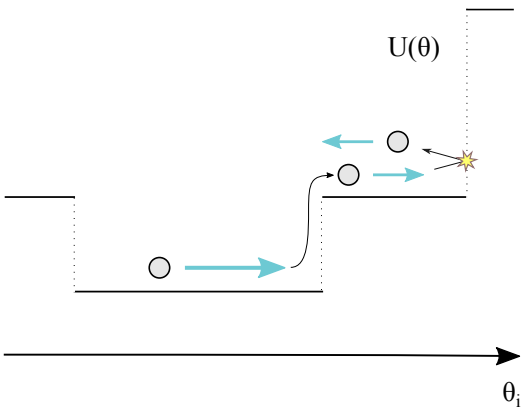


FIGURE 2.1: Example trajectory $\boldsymbol{\theta}(t)$ of discontinuous Hamiltonian dynamics. The trajectory has enough kinetic energy to move across the first discontinuity by transferring some of kinetic energy to potential energy. Across the second discontinuity, however, the trajectory has insufficient kinetic energy to compensate for the potential energy increase and bounces back as a result.

the coordinate-wise equation (2.9) as given in Algorithm 2.1. While the parameter $\boldsymbol{\theta}$ does not get updated when $m_i^{-1}|p_i| < \Delta U$ (line 5), the momentum flip $p_i \leftarrow -p_i$ (line 9) ensures that the next numerical integration step leads the trajectory toward a higher density of $\pi_\Theta(\boldsymbol{\theta})$.

The solution of the original (unsplit) differential equation (2.7) is approximated by sequentially updating each coordinate of $(\boldsymbol{\theta}, \boldsymbol{p})$ with Algorithm 2.1. The reversibility of the resulting proposal is guaranteed by randomly permuting the order of the coordinate updates. (Alternatively, one can split the operator symmetrically to obtain a reversible integrator; for example, first update $\boldsymbol{\theta}$ in the order $\theta_1, \theta_2, \ldots, \theta_d$ and then in the order $\theta_d, \theta_{d-1}, \ldots, \theta_1$. See Section 2.4.1 as well as McLachlan and Quispel (2002) for more details.) The integrator does not reproduce the exact solution but nonetheless preserves the Hamiltonian exactly, yielding a rejection-free proposal. While this remains true with any stepsize ϵ, for good mixing the stepsize needs to be chosen small enough that the condition on Line 5 is satisfied with high probability; see Section 2.4.3 for further discussion on tuning ϵ.

Algorithm 2.1 Coordinate-wise integrator for dynamics with Laplace momentum

function CoordIntegrator($\boldsymbol{\theta}, \boldsymbol{p}, i, \epsilon$)
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
  $\theta_i^* \leftarrow \theta_i^* + \epsilon m_i^{-1} \text{sign}(p_i)$
  $\Delta U \leftarrow U(\boldsymbol{\theta}^*) - U(\boldsymbol{\theta})$
  if $m_i^{-1}|p_i| > \Delta U$ **then**
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$
    $p_i \leftarrow p_i - m_i \Delta U$
  else
    $p_i \leftarrow -p_i$
  end if
  return $\boldsymbol{\theta}$, $\boldsymbol{p}$
end function

2.3.4 Mixing momentum distributions for continuous and discrete parameters

The potential energy $U(\boldsymbol{\theta})$ would be a smooth function of θ_i if both the prior and likelihood depend smoothly on θ_i as is often the case for a continuous parameter. On the other hand, $U(\boldsymbol{\theta})$ will be a discontinuous function of θ_i if θ_i is a continuous embedding of a discrete parameter. The coordinate-wise update of Algorithm 2.1 leads to a valid proposal mechanism whether or not $U(\boldsymbol{\theta})$ has discontinuities along the coordinate θ_i. If $U(\boldsymbol{\theta})$ varies smoothly along some coordinates of $\boldsymbol{\theta}$, however, we can devise an integrator that takes advantage of such smooth dependence.

To describe the integrator, we write $\boldsymbol{\theta} = (\boldsymbol{\theta}_I, \boldsymbol{\theta}_J)$ where the collections of indices I and J are defined as

$$I = \{i \in \{1, \ldots, d\} : U(\boldsymbol{\theta}) \text{ is a smooth function of } \theta_i\}, \quad J = \{1, \ldots, d\} \backslash I \tag{2.13}$$

We write $\boldsymbol{p} = (\boldsymbol{p}_I, \boldsymbol{p}_J)$ correspondingly and define the distribution of $\boldsymbol{p}$ as a product of Gaussian and independent Laplace so that the kinetic energy is given by

$$K(\boldsymbol{p}) = \frac{1}{2} \boldsymbol{p}_I^\intercal \boldsymbol{M}_I^{-1} \boldsymbol{p}_I + \sum_{j \in J} m_j^{-1} |p_j| \tag{2.14}$$

where $\boldsymbol{M}_I$ and $\boldsymbol{M}_J = \text{diag}(m_J)$ are *mass matrices* (Neal, 2010). With a slight abuse

of terminology, we will refer to the parameters with discontinuous conditional densities $\boldsymbol{\theta}_J$ as discontinuous parameters.

The integrator is again based on operator splitting; we update the smooth parameter $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ first, then the discontinuous parameter $(\boldsymbol{\theta}_J, \boldsymbol{p}_J)$, followed by another update of $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$. The discontinuous parameters are updated according to the coordinate-wise operators (2.9) as described in Section 2.3.3. The update of $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ is based on a decomposition familiar from the leapfrog scheme:

$$\frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = \nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta}), \quad \frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}\boldsymbol{\theta}_J}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_J}{\mathrm{d}t} = 0 \tag{2.15}$$

$$\frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = \boldsymbol{M}_I^{-1}\boldsymbol{p}_I, \qquad \frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}\boldsymbol{\theta}_J}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_J}{\mathrm{d}t} = 0 \tag{2.16}$$

The pseudo code of Algorithm 2.2 describes the integrator with all the ingredients put together. The symbol φ denotes a bijective map (i.e. permutation) on J.

Compared to the coordinate-wise updates, the joint update of continuous parameters in this integrator has a couple of advantages. First, the joint update is much more efficient when there is little conditional independence structure that the coordinate-wise updates can take advantage of. Secondly, even with conditional independence structure, the joint update likely has a lower computational cost as an interpreter or compiler (of a programing language) can more easily optimize the required computation. On the other hand, the coordinate-wise updates have an advantage of being rejection-free by virtue of exact energy preservation and may be preferable for posteriors with substantial conditional independence structure such as in latent Markov random field models. Our numerical results in Section 2.5 use the integrator developed in this section.

Algorithm 2.2 Integrator for discontinuous HMC

function DISCINTEGRATOR($\boldsymbol{\theta}, \boldsymbol{p}, \epsilon, \varphi = \text{PERMUTE}(J)$)

$\boldsymbol{p}_I \leftarrow \boldsymbol{p}_I + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta})$

$\boldsymbol{\theta}_I \leftarrow \boldsymbol{\theta}_I + \frac{\epsilon}{2}\boldsymbol{M}_I^{-1}\boldsymbol{p}_I$

for j in J **do**

$\boldsymbol{\theta}, \boldsymbol{p} \leftarrow$ COORDINTEGRATOR$(\boldsymbol{\theta}, \boldsymbol{p}, \varphi(j), \epsilon)$ ▷ update discontinuous params

end for

$\boldsymbol{\theta}_I \leftarrow \boldsymbol{\theta}_I + \frac{\epsilon}{2}\boldsymbol{M}_I^{-1}\boldsymbol{p}_I$

$\boldsymbol{p}_I \leftarrow \boldsymbol{p}_I + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta})$

return $\boldsymbol{\theta}, \boldsymbol{p}$

end function

2.4 Theoretical properties of discontinuous HMC

2.4.1 Reversibility

As in the existing HMC variants, the reversibility of DHMC is a direct consequence of the reversibility and volume-preserving property of our integrator in Algorithm 2.2. We will establish these properties of the integrator momentarily, but first we comment that our integrator has an unconventional feature of being reversible only "in distribution." To explain this, let $\boldsymbol{F}$ denote a bijective map on the space $(\boldsymbol{\theta}, \boldsymbol{p})$ corresponding to the approximation of (discontinuous) Hamiltonian dynamics through multiple numerical integration steps. *Reversibility* of an integrator means that $\boldsymbol{F}$ satisfies

$$(\boldsymbol{R} \circ \boldsymbol{F})^{-1} = \boldsymbol{R} \circ \boldsymbol{F} \quad \text{or equivalently} \quad \boldsymbol{F}^{-1} = \boldsymbol{R} \circ \boldsymbol{F} \circ \boldsymbol{R} \tag{2.17}$$

where $\boldsymbol{R} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}, -\boldsymbol{p})$ is a momentum flip operator. Due to the updates of discrete parameters in a random order, the map $\boldsymbol{F}$ induced by our integrator is non-deterministic and satisfies the reversibility condition (2.17) in distribution: $(\boldsymbol{R} \circ \boldsymbol{F})^{-1} \overset{d}{=} \boldsymbol{R} \circ \boldsymbol{F}$.

Once we establish the reversibility (in distribution) and volume-preserving property

of the integrator, the reversibility of DHMC proposals follows from these properties of our integrator by the standard arguments with only a minor modification. For these arguments, we refer the readers to Neal (2010) and Betancourt (2017) for a heuristic presentation and to Fang et al. (2014) for a more general and mathematically precise treatment.

Analysis of our integrator'ic reversibility and volume-preserving property is similar to that of typical HMC integrators except for the coordinate-wise integration part of Algorithm 2.1. These properties for the coordinate-wise integrator are first established in Lemma 2.1, based on which the same properties for Algorithm 2.2 are established in Theorem 2.2.

Lemma 2.1. *For a piecewise smooth potential energy $U(\boldsymbol{\theta})$, the coordinate-wise integrator of Algorithm 2.1 is volume-preserving and almost everywhere reversible for any coordinate index i. Moreover, updating multiple coordinates with the integrator in a random index order $\varphi(1), \ldots, \varphi(d)$ is again reversible (in distribution) provided that the random permutation φ satisfies $(\varphi(1), \varphi(2), \ldots, \varphi(d)) \overset{d}{=} (\varphi(d), \varphi(d-1), \ldots, \varphi(1))$.*

Theorem 2.2. *For a piecewise smooth potential energy $U(\boldsymbol{\theta})$, the integrator of Algorithm 2.2 is volume-preserving and almost everywhere reversible.*

The proofs are in Appendix 2.A.

We also establish in Theorem 2.3 a more general result on the reversibility and volume-preserving property of discontinuious Hamiltonian dynamics. The result in particular justifies the use of the alternative event-driven integrator Algorithm 2.3 in the appendix and may be useful in constructing other event-driven integrators with various momentum distributions. A *solution operator* $\boldsymbol{\Psi}_t$ of a differential equation (or more generally of a differential inclusion) is a map such that $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ is a solution of the equation with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. Also,

symplecticity is a property of Hamiltonian dynamics which in particular implies volume-preservation; see Appendix 2.B for the definition.

Theorem 2.3. *Let $U(\boldsymbol{\theta})$ be a piecewise constant potential energy function whose discontinuity set is piecewise linear. Suppose that a kinetic energy $K(\boldsymbol{p})$ is symmetric, convex, piecewise smooth, and satisfies the growth condition $K(\boldsymbol{p}) \to \infty$ as $\|\boldsymbol{p}\| \to \infty$. Then the solution operator $\boldsymbol{\Psi}_t$ of discontinuous Hamiltonian dynamics as defined in Section 2.2.3 is symplectic and almost everywhere reversible.*

The proof is in Appendix 2.B. Theorem 2.3 generalizes the result of Afshar and Domke (2015) in three ways: it holds for any kinetic energy satisfying the assumption, implies a stronger conclusion of symplecticity, and provides a rigorous quantification of non-differentiable sets. We believe that the conclusion of Theorem 2.3 holds under a much more general potential and kinetic energy since the reversibility and symplecticity are essential properties of smooth Hamiltonian dynamics. It is well beyond the scope of this thesis to analyze the properties of more general non-smooth Hamiltonian dynamics, however, as the study of such dynamics requires more sophisticated mathematical tools; see Fetecau (2003) and Brogliato (2016) for more on this topic.

2.4.2 Ergodicity

Care needs to be taken when applying the coordinate-wise integrator as its use with a fixed stepsize ϵ results in a reducible Markov chain which is not ergodic; we discuss this issue here and present a simple remedy.

Consider the transition probability of (multiple iterations of) DHMC based on the integrator of Algorithm 2.2. Given the initial state $\boldsymbol{\theta}_0$, the integrator of Algorithm 2.1 moves the i-th coordinate of $\boldsymbol{\theta}$ only by the distance $\pm\epsilon m_i^{-1}$ regardless of the values of the momentum variable. The transition probability in the $\boldsymbol{\theta}$-space with $\boldsymbol{p}$ marginalized

out, therefore, is supported on a grid

$$\Omega = \{(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) : \boldsymbol{\theta}_J = \boldsymbol{\theta}_{0,J} + \epsilon \boldsymbol{m} \odot \boldsymbol{k} \text{ for a vector of integers } \boldsymbol{k}\} \tag{2.18}$$

where $\boldsymbol{\theta}_J$ as in (2.13) denotes the coordinates of $\boldsymbol{\theta}$ with discontinuous conditionals.

The pathological behavior above can be avoided simply by randomizing the stepsize at each iteration, say $\epsilon \sim \text{Uniform}(\epsilon_{\min}, \epsilon_{\max})$. In fact, randomizing the stepsize over a small interval is considered a good practice in any case to account for the possibility that some regions of the parameter space require smaller stepsizes for efficient exploration (Neal, 2010). While the coordinate-wise integrator does not suffer from the stability issue of the leapfrog scheme, the quantity ϵm_i^{-1} nonetheless needs to be in the same order of magnitude as the length scale of θ_i; see Section 2.4.3.

2.4.3 Role of mass matrix and stepsize

As in the case of standard HMC, using a non-identity mass matrix has the effect of preconditioning a target distribution through reparametrization (Neal, 2010; Nishimura and Dunson, 2016). More precisely, for a matrix $\boldsymbol{A}_I$ and a diagonal matrix $\boldsymbol{A}_J$, the performance of DHMC in a sampling space $(\boldsymbol{A}_I\boldsymbol{\theta}_I, \boldsymbol{A}_J\boldsymbol{\theta}_J, \boldsymbol{p}_I, \boldsymbol{p}_J)$ is identical to that in $(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J, \boldsymbol{A}_I^\intercal\boldsymbol{p}_I, \boldsymbol{A}_J^\intercal\boldsymbol{p}_J)$. Since the choice of a mass matrix for a Gaussian momentum is a well-studied topic (Neal, 2010; Girolami and Calderhead, 2011), we focus on the choice of mass m_j for a Laplace momentum $p_j \sim \text{Laplace}(\text{scale} = m_j)$.

We generally expect that sampling is facilitated by a reparametrization $\boldsymbol{\theta}_J \to \boldsymbol{\Lambda}_J^{-1/2}\boldsymbol{\theta}_J$ for $\boldsymbol{\Lambda}_J = \text{diag}(\text{var}(\boldsymbol{\theta}_J))$. This suggests that, together with the property of DHMC under parameter transformation discussed above, the mass should be chosen as $m_j \approx \text{var}(\theta_j)^{-1/2}$. This choice is further motivated by the fact that a parameter θ_j is updated by an increment of $\pm\epsilon m_j^{-1}$ by the coordinate-wise integrator of Algorithm 2.1, and is related to the issue of choosing stepsize ϵ we discuss now.

The stepsize ϵ should be adjusted so that ϵm_j^{-1} has the same order of magnitude

as a typical scale of the conditional distribution of θ_j. Unlike a leapfrog integrator that becomes unstable as ϵ increases, the coordinate-wise integrator remains exactly energy-preserving but at some point a large stepsize will cause DHMC to "get stuck" at the current state. The numerical integration scheme of DHMC will keep flipping the momentum $p_j \leftarrow -p_j$ (Line 9 of Algorithm 2.1) without updating θ_j until the following condition is met:

$$U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big) - U(\boldsymbol{\theta}) < m_j^{-1}|p_j| \overset{d}{=} \text{Exp}(1) \tag{2.19}$$

where $\boldsymbol{e}_j$ denotes the j-th standard basis vector. When ϵm_j^{-1} becomes larger than a typical scale of θ_j, it becomes unlikely for the condition (2.19) to be satisfied and leads to infrequent updates of θ_j.

Based on these observations, useful statistics for tuning the stepsize would be

$$\begin{aligned}&\mathbb{P}_{\pi_\Theta \times \pi_P}\left\{U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big) - U(\boldsymbol{\theta}) > m_j^{-1}|p_j|\right\} \\ &\qquad = \mathbb{E}_{\pi_\Theta \times \pi_P}\left[\min\left\{1,\, \exp\left(U(\boldsymbol{\theta}) - U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big)\right)\right\}\right]\end{aligned} \tag{2.20}$$

which plays a role analogous to the acceptance rate of Metropolis proposals. The statistics (2.20) can be estimated, for example, by counting the frequency of momentum flips during each DHMC iteration, and can then be used to tune the stepsize through stochastic optimization (Andrieu and Thoms, 2008; Hoffman and Gelman, 2014). One would want the statistics to be well above zero but not too close to 1, balancing the mixing rate and computational cost of each DHMC iteration. Theoretical analysis of the optimal statistics value is beyond the scope of this thesis, but the value $0.7 \sim 0.9$ is perhaps reasonable in analogy with the optimal acceptance rate of HMC (Betancourt et al., 2014).

2.4.4 Metropolis-within-Gibbs with momentum

Consider a version of DHMC in which all the parameters are updated with the coordinate-wise integrator of Algorithm 2.1; in other words, the integrator of Algorithm 2.2 is applied with $J = \{1, \ldots, d\}$ and an empty indexing set I. This version of DHMC turns out to be a generalization of random-scan Metropolis-within-Gibbs (also known as one-variable-at-a-time Metropolis) algorithm. We therefore refer to this version of DHMC alternatively as *Metropolis-within-Gibbs with momentum*.

To make the connection to Metropolis-within-Gibbs clear, we first write out this version of DHMC explicitly. We use $\pi_{\mathcal{E}}(\cdot)$ and $\pi_{\Phi}(\cdot)$ to denote the distribution of a stepsize ϵ and a permutation φ of $\{1, \ldots, d\}$. As before, we require that $\varphi \sim \pi_{\Phi}(\cdot)$ satisfies $(\varphi(1), \ldots, \varphi(d)) \overset{d}{=} (\varphi(d), \ldots, \varphi(1))$. With these notations, one iteration of Metropolis-within-Gibbs with momentum can be expressed as follows:

1. Draw $\epsilon \sim \pi_{\mathcal{E}}(\cdot)$, $\varphi \sim \pi_{\Phi}(\cdot)$, and $p_j \sim \text{Laplace}(\text{scale} = m_j)$ for $j = 1, \ldots, d$.
2. Repeat for L times a sequential update of the coordinate (θ_j, p_j) for $j = \varphi(1), \ldots, \varphi(d)$ via Algorithm 2.1 with stepsize ϵ.

In this version of DHMC, the integrator exactly preserves the Hamiltonian and the acceptance-rejection step can be omitted.

When $L = 1$, the above algorithm recovers Metropolis-within-Gibbs with a random scan order $\varphi \sim \pi_{\Phi}(\cdot)$. This can be seen by realizing that Lines 5 – 9 of Algorithm 2.1 coincide with the standard Metropolis acceptance-rejection procedure for θ_j. More precisely, the coordinate-wise integrator updates θ_j to $\theta_j + \epsilon m_j^{-1}\text{sign}(p_j)$ only if

$$\exp\left(-U(\boldsymbol{\theta}^*) + U(\boldsymbol{\theta})\right) > \exp\left(-m_j^{-1}|p_j|\right) \overset{d}{=} \text{Uniform}(0, 1) \tag{2.21}$$

where the last distributional equality follows from the fact $m_j^{-1}|p_j| \overset{d}{=} \text{Exp}(1)$. Theorem 2.4 below summarizes the above discussion.

Theorem 2.4. *Consider a version of DHMC updating all the parameters with the coordinate-wise integrator of Algorithm 2.1. When taking only one numerical integration step, this version of DHMC coincides with Metropolis-within-Gibbs with a random scan order $\varphi \sim \pi_\Phi(\cdot)$ and a symmetric proposal $\theta_j \pm \epsilon m_j^{-1}$ for each parameter with $\epsilon \sim \pi_\mathcal{E}(\cdot)$.*

As formulated in Theorem 2.4, the version of DHMC corresponds to a Metropolis-within-Gibbs with the univariate proposal distributions coupled via the shared parameter $\epsilon \sim \pi_\mathcal{E}(\cdot)$. We could also consider a version of DHMC with a fixed stepsize $\epsilon = 1$ but with a mass matrix randomized $(m_1^{-1}, \ldots, m_d^{-1}) \sim \pi_{\boldsymbol{M}^{-1}}(\cdot)$ before each numerical integration step. This version would correspond to a more standard Metropolis-within-Gibbs with independent univariate proposals.

Being a generalization of Metropolis-within-Gibbs, DHMC is guaranteed a superior performance if tuned properly:

Corollary 2.5. *Under any efficiency metric (which may account for computational costs), an optimally tuned DHMC outperforms random-scan Metropolis-within-Gibbs samplers.*

In practice, the addition of momentum to Metropolis-within-Gibbs allows for a more efficient update of correlated parameters as empirically shown in Appendix 2.E.1.

2.4.5 Relation to zig-zag sampler

Zig-zag sampler is a state-of-the-art rejection-free non-reversible Monte Carlo algorithm based on a piece-wise deterministic Markov process called a *zig-zag process* (Bierkens et al., 2016; Fearnhead et al., 2016; Bierkens et al., 2017). As we will show now, the Laplace momentum based Hamiltonian dynamics (2.7) with unit masses (i.e. $m_j = 1$ for all j) has a remarkable similarity to a zig-zag process.

Consider a zig-zag process and Hamiltonian dynamics both starting from the state $\boldsymbol{\theta}_0$. Let $\boldsymbol{v}_0$ drawn uniformly drawn from $\{-1, +1\}^d$ be the initial velocity of the zig-zag process and $\boldsymbol{p}_0 = (p_{0,1}, \ldots, p_{0,d})$ drawn from the independent Laplace distribution be the initial momentum of the Hamiltonian dynamics. Under both the zig-zag process and Hamiltonian dynamics, the velocity and momentum remain constant while the parameter $\boldsymbol{\theta}$ moves along a straight line $\boldsymbol{\theta}^Z(t) = \boldsymbol{\theta}_0 + t\boldsymbol{v}_0$ and $\boldsymbol{\theta}^H(t) = \boldsymbol{\theta}_0 + t\,\mathrm{sign}(\boldsymbol{p}_0)$ for $t > 0$ until their respective first event times. The first event time for the zig-zag process is given as $t_e^Z = \min\{t_1^Z, \ldots, t_d^Z\}$ where

$$t_i^Z = \inf_{t'>0} \left\{ \tau_i = \int_0^{t'} \left[v_{0,i} \partial_{\theta_i} U(\boldsymbol{\theta}_0 + t\boldsymbol{v}_0) \right]^+ \mathrm{d}t' \right\} \tag{2.22}$$

with $[x]^+ = \max\{0, x\}$ and τ_i's drawn from Exp(1). For the Hamiltonian dynamics, the first event time is given as $t_e^H = \min\{t_1^H, \ldots, t_d^H\}$ where

$$t_i^H = \inf_{t'>0} \left\{ |p_{0,i}| = \int_0^{t'} \mathrm{sign}(p_{0,i})\, \partial_{\theta_i} U(\boldsymbol{\theta}_0 + t\,\mathrm{sign}(\boldsymbol{p}_0)) \mathrm{d}t' \right\} \tag{2.23}$$

For both processes, the events result in the velocity change $v_k \leftarrow -v_k$ and $\mathrm{sign}(p_\ell) \leftarrow -\mathrm{sign}(p_\ell)$ for $k = \mathrm{argmin}_i\{t_i^Z\}$ and $\ell = \mathrm{argmin}_i\{t_i^H\}$.

Given that $(\boldsymbol{v}_0, \boldsymbol{\tau}) \overset{d}{=} (\mathrm{sign}(\boldsymbol{p}_0), |\boldsymbol{p}_0|)$, the similarity between (2.22) and (2.23) is striking. In fact, if $U(\boldsymbol{\theta})$ were convex and $\boldsymbol{\theta}_0$ was the minimum of $U(\boldsymbol{\theta})$, then the two processes $\{\boldsymbol{\theta}^Z(t), 0 \leqslant t \leqslant t_e^Z\}$ and $\{\boldsymbol{\theta}^H(t), 0 \leqslant t \leqslant t_e^H\}$ coincide in distribution. After the first event time or in more general settings, however, the two processes diverge because a zig-zag process $(\boldsymbol{\theta}^Z, \mathrm{d}\boldsymbol{\theta}^Z/\mathrm{d}t) = (\boldsymbol{\theta}^Z, \boldsymbol{v})$ is Markovian while its Hamiltonian dynamics counter-part $(\boldsymbol{\theta}^H, \mathrm{d}\boldsymbol{\theta}^H/\mathrm{d}t) = (\boldsymbol{\theta}^H, \mathrm{sign}(\boldsymbol{p}))$ is not. More specifically, Hamiltonian dynamics after each event retains the magnitudes of its momentum variable $|p_i|$'s from the previous moment. Also, Hamiltonian dynamics accumulates kinetic energy while going potential energy downhill such that $\mathrm{sign}\left(p_i(t)\right) \partial_{\theta_i} U(\boldsymbol{\theta}^H(t)) < 0$.

This creates a tendency for each coordinate of a Hamiltonian dynamics trajectory $\boldsymbol{\theta}^H(t)$ to travel longer in the same direction before switching its direction compared to that of a zig-zag process.

Its close connection to a state-of-the-art sampler partially explains the empirical success of DHMC in Section 2.E.1, thought the application of DHMC to smooth target distributions is outside the main focus of this chapter. Some advantages of zig-zag sampler over others have been considered to be its non-reversibility and the fact that its entire trajectory can be used as valid samples from the target. In fact, DHMC can also be made non-reversible through partial momentum refreshments (Neal, 2010) and can utilize the entire trajectories as valid samples (Nishimura and Dunson, 2015). These strategies will likely further boost the performance of DHMC. We will leave further theoretical and empirical comparisons of DHMC and zig-zag sampler to a future work.

2.5 Numerical results

We use two challenging posterior inference problems to demonstrate that DHMC is a highly efficient and general-purpose sampler that extends the scope of applicable models well beyond traditional HMC. Additional numerical results in Appendix 2.E further illustrate the breadth of DHMC's capability.

Currently, few general and efficient approaches exist for sampling from a discrete parameter or a discontinuous target density when the posterior is not conjugate. While adaptive Metropolis algorithms (Haario et al., 2001, 2006) based on Gaussian proposal distributions can be effective on a low-dimensional target distribution with limited non-linearity, such algorithms scale poorly in the number of parameters (Roberts et al., 1997). Metropolis-within-Gibbs with component-wise adaptation can scale better provided the conditional densities can be efficiently evaluated, but suffers from strong dependence among the parameters (Haario et al., 2005). We

will use these algorithms as benchmarks when they are viable options. As another benchmark against discontinuous HMC, we use a best current practice implemented in a probabilistic programming language PyMC (Salvatier et al., 2016). Other HMC-based probabilistic programming languages do not support inference on discrete parameters to our knowledge.

PyMC samples continuous and discrete parameters alternately, sampling continuous ones with NUTS, a variant of HMC with automatic path length adjustments (Hoffman and Gelman, 2014), and discrete ones with univariate Metropolis updates. We will refer to this approach as a NUTS-Gibbs sampler. We tilt the comparison in favor of NUTS-Gibbs by updating each discrete parameter with a full conditional update through multinomial sampling from all possible values, truncated to a reasonable range if the sampling space is infinite. With our high-level language (Python) implementation, calculations required for such multinomial sampling require a small amount of computer time relative to continuous parameter updates, with these calculations more easily optimized through vectorization.

For each example, the stepsize and path length (i.e. the number of numerical integration steps) for DHMC were manually adjusted over short preliminary runs, visually examining the trace plots of the worst mixing parameters. We then used the same stepsize for sampling the continuous parameters in the NUTS-Gibbs sampler. In both the samplers, continuous parameters with range constraints are transformed into unconstrained ones to facilitate sampling (Stan Development Team, 2016; Salvatier et al., 2016). More precisely, the constraint $\theta > 0$ is handled by a log transform $\theta \to \log \theta$ and $\theta \in [0, 1]$ by a log odds transform $\theta \to \log(\theta/(1-\theta))$ as done in Stan and PyMC. DHMC handles constraints on discrete parameters automatically through the coordinate-wise integrator of Algorithm 2.1.

Efficiencies of the algorithms are compared through effective sample sizes (ESS) (Geyer, 2011). As is commonly done in the MCMC literature, we compute the effective

sample sizes of the first and second moment estimators for each parameter and report the minimum ESS across all the parameters. ESS's are estimated using the method of batch means with 25 batches (Geyer, 2011), averaged over the estimates from 8 independent chains.[2] We report a confidence interval for the ESS estimator of the form $\pm 1.96\,\hat{\sigma}$ with the empirical variance $\hat{\sigma}^2$.

2.5.1 Jolly-Seber model: estimation of unknown open population size and survival rate from multiple capture-recapture data

The Jolly-Seber model and its numerous extensions are widely used in ecology to estimate unknown animal population sizes as well as related parameters of interest (Schwarz and Seber, 1999). The method is based on the following experimental design. Animals of a particular species having an unknown population size are captured, marked (for example by tagging), and released bask to the environment. This procedure is repeated over multiple capture occasions. At each occasion, the number of marked and unmarked animals among the captured ones are recorded. Individuals survive from one capture occasion to another with an unknown survival rate. Also, the population is assumed to be "open" so that individuals may enter (either through birth or immigration) or leave the area under the study.

In order to be consistent with the literature on capture-recapture models, the notations within this section will deviate from the rest of the chapter. Assuming that data are collected over $i = 1, \ldots, T$ capture occasions, the unknown parameters of the model are $\{U_i, p_i\}_{i=1}^{T}$ and $\{\phi_i\}_{i=1}^{T-1}$, each of which represents

[2] While the consistency of a batch means estimator requires the number of batches to grow with the length of a chain, some of our benchmark samplers mix so slowly that it seemed practically impossible to run them long enough for such an estimator to be unbiased (Flegal and Jones, 2010). An estimator with a fixed number of batches is at least unbiased for a long enough chain, so we chose to use a fixed number of batches and assess its accuracy by running multiple independent chains. For reversible chains, our batch means estimates were similar to those of the monotone positive sequence estimator of Geyer (1992). The R package CODA (Plummer et al., 2006) provides an alternative estimator but it appeared to overestimate ESS's for very slowly mixing chains while the batch means and monotone sequence estimators agreed.

- $U_i =$ number of unmarked animals right before the ith capture occasion.
- $p_i =$ capture probability of each animal at the ith capture occasion.
- $\phi_i =$ survival probability of each animal from the ith to $(i+1)$th capture occasion.

We assign standard objective priors $p_i, \phi_i \sim \text{Unif}(0,1)$ and $\pi(U_1) \propto U_1^{-1}$. The prior conditional distributions $U_{i+1} \,|\, U_i, \phi_i$ are described in the supplement Section 2.D, along with the expression for the likelihood function and other details on the Jolly-Seber model.

We take the black-kneed capsid population data from Jolly (1965) as summarized in Seber (1982). The data records the capture-recapture information over $T = 13$ successive capture occasions. The posterior distribution of U_i may be heavy-tailed with conditional variance that depends highly on the capture probability p_i, so we alleviate these potential issues through log-transformed embedding of U_i's (Section 2.2.1). NUTS-Gibbs sampler updates U_i's through multinomial samplings from the integers between 0 and 5,000. Also, as this example happens to be low-dimensional enough, we try a random walk Metropolis with optimal Gaussian proposals by pre-computing the true posterior covariance with a long adaptive Metropolis chain (Roberts et al., 1997; Haario et al., 2001). DHMC can also take advantage of the posterior covariance information through the mass matrix, so we also try DHMC with a diagonal mass matrix whose entries are set according to the estimated posterior variance of each parameter. Starting from stationarity, we run 10^4 iterations of DHMC and NUTS-Gibbs and 5×10^5 iterations of Metropolis.

The performance of each algorithm is summarized in Table 2.1 where "DHMC (diagonal)" and "DHMC (identity)" indicate DHMC with a diagonal and identity mass matrix respectively. The table clearly indicates a superior performance of DHMC over NUTS-Gibbs and Metropolis with approximately 60 and 7-fold efficiency

Table 2.1: Performance summary of each algorithm on the Jolly-Serber model example. The term ($\pm \ldots$) is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC (diagonal)	45.5 ($\pm$ 5.2)	424	45	87.7
DHMC (identity)	24.1 ($\pm$ 2.6)	126	77.5	157
NUTS-Gibbs	1.04 ($\pm$ 0.087)	6.38	150	133
Metropolis	0.0714 ($\pm$ 0.016)	58.5	1	1

increase respectively when using a diagonal mass matrix. The posterior distribution exhibits high negative correlations between U_i and p_i, and all the algorithms recorded their worst ESS in the first capture probability p_1; see Figure 2.2. Some correlations are observed among the other pairs of parameters, but not to the same degree.

Our results demonstrate that, although the continuous and discontinuous parameters are not updated simultaneously in the DHMC integrator of Algorithm 2.2, the trajectory can nonetheless explore the joint distributions of highly dependent variables through the momentum information. It is also worth noting that the advantage of DHMC over Metropolis will likely increase as the parameter space dimension grows because of their scaling properties (Roberts et al., 1997; Beskos et al., 2013); see also Section 2.5.2 for the comparison of the two algorithms in a higher dimensional problem.

2.5.2 Generalized Bayesian belief update based on loss functions

Motivated by model misspecification and difficulty in modeling all aspects of a data generating process, Bissiri et al. (2016) propose a generalized Bayesian framework, which replaces the log-likelihood with a surrogate based on a utility function. Given an additive loss $\ell(\boldsymbol{y}, \boldsymbol{\theta})$ for the data $\boldsymbol{y}$ and parameter of interest $\boldsymbol{\theta}$, the prior $\pi(\boldsymbol{\theta})$ is updated to obtain the generalized posterior:

$$\pi_{\text{post}}(\boldsymbol{\theta}) \propto \exp\left(-\ell(\boldsymbol{y}, \boldsymbol{\theta})\right) \pi(\boldsymbol{\theta}) \tag{2.24}$$

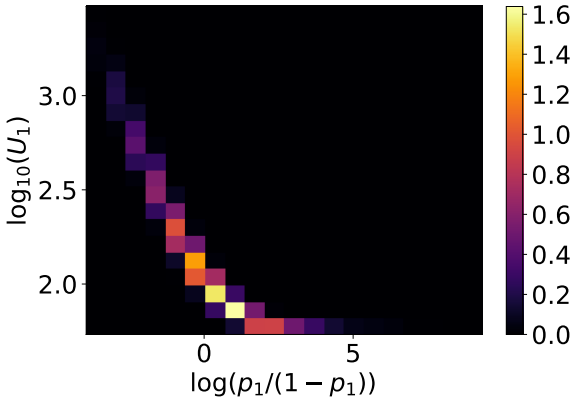


FIGURE 2.2: Two-dimensional empirical density function showing the posterior marginal of (p_1, U_1) with parameter transformations.

While (2.24) coincides with a pseudo-likelihood type approach, Bissiri et al. (2016) derives the formula as a coherent and optimal update from a decision theoretic perspective.

Here we consider a binary classification problem with an error-rate loss:

$$\ell(\boldsymbol{y}, \boldsymbol{\beta}) = \sum_{i=1} \mathbb{1}\left\{y_i \boldsymbol{x}_i^\intercal \boldsymbol{\beta} < 0\right\} \tag{2.25}$$

where $y_i \in \{-1, 1\}$, $\boldsymbol{x}_i$ is a vector of predictors, and $\boldsymbol{\beta}$ is a regression coefficient. The target distribution of the form (2.24) based on the loss function (2.25) is suggested as a challenging test case for an MCMC algorithm by Chopin and Ridgway (2017). We use the SECOM data from UCI machine learning repository, which records various sensor data that can be used to predict the production quality (pass or fail) of a semi-conductor. We first removed the predictors with more than 20 missing cases and then removed the observations that still had missing predictors, leaving us 1,477 cases with 376 predictors. All the predictors were then normalized and the regression coefficients β_i's were given $\mathcal{N}(0, 1)$ priors.

Chopin and Ridgway (2017) present a wide range of algorithms to approximate or sample from a posterior distribution of binary classification problems, but none of

them except random-walk Metropolis seems to apply to the target distribution of interest here — Figure 2.3 shows the conditional density of the intercept parameter as an illustration. Their numerical results also indicated that, after accounting for computational cost, none of the more complex algorithms consistently outperform Metropolis even for a parameter space of dimension as large as 180. We therefore compare DHMC to Metropolis with a proposal covariance matrix proportional to the empirical posterior covariance estimated by 10^5 iterations of DHMC. The scaling of the proposal covariance as suggested by Haario et al. (2001) resulted in an acceptance probability of less than 4%, so we scaled the proposal covariance to achieve the acceptance probability of 0.234 with stochastic optimization (Andrieu and Thoms, 2008). We also run Metropolis-within-Gibbs that updates one parameter at a time with the acceptance probability calibrated to be around 0.44 as recommended in Gelman et al. (1996). We run DHMC for 10^4 iterations, Metropolis for 10^7 iterations, and Metropolis-within-Gibbs for 5×10^4 iterations from stationarity.

Table 2.2 summarizes the performance of each algorithm. DHMC outperforms Metropolis and Metropolis-within-Gibbs approximately by a factor of 330 and 2 respectively. The mixing of random-walk Metropolis suffers substantially from the dimensionality of the target (377 parameters).[3] Conditional updates of Metropolis-within-Gibbs mix relatively well as the posterior correlation happens to be quite modest — the condition number of the estimated posterior covariance matrix was approximately 6.8^2.

2.6 Discussion

We have presented discontinuous HMC, an extension of HMC that can sample from discontinuous target densities while inheriting generality and efficiency of HMC.

[3] We could not store all the 10^7 samples in memory, so we kept only every 100 iterations for computing the ESS's. The discarded samples contribute little to the ESS's due to extremely high auto-correlations.

Table 2.2: Performance summary of each algorithm on the generalized Bayesian posterior example. The term $(\pm \ldots)$ is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC	26.3 ($\pm$ 3.2)	76	25	972
Metropolis	0.00809 ($\pm$ 0.0018)	0.227	1	1
Metropolis-Gibbs	0.514 ($\pm$ 0.039)	39.8	1	36.2

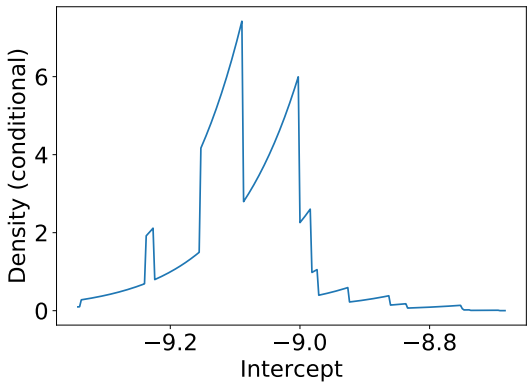


FIGURE 2.3: The posterior conditional density of the intercept parameter in the generalized Bayesian posterior example. The other parameters are fixed at the posterior draw with the highest posterior density among the DHMC samples. The density is not continuous since the loss function is not.

DHMC easily accommodates discrete parameters through the embedding strategy presented in Section 2.2.1. When using a Laplace momentum for each parameter, DHMC turns out to be a generalization of Metropolis-within-Gibbs that takes advantage of momentum information (Section 2.4.4). A preliminary result in Appendix 2.E.1 indicates that this version of DHMC may be useful on its own — regardless of discontinuity in a target density.

