
Hamiltonian Monte Carlo Without Detailed Balance

Jascha Sohl-Dickstein

Stanford University, Palo Alto. Khan Academy, Mountain View

JASCHA@STANFORD.EDU

Mayur Mudigonda

Redwood Institute for Theoretical Neuroscience, University of California at Berkeley

MUDIGONDA@BERKELEY.EDU

Michael R. DeWeese

Redwood Institute for Theoretical Neuroscience, University of California at Berkeley

DEWEESE@BERKELEY.EDU

Abstract

We present a method for performing Hamiltonian Monte Carlo that largely eliminates sample rejection. In situations that would normally lead to rejection, instead a longer trajectory is computed until a new state is reached that can be accepted. This is achieved using Markov chain transitions that satisfy the fixed point equation, but do not satisfy detailed balance. The resulting algorithm significantly suppresses the random walk behavior and wasted function evaluations that are typically the consequence of update rejection. We demonstrate a greater than factor of two improvement in mixing time on three test problems. We release the source code as Python and MATLAB packages.

1. Introduction

High dimensional and otherwise computationally expensive probabilistic models are of increasing importance for such diverse tasks as modeling the folding of proteins (Schütte & Fischer, 1999), the structure of natural images (Culpepper et al., 2011), or the activity of networks of neurons (Cadieu & Koepsell, 2010).

Sampling from the described distribution is typically the bottleneck when working with these probabilistic models. Sampling is commonly required when training a probabilistic model, when evaluating the model’s performance, when performing inference, and when taking expectations (MacKay, 2003). Therefore, work that improves sampling is fundamentally important.

The most common way to guarantee that a sampling algo-

rithm converges to the correct distribution is via a concept known as detailed balance. Sampling algorithms based on detailed balance are powerful because they allow samples from any target distribution to be generated from almost any proposal distribution, using for instance Metropolis-Hastings acceptance criteria (Hastings, 1970). However, detailed balance also suffers from a critical flaw. Precisely because the forward and reverse transitions occur with equal probability, detailed balance driven samplers go backwards exactly as often as they go forwards. The state space is thus explored via a random walk over distances longer than those traversed by a single draw from the proposal distribution. A random walk only travels a distance $dN^{\frac{1}{2}}$ in N steps, where d is the characteristic step length.

The current state-of-the-art sampling algorithm for probability distributions with continuous state spaces is Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2010). By extending the state space to include auxiliary momentum variables, and then using Hamiltonian dynamics to traverse long iso-probability contours in this extended state space, HMC is able to move long distances in state space in a single update step. However, HMC still relies on detailed balance to accept or reject steps, and as a result still behaves like a random walk – just a random walk with a longer step length. Previous attempts to address this have combined multiple Markov steps that individually satisfy detailed balance into a composite step that does not (Horowitz, 1991), with limited success (Kennedy & Pendleton, 1991).

The No-U-Turn Sampler (NUTS) sampling package (Hoffman & Gelman, 2011) and the windowed acceptance method of (Neal, 1994) both consider Markov transitions within a set of discrete states generated by repeatedly simulating Hamiltonian dynamics. NUTS generates a set of candidate states around the starting state by running Hamiltonian dynamics forwards and backwards until the trajectory doubles back on itself, or a slice variable constraint is violated. It then chooses a new state at uniform from the

candidate states. In windowed acceptance, a transition is proposed between a window of states at the beginning and end of a trajectory, rather than the first state and last state. With the selected window, a single state is then chosen using Boltzmann weightings. Both NUTS and the windowed acceptance method rely on detailed balance to choose the candidate state from the discrete set.

Here we present a novel discrete representation of the HMC state space and transitions. Using this representation, we derive a method for performing HMC while abandoning detailed balance altogether, by directly satisfying the fixed point equation restricted to the discrete state space. As a result, random walk behavior in the sampling algorithm is greatly reduced, and the mixing rate of the sampler is substantially improved.

2. Sampling

We begin by briefly reviewing some key concepts related to sampling. The goal of a sampling algorithm is to draw characteristic samples $\mathbf{x} \in \mathcal{R}^N$ from a target probability distribution $p(\mathbf{x})$. Without loss of generality, we will assume that $p(\mathbf{x})$ is determined by an energy function $E(\mathbf{x})$,

$$p(\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{x})). \quad (1)$$

2.1. Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) (Neal, 1993) is commonly used to sample from probabilistic models. In MCMC a chain of samples is generated by repeatedly drawing new samples \mathbf{x}' from a conditional probability distribution $T(\mathbf{x}'|\mathbf{x})$, where \mathbf{x} is the previous sample. Since $T(\mathbf{x}'|\mathbf{x})$ is a probability density over \mathbf{x}' , $\int T(\mathbf{x}'|\mathbf{x}) d\mathbf{x}' = 1$ and $T(\mathbf{x}'|\mathbf{x}) \geq 0$.