We are currently developing methods for automatic tuning of DHMC based on its properties described in Section 2.4 as well as various techniques that have been developed for tuning HMC (Hoffman and Gelman, 2014; Wang et al., 2013; Stan Development Team, 2016). We hope to integrate DHMC into an existing probabilistic programming language so that it can be deployed more widely and have its capability

and limitations fully tested. Since DHMC requires calculations of log-differences in full conditional distributions, it is critical for its optimized implementation that conditional independence structures are exploited and redundant calculations avoided. Such operations are already automated, for example, by PyMC using the Theano library as a back-end (Theano Development Team, 2016).

In addition to being a highly practical algorithm, DHMC also motivates new directions in methodological and theoretical research on Monte Carlo methods. In the HMC literature to this date, there have been limited interests in non-Gaussian momenta as well as limited research effort to develop a new class of integrators aside from incremental improvements in numerical accuracy (Betancourt, 2017). Our work demonstrates a successful use of a non-Gaussian momentum along with a custom-made integrator in practical applications. The ideas introduced here could conceivably be extended to devise HMC-like algorithms in more complex parameter spaces as considered in Dinh et al. (2017). Also, the remarkable similarity between a zig-zag process and Hamiltonian dynamics underlying DHMC (Section 2.4.5) may indicate a deeper connection between piecewise deterministic Markov processes and Hamiltonian dynamics based samplers. A unifying framework for the two approaches, if it exists, could provide recipes for further innovations in Monte Carlo algorithms.

Appendix for Chapter 2

2.A Proof of Lemma 2.1 and Theorem 2.2

Proof of Lemma 2.1. One step of Algorithm 2.1 corresponds to a map $\boldsymbol{F}_{i,\epsilon} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ as follows assuming $p_i \neq 0$. Let $\boldsymbol{e}_i$ denote the ith standard basis vector. We can express $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ in terms of $(\boldsymbol{\theta}, \boldsymbol{p})$ as

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i) \boldsymbol{e}_i, \quad \boldsymbol{p}^* = \boldsymbol{p} - m_i \left\{ U(\boldsymbol{\theta}^*) - U(\boldsymbol{\theta}) \right\} \boldsymbol{e}_i \tag{2.26}$$

if $U(\boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i) \boldsymbol{e}_i) - U(\boldsymbol{\theta}) > m_i^{-1} p_i$, and otherwise

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}, \quad \boldsymbol{p}^* = -\boldsymbol{p} \tag{2.27}$$

Under both (2.26) and (2.27), we have $\partial \boldsymbol{\theta}^* / \partial \boldsymbol{p} = 0$ and it can be easily shown that

$$\det \left(\frac{\partial (\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial (\boldsymbol{\theta}, \boldsymbol{p})} \right) = \det \left(\frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{\theta}} \right) \det \left(\frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}} \right) = 1 \tag{2.28}$$

establishing the volume-preservation. The reversibility as defined in (2.17) can be directly verified by solving the update equations (2.26) and (2.27) for $(\boldsymbol{\theta}, -\boldsymbol{p})$ as a function of $(\boldsymbol{\theta}^*, -\boldsymbol{p}^*)$.

To see that the above argument for reversibility and volume-preservation holds almost everywhere, let $\mathcal{D}$ denote the discontinuity set of $U(\boldsymbol{\theta})$ and $\mathcal{D} + \boldsymbol{v}$ denote a set of points in $\mathcal{D}$ shifted by a vector $\boldsymbol{v}$. Observe that the update equations (2.26) and (2.27) are well-defined and differentiable except when $(\boldsymbol{\theta}, \boldsymbol{p})$ belongs to one of

the sets below:

$$\mathcal{D} \times \mathbb{R}^d,\ \left(\mathcal{D} \pm \epsilon m_i^{-1} \boldsymbol{e}_i\right) \times \mathbb{R}^d,\ \{p_i = 0\},\ \left\{U(\boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i)\boldsymbol{e}_i) - U(\boldsymbol{\theta}) = m_i^{-1} p_i\right\} \tag{2.29}$$

Each of the sets above consists of lower-dimensional manifolds of the parameter space and hence has measure zero.

Lastly, we prove the reversibility of multiple coordinate updates corresponding to a map $\boldsymbol{F}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}$ with a random permutation φ. From the reversibility of each $\boldsymbol{F}_{\epsilon,i}$, we deduce that

$$\boldsymbol{R} \circ \left(\boldsymbol{F}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}\right) \circ \boldsymbol{R} = \boldsymbol{F}_{\varphi(d),\epsilon}^{-1} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}^{-1} = \left(\boldsymbol{F}_{\varphi(1),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(d),\epsilon}\right)^{-1} \tag{2.30}$$

By our assumption on the distribution of φ, we have

$$\left(\boldsymbol{F}_{\varphi(1),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(d),\epsilon}\right)^{-1} \overset{d}{=} \left(\boldsymbol{F}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}\right)^{-1} \tag{2.31}$$

establishing the reversibility of $\boldsymbol{F}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}$ in distribution. □

Proof of Theorem 2.2. Let $\boldsymbol{F}_{J,\varphi,\epsilon} = \boldsymbol{F}_{\varphi(d'),\epsilon} \circ \ldots \circ \boldsymbol{F}_{\varphi(1),\epsilon}$ where $\boldsymbol{F}_{j,\epsilon} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is defined as in (2.26) and (2.27) and $\varphi(1), \ldots, \varphi(d')$ is a permutation of the indexing set J. Also define $\boldsymbol{F}_{\Theta,I,\epsilon/2}$ and $\boldsymbol{F}_{P,I,\epsilon/2}$ as a function of $(\boldsymbol{\theta}, \boldsymbol{p})$ such that

$$\boldsymbol{F}_{\Theta,I,\epsilon/2} : \boldsymbol{\theta}_I \to \boldsymbol{\theta}_I + \frac{\epsilon}{2} \boldsymbol{M}_I^{-1} \boldsymbol{p}_I, \quad \boldsymbol{F}_{P,I,\epsilon/2} : \boldsymbol{p}_I \to \boldsymbol{p}_I - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}) \tag{2.32}$$

while leaving all the other coordinate unchanged. The integrator of Algorithm 2.2 can then be formally expressed as a map

$$\boldsymbol{F}_{\Theta,I,\epsilon/2} \circ \boldsymbol{F}_{P,I,\epsilon/2} \circ \boldsymbol{F}_{J,\varphi,\epsilon} \circ \boldsymbol{F}_{P,I,\epsilon/2} \circ \boldsymbol{F}_{\Theta,I,\epsilon/2} \tag{2.33}$$

Being a symmetric composition of reversible maps, the map (2.33) is again reversible. The maps $\boldsymbol{F}_{\Theta,I,\epsilon/2} \circ \boldsymbol{F}_{P,I,\epsilon/2}$ and $\boldsymbol{F}_{P,I,\epsilon/2} \circ \boldsymbol{F}_{\Theta,I,\epsilon/2}$ coincide with symplectic Euler schemes in the coordinate $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ and hence are volume preserving (Hairer et al., 2006). Since $\boldsymbol{F}_{J,\varphi,\epsilon}$ is also volume-preserving by the results of Lemma 2.1, the composition (2.33) is volume-preserving. □

2.B Proof of Theorem 2.3

First we define a notion of *symplecticity*, a property of Hamiltonian dynamics that implies a volume preservation and further has important consequences in the stability of numerical approximation schemes (Hairer et al., 2006).

Definition 2.6. A differentiable map $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is called *symplectic* if

$$\frac{\partial(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial(\boldsymbol{\theta}, \boldsymbol{p})}^\intercal \boldsymbol{J} \frac{\partial(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial(\boldsymbol{\theta}, \boldsymbol{p})} = \boldsymbol{J} \quad \text{for } \boldsymbol{J} = \begin{bmatrix} 0 & \boldsymbol{I}_d \\ -\boldsymbol{I}_d & 0 \end{bmatrix} \tag{2.34}$$

where $\boldsymbol{I}_d$ denotes a d-dimensional identity matrix. A dynamics is called symplectic if its solution operator is.

Proof of Theorem 2.3. Reversibility is a standard property of smooth Hamiltonian dynamics with a symmetric kinetic energy (Hairer et al., 2006). Defined as a (pointwise) limit of smooth dynamics, discontinuous dynamics therefore is also reversible.

We turn to the proof of symplecticity. Under the assumption of Theorem 2.3, the evolution of discontinuous Hamiltonian dynamics from a state $(\boldsymbol{\theta}, \boldsymbol{p})$ at $t = 0$ to $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ at $t = \tau$ is given as follows. Dividing up the time intervals into a smaller pieces if necessary, we can without loss of generality assume that a trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ encounters only one discontinuity at $\boldsymbol{\theta}(t_e)$ during the interval $[0, \tau]$. Since $U(\boldsymbol{\theta})$ is piecewise constant, the momentum remains constant and $\boldsymbol{\theta}(t)$ travels in a straight line except when hitting the discontinuity. The relationship between $(\boldsymbol{\theta}, \boldsymbol{p})$ and $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is therefore given by

$$\begin{aligned} \boldsymbol{\theta}^* &= \boldsymbol{\theta} + t_e \nabla_{\boldsymbol{p}} K(\boldsymbol{p}) + (\tau - t_e) \nabla_{\boldsymbol{p}} K(\boldsymbol{p}^*) \\ \boldsymbol{p}^* &= \boldsymbol{p} + \gamma(\boldsymbol{p}) \boldsymbol{\nu}_e \end{aligned} \tag{2.35}$$

where $\gamma(\boldsymbol{p})$ is defined implicitly as a solution of the following relations. If $\Delta U_e < K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}_e)$ for ΔU_e defined as in (2.5), we define $\gamma(\boldsymbol{p})$ as a value that

satisfies:

$$K(\boldsymbol{p} - \gamma \boldsymbol{\nu}_e) = K(\boldsymbol{p}) + \Delta U_e \quad \text{with } \gamma > 0 \tag{2.36}$$

Otherwise, $\gamma(\boldsymbol{p})$ is defined through the relation:

$$K(\boldsymbol{p} - \gamma \boldsymbol{\nu}_e) = K(\boldsymbol{p}) \quad \text{with } \gamma > 0 \tag{2.37}$$

The uniqueness of solutions to the above relations is guaranteed by the convexity and growth condition on $K(\boldsymbol{p})$, and hence $\gamma(\boldsymbol{p})$ is well-defined. The event time t_e is also a function of $(\boldsymbol{\theta}, \boldsymbol{p})$ and can easily shown to be

$$t_e(\boldsymbol{\theta}, \boldsymbol{p}) = \frac{\alpha - \langle \boldsymbol{\theta}, \boldsymbol{\nu}_e \rangle}{\langle \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \boldsymbol{\nu}_e \rangle} \tag{2.38}$$

where α is the distance from the origin of the discontinuity plane of U at $\boldsymbol{\theta}(t_e)$. Assuming that $\boldsymbol{\theta}(t_e)$ is not at the intersection of the linear discontinuity planes and that $\Delta U_e \neq K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}_e)$, the relation (2.35) correctly describes the evolution of the dynamics on a small enough neighborhood of $(\boldsymbol{\theta}, \boldsymbol{p})$ with $\gamma(\boldsymbol{p})$ defined either through (2.36) or (2.37). The map $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ therefore is differentiable and Lemma 2.7 establishes the symplecticity through direct computation.

Lastly, we turn to the almost everywhere differentiability of discontinuous Hamiltonian dynamics. To characterize a state at which the solution operator fails to be differentiable, we first define the following sets:

- $\mathcal{D} = \{\, \boldsymbol{\theta} : \text{ multiple discontinuity boundaries of } U \text{ intersects at } \boldsymbol{\theta} \}$
- $\mathcal{U} = \{\, \Delta > 0 : \Delta = U(\boldsymbol{\theta}) - U(\boldsymbol{\theta}') \text{ for some } \boldsymbol{\theta}, \boldsymbol{\theta}' \}$
- $\mathcal{V} = \{\, \boldsymbol{\nu} : \ \boldsymbol{\nu} \text{ is orthonormal to a discontinuity boundary of } U \}$

The above sets are all countable by our assumption on $U(\boldsymbol{\theta})$. Based on the behavior of a trajectory as described in the previous paragraph, a trajectory from the initial

state $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ potentially experiences a non-differentiable behavior at time t only if the initial state belongs to one of the sets below:

$$\bigcup_{\boldsymbol{\theta}\in\mathcal{D}} \{(\boldsymbol{\theta} + s\nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \boldsymbol{p}) : s \in \mathbb{R}\}, \quad \bigcup_{\Delta\in\mathcal{U},\boldsymbol{\nu}\in\mathcal{V}} \left\{(\boldsymbol{\theta}, \boldsymbol{p}) : K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}) = \Delta\right\}$$
$$\left\{(\boldsymbol{\theta}, \boldsymbol{p}) : t = \frac{\alpha - \langle \boldsymbol{\theta}, \boldsymbol{\nu}_e\rangle}{\langle \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \boldsymbol{\nu}_e\rangle}\right\} \tag{2.39}$$

Being a countable union of lower dimensional manifolds, all of the sets above have measure zero. □

Lemma 2.7. *The map* (2.35) *is symplectic for* $\gamma(\boldsymbol{p})$ *and* $t_e(\boldsymbol{\theta}, \boldsymbol{p})$ *as defined through* (2.36), (2.37), *and* (2.38).

Proof of Lemma 2.7. To simplify expressions, we denote $\boldsymbol{w} = \nabla_{\boldsymbol{p}} K(\boldsymbol{p})$, $\boldsymbol{w}^* = \nabla_{\boldsymbol{p}} K(\boldsymbol{p}^*)$, and let $\boldsymbol{A}$ and $\boldsymbol{A}^*$ denote the Hessians of K at $\boldsymbol{p}$ and $\boldsymbol{p}^*$. First, an implicit differentiation of either (2.36) or (2.37) with some algebra yields

$$\frac{\partial \gamma}{\partial \boldsymbol{p}} = \frac{\boldsymbol{w}^\intercal - \boldsymbol{w}^{*\intercal}}{\langle \boldsymbol{w}^*, \boldsymbol{\nu}_e\rangle} \tag{2.40}$$

Differentiating (2.35) with respect to $(\boldsymbol{\theta}, \boldsymbol{p})$, we obtain

$$\frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{\theta}} = \boldsymbol{I} - \frac{(\boldsymbol{w} - \boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal}{\langle \boldsymbol{w}, \boldsymbol{\nu}_e\rangle}, \quad \frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{p}} = t_e \boldsymbol{A} - \frac{t_e}{\langle \boldsymbol{w}, \boldsymbol{\nu}_e\rangle}(\boldsymbol{w} - \boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal \boldsymbol{A} + (\tau - t_e)\boldsymbol{A}^* \frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}}$$
$$\frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{\theta}} = 0, \qquad \frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}} = \boldsymbol{I} + \frac{\boldsymbol{\nu}_e(\boldsymbol{w} - \boldsymbol{w}^*)^\intercal}{\langle \boldsymbol{w}^*, \boldsymbol{\nu}_e\rangle} \tag{2.41}$$

When $\partial \boldsymbol{p}^*/\partial \boldsymbol{\theta} = 0$, the symplecticity condition (2.34) simplifies to:

$$\frac{\partial \boldsymbol{\theta}^{*\intercal}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}} = \boldsymbol{I}, \quad \frac{\partial \boldsymbol{p}^{*\intercal}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{p}} = \left(\frac{\partial \boldsymbol{p}^{*\intercal}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{p}}\right)^\intercal \tag{2.42}$$

The first equality in (2.42) is easily verified from (2.41). To establish the second equality of (2.42), we need to verify the symmetry of the matrix

$$\frac{\partial \boldsymbol{p}^{*\intercal}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{p}} = t_e \frac{\partial \boldsymbol{p}^{*\intercal}}{\partial \boldsymbol{p}} \left(\boldsymbol{I} - \frac{(\boldsymbol{w} - \boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal}{\langle \boldsymbol{w}, \boldsymbol{\nu}_e \rangle} \right) \boldsymbol{A} + (\tau - t_e) \frac{\partial \boldsymbol{p}^{*\intercal}}{\partial \boldsymbol{p}} \boldsymbol{A}^* \frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}} \tag{2.43}$$

The first term of (2.43) simplifies to $t_e \boldsymbol{A}$, which is symmetric, and the second term is obviously symmetric. □

2.C Event-driven integrator for Gaussian momentum

For simplicity, we assume that a parameter space $\boldsymbol{\theta}$ consists only of the embedded discrete parameters as described in Section 2.2.1, so that the target $\pi_\Theta(\cdot)$ is piecewise constant with the discontinuity set consisting of the boundaries of hyper-cubes. The pseudo code of an event-driven integrator in this setting is given in Algorithm 2.3. The integrator is energy-preserving and hence yields a rejection-free proposal.

2.D Additional details on Jolly-Seber model

As mentioned earlier, the notations in our description of the Jolly-Seber model follows the literature on capture-recapture models and deviate from the rest of the chapter.

2.D.1 Observed quantities / statistics

Under appropriate assumptions, details of which we refer the reader to Seber (1982), the likelihood of the Jolly-Seber model depends only on the following statistics from a capture-recapture experiment carried over $i = 1, \ldots, T$ capture occasions:

- $R_i =$ number of marked animals released after the ith capture occasion.
- $r_i =$ number of animals from the released R_i animals that are subsequently captured.

Algorithm 2.3 Event-driven integrator for $K(\boldsymbol{p}) = \|\boldsymbol{p}\|^2/2$

Input: initial state $(\boldsymbol{\theta}, \boldsymbol{p})$, integration time τ

```
t ← 0
while t < τ do
    t_e ← the time until reaching the next discontinuity
    if t + t_e < τ then
        θ ← θ + (τ − t)p
        t ← τ
    else
        θ ← θ + t_e p
        i ← the index of the axis orthogonal to the discontinuity plane at θ
        ΔU_e ← the potential energy difference
        if p_i^2/2 > ΔU_e then
            p_i ← sqrt(p_i^2 − 2ΔU_e)
        else
            p_i ← −p_i
        end if
        t ← t + t_e
    end if
end while
```

- $z_i =$ number of animals that are caught before ith capture occasion, not caught in the ith capture occasion, but caught subsequently.
- $m_i =$ number of marked animals caught at the ith capture occasion.
- $u_i =$ number of unmarked animals caught at the ith capture occasion.

2.D.2 *Likelihood function*

The likelihood decomposes into two parts: one for the first captures of previously unmarked animals and another for their re-captures. More precisely,

$$\begin{aligned} L(\text{data} \,|\, \boldsymbol{U}, \boldsymbol{p}, \boldsymbol{\phi}) &= L(\text{first captures}) \times L(\text{re-captures}) \\ L(\text{first captures}) &\propto \prod_{i=1}^{T} \frac{U_i!}{U_i - u_i!} p_i^{u_i} (1-p_i)^{U_i - u_i} \\ L(\text{re-captures}) &\propto \prod_{i=1}^{T-1} \chi_i^{R_i - r_i} \{\phi_i (1-p_{i+1})\}^{z_{i+1}} (\phi_i p_{i+1})^{m_{i+1}} \end{aligned} \tag{2.44}$$

where χ_i represents the conditional probability that a marked animal released after the ith capture occasion is not caught again. Mathematically, χ_i is defined recursively as

$$\chi_{T-1} = 1 - \phi_{T-1} p_T, \quad \chi_i = 1 - \phi_i \{p_{i+1} + (1-p_{i+1})(1-\chi_{i+1})\} \tag{2.45}$$

2.D.3 *Prior distribution for* $U_{i+1} \,|\, U_i, \phi_i$

Let B_i denote the number of "births," representing animals that are born, enters (immigration), or leaves (emigration) the population after the ith occasion and remains so until the $(i+1)$th occasion. Also let S_i denote the number of animals that are unmarked right after the ith capture occasion and survives until the next capture occasion. Then we have $U_{i+1} = B_i + S_i$ where $S_i \,|\, U_i, u_i, \phi_i \sim \text{Binom}(\phi_i, U_i - u_i)$.

The prior distribution of $\{U_i\}_{i=1}^T$ can thus be induced by assigning a prior on B_i's. In our example, we assign a convenient prior on U_i's based on the assumptions that 1) $\text{Binom}(\phi_i, U_i - u_i)$ can be approximated by $\mathcal{N}(U_i - u_i, \phi_i(1-\phi_i))$ and 2) B_i's are approximately i.i.d. $\mathcal{N}(0, \sigma_B^2)$. These assumptions motivates a prior

$$U_{i+1} \,|\, U_i, u_i, \phi_i, \sigma_B \sim \lfloor \mathcal{N}(U_i - u_i, \sigma_B^2 + \phi_i(1-\phi_i)) \rfloor \tag{2.46}$$

where $\lfloor \cdot \rfloor$ is a floor function. We used $\sigma_B = 500$ in our example of Section 2.5.1. An alternative prior on $\{U_i\}_{i=1}^T$ can be assigned to reflect different model and prior

assumptions on the number of births. For instance, it is more natural to constrain $B_i \geqslant 0$ in some cases (Schwarz and Arnason, 1996) and a binomial distribution on B_i will for example induce a poisson-binomial distribution on the conditional $U_{i+1} \,|\, U_i, u_i, \phi_i$ after marginalizing over B_i and S_i.

2.D.4 Inference on unknown population sizes

In case the total population sizes $\{N_i\}_{i=1}^T$ at each capture occasion are of interest, we can generate their posterior samples using the relation $N_i = M_i + U_i$ where M_i denotes the number of marked animals right before the $(i+1)$th capture occasion. The distribution of $\{M_i\}_{i=1}^T$ follows $M_0 = 0$ and $M_{i+1} \,|\, M_i, \phi_i \sim \text{Binom}(M_i, \phi_i)$.

2.E Additional numerical results

2.E.1 Comparison of DHMC and Gibbs in synthetic example

We use a synthetic target distribution to demonstrate the difference between Metropolis-within-Gibbs with and without momentum as discussed in Section 2.4.4. While DHMC requires neither conjugacy or smoothness of the conditional densities, we choose a multivariate Gaussian target distribution so that we can compare DHMC to an optimal Metropolis-within-Gibbs implementation with the univariate proposal variances chosen according to the theory of Gelman et al. (1996). In particular, we assume that the target distribution of $\boldsymbol{\theta}$ follows that of a stationary unit variance auto-regressive process of the form

$$\theta_t = \alpha\theta_{t-1} + \sqrt{1-\alpha^2}\eta_t, \quad \theta_1, \eta_t \sim \mathcal{N}(0,1) \tag{2.47}$$

for $t = 2, \ldots, 1000$ with $\alpha = 0.9$.

We compare the performances of four algorithms: DHMC (coordinate-wise), Gibbs (full conditional updates), Metropolis-within-Gibbs (univariate updates with optimal proposal variances), and NUTS (No-U-Turn-Sampler of Hoffman and Gelman

Table 2.3: Performance summary of each algorithm on the auto-regressive process example. The term ($\pm \ldots$) is the error estimate of our ESS estimators. ESS per unit time normalizes the ESS's with computational efforts. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per unit time	Path length	Iter time
DHMC	77.4 ($\pm$ 5.2)	1.56	49.5	49.5
NUTS	52.4 ($\pm$ 3.2)	N/A	142	N/A
Gibbs	0.949 ($\pm$ 0.076)	0.949	1	1
Metropolis-within	0.219 ($\pm$ 0.015)	0.219	1	1

(2014)). The performance of each algorithm is summarized in Table 2.3, which shows that DHMC outperforms not only Metropolis-within-Gibbs but also Gibbs (recall that DHMC requires no closed form conditionals at all). After accounting for the computational costs, DHMC improves Gibbs by over 50% and Metropolis-within-Gibbs by over 600%. In general, the advantage of DHMC over Gibbs is expected to increase as the correlations among the parameters increase because the use of momentum can suppress the "random walk behavior" (Neal, 2010). The covariance matrix of the target distribution here has a condition number $\approx 19^2$, which corresponds to a substantial but not particularly severe correlations.

In computing ESS per unit time, we estimated theoretical and platform-independent relative computational time of the algorithms as follows. In reasonable low-level language implementations, the computation of conditional densities should account for the majority of computational times for a typical target distribution. Therefore, computational efforts should be roughly equivalent between one numerical integration step of DHMC and one iteration of (Metropolis-within) Gibbs sampler. The computational cost of NUTS relative to these algorithms is more specific to individual target distributions, depending strongly on specific structures such as conditional independence among the parameters. For this reason, we do not attempt to compare NUTS to the other algorithms in terms of ESS per unit time.

2.E.2 *Multiple change-point detection for auto-regressive conditional heteroscedastic processes*

Auto-regressive conditional heteroscedastic processes (ARCH) are a popular model for log-returns of speculative prices such as stock market indices. A non-stationary first-order ARCH process $\{y_t\}_{t=1}^T$ with parameters $\{a(t), b(t)\}_{t=1}^T$ assumes the distribution

$$y_t \,|\, y_{t-1}, a, b \sim \mathcal{N}(0, \sigma_t^2) \quad \text{where} \quad \sigma_t^2 = a(t) + b(t)\, y_{t-1}^2 \tag{2.48}$$

Motivated by its interpretability and advantage in forecasting, Fryzlewicz and Subba Rao (2014) propose a piecewise constant parametrization of $a(t)$ and $b(t)$ as follows:

$$\big(a(t), b(t)\big) = (a_k, b_k) \;\text{ if }\; \tau_{k-1} < t \leqslant \tau_k \tag{2.49}$$

for $k = 1, \ldots, K$ where the number of change points K and their locations $1 = \tau_0 < \tau_1 < \ldots < \tau_K$ are to be estimated along with (a_k, b_k)'s.

To fit the above model within a Bayesian paradigm, we infer the change points through a variable selection type approach as follows, using the horseshoe shrinkage priors of Carvalho et al. (2010). We first choose an upper bound $K_{\max}$ on the number of change points and assume a uniform prior on τ_k's on the constrained space $1 < \tau_1 < \ldots < \tau_{K_{\max}} < T$. We then model the changes in the values of $a(t)$ and $b(t)$ through a prior

$$\begin{array}{l} \log(a_k/a_{k-1}) \sim \mathcal{N}\left(0, \sigma_a \eta_{a,k}\right) \\ \log(b_k/b_{k-1}) \sim \mathcal{N}\left(0, \sigma_b \eta_{b,k}\right) \end{array} \quad \text{with} \quad \eta_{a,k}, \eta_{b,k} \sim \text{Cauchy}^+(0,1) \tag{2.50}$$

where $\text{Cauchy}^+(0,1)$ denotes the standard half-Cauchy prior and σ_a and σ_b are the global shrinkage parameters (Carvalho et al., 2010). The above approach can "select" a subset of $\tau_1, \ldots, \tau_{K_{\max}}$ as real change points by removing the others through shrinkage $a_k \approx a_{k-1}$ and $b_k \approx b_{k-1}$. We place a default prior $\sigma_a, \sigma_b \sim \text{Cauchy}^+(0,1)$ for the global shrinkage parameters (Gelman, 2006), and a weak prior $a_0, b_0 \sim \mathcal{N}(0,1)$ for the initial volatility parameters.

Following Fryzlewicz and Subba Rao (2014), we fit our model to the log return values of a stock market index over a period that includes the subprime mortgage crisis. In particular, we use the daily closing values of S&P 500 on the market opening days during the period from Jan 1st, 2005 to Dec 31st, 2009.[4] The model parameters in this example are largely nonidentifiable even with the order constraint $\tau_1, \ldots, \tau_{K_{\max}}$. In such cases, it is not clear if the minimum ESS across the individual parameters is a good measure of efficiency. For this example, therefore, we calculate the minimum ESS over the first and second moments of the following quantities: the hyper-parameters σ_a and σ_b, log posterior density, and four summary statistics of the estimated functions $a(t)$ and $b(t)$ as defined in the footnote.[5] Both algorithms are run for 2.5×10^4 iterations from stationarity.

The simulation results are summarized in Table 2.4. This example is challenging for DHMC as the posterior of τ_k's are multi-modal conditionally on the continuous parameters, while the full conditional update of τ_k's by NUTS-Gibbs does not suffer from such multi-modality. The complex dependency between the local shrinkage and the other parameters creates potential paths among the modes, however. It seems that DHMC can exploit this complex posterior geometry efficiently and be competitive with NUTS-Gibbs. Figure 2.4 plots 100 DHMC posterior samples of the piecewise constant volatility functions $a(t)$ and $b(t)$ to illustrate the posterior structure of the model.

---

[4] The log return value cannot be computed when a daily closing value exactly coincides with the previous one. There were four such days during the period and these data points were removed.

[5] We define the four summary statistics $\log(\|a\|_2), \log(\|b\|_2), C_a$, and C_b as follows. The quantity $\|a\|_2$ summarizes the deviation of $a(t)$ from its posterior (pointwise empirical) mean $\hat{a}(t)$ and is defined as $\|a\|_2 = \sum_{t=1}^{T} |a(t) - \hat{a}(t)|^2$. The statistics C_a is a surrogate for the number of "change points" in the function $a(t)$:

$$C_a = \left| \{k \in \{1, \ldots, K_{\max}\} : |\log(a_k / a_{k-1})| > .1\} \right| \tag{2.51}$$

The statistics $\|b\|_2$ and C_b are defined analogously.

Table 2.4: Performance summary of each algorithm on the change points detection example. The term $(\pm \ldots)$ is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC	13.7 ($\pm$ 1.1)	38.7	87.3	1.03
NUTS-Gibbs	11.6 ($\pm$ 3.2)	33.5	218	1

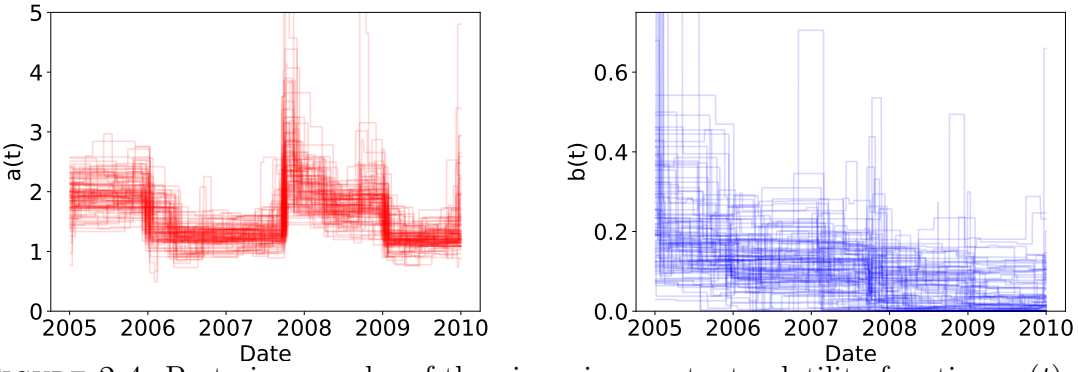


FIGURE 2.4: Posterior samples of the piecewise constant volatility functions $a(t)$ and $b(t)$ from 100 iterations of DHMC.

3

Geometrically tempered Hamiltonian Monte Carlo

3.1 Introduction

HMC faces major problems when posterior distributions are multimodal. This chapter attempts to address this problem to obtain a general approach for accelerating mixing of HMC including in multi-modal cases.

Hamiltonian dynamics generates trajectories that move along the level sets of a scalar function commonly referred to as a Hamiltonian or energy. This property is known as *conservation of energy* in physics. HMC exploits this property to generate proposals that are far away from the current state yet are accepted with high probability. If the parameter of interest has a distribution with multiple modes separated by a region of low probability density, however, the conservation of energy almost completely eliminates the possibility of HMC transitioning from one mode to another in a small number of iterations (Section 3.2.2; Neal (2010)). This issue is inherent in the choice of Hamiltonian dynamics underlying HMC's proposal mechanism and consequently most variations of HMC (Hoffman and Gelman, 2014; Neal, 1994; Shahbaba et al., 2014; Sohl-Dickstein et al., 2014) similarly suffer in the presence of

multi-modality.

Girolami and Calderhead (2011) proposed Riemann manifold HMC (RMHMC), an extension of HMC that modifies the underlying Hamiltonian dynamics through distortion of local distances by a Riemannian metric. Their choice of metric, Fisher information, is not designed to facilitate sampling from a multimodal target distribution, but their work spurred a question: can a metric be chosen to help HMC sample more efficiently from multi-modal distributions? (See "Discussion on the paper" section in Girolami and Calderhead (2011).) In this chapter, we provide a positive answer to this question by proposing a class of metric specifically designed to lower the "energy barriers" among the modes, thereby enabling trajectories of Hamiltonian dynamics to transition from one mode to another more frequently. We call RMHMC under this class of metric as geometrically tempered HMC (GTHMC) due to its similarities to other tempering methods. While geometric methods in statistics are usually motivated using the language of intrinsic geometry (Amari and Nagaoka, 2007; Girolami and Calderhead, 2011; Xifara et al., 2014), we develop a geometric theory behind RMHMC using the language of extrinsic geometry, thereby making the results more explicit and intuitive as well as accessible to a wider audience.

Choosing a metric to adapt HMC to multimodal target distributions was previously considered by Lan et al. (2015). Their approach, however, requires knowledge of the mode locations, substantial hand tuning, and ad hoc additions of drifts to the dynamics which can in general undermine the desirable properties of RMHMC. Many of these issues arise from the lack of precise treatment of geometry behind RMHMC and are all solved by GTHMC. Another related work is Roberts and Stramer (2002) where they consider using what they call Langevin tempered dynamics as a proposal generation mechanism. This dynamics is a Langevin dynamics analogue of Hamiltonian dynamics under isometric tempering, a special case of our geometric tempering method discussed in Section 3.3. Both Langevin and Hamiltonian dynamics

explore the parameter space with highly variable velocities under geometric tempering, making their discrete approximation challenging (Section 3.4; Roberts and Stramer (2002)). The deterministic nature of Hamiltonian dynamics, however, allows an accurate approximation of the dynamics in a relatively efficient manner through the variable stepsize integrator proposed in Section 3.4.

The rest of the chapter is organized as follows. In Section 3.2, we motivate our choice of metric for GTHMC by developing geometric intuitions behind RMHMC using the language of extrinsic geometry. Section 3.3 provides two example classes of GTHMC algorithms. Section 3.4 develops a novel variable stepsize integrator for Hamiltonian dynamics, motivated by the need for an improvement over the standard Störmer-Verlet scheme that produces unstable trajectories in GTHMC settings. An effective application of the variable stepsize integrator to GTHMC and other HMC variants calls for an improved acceptance-rejection mechanism, and this is also described in Section 3.4. Both the integrator and acceptance-rejection algorithm are general tools of independent interest. In Section 3.5, we compare the performance of GTHMC to HMC on various examples and demonstrate its superiority in terms of effective sample sizes.

3.2 Motivation and geometric theory behind GTHMC

We begin this section with a brief review of RMHMC. We then discuss why HMC variants in general perform poorly on multimodal target distributions, which leads to a simple motivation for GTHMC defined in Section 3.2.3. In the subsequent subsections, we develop more precise geometric theory behind GTHMC.

3.2.1 Hamiltonian dynamics and RMHMC

RMHMC falls under the general framework of HMC variants described in Section 1. Under RMHMC, the momentum distribution $\pi_{P|\Theta}(\boldsymbol{p}\,|\,\boldsymbol{\theta})$ is chosen as $\mathcal{N}(\boldsymbol{p}; \mathbf{0}, \boldsymbol{G}(\boldsymbol{\theta}))$,

corresponding to the Hamiltonian

$$H(\boldsymbol{\theta}, \boldsymbol{p}) = -\log \pi(\boldsymbol{\theta}) + \log |\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} + \frac{1}{2}\boldsymbol{p}^T \boldsymbol{G}^{-1}(\boldsymbol{\theta})\boldsymbol{p}. \tag{3.1}$$

The conditional covariances of the momentum $\{\boldsymbol{G}(\boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ are referred to as a *mass tensor* or *Riemannian metric* (Bennett, 1975; Neal, 2010; Girolami and Calderhead, 2011).

For the purpose of our discussion here, let us suppose that Hamiltonian dynamics (1.4) corresponding to the Hamiltonian (3.1) can be solved exactly. Rephrasing Algorithm 1.1 in the specific case here, RMHMC works as follows:

Algorithm 3.1 (RMHMC w/o numerical approximation)**.** RMHMC generates a Markov chain with the following transition rule $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$:

1) Sample the momentum from its conditional distribution $\boldsymbol{p} \,|\, \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{p}; \boldsymbol{0}, \boldsymbol{G}(\boldsymbol{\theta}))$.

2) Propose $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$, where $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \in [0,\tau]}$ is the solution of the dynamics corresponding to the Hamiltonian (3.1) with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}, \boldsymbol{p})$.

3) Accept the proposal with probability 1 (because we assumed no numerical approximation error; see Section 1.1).

RMHMC recovers a familiar HMC (Duane et al., 1987; Neal, 2010) when the mass matrix is independent of the position variable. See Neal (2010) and Girolami and Calderhead (2011) for more detailed presentations on HMC and RMHMC.

3.2.2 Multi-modality and conservation of Energy

We now explain how existing HMC variants suffer from multimodality in the target distribution. For simplicity we consider the basic version of (Riemann manifold) HMC as in Algorithm 3.1 with a constant mass matrix $\boldsymbol{M}$, but the following analysis applies equally to the other HMC variants.

Consider a Hamiltonian as a sum of potential energy $U(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta})$ and kinetic energy $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T \boldsymbol{M}^{-1}\boldsymbol{p}$ (up to an additive constant). The *energy barrier* with respect to a potential energy function U from a position $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$ is the smallest possible energy increase along a continuous path from $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$:

$$B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U) := \inf_{\boldsymbol{\gamma} \in C^0} \left\{ \max_{0 \leqslant t \leqslant 1} U(\boldsymbol{\gamma}(t)) - U(\boldsymbol{\theta}_1) \,\middle|\, \boldsymbol{\gamma}(0) = \boldsymbol{\theta}_1 \text{ and } \boldsymbol{\gamma}(1) = \boldsymbol{\theta}_2 \right\} \tag{3.2}$$

where C^0 denotes a class of continuous functions. The quantity $B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U)$ is the minimum amount of kinetic energy from Step 1 of Algorithm 3.1 needed for HMC to reach $\boldsymbol{\theta}_2$ from $\boldsymbol{\theta}_1$ in a single iteration; to see this, notice that a trajectory of Hamiltonian dynamics satisfies the following relation due to the conservation of energy:

$$U(\boldsymbol{\theta}(t)) - U(\boldsymbol{\theta}_0) = K(\boldsymbol{p}_0) - K(\boldsymbol{p}(t)) \leqslant K(\boldsymbol{p}_0) \tag{3.3}$$

where $\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$ and $\boldsymbol{p}(0) = \boldsymbol{p}_0$. The quantity $K(\boldsymbol{p}_0) - K(\boldsymbol{p}(t))$ is the amount of energy transferred from kinetic to potential at time t. Since the increase in the potential energy along a trajectory is upper bounded by $K(\boldsymbol{p}_0)$, the trajectory generated in Step 2 of HMC will not be able to reach $\boldsymbol{\theta}_2$ if the kinetic energy from Step 1 is smaller than $B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U)$. This is problematic for HMC as the energy barrier $B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U)$ would be high if $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ were two modes of $\pi(\boldsymbol{\theta})$ with a region of low probability in between. To make things worse, there is no guarantee that a momentum variable with minimum required kinetic energy will actually generate a path between two modes.

3.2.3 Simple motivation for GTHMC

We now define GTHMC and provide a simple motivation behind it.

Definition 3.1 (GTHMC)**.** GTHMC is a sub-class of RMHMC in which a metric

$\boldsymbol{G}_T(\boldsymbol{\theta})$ satisfies the following relation for $T > 1$

$$|\boldsymbol{G}_T(\boldsymbol{\theta})|^{1/2} \propto \pi(\boldsymbol{\theta})^{1-\frac{1}{T}} \tag{3.4}$$

where $|\boldsymbol{M}|$ denotes the determinant of a matrix $\boldsymbol{M}$.

For GTHMC at temperature T, the Hamiltonian decomposes into a potential energy $U_T(\boldsymbol{\theta})$ and kinetic energy $K_T(\boldsymbol{\theta}, \boldsymbol{p})$ where

$$U_T(\boldsymbol{\theta}) = -\frac{1}{T}\log \pi(\boldsymbol{\theta}), \quad K_T(\boldsymbol{\theta}, \boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T \boldsymbol{G}_T^{-1}(\boldsymbol{\theta})\boldsymbol{p}$$

and $K_T(\boldsymbol{\theta}, \boldsymbol{p}) \sim \chi^2_d/2$ irrespective of T. As in the HMC setting (3.3), the conservation of energy implies that

$$U_T(\boldsymbol{\theta}(t)) - U_T(\boldsymbol{\theta}_0) = K_T(\boldsymbol{\theta}_0, \boldsymbol{p}_0) - K_T(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$$

where $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ denotes a trajectory of the corresponding Hamiltonian dynamics with the initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. Again, $K_T(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ is the maximum possible increase in the potential energy along the trajectory of Hamiltonian dynamics. Now notice that $U_T(\boldsymbol{\theta}) = U_1(\boldsymbol{\theta})/T$ and therefore we have

$$B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U_T) = \frac{1}{T} B(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2; U_1)$$

Hence, the energy barrier from $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$ becomes lower as T becomes large, requiring less kinetic energy for the trajectory to reach $\boldsymbol{\theta}_2$ from $\boldsymbol{\theta}_1$. As we discuss in more detail below (in particular, see the remark in Section 3.2.5), a property similar to (3.4) is not only a convenient way but also a requirement to allow RMHMC to move from one mode to another in a small number of iterations.

3.2.4 Geometric intuition behind RMHMC

To further motivate GTHMC, we establish a theoretical result on RMHMC that provides a novel geometric intuition behind the algorithm. Our approach is to

describe the Hamiltonian dynamics underlying RMHMC in terms of a more intuitive Newtonian dynamics on a manifold embedded in a Euclidean space. The required knowledge of Riemannian geometry is minimal and Appendix 3.D provides further background information.

Newtonian Dynamics on a Manifold

We first review Newtonian dynamics on a Euclidean space, which can be considered as a special case of Hamiltonian dynamics when the mass matrix is proportional to the identity and the Hamiltonian takes the form $H(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + \boldsymbol{p}^T\boldsymbol{p}/2$ for a potential energy function $U(\boldsymbol{\theta})$. In this case, the Hamilton's equation recovers Newtonian mechanics' description of the motion of a unit-mass particle in the potential energy field $U(\boldsymbol{\theta})$:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \boldsymbol{p}, \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla U(\boldsymbol{\theta})$$

The first equation simply expresses the fact that velocity is a time derivative of position. The second equation expresses Newton's second law; acceleration is proportional to force, the negative gradient of potential energy in our case. Borrowing from Neal (2010), Newtonian dynamics in two dimensions can be imagined as a motion of a frictionless puck that slides over a surface of height $U(\boldsymbol{\theta})$. At the position $\boldsymbol{\theta}$, the puck experiences a force in the direction of greatest descent $-\nabla U(\boldsymbol{\theta})$. If the surface is flat around $\boldsymbol{\theta}$ (i.e. $\nabla U \equiv \mathbf{0}$), the puck continues to move at a constant velocity in this area.

Now consider a potential energy $\tilde{U}(\tilde{\boldsymbol{\theta}})$ defined on a d-dimensional manifold $M \subset \mathbb{R}^{\tilde{d}}$ and let $T_{\tilde{\boldsymbol{\theta}}}M \subset \mathbb{R}^{\tilde{d}}$ denote the tangent space of M at $\tilde{\boldsymbol{\theta}}$. As in a Euclidean space, Newtonian dynamics on a manifold describes the motion of a particle under the potential energy field $\tilde{U}(\tilde{\boldsymbol{\theta}})$ driven in the direction of the greatest energy decrease, except that the particle is now constrained on a manifold M. Denoting the gradient

of $\tilde{U}$ on M by $\nabla^M \tilde{U}(\tilde{\boldsymbol{\theta}})$, Newtonian dynamics on a manifold is defined as follows:

Definition 3.2 (Newtonian dynamics on a manifold)**.** A trajectory of Newtonian dynamics on a manifold M under the potential energy field $\tilde{U}(\tilde{\boldsymbol{\theta}})$ with an initial condition $\tilde{\boldsymbol{\theta}}_0 \in M$ and $\tilde{\boldsymbol{p}}_0 \in T_{\tilde{\boldsymbol{\theta}}_0} M$ is a unique solution $(\tilde{\boldsymbol{\theta}}(t), \tilde{\boldsymbol{p}}(t)) \in M \times T_{\tilde{\boldsymbol{\theta}}(t)} M$ of the differential equation

$$\frac{\mathrm{d}\tilde{\boldsymbol{\theta}}}{\mathrm{d}t} = \tilde{\boldsymbol{p}}, \quad \frac{\mathrm{d}\tilde{\boldsymbol{p}}}{\mathrm{d}t} = -\nabla^M \tilde{U}(\tilde{\boldsymbol{\theta}})$$

such that $\tilde{\boldsymbol{\theta}}(0) = \tilde{\boldsymbol{\theta}}_0$ and $\tilde{\boldsymbol{p}}(0) = \tilde{\boldsymbol{p}}_0$.

RMHMC in terms of Newtonian Dynamics on a Manifold

Just as in Euclidean space, we can run HMC on a manifold by solving the Newtonian dynamics to generate samples from a given target distribution. We introduce HMC on a manifold as a theoretical tool to enhance our understanding of RMHMC, so we do not concern ourselves with how the Newtonian dynamics on a manifold may be numerically approximated.

Definition 3.3 (HMC on a manifold)**.** Given a pdf $\tilde{\pi}(\tilde{\boldsymbol{\theta}})$ on a manifold M, the following procedures generate a Markov chain $\{(\tilde{\boldsymbol{\theta}}^{(i)}, \tilde{\boldsymbol{p}}^{(i)})\}_{i=1}^{\infty}$ whose stationary distribution has the marginal $\tilde{\pi}(\tilde{\boldsymbol{\theta}})$. 1. Sample $\tilde{\boldsymbol{p}}^{(i)}$ from the standard Gaussian on $T_{\tilde{\boldsymbol{\theta}}^{(i)}} M$. 2. Set $(\tilde{\boldsymbol{\theta}}^{(i+1)}, \tilde{\boldsymbol{p}}^{(i+1)}) = (\tilde{\boldsymbol{\theta}}^{(i)}(\tau), -\tilde{\boldsymbol{p}}^{(i)}(\tau))$ where $\{(\tilde{\boldsymbol{\theta}}^{(i)}(t), \tilde{\boldsymbol{p}}^{(i)}(t))\}_t$ is a solution of the Newtonian dynamics as in Definition 3.2 with the potential energy $\tilde{U}(\tilde{\boldsymbol{\theta}}) = -\log \tilde{\pi}(\tilde{\boldsymbol{\theta}})$ and the initial condition $(\tilde{\boldsymbol{\theta}}^{(i)}, \tilde{\boldsymbol{p}}^{(i)})$.

We now state our main theoretical result. Theorem 3.4 below provides valuable insights into the behaviors of RMHMC trajectories that are hard to predict otherwise.

Theorem 3.4 (RMHMC as reparametrization of HMC)**.** *Given a random variable $\boldsymbol{\theta} \sim \pi(\cdot)$ on $\mathbb{R}^d$, let $\tilde{\pi}(\cdot)$ denote the density of a random variable $\boldsymbol{g}(\boldsymbol{\theta})$. For the initial*

input $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ *and* $\tilde{\boldsymbol{\theta}}_0 = \boldsymbol{g}(\boldsymbol{\theta}_0)$*, let* $\left\{(\tilde{\boldsymbol{\theta}}^{(i)}, \tilde{\boldsymbol{p}}^{(i)})\right\}_{i=0}^N$ *be a Markov chain generated by HMC on a manifold as in Definition 3.3. Then a Markov chain* $\left\{\boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^T(\tilde{\boldsymbol{\theta}}^{(i)}, \tilde{\boldsymbol{p}}^{(i)})\right\}_{i=0}^N$ *on* $\mathbb{R}^d \times \mathbb{R}^d$ *defined through the map*

$$\boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^T(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{p}}) := \left(\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}),\ \mathbf{D}\boldsymbol{g}^T_{\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})}\tilde{\boldsymbol{p}}\right)$$

has the same distribution as the Markov chain generated by running RMHMC on $\mathbb{R}^d$ *with a metric* $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbf{D}\boldsymbol{g}^T_{\boldsymbol{\theta}}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$*.*

Theorem 3.4 is a consequence of the fact that $\boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^T$ bijectively maps Newtonian dynamics on a manifold onto the corresponding Hamiltonian dynamics on $\mathbb{R}^d$, formally stated as follows:

Theorem 3.5. *If* $(\tilde{\boldsymbol{\theta}}(t), \tilde{\boldsymbol{p}}(t))$ *is a solution of the Newtonian dynamics on* M *with a potential energy* $\tilde{U}(\tilde{\boldsymbol{\theta}}) = -\log \tilde{\pi}(\tilde{\boldsymbol{\theta}})$*, then* $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^T(\tilde{\boldsymbol{\theta}}(t), \tilde{\boldsymbol{p}}(t))$ *is a solution of Hamiltonian dynamics in* $\mathbb{R}^d$ *corresponding to the Hamiltonian (3.1) with* $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbf{D}\boldsymbol{g}^T_{\boldsymbol{\theta}}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$*.*

Theorems 3.4 and 3.5 in essence state that, up to numerical approximation errors in simulating trajectories, running RMHMC to sample from a parameter space $\boldsymbol{\theta} \in \mathbb{R}^d$ is equivalent to running HMC to sample from the reparametrization $\tilde{\boldsymbol{\theta}} = \boldsymbol{g}(\boldsymbol{\theta}) \in M$. This means that the metric $\boldsymbol{G}(\boldsymbol{\theta})$ should be chosen so that the reparametrization defines a well-conditioned distribution from which HMC can sample efficiently. In the special case when $\boldsymbol{g} = \widehat{\boldsymbol{\Sigma}}^{-1/2}$ is a linear operator and $\boldsymbol{G} = \widehat{\boldsymbol{\Sigma}}^{-1}$, Theorem 3.4 recovers a well-known fact on the effect of using a non-identity mass matrix in HMC (Neal, 2010). The Langevin dynamics analogue of Theorem 3.4 can also be established: see Supplement Section 3.B.

Remark 3.2.1. Theorem 3.4 and 3.5 start with a reparametrization $\boldsymbol{g}$ and identify the corresponding Riemannian metric as $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbf{D}\boldsymbol{g}^T_{\boldsymbol{\theta}}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$. Nash embedding theorem

(Nash, 1954) tells us that the construction can go in the other direction as well; given a metric $\boldsymbol{G}(\boldsymbol{\theta})$, there is a corresponding (local) reparametrization $\boldsymbol{g}$ so that HMC in the space $\boldsymbol{g}(\boldsymbol{\theta})$ is equivalent to RMHMC in the space $\boldsymbol{\theta}$ with a metric $\boldsymbol{G}(\boldsymbol{\theta})$.

3.2.5 Theory behind geometric tempering

Tempering methods are motivated by the fact that a distribution $\pi(\boldsymbol{\theta})^{1/T}/Z_T$, where Z_T is a normalizing constant, has less severe multi-modality than $\pi(\boldsymbol{\theta})$ for $T > 1$ (Earl and Deem, 2005; Geyer and Thompson, 1995; Marinari and Parisi, 1992; Neal, 2001). The main challenge is to relate the samples from the tempered distribution $\pi(\boldsymbol{\theta})^{1/T}/Z_T$ back to the original target $\pi(\boldsymbol{\theta})$.

GTHMC works by implicitly sampling from a transformed variable $\tilde{\boldsymbol{\theta}}$ such that the transformation $\boldsymbol{\theta} \to \tilde{\boldsymbol{\theta}}$ alleviates the multi-modality of $\pi(\boldsymbol{\theta})$. This implicit transformation is achieved by appropriately modifying Hamiltonian dynamics for the parameter $\boldsymbol{\theta}$, and therefore the samples generated from GTHMC retain the target distribution $\pi(\boldsymbol{\theta})$. Theorem 3.4 implies that the use of a metric with a property $|\boldsymbol{G}_T(\boldsymbol{\theta})|^{1/2} = \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$ corresponds to an implicit transformation $\tilde{\boldsymbol{\theta}} = \boldsymbol{g}(\boldsymbol{\theta})$ through a map $\boldsymbol{g}$ such that $\left|\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^T \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}\right|^{1/2} = \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$. This means that the transformed variable $\tilde{\boldsymbol{\theta}}$ would have the distribution

$$\tilde{\pi}(\tilde{\boldsymbol{\theta}}) \propto \pi \circ \boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})^{1/T}$$

by virtue of the (generalized) change of variable formula $\tilde{\pi}\left(\tilde{\boldsymbol{\theta}}\right) = \left|\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^T \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}\right|^{-1/2} \pi(\boldsymbol{\theta})$ (Federer, 1969). This is how GTHMC effectively lowers the energy barriers among the modes of $\pi(\boldsymbol{\theta})$ by a factor of $1/T$. Geometric tempering does not compete with existing tempering methods and in fact can be combined with them; see Section 3.6.

Remark 3.2.2. The implicit reparametrization $\boldsymbol{g} : \boldsymbol{\theta} \to \tilde{\boldsymbol{\theta}}$ under RMHMC has no effects on energy barriers among the modes if a metric $\boldsymbol{G}(\boldsymbol{\theta})$ has a constant volume factor $|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} = c$. More generally, the difference in the potential energy $\tilde{U}$ between

two positions $\tilde{\boldsymbol{\theta}}_1$ and $\tilde{\boldsymbol{\theta}}_2$ is given by

$$\log \frac{\tilde{\pi}(\tilde{\boldsymbol{\theta}}_2)}{\tilde{\pi}(\tilde{\boldsymbol{\theta}}_1)} = \log \frac{\pi(\boldsymbol{\theta}_2)}{\pi(\boldsymbol{\theta}_1)} - \frac{1}{2} \log \frac{|\boldsymbol{G}(\boldsymbol{\theta}_2)|}{|\boldsymbol{G}(\boldsymbol{\theta}_1)|}$$

where $\boldsymbol{\theta}_i = \boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}_i)$. The above equation shows that RMHMC has a measurable effect on the energy difference between $\tilde{\boldsymbol{\theta}}_1$ and $\tilde{\boldsymbol{\theta}}_2$ only if

$$\frac{|\boldsymbol{G}(\boldsymbol{\theta}_2)|}{|\boldsymbol{G}(\boldsymbol{\theta}_1)|} \underset{\sim}{\propto} \left(\frac{\pi(\boldsymbol{\theta}_2)}{\pi(\boldsymbol{\theta}_1)} \right)^{\alpha} \quad \text{for } \alpha > 0.$$

Thus any metric designed to promote the movements among the modes must locally have a property like $|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} \propto \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$ for $T > 1$, the defining characteristic of GTHMC.

3.3 Concrete examples of GTHMC

We have only assumed $|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} \propto \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$ in our development of GTHMC, leaving substantial flexibility in the choice of metric. We propose two computationally convenient variants of GTHMC, illustrating how the choice of metric affects performance. Simulation results are presented in Section 3.5, preceded by discussion on how to efficiently approximate the dynamics underlying GTHMC in Section 3.4.

3.3.1 Isometrically tempered HMC (ITHMC)

The choice $\boldsymbol{G}(\boldsymbol{\theta}) = g(\boldsymbol{\theta})\, \boldsymbol{I}$ with $g(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})^{\frac{2}{d}(1-\frac{1}{T})}$ is arguably the simplest way to satisfy the requirement $|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} \propto \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$. This metric modifies the local distance of the parameter space uniformly in all the directions, and therefore we call GTHMC with this choice of metric *isometrically tempered HMC* (ITHMC).

3.3.2 Directionally tempered HMC (DTHMC)

As mentioned in Section 3.2.2, an iteration of HMC with sufficient kinetic energy to overcome energy barriers does not guarantee a transition from one mode to another.

The transition can be infrequent even for GTHMC when a randomly generated trajectory from one mode tends not to travel in the direction of another. For this reason, a non-uniform distortion of the local distances can improve efficiency in certain situations (see Section 3.3.3). We let *directionally tempered HMC* (DTHMC) refer to a version of GTHMC in which the local distance in a particular direction is modified differently from the other directions. More precisely, we set

$$\boldsymbol{G}(\boldsymbol{\theta}) = g_{\parallel}(\boldsymbol{\theta})\, \boldsymbol{u}\boldsymbol{u}^T + g_{\perp}(\boldsymbol{\theta})\, (\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^T) \tag{3.5}$$

where $g_{\parallel}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})^{2\gamma(1-\frac{1}{T})}$, $g_{\perp}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})^{2\frac{1-\gamma}{d-1}(1-\frac{1}{T})}$, and $d^{-1} < \gamma \leqslant 1$. This metric modifies the distance only in the direction of $\boldsymbol{u}$ when $\gamma = 1$ while it coincides with the metric for ITHMC when $\gamma = d^{-1}$. This kind of metric is appropriate when it is known that the multi-modality is more severe in a particular direction.

The metric proposed in Lan et al. (2015) has an apparent similarity to (3.5) but lacks the crucial property $|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2} \propto \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$. Some degree of geometric tempering is achieved in their examples as a result of substantial manual tuning of the metrics based on the knowledge of mode locations. Even then, they have to resort to ad hoc additions of drifts to their dynamics to induce more frequent transitions among the modes. GTHMC provides more effective geometric tempering without such an extensive manual tuning.

3.3.3 Illustration of trajectories generated by GTHMC

We simulate some trajectories of HMC, ITHMC and DTHMC to illustrate the effect of geometric tempering as well as the difference between isometric and directional tempering. We construct a bi-modal target distribution $\pi(\cdot)$ as a mixture of 2-d standard Gaussians centered at $(-4, 0)$ and $(4, 0)$. For each of the algorithms, trajectories are simulated for $t = 3$ from a high density region near $(-4, 0)$, all having the same initial kinetic energy $K(\boldsymbol{\theta}_0, \boldsymbol{p}_0) = \boldsymbol{p}_0^T \boldsymbol{G}(\boldsymbol{\theta}_0)^{-1} \boldsymbol{p}_0 \,/\, 2 = 0.8$. For DTHMC, the

tempering direction is along the x-axis (i.e. $\boldsymbol{u} = (1, 0)$ in Equation (3.5)) and the temperature is set at $T = 15$ for both DTHMC and ITHMC. Between the two modes, the energy barrier with respect to the potential energy $U = -\log \pi$ is roughly given by $U(0, 0) - U(-4, 0) \approx 7.3$, so the geometrically tempered trajectories have more than enough kinetic energy to overcome the barrier as $K(\boldsymbol{\theta}_0, \boldsymbol{p}_0) = 0.8 > 7.3 \,/\, T$.

Figure 3.1 shows the trajectories generated as described above. For HMC and DTHMC, the trajectories of the same color are meant to be directly comparable as they have exactly the same value of $(\boldsymbol{\theta}_0, \boldsymbol{G}(\boldsymbol{\theta}_0)^{-1/2}\boldsymbol{p}_0)$ (recall that $\boldsymbol{G}(\boldsymbol{\theta})^{-1/2}\boldsymbol{p} \,|\, \boldsymbol{\theta} \sim$ Normal$(\mathbf{0}, \boldsymbol{I})$ irrespective of the choice of a metric). The ITHMC trajectories were given similar but not necessarily comparable values of $(\boldsymbol{\theta}_0, \boldsymbol{G}(\boldsymbol{\theta}_0)^{-1/2}\boldsymbol{p}_0)$; the initial conditions were instead chosen to better highlight the difference between the isometric and directional tempering.

As can be seen, none of the HMC trajectories have sufficient (total) energy to reach the other mode and consequently are trapped near the left mode. On the other hand, the DTHMC trajectories can easily reach the other mode with high probability. The ITHMC trajectories also have enough energy to travel through the low probability and clearly improve on HMC, but are not as successful as DTHMC in locating the other mode. In general, geometrically tempered trajectories tend to drift toward regions of lower probability as the distances to those regions are closer than to regions of higher probability under the metric of the form (3.5). Benefits of geometric tempering therefore are greater if done in particular directions of interest to limit the exploration of irrelevant regions.

Along each of the trajectories, asterisk signs are placed at $\{\boldsymbol{\theta}(t_i)\}_{i=0}^n$, where $\{t_i\}_{i=0}^n$ partitions $[0, t]$ into n equally spaced intervals. This is done to demonstrate how the velocity of a trajectory changes along its path. The tempered trajectories travel through low probability density regions in a relatively small amount of time, a property we discuss further in Section 3.4.1.

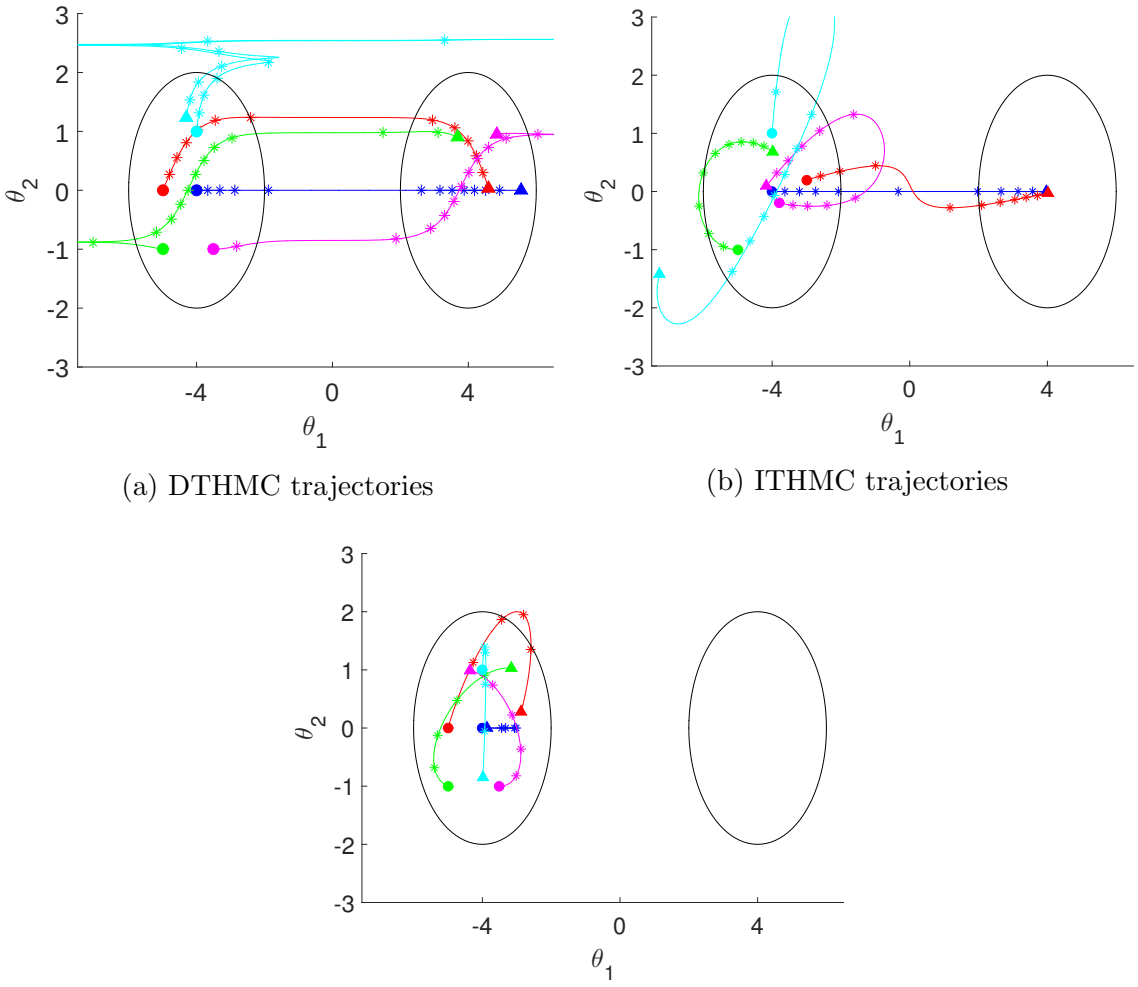
The cyan coloured DTHMC trajectory deserves some attention. The large oscillation in the tempered direction can be understood as follows in view of Theorem 3.4: a map $\boldsymbol{g} : \mathbb{R}^2 \to M$ corresponding to the DTHMC metric heavily compresses the distance along the x-axis in low probability regions. Therefore, a small oscillation of a trajectory on M manifests as a large oscillation in the original parameter space $\mathbb{R}^2$. This phenomenon does not negatively affect the mixing of DTHMC but it does increase the computational cost; see our simulation results in Section 3.5.

3.4 Reversible variable stepsize integrator for GTHMC

Until this point, we have put aside the issue that Hamiltonian dynamics in general cannot be solved exactly. The usual Störmer-Verlet scheme for approximating Hamiltonian dynamics encounters numerical stability issues in GTHMC. This is because the velocity $\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t = \boldsymbol{G}_T^{-1}(\boldsymbol{\theta})\boldsymbol{p}$ can become unboundedly large in regions of low probability. We begin this section by quantifying this phenomena and follow it up with the development of a novel reversible integrator that overcomes this shortcoming of Störmer-Verlet and enables practical applications of GTHMC. We then provide concrete examples of the integrator applied to ITHMC and DTHMC in Section 3.4.3.

3.4.1 Velocity of GTHMC trajectories

The velocity of a GTHMC trajectory grows rapidly as it enters a low probability region in which $\pi(\boldsymbol{\theta})/\left\|\pi\right\|_\infty \ll 1$ where $\left\|\pi\right\|_\infty = \max_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta})$. This is a necessary consequence of the fact that GTHMC travels through such regions without modifying the target distribution $\pi(\boldsymbol{\theta})$; a dynamics would distort a distribution if it spends as much time in low probability regions as in high probability regions. The position coordinate of a GTHMC trajectory $\boldsymbol{\theta}(t)$ travels faster and faster as $\pi(\boldsymbol{\theta}(t))$ becomes smaller, thereby spending less time in regions with lower probability. While this enables GTHMC to transition from one mode to another, this property also makes it difficult to



(a) DTHMC trajectories

(b) ITHMC trajectories

(c) HMC trajectories

FIGURE 3.1: Comparison of trajectories generated (a) with directional, (b) with isometric, and (c) without tempering. The black circles indicate a high probability density region. The circular and triangular markers indicate the start and end point of the trajectories. The star marks are placed at equal time intervals. (The time interval varies from plot to plot but is constant within each plot.)

approximate GTHMC trajectories with a fixed stepsize integrator like Störmer-Verlet.

To quantify how the velocity of a GTHMC trajectory depends on position, consider an exact (not numerically approximated) GTHMC trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ with an initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ drawn from the stationary distribution $\pi(\boldsymbol{\theta}, \boldsymbol{p})$. The energy and volume conservation property of Hamiltonian dynamics implies $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) \stackrel{d}{=} (\boldsymbol{\theta}_0, \boldsymbol{p}_0)$

and therefore $\boldsymbol{G}_T^{-1/2}(\boldsymbol{\theta}(t))\boldsymbol{p}(t) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ for all t. This suggests that the magnitude of the velocity $\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t = \boldsymbol{G}_T^{-1}(\boldsymbol{\theta})\boldsymbol{p}$ can grow as large as $\left\|\boldsymbol{G}_T(\boldsymbol{\theta}(t))^{-1/2}\right\|$ along a typical trajectory of GTHMC. Notice that, due to the constraint $\left|\boldsymbol{G}_T(\boldsymbol{\theta})^{1/2}\right| \propto \pi(\boldsymbol{\theta})^{(1-\frac{1}{T})}$, the matrix norm $\left\|\boldsymbol{G}_T(\boldsymbol{\theta})^{-1/2}\right\|$ necessarily becomes unbounded as $\pi(\boldsymbol{\theta}) \to 0$ for $T > 1$.

3.4.2 Explicit adaptive integrator with time rescaling

The discussion in Section 3.4.1 suggests that GTHMC requires a *variable stepsize* or *adaptive* integrator that adjusts stepsize locally according to the current position. Variable stepsize integrators can be interpreted as fixed stepsize integrators of a differential equation under *time rescaling*. If $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ denotes a solution of Hamilton's equations and a new time-scale s is defined via the relation $\eta(\boldsymbol{\theta})\mathrm{d}s = \mathrm{d}t$, the trajectory $(\boldsymbol{\theta}(s), \boldsymbol{p}(s))$ satisfies the following *time rescaled Hamilton's equations*:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \eta(\boldsymbol{\theta})\nabla_{\boldsymbol{p}}H(\boldsymbol{\theta}, \boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}s} = -\eta(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}H(\boldsymbol{\theta}, \boldsymbol{p}) \tag{3.6}$$

An implicit integrator similar to *adaptive Störmer-Verlet* of Huang and Leimkuhler (1997) can be used to solve (3.6) The implicit updates of (adaptive) Störmer-Verlet, however, require numerically solving for fixed points of non-linear functions and is a significant computational burden (Hairer et al., 2006).

In order to address the above issues, we develop an explicit reversible integrator with built-in local stepsize adjustment. The integrator is a generalization of the one proposed by Lan et al. (2015) based on a similar variable transformation idea. In RMHMC settings, a Hamiltonian has the form $H(\boldsymbol{\theta}, \boldsymbol{p}) = \phi(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{p}^T\boldsymbol{G}^{-1}(\boldsymbol{\theta})\boldsymbol{p}$, and (3.6) can be written as:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \eta\,\boldsymbol{G}^{-1}\boldsymbol{p}, \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}s} = -\eta\nabla_{\boldsymbol{\theta}}\phi + \frac{1}{2}\eta\,\boldsymbol{p}^T\boldsymbol{G}^{-1}(\nabla_{\boldsymbol{\theta}}\boldsymbol{G})\boldsymbol{G}^{-1}\boldsymbol{p} \tag{3.7}$$

where $\boldsymbol{u}^T(\nabla\boldsymbol{A})\boldsymbol{w}$ denotes a vector whose kth entry is $\boldsymbol{u}^T(\partial_k\boldsymbol{A})\boldsymbol{w}$ for $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{R}^d$ and a $d \times d$ matrix valued function $\boldsymbol{A}$. With an appropriately chosen time-rescaling $\eta(\boldsymbol{\theta})$,

the differential equation (3.7) is much better-behaved than the equation in the original time scale. In fact, the choice $1/\eta(\boldsymbol{\theta}) \propto \left\|\boldsymbol{G}^{-1/2}(\boldsymbol{\theta})\right\|$ stabilizes RMHMC trajectories in general as can be shown by an analysis similar to that of Section 3.4.1. We now reparametrize the differential equation (3.7) in terms of the variables $(\boldsymbol{\theta}(s), \boldsymbol{v}(s))$ where $\boldsymbol{v} = \eta\, \boldsymbol{G}^{-1}\boldsymbol{p}$. After carrying out calculations described in Supplement Section 3.C, we find that a trajectory $(\boldsymbol{\theta}(s), \boldsymbol{v}(s))$ satisfies the following differential equation:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \boldsymbol{v}, \quad \frac{\mathrm{d}v_k}{\mathrm{d}s} = -\eta^2 \left[\boldsymbol{G}^{-1}\nabla_{\boldsymbol{\theta}}\phi\right]_k + \boldsymbol{v}^T\boldsymbol{\Gamma}^k\boldsymbol{v} \quad \text{for } k = 1, \ldots, d \tag{3.8}$$

where $[\boldsymbol{w}]_k$ denotes the k-th coordinate of $\boldsymbol{w}$ and $\boldsymbol{\Gamma}^k = \boldsymbol{\Gamma}^k(\boldsymbol{\theta})$ denotes a symmetric matrix whose entries are defined as

$$\boldsymbol{\Gamma}^k_{ij} = \sum_{\ell}(\boldsymbol{G}^{-1})_{k\ell}\left[\frac{1}{2}\frac{\partial}{\partial\theta_\ell}G_{ij} - \frac{\eta}{2}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}G_{\ell j}\right) - \frac{\eta}{2}\frac{\partial}{\partial\theta_j}\left(\frac{1}{\eta}G_{\ell i}\right)\right] \tag{3.9}$$

A reversible integrator of (3.8) can be obtained by a symmetric linearly implicit scheme of Kahan (Lan et al., 2015; Sanz-Serna, 1994), which results in the following update equations:

$$\begin{aligned}
\boldsymbol{v}_{1/2} &= \left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}_0^T\boldsymbol{\Gamma}(\boldsymbol{\theta}_0)\right)^{-1}\left(\boldsymbol{v}_0 - \frac{\epsilon}{2}\eta^2(\boldsymbol{\theta}_0)\boldsymbol{G}^{-1}(\boldsymbol{\theta}_0)\nabla\phi(\boldsymbol{\theta}_0)\right) \\
\boldsymbol{\theta}_1 &= \boldsymbol{\theta}_0 + \epsilon\boldsymbol{v}_{1/2} \\
\boldsymbol{v}_1 &= \left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}_{1/2}^T\boldsymbol{\Gamma}(\boldsymbol{\theta}_1)\right)^{-1}\left(\boldsymbol{v}_{1/2} - \frac{\epsilon}{2}\eta^2(\boldsymbol{\theta}_1)\boldsymbol{G}^{-1}(\boldsymbol{\theta}_1)\nabla\phi(\boldsymbol{\theta}_1)\right)
\end{aligned} \tag{3.10}$$

where ϵ is a fixed stepsize and $\boldsymbol{v}^T\boldsymbol{\Gamma}$ denotes a matrix whose kth row corresponds to $\boldsymbol{v}^T\boldsymbol{\Gamma}^k$. The symmetry of the integrator implies that the local error is of order $O(\epsilon^3)$ i.e.

$$(\boldsymbol{\theta}_1, \boldsymbol{v}_1)(\epsilon) = \boldsymbol{F}_\epsilon(\boldsymbol{\theta}_0, \boldsymbol{v}_0) + O(\epsilon^3)$$

where $\boldsymbol{F}_\epsilon$ is the solution operator of the dynamics (3.8) (Leimkuhler and Reich, 2005; Neal, 2010). Unlike Störmer-Verlet, this integrator is not volume-preserving, therefore

the determinant of the Jacobian $\frac{\partial(\boldsymbol{\theta}_1,\boldsymbol{v}_1)}{\partial(\boldsymbol{\theta}_0,\boldsymbol{v}_0)}$ needs to be included in the calculation of the acceptance probability in RMHMC applications (see Section 3.4.4). We provide the derivation and further properties of the integrator in Supplement Section 3.C.

3.4.3 Examples: explicit adaptive integrator for ITHMC and DTHMC

We illustrate how the time rescaling of Hamiltonian dynamics and resulting explicit integrator works in practice. With a metric defined as in Section 3.3.1 for ITHMC, we have $\left\|\boldsymbol{G}^{-1/2}(\boldsymbol{\theta})\right\| = 1/\sqrt{g(\boldsymbol{\theta})}$, so we set $\eta(\boldsymbol{\theta}) = \sqrt{g(\boldsymbol{\theta})}$. In this case, the matrix $\boldsymbol{\Gamma}^k$ defined as (3.9) becomes

$$\boldsymbol{\Gamma}^k = \frac{1}{2}\frac{\partial \log g}{\partial \theta_k}\boldsymbol{I} - \frac{1}{4}\nabla_{\boldsymbol{\theta}} \log g \cdot \boldsymbol{e}_k^T - \frac{1}{4}\boldsymbol{e}_k \cdot \nabla_{\boldsymbol{\theta}}^T \log g \tag{3.11}$$

So we have

$$\boldsymbol{v}^T\boldsymbol{\Gamma} = \frac{1}{2}\nabla \log g \cdot \boldsymbol{v}^T - \frac{1}{4}\langle \boldsymbol{v}, \nabla \log g\rangle \boldsymbol{I} - \frac{1}{4}\boldsymbol{v} \cdot \nabla^T \log g \tag{3.12}$$

$$\Longrightarrow \left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}^T\boldsymbol{\Gamma}\right) = \left(1 + \frac{\epsilon}{8}\langle \boldsymbol{v}, \nabla \log g\rangle\right)\boldsymbol{I} - \frac{\epsilon}{4}\nabla \log g \cdot \boldsymbol{v}^T + \frac{\epsilon}{8}\boldsymbol{v} \cdot \nabla^T \log g$$

Since the above matrix is a rank-2 perturbation of an identity, it can be inverted in $O(d)$ using the Sherman-Morrison formula to carry out the velocity updates in (3.10):

$$\boldsymbol{v}^* = \left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}^T\boldsymbol{\Gamma}\right)^{-1}\left(\boldsymbol{v} + \frac{\epsilon}{2T}\nabla \log \pi\right)$$

The determinant $|\mathbf{D}\boldsymbol{F}_\epsilon|$ needed in the acceptance probability calculation can also be computed in $O(d)$ using the matrix determinant lemma (see (3.30) in Supplement Section 3.C for the formula of the Jacobian).

For DTHMC with a metric as in (3.5), we have $\left\|\boldsymbol{G}^{-1/2}(\boldsymbol{\theta})\right\| = 1/\sqrt{g_{\|}(\boldsymbol{\theta})}$, so we set $\eta(\boldsymbol{\theta}) = \sqrt{g_{\|}(\boldsymbol{\theta})}$. As in ITHMC, the numerical integration and determinant computation can be carried out in $O(d)$ because the matrix $\left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}^T\boldsymbol{\Gamma}\right)$ is a rank-3 perturbation of identity. The formulas for $\boldsymbol{\Gamma}^k$ and $\boldsymbol{v}^T\boldsymbol{\Gamma}$ are more complicated than those for ITHMC, however, and we refer the readers to Appendix 3.C.3 for their full expressions.

3.4.4 Variable trajectory length compressible HMC

Although the variable stepsize integrator of Section 3.4 enables an efficient and accurate approximation of otherwise unstable trajectories, the required time-rescaling of Hamiltonian dynamics destroys its volume-preserving property. As we discuss further in Section 4.2, a non-volume preserving integrator can be used to generate a proposal as long as the acceptance probability is modified to include the Jacobian (i.e. change of volume) factor (see). The extension of HMC with this generalized acceptance-rejection scheme is known as *compressible HMC* (CHMC) (Fang et al., 2014; Lan et al., 2015). In GTHMC settings, however, CHMC in general suffers from low acceptance probabilities and poor mixing. This is because Hamiltonian dynamics no longer preserves the original target distribution after time-rescaling (3.6), which means that a substantial fraction of proposals must be rejected in order for the transition kernel to preserve the target distribution.

We address this shortcoming of CHMC through *variable trajectory length CHMC* (VTL-CHMC) developed in Chapter 4. By allowing individual trajectories to have different path lengths, VTL-CHMC constructs a transition kernel that better approximates the original dynamics and has a guaranteed high acceptance probability. Section 4.3 describes the special case of VTL-CHMC used for GTHMC in detail along with the theoretical results on the advantage of VTL-CHMC over the standard CHMC. Without getting into details, for now we only mention that VTL-CHMC proposals are guaranteed to have the acceptance probability of at least $1/2$ in the small stepsize limit $\epsilon \to 0$. Empirically, in our simulations of Section 3.5, VTL-CHMC achieved the acceptance rate of $0.6 \sim 0.8$ with stepsizes reasonable for overall computational efficiency. CHMC has no such guarantee of high acceptance rate even in the small stepsize limit. An empirical comparison of CHMC and VTL-CHMC are given in Section 4.5, and VTL-CHMC demonstrates $5 \sim 10$ folds efficiency improvement over

CHMC.

3.5 Numerical results

We compare the performance of HMC and GTHMC on various multi-modal target distributions to demonstrate the advantage of GTHMC. The effect of different temperatures and tempering schemes are also illustrated.