2.2. Fixed Point Equation

An MCMC algorithm must satisfy two conditions in order to generate samples from the target distribution $p(\mathbf{x})$. The first is mixing, which requires that repeated application of $T(\mathbf{x}'|\mathbf{x})$ must eventually explore the full state space of $p(\mathbf{x})$. The second condition is that the target distribution $p(\mathbf{x})$ must be a fixed point of $T(\mathbf{x}'|\mathbf{x})$. This second condition can be expressed by the fixed point equation,

$$\int p(\mathbf{x}) T(\mathbf{x}'|\mathbf{x}) d\mathbf{x} = p(\mathbf{x}'), \quad (2)$$

which requires that when $T(\mathbf{x}'|\mathbf{x})$ acts on $p(\mathbf{x})$, the resulting distribution is unchanged.

2.3. Detailed Balance

Detailed balance is the most common way of guaranteeing that the Markov transition distribution $T(\mathbf{x}'|\mathbf{x})$ satisfies the

fixed point equation (Equation 2). Detailed balance guarantees that if samples are drawn from the equilibrium distribution $p(\mathbf{x})$, then for every pair of states \mathbf{x} and \mathbf{x}' the probability of transitioning from state \mathbf{x} to state \mathbf{x}' is identical to that of transitioning from state \mathbf{x}' to \mathbf{x} ,

$$p(\mathbf{x}) T(\mathbf{x}'|\mathbf{x}) = p(\mathbf{x}') T(\mathbf{x}|\mathbf{x}'). \quad (3)$$

By substitution for $T(\mathbf{x}'|\mathbf{x})$ in the left side of Equation 2, it can be seen that if Equation 3 is satisfied, then the fixed point equation is also satisfied.

An appealing aspect of detailed balance is that a transition distribution satisfying it can be easily constructed from nearly any proposal distribution, using Metropolis-Hastings acceptance/rejection rules (Hastings, 1970). A primary drawback of detailed balance, and of Metropolis-Hastings, is that the resulting Markov chains always engage in random walk behavior, since by definition detailed balance depends on forward and reverse transitions happening with equal probability.

The primary advance in this paper is demonstrating how HMC sampling can be performed without resorting to detailed balance.

3. Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) can traverse long distances in state space with single Markov transitions. It does this by extending the state space to include auxiliary momentum variables, and then simulating Hamiltonian dynamics to move long distances along iso-probability contours in the expanded state space.

3.1. Extended state space

The state space is extended by the addition of momentum variables $\mathbf{v} \in \mathcal{R}^N$, with identity-covariance Gaussian distribution,

$$p(\mathbf{v}) = (2\pi)^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\mathbf{v}^T\mathbf{v}\right). \quad (4)$$

We refer to the combined state space of \mathbf{x} and \mathbf{v} as ζ , such that $\zeta = \{\mathbf{x}, \mathbf{v}\}$. The corresponding joint distribution is

$$p(\zeta) = p(\mathbf{x}, \mathbf{v}) = p(\mathbf{x}) p(\mathbf{v}) = \frac{(2\pi)^{-\frac{N}{2}}}{Z} \exp(-H(\zeta)), \quad (5)$$

$$H(\zeta) = H(\mathbf{x}, \mathbf{v}) = E(\mathbf{x}) + \frac{1}{2}\mathbf{v}^T\mathbf{v}. \quad (6)$$

$H(\zeta)$ has the same form as total energy in a physical system, where $E(\mathbf{x})$ is the potential energy for position \mathbf{x} and $\frac{1}{2}\mathbf{v}^T\mathbf{v}$ is the kinetic energy for momentum \mathbf{v} (mass is set to one).

In HMC samples from $p(\mathbf{x})$ are generated by drawing samples from the joint distribution $p(\mathbf{x}, \mathbf{v})$, and using only the \mathbf{x} variables as samples from the desired distribution.

3.2. Hamiltonian dynamics

Hamiltonian dynamics govern how physical systems evolve with time. It might be useful to imagine the trajectory of a skateboarder rolling in an empty swimming pool. As she rolls downwards she exchanges potential energy for kinetic energy, and the magnitude of her velocity increases. As she rolls up again she exchanges kinetic energy back for potential energy. In this fashion she is able to traverse long distances across the swimming pool, while at the same time maintaining constant total energy over her entire trajectory.

In HMC, we treat $H(\zeta)$ as the total energy of a physical system, with spatial coordinate \mathbf{x} , velocity \mathbf{v} , potential energy $E(\mathbf{x})$, and kinetic energy $\frac{1}{2}\mathbf{v}^T\mathbf{v}$. In an identical fashion to the case of the skateboarder in the swimming pool, running Hamiltonian dynamics on this system traverses long distances in \mathbf{x} while maintaining constant total energy $H(\zeta)$. By Equation 5, moving along a trajectory with constant energy is identical to moving along a trajectory with constant probability.