Finding an optimal value of integration time $\tau = \epsilon L$ for HMC is known to be difficult (Neal, 2010). As a benchmark against GTHMC, therefore, we use a variant of HMC known as the No-U-Turn-Sampler (NUTS) by Hoffman and Gelman (2014) which automatically adapts the path length for individual trajectories of Hamiltonian dynamics. The use of NUTS to benchmark against GTHMC is appropriate since NUTS uses the same underlying dynamics as HMC and has been shown empirically to perform as well as optimally tuned HMC in a variety of situations. The mass matrices of ITHMC and DTHMC as in Section 3.3 degenerate to the identity when $T = 1$, so for fair comparison we used the identity mass matrix for NUTS. The stepsize ϵ was tuned using the dual-averaging algorithm of Hoffman and Gelman (2014) so that the average acceptance probability corresponds to a pre-specified value $\delta \in (0, 1)$. Theoretical and empirical studies suggest the values of $\delta \in [0.6, 0.8]$ to be optimal (Beskos et al., 2013; Hoffman and Gelman, 2014) and the values of $\delta = 0.5, 0.6, \ldots, 0.9$ were tried for each target distribution.

For ITHMC and DTHMC, the parameters ϵ and τ were tuned alternately for a few times with one of them fixed while the other is adjusted. A modified dual-averaging algorithm was used to tune ϵ to achieve an appropriate acurracy in the numerical approximation of Hamiltonian dynamics. The path length τ was tuned to maximize a normalized expected squared jumping distance (Wang et al., 2013).

Efficiency of the algorithms are compared through effective sample sizes (ESS). Following Hoffman and Gelman (2014), we summarize and normalize ESS's as

below. We first compute the ESS's of marginal mean and variance estimators for each coordinate of a target distribution and then report the minimum of these values. For the majority of posterior distributions encountered in practice, the most computationally expensive parts of the algorithms are evaluations of $\nabla_{\boldsymbol{\theta}} \log \pi(\cdot)$. ESS is therefore normalized by the number of the gradient evaluations to account for the costs of each iteration. We also report ESS per 100 MCMC samples so that the qualities of the samples can be compared to independent ones.

Some algorithms experience very slow mixing on some of the examples due to multi-modalities, and one might worry about the accuracy of ESS estimates in such situations. To address this concern, we estimate ESS's using the monotone sequence estimator of Geyer (1992) with the following modification; in estimating the lag k auto-covariance $a(k)$ of a statistic $g(\boldsymbol{\theta})$, the true mean $\mu(g) := \mathbb{E}[g(\boldsymbol{\theta})]$ is used in place of the empirical mean. This procedure leads to more reliable estimates of ESS's (Hoffman and Gelman, 2014). The expectations were computed analytically or numerically with high accuracy.

3.5.1 Bi-modal Gaussian mixture

We first compared the performance of NUTS and GTHMC on a simple bi-modal target distribution, a mixture of 2-d standard Gaussians centered at $(0, -4)$ and $(0, 4)$ with equal weights as in Figure 3.1. We ran ITHMC, DTHMC with $\gamma = .75$, and DTHMC with $\gamma = 1$ at different temperatures. DTHMC was tempered along the first coordinate. The performance of each algorithm is summarized in Table 3.1. ITHMC improves over NUTS substantially in terms of ESS, with further improvement obtained by DTHMC. Figure 3.2 compares the traceplot of the best performing NUTS ($\delta = 0.7$) and DTHMC ($T = 20, \gamma = 1$). The efficiency gain by ITHMC and DTHMC are partially offset by the increased number of numerical integration steps required to accurately simulate GTHMC trajectories, as seen in ESS per gradients. The minimum

Table 3.1: Comparison of minimum ESS at different temperatures for the 2-d bimodal target. ESS per 100 MCMC samples or per 6656 gradients evaluations are shown.

Temperature	5	10	15	20	25
ITHMC ESS per samples	0.279	0.421	0.445	0.469	0.510
DTHMC ($\gamma = .75$) ESS per samples	1.10	2.56	3.20	3.67	3.63
DTHMC ($\gamma = 1$) ESS per samples	3.91	13.0	17.9	18.2	16.4
NUTS ($\delta = .7$) ESS per samples			0.0342		
ITHMC ESS per gradients	3.37	4.90	5.11	5.27	5.80
DTHMC ($\gamma = .75$) ESS per gradients	8.60	17.6	21.3	21.4	22.1
DTHMC ($\gamma = 1$) ESS per gradients	23.0	49.8	59.4	65.3	52.2
NUTS ($\delta = .7$) ESS per gradients			1		

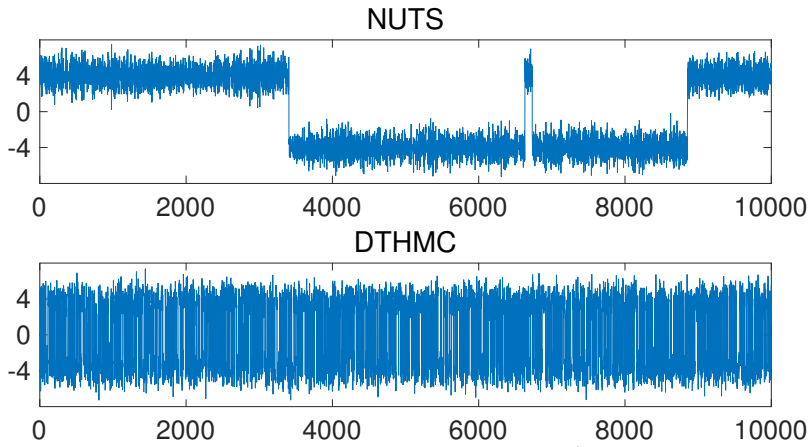


FIGURE 3.2: Traceplot of the first coordinate from 10^4 samples generated by NUTS ($\delta = 0.7$) and DTHMC ($T = 20, \gamma = 1$).

ESS came from the mean estimator along the first coordinate for all the simulations, except for DTHMC with $\gamma = 1$ and $T = 25$; in general the directions orthogonal to the tempered one are explored less efficiently by DTHMC as the parameter γ and the temperature T increases.

3.5.2 Swiss roll distribution

For a "swiss roll" target as shown in Figure 3.3, defined as a Gaussian mixture, we ran NUTS, ITHMC, and DTHMC with $\gamma = .75$. The tempering direction for DTHMC was generated uniformly from a space of unit vectors and independently at

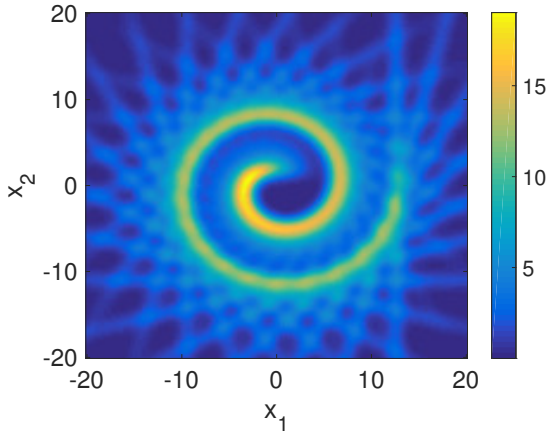


FIGURE 3.3: Plot of unnormalized swiss roll target distribution.

Table 3.2: Comparison of ESS across different temperatures for the swiss roll target. ESS per 100 MCMC samples or per 214 gradients evaluations are shown.

Temperature	5	10	15	20	25
ITHMC ESS per samples	50.4	42.1	42.4	46.8	42.6
DTHMC ($\gamma = .75$) ESS per samples	10.5	10.6	10.3	10.6	11.1
NUTS ($\delta = .8$) ESS per samples			6.48		
ITHMC ESS per gradients	1.80	1.73	1.68	1.59	1.54
DTHMC ($\gamma = .75$) ESS per gradients	0.637	0.581	0.598	0.528	0.495
NUTS ($\delta = .8$) ESS per gradients			1		

each iteration. The performance of each algorithm is summarized in Table 3.2. The potential energy barrier between the "inner" and "outer" roll is not large, so ITHMC can easily move between them even at $T = 5$. It appears that increasing temperature beyond this point is wasteful in terms of the number of gradient evaluations as the trajectories spend more time exploring the low probability region before finally arriving at the high probability region. It is possible, however, the decrease in ESS per gradients is an artifact of our tuning algorithm. The efficiency of DTHMC here is limited by the lack of preferred direction in the target distribution.

Table 3.3: Comparison of ESS along a coordinate and along the radial direction. ESS per 100 MCMC samples or per 831 gradient evaluations are shown.

	Coordinate-wise	Radial
ITHMC ($T = 5$) ESS per samples	12.7	3.28
NUTS ($\delta = 0.8$) ESS per samples	7.30	1.13
ITHMC ($T = 5$) ESS per gradients	13.1	3.43
NUTS ($\delta = 0.8$) ESS per gradients	6.43	1

3.5.3 Spherically symmetric "donut" distribution

To see how GTHMC performs in higher dimensions, we ran NUTS and ITHMC on a 25-dimensional spherically symmetric distribution defined as follows:

$$\pi(\boldsymbol{\theta}) = \sum_{i=1}^{3} \frac{1}{\sigma} \exp\left(-\frac{(\|\boldsymbol{\theta}\| - \mu_i)^2}{2\sigma^2}\right) \quad \text{where } \mu_i = i/2,\ \sigma = 0.1$$

The probabilities are therefore concentrated at the spherical shells of radius μ_i's. One may wonder if the bottleneck in this example is multi-modality or other geometric features, so we additionally report the ESS of a statistic $\|\boldsymbol{\theta}\|$ as a measure of efficiency in exploring the radial direction. The results are summarized in Table 3.3. The ESS along the radial direction are much smaller, clearly indicating the multimodality to be the bottleneck. Also clear is ITHMC's ability to better deal with the multimodality. In addition, the higher coordinate-wise ESS shows that ITHMC inherits the ability of HMC to explore a complex distribution relatively efficiently.

The temperature of ITHMC was fixed at $T = 5$ since, as in the swiss roll example, the performance did not change significantly at higher temperature. DTHMC was not tried on this example since DTHMC does not scale well to higher dimensions without localizing the Riemannian metric, which is beyond the scope of this thesis.

3.6 Discussion

This chapter presented a theoretical and practical framework for alleviating HMC's tendency to get stuck at local modes. We established the necessary condition on a Riemannian metric and studied the properties of the corresponding Hamiltonian dynamics. In addition, we developed a novel adaptive reversible integrator as well as improved acceptance-rejection mechanism to address the shortcomings of the standard Störmer-Verlet.

GTHMC clearly has room for further improvement in two aspects. First, more research effort is needed to develop better numerical integrators for RMHMC and GTHMC applications. Numerical integrators traditionally have been developed to achieve highly accurate trajectories for a long integration time, while in an RMHMC application a required integration time is usually shorter and accuracy is not so important as overall computational efficiency. Blanes et al. (2014) is one of the first attempts to develop an integrator tailor-made for HMC beyond the standard Störmer-Verlet. To our knowledge, the explicit adaptive reversible integrator for non-separable Hamiltonians presented in Section 3.4.2 is the first of its kind, and a better numerical integrator can likely be developed with increased research effort in this area. Our work on GTHMC has already prompted a new work (Okudo and Suzuki, 2016) and we expect more to come.

Second, GTHMC can benefit from a metric more specifically chosen for each multimodal target distribution rather than the generic ones considered in this chapter. ITHMC is a clear improvement over HMC, but is still not efficient in the absolute sense. In fact, it was observed that ITHMC barely performs better than HMC in higher dimensions when modes are isolated (not reported in the thesis). This is because a randomly generated trajectory is unlikely to travel in the right direction in a high dimension without encoding more information in the metric. On the other

hand, the bi-modal example in Section 3.5.1 demonstrates that GTHMC has potential to sample efficiently even from a target distribution with substantial multi-modality.

It is also worth noting that GTHMC can be combined with other (non-geometric) tempering approaches to further promote transitions among the modes in the presence of severe multi-modality. These tempering methods are meta-algorithms and in practice require an additional specification of a transition kernel to sample from tempered distributions $\propto \pi(\boldsymbol{\theta})^{1/T_i}$ where the sequence of temperatures $1 = T_1 < T_2 < \ldots < T_M$ must also be specified by a user (Earl and Deem, 2005; Geyer and Thompson, 1995; Marinari and Parisi, 1992). The largest temperature T_M must be large enough that the transition kernel can easily induce transitions from one mode to another. Increasing T_M however comes at the cost of increasing the computational time in relating the tempered distribution $\propto \pi(\boldsymbol{\theta})^{1/T_M}$ back to the original distribution. For this reason, even within the tempering algorithms it is desirable to use a transition kernel less prone to be stuck at local modes so that the temperatures do not need to be unnecessarily large. GTHMC can provide such a transition kernel, inheriting otherwise desirable characteristics of HMC.

Appendix for Chapter 3

3.A Proof of Theorem 3.5

For the purpose of the proof, we consider the Jacobians $\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$ and $\mathbf{D}(\boldsymbol{g}^{-1})_{\tilde{\boldsymbol{\theta}}}$ as bijective maps between $\mathbb{R}^d$ and $T_{\tilde{\boldsymbol{\theta}}}M$ rather than non-square matrices, so that the inverse $(\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1} = \mathbf{D}(\boldsymbol{g}^{-1})_{\tilde{\boldsymbol{\theta}}}$ makes sense. One may think of these Jacobians as a square matrix with respect to some basis for $T_{\tilde{\boldsymbol{\theta}}}M$. One can easily verify that the calculations in the proof are independent of choice of basis. Additionally, for notational convenience we suppress the superscript M from the gradient ∇^M for a function defined on a manifold M.

Proof. By direct computation, we will prove the equivalence between the differential equations for $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ and Hamilton's equations with the Hamiltonian $H(\boldsymbol{\theta}, \boldsymbol{p}) = -\log \pi(\boldsymbol{\theta}) + \frac{1}{2}\log|\boldsymbol{G}(\boldsymbol{\theta})| + \frac{1}{2}\boldsymbol{p}^\intercal \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p}$. Recalling the relations $\boldsymbol{\theta} = \boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})$, $\frac{\mathrm{d}\tilde{\boldsymbol{\theta}}}{\mathrm{d}t} = \tilde{\boldsymbol{p}}$, and $\boldsymbol{p} = \mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^\intercal \tilde{\boldsymbol{p}}$, we find

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \mathbf{D}(\boldsymbol{g}^{-1})_{\tilde{\boldsymbol{\theta}}}\frac{\mathrm{d}\tilde{\boldsymbol{\theta}}}{\mathrm{d}t} = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1}\tilde{\boldsymbol{p}} = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1}(\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-T}\boldsymbol{p} = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p} = \nabla_{\boldsymbol{p}}H(\boldsymbol{\theta}, \boldsymbol{p})$$

The computation for $\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}$ is a bit more involved. First note that

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left\langle \frac{\partial \boldsymbol{g}}{\partial \theta_i}(\boldsymbol{\theta}), \tilde{\boldsymbol{p}} \right\rangle = \left\langle \left(\mathbf{D}\frac{\partial \boldsymbol{g}}{\partial \theta_i}\right)_{\boldsymbol{\theta}} \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}, \tilde{\boldsymbol{p}} \right\rangle + \left\langle \frac{\partial \boldsymbol{g}}{\partial \theta_i}, \nabla_{\tilde{\boldsymbol{\theta}}} \log \tilde{\pi} \right\rangle \tag{3.13}$$

The first term in the last equation will simplify as follows:

$$\begin{aligned}\left\langle \left(\mathbf{D}\frac{\partial \boldsymbol{g}}{\partial \theta_i}\right)_{\boldsymbol{\theta}} \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}, \tilde{\boldsymbol{p}} \right\rangle &= \left\langle \left(\frac{\partial}{\partial \theta_i}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}\right)\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p}, \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p} \right\rangle \\ &= \frac{1}{2}\boldsymbol{p}^{\intercal}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\frac{\partial}{\partial \theta_i}\left(\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}\right)\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p} \\ &= \frac{1}{2}\boldsymbol{p}^{\intercal}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\frac{\partial \boldsymbol{G}(\boldsymbol{\theta})}{\partial \theta_i}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p} \end{aligned} \tag{3.14}$$

We can simplify the second term in (3.13) using Lemma 3.6 follows:

$$\begin{aligned}\left\langle \frac{\partial \boldsymbol{g}}{\partial \theta_i}, \nabla_{\tilde{\boldsymbol{\theta}}} \log \tilde{\pi} \right\rangle &= \left\langle (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta_i}, \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}) - \frac{1}{2}\nabla_{\boldsymbol{\theta}} \log |\boldsymbol{G}(\boldsymbol{\theta})| \right\rangle \\ &= \frac{\partial}{\partial \theta_i}\log \pi(\boldsymbol{\theta}) - \frac{1}{2}\frac{\partial}{\partial \theta_i}\log |\boldsymbol{G}(\boldsymbol{\theta})| \end{aligned} \tag{3.15}$$

From (3.13), (3.14), and (3.15), we conclude that

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \frac{\partial}{\partial \theta_i}\log \pi(\boldsymbol{\theta}) - \frac{1}{2}\frac{\partial}{\partial \theta_i}\log |\boldsymbol{G}(\boldsymbol{\theta})| + \frac{1}{2}\boldsymbol{p}^{\intercal}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\frac{\partial \boldsymbol{G}(\boldsymbol{\theta})}{\partial \theta_i}\boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p} = -\frac{\partial}{\partial \theta_i}H(\boldsymbol{\theta}, \boldsymbol{p})$$

□

Lemma 3.6. *If $\pi(\boldsymbol{\theta})$ is a pdf on $\mathbb{R}^d$ and $\tilde{\pi}(\tilde{\boldsymbol{\theta}})$ is a pdf on a manifold M induced by the bijective map $\boldsymbol{g} : \mathbb{R}^d \to M$, then*

$$\nabla_{\tilde{\boldsymbol{\theta}}} \log \tilde{\pi}(\tilde{\boldsymbol{\theta}}) = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-T}\left(\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}) - \frac{1}{2}\nabla_{\boldsymbol{\theta}} \log |\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}|\right)$$

Proof. By the change of variable formula, we have

$$\log \tilde{\pi}\left(\tilde{\boldsymbol{\theta}}\right) = -\frac{1}{2}\log |\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}| + \log \pi(\boldsymbol{\theta}) \tag{3.16}$$

Now we only need to observe that the following equality holds for any scalar-valued function $f(\boldsymbol{\theta})$ on $\mathbb{R}^d$:

$$\nabla_{\tilde{\boldsymbol{\theta}}} f \circ \boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}) = \mathbf{D}(\boldsymbol{g}^{-1})_{\tilde{\boldsymbol{\theta}}}^{\intercal}\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-T}\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$$

□

3.B Geometric theory of manifold Langevin algorithm

Riemann manifold Metropolis adjusted Langevin algorithm (MMALA) is the Langevin dynamics analogue of RMHMC and described by Girolami and Calderhead (2011) as a potentially useful alternative to RMHMC. Given a metric $\boldsymbol{G}(\boldsymbol{\theta})$, MMALA generates a proposal by approximating the following SDE for $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$:

$$\begin{aligned} \mathrm{d}\theta_i = \frac{1}{2}\left\{\boldsymbol{G}^{-1}(\boldsymbol{\theta})\left(\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}) - \frac{1}{2}\nabla_{\boldsymbol{\theta}} \log |\boldsymbol{G}(\boldsymbol{\theta})|\right)\right\}_i \mathrm{d}t \\ + \left\{\boldsymbol{G}^{-1/2}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{B}_t\right\}_i + \frac{1}{2}|\boldsymbol{G}(\boldsymbol{\theta})|^{-1/2}\sum_{j=1}^{d}\frac{\partial}{\partial\theta_j}\left\{|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2}\left(\boldsymbol{G}^{-1}(\boldsymbol{\theta})\right)_{ij}\right\}\mathrm{d}t \end{aligned} \tag{3.17}$$

where $\boldsymbol{B}(t)$ is a Brownian motion. Note that the above equation differs from the one originally presented in Girolami and Calderhead (2011) which contains a transcription error (Xifara et al., 2014).

Theorem 3.7 below is a Langevin dynamics analogue of Theorem 3.4, establishing a geometric connection between the standard Langevin dynamics (3.18) and the SDE (3.17). Due to the stochastic nature of Langevin dyanmics, defining it on a manifold through the language of extrinsic geometry turns out to be far more challenging than doing the same for Hamiltonian dynamics (Rogers and Williams, 2000). For simplicity, therefore, Theorem 3.7 invokes a stronger assumption than Theorem 3.4 and assumes that the reparametrization $\boldsymbol{g}$ is a map between subsets of $\mathbb{R}^d$.

Theorem 3.7 (Manifold Langevin as reparametrization)**.** *Given a pdf $\pi(\boldsymbol{\theta})$ on $\mathbb{R}^d$, let $\tilde{\pi}$ denote the pdf on a domain $\Omega \subset \mathbb{R}^d$ induced by a smooth bijection $\boldsymbol{g} : \mathbb{R}^d \to \Omega$. For the initial condition $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ and $\tilde{\boldsymbol{\theta}}_0 = g(\boldsymbol{\theta}_0)$, let $\tilde{\boldsymbol{\theta}}(t)$ denote a weak solution of the SDE*

$$d\tilde{\boldsymbol{\theta}} = \frac{1}{2}\nabla_{\tilde{\boldsymbol{\theta}}} \log \tilde{\pi}(\tilde{\boldsymbol{\theta}})dt + d\tilde{\boldsymbol{B}}(t) \tag{3.18}$$

where $\tilde{\boldsymbol{B}}(t)$ *is a Brownian motion. Then the stochastic process* $\boldsymbol{\theta}(t) = \boldsymbol{g}^{-1}\big(\tilde{\boldsymbol{\theta}}(t)\big)$ *is a weak solution of the SDE* (3.17) *with* $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$.

Proof. Let $\tilde{\boldsymbol{\theta}}(t)$ be a solution of the SDE (3.18). By Ito's lemma, the stochastic process $\boldsymbol{\theta}(t) = \boldsymbol{g}^{-1}\big(\tilde{\boldsymbol{\theta}}(t)\big)$ solves the following SDE in a weak sense:

$$\mathrm{d}\boldsymbol{\theta}(t) = \frac{1}{2}\mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-1}\nabla_{\tilde{\boldsymbol{\theta}}}\log\tilde{\pi}(\tilde{\boldsymbol{\theta}}) + \mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-1}\mathrm{d}\boldsymbol{B}(t) + \frac{1}{2}\Delta_{\tilde{\boldsymbol{\theta}}}\,\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})\mathrm{d}t \tag{3.19}$$

where $\boldsymbol{B}(t)$ is a Brownian motion and $\Delta_{\tilde{\boldsymbol{\theta}}} = \sum_i \partial^2/\partial\tilde{\theta}_i^2$ is the Laplacian. Since $\boldsymbol{G}^{-1}(\boldsymbol{\theta}) = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1} = \mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-T}$, we have

$$\boldsymbol{G}^{-1/2}(\boldsymbol{\theta}(t))\,(\boldsymbol{B}(t+\epsilon) - \boldsymbol{B}(t)) \overset{d}{=} \mathbf{D}\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}(t))\,(\boldsymbol{B}(t+\epsilon) - \boldsymbol{B}(t)) \tag{3.20}$$

and the term $\mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-1}\mathrm{d}\boldsymbol{B}(t)$ in (3.19) can equivalently be written as $\boldsymbol{G}^{-1/2}\big(\boldsymbol{\theta}\big)\mathrm{d}\boldsymbol{B}(t)$. Also rewriting the term $\nabla_{\tilde{\boldsymbol{\theta}}}\log\tilde{\pi}(\tilde{\boldsymbol{\theta}})$ using Lemma 3.6, the SDE (3.19) can expressed as

$$\mathrm{d}\boldsymbol{\theta} = \frac{1}{2}\boldsymbol{G}^{-1}(\boldsymbol{\theta})\left(\nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{\theta}) - \frac{1}{2}\nabla_{\boldsymbol{\theta}}\log|\boldsymbol{G}(\boldsymbol{\theta})|\right) + \boldsymbol{G}^{-1/2}\big(\boldsymbol{\theta}\big)\mathrm{d}\boldsymbol{B}(t) + \frac{1}{2}\Delta_{\tilde{\boldsymbol{\theta}}}\,\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})\mathrm{d}t \tag{3.21}$$

To express the term $\Delta_{\tilde{\boldsymbol{\theta}}}\,\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})$ in terms of $\boldsymbol{\theta}$, note that

$$\nabla_{\tilde{\boldsymbol{\theta}}}\left\{\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})\right\}_i = \left(\boldsymbol{e}_i^{\intercal}\,\mathbf{D}\boldsymbol{g}_{\tilde{\boldsymbol{\theta}}}^{-1}\right)^{\intercal} = (\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-T}\boldsymbol{e}_i$$

Substituting this to Lemma 3.8, we conclude that

$$\nabla_{\tilde{\boldsymbol{\theta}}}\cdot\nabla_{\tilde{\boldsymbol{\theta}}}\left\{\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}})\right\}_i = |\boldsymbol{G}(\boldsymbol{\theta})|^{-1/2}\sum_{j=1}^{d}\frac{\partial}{\partial\theta_j}\left\{|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2}\left(\boldsymbol{G}^{-1}(\boldsymbol{\theta})\right)_{ij}\right\}\mathrm{d}t \qquad \square$$

Lemma 3.8. *If* $\boldsymbol{g}:\boldsymbol{\theta}\to\tilde{\boldsymbol{\theta}}$ *is a smooth bijection between subsets of* $\mathbb{R}^d$ *and* $\boldsymbol{v}(\tilde{\boldsymbol{\theta}})$ *is a vector-valued function, then*

$$\nabla_{\tilde{\boldsymbol{\theta}}}\cdot\boldsymbol{v}(\tilde{\boldsymbol{\theta}}) = |\boldsymbol{G}(\boldsymbol{\theta})|^{-1/2}\,\nabla_{\boldsymbol{\theta}}\cdot\left\{|\boldsymbol{G}(\boldsymbol{\theta})|^{1/2}(\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}})^{-1}\boldsymbol{v}(\boldsymbol{\theta})\right\} \tag{3.22}$$

where $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^{\intercal}\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$, $\boldsymbol{v}(\boldsymbol{\theta}) := \boldsymbol{v} \circ \boldsymbol{g}(\boldsymbol{\theta})$, *and* $\nabla_{\boldsymbol{\theta}} \cdot = \sum_i \partial/\partial\theta_i$ *is the divergence operator.*

Proof. The proof only requires elementary calculus, but the computation is lengthy, involved and hence is omitted here. The details can be found in, for example, Chapter 3 of Leonhardt and Philbin (2010). □

3.C Explicit adaptive integrator: further details

Here we provide further details on the derivation and the properties of the explicit adaptive integrator described in Section 3.4.2.

3.C.1 Derivation of Equation (3.8)

We first show how one can derive the differential equation (3.8) for the parameters $(\boldsymbol{\theta}, \boldsymbol{v})$ from (3.7). Similar calculations in the case $\eta(\boldsymbol{\theta}) \equiv 1$ are carried out in Lan et al. (2015) and Fang et al. (2014). Letting $\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{p}) = (\boldsymbol{\theta}, \eta(\boldsymbol{\theta})\boldsymbol{G}^{-1}(\boldsymbol{\theta})\boldsymbol{p})$ denote the change of variable from $(\boldsymbol{\theta}, \boldsymbol{p})$ to $(\boldsymbol{\theta}, \boldsymbol{v})$, we have

$$\frac{\mathrm{d}}{\mathrm{d}s}(\boldsymbol{\theta}(s), \boldsymbol{v}(s)) = \mathbf{D}\boldsymbol{h}(\boldsymbol{\theta}(s), \boldsymbol{p}(s))\frac{\mathrm{d}}{\mathrm{d}s}(\boldsymbol{\theta}(s), \boldsymbol{p}(s)) \tag{3.23}$$

It is not difficult to show that the Jacobian $\mathbf{D}\boldsymbol{h}$ is given in terms of the variable $(\boldsymbol{\theta}, \boldsymbol{v})$ as:

$$\mathbf{D}\boldsymbol{h} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ -\eta\boldsymbol{G}^{-1}\left(\sum_i \frac{\partial}{\partial\theta_i}(\frac{1}{\eta}\boldsymbol{G})\boldsymbol{v}\boldsymbol{e}_k^{\intercal}\right) & \eta\boldsymbol{G}^{-1} \end{bmatrix} \tag{3.24}$$

By plugging (3.24) and (3.23) into the differential equation (3.7) for $(\boldsymbol{\theta}, \boldsymbol{p})$, we obtain

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \boldsymbol{v}, \quad \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = -\eta^2\boldsymbol{G}^{-1}\nabla_{\boldsymbol{\theta}}\phi - \eta\sum_{i=1}^{d} v_i\boldsymbol{G}^{-1}\frac{\partial}{\partial\theta_i}\Big(\frac{1}{\eta}\boldsymbol{G}\Big)\boldsymbol{v} + \frac{1}{2}\boldsymbol{G}^{-1}\boldsymbol{v}^{\intercal}(\nabla_{\boldsymbol{\theta}}\boldsymbol{G})\boldsymbol{v} \tag{3.25}$$

With straightforward algebra, the expression for $\mathrm{d}\boldsymbol{v}/\mathrm{d}s$ can be re-written as:

$$\begin{aligned} \frac{\mathrm{d}v_k}{\mathrm{d}s} &= -\eta^2 \left[\boldsymbol{G}^{-1}\nabla_{\boldsymbol{\theta}}\phi\right]_k + \boldsymbol{v}^\intercal \boldsymbol{\Gamma}^k \boldsymbol{v} \\ \text{where } \boldsymbol{\Gamma}^k_{ij} &= \sum_\ell (\boldsymbol{G}^{-1})_{k\ell}\left[\frac{1}{2}\frac{\partial}{\partial\theta_\ell}G_{ij} - \eta\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}G_{\ell j}\right)\right] \end{aligned} \tag{3.26}$$

Since $\boldsymbol{v}^\intercal\boldsymbol{\Gamma}^k\boldsymbol{v} = \boldsymbol{v}^\intercal(\boldsymbol{\Gamma}^k)^\intercal\boldsymbol{v}$, we can replace $\boldsymbol{\Gamma}^k$ with its symmetrization $\frac{1}{2}(\boldsymbol{\Gamma}^k + (\boldsymbol{\Gamma}^k)^\intercal)$ without changing the equation (3.26). So we re-define $\boldsymbol{\Gamma}^k$ to be a matrix such that

$$\boldsymbol{\Gamma}^k_{ij} = \sum_\ell (\boldsymbol{G}^{-1})_{k\ell}\left[\frac{1}{2}\frac{\partial}{\partial\theta_\ell}G_{ij} - \frac{\eta}{2}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}G_{\ell j}\right) - \frac{\eta}{2}\frac{\partial}{\partial\theta_j}\left(\frac{1}{\eta}G_{\ell i}\right)\right]$$

Although the symmetrization of $\boldsymbol{\Gamma}^k$ does not alter the differential equation at all, it will guarantee $(\boldsymbol{v}^*)^\intercal\boldsymbol{\Gamma}^k\boldsymbol{v} = \boldsymbol{v}^\intercal\boldsymbol{\Gamma}^k\boldsymbol{v}^*$ for all $\boldsymbol{v}$ and $\boldsymbol{v}^*$ — a crucial property in ensuring the reversibility of our explicit adaptive integrator. Finally, if we let $\boldsymbol{v}^\intercal\boldsymbol{\Gamma}$ denote a matrix whose k-th row is given by $\boldsymbol{v}^\intercal\boldsymbol{\Gamma}^k$, we can express the differential equation (3.25) in the following form, which agrees with (3.8):

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \boldsymbol{v}, \quad \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = -\eta^2\boldsymbol{G}^{-1}\nabla\phi + \boldsymbol{v}^\intercal\boldsymbol{\Gamma}\boldsymbol{v} \tag{3.27}$$

3.C.2 Reversible explicit discretization

We now describe how to obtain the explicit reversible integrator (3.10) of the differential equation (3.27). We also derive the formula for the Jacobian of the integrator, which is needed to calculate the acceptance probability of the variable-length trajectory CHMC algorithm in Section 3.4.4. A reversible explicit update $\boldsymbol{v} \to \boldsymbol{v}^*$ is obtained by the following discretization based on a linearly implicit scheme of Kahan (Lan et al., 2015; Sanz-Serna, 1994):

$$\frac{\boldsymbol{v}^* - \boldsymbol{v}}{\epsilon} = -\eta^2\boldsymbol{G}^{-1}\nabla_{\boldsymbol{\theta}}\phi + \boldsymbol{v}^\intercal\boldsymbol{\Gamma}\boldsymbol{v}^* \tag{3.28}$$

$$\iff \boldsymbol{v}^* = (\boldsymbol{I} - \epsilon\boldsymbol{v}^\intercal\boldsymbol{\Gamma})^{-1}\left(\boldsymbol{v} - \epsilon\eta^2\boldsymbol{G}^{-1}\nabla_{\boldsymbol{\theta}}\phi\right) \tag{3.29}$$

Now let $\boldsymbol{F}_{\boldsymbol{v},\epsilon}$ denote the map $\boldsymbol{F}_{\boldsymbol{v},\epsilon}(\boldsymbol{\theta},\boldsymbol{v}) = (\boldsymbol{\theta},\boldsymbol{v}^*)$ corresponding to the update equation (3.29). Note that $\boldsymbol{F}_{\boldsymbol{v},\epsilon}$ is reversible thanks to the symmetry $\boldsymbol{v}^\intercal\boldsymbol{\Gamma}\boldsymbol{v}^* = (\boldsymbol{v}^*)^\intercal\boldsymbol{\Gamma}\boldsymbol{v}$. The Jacobian of the map $\boldsymbol{v} \to \boldsymbol{v}^*$ is obtained by differentiating Equation (3.28) implicitly in $\boldsymbol{v}$:

$$\begin{aligned} &\frac{\frac{\partial \boldsymbol{v}^*}{\partial \boldsymbol{v}} - \boldsymbol{I}}{\epsilon} = \boldsymbol{v}^\intercal\boldsymbol{\Gamma}\frac{\partial \boldsymbol{v}^*}{\partial \boldsymbol{v}} + (\boldsymbol{v}^*)^\intercal\boldsymbol{\Gamma} \\ \iff &\frac{\partial \boldsymbol{v}^*}{\partial \boldsymbol{v}} = \left(\boldsymbol{I} - \frac{\epsilon}{2}\boldsymbol{v}^\intercal\boldsymbol{\Gamma}\right)^{-1}\left(\boldsymbol{I} + \frac{\epsilon}{2}(\boldsymbol{v}^*)^\intercal\boldsymbol{\Gamma}\right) \end{aligned} \tag{3.30}$$

A reversible explicit update for $\boldsymbol{\theta}$ is given by a map $\boldsymbol{F}_{\boldsymbol{\theta},\epsilon}(\boldsymbol{\theta},\boldsymbol{v}) = (\boldsymbol{\theta} + \epsilon\boldsymbol{v}, \boldsymbol{v})$, which is obviously reversible and volume preserving. The integrator (3.10) is obtained by the composition $\boldsymbol{F}_{\boldsymbol{v},\epsilon/2} \circ \boldsymbol{F}_{\boldsymbol{\theta},\epsilon} \circ \boldsymbol{F}_{\boldsymbol{v},\epsilon/2}$, which is reversible and explicit because both $\boldsymbol{F}_{\boldsymbol{v},\epsilon/2}$ and $\boldsymbol{F}_{\boldsymbol{\theta},\epsilon}$ are.

3.C.3 Derivation of explicit adaptive integrator for DTHMC

Here we derive the necessary formulas to carry out an efficient implementation of the integrator (3.10) in DTHMC settings. In particular, we show how to simplify the formula of $1 - \frac{\epsilon}{2}\boldsymbol{v}^\intercal\boldsymbol{\Gamma}$; the rest of the quantities in (3.10) are relatively straightforward to compute. To find a formula for the matrix $\boldsymbol{\Gamma}^k$ as defined in (3.9), we start by computing the last two terms of $\mathrm{d}\boldsymbol{v}/\mathrm{d}s$ in (3.25) namely the term $\frac{1}{2}\boldsymbol{G}^{-1}\boldsymbol{v}^\intercal(\nabla_{\boldsymbol{\theta}}\boldsymbol{G})\boldsymbol{v}$ and $\eta\sum_{i=1}^d v_i\boldsymbol{G}^{-1}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}$. Observe that

$$\begin{aligned} &\boldsymbol{v}^\intercal\partial_{\theta_i}\boldsymbol{G}\,\boldsymbol{v} = \langle\boldsymbol{u},\boldsymbol{v}\rangle^2\,\partial_{\theta_i}g_{\parallel} - \langle\boldsymbol{u},\boldsymbol{v}\rangle^2\,\partial_{\theta_i}g_{\perp} + \|\boldsymbol{v}\|^2\,\partial_{\theta_i}g_{\perp} \\ \implies &\boldsymbol{v}^\intercal\nabla\boldsymbol{G}\,\boldsymbol{v} = \langle\boldsymbol{u},\boldsymbol{v}\rangle^2\,(\nabla g_{\parallel} - \nabla g_{\perp}) + \|\boldsymbol{v}\|^2\,\nabla g_{\perp} \\ \implies &\boldsymbol{G}^{-1}\boldsymbol{v}^\intercal\nabla\boldsymbol{G}\,\boldsymbol{v} = \langle\boldsymbol{u},\boldsymbol{v}\rangle^2\left(\left(\frac{1}{g_{\parallel}} - \frac{1}{g_{\perp}}\right)\langle\nabla g_{\parallel} - \nabla g_{\perp}, \boldsymbol{u}\rangle\,\boldsymbol{u} + \frac{1}{g_{\perp}}(\nabla g_{\parallel} - \nabla g_{\perp})\right) \\ &\qquad + \|\boldsymbol{v}\|^2\left(\left(\frac{1}{g_{\parallel}} - \frac{1}{g_{\perp}}\right)\langle\nabla g_{\perp}, \boldsymbol{u}\rangle\,\boldsymbol{u} + \frac{1}{g_{\perp}}\nabla g_{\perp}\right) \end{aligned}$$

$$\implies \left(\frac{1}{2}\boldsymbol{G}^{-1}\boldsymbol{v}^\intercal\nabla\boldsymbol{G}\,\boldsymbol{v}\right)_k =$$

$$\boldsymbol{v}^\intercal\left[\left(\left(\frac{1}{g_\parallel}-\frac{1}{g_\perp}\right)\left\langle\nabla g_\parallel-\nabla g_\perp,\boldsymbol{u}\right\rangle u_k+\frac{1}{g_\perp}(\partial_{\theta_k}g_\parallel-\partial_{\theta_k}g_\perp)\right)\frac{1}{2}\boldsymbol{u}\boldsymbol{u}^\intercal\right]\boldsymbol{v}$$

$$+\boldsymbol{v}^\intercal\left[\left(\left(\frac{1}{g_\parallel}-\frac{1}{g_\perp}\right)\left\langle\nabla g_\perp,\boldsymbol{u}\right\rangle u_k+\frac{1}{g_\perp}\partial_{\theta_k}g_\perp\right)\frac{1}{2}\boldsymbol{I}\right]\boldsymbol{v}$$

For the other term, we have

$$\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)=\left(\frac{\partial}{\partial\theta_i}\frac{1}{\eta}\right)\boldsymbol{G}+\frac{1}{\eta}\left(\left(\partial_{\theta_i}g_\parallel-\partial_{\theta_i}g_\perp\right)\boldsymbol{u}\boldsymbol{u}^\intercal+\partial_{\theta_i}g_\perp\,\boldsymbol{I}\right)$$

$$\implies\quad \eta\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}=\eta\left(\frac{\partial}{\partial\theta_i}\frac{1}{\eta}\right)\boldsymbol{G}\boldsymbol{v}+\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\left(\partial_{\theta_i}g_\parallel-\partial_{\theta_i}g_\perp\right)\boldsymbol{u}+\partial_{\theta_i}g_\perp\,\boldsymbol{v}$$

$$\implies\quad \eta\,\boldsymbol{G}^{-1}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}=\eta\left(\frac{\partial}{\partial\theta_i}\frac{1}{\eta}\right)\boldsymbol{v}+\frac{1}{g_\parallel}\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\left(\partial_{\theta_i}g_\parallel-\partial_{\theta_i}g_\perp\right)\boldsymbol{u}$$

$$+\left(\frac{1}{g_\parallel}-\frac{1}{g_\perp}\right)\partial_{\theta_i}g_\perp\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\boldsymbol{u}+\frac{1}{g_\perp}\partial_{\theta_i}g_\perp\,\boldsymbol{v}$$

$$=\eta\left(\frac{\partial}{\partial\theta_i}\frac{1}{\eta}\right)\boldsymbol{v}+\frac{1}{g_\parallel}\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\left(\partial_{\theta_i}g_\parallel\right)\boldsymbol{u}$$

$$-\frac{1}{g_\perp}\partial_{\theta_i}g_\perp\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\boldsymbol{u}+\frac{1}{g_\perp}\partial_{\theta_i}g_\perp\,\boldsymbol{v}$$

$$\implies \eta\sum_i v_i\boldsymbol{G}^{-1}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}=$$

$$\eta\left\langle\boldsymbol{v},\nabla\frac{1}{\eta}\right\rangle\boldsymbol{v}+\left\langle\boldsymbol{u},\boldsymbol{v}\right\rangle\left\langle\boldsymbol{v},\nabla(\log g_\parallel-\log g_\perp)\right\rangle\boldsymbol{u}+\left\langle\boldsymbol{v},\nabla\log g_\perp\right\rangle\boldsymbol{v}$$

$$\implies\left(-\eta\sum_i v_i\boldsymbol{G}^{-1}\frac{\partial}{\partial\theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}\right)_k=$$

$$\boldsymbol{v}^\intercal\left[-\eta\nabla\frac{1}{\eta}\cdot\boldsymbol{e}_k^\intercal-\boldsymbol{u}\cdot\nabla^\intercal(\log g_\parallel-\log g_\perp)\,u_k-\nabla\log g_\perp\cdot\boldsymbol{e}_k^\intercal\right]\boldsymbol{v}$$

Since $\eta = \sqrt{g_\parallel}$, we have $\eta \nabla 1/\eta = -\frac{1}{2}\nabla \log g_\parallel$ and therefore

$$\left(-\eta \sum_i v_i \boldsymbol{G}^{-1} \frac{\partial}{\partial \theta_i}\left(\frac{1}{\eta}\boldsymbol{G}\right)\boldsymbol{v}\right)_k$$

$$= \boldsymbol{v}^\intercal \left[\frac{1}{2}(\nabla \log g_\parallel - 2\nabla \log g_\perp)\cdot \boldsymbol{e}_k^\intercal - \boldsymbol{u}\cdot \nabla^\intercal (\log g_\parallel - \log g_\perp)\, u_k\right]\boldsymbol{v}$$

Thus the (symmetrized) matrix $\boldsymbol{\Gamma}^k$ must be given by

$$\begin{aligned}
\boldsymbol{\Gamma}^k = {} & \frac{1}{2}\left(\left(\frac{1}{g_\parallel} - \frac{1}{g_\perp}\right)\big\langle \nabla g_\parallel - \nabla g_\perp, \boldsymbol{u}\big\rangle\, u_k + \frac{1}{g_\perp}(\partial_{\theta_k} g_\parallel - \partial_{\theta_k} g_\perp)\right)\boldsymbol{u}\boldsymbol{u}^\intercal \\
& + \frac{1}{2}\left(\left(\frac{1}{g_\parallel} - \frac{1}{g_\perp}\right)\langle \nabla g_\perp, \boldsymbol{u}\rangle\, u_k + \frac{1}{g_\perp}\partial_{\theta_k} g_\perp\right)\boldsymbol{I} \\
& + \frac{1}{4}(\nabla \log g_\parallel - 2\nabla \log g_\perp)\cdot \boldsymbol{e}_k^\intercal + \frac{1}{4}\boldsymbol{e}_k \cdot (\nabla \log g_\parallel - 2\nabla \log g_\perp)^\intercal \\
& - \frac{1}{2}\boldsymbol{u}\cdot \nabla^\intercal (\log g_\parallel - \log g_\perp)\, u_k - \frac{1}{2}\nabla(\log g_\parallel - \log g_\perp)\cdot \boldsymbol{u}^\intercal\, u_k
\end{aligned}$$

From this formula it easily follows that

$$\begin{aligned}
\boldsymbol{v}^\intercal \boldsymbol{\Gamma} = {} & \frac{1}{2}\left(\frac{1}{g_\parallel} - \frac{1}{g_\perp}\right)\big\langle \nabla g_\parallel - \nabla g_\perp, \boldsymbol{u}\big\rangle \langle \boldsymbol{v}, \boldsymbol{u}\rangle\, \boldsymbol{u}\boldsymbol{u}^\intercal + \frac{1}{2h}\langle \boldsymbol{v}, \boldsymbol{u}\rangle (\nabla g_\parallel - \nabla g_\perp)\cdot \boldsymbol{u}^\intercal \\
& + \frac{1}{2}\left(\frac{1}{g_\parallel} - \frac{1}{g_\perp}\right)\langle \nabla g_\perp, \boldsymbol{u}\rangle\, \boldsymbol{u}\boldsymbol{v}^\intercal + \frac{1}{2}\nabla \log g_\perp \cdot \boldsymbol{v}^\intercal \\
& + \frac{1}{4}\big\langle \boldsymbol{v}, \nabla \log g_\parallel - 2\nabla \log g_\perp\big\rangle \boldsymbol{I} + \frac{1}{4}\boldsymbol{v}\cdot (\nabla \log g_\parallel - 2\nabla \log g_\perp)^\intercal \\
& - \frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u}\rangle\, \boldsymbol{u}\cdot \nabla^\intercal (\log g_\parallel - \log g_\perp) - \frac{1}{2}\big\langle \boldsymbol{v}, \nabla(\log g_\parallel - \log g_\perp)\big\rangle\, \boldsymbol{u}\boldsymbol{u}^\intercal \\
= {} & \frac{1}{2}\left(1 - \frac{g_\parallel}{g_\perp}\right)\big\langle \nabla \log g_\parallel, \boldsymbol{u}\big\rangle \langle \boldsymbol{v}, \boldsymbol{u}\rangle\, \boldsymbol{u}\boldsymbol{u}^\intercal + \frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u}\rangle \left(\frac{g_\parallel}{g_\perp}\nabla \log g_\parallel - \nabla \log g_\perp\right)\cdot \boldsymbol{u}^\intercal \\
& + \frac{1}{2}\left(\frac{g_\perp}{g_\parallel} - 1\right)\langle \nabla \log g_\perp, \boldsymbol{u}\rangle\, \boldsymbol{u}\cdot (\boldsymbol{v}^\intercal - \langle \boldsymbol{v}, \boldsymbol{u}\rangle\, \boldsymbol{u}^\intercal) + \frac{1}{2}\nabla \log g_\perp \cdot \boldsymbol{v}^\intercal \\
& + \frac{1}{4}\big\langle \boldsymbol{v}, \nabla \log g_\parallel - 2\nabla \log g_\perp\big\rangle \boldsymbol{I} + \frac{1}{4}\boldsymbol{v}\cdot (\nabla \log g_\parallel - 2\nabla \log g_\perp)^\intercal \\
& - \frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u}\rangle\, \boldsymbol{u}\cdot \nabla^\intercal (\log g_\parallel - \log g_\perp) - \frac{1}{2}\big\langle \boldsymbol{v}, \nabla(\log g_\parallel - \log g_\perp)\big\rangle\, \boldsymbol{u}\boldsymbol{u}^\intercal
\end{aligned}$$

To express $\boldsymbol{v}^\intercal \boldsymbol{\Gamma}$ as a low-rank perturbation of identity, we first note that $\log g_\perp / \log g_\parallel = c$ where $c = \frac{1-\gamma}{\gamma(d-1)}$. Using this relation, we have the following three equalities:

$$\begin{aligned}
&\frac{1}{4}\boldsymbol{v} \cdot (\nabla \log g_\parallel - 2\nabla \log g_\perp)^\intercal - \frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u} \cdot \nabla^\intercal (\log g_\parallel - \log g_\perp) \\
&\qquad = \left(\frac{1}{4}(1-2c)\boldsymbol{v} - \frac{1}{2}(1-c)\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u} \right) \cdot \nabla^\intercal \log g_\parallel \\
&\frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u} \rangle \left(\frac{g_\parallel}{g_\perp} \nabla \log g_\parallel - \nabla \log g_\perp \right) \cdot \boldsymbol{u}^\intercal + \frac{1}{2}\nabla \log g_\perp \cdot \boldsymbol{v}^\intercal \\
&\qquad = \nabla \log g_\parallel \cdot \left(\frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u} \rangle \left(\frac{g_\parallel}{g_\perp} - c \right) \boldsymbol{u}^\intercal + \frac{c}{2}\boldsymbol{v}^\intercal \right)
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{2}\left(1 - \frac{g_\parallel}{g_\perp}\right) \big\langle \nabla \log g_\parallel, \boldsymbol{u} \big\rangle \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u}\boldsymbol{u}^\intercal \\
&\qquad + \frac{1}{2}\frac{g_\perp}{g_\parallel}\left(1 - \frac{g_\parallel}{g_\perp}\right) \langle \nabla \log g_\perp, \boldsymbol{u} \rangle \boldsymbol{u} \cdot (\boldsymbol{v}^\intercal - \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u}^\intercal) \\
&\qquad - \frac{1}{2}\big\langle \boldsymbol{v}, \nabla(\log g_\parallel - \log g_\perp) \big\rangle \boldsymbol{u}\boldsymbol{u}^\intercal \\
&= \frac{1}{2}\boldsymbol{u} \cdot \Bigg[\left(1 - \frac{g_\parallel}{g_\perp}\right) \big\langle \nabla \log g_\parallel, \boldsymbol{u} \big\rangle \left(\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u} + \frac{c g_\perp}{g_\parallel}(\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{u}) \right) \\
&\qquad\qquad - (1-c)\big\langle \boldsymbol{v}, \nabla \log g_\parallel \big\rangle \boldsymbol{u} \Bigg]^\intercal
\end{aligned}$$

So the formula for $\boldsymbol{v}^\intercal \boldsymbol{\Gamma}$ can be simplified as

$$\begin{aligned}\boldsymbol{v}^\intercal \boldsymbol{\Gamma} = \frac{1}{4}(1-2c)\left\langle \boldsymbol{v}, \nabla \log g_\parallel \right\rangle \boldsymbol{I} \\ + \left(\frac{1}{4}(1-2c)\boldsymbol{v} - \frac{1}{2}(1-c)\langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u}\right) \cdot \nabla^\intercal \log g_\parallel \\ + \nabla \log g_\parallel \cdot \left(\frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u}\rangle \left(\frac{g_\parallel}{g_\perp} - c\right)\boldsymbol{u}^\intercal + \frac{c}{2}\boldsymbol{v}^\intercal\right) \\ + \frac{1}{2}\boldsymbol{u} \cdot \Bigg[\left(1 - \frac{g_\parallel}{g_\perp}\right)\left\langle \nabla \log g_\parallel, \boldsymbol{u}\right\rangle \left(\langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u} + \frac{c g_\perp}{g_\parallel}(\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u})\right) \\ - (1-c)\left\langle \boldsymbol{v}, \nabla \log g_\parallel \right\rangle \boldsymbol{u}\Bigg]^\intercal\end{aligned}$$

And finally we obtain

$$\begin{aligned}1 - \frac{\epsilon}{2}\boldsymbol{v}^\intercal \boldsymbol{\Gamma} = \left(1 - \frac{\epsilon}{8}(1-2c)\left\langle \boldsymbol{v}, \nabla \log g_\parallel \right\rangle\right) \boldsymbol{I} \\ - \epsilon\left(\frac{1}{8}(1-2c)\boldsymbol{v} - \frac{1}{4}(1-c)\langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u}\right) \cdot \nabla^\intercal \log g_\parallel \\ - \frac{\epsilon}{2}\nabla \log g_\parallel \cdot \left(\frac{1}{2}\langle \boldsymbol{v}, \boldsymbol{u}\rangle \left(\frac{g_\parallel}{g_\perp} - c\right)\boldsymbol{u}^\intercal + \frac{c}{2}\boldsymbol{v}^\intercal\right) \\ - \frac{\epsilon}{4}\boldsymbol{u} \cdot \Bigg[\left(1 - \frac{g_\parallel}{g_\perp}\right)\left\langle \nabla \log g_\parallel, \boldsymbol{u}\right\rangle \left(\langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u} + \frac{c g_\perp}{g_\parallel}(\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u})\right) \\ - (1-c)\left\langle \boldsymbol{v}, \nabla \log g_\parallel \right\rangle \boldsymbol{u}\Bigg]^\intercal\end{aligned}$$

3.D Relevant geometric notions

3.D.1 Gradient on manifold

Consider a function $\tilde{U}(\tilde{\boldsymbol{\theta}})$ defined on a d-dimensional manifold $M \subset \mathbb{R}^{\tilde{d}}$ and let $T_{\tilde{\boldsymbol{\theta}}}M \subset \mathbb{R}^{\tilde{d}}$ denote the tangent space of M at $\tilde{\boldsymbol{\theta}}$. The gradient $\nabla^M \tilde{U}(\tilde{\boldsymbol{\theta}})$ can be defined as a unique vector in $T_{\tilde{\boldsymbol{\theta}}}M$ such that

$$\left\langle \nabla^M \tilde{U}(\tilde{\boldsymbol{\theta}}), \mathbf{c}'(0)\right\rangle = \frac{\mathrm{d}}{\mathrm{d}t}\tilde{U}(\mathbf{c}(t))\Big|_{t=0} \tag{3.31}$$

for all differentiable curves $\mathbf{c}(t)$ on M with $\mathbf{c}(0) = \tilde{\boldsymbol{\theta}}$. Notice that, under the constraint $\|\mathbf{c}'(0)\| = 1$, the left hand side in (3.31) is maximized when $\mathbf{c}'(0)$ is parallel to $\nabla^M \tilde{U}(\tilde{\boldsymbol{\theta}})$, agreeing with our intuition of the gradient as the direction of the greatest increase in $\tilde{U}(\tilde{\boldsymbol{\theta}})$.

3.D.2 Probability density function on parametrized manifold

Due to the difference in the integration theory over a Euclidean space and a manifold, a pdf on a manifold is defined slightly differently from those on a Euclidean space. Here we describe one way to define a pdf on a parametrized manifold through a generalized change of variable formula.

Suppose a random variable $\boldsymbol{\theta} \in \mathbb{R}^d$ has a pdf $\pi(\boldsymbol{\theta})$. Given a parametrization (i.e. differentiable bijection) $\boldsymbol{g}$ of a manifold M, a random variable $\tilde{\boldsymbol{\theta}} = \boldsymbol{g}(\boldsymbol{\theta}) \in M$ has a pdf

$$\tilde{\pi}(\tilde{\boldsymbol{\theta}}) = |\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^\intercal \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}|^{-1/2} \, \pi(\boldsymbol{\theta}) \quad \text{where } \boldsymbol{\theta} = \boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}) \tag{3.32}$$

If $\boldsymbol{g}$ were a bijection between Euclidean spaces and $\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}$ were a square matrix, then the above formula reduces to the standard change of variable formula, where $|\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}|$ is the change of volume factor. More generally, it can be shown that $|\mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}^\intercal \mathbf{D}\boldsymbol{g}_{\boldsymbol{\theta}}|^{1/2}$ represents the volume of a d-dimensional parallelepiped

$$P = \left\{ \sum_{i=1}^{d} c_i \frac{\partial \boldsymbol{g}}{\partial \theta_i}(\boldsymbol{\theta}) \in T_{\boldsymbol{\theta}} M \ : \ \sum_i c_i \leqslant 1, \, c_i \geqslant 0 \right\}$$

3.D.3 Mapping dynamics on manifold to one on Euclidean space

Given a parametrization $\boldsymbol{g} : \Omega \subset \mathbb{R}^d \to M$ of a manifold $M \subset \mathbb{R}^{\tilde{d}}$, the d by $\tilde{d}$ matrix $\mathbf{D}\boldsymbol{g}^\intercal_{g^{-1}(\tilde{\boldsymbol{\theta}})}$ is a bijection from the tangent space $T_{\tilde{\boldsymbol{\theta}}} M \subset \mathbb{R}^{\tilde{d}}$ to $\mathbb{R}^d$. This is due to the following elementary fact from linear algebra: given a full rank $\tilde{d} \times d$ matrix $\boldsymbol{A}$, its transpose $\boldsymbol{A}^\intercal$ is a bijection from range($\boldsymbol{A}$) to $\mathbb{R}^d$. It then follows that the product

map $\boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^{\intercal}$ defined as

$$\boldsymbol{g}^{-1} \times \mathbf{D}\boldsymbol{g}^{\intercal}(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{p}}) = (\boldsymbol{g}^{-1}(\tilde{\boldsymbol{\theta}}), \mathbf{D}\boldsymbol{g}^{\intercal}_{g^{-1}(\tilde{\boldsymbol{\theta}})}\tilde{\boldsymbol{p}}) \tag{3.33}$$

is a bijection from a collection of tangent space $\bigcup_{\tilde{\boldsymbol{\theta}} \in M} T_{\tilde{\boldsymbol{\theta}}} M$ to $\Omega \times \mathbb{R}^d$. (The collection $TM = \bigcup_{\tilde{\boldsymbol{\theta}} \in M} T_{\tilde{\boldsymbol{\theta}}} M$ is also known as a *tangent bundle*.) Therefore the product map bijectively relates a dynamics on a manifold M to one on a Euclidean space.

4

Variable trajectory length compressible Hamiltonian Monte Carlo

4.1 Introduction

Within the standard HMC framework, an integrator for simulating Hamiltonian dynamics must be reversible and volume-preserving to produce a valid Metropolis proposal (Neal, 2010). In fact, the volume-preserving property can be relaxed by including a Jacobian factor in the calculation of the Metropolis-Hastings acceptance probability (Leimkuhler and Reich, 2009; Lan et al., 2015). Under this generalization of HMC, any reversible dynamics / bijective map can be applied to generate a proposal state. This algorithm is formalized as *compressible HMC* (CHMC) in Fang et al. (2014) and is a generalization of Algorithm 1.1 described in Chapter 1.

This chapter presents a further generalization of Algorithm 1.1, relaxing another condition required by (compressible) HMC. Given a reversible map $\boldsymbol{F}$ and state $(\boldsymbol{\theta}, \boldsymbol{p})$, CHMC proposes the next state by applying the map n times, where the number of steps n can be drawn randomly at each iteration. Though often not stated explicitly, the detailed balance requires the number of steps to be determined independently

of the trajectory $\{(\boldsymbol{\theta}, \boldsymbol{p}), \boldsymbol{F}(\boldsymbol{\theta}, \boldsymbol{p}), \boldsymbol{F}^2(\boldsymbol{\theta}, \boldsymbol{p}), \ldots\}$. As we will show in Section 4.3, this constraint can prevent realizing the full potential of MCMC algorithms based on reversible dynamics.

Our algorithm generalizes the acceptance-rejection mechanism behind CHMC to allow the number of steps to depend on each trajectory of the dynamics while preserving the detailed balance. The number of numerical integration steps taken in simulating a trajectory of HMC is commonly referred to as the *(path) length* of a trajectory in the statistics literature. We therefore call our algorithm *variable trajectory length CHMC* (VTL-CHMC). It should be mentioned that the No-U-Turn-Sampler (NUTS) is another variant of HMC that allows the path lengths to vary from one trajectory to another (Hoffman and Gelman, 2014). However, the motivation behind NUTS is to spare a user the trouble of manually tuning the number of steps, and NUTS in general performs no better than HMC with well-chosen path lengths (Hoffman and Gelman, 2014; Wang et al., 2013). On the other hand VTL-CHMC can improve the performance of CHMC in a more fundamental and significant way. In particular, VTL-CHMC enables an effective application of reversible variable stepsize integrators to HMC-type sampling algorithms based on reversible dynamics.

The rest of the chapter is organized as follows. Section 4.2 reviews the main ideas behind CHMC and provides an example in which the compressible dynamics arises from the use of non-traditional integrators in HMC settings. Such integrators have proven to be more efficient than the commonly used volume-preserving integrators in various applications. The example also serves to introduce the notations and concepts needed in the next section, where VTL-CHMC is motivated as a method to effectively apply variable stepsize integrators in HMC settings. Section 4.3 explains how the existing framework limits the utility of variable stepsize integrators to sampling algorithms. The key observation in addressing this issue leads to a special case of VTL-CHMC. More general construction of VTL-CHMC is provided in Section 4.4.

Section 4.4.1 presents another use case of VTL-CHMC, where HMC is modified to reduce the wasted computation due to unstable numerical approximations and corresponding rejected proposals. The simulation results are shown in Section 4.5 to demonstrate the potential gains from the framework of VTL-CHMC.

4.2 Review of compressible HMC

4.2.1 Basic theory

CHMC generates a valid MCMC algorithm from any reversible dynamics, which does not have to operate in the typical joint parameter space $(\boldsymbol{\theta}, \boldsymbol{p})$ of HMC variants. We therefore use $\boldsymbol{z}$ to denote a parameter space and use a more general notion of reversibility than the one previously introduced. To keep the description of CHMC and the subsequent development of VTL-CHMC more intuitive, the notion of reversibility used here is slightly less general than the one in Fang et al. (2014). It is straightforward to extend the variable trajectory length algorithm of Section 4.4 to the general settings.

A bijective map $\boldsymbol{F}$ is said to be *reversible* if

$$\boldsymbol{F}^{-1} = \boldsymbol{R} \circ \boldsymbol{F} \circ \boldsymbol{R} \quad \text{or equivalently} \quad (\boldsymbol{R} \circ \boldsymbol{F})^{-1} = \boldsymbol{R} \circ \boldsymbol{F} \tag{4.1}$$

for an *involution* $\boldsymbol{R}$ i.e. $\boldsymbol{R} \circ \boldsymbol{R}$ is an identity map. Note that the reversiblity of $\boldsymbol{F}$ implies that of $\boldsymbol{F}^n$ for any n. Let $\mathbf{D}\boldsymbol{F}^n$ denote the Jacobian matrix of $\boldsymbol{F}^n$ and $|\mathbf{D}\boldsymbol{F}^n|$ its determinant. Given a state $\boldsymbol{z}$ and integer n, CHMC proposes the state $\boldsymbol{z}^* = \boldsymbol{R} \circ \boldsymbol{F}^n(\boldsymbol{z})$[1] and accepts or rejects the proposal with probability

$$\min \left\{ 1, \frac{\pi(\boldsymbol{z}^*)|\mathbf{D}\boldsymbol{F}^n(\boldsymbol{z})|}{\pi(\boldsymbol{z})} \right\} \tag{4.2}$$

[1] Within the typical HMC framework as in Algorithm 1.1, the involution (momentum flip) map $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) \to (\boldsymbol{\theta}^*, -\boldsymbol{p}^*)$ can be ignored during the proposal generation since $\pi(\boldsymbol{\theta}, \boldsymbol{p})$ is assumed to be symmetric in $\boldsymbol{p}$.

To see that this transition rule satisfies the detailed balance with respect to $\pi(\cdot)$, consider a small neighborhood B around $\boldsymbol{z}$ and $B^* = \boldsymbol{R} \circ \boldsymbol{F}^n(B)$ around $\boldsymbol{z}^*$, so that $\boldsymbol{R} \circ \boldsymbol{F}^n(B^*) = B$. The proposal move sends the probability mass

$$\int_B \pi(\boldsymbol{z}')\mathrm{d}\boldsymbol{z}' \approx \pi(\boldsymbol{z})\mathrm{vol}(B)$$

from B to B^*. On the other hand, the mass sent from B^* to B by the proposal move can be seen to be

$$\int_{B^*} \pi(\boldsymbol{z}')\mathrm{d}\boldsymbol{z}' = \int_B \pi(\boldsymbol{z}')|\mathbf{D}(\boldsymbol{R} \circ \boldsymbol{F}^n)(\boldsymbol{z}')|\mathrm{d}\boldsymbol{z}' \approx \pi(\boldsymbol{z}^*)|\mathbf{D}\boldsymbol{F}^n(\boldsymbol{z})|\,\mathrm{vol}(B)$$

by the change of variable formula and the fact $|\boldsymbol{R}| = 1$. The acceptance and rejection step of CHMC amounts to rejecting the fraction of move by the ratio of the probability fluxes and thus imposes the detailed balance.

The steps of CHMC are summarized in Algorithm 4.1 below. (The number of steps n can be drawn randomly at each iteration of CHMC.) The use of a deterministic map as a proposal distribution does not yield an ergodic Markov chain. Randomness must be introduced by alternating a deterministic transition rule with a stochastic transition rule that preserves the target density $\pi(\cdot)$, as done in Step 1 of the algorithm. We do not concern ourselves here with how to choose a stochastic transition rule since the choice depends critically on the particular form of $\pi(\cdot)$.

Algorithm 4.1 (Compressible HMC)**.** CHMC generates a Markov chain $\{\boldsymbol{z}^{(m)}\}_m$ with the following transition rule $\boldsymbol{z}^{(m)} \to \boldsymbol{z}^{(m+1)}$:

1. Make a random change $\boldsymbol{z}^{(m)} \to \boldsymbol{z}$ that preserves the target density $\pi(\cdot)$.
2. Let $\boldsymbol{z}^{(m+1)} = \boldsymbol{z}^*$ with probability

$$\min\left\{1, \frac{\pi(\boldsymbol{z}^*)|\mathbf{D}\boldsymbol{F}^n(\boldsymbol{z})|}{\pi(\boldsymbol{z})}\right\}$$

Otherwise, let $\boldsymbol{z}^{(m+1)} = \boldsymbol{z}$.

4.2.2 Example: (Riemann manifold) HMC with non-volume-preserving integrators

HMC and its extension Riemann manifold HMC (RMHMC) require a geometric integrator that preserves the reversibility and volume-preservation property of Hamiltonian dynamics. Under the CHMC framework, however, Hamiltonian dynamics can be approximated using a wider range of integration techniques.

As described earlier in Section 3.2.1, RMHMC follows the framework of Algorithm 1.1 with a choice of the momentum distribution $\boldsymbol{p} \,|\, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{G}(\boldsymbol{\theta}))$. Solving the corresponding Hamiltonian dynamics (1.4) using a reversible integrator with a constant stepsize Δt yields a reversible map $\boldsymbol{F}_{\Delta t}$ so that

$$\boldsymbol{F}_{\Delta t}^{n}(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \approx (\boldsymbol{\theta}(n\Delta t), \boldsymbol{p}(n\Delta t)) \tag{4.3}$$

where $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \geqslant 0}$ denotes the exact solution with the initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. If the reversible map $\boldsymbol{F}_{\Delta t}$ is further required to be volume preserving, then we have $|\mathbf{D}\boldsymbol{F}_{\Delta t}^{n}| = 1$ and the Jacobian factor drops from (4.2), recovering HMC and RMHMC algorithms of Duane et al. (1987) and Girolami and Calderhead (2011). In some applications however, non-volume-preserving approximations of (1.4) have been shown to offer substantial gains in computational efficiency (Lan et al., 2015; Fang et al., 2014).

For example, Lan et al. (2015) considers the ODE corresponding to (1.4) in terms of reparametrization $(\boldsymbol{\theta}, \boldsymbol{v}) = (\boldsymbol{\theta}, \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p})$. The reparametrized ODE admits semi-explicit and explicit reversible approximations, requiring fewer or no fixed point iterations compared to the Störmer-Verlet integrator typically employed in RMHMC. The proposal move using a simulated trajectory is alternated with sampling $\boldsymbol{v}$ from its conditional density $\boldsymbol{v} \,|\, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{G}(\boldsymbol{\theta})^{-1})$, a random move corresponding to Step 1 in Algorithm 4.1. The CHMC algorithm based on the semi-explicit and explicit integrator are found to significantly outperform RMHMC based on the Störmer-Verlet

integrator over a range of examples.

4.3 Special case of variable trajectory length CHMC

Variable trajectory length CHMC (VTL-CHMC) is most naturally motivated as a method to effectively apply variable stepsize integrators in RMHMC settings. For this reason, we first develop this special case of VTL-CHMC in Section 4.3. A more general theory is developed in Section 4.4 along with its application.

4.3.1 Motivation: RMHMC with variable stepsize integrators and limitations of CHMC

In Section 4.2.2, we discussed how CHMC allows us to approximate Hamiltonian dynamics with non-volume-preserving integrators and still generate a valid Metropolis-Hastings proposal. We in particular considered the use of a reversible integrator with a constant stepsize. A wider range of reversible integration techniques for Hamiltonian systems are available in the literature, however, including a number of variable stepsize integrators (Calvo et al., 1998; Blanes and Iserles, 2012; Leimkuhler and Reich, 2005; Hairer et al., 2006). In theory, a variable stepsize integrator similarly produces a valid CHMC proposal as long as the integrator is reversible. However, the use of such an integrator under the existing CHMC framework generally leads to an algorithm with suboptimal sampling efficiency, for the reasons we describe now.

Each step of a variable stepsize integrator approximates the evolution $(\boldsymbol{\theta}(t_n), \boldsymbol{p}(t_n)) \rightarrow (\boldsymbol{\theta}(t_n + \Delta t_n), \boldsymbol{p}(t_n + \Delta t_n))$ where the stepsize Δt_n depends on the current state $(\boldsymbol{\theta}(t_n), \boldsymbol{p}(t_n))$ through a *stepsize controller* $\eta(\boldsymbol{\theta}, \boldsymbol{p})$. The simplest choice of stepsize would be $\Delta t_n = \eta(\boldsymbol{\theta}(t_n), \boldsymbol{p}(t_n))\Delta s$, but the reversibility requires a slightly more sophisticated relationship and the condition $\eta(\boldsymbol{\theta}, \boldsymbol{p}) = \eta(\boldsymbol{\theta}, -\boldsymbol{p})$ (see Section 4.3.2). Most importantly for our discussion, a variable stepsize scheme is equivalent to approximating the following *time-rescaled* Hamiltonian dynamics in a

new time scale $\mathrm{d}s = \eta(\boldsymbol{\theta}, \boldsymbol{p})^{-1}\mathrm{d}t$ with a constant stepsize Δs:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s} = \eta(\boldsymbol{\theta}, \boldsymbol{p})\nabla_{\boldsymbol{p}} H(\boldsymbol{\theta}, \boldsymbol{p}), \ \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}s} = -\eta(\boldsymbol{\theta}, \boldsymbol{p})\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \boldsymbol{p}) \tag{4.4}$$

In other words, a reversible variable stepsize approximation of (1.4) yields a reversible map $\boldsymbol{F}_{\Delta s}$ such that

$$\boldsymbol{F}_{\Delta s}^n(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \approx (\boldsymbol{\theta}(n\Delta s), \boldsymbol{p}(n\Delta s)) \tag{4.5}$$

where $\{\boldsymbol{\theta}(s), \boldsymbol{p}(s)\}_s$ is the solution to the time-rescaled dynamics (4.4) with the initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$.

This implicit time-rescaling behind variable stepsize integration causes trouble for CHMC. The utility of Hamiltonian dynamics (1.4) as a proposal generation mechanism stems from the fact that $\pi(\boldsymbol{\theta}, \boldsymbol{p}) \propto \exp(-H(\boldsymbol{\theta}, \boldsymbol{p}))$ is the *invariant distribution* of the dynamics i.e. if $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ has the distribution $\pi(\cdot)$, then $\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ also has the same distribution $\pi(\cdot)$ for all t. As a consequence, the proposal generated by an approximate solution $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = \boldsymbol{F}_{\Delta t}^n(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ as in (4.3) can be accepted with probability 1 in the limit $\Delta t \to 0$ and $n\Delta t \to t'$. On the other hand, the time-rescaled dynamics (4.4) in general does not preserve the target density $\pi(\boldsymbol{\theta}, \boldsymbol{p})$, and the proposal generated by the approximate solution $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = \boldsymbol{F}_{\Delta s}^n(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ may not be accepted with high probability even in the limit $\Delta s \to 0$ and $n\Delta s \to s'$. In fact, we have the following result:

Theorem 4.1. *Consider a CHMC proposal $(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \to \boldsymbol{F}_{\Delta s}^n(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ generated by a map $\boldsymbol{F}_{\Delta s}^n$ approximating the dynamics (4.4). In the limit $\Delta s \to 0$ and $n\Delta s \to s'$, its acceptance probability is given by*

$$\min\left\{1, \frac{\eta(\boldsymbol{\theta}(s'), \boldsymbol{p}(s'))}{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)}\right\} \tag{4.6}$$

where $\{\boldsymbol{\theta}(s), \boldsymbol{p}(s)\}_s$ denotes the solution to (4.4) with the initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$.

The proof is given in Appendix 4.A.

4.3.2 Algorithm: variable trajectory length for time-rescaled dynamics

In order to address the issue caused by the implicit time-rescaling associated with variable stepsize integrators, VTL-CHMC approximates the dynamics in the original time scale as follows. Fix the initial condition $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ and denote $(\boldsymbol{\theta}_i, \boldsymbol{p}_i) = \boldsymbol{F}^i_{\Delta s}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ where $\boldsymbol{F}_{\Delta s}$ approximates the dynamics in the time scale s as in (4.5). The evolution $(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \to (\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ in the original time scale can be approximated by taking the trajectory dependent number of steps $N(\boldsymbol{\theta}_0, \boldsymbol{p}_0) = N(t, \boldsymbol{\theta}_0, \boldsymbol{p}_0)$ defined as

$$N(\boldsymbol{\theta}_0, \boldsymbol{p}_0) = \min \left\{ n : \sum_{i=1}^{n} \frac{\Delta s}{2} \left(\eta(\boldsymbol{\theta}_{i-1}, \boldsymbol{p}_{i-1}) + \eta(\boldsymbol{\theta}_i, \boldsymbol{p}_i) \right) > t \right\}. \tag{4.7}$$

Then the map $\boldsymbol{F}^N_{\Delta s}$ defined as

$$\boldsymbol{F}^N_{\Delta s}(\boldsymbol{\theta}, \boldsymbol{p}) = \boldsymbol{F}^{N(\boldsymbol{\theta}, \boldsymbol{p})}_{\Delta s}(\boldsymbol{\theta}, \boldsymbol{p}) \tag{4.8}$$

approximates the evolution in the original time scale. The map however cannot be used directly to generate a proposal because in general it is neither reversible or even bijective. The map would be reversible if $N(\boldsymbol{\theta}^*_0, \boldsymbol{p}^*_0) = N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ where $(\boldsymbol{\theta}^*_0, \boldsymbol{p}^*_0) = \boldsymbol{R} \circ \boldsymbol{F}^N_{\Delta s}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, but (4.7) only implies $N(\boldsymbol{\theta}^*_0, \boldsymbol{p}^*_0) \leqslant N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. For example when $\eta(\boldsymbol{\theta}^*_0, \boldsymbol{p}^*_0) \gg \eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, the simulated time along the reverse trajectory $\{(\boldsymbol{\theta}^*_i, \boldsymbol{p}^*_i) = \boldsymbol{F}^i_{\Delta s}(\boldsymbol{\theta}^*_0, \boldsymbol{p}^*_0)\}^n_{i=0}$

$$\sum_{i=1}^{n} \frac{\Delta s}{2} \big(\eta(\boldsymbol{\theta}^*_{i-1}, \boldsymbol{p}^*_{i-1}) + \eta(\boldsymbol{\theta}^*_i, \boldsymbol{p}^*_i) \big)$$

will likely reach the threshold t before $n = N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ steps. Figure 4.1 visually illustrates this phenomenon as well the VTL-CHMC algorithm we describe now. The parameter $(\boldsymbol{\theta}_i, \boldsymbol{p}_i)$ is represented by $\boldsymbol{z}_i$ in the figure.

The key observation behind VTL-CHMC is that we can nonetheless construct

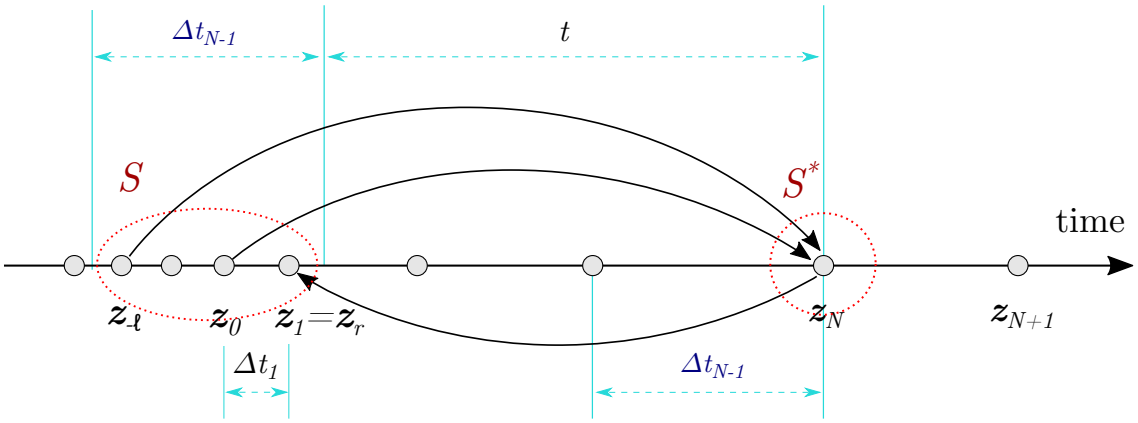


FIGURE 4.1: Visual illustration of the VTL-CHMC (Algorithm 4.2). The forward map $\boldsymbol{F}_{\Delta s}^N$ sends $\boldsymbol{z}_0$ to $\boldsymbol{z}_N = \boldsymbol{F}_{\Delta s}^N(\boldsymbol{z}_0)$ and the "inverse" map $\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N \circ \boldsymbol{R}$ sends $\boldsymbol{z}_N$ to $\boldsymbol{z}_r$ for $r \geqslant 0$. The sets S and S^* can be constructed around $\boldsymbol{z}_0$ and $\boldsymbol{R}(\boldsymbol{z}_N)$ so that the generalize reversibility (4.9) is satisfied.

collections of states S and S^* containing $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ and $(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)$ such that

$$\begin{aligned} \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N(S) \subset S^* \quad \text{and} \quad \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N(S^*) \subset S \\ \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N(S^c) \subset (S^*)^c \quad \text{and} \quad \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N\left((S^*)^c\right) \subset S^c \end{aligned} \tag{4.9}$$

The existence of such sets S and S^* is a property of the map $\boldsymbol{F}_{\Delta s}^N$ and generalizes the notion of reversibility (4.1). The set S is essentially the pre-image of $\{(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)\}$ under $\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N$ and can be constructed by defining $S = \{(\boldsymbol{\theta}_{-\ell}, \boldsymbol{p}_{-\ell}), \ldots, (\boldsymbol{\theta}_r, \boldsymbol{p}_r)\}$ by choosing $\ell, r \geqslant 0$ such that

$$\begin{aligned} \ell &= \max\left\{j \geqslant 0 : \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_{-j}, \boldsymbol{p}_{-j}) = \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right\} \\ r &= \max\left\{j \geqslant 0 : \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_j, \boldsymbol{p}_j) = \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right\} \end{aligned} \tag{4.10}$$

Algorithmically, ℓ and r can be found by solving the dynamics backward and forward

from $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ using the equivalent definitions below:

$$\begin{aligned} \ell &= \max\left\{ j \geqslant 0 : \sum_{i=-j}^{N(t,\boldsymbol{\theta}_0,\boldsymbol{p}_0)-1} \Delta t_i < t \right\} \\ r &= \max\left\{ j \geqslant 0 : \sum_{i=j}^{N(t,\boldsymbol{\theta}_0,\boldsymbol{p}_0)} \Delta t_i > t \right\} \\ &\text{where } \Delta t_i = \frac{\Delta s}{2}\left(\eta(\boldsymbol{\theta}_{i-1}, \boldsymbol{p}_{i-1}) + \eta(\boldsymbol{\theta}_i, \boldsymbol{p}_i)\right) \end{aligned} \tag{4.11}$$

The set S^* is the pre-image of $\{(\boldsymbol{\theta}_r, \boldsymbol{p}_r)\}$ under $\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N$ and can analogously be constructed. Denoting $(\boldsymbol{\theta}_i^*, \boldsymbol{p}_i^*) = \boldsymbol{F}_{\Delta s}^i(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)$, let $S^* = \left\{(\boldsymbol{\theta}_{-\ell^*}^*, \boldsymbol{p}_{-\ell^*}^*), \ldots, (\boldsymbol{\theta}_{r^*}^*, \boldsymbol{p}_{r^*}^*)\right\}$ where $\ell^*, r^* \geqslant 0$ is defined as

$$\begin{aligned} \ell^* &= \max\left\{ j \geqslant 0 : \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_{-j}^*, \boldsymbol{p}_{-j}^*) = \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*) \right\} \\ r^* &= \max\left\{ j \geqslant 0 : \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_j^*, \boldsymbol{p}_j^*) = \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*) \right\} \end{aligned} \tag{4.12}$$

It is shown in Appendix 4.B that the above definition actually implies $r^* = 0$. The proof of (4.9) and of other facts regarding S and S^* are also given in Appendix 4.B.