Hamiltonian dynamics can be run exactly in reverse by reversing the velocity vector. They also preserve volume in ζ . As we will see, all these properties together mean that Hamiltonian dynamics can be used to propose update steps that move long distances in state space while retaining high acceptance probability.

3.3. Operators

The Markov transitions from which HMC is constructed can be understood in terms of several operators acting on ζ . These operators are illustrated in Figure 1a. This representation of the actions performed in HMC, and the corresponding state space, is unique to this paper and diverges from the typical presentation of HMC.

3.3.1. MOMENTUM FLIP

The momentum flip operator \mathbf{F} reverses the direction of the momentum. It is its own inverse, leaves the total energy unchanged, and preserves volume in state space:

$$\mathbf{F}\zeta = \mathbf{F}\{\mathbf{x}, \mathbf{v}\} = \{\mathbf{x}, -\mathbf{v}\}, \quad (7)$$

$$\mathbf{F}^{-1}\zeta = \mathbf{F}\zeta, \quad (8)$$

$$H(\mathbf{F}\zeta) = H(\zeta), \quad (9)$$

$$\left| \det \left(\frac{\partial \mathbf{F}\zeta}{\partial \zeta^T} \right) \right| = 1. \quad (10)$$

The momentum flip operator \mathbf{F} causes movement between the left and right sides of the state ladder in Figure 1b.

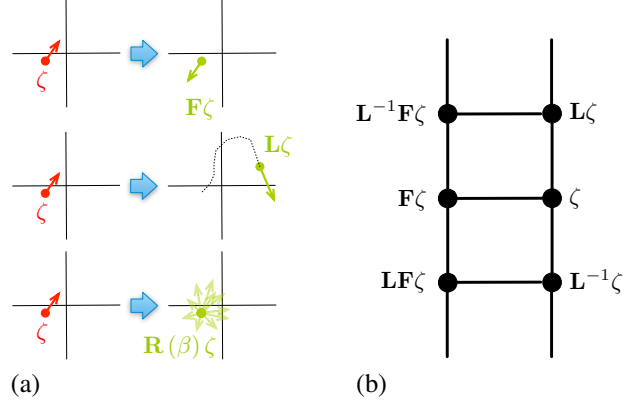


Figure 1. (a) The action of operators involved in Hamiltonian Monte Carlo (HMC). The base of each red or green arrow represents the position \mathbf{x} , and the length and direction of each of these arrows represents the momentum \mathbf{v} . The flip operator \mathbf{F} reverses the momentum. The leapfrog operator \mathbf{L} approximately integrates Hamiltonian dynamics. The trajectory taken by \mathbf{L} is indicated by the dotted line. The randomization operator $\mathbf{R}(\beta)$ corrupts the momentum with an amount of noise that depends on β . (b) The ladder of discrete states that are accessible by applying \mathbf{F} and \mathbf{L} starting at state ζ . Horizontal movement on the ladder occurs by flipping the momentum, whereas vertical movement occurs by integrating Hamiltonian dynamics.

3.3.2. LEAPFROG INTEGRATOR

Leapfrog, or Störmer-Verlet, integration provides a discrete time approximation to Hamiltonian dynamics (Hairer et al., 2003). The operator $\mathbf{L}(\epsilon, M)$ performs leapfrog integration for M leapfrog steps with step length ϵ . For conciseness, $\mathbf{L}(\epsilon, M)$ will be written only as \mathbf{L} ,

$$\mathbf{L}\zeta = \begin{cases} \text{The state resulting from } M \text{ steps of} \\ \text{leapfrog integration of Hamiltonian} \\ \text{dynamics with step length } \epsilon. \end{cases} \quad (11)$$

Like exact Hamiltonian dynamics, leapfrog dynamics are exactly reversible by reversing the velocity vector, and they also exactly preserve volume in state space. \mathbf{L} can be inverted by reversing the sign of the momentum, tracing out the reverse trajectory, and then reversing the sign of the momentum again so that it points in the original direction;

$$\mathbf{L}^{-1}\zeta = \mathbf{F}\mathbf{L}\mathbf{F}\zeta, \quad (12)$$

$$\left| \det \left(\frac{\partial \mathbf{L}\zeta}{\partial \zeta^T} \right) \right| = 1. \quad (13)$$

Unlike for exact dynamics, the total energy $H(\zeta)$ is only approximately conserved by leapfrog integration, and the energy accumulates errors due to discretization. This discretization error in the energy is the source of all rejections of proposed updates in HMC.