Having constructed the sets S and S^* with the property (4.9), VTL-CHMC imposes the detailed balance by rejecting a fraction of moves between S and S^* as described in Algorithm 4.2 below.

Algorithm 4.2 (VTL-CHMC)**.** Given a reversible map $\boldsymbol{F}_{\Delta s}$ as in (4.5) and a trajectory length function N as in (4.7), VTL-CHMC generates a Markov chain $\{(\boldsymbol{\theta}^{(m)}, \boldsymbol{p}^{(m)})\}_m$ with the following transition rule $(\boldsymbol{\theta}^{(m)}, \boldsymbol{p}^{(m)}) \to (\boldsymbol{\theta}^{(m+1)}, \boldsymbol{p}^{(m+1)})$:

1. Sample $\boldsymbol{p}_0$ from the conditional density $\boldsymbol{p} \,|\, \boldsymbol{\theta}^{(m)}$ and set $\boldsymbol{\theta}_0 = \boldsymbol{\theta}^{(m)}$.
2. Find the indices ℓ, r, ℓ^*, r^* as in (4.10) and (4.12) by simulating the dynamics forward and backward from $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ and $\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. Then set
$$S = \left\{\boldsymbol{F}_{\Delta s}^{-\ell}(\boldsymbol{\theta}_0, \boldsymbol{p}_0), \ldots, \boldsymbol{F}_{\Delta s}^{r}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right\}$$
$$S^* = \left\{\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0 - r^*}(\boldsymbol{\theta}_0, \boldsymbol{p}_0), \ldots, \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0 + \ell^*}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right\}$$

where $N_0 = N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$.

3. Propose the transition from S to S^* with the acceptance probability which is the smaller of 1 and

$$\frac{\sum_{j=-r*}^{\ell*} \pi\left(\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right|}{\sum_{i=-\ell}^{r} \pi\left(\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right|} \tag{4.13}$$

4. If the transition in Step 3 is accepted, choose a state $\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ from S^* with the probability proportional to

$$\pi\left(\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| \tag{4.14}$$

and set $(\boldsymbol{\theta}^{(m+1)}, \boldsymbol{p}^{(m+1)}) = \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. Otherwise, choose a state $\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ from S with the probability proportional to

$$\pi\left(\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| \tag{4.15}$$

and set $(\boldsymbol{\theta}^{(m+1)}, \boldsymbol{p}^{(m+1)}) = \boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$.

4.3.3 Theory: VTL-CHMC and detailed-balance condition

Theorem 4.2. *The VLT-CHMC of Algorithm 4.2 satisfies the detailed balance condition. To state more precisely, let $P(\boldsymbol{z}^* \mid \boldsymbol{z})$ for $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{p})$ and $\boldsymbol{z}^* = (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ denote the the transition kernel of the algorithm. Then the following holds for any pair of Borel sets Ω and Ω^*:*

$$\int_{\Omega^*} \int_{\Omega} P(\boldsymbol{z}^* | \boldsymbol{z}) \pi(\boldsymbol{z}) \, d\boldsymbol{z} \, d\boldsymbol{z}^* = \int_{\Omega} \int_{\Omega^*} P(\boldsymbol{z} | \boldsymbol{z}^*) \pi(\boldsymbol{z}^*) \, d\boldsymbol{z}^* \, d\boldsymbol{z} \tag{4.16}$$

Proof. We provide an informal argument that can easily be translated into a formal measure theoretic proof. Consider a small neighborhood B_0 around $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. The

total probability in the neighborhood $B = \cup_{i=-\ell}^{r} \boldsymbol{F}_{\Delta s}^{i}(B_0)$ of S is

$$\int_B \pi(\boldsymbol{\theta}, \boldsymbol{p}) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\boldsymbol{p} \approx \sum_{i=-\ell}^{r} \pi\left(\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| \left|B_0\right| \tag{4.17}$$

assuming that B_0 is small enough that $\boldsymbol{F}_{\Delta s}^{i}(B_0)$'s are disjoint. Similarly, the total probability in the neighborhood $B^* = \cup_{j=-r^*}^{\ell^*} \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(B_0)$ of S^* is

$$\int_{B^*} \pi(\boldsymbol{\theta}, \boldsymbol{p}) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\boldsymbol{p} \approx \sum_{j=-r^*}^{\ell^*} \pi\left(\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| \left|B_0\right| \tag{4.18}$$

Comparing the acceptance probability (4.13) with the probability fluxes (4.17) and (4.18), one can see that the acceptance-rejection procedure of Step 3 controls the probability fluxes appropriately to achieve the detailed balance between the neighborhoods B and B^*. Step 4 then imposes the detailed balance within B and B^* by sampling a state according to the relative amount of probability in the individual components $\{\boldsymbol{F}_{\Delta s}^{i}(B_0)\}_{i=-\ell}^{r}$ of B and $\{\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(B_0)\}_{j=-r^*}^{\ell^*}$ of B^*. □

4.3.4 Theoretical efficiency: improvement over CHMC

Throughout Section 4.3 we considered the compressible dynamics (4.4) arising from a variable stepsize integration of Hamiltonian dynamics. In this specific setting with the trajectory length function N as defined in (4.7), VTL-CHMC is guaranteed to have a high average acceptance probability.

Let $\boldsymbol{\Psi}_t$ denote the solution operator of Hamiltonian dynamics (1.4) in the original time scale i.e. $\boldsymbol{\Psi}_t$ is a map such that a trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ is the solution of (1.4) with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. We have the following result:

Theorem 4.3. *In the limit $\Delta s \to 0$ with t fixed, the acceptance probability (4.13) of*

a VTL-CHMC proposal from $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ *converges to a value bounded below by*

$$\frac{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))}{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)} \left\lfloor \frac{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)}{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))} \right\rfloor \tag{4.19}$$

when $\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)) < \eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. *In case* $\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)) > \eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, *a similar lower bound holds for the proposal from* $\boldsymbol{R} \circ \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$.

The proof is given in Appendix 4.A along with a more precise expression for the acceptance probability. Note that the quantity (4.19) is always larger than 1/2 and it tends to 1 as the ratio $\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))/\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ increases, in contrast with the acceptance probability (4.6) of CHMC.

Of course, the acceptance rate of a proposal distribution is not the only factor determining the efficiency of an MCMC algorithm. Nonetheless, the theoretical result above highlights an advantage VTL-CHMC has over the usual CHMC. The bottom line is that VTL-CHMC proposals approximate the original dynamics (1.4) while CHMC proposals approximate the time-rescaled dynamics (4.4). Therefore, VTL-CHMC will generally outperform CHMC whenever the exact solution of the original dynamics constitutes an efficient Markov chain propagator, which typically is the case in RMHMC applications (Girolami and Calderhead, 2011; Chapter 3). This is substantiated by our simulation study in Section 4.5.

4.4 General variable trajectory length CHMC

The key step in Algorithm 4.2 is the construction of the sets S and S^* with the property (4.9). More generally, the detailed balance can be imposed by the same type of acceptance-rejection mechanism whenever the parameter space can be partitioned into a collection of pairs S and S^* such that the set $S \cup S^*$ and $(S \cup S^*)^c$ is closed under a (deterministic) transition rule. Conceivably, a wide range of algorithms can

be devised under this general condition. In this section we present one systematic way to generalize the framework of Section 4.3.

Consider a generic reversible map $\boldsymbol{F}$ on a state space $\boldsymbol{z}$ and associated involution $\boldsymbol{R}$. Fix $\boldsymbol{z}_0$ and denote $\boldsymbol{z}_i = \boldsymbol{F}^i(\boldsymbol{z}_0)$. Choose a *trajectory termination criteria*, or more precisely boolean valued functions $b_n(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) \in \{0, 1\}$, with the following property

$$b_n(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) = b_n(\boldsymbol{R}(\boldsymbol{z}_n), \ldots, \boldsymbol{R}(\boldsymbol{z}_0)) \tag{4.20}$$

as well as the property

$$b_n(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) = 1 \;\text{ only if }\; b_{n-i}(\boldsymbol{z}_i, \ldots, \boldsymbol{z}_n) = 1 \tag{4.21}$$

for any $i > 0$. These properties are satisfied, for example, by a termination criteria $\sum_{i=1}^{n} a(\boldsymbol{z}_i) + a(\boldsymbol{z}_{i-1}) > c$ for a scalar function $a(\boldsymbol{z}) \geqslant 0$. Define a corresponding trajectory length function $N(\boldsymbol{z}_0)$ as

$$N(\boldsymbol{z}_0) = \min\{N'(\boldsymbol{z}_0), N_{\max}\} \;\text{ for }\; N'(\boldsymbol{z}_0) = \min\left\{n : b_n(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) = 1\right\} \tag{4.22}$$

With the reversible map $\boldsymbol{F}_{\Delta s}$ and trajectory length function N of (4.7) replaced by generic ones as above, Algorithm 4.2 remains a valid MCMC scheme. This is because the justification of the algorithm (in Appendix 4.B) only require a trajectory length function N to satisfy the *short return* condition

$$N(\boldsymbol{z}^*) \leqslant N(\boldsymbol{z}) \;\text{ where }\; \boldsymbol{z}^* = \boldsymbol{R} \circ \boldsymbol{F}^{N(\boldsymbol{z})}(\boldsymbol{z}) \tag{4.23}$$

and *order preserving* condition

$$N(\boldsymbol{z}) - n \leqslant N(\boldsymbol{F}^n(\boldsymbol{z})) \;\text{ for any } n \tag{4.24}$$

The intuition behind the terminologies are explained in Appendix 4.B along with the proof of Theorem the general VTL-CHMC algorithm.

Theorem 4.4. *Consider a generalization of Algorithm 4.2 by replacing the reversible map $\boldsymbol{F}_{\Delta s}$ with any reversible map $\boldsymbol{F}(\boldsymbol{z})$ and trajectory length function N of (4.7) with any positive integer-valued function satisfying (4.23) and (4.24). Under this generalization, the proposal scheme continues to satisfy the detailed balance condition.*

4.4.1 Example: rejection avoiding HMC

Here we illustrate a use of the general VTL-CHMC framework through an algorithm of very different flavor from the special case presented in Section 4.3.

A stepsize required for stable numerical integration of Hamilton's equation (1.4) can vary significantly at different regions of a parameter space in some application areas of HMC (Neal, 2010). In such situations, the Hamiltonian may be approximately preserved along a simulated trajectory for a while until it suddenly starts to deviate wildly, leading to a proposal with little chance of acceptance. VTL-CHMC provides a way to "detect" when the trajectory becomes unstable and select an alternate state along the trajectory to transition to.

Let $\boldsymbol{F}_{\Delta t}$ be a volume-preserving and reversible map as in (4.3), approximating Hamiltonian dynamics. Consider a trajectory $\{(\boldsymbol{\theta}_i, \boldsymbol{p}_i) = \boldsymbol{F}^i_{\Delta t}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\}_{i=0,1,2,\ldots}$. When the trajectory becomes unstable, it can be detected by a trajectory termination criteria such as

$$b_n = \mathbb{1}\left\{\max_{0 \leqslant i \leqslant n} H(\boldsymbol{\theta}_i, \boldsymbol{p}_i) - \min_{0 \leqslant i \leqslant n} H(\boldsymbol{\theta}_i, \boldsymbol{p}_i) \geqslant \delta\right\} \tag{4.25}$$

where $\delta > 0$ and $\mathbb{1}$ is an indicator function. We will actually use an alternative criteria below since this leads to a simpler algorithm implementation:

$$b_n = \mathbb{1}\Big\{ |H(\boldsymbol{\theta}_i, \boldsymbol{p}_i) - H(\boldsymbol{\theta}_{i-1}, \boldsymbol{p}_{i-1})| \geqslant \delta \text{ for some } i = 1, \ldots, n\Big\} \tag{4.26}$$

It is easy to check that the criteria (4.25) and (4.26) satisfy the properties (4.20) and (4.21) and define a valid trajectory length function N of the form (4.22) for

Algorithm 4.2. We refer to the version of VTL-CHMC based on the criteria (4.26) as *rejection avoiding HMC*.

A proposal of rejection avoiding HMC recovers the usual HMC proposal with the trajectory length $N_{\max}$ when the fluctuation of a Hamiltonian at each step is within the error tolerance δ. However, upon detecting the fluctuation of magnitude larger than δ at the step $(\boldsymbol{\theta}_{i-1}, \boldsymbol{p}_{i-1}) \to (\boldsymbol{\theta}_i, \boldsymbol{p}_i)$, the algorithm proceeds to simulate the trajectory backward from $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ and $(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*) = (\boldsymbol{\theta}_i, -\boldsymbol{p}_i)$ to determine the sets S and S^* according to the rule in Step 2 of Algorithm 4.2.

4.5 Numerical results

4.5.1 Geometrically tempered HMC with variable stepsize integrator

We compare the performance of CHMC and VTL-CHMC applied to the variable stepsize integrator for geometrically tempered HMC developed earlier in Section 3.4. We use the bi-modal target example of Section 3.5.1 with $\gamma = 1$ and $T = 15$ for the directional tempering metric (3.5). VTL-CHMC is run with the trajectory length function (4.7). The main challenge in this example to explore the parameter space along the first coordinate of $\boldsymbol{\theta}$ due to the multi-modality along this direction. Therefore the efficiency of the sampling algorithms is summarized by the effective sample sizes (ESS) along the first coordinate of $\boldsymbol{\theta}$. The ESS's as well as the acceptance probabilities at different parameter settings of CHMC and VTL-CHMC are summarized in Table 4.1 and 4.2. As predicted by our discussion in Section 4.3, VTL-CHMC has substantially higher acceptance probabilities and, across various parameter settings, is five times more efficient than CHMC with the optimal parameter choice. The time stepsize $\Delta s = .75$ for the variable stepsize integrator was used for all the simulations and was chosen to control the error in the Hamiltonian within a reasonable level along the trajectories. ESS's were computed using the initial monotone sequence estimator of Geyer (1992).

Table 4.1: ESS of CHMC along the first coordinate per 10^5 force evaluations at the various numbers of numerical integration steps. The number of steps coincides with that of force evaluations.

Number of steps	5	10	15	20	25	30	35
Acceptance rate	0.48	0.38	0.37	0.36	0.34	0.33	0.33
ESS	75.7	180	145	83.1	103	123	101

Table 4.2: ESS of VTL-CHMC along the first coordinate per 10^5 force evaluations. The integration time t determines the trajectory lengths through the termination criteria in (4.7).

t	0.50	0.75	1.00	1.25	1.50	1.75	2.00
Number of steps	13	17	21	24	27	30	33
Acceptance rate	0.81	0.78	0.76	0.75	0.73	0.72	0.71
ESS	899	966	924	992	925	921	805

4.5.2 Rejection avoiding HMC

To illustrate the benefit of the rejection avoiding algorithm described in Section 4.4.1, we consider the problem of sampling from a probability density function $\pi(x, y) \propto \exp(-U(x, y))$ as plotted in Figure 4.2. The density $\pi(x, y)$ is constructed as a (continuous) Gaussian mixture

$$\pi(x, y) \propto \int_1^{10} \frac{1}{\sigma_\mu} \exp\left(-\frac{(x-\mu)^2}{2\,\sigma_\mu^2} - y^2\right) \mathrm{d}\mu \tag{4.27}$$

where $\sigma_\mu = 0.1 + (\mu/10)^2$. The density has a property that, along the x-axis, the partial derivative $\partial_y U(x, y)$ varies substantially and so does the stable stepsize for the leap-frog integrator typically employed in HMC. For example, the leap-frog integrator with the stepsize $\Delta t \geqslant 0.4$ approximates the Newton's equations of motion quite accurately in the region $x > 4$, while the stepsize of $\Delta t \approx 0.2$ is required for a numerically stable approximation in the region $x < 2$. In in practice, such a knowledge is obviously not available to us and the appropriate stepsize must be determined empirically from preliminary runs of HMC. A common strategy is to pick a target acceptance rate for the HMC proposals, typically in the range $0.65 \sim 0.8$,

and tune the stepsize accordingly (Beskos et al., 2013; Neal, 2010; Stan Development Team, 2016). This approach would suggest a stepsize well above the stability in this example, however. Figure 4.3 shows that the acceptance rate of HMC to be quite high even for the stepsize $\Delta t = 0.4$. The acceptance rate can be high despite some unstable trajectories because the region where the approximation become unstable contains relatively small, though not negligible, probability. On the other hand, the performance of HMC is severely undermined by the choice of a too large stepsize as can be seen in Figure 4.4. The ESS's for 10^6 force evaluations, estimated from ten independent simulations, are shown so that the computational cost is fixed across the experiments. The error tolerance in Hamiltonian, as in (4.26), for rejection avoiding HMC is set to $\delta = 3$. When $\Delta t = 0.2$, less than 1% of trajectories experience the error in Hamiltonian above the tolerance, so there is no practical difference between HMC with and without rejection avoidance. However, without rejection avoidance, the ESS is reduced by the factor as large as five when increasing the stepsize from $\Delta t = 0.2$ to $\Delta t = 0.3$. The performance degradation is less severe for rejection avoiding HMC as the algorithm concentrates the computational efforts on the stable portions of approximated trajectories.

In summary, choosing an optimal stepsize for HMC is difficult in practice as the choice must be made without the detailed knowledge of a target density. A stepsize can appear to approximate the dynamics accurately but be above the stability limit in some regions. Rejection avoiding HMC can alleviate the effect of a suboptimal stepsize choice and provides far more ESS's than the standard HMC in such situations.

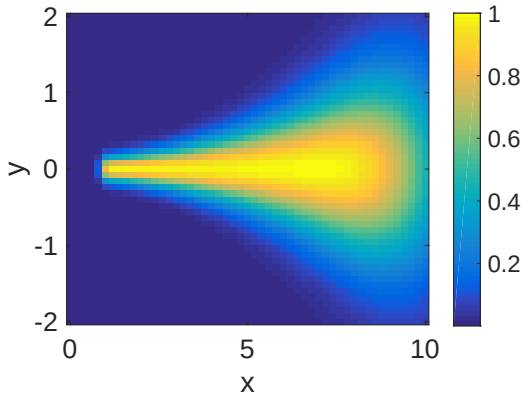


FIGURE 4.2: Plot of (unnormalized) probability density function $\pi(x, y) \propto \exp(-U(x, y))$ used to illustrate the benefit of rejection avoiding HMC.

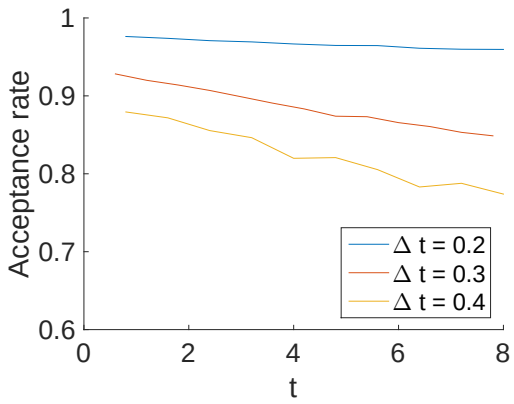


FIGURE 4.3: Acceptance rate of HMC proposals at various settings of stepsize and integration time when sampling from the density shown in Figure 4.2.

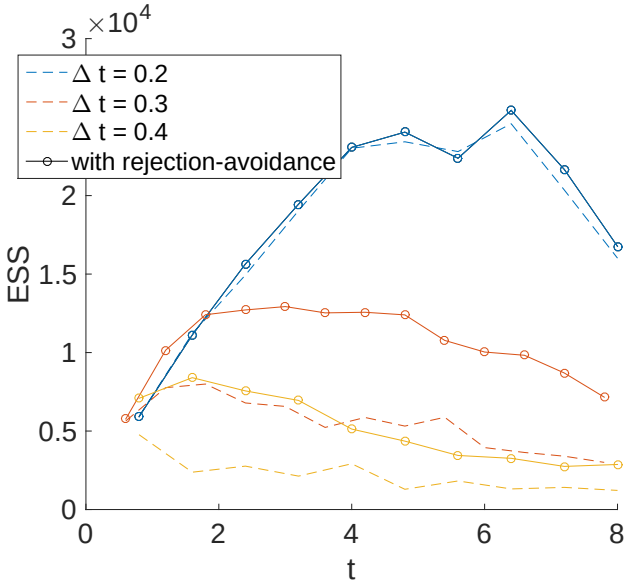


FIGURE 4.4: ESS per 10^6 force evaluations at various settings of stepsize and integration time. The ESS's are for the mean estimation along the x-axis.

Appendix for Chapter 4

4.A Derivation of limiting acceptance probability

In this section we analyse the acceptance probability of CHMC and VTL-CHMC algorithms in the special case of RMHMC with variable stepsize integrators as described in Section 4.3. We derive explicit formulas as well as useful bounds on the acceptance probabilities in the limit $\Delta s \to 0$.

4.A.1 Acceptance probability of CHMC

When approximating a time-rescaled Hamiltonian dynamics (4.4) with a reversible map $\boldsymbol{F}_{\Delta s}$ as in (4.5), the acceptance probability of the CHMC proposal from $(\boldsymbol{\theta}, \boldsymbol{p})$ is calculated by the formula

$$1 \wedge \frac{\pi(\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^n(\boldsymbol{\theta}, \boldsymbol{p}))\, |\mathbf{D}\boldsymbol{F}_{\Delta s}^n(\boldsymbol{\theta}, \boldsymbol{p})|}{\pi(\boldsymbol{\theta}, \boldsymbol{p})}$$

In the limit $\Delta s \to 0$ and $n\Delta s \to s'$, the above quantity converges to

$$1 \wedge \frac{\pi(\boldsymbol{R} \circ \boldsymbol{\Psi}_{s'}(\boldsymbol{\theta}, \boldsymbol{p}))\, |\mathbf{D}\boldsymbol{\Psi}_{s'}(\boldsymbol{\theta}, \boldsymbol{p})|}{\pi(\boldsymbol{\theta}, \boldsymbol{p})}$$

where $\boldsymbol{\Psi}_s$ is the solution operator of the dynamics (4.4) i.e. $\boldsymbol{\Psi}_s$ is the solution of a (vector-valued) differential equation

$$\frac{\mathrm{d}\boldsymbol{\Psi}_s}{\mathrm{d}s} = (\eta \circ \boldsymbol{\Psi}_s)\ \boldsymbol{f} \circ \boldsymbol{\Psi}_s \quad \text{where} \quad \boldsymbol{f} = (\nabla_{\boldsymbol{p}} H, -\nabla_{\boldsymbol{\theta}} H) \tag{4.28}$$

with the initial condition $\boldsymbol{\Psi}_0 = \boldsymbol{I}$. We have $\pi \circ \boldsymbol{\Psi}_{s'} = \pi$ since Hamiltonian dynamics conserves the energy and so does the time-rescaled dynamics. We also have $\pi \circ \boldsymbol{R} = \pi$,

so that $\pi(\boldsymbol{R} \circ \boldsymbol{\Psi}_{s'}(\boldsymbol{\theta}, \boldsymbol{p})) = \pi(\boldsymbol{\theta}, \boldsymbol{p})$. To establish the limiting acceptance probability (4.6), therefore, it remains to show that $|\mathbf{D}\boldsymbol{\Psi}_{s'}(\boldsymbol{\theta}, \boldsymbol{p})| = \eta(\boldsymbol{\Psi}_{s'}(\boldsymbol{\theta}, \boldsymbol{p}))/\eta(\boldsymbol{\theta}, \boldsymbol{p})$. The Jacobian $\mathbf{D}\boldsymbol{\Psi}_s$ satisfies a matrix-valued differential equation $\frac{\partial}{\partial s}\mathbf{D}\boldsymbol{\Psi}_s = \mathbf{D}\boldsymbol{f} \circ \boldsymbol{\Psi}_s \, \mathbf{D}\boldsymbol{\Psi}_s$ and therefore Liouville's formula tells us that

$$|\mathbf{D}\boldsymbol{\Psi}_{s'}| = \exp\left(\int_0^{s'} \operatorname{tr}\left(\mathbf{D}\boldsymbol{f} \circ \boldsymbol{\Psi}_s\right) \mathrm{d}s\right)$$

A straightforward calculation shows that $\operatorname{tr}\left(\mathbf{D}\boldsymbol{f} \circ \boldsymbol{\Psi}_s\right) = \frac{\partial}{\partial s} \log \eta \circ \boldsymbol{\Psi}_s$, from which the identity $|\mathbf{D}\boldsymbol{\Psi}_{s'}| = \eta \circ \boldsymbol{\Psi}_{s'}/\eta$ follows.

4.A.2 Acceptance probability of VTL-CHMC

In the derivation below, we will follow the notations of Section 4.3.2. Namely, we set $(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*) = \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, $(\boldsymbol{\theta}_i, \boldsymbol{p}_i) = \boldsymbol{F}_{\Delta s}^i(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, and $(\boldsymbol{\theta}_i^*, \boldsymbol{p}_i^*) = \boldsymbol{F}_{\Delta s}^i(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)$. The trajectory length function $N = N(t)$ is defined as in (4.7) and the sets S and S^* as in Algorithm 4.2. Note that $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ is fixed, but other quantities depend on Δs, including but not limited to $(\boldsymbol{\theta}_i, \boldsymbol{p}_i)$'s, $N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, and S. We do not denote the dependence explicitly but it is implied.

We will show that the acceptance probability of the transition from S to S^* converges to

$$1 \wedge \frac{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))|S^*|}{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)|S|} \tag{4.29}$$

as $\Delta s \to 0$ while t fixed. Moreover, if $\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)) < \eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, then in the limit $\Delta s \to 0$ we have $|S| = 1$ and

$$\frac{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)}{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))} - 1 \leqslant |S^*| \leqslant \frac{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)}{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))} + 1 \tag{4.30}$$

The claimed lower bound (4.19) on the acceptance probability follows immediately from (4.29) and (4.30).

It is not difficult to show that $\operatorname{diam}(S) \to 0$ and $\operatorname{diam}(S^*) \to 0$ as $\Delta s \to 0$. This means that the elements of S (and of S^*) collapse to a single state as $\Delta s \to 0$. More precisely, for all $-\ell \leqslant i \leqslant r$ and $-r^* \leqslant j \leqslant \ell^*$,

$$\begin{aligned} \boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0) &\to (\boldsymbol{\theta}_0, \boldsymbol{p}_0) \\ \boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0) &\to \boldsymbol{R} \circ \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \end{aligned} \tag{4.31}$$

where $N_0 = N(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ and r, ℓ, r^*, ℓ^* are defined as in (4.10) and (4.12). It follows that

$$\begin{aligned} \pi\left(\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{i}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| &\to \pi(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \\ \pi\left(\boldsymbol{R} \circ \boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right) \left|\mathbf{D}\boldsymbol{F}_{\Delta s}^{N_0+j}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)\right| &\to \pi(\boldsymbol{R} \circ \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)) \left|\mathbf{D}\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))\right| \end{aligned} \tag{4.32}$$

By the same argument as in Section 4.A.1, we can show that

$$\pi(\boldsymbol{R} \circ \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)) \left|\mathbf{D}\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))\right| = \pi(\boldsymbol{\theta}_0, \boldsymbol{p}_0) \frac{\eta(\boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0))}{\eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0)} \tag{4.33}$$

establishing the claimed formula (4.29).

We now turn to the proof of the inequality (4.30). The intuition behind the inequality and the proof below is that the size of the set $|S^*|$ is roughly equal to the number of intervals of length $\Delta s \cdot \eta(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)$ that can be fit inside the interval $(t,\, t + \Delta s \cdot \eta(\boldsymbol{\theta}_0, \boldsymbol{p}_0))$. Denote $N_0^* = N(\boldsymbol{\theta}_0^*, \boldsymbol{p}_0^*)$. By the definition of N_0^*, r^*, and ℓ^*, we must have

$$\sum_{i=-\ell^*+1}^{N_0^*-1} \frac{\Delta s}{2} \left(\eta(\boldsymbol{\theta}_{i-1}^*, \boldsymbol{p}_{i-1}^*) + \eta(\boldsymbol{\theta}_i^*, \boldsymbol{p}_i^*)\right) < t < \sum_{i=r^*+1}^{N_0^*} \frac{\Delta s}{2} \left(\eta(\boldsymbol{\theta}_{i-1}^*, \boldsymbol{p}_{i-1}^*) + \eta(\boldsymbol{\theta}_i^*, \boldsymbol{p}_i^*)\right) \tag{4.34}$$

which implies that

$$\sum_{i=-\ell^*+1}^{r^*} \frac{1}{2}\left(\eta(\boldsymbol{\theta}^*_{i-1},\boldsymbol{p}^*_{i-1})+\eta(\boldsymbol{\theta}^*_{i},\boldsymbol{p}^*_{i})\right) < \frac{1}{2}\left(\eta(\boldsymbol{\theta}^*_{N_0^*-1},\boldsymbol{p}^*_{N_0^*-1})+\eta(\boldsymbol{\theta}^*_{N_0^*},\boldsymbol{p}^*_{N_0^*})\right) \tag{4.35}$$

Also by the definition N_0^*, r^*, and ℓ^*, we must have

$$\sum_{i=r^*+2}^{N_0^*} \frac{\Delta s}{2}\left(\eta(\boldsymbol{\theta}^*_{i-1},\boldsymbol{p}^*_{i-1})+\eta(\boldsymbol{\theta}^*_{i},\boldsymbol{p}^*_{i})\right) < t < \sum_{i=-\ell^*}^{N_0^*-1} \frac{\Delta s}{2}\left(\eta(\boldsymbol{\theta}^*_{i-1},\boldsymbol{p}^*_{i-1})+\eta(\boldsymbol{\theta}^*_{i},\boldsymbol{p}^*_{i})\right) \tag{4.36}$$

which implies that

$$\frac{1}{2}\left(\eta(\boldsymbol{\theta}^*_{N_0^*-1},\boldsymbol{p}^*_{N_0^*-1})+\eta(\boldsymbol{\theta}^*_{N_0^*},\boldsymbol{p}^*_{N_0^*})\right) < \sum_{i=-\ell^*}^{r^*+1} \frac{1}{2}\left(\eta(\boldsymbol{\theta}^*_{i-1},\boldsymbol{p}^*_{i-1})+\eta(\boldsymbol{\theta}^*_{i},\boldsymbol{p}^*_{i})\right) \tag{4.37}$$

Since $\operatorname{diam}(S) \to 0$ and $\operatorname{diam}(S^*) \to 0$ as $\Delta s \to 0$, the inequalities (4.35) and (4.37) converge to

$$\eta(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0)(|S^*|-1) \leqslant \eta(\boldsymbol{\theta}_0,\boldsymbol{p}_0) \leqslant \eta(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0)(|S^*|+1) \tag{4.38}$$

The desired inequality (4.30) is obtained by rearranging the terms in the above inequality.

Finally, we turn to the proof of the fact that $|S| \to 1$ as $\Delta s \to 0$ when $\eta(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0) < \eta(\boldsymbol{\theta}_0,\boldsymbol{p}_0)$. To this end, we only need to note that all the arguments in the proof of (4.38) remain valid if we switch the role of $(\boldsymbol{\theta}^*_i,\boldsymbol{p}^*_i)$, r^*, ℓ^* and N_0^* with $(\boldsymbol{\theta}_i,\boldsymbol{p}_i)$, r, ℓ and N_0. This means that the inequality (4.38) still holds if we switch the role of S^* with S and of $(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0)$ with $(\boldsymbol{\theta}_0,\boldsymbol{p}_0)$, yielding the inequality

$$\eta(\boldsymbol{\theta}_0,\boldsymbol{p}_0)(|S|-1) \leqslant \eta(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0) \leqslant \eta(\boldsymbol{\theta}_0,\boldsymbol{p}_0)(|S|+1)$$

In particular, we have $|S| \leqslant \frac{\eta(\boldsymbol{\theta}^*_0,\boldsymbol{p}^*_0)}{\eta(\boldsymbol{\theta}_0,\boldsymbol{p}_0)} + 1$ and hence $|S| = 1$.

4.B Proof of general VTL-CHMC algorithm

Proof of Theorem 4.4. In Section 4.3.3, the detailed balance condition of VTL-CHMC was derived using the notations of Algorithm 4.2. However, it is easy to see that the same analysis as in the proof of Theorem 4.2 carries through when we replace the reversible map $\boldsymbol{F}_{\Delta s}$ of Section 4.3 with any reversible map $\boldsymbol{F}(\boldsymbol{z})$ as long as the set S and S^* satisfies (4.9). Here we establish the last piece in our proof of the general VTL-CHMC algorithm; the property (4.9) holds whenever N satisfies the short-return (4.23) and order-preserving condition (4.24). Aside from a more general choice of $\boldsymbol{F}$ and N, all the notations and definitions below directly parallel those in Section 4.3.2 used for our presentation of the VTL-CHMC special case.

Fix $\boldsymbol{z}_0$ and denote $\boldsymbol{z}_i = \boldsymbol{F}^i(\boldsymbol{z}_0)$, $\boldsymbol{z}_0^* = \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_0)$, and $\boldsymbol{z}_i^* = \boldsymbol{F}^i(\boldsymbol{z}_0^*)$. A trajectory function N determines the sets via the formula $S = \{\boldsymbol{z}_{-\ell}, \ldots, \boldsymbol{z}_r\}$ and $S^* = \{\boldsymbol{z}_{-\ell^*}^*, \ldots, \boldsymbol{z}_{r^*}^*\}$ where $\ell, r, \ell^*, r^* \geqslant 0$ are defined as

$$\begin{aligned} \ell &= \max\{i \geqslant 0 : \boldsymbol{F}^N(\boldsymbol{z}_{-i}) = \boldsymbol{F}^N(\boldsymbol{z}_0)\} \\ r &= \max\{i \geqslant 0 : \boldsymbol{F}^N(\boldsymbol{z}_i) = \boldsymbol{F}^N(\boldsymbol{z}_0)\} \\ \ell^* &= \max\{i \geqslant 0 : \boldsymbol{F}^N(\boldsymbol{z}_{-i}^*) = \boldsymbol{F}^N(\boldsymbol{z}_0^*)\} \\ r^* &= \max\{i \geqslant 0 : \boldsymbol{F}^N(\boldsymbol{z}_i^*) = \boldsymbol{F}^N(\boldsymbol{z}_0^*)\} \end{aligned} \tag{4.39}$$

To build the intuition behind the proof, we define a partial ordering $\preceq$ on the parameter space as follows:

$$\boldsymbol{z} \preceq \tilde{\boldsymbol{z}} \quad \text{if} \quad \boldsymbol{F}^i(\boldsymbol{z}) = \tilde{\boldsymbol{z}} \text{ for } i \geqslant 0 \tag{4.40}$$

Note that $\boldsymbol{z} \preceq \tilde{\boldsymbol{z}}$ if and only if $\boldsymbol{R}(\tilde{\boldsymbol{z}}) \preceq \boldsymbol{R}(\boldsymbol{z})$, due to the reversibility of $\boldsymbol{F}$. With this notation, the short-return condition can be expressed as

$$\boldsymbol{z} \preceq \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}^*) \quad \text{for} \quad \boldsymbol{z}^* = \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}) \tag{4.41}$$

The condition (4.41) can be interpreted intuitively as follows; according to the trajectory termination criteria imposed by N, the reverse trajectory $\boldsymbol{z}_0^*, \boldsymbol{z}_1^*, \ldots$ must terminate at $\boldsymbol{z}_0$ or at $\boldsymbol{z}_i$ for $i > 0$ before coming all the way back to $\boldsymbol{z}_0$. The order-preserving condition simply amounts to

$$\boldsymbol{F}^N(\boldsymbol{z}) \preceq \boldsymbol{F}^N(\tilde{\boldsymbol{z}}) \quad \text{if} \quad \boldsymbol{z} \preceq \tilde{\boldsymbol{z}} \tag{4.42}$$

We now show how the order-preserving and short-return condition implies (4.9). By the order-preserving condition, we know that

$$\boldsymbol{F}^N(\boldsymbol{z}_{-\ell}) \preceq \boldsymbol{F}^N(\boldsymbol{z}_i) \preceq \boldsymbol{F}^N(\boldsymbol{z}_r) \tag{4.43}$$

for all $-\ell \leqslant i \leqslant r$. On the other hand, we have $\boldsymbol{F}^N(\boldsymbol{z}_{-\ell}) = \boldsymbol{F}^N(\boldsymbol{z}_r) = \boldsymbol{R}(\boldsymbol{z}_0^*)$ by the definition of ℓ and r, so it follows that $\boldsymbol{R} \circ \boldsymbol{F}^N(\{\boldsymbol{z}_{-\ell}, \ldots, \boldsymbol{z}_r\}) = \{\boldsymbol{z}_0^*\}$.

We now turn to demonstration of $\boldsymbol{R} \circ \boldsymbol{F}^N(S^*) = \{\boldsymbol{z}_r\}$. To this end, it suffices to show $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_0^*) = \boldsymbol{z}_r$ as the definition of ℓ^* and r^* combined with the order-preserving condition implies $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_i^*) = \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_0^*)$ for all $-\ell^* \leqslant i \leqslant r^*$. Since $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_r) = \boldsymbol{z}_0^*$, the short-return condition tells us $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_0^*) = \boldsymbol{z}_{r+k}$ for some $k \geqslant 0$. To show that $k = 0$, first observe that an application of the short-return condition to the state $\boldsymbol{z}_{r+k}$ implies $\boldsymbol{z}_0^* \preceq \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_{r+k})$. On the other hand, the order-preserving condition implies $\boldsymbol{F}^N(\boldsymbol{z}_r) \preceq \boldsymbol{F}^N(\boldsymbol{z}_{r+k})$ and hence $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_{r+k}) \preceq \boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_r) = \boldsymbol{z}_0^*$. The preceding inequalities together show that $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_{r+k}) = \boldsymbol{z}_0^*$. Since r was defined as the largest integer i such that $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_i) = \boldsymbol{z}_0^*$, it follows that $k = 0$ and $\boldsymbol{R} \circ \boldsymbol{F}^N(\boldsymbol{z}_0^*) = \boldsymbol{z}_r$.

The remaining relations in (4.9) as well as the fact $r^* = 0$ can be proved similarly with repeated applications of the short-return and order-preserving properties. □

5

Recycling intermediate steps to improve Hamiltonian Monte Carlo

5.1 Introduction

Let $\boldsymbol{F}_\epsilon : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ denote a map corresponding to one numerical integration step of size ϵ, for example the leapfrog step as in (1.7). As described in earlier sections, HMC and related algorithms usually use $M > 1$ steps of an integrator to generate a Metropolis proposal. (Throughout this chapter, a non-bold M denotes a positive integer and not to be confused a mass matrix $\boldsymbol{M}$.) All the intermediate values $\boldsymbol{F}_\epsilon^k(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ for $k < M$ are discarded under current practice. As we will show, this is wasteful since the intermediate values can be *recycled* to generate additional samples from posterior distributions. The recycling algorithm only requires quantities that have already been sampled or computed, so there is essentially no extra computational cost. Our proposed recycling approach can also be applied directly to a wide variety of modified HMC algorithms (Neal, 2010; Girolami and Calderhead, 2011; Pakman and Paninski, 2013, 2014; Lan et al., 2015; Shahbaba et al., 2014; Fang et al., 2014; Zhang et al., 2016; Lu et al., 2016). Extensions to more complex variants are also

possible, including the No-U-Turn-Sampler (NUTS) (Hoffman and Gelman, 2014; Stan Development Team, 2016).

Our algorithm is distinguished by its simplicity and generality compared to alternative algorithms for utilizing the intermediate values of HMC (Neal, 1994; Calderhead, 2014; Bernton et al., 2015). Under our framework, one can typically implement of an HMC variant as usual and simply add several lines of code to recycle the intermediate values using the familiar acceptance and rejection probabilities. The underlying idea behind our algorithm is similar to Neal (1994). He realized that, in the variant of HMC that uses a collection of states in computing the acceptance probability, those states can be re-used when computing the posterior summaries through conditional expectation. Our theory is more general and easily translated into practical methods to improve a variety of multi-proposal algorithm. Our theory can also justify various schemes to select only a subset of the intermediate states to recycle, which is an important feature for scalability as the extra memory requirement to store the additional samples becomes substantial in a high-dimensional parameter space (see Section 5.5). Another method to make use of the intermediate states was proposed by Calderhead (2014) and its Rao-Blackwellization by Bernton et al. (2015) as a special instance of a multi-proposal MCMC algorithm based on the *super-detailed balance* condition (Frenkel, 2004; Tjelmeland, 2004). Their algorithm is more complex; it requires a trajectory to be simulated forward and backward in a symmetric manner, followed by the acceptance-rejection step using the generalized Metropolis-Hastings algorithm (Calderhead, 2014) or assignment of appropriate weights to the intermediate values (Bernton et al., 2015). Importantly, while our algorithm applies straightforwardly to NUTS, arguably the most popular variant of HMC (Stan Development Team, 2016), theirs does not. This is because NUTS yields a variable number of intermediate states and does not constitute a multi-proposal scheme necessary for using their algorithms.

5.2 Recycled Hamiltonian Monte Carlo

The following non-standard HMC algorithm accepts or rejects each of the intermediate values, enabling recycling of these samples. The number of steps $L^{(i)}$ is randomized as recommended in the literature to avoid periodic behavior in the trajectories of (1.4) (Neal, 2010).

Algorithm 5.1 (Recycled HMC)**.** Generate random variables $\{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 0, 1, \ldots, M\}_{i \geqslant 1}$ so that the sequence $\{(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\}_{i \geqslant 1}$ forms a Markov chain with transition rule $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)})$ as follows:

1. Resample a momentum: $\boldsymbol{p}_0^{(i+1)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{M})$.
2. For $k = 1, \ldots, M$, let $(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)}) = \boldsymbol{F}_\epsilon^k(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})$ with probability
$$\min\left\{1, \frac{\pi\big(\boldsymbol{F}_\epsilon^k(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\big)}{\pi\big((\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\big)}\right\} \tag{5.1}$$
and $(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)}) = (\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})$ otherwise.
3. Set $(\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)}) = (\boldsymbol{\theta}_{L^{(i)}}^{(i)}, \boldsymbol{p}_{L^{(i)}}^{(i)})$ for $L^{(i)}$ drawn from a distribution $\pi_{\mathcal{L}}(\cdot)$ on $\{1, \ldots, M\}$.

The transition rule $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)})$ above coincides with that of the standard HMC algorithm. Although HMC discards $(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})$ for all $k \neq 0$, the intermediate samples can be recycled as valid draws from the target distribution, a consequence of a more general theory given in the next section.

Theorem 5.1. *If the samples $(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})$ for $k = 1, \ldots, M$ are generated as in Algorithm 5.1, then*
$$\frac{1}{NM}\sum_{i=1}^{N}\sum_{k=1}^{M} \delta_{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})}(\cdot) \xrightarrow{w} \pi(\cdot) \quad \textit{as} \quad N \to \infty, \tag{5.2}$$
where $\xrightarrow{w}$ denotes the weak convergence of a measure.

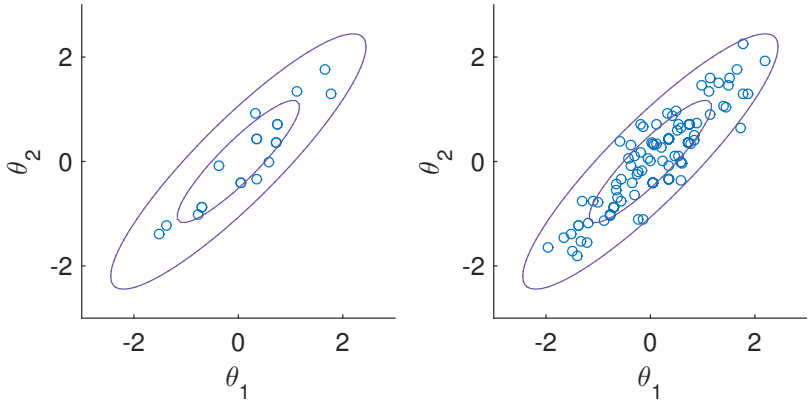


FIGURE 5.1: Comparison of HMC with and without recycling. The samples are drawn from a bivariate Gaussian with correlation 0.9. The contours indicate the 50% and 95% highest density region. The tuning parameters were chosen as $\epsilon = 0.486$ and $L^{(i)} \sim \text{Uniform}\{4, 5, 6\}$.

The benefit of recycling is visually illustrated in Fig. 5.1. Recycling requires Metropolis-type acceptance-rejection for the intermediate steps $\boldsymbol{F}_\epsilon^k(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})$ as in (5.1), but the calculation of acceptance probabilities typically takes little additional computational time. The unnormalized target densities at the intermediate values are already computed in common variants of HMC (Neal, 2010; Hoffman and Gelman, 2014) or can typically be obtained cheaply as a by-product of computing the gradients $\nabla \log \pi_{\boldsymbol{\theta}}$.

The recycled HMC algorithm above requires us to simulate trajectories for M steps at each iteration of HMC even if we use the $L^{(i)}$th leap-frog step with $L^{(i)} < M$ as the proposal for the starting point of the next trajectory. This is not necessary in an alternative version of the recycling algorithm described in Theorem 5.2, leading to a more direct modification of the standard HMC algorithm.

Theorem 5.2. *If the samples* $(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})$ *for* $k = 1, \ldots, L^{(i)}$ *are generated as in Algorithm 5.1, then*

$$\frac{1}{\sum_{i=1}^N L^{(i)}} \sum_{i=1}^N \sum_{k=1}^{L^{(i)}} \delta_{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})}(\cdot) \xrightarrow{w} \pi(\cdot) \text{ as } N \to \infty. \tag{5.3}$$

5.3 Theory behind recycling algorithm

The validity of recycled HMC as in Theorem 5.1 and 5.2 follows from a more general principle below.

Theorem 5.3. *Let $P_k(\cdot\,|\,\cdot)$ for $k = 0, 1, \ldots, M$ be transition kernels with a common stationary measure $\pi(\cdot)$ and suppose $P_0(\cdot\,|\,\cdot)$ is uniquely ergodic.*[1] *Consider a Markov chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$ on a product space $\boldsymbol{z} = (\boldsymbol{z}_0, \ldots, \boldsymbol{z}_M)$ whose transition probability $z \to \boldsymbol{z}^*$ only depends on the coordinate $\boldsymbol{z}_0$ i.e.*

$$P(\boldsymbol{z}_0^*, \ldots, \boldsymbol{z}_M^* \,|\, \boldsymbol{z}_0, \ldots, \boldsymbol{z}_M) = P(\boldsymbol{z}_0^*, \ldots, \boldsymbol{z}_M^* \,|\, \boldsymbol{z}_0) \tag{5.4}$$

and has the marginal densities

$$\int P(\boldsymbol{z}_0^*, \ldots, \boldsymbol{z}_M^* \,|\, \boldsymbol{z}_0)\, \mathrm{d}\boldsymbol{z}_{-k}^* = P_k(\boldsymbol{z}_k^* \,|\, \boldsymbol{z}_0) \tag{5.5}$$

where $\boldsymbol{z}_{-k}^* = (\boldsymbol{z}_0^*, \ldots, \boldsymbol{z}_{k-1}^*, \boldsymbol{z}_{k+1}^*, \ldots, \boldsymbol{z}_M^*)$ for $k = 0, 1, \ldots, M$. Then the following result holds:

$$\frac{1}{NM} \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{\boldsymbol{z}_k^{(i)}}(\cdot) \xrightarrow{w} \pi(\cdot) \quad \textit{as} \quad N \to \infty, \tag{5.6}$$

Additionally, the Markov chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$ is geometrically (or uniformly) ergodic if $P_0(\cdot\,|\,\cdot)$ is geometrically (or uniformly) ergodic.

The proof is given in Appendix 5.A. Theorem 5.3 has a subtle but important difference from "composition sampling," in which one would first generate a Markov chain $\{\boldsymbol{z}_0^{(i)}\}_{i\geqslant 0}$ and then sample $(\boldsymbol{z}_1^{(i+1)}, \ldots, \boldsymbol{z}_M^{(i+1)})$ from a conditional distribution $\pi^*(\,\cdot\,|\,\boldsymbol{z}_0^{(i)})$. For a Markov chain generated as in Theorem 5.3, the conditional distribution $\boldsymbol{z}_1^{(i+1)}, \ldots, \boldsymbol{z}_M^{(i+1)} \,|\, \boldsymbol{z}_0^{(i)}$ may have dependency on $\boldsymbol{z}_0^{(i+1)}$. This additional flexibility is critical for the recycling algorithms presented in this chapter.

[1] A transition kernel (or a Markov chain) with a unique stationary measure is called *uniquely ergodic*. The uniqueness of a stationary measure implies ergodicity by the ergodic decomposition theorem (Kallenberg, 2002).

Theorem 5.3 reduces to Theorem 5.1 when the transition kernel $P_k(\cdot \,|\, \cdot)$ in the parameter space $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{p})$ is constructed as one iteration of HMC with k leapfrog steps for $k \geqslant 1$ and $P_0(\cdot \,|\, \cdot)$ with $L \sim \pi_{\mathcal{L}}(\cdot)$ leapfrog steps. Theorem 5.2 is similarly justified by the following extension of Theorem 5.3, which also justifies various subsampling schemes for recycled samples.

Theorem 5.4. *Under the assumptions of Theorem 5.3, the following convergence result holds for i.i.d. random subsets $S^{(i)} \subset \{1, \ldots, M\}$ independent of $\{\boldsymbol{z}^{(i)}\}_{i \geqslant 1}$:*

$$\frac{1}{\sum_{i=1}^{N} |S^{(i)}|} \sum_{i=1}^{N} \sum_{k \in S^{(i)}} \delta_{\boldsymbol{z}_k^{(i)}}(\cdot) \xrightarrow{w} \pi(\cdot) \text{ as } N \to \infty. \tag{5.7}$$

The general formulation of the recycling algorithm as in Theorem 5.3 and 5.4 is of practical value for any MCMC algorithm that simultaneously yields multiple valid transition kernels $P_k(\cdot \,|\, \cdot)$'s. Indeed, in many variants of HMC (Neal, 2010; Girolami and Calderhead, 2011; Shahbaba et al., 2014; Fang et al., 2014), a proposal is generated by computing a long trajectory whose intermediate steps constitute valid proposal states that can be all recycled by simply adding acceptance-rejection steps as in Algorithm 5.1. Our theory also provides an alternate and simpler justification of the algorithms by Calderhead (2014) and Bernton et al. (2015) as shown in Appendix 5.C. Our recycling algorithm can also be applied to more complex proposal generation mechanisms as we illustrate in the next section.

5.4 Recycled No-U-Turn-Sampler

No-U-Turns-Sampler (NUTS) of Hoffman and Gelman (2014) automates choice of path lengths by simulating each trajectory of Hamiltonian dynamics until it starts moving back towards the starting point, a criteria they termed the *U-turn* condition. The lengths of trajectories are recursively doubled forward or backward in a randomly chosen direction. This generates a collection of states with a binary tree structure

and reversibility can be ensured by checking the U-turn condition for the entire tree as well as all its subtrees.

Unlike the simpler trajectory simulation procedure behind HMC, the trajectory doubling procedure of NUTS does not yield a sequence of valid intermediate proposals. In particular, the empirical distribution does not converges to the correct target distribution if we naively recycle all the intermediate states of NUTS as in Algorithm 5.1. A simple recycling algorithm for NUTS can nonetheless be devised by taking advantage of the following fact.

Fact 5.5. The following transition rule $P_1 : (\boldsymbol{\theta}_0, \boldsymbol{p}_0) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ preserves the target distribution $\pi(\cdot)$. Let $\mathcal{T} = \mathcal{T}(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ denote a (random) collection of 2^d states generated by an iteration of NUTS from the initial state $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$, including $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ itself. Generate $u \sim \text{Unif}([0, \pi(\boldsymbol{\theta}_0, \boldsymbol{p}_0)])$ and sample $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ uniformly from the collection of acceptable states

$$\mathcal{A} = \mathcal{A}(\mathcal{T}, u) = \{(\boldsymbol{\theta}, \boldsymbol{p}) \in \mathcal{T} \,|\, \pi(\boldsymbol{\theta}, \boldsymbol{p}) > u\} \tag{5.8}$$

The stationarity of $\pi(\cdot)$ under the above transition rule follows from the discussion in Hoffman and Gelman (2014). Fact 5.5 motivates the following simple algorithm to utilize the intermediate states generated during each iteration of NUTS.

Algorithm 5.2 (Simple recycled NUTS)**.** Run NUTS to generate a sequence of random variables $\{(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\}_{i \geqslant 1}$. Additionally at each iteration of NUTS, generate $\{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 1, \ldots, M\}$ by sampling M variables without replacement from the acceptable states $\mathcal{A}\big(\mathcal{T}(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})\big)$ as in (5.8).

Algorithm 5.2 is justified with a straightforward application of Theorem 5.3, observing from Fact 5.5 that the transition $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})$ preserves the target distribution $\pi(\cdot)$ for each $k = 1, \ldots, M$. In fact, it is more statistically efficient to sample $(\boldsymbol{\theta}_1^{(i)}, \boldsymbol{p}_1^{(i)}), \ldots, (\boldsymbol{\theta}_M^{(i)}, \boldsymbol{p}_M^{(i)})$ from $\mathcal{A}\big(\mathcal{T}(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})\big)$ so that they are evenly

spread along a NUTS trajectory. Such a sampling scheme can be implemented in a simple and memory efficient (i.e. without storing all the intermediate states in memory) manner by taking advantage of the binary tree structure of a NUTS trajectory. This is described in Appendix 5.B

When we are not constrained by memory, the following Rao-Blackwellized version of recycled NUTS allows us to simply collect and use all the acceptable states of each NUTS iteration by assigning appropriate weights.

Algorithm 5.3 (Rao-Blackwellized recycled NUTS)**.**
Let $\mathcal{A}_i = \{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 1, \ldots, |\mathcal{A}_i|\}$ denote the collection of acceptable states from the i-th iteration of NUTS. Return the samples $\{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 1, \ldots, |\mathcal{A}_i|\}$ with weight $\propto |\mathcal{A}_i|^{-1}$ for $i = 1, \ldots, N$ as the draws from the target distribution, yielding an empirical measure:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{A}_i|}\sum_{k=1}^{|\mathcal{A}_i|}\delta_{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})}(\cdot)$$

The validity of Algorithm 5.3 follows simply by taking an expectation over the sampling step $(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}) \sim \text{Uniform}(\mathcal{A}_i)$ of Algorithm 5.2.

5.5 Numerical results

We take the test cases from Hoffman and Gelman (2014). In all our simulations we chose the stepsizes ϵ such that the corresponding average acceptance rates are approximately 70%, as values between 60% and 80% are typically considered optimal (Neal, 2010; Beskos et al., 2013; Hoffman and Gelman, 2014). The dual averaging algorithm of Hoffman and Gelman (2014) was used to find such stepsizes. The choice of path lengths $\tau^{(i)} = \epsilon L^{(i)}$ for HMC is discussed within the individual test cases below. Also, the identity mass matrix was used in all our simulations except when investigating the use of recycling in mass matrix tuning (see below).

In comparing the algorithms with and without recycling, we use effective sample sizes (ESS) as a commonly used measure of efficiency of Monte Carlo algorithms (Geyer, 2011). The standard definition of ESS applies only to estimators of the form $N^{-1}\sum_{i=1}^{N} g(\boldsymbol{\theta}^{(i)})$ for a real-valued function g, so we extend the standard definition to a more complex estimator $G : \{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N} \to \mathbb{R}$ of a quantity $\mathbb{E}[g(\boldsymbol{\theta})]$ by defining

$$\mathrm{ESS}_G\left(\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N}\right) = N\frac{\mathrm{MSE}\left(G\left(\left\{\boldsymbol{\theta}^{*(i)} \overset{\text{i.i.d.}}{\sim} \pi(\cdot)\right\}\right)\right)}{\mathrm{MSE}\left(G(\{\boldsymbol{\theta}^{(i)}\})\right)} \tag{5.9}$$

where $\mathrm{MSE}(\cdot)$ denotes the mean squared error of an estimator. The definition (5.9) agrees with the standard one when $G\left(\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N}\right) = N^{-1}\sum_{i=1}^{N} g(\boldsymbol{\theta}^{(i)})$. Additional computer time incurred by recycling is typically insignificant (e.g. $1 \sim 6\%$ in our NUTS examples implemented in MATLAB), so the comparison in terms of ESS practically accounts for computational time.

In our simulation, we also study the relationship between the number of recycled samples and statistical efficiency, in order to demonstrate that it is not necessary to recycle all the intermediate steps to reap the benefit of recycling. This is relevant in a high dimensional parameter space where the amount of memory required to store the extra samples becomes substantial.[2] For long trajectories, there is substantial correlation among the intermediate states and we can expect that recycling a subset of the intermediate states would provide as much statistical efficiency as recycling all. To quantify this, we first ran the algorithm recycling all the intermediate states. We then repeatedly reduced the number of samples per iteration M (recycled intermediate states plus the final state) by a factor of 2. The results presented for our examples are based on the smallest M for which the ESS averaged across all the estimators is within 5% of that when recycling all the intermediate states. Section 5.5.4 investigates

[2] In the stochastic volatility model of Section 5.5.3, for example, it requires 4GB of memory to store 100 extra samples per iteration from a Markov chain of length 3,200 in a 3000-dimensional parameter space.

in more detail the relationship between the statistical efficiency and the number of recycled samples.

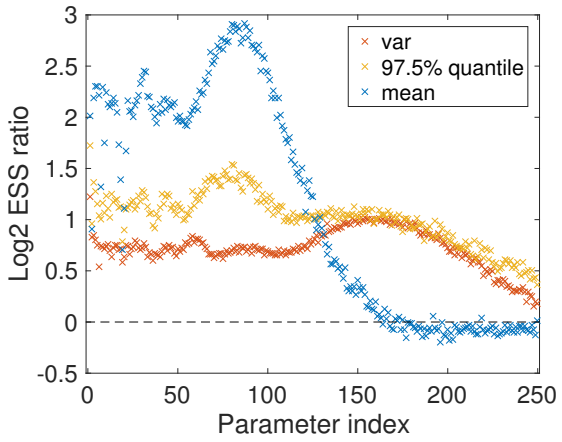
Finally, we investigate the utility of recycling during the tuning phase of HMC / NUTS, in which a covariance matrix of the target $\pi(\cdot)$ is estimated. Such use of recycling requires no extra memory, and a good covariance estimator $\widehat{\boldsymbol{\Sigma}}$ can enhance both the speed of one iteration as well as the mixing rate of HMC / NUTS later on by setting the mass matrix $\boldsymbol{M} = \widehat{\boldsymbol{\Sigma}}^{-1}$ (Stan Development Team, 2016; Neal, 2010).

5.5.1 Multivariate Gaussian

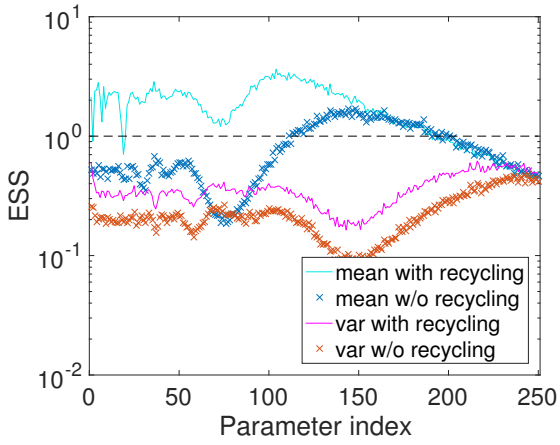
The first test case is sampling from a 250-dimensional multivariate Gaussian $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is drawn from a Wishart distribution with 250 degrees of freedom and mean equal to the identity matrix. A covariance matrix drawn from this distribution exhibits strong correlations, and in our case the ratio between the largest and smallest eigenvalue of $\boldsymbol{\Sigma}$ was approximately 9.5×10^4. Since HMC and NUTS with the identity mass matrix are invariant under rotations, for convenience we assume that $\boldsymbol{\Sigma}$ is diagonal with $\Sigma_{i,i} = \sigma_i^2$, where σ_i^2 corresponds to the ith smallest eigenvalue of the original covariance matrix. For the path length of HMC, we first found the smallest value of τ for which the samples in the leading principal component direction are roughly independent. The typical practice would be then to jitter $\tau^{(i)}$'s within the range $[0.9\,\tau, 1.1\,\tau]$ to avoid periodicity (Neal, 2010), but this still resulted in near perfect periodicity and hence poor mixing for some parameters. After some experiments, we found jittering $\tau^{(i)}$ in the range $[\tau/2, \tau]$ to provide decent mixing along all the coordinates.

We simulated 800 independent Markov chains of length 1600 starting from stationarity. We then computed the MSE in Monte Carlo estimates of the mean, variance, and 97.5% quantile along each dimension. Fig. 5.2a shows $\log_2$ of the ratios between ESS of HMC with and without recycling, calculated from the MSE using the relation

(5.9). Values above zero indicate superior performance of our recycling algorithm. Recycling uniformly and substantially improves on estimating variance and quantiles: about 100% increase in ESS on average. Though the mean estimates for parameters with larger variances are not improved, Fig. 5.2b clearly demonstrates gains in the worst case performance. Out of 251 recyclable samples generated on average from each iteration of HMC, we recycled $251/8 \approx 31$ samples.



(a) $\log_2$ ratios of ESS with recycling (numerator) and without recycling (denominator). The horizontal line at zero corresponds to no gain from recycling. The x-axis corresponds to different parameters.



(b) ESS per HMC step for the first and second moment estimators. The y-axis is in $\log_{10}$ scale.

FIGURE 5.2: Performance comparison between HMC with and without recycling in estimating mean, variance, and quantiles for the Gaussian example.

We were also interested in whether recycling helps estimate the covariance structure of the target distribution. To investigate this, we computed the top eigenvalue and eigenvector of the empirical covariance matrix for each chain. We then calculated the angle between the empirical eivenvector and the plane spanned by the ℓ true leading principal components. This angle should be close to 0 when the eigenvector is estimated well. To ensure identifiability of the direction, we chose $\ell = \min\{j : \sigma_j^2 < \sigma_1^2/2\}$ in all our simulations, where σ_j^2 denotes the jth largest eigenvalue of the true covariance matrix. The ratios of ESSs in estimating the angle as well as the

eigenvalues are shown in Fig. 5.3. We plotted the ratios against the length of Markov chains. The direction of the principal component cannot be well estimated by shorter chains of lengths ~ 200 even with recycling, but recycling conveys a substantial advantage as the chains are run longer.

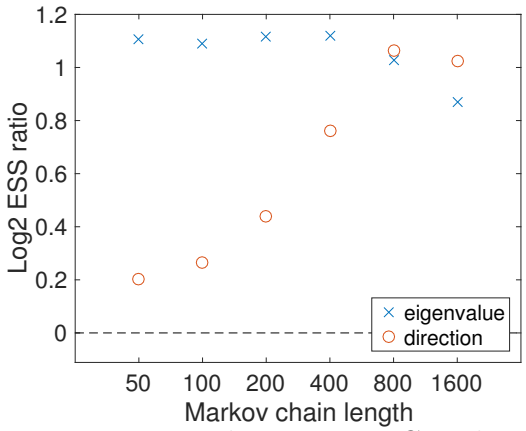


FIGURE 5.3: Performance comparison between HMC with and without recycling in estimating the direction and magnitude of the leading principal component for the covariance matrix in the Gaussian example.

The relative performance of NUTS with and without recycling is similarly summarized in Fig. 5.4. The average trajectory length was $2^9 = 512$, out of which $2^4 - 1 = 15$ samples were recycled.

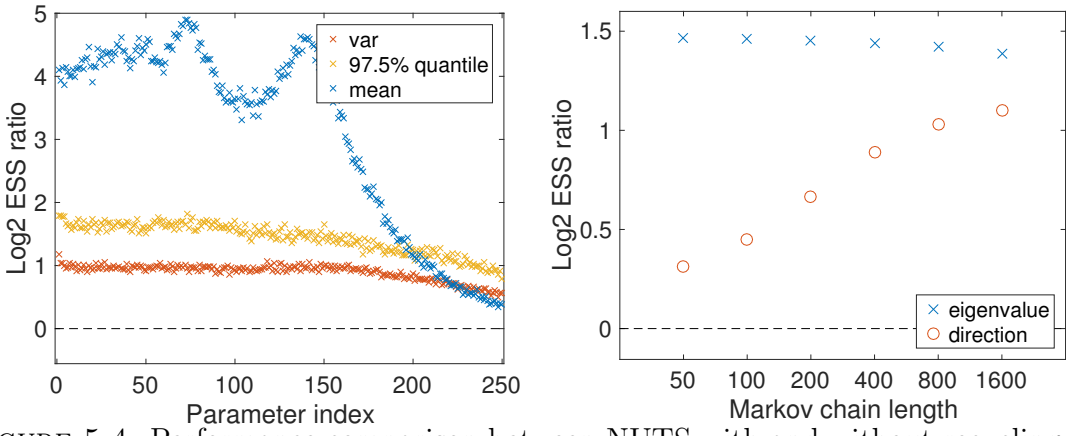


FIGURE 5.4: Performance comparison between NUTS with and without recycling for the Gaussian example.

Our last experiment explores the use of recycling in tuning the NUTS mass matrix

$\boldsymbol{M} = \widehat{\boldsymbol{\Sigma}}^{-1}$ with a covariance estimator $\widehat{\boldsymbol{\Sigma}}$. To this end, we tune the mass matrix while running NUTS with and without recycling, and then run two independent chains with the two different mass matrices to compare their ESSs. Recycling is only applied during the tuning phase for covariance estimation. This experiment also serves as an alternate and more holistic evaluation of covariance estimation with and without recycling. Aside from some simplifications, our experimental set-up closely follows the default settings of Stan for tuning the stepsize and mass matrix (Stan Development Team, 2016). First, 50 iterations of the dual-averaging algorithm are run to tune the stepsize with the identity mass matrix, followed by N_{adap} iterations with a fixed stepsize to estimate the covariance matrix, and finally another 75 iterations of dual-averaging to re-adjust the stepsize with the tuned mass matrix. After the covariance estimation phase with N_{adap} iterations, we set $\boldsymbol{M}^{-1} = \widehat{\boldsymbol{\Sigma}}$ where

$$\widehat{\boldsymbol{\Sigma}} = \frac{N_{\text{adap}}}{5 + N_{\text{adap}}} \widehat{\boldsymbol{\Sigma}}_{\text{emp}} + \frac{5}{5 + N_{\text{adap}}} 10^{-3} \cdot \boldsymbol{I} \tag{5.10}$$

with $\widehat{\boldsymbol{\Sigma}}_{\text{emp}}$ the empirical covariance matrix and $\boldsymbol{I}$ the identity matrix. After the tuning phase, we run NUTS until the total number of gradient evaluations reaches 10^4. This procedure is repeated 800 times and the ESS for each statistic is averaged across the repetitions.

Figure 5.5 shows the ratio of average ESS with and without recycling during the covariance estimation phase for $N_{\text{adap}} = 400$. Again, in this experiment recycling is only carried out during the tuning phase and the difference in ESS comes purely from difference in mass matrix parameters. The benefit of recycling diminishes as N_{adap} increases as the covariance matrix can be adequately approximated without recycling and we found no advantage of recycling when $N_{\text{adap}} \geqslant 800$.

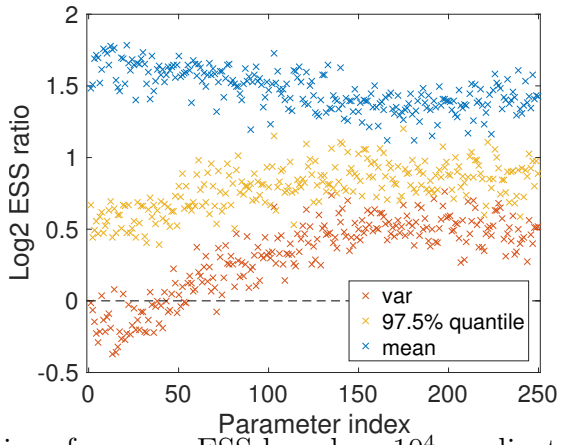


FIGURE 5.5: $\log_2$ ratios of average ESS based on 10^4 gradient evaluations when the mass matrix is tuned with and without recycling for the Gaussian example.

5.5.2 Hierarchical Bayesian logistic regression

The second test case is a hierarchical Bayesian logistic regression model applied to the German credit data set available from the University of California Irvine Machine Learning Repository. Including two-way interaction terms and an intercept, there are 301 predictors and the regression coefficients $\boldsymbol{\beta}$ are given a $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ prior. A hyper-prior is placed on σ^2, which makes the posterior inference more challenging through the strong dependence between σ and $\boldsymbol{\beta}$. We made one modification to the corresponding example in Hoffman and Gelman (2014) by defining our parameters to be $(\log(\sigma), \boldsymbol{\beta})$ instead of $(\sigma^2, \boldsymbol{\beta})$ since such a transformation of constrained variables has become standard (Stan Development Team, 2016). A default flat prior was placed on σ.

The 800 independent chains were run for 3200 iterations starting from stationarity. In computing the ESSs, the statistics from an independent chain of 10^7 NUTS iterations after 10^3 burn-in samples were used as the ground truth.

A performance comparison as in Section 5.5.1 is shown in Fig. 5.6, 5.7, and 5.8. To facilitate the comparison of algorithms with and without recycling, the parameters are sorted in increasing order of the ESS ratios in mean estimation. For some parameters,

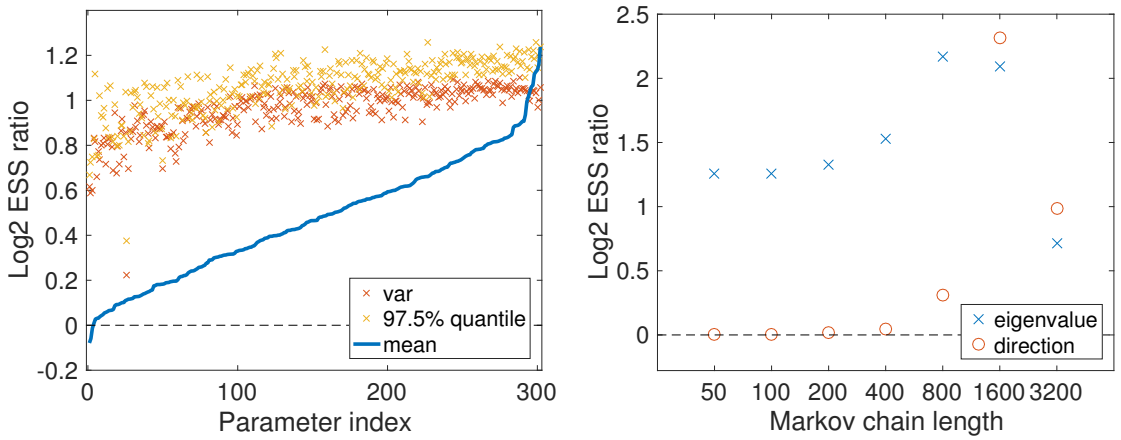


FIGURE 5.6: Performance comparison between HMC with and without recycling for the hierarchical logistic model.

recycling seems to produce little gains in terms of mean estimation but provides clear benefits in terms of variance and quantile estimation. In the mass matrix tuning experiment shown in Figure 5.8, we tried $N_{\text{adap}} = 500, 1000, 2000$ and observed substantial improvement in the average ESS from recycling for $N_{\text{adap}} \leqslant 1000$.

For the path lengths for HMC, we first found the value τ to maximize the normalized expected square jumping distance $\tau^{-1/2}\mathbb{E}\|\boldsymbol{\theta}^{(i+1)}(\tau) - \boldsymbol{\theta}^{(i)}(\tau)\|$ as in Wang et al. (2013), then jittered each path length $\tau^{(i)}$ in the range $[0.9\,\tau, 1.1\,\tau]$. The average trajectory length of HMC was 9 and all the intermediate states were recycled. The average trajectory length of NUTS was $2^4 = 16$, out of which 7 were recycled.

5.5.3 Stochastic volatility model

The last test case is a stochastic volatility (SV) model fit to a time series y taken from the closing values of S&P 500 index for 3000 days ending on Dec 31st, 2015. The model is specified as follows:

$$\log\left(\frac{y_i}{y_{i-1}}\right) \sim \mathcal{N}(0, s_i^2),\ \ 100\log\left(\frac{s_i}{s_{i-1}}\right) \sim \mathcal{N}(0, \tau^{-1})$$

with priors $s_0 \sim \text{Exp}(\text{mean} = 1/10)$ and $\tau \sim \text{Gamma}\,(1/2, 1/2)$. The observed value on Jan 2nd, 2008 was removed from the original data as this simple SV model could

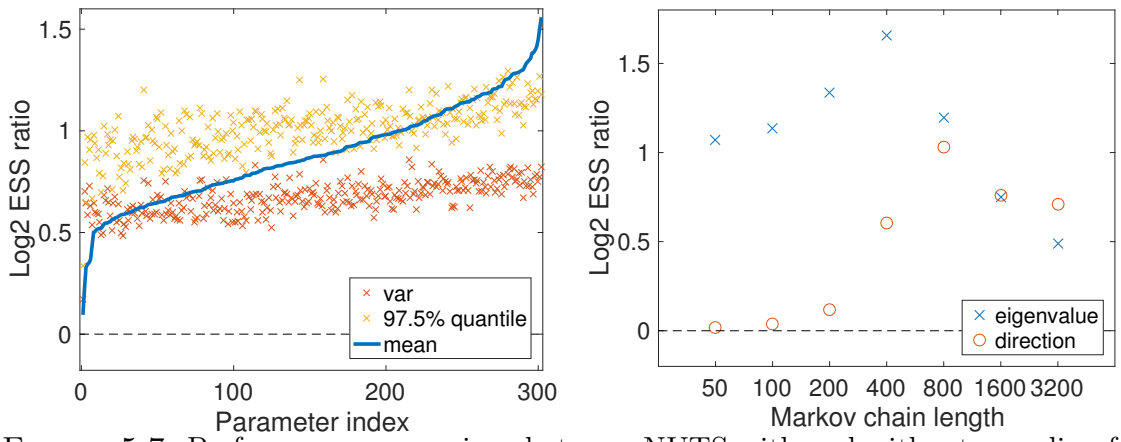


FIGURE 5.7: Performance comparison between NUTS with and without recycling for the hierarchical logistic model.

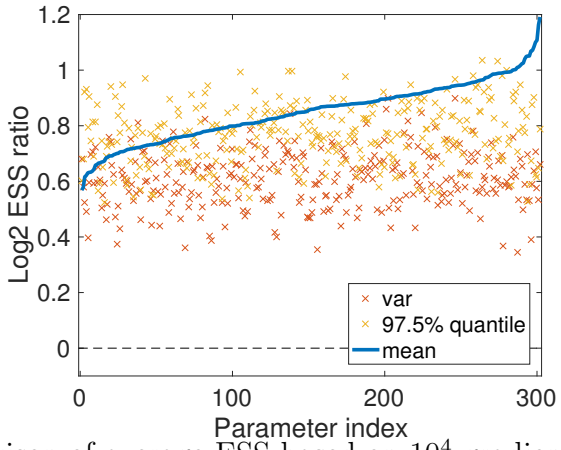


FIGURE 5.8: Comparison of average ESS based on 10^4 gradient evaluations between NUTS with a mass matrix tuned with and without recycling for $N_{\text{adap}} = 500$ in the hierarchical logistic model.

not fit this observation well. The model is identical to the one in Hoffman and Gelman (2014) except for minor changes to simplify the analytical formula of posterior density. After integrating out τ to accelerate mixing, we are left with a 3000 dimensional parameter space for $\log \boldsymbol{s}$.

A performance comparison is shown in Fig. 5.9 and 5.10 with the parameters sorted according to the ESS ratios for mean estimation as in Section 5.5.2. The 400 independent chains were run for 3200 iterations starting from stationarity. In computing the ESSs, the statistics from an independent NUTS chain of length 2.5×10^6

after 10^3 burn-in were used as the ground truth. The path length for HMC was chosen as in Section 5.5.2. The mass matrix tuning experiment was not carried out for this example as tuning a mass matrix for a 3000 dimensional space is impractical. On average, 44 samples out of 90 per iteration were recycled for HMC and 7 out of $2^7 = 128$ were recycled for NUTS.

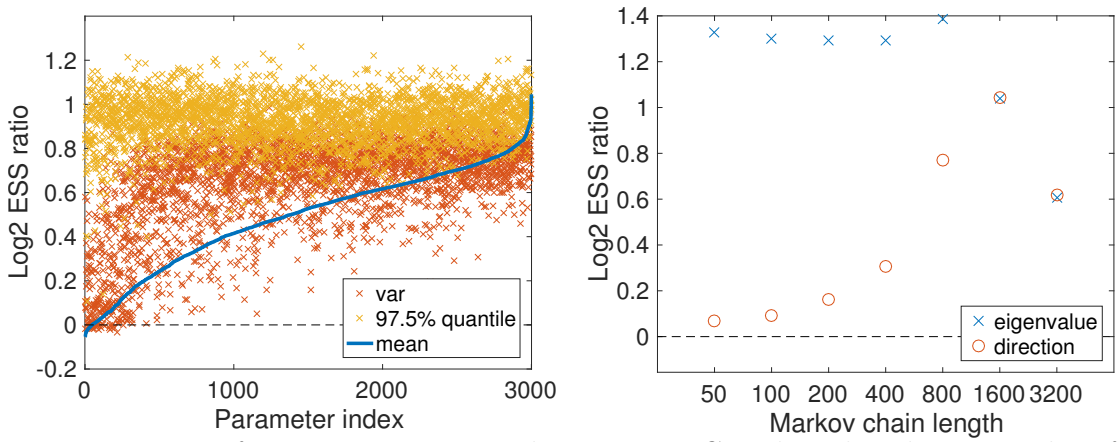


FIGURE 5.9: Performance comparison between HMC with and without recycling for the SV model.

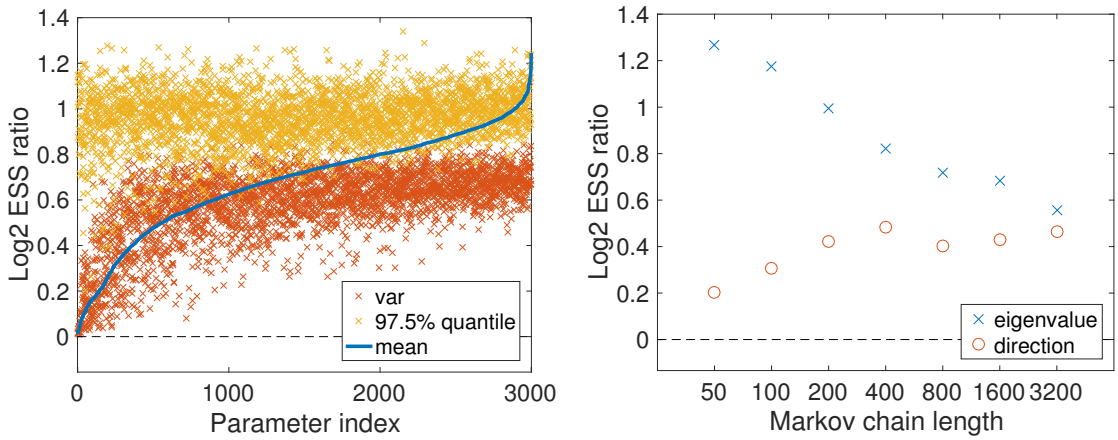


FIGURE 5.10: Performance comparison between NUTS with and without recycling for the SV model.

5.5.4 Number of recycled samples and statistical efficiency

As mentioned earlier, in the simulation results above we recycle enough of the intermediate states to achieve near-optimal efficiency gains. Here we take a closer look at how the efficiency gain from recycling depends on the number of recycled samples. Our results here in particular provide a practical guidance on how one might trade off statistical efficiency for memory efficiency when needed.

For our experiments here, we focus on the problem of estimating a quantile; the dependence of mean and variance estimators on the number of recycled samples was found to be similar. The number of samples per iteration (recycled states plus the final state) was repeatedly reduced by a factor of 2 until the benefit of recycling became almost negligible. The results are summarized in the $\log_2$ ESS ratio plots as presented earlier; Figure 5.11 for the multi-variate Gaussian example, Figure 5.12 for the hierarchical Bayesian logistic regression example, and Figure 5.13 for the stochastic volatility example. The parameter indices are sorted in the increasing order of the ESS ratio at the largest number of recycled samples (the dark solid line). The green dotted line corresponds to the number of recycled samples at which the efficiency decrease relative to the optimal one becomes visually noticeable. The cyan dashed line corresponds to the number of recycled samples below which the benefit from recycling becomes negligible.

The performance of recycled NUTS is particularly remarkable, not only offering the near-optimal efficiency gain well-below the maximal recycling size but also demonstrating over 40% ($\approx 2^{0.5}$) efficiency gain with just one recycled sample. For recycled HMC, the efficiency gains remain substantial well-below the maximal recycling size but start to diminish much earlier than NUTS. Two design features of NUTS likely explain this phenomenon. First, NUTS simulates a trajectory in both the forward and backward direction, which means that some of the intermediate

states lie in the direction opposite to the final proposal state relative to the starting point of a trajectory. Secondly, while HMC simulates a trajectory to construct one high-quality proposal state, NUTS generates a collection of states — any of which likely constitutes a good proposal state — and select one state from the collection as a final proposal. These two features of NUTS suggest that, compared to those of HMC, the recyclable states of NUTS individually have smaller correlations with the final proposal state. Even if the efficiency gain is comparable between HMC and NUTS when recycling all the intermediate states, it seems that the smaller pair-wise correlations of recyclable states with the final state provides NUTS with a greater benefit when recycling a small subset.

It is worth noting that NUTS is actually a meta-algorithm that provides a useful trajectory termination criterion for any MCMC algorithm based on reversible dynamics. NUTS and our recycled version therefore apply straightforwardly to most of the HMC variants mentioned in this paper. The U-turn condition of NUTS can be adjusted to suit particular objectives as illustrated in Betancourt (2013). Our experiments here suggest that recycled NUTS may be a particularly practical alternative to the standard implementation of HMC-type algorithms; it not only eliminates the need to tune the path length but also provides a significant boost in efficiency with a rather small increase in memory requirement.

5.6 Discussion

We have proposed a simple and general algorithm for improving the efficiency of HMC and variants with essentially no extra computational overhead. The trade-off between statistical and memory efficiency was also addressed as it is an important scalability issue. Our simulations demonstrate the substantial gains in computational efficiency without excessive memory use. In practice, conceptual complexity, ease of implementation, and memory efficiency are just as important considerations as

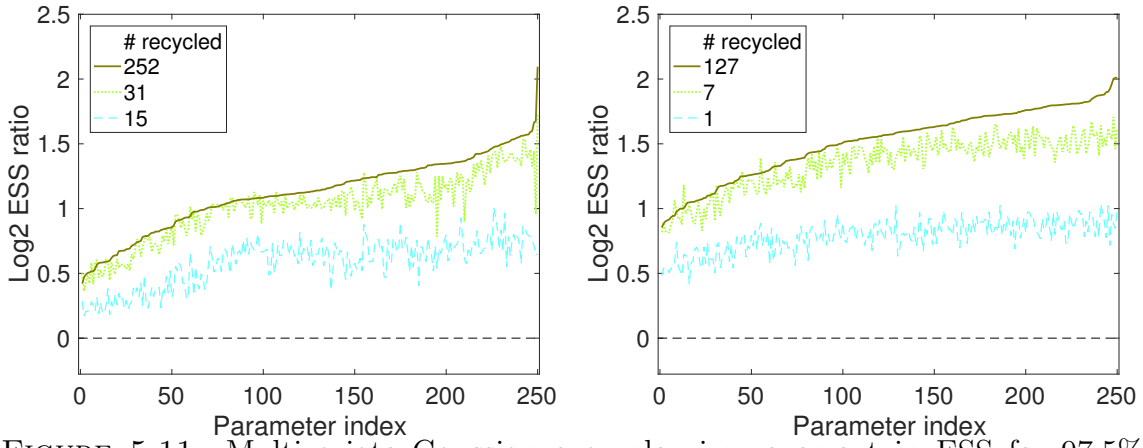


FIGURE 5.11: Multivariate Gaussian example: improvement in ESS for 97.5% quantile estimation with different number of recycled samples.

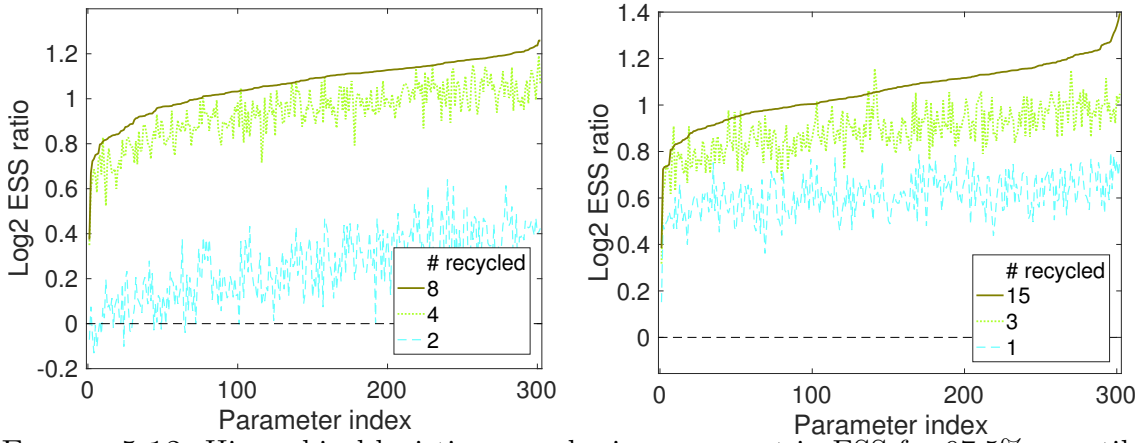


FIGURE 5.12: Hierarchical logistic example: improvement in ESS for 97.5% quantile estimation with different number of recycled samples.

statistical efficiency, which likely explains why related ideas to improve the efficiency of HMC variants have not gained traction. Our algorithm provides a more practical and user-friendly alternative that applies straightforwardly to a wide range of multi-proposal schemes.

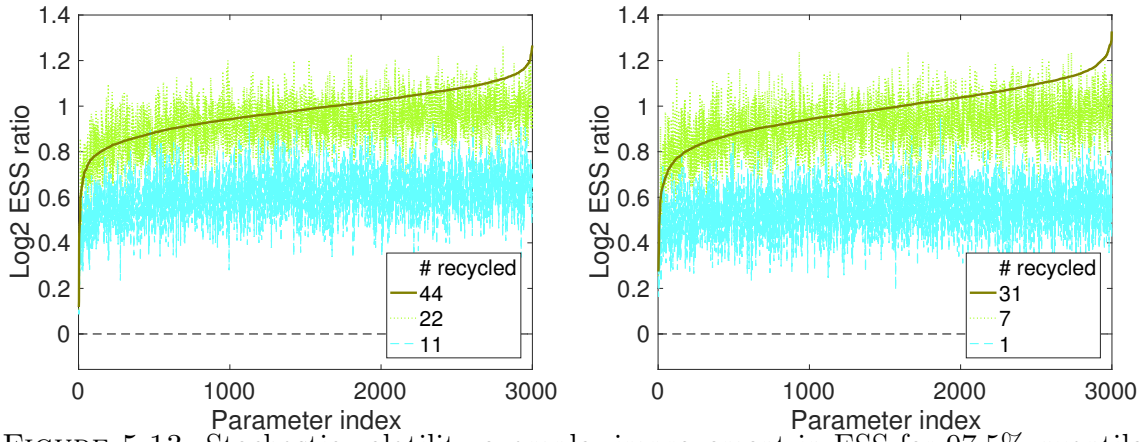


FIGURE 5.13: Stochastic volatility example: improvement in ESS for 97.5% quantile estimation with different number of recycled samples.

Appendix for Chapter 5

5.A Proofs of Theorem 5.3 and 5.4

Theorem 5.3. A stationary distribution $\pi^*(\cdot)$ of the Markov chain $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots$ is given by

$$\pi^*(\cdot) = \int P(\,\cdot \,|\, \boldsymbol{z}_0)\pi(\boldsymbol{z}_0)\,\mathrm{d}\boldsymbol{z}_0 \tag{5.11}$$

By the assumption (5.5), the marginal $\pi^*(\boldsymbol{z}_k)$ coincides with $\pi(\boldsymbol{z}_k)$ for all $k = 0, \ldots, M$. Once we establish the unique ergodicity of the chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$, therefore, the conclusion (5.6) follows by averaging the coordinates $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ of the empirical measure $N^{-1}\sum_{i=1}^N \delta_{z^{(i)}}$. Suppose $\tilde{\pi}^*(\cdot)$ is another stationary measure of $P(\cdot \,|\, \cdot)$. This means that, by the assumption (5.4),

$$\tilde{\pi}^*(\cdot) = \int P(\,\cdot \,|\, \boldsymbol{z}_0)\tilde{\pi}^*(\boldsymbol{z}_0)\,\mathrm{d}\boldsymbol{z}_0. \tag{5.12}$$

In particular, the marginal $\tilde{\pi}^*(\boldsymbol{z}_0^*)$ satisfies $\tilde{\pi}^*(\boldsymbol{z}_0^*) = \int P_0(\boldsymbol{z}_0^* \,|\, \boldsymbol{z}_0)\tilde{\pi}^*(\boldsymbol{z}_0)\,\mathrm{d}\boldsymbol{z}_0$ by the assumption (5.5). The unique ergodicity of $P_0(\cdot \,|\, \cdot)$ then implies $\tilde{\pi}^*(\boldsymbol{z}_0) = \pi(\boldsymbol{z}_0)$. Substituting this equality into (5.12) establishes $\tilde{\pi}^*(\cdot) = \pi^*(\cdot)$ and hence the unique ergodicity of the chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$.

We turn to the proof of a convergence rate (geometric or uniform ergodicity) of the chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$ under the corresponding assumption on $P_0(\cdot \,|\, \cdot)$. For the conditional distribution of $\boldsymbol{z}^{(n)} \,|\, \boldsymbol{z}_0^{(0)}$, we have

$$\left|\mathbb{P}(\boldsymbol{z}^{(n)} \in A \,|\, \boldsymbol{z}_0^{(0)}) - \pi^*(A)\right| = \left|\int P(A \,|\, \boldsymbol{z}_0')\left(P_0^n(\boldsymbol{z}_0' \,|\, \boldsymbol{z}_0^{(n)}) - \pi(\boldsymbol{z}_0')\right)\mathrm{d}\boldsymbol{z}_0'\right| \tag{5.13}$$

It follows that

$$\left\|\mathbb{P}(\boldsymbol{z}^{(n)} \in \cdot \,|\, \boldsymbol{z}_0^{(0)}) - \pi^*(\cdot)\right\|_{\text{tv}} \leqslant \left\|P_0^n(\,\cdot \,|\, \boldsymbol{z}_0^{(n)}) - \pi(\cdot)\right\|_{\text{tv}}$$

where $\|\cdot\|_{\text{tv}}$ denotes a total variation norm. Hence the chain $\{\boldsymbol{z}^{(i)}\}_{i\geqslant 1}$ inherits the convergence rate of $P_0(\cdot\,|\,\cdot)$. □

Theorem 5.4. Re-write the empirical measure in (5.7) as

$$\frac{1}{M}\sum_{k=1}^{M} \frac{1}{N^{-1}\big(\sum_{i=1}^{N}|S^{(i)}|\big)} \left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{k \in S^{(i)}\}\, \delta_{\boldsymbol{z}_k^{(i)}}(\cdot)\right) \tag{5.14}$$

The law of iterated expectations $\mathbf{E}[\,\cdot\,] = \mathbf{E}[\mathbf{E}\{\,\cdot\,|\,\mathcal{S}\}]$ for $\mathcal{S} = (S^{(1)}, S^{(2)}, \ldots)$ implies that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{k \in S^{(i)}\}\, \delta_{\boldsymbol{z}_k^{(i)}}(\cdot) = \lim_{N\to\infty} \frac{\mathbb{P}(k \in S^{(1)})}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{z}_k^{(i)}}(\cdot) \tag{5.15}$$

where the limit denotes the convergence in distribution. Also, we have the almost sure convergence

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} |S^{(i)}| = \mathbb{E}|S^{(1)}| = \sum_{\ell=1}^{M} \mathbb{P}(\ell \in S^{(1)}) \tag{5.16}$$

From (5.14), (5.15), and (5.16), it follows that the empirical measure in (5.7) has the same limiting distribution as the following sequence of measures:

$$\frac{1}{M}\sum_{k=1}^{M} \frac{\mathbb{P}(k \in S^{(1)})}{\sum_{\ell=1}^{M}\mathbb{P}(\ell \in S^{(1)})} \left(\frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{z}_k^{(i)}}(\cdot)\right) \tag{5.17}$$

We know from the proof of Theorem 5.3 that $\nu_{k,N}(\cdot) = N^{-1}\sum_{i=1}^{N}\delta_{\boldsymbol{z}_k^{(i)}}(\cdot)$ converges to $\pi(\cdot)$ for all $k = 1, \ldots, M$. The measure (5.17) is simply a weighted average of $\nu_{k,N}$'s and therefore converges to $\pi(\cdot)$. □

5.B Efficient recycled NUTS

Rao-Blackwellized recycled NUTS of Algorithm 5.3 is statistically efficient, but requires all the intermediate states. As an alternative, we here describe a modification of Algorithm 5.2 which improves statistical efficiency without increasing the number of recycled states by ensuring that the recycled samples are evenly spread across a NUTS trajectory $\mathcal{T}(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})$. As mentioned in Section 5.4, we can take advantage of the binary tree structure of $\mathcal{T}(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})$ in order to implement it in a simple and memory efficient manner. To explain the main idea, suppose that an iteration of NUTS from $(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})$ takes $2^d - 1$ leapfrog steps, generating a binary tree of depth d which we denote by $\mathcal{T}_d$. Let $\mathcal{T}_{d-j,\ell}$ and $\mathcal{A}_{d-j,\ell} = \mathcal{A}(\mathcal{T}_{d-j,\ell})$ for $\ell = 1, \ldots, 2^j$ denote the subtrees of depth $d-j$ and the collections of acceptable states within each subtree. At the depth $d-j$, we assign the minimal number of samples from the subtree $\mathcal{T}_{d-j,\ell}$ to be

$$\left\lfloor M \frac{|\mathcal{A}_{d-j,\ell}|}{\sum_\ell |\mathcal{A}_{d-j,\ell}|} \right\rfloor$$

where $\lfloor \cdot \rfloor$ denotes a floor function. Enforcing this recursively at each depth $d-j$ for $j = 1, \ldots, d$ ensure that the recycled states are evenly spread along the trajectory. An actual procedure is described in Algorithm 5.4 below. In order to avoid references to the algorithm implementation details of NUTS, we describe how to carry out the recycling procedure assuming we store all the intermediate states and its binary tree structure during each NUTS iteration. It is however easy to add the recycling algorithm on top of the NUTS implementation of Hoffman and Gelman (2014) so that no more than M states are stored in memory during each NUTS iteration.

Algorithm 5.4 (Recycled NUTS)**.** Run NUTS to generate a sequence of random variables $\{(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\}_{i \geqslant 1}$. Additionally at each iteration of NUTS, recycle variables

$\{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 1, \ldots, M\}$ from the collection of acceptable states $\mathcal{A}(\boldsymbol{\theta}_0^{(i-1)}, \boldsymbol{p}_0^{(i-1)})$ by calling the function RECYCLE below:

---

function RECYCLE($\mathcal{A}$, M)
  if depth($\mathcal{A}$) = 0 **then** ▷ *A is a singleton set*
    return M copies of the variable from $\mathcal{A}$
  else
    let $\mathcal{A}'$ and $\mathcal{A}''$ be the left and right subtree of $\mathcal{A}$
    $n \leftarrow \lfloor w \rfloor + \text{Bernoulli}(w - \lfloor w \rfloor)$ for $w = M/|\mathcal{A}'|$
    $\{(\boldsymbol{\theta}_k, \boldsymbol{p}_k)\}_{k=1}^{n} \leftarrow \text{Recycle}(\mathcal{A}', n)$
    $\{(\boldsymbol{\theta}_k, \boldsymbol{p}_k)\}_{k=n+1}^{M} \leftarrow \text{Recycle}(\mathcal{A}'', M - n)$
    return $\{(\boldsymbol{\theta}_1, \boldsymbol{p}_1), \ldots, (\boldsymbol{\theta}_M, \boldsymbol{p}_M)\}$
  end if
end function

5.C Simple proof of algorithm by Calderhead and Bernton et. al.

Here we describe how Theorem 5.3 provides an alternative and simpler proof for a version of the algorithms by Calderhead (2014) and Bernton et al. (2015). The proof in particular requires no understanding of the super-detailed balance condition (Frenkel, 2004; Tjelmeland, 2004). The algorithms below are presented as "Version 2" of modified Calderhead's algorithms in Bernton et al. (2015). As before, the map $\boldsymbol{F}_\epsilon : (\boldsymbol{\theta}_0, \boldsymbol{p}_0) \to (\boldsymbol{\theta}_1, \boldsymbol{p}_1)$ corresponds to one leap-frog step with stepsize ϵ.

Algorithm 5.5 (Calderhead and Bernton et. al.)**.** Generate a Markov chain $\{(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\}_{i \geqslant 1}$ with the transition rule $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)})$ as follows:

1. Sample $L^{(i)} \sim \text{Uniform}(\{0, 1, \ldots, M\})$. Set $\ell = M - L^{(i)}$ if $M - L^{(i)} \geqslant L^{(i)}$ and $\ell = -L^{(i)}$ otherwise.
2. Set $(\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)}) = \boldsymbol{F}_\epsilon^\ell(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})$ with probability

$$\min\left\{1, \frac{\pi\big(\boldsymbol{F}_\epsilon^\ell(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\big)}{\pi\big((\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})\big)}\right\} \tag{5.18}$$

and $(\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)}) = (\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)})$ otherwise.

3. Generate a new momentum: $\boldsymbol{p}_0^{(i+1)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{M})$.

Additionally at each iteration, generate $\{(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)}), k = 1, \ldots, M\}$ as follows:

4. Define a collection of states

$$\mathcal{A}_{i+1} = \mathcal{A}(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}, L^{(i)}) = \{F^k(\boldsymbol{\theta}^{(i)}, \boldsymbol{p}^{(i)}), k = -L^{(i)}, \ldots, M - L^{(i)}\} \tag{5.19}$$

Sample $(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)})$'s by independently setting $(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)}) = (\boldsymbol{\theta}^*, \boldsymbol{p}^*) \in \mathcal{A}_{i+1}$ with probability

$$w_{i+1}(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = \frac{\pi(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\sum_{(\boldsymbol{\theta}, \boldsymbol{p}) \in \mathcal{A}_{i+1}} \pi(\boldsymbol{\theta}, \boldsymbol{p})}. \tag{5.20}$$

Taking an expectation over the sampling procedure of $\{(\boldsymbol{\theta}_k^{(i+1)}, \boldsymbol{p}_k^{(i+1)}), k = 1, \ldots, M\}$ in Step 4 above, we obtain the Rao-Blackwellized version of Algorithm 5.5.

Algorithm 5.6 (Rao-Blackwellization of Algorithm 5.5)**.** Given the collection of states $\mathcal{A}_i$ with the weights w_i as in (5.19) and (5.20), return the weighted empirical measure

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{(\boldsymbol{\theta}^*, \boldsymbol{p}^*) \in \mathcal{A}_i} w_i(\boldsymbol{\theta}^*, \boldsymbol{p}^*) \delta_{(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}(\cdot)$$

as a Monte Carlo estimate of the target distribution.

Proof of Validity of Algorithm 5.5. We will establish the weak convergence

$$\frac{1}{NM} \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)})}(\cdot) \xrightarrow{w} \pi(\cdot) \text{ as } N \to \infty. \tag{5.21}$$

for the samples $\{(\boldsymbol{\theta}_k^{(i)}, \boldsymbol{p}_k^{(i)}), k = 1, \ldots, M\}$ generated as in Algorithm 5.5.

Let $P_0(\cdot \,|\, \cdot)$ denote the transition kernel corresponding to the transition rule $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_0^{(i+1)}, \boldsymbol{p}_0^{(i+1)})$ as in Step 1–3 of Algorithm 5.5. Also let $P_1(\cdot \,|\, \cdot) =$

$\ldots = P_M(\cdot \,|\, \cdot)$ denote the kernel corresponding to the transition rule $(\boldsymbol{\theta}_0^{(i)}, \boldsymbol{p}_0^{(i)}) \to (\boldsymbol{\theta}_1^{(i+1)}, \boldsymbol{p}_1^{(i+1)})$ as in Step 4 of Algorithm 5.5. By virtue of Theorem 5.3, we simply need to verify that $\pi(\cdot)$ is the stationary distribution of the transition kernels $P_0(\cdot \,|\, \cdot)$ and $P_1(\cdot \,|\, \cdot)$. The kernel $P_0(\cdot \,|\, \cdot)$ represents the transition rule of HMC with a randomized number of leap-frog steps and hence is reversible with respect to $\pi(\cdot)$. The reversibility of $P_1(\cdot \,|\, \cdot)$ also follows from the standard HMC theory; the only additional observation needed for the proof is the following "symmetry" in the collection of proposed states at Step 4. For $L \sim \text{Uniform}(\{0, 1, \ldots, M\})$ and a pair of states $(\boldsymbol{\theta}, \boldsymbol{p})$ and $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$, the following conditional distributions of random sets $\mathcal{A}(\boldsymbol{\theta}, \boldsymbol{p}, L)$ and $\mathcal{A}(\boldsymbol{\theta}^*, \boldsymbol{p}^*, L)$ as defined in (5.19) coincide:

$$\mathcal{A}(\boldsymbol{\theta}, \boldsymbol{p}, L) \,|\, (\boldsymbol{\theta}^*, \boldsymbol{p}^*) \in \mathcal{A}(\boldsymbol{\theta}, \boldsymbol{p}, L) \stackrel{d}{=} \mathcal{A}(\boldsymbol{\theta}^*, \boldsymbol{p}^*, L) \,|\, (\boldsymbol{\theta}, \boldsymbol{p}) \in \mathcal{A}(\boldsymbol{\theta}^*, \boldsymbol{p}^*, L) \qquad \square$$

Bibliography

Abraham, R. and Marsden, J. E. (1978), *Foundations of mechanics*, vol. 36, Addison-Wesley Publishing Company.

Afshar, H. M. and Domke, J. (2015), "Reflection, Refraction, and Hamiltonian Monte Carlo," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 3007–3015.

Alder, B. J. and Wainwright, T. E. (1959), "Studies in Molecular Dynamics. I. General Method," *The Journal of Chemical Physics*, 31, 459–466.

Amari, S. and Nagaoka, H. (2007), *Methods of Information Geometry*, vol. 191, American Mathematical Society.

Ambrosio, L. (2008), "Transport equation and Cauchy problem for non-smooth vector fields," in *Calculus of variations and nonlinear partial differential equations*, pp. 1–41, Springer.

Andrieu, C. and Thoms, J. (2008), "A tutorial on adaptive MCMC," *Statistics and Computing*, 18, 343–373.

Barp, A., Briol, F.-X., Kennedy, A. D., and Girolami, M. (2017), "Geometry and Dynamics for Markov Chain Monte Carlo," *arXiv:1705.02891*.

Basu, S. and Ebrahimi, N. (2001), "Bayesian Capture-Recapture Methods for Error Detection and Estimation of Population Size: Heterogeneity and Dependence," *Biometrika*, 88, 269–279.

Bennett, C. H. (1975), "Mass tensor molecular dynamics," *Journal of Computational Physics*, 19, 267–279.

Berg, B. A. and Neuhaus, T. (1992), "Multicanonical ensemble: a new approach to simulate first-order phase transitions," *Physical Review Letters*, 68, 9.

Berger, J. O. (2013), *Statistical decision theory and Bayesian analysis*, Springer Science & Business Media.

Berger, J. O., Bernardo, J. M., and Sun, D. (2012), "Objective Priors for Discrete Parameter Spaces," *Journal of the American Statistical Association*, 107, 636–648.

Bernton, E., Yang, S., Chen, Y., Shephard, N., and Liu, J. S. (2015), "Locally weighted Markov chain Monte Carlo," *arXiv:1506.08852*.

Beskos, A., Pinski, F. J., Sanz-Serna, J. M., and Stuart, A. M. (2011), "Hybrid Monte Carlo on hilbert spaces," *Stochastic Processes and their Applications*, 121, 2201–2230.

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

Betancourt, M. (2015), "The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 533–540.

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

Betancourt, M., Byrne, S., and Girolami, M. (2014), "Optimizing the integrator step size for Hamiltonian Monte Carlo," *arXiv:1411.6669*.

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

Bierkens, J., Fearnhead, P., and Roberts, G. (2016), "The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data," *arXiv:1607.03188*.

Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A. B., Fearnhead, P., Roberts, G., and Vollmer, S. J. (2017), "Piecewise Deterministic Markov Processes for Scalable Monte Carlo on Restricted Domains," *arXiv:1701.04244*.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016), "A general framework for updating belief distributions," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78, 1103–1130.

Blanes, S. and Iserles, A. (2012), "Explicit adaptive symplectic integrators for solving Hamiltonian systems," *Celestial Mechanics and Dynamical Astronomy*, 114, 297–317.

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

Bou-Rabee, N. and Sanz-Serna, J. M. (2015), "Randomized Hamiltonian Monte Carlo," *arXiv:1511.09382*.

Brogliato, B. (2016), *Nonsmooth Mechanics. Models, Dynamics and Control*, Springer.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (eds.) (2011), *Handbook of Markov Chain Monte Carlo*, CRC Press.

Byrne, S. and Girolami, M. (2013), "Geodesic Monte Carlo on embedded manifolds," *Scandinavian Journal of Statistics*, 40, 825–845.

Calderhead, B. (2014), "A general construction for parallelizing Metropolis-Hastings algorithms," *Proceedings of the National Academy of Sciences*, 111, 17408–17413.

Calvo, M., López-Marcos, M., and Sanz-Serna, J. M. (1998), "Variable step implementation of geometric integrators," *Applied Numerical Mathematics*, 28, 1–16.

Carpenter, B., Hoffman, M. D., Brubaker, M., Lee, D., Li, P., and Betancourt, M. (2015), "The Stan Math Library: Reverse-Mode Automatic Differentiation in C++," *arXiv:1509.07164*.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The horseshoe estimator for sparse signals," *Biometrika*, 97, 465–480.

Chen, T., Fox, E., and Guestrin, C. (2014), "Stochastic gradient Hamiltonian Monte Carlo," in *Proceedings of the 31st International Conference on Machine Learning*, pp. 1683–1691.

Chib, S. (1998), "Estimation and comparison of multiple change-point models," *Journal of Econometrics*, 86, 221–241.

Chopin, N. and Ridgway, J. (2017), "Leave Pima Indians alone: binary regression as a benchmark for Bayesian computation," *Statistical Science*, 32, 64–87.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014), "Bayesian sampling using stochastic gradient thermostats," in *Advances in Neural Information Processing Systems*, pp. 3203–3211.

Dinh, V., Bilge, A., Zhang, C., and Matsen, F. A. (2017), "Probabilistic Path Hamiltonian Monte Carlo," *arXiv:1702.07814*.

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

Earl, D. J. and Deem, M. W. (2005), "Parallel tempering: Theory, applications, and new perspectives," *Physical Chemistry Chemical Physics*, 7, 3910–3916.

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

Fearnhead, P., Bierkens, J., Pollock, M., and Roberts, G. O. (2016), "Piecewise Deterministic Markov Processes for Continuous-Time Monte Carlo," *arXiv:1611.07873*.

Federer, H. (1969), *Geometric measure theory*, Springer-Verlag, Berlin.

Fetecau, R. C. (2003), "Variational methods for nonsmooth mechanics," Ph.D. thesis, California Institute of Technology.

Flegal, J. M. and Jones, G. L. (2010), "Batch means and spectral variance estimators in Markov chain Monte Carlo," *The Annals of Statistics*, 38, 1034–1070.

Frenkel, D. (2004), "Speed-up of Monte Carlo simulations by sampling of rejected states," *Proceedings of the National Academy of Sciences*, 101, 17571–17575.

Fryzlewicz, P. and Subba Rao, S. (2014), "Multiple-change-point detection for auto-regressive conditional heteroscedastic processes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 903–924.

Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models," *Bayesian Analysis*, 1, 515–534.

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996), "Efficient Metropolis jumping rules," *Bayesian Statistics*, 5, 599–607.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian Data Analysis*, CRC Press.

Gelman, A., Lee, D., and Guo, J. (2015), "Stan: a probabilistic programming language for Bayesian inference and optimization," *Journal of Educational and Behavior Science*, 40, 530–543.

Geyer, C. (2011), "Introduction to Markov chain Monte Carlo," *Handbook of Markov Chain Monte Carlo*, pp. 3–48.

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

Geyer, C. J. and Thompson, E. A. (1995), "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90, 909–920.

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

Griewank, A. and Walther, A. (2008), *Evaluating derivatives: principles and techniques of algorithmic differentiation*, Society for Industrial and Applied Mathematics.

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

Haario, H., Saksman, E., and Tamminen, J. (2005), "Componentwise adaptation for high dimensional MCMC," *Computational Statistics*, 20, 265–273.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006), "DRAM: efficient adaptive MCMC," *Statistics and Computing*, 16, 339–354.

Hairer, E., Lubich, C., and Wanner, G. (2006), *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer-Verlag.

Hoffman, M. D. and Gelman, A. (2014), "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, 15, 1593–1623.

Huang, W. and Leimkuhler, B. (1997), "The Adaptive Verlet Method," *SIAM Journal on Scientific Computing*, 18, 239–18.

Jolly, G. M. (1965), "Explicit estimates from capture-recapture data with both death and immigration-stochastic model," *Biometrika*, 52, 225–247.

Kallenberg, O. (2002), *Foundations of Modern Probability*, Probability and Its Applications, Springer New York.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006), "Equi-energy sampler with applications in statistical inference and statistical mechanics," *The annals of Statistics*, pp. 1581–1619.

Kruschke, J. (2014), *Doing Bayesian Data Analysis, Second Edition: A Tutorial with R, JAGS, and Stan*, Academic Press.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2015), "Automatic variational inference in Stan," in *Advances in Neural Information Processing Systems*, pp. 568–576.

Lan, S., Streets, J., and Shahbaba, B. (2014), "Wormhole Hamiltonian Monte Carlo," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

Lan, S., Stathopoulos, V., Shahbaba, B., and Girolami, M. (2015), "Markov Chain Monte Carlo from Lagrangian Dynamics," *Journal of Computational and Graphical Statistics*, 24, 357–378.

Leimkuhler, B. and Reich, S. (2005), *Simulating Hamiltonian Dynamics*, Cambridge University Press.

Leimkuhler, B. and Reich, S. (2009), "A Metropolis adjusted Nosé-Hoover thermostat," *ESAIM: Mathematical Modelling and Numerical Analysis*, 43, 743–755.

Leonhardt, U. and Philbin, T. (2010), *Geometry and light: the science of invisibility*, Dover Books on Physics, Dover, Mineola, NY.

Liang, F., Liu, C., and Carroll, R. (2011), *Advanced Markov chain Monte Carlo methods: learning from past samples*, vol. 714, John Wiley & Sons.

Livingstone, S., Betancourt, M., Byrne, S., and Girolami, M. (2016), "On the Geometric Ergodicity of Hamiltonian Monte Carlo," *arXiv:1601.08057*.

Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., and Vollmer, S. J. (2016), "Relativistic Monte Carlo," *arXiv:1609.04388*.

Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evolution, critique and future directions," *Statistics in medicine*, 28, 3049–3067.

Marinari, E. and Parisi, G. (1992), "Simulated tempering: a new Monte Carlo Scheme," *Europhysics Letters*, 19, 451.

McLachlan, R. I. and Quispel, G. R. W. (2002), "Splitting methods," *Acta Numerica*, 11, 341–434.

Monnahan, C. C., Thorson, J. T., and Branch, T. A. (2016), "Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo," *Methods in Ecology and Evolution*, pp. 339–348.

Nakajima, J. and West, M. (2013), "Bayesian analysis of latent threshold dynamic models," *Journal of Business & Economic Statistics*, 31, 151–164.

Nash, J. (1954), "C1 Isometric Imbeddings," *Annals of Mathematics*, 60, 383–396.

Neal, R. M. (1994), "An Improved Acceptance Procedure for the Hybrid Monte Carlo Algorithm," *Journal of Computational Physics*, 111, 194–203.

Neal, R. M. (1996a), *Bayesian Learning for Neural Networks*, Springer-Verlag.

Neal, R. M. (1996b), "Sampling from multimodal distributions using tempered transitions," *Statistics and Computing*, 6, 353–366.

Neal, R. M. (2001), "Annealed importance sampling," *Statistics and Computing*, 11, 125–139.

Neal, R. M. (2010), "MCMC using Hamiltonian Dynamics," in *Handbook of Markov chain Monte Carlo*, CRC Press.

Neelon, B. and Dunson, D. B. (2004), "Bayesian isotonic regression and trend analysis," *Biometrics*, 60, 398–406.

Nishimura, A. and Dunson, D. (2015), "Recycling intermediate steps to improve Hamiltonian Monte Carlo," *arXiv:1511.06925*.

Nishimura, A. and Dunson, D. (2016), "Geometrically Tempered Hamiltonian Monte Carlo," *arXiv:1604.00872*.

Okudo, M. and Suzuki, H. (2016), "Hamiltonian Monte Carlo with explicit, reversible, and volume-preserving adaptive step size control," Tech. rep., Department of Mathematical Informatics, University of Tokyo.

Pakman, A. and Paninski, L. (2013), "Auxiliary-variable Exact Hamiltonian Monte Carlo Samplers for Binary Distributions," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 2490–2498.

Pakman, A. and Paninski, L. (2014), "Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians," *Journal of Computational and Graphical Statistics*, 23, 518–542.

Parkin, D. M. and Bray, F. (2009), "Evaluation of data quality in the cancer registry: Principles and methods Part II. Completeness," *European Journal of Cancer*, 45, 756–764.

Peters, E. A. J. F. and de With, G. (2012), "Rejection-free Monte Carlo sampling for general potentials," *Physical Review E*, 85, 026703.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11.

Roberts, G. O. and Rosenthal, J. S. (2001), "Optimal scaling for various Metropolis-Hastings algorithms," *Statistical Science*, 16, 351–367.

Roberts, G. O. and Stramer, O. (2002), "Langevin diffusions and Metropolis-Hastings algorithms," *Methodology and Computing in Applied Probability*, 4, 337–357.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997), "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The Annals of Applied Probability*, 7, 110–120.

Rogers, L. C. G. and Williams, D. (2000), *Diffusions, Markov processes, and martingales. Volume 2, Itô calculus*, Cambridge University Press, Cambridge.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016), "Probabilistic programming in Python using PyMC3," *PeerJ Computer Science*.

Sanz-Serna, J. M. (1994), "An unconventional symplectic integrator of W. Kahan," *Applied Numerical Mathematics*, 16, 245–250.

Schwarz, C. J. and Arnason, A. N. (1996), "A general methodology for the analysis of capture-recapture experiments in open populations," *Biometrics*, pp. 860–873.

Schwarz, C. J. and Seber, G. A. F. (1999), "Estimating Animal Abundance: Review III," *Statistical Science*, 14, 427–456.

Seber, G. A. F. (1982), *The estimation of animal abundance*, Griffin London.

Shahbaba, B., Lan, S., Johnson, W. O., and Neal, R. M. (2014), "Split Hamiltonian Monte Carlo," *Statistics and Computing*, 24, 339–349.

Shang, X., Zhu, Z., Leimkuhler, B., and Storkey, A. J. (2015), "Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling," in *Advances in Neural Information Processing Systems*, pp. 37–45.

Sohl-Dickstein, J., Mudigonda, M., and DeWeese, M. (2014), "Hamiltonian Monte Carlo without detailed balance," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, pp. 719–726.

Stan Development Team (2016), *Stan Modeling Language Users Guide and Reference Manual, Version 2.14.0*.

Stewart, D. E. (2000), "Rigid-body dynamics with friction and impact," *SIAM Review*, 42, 3–39.

Theano Development Team (2016), "Theano: A Python framework for fast computation of mathematical expressions," *arXiv:1605.02688*.

Tjelmeland, H. (2004), "Using all Metropolis-Hastings proposals to estimate mean values," Tech. rep., Norwegian University of Science and Technology.

Wagner, A. K., Soumerai, Stephen B .and Zhang, F., and Ross-Degnan, D. (2002), "Segmented regression analysis of interrupted time series studies in medication use research," *Journal of Clinical Pharmacy and Therapeutics*, 27, 299–309.

Wang, F. and Landau, D. P. (2001), "Efficient, multiple-range random walk algorithm to calculate the density of states," *Physical Review Letters*, 86, 2050.

Wang, Z., Mohamed, S., and de Freitas, N. (2013), "Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, pp. 1462–1470.

Warren, R. and Warren, J. R. (2013), "Unauthorized Immigration to the United States: Annual Estimates and Components of Change, by State, 1990 to 2010," *International Migration Review*, 47, 296–329.

Welling, M. and Teh, Y. W. (2011), "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688.

Xifara, T., Sherlock, C., Livingstone, S., Byrne, S., and Girolami, M. (2014), "Langevin diffusions and the Metropolis-adjusted Langevin algorithm," *Statistics & Probability Letters*, 91, 14–19.

Zhang, Y., Sutton, C., Storkey, A., and Ghahramani, Z. (2012), "Continuous Relaxations for Discrete Hamiltonian Monte Carlo," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 3194–3202.

Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K., and Carin, L. (2016), "Towards Unifying Hamiltonian Monte Carlo and Slice Sampling," *arXiv:1602.07800*.

Biography

Akihiko Nishimura was born September 19th, 1987 in Tokyo, Japan. He attended Stanford University and earned a Bachelor of Science in Mathematics with distinction and with minor in Physics in 2010. He also completed a Master of Science in Statistics at Stanford University in 2011. He will be graduating with his Ph.D. from Duke University under the supervision of Prof. David Dunson.

He is a recipient of James B. Duke fellowship from Duke University and a winner of 2016 Laplace award from the Section on Bayesian Statistical Science of the American Statistical Association.

After completion of his Ph.D., he will be joining the Department of Biomathematics and Biostatistics at University of California Los Angeles as a postdoctoral research scholar under the supervision of Prof. Marc Suchard.