The leapfrog operator \mathbf{L} causes movement up the right side

of the state ladder in Figure 1b, and down the left side of the ladder.

3.3.3. MOMENTUM RANDOMIZATION

The momentum randomization operator $\mathbf{R}(\beta)$ mixes an amount of Gaussian noise determined by $\beta \in [0, 1]$ into the velocity vector,

$$\mathbf{R}(\beta)\zeta = \mathbf{R}(\beta)\{\mathbf{x}, \mathbf{v}\} = \{\mathbf{x}, \mathbf{v}'\}, \quad (14)$$

$$\mathbf{v}' = \mathbf{v}\sqrt{1-\beta} + \mathbf{n}\sqrt{\beta}, \quad (15)$$

$$\mathbf{n} \sim N(\mathbf{0}, \mathbf{I}). \quad (16)$$

Unlike the previous two operators, the momentum randomization operator is not deterministic. $\mathbf{R}(\beta)$ is however a valid Markov transition operator for $p(\zeta)$ on its own, in that it satisfies both Equation 2 and Equation 3.

The momentum randomization operator $\mathbf{R}(\beta)$ causes movement off of the current state ladder and on to a new state ladder.

3.4. Discrete State Space

As illustrated in Figure 1b, the operators \mathbf{L} and \mathbf{F} generate a discrete state space ladder, with transitions only occurring between ζ and three other states. Note that every state on the ladder can be represented many different ways, depending on the series of operators used to reach it. For instance, the state in the upper left of the figure pane can be written $\mathbf{L}^{-1}\mathbf{F}\zeta = \mathbf{F}\mathbf{L}\zeta = \mathbf{L}\mathbf{F}\mathbf{L}\zeta = \dots$.

Standard HMC can be viewed in terms of transitions on this ladder. Additionally, we will see that this discrete state space view allows Equation 2 to be solved directly by replacing the integral over all states with a short sum.

3.5. Standard HMC

HMC as typically implemented consists of the following steps. Here, $\zeta^{(t,s)}$ represents the state at sampling step t , and sampling substep s . Each numbered item below corresponds to a valid Markov transition for $p(\zeta)$, satisfying detailed balance. A full sampling step consists of the composition of all three Markov transitions.

1. (a) Generate a proposed update,

$$\zeta' = \mathbf{F}\mathbf{L}\zeta^{(t,0)}. \quad (17)$$

On the state ladder in Figure 1b, this corresponds to moving up one rung (\mathbf{L}), and then moving from the right to the left side (\mathbf{F}).

- (b) Accept or reject the proposed update using

Metropolis-Hastings rules,

$$\pi_{accept} = \min\left(1, \frac{p(\zeta')}{p(\zeta)}\right), \quad (18)$$

$$\zeta^{(t,1)} = \begin{cases} \zeta' & \text{with probability } \pi_{accept} \\ \zeta^{(t,0)} & \text{with probability } 1 - \pi_{accept} \end{cases} \quad (19)$$

Note that since the transition $\mathbf{F}\mathbf{L}$ is its own inverse, the forward and reverse proposal distribution probabilities cancel in the Metropolis-Hastings rule in Equation 18.

On rejection, the computations performed in Equation 17 are discarded. In our new technique, this will no longer be true.

2. Flip the momentum,

$$\zeta^{(t,2)} = \mathbf{F}\zeta^{(t,1)}. \quad (20)$$

If the proposed update from Step 1 was accepted, then this moves $\zeta^{(t,1)}$ from the left back to the right side of the state ladder in Figure 1b, and prevents the trajectory from doubling back on itself. If the update was rejected however, and $\zeta^{(t,1)}$ is already on the right side of the ladder, then this causes it to move to the left side of the ladder, and the trajectory to double back on itself.

Doubling back on an already computed trajectory is wasteful in HMC, both because it involves recomputing nearly redundant trajectories, and because the distance traveled before the sampler doubles back is the characteristic length scale beyond which HMC explores the state space by diffusion.

3. Corrupt the momentum with noise,

$$\zeta^{(t+1,0)} = \mathbf{R}(\beta)\zeta^{(t,2)}. \quad (21)$$

It is common to set $\beta = 1$, in which case the momentum is fully randomized every sampling step. In our experiments (Section 5) however, we found that smaller values of β produced large improvements in mixing time. This is therefore a hyperparameter that is probably worth adjusting¹.

4. Look Ahead HMC

Here we introduce an HMC algorithm that relies on Markov transitions that do not obey detailed balance, but

¹One method for choosing β (Culpepper et al., 2011) which we have found to be effective is to set it such that it randomizes a fixed fraction α of the momentum per unit simulation time,

$$\beta = \alpha \frac{1}{\tau M}. \quad (22)$$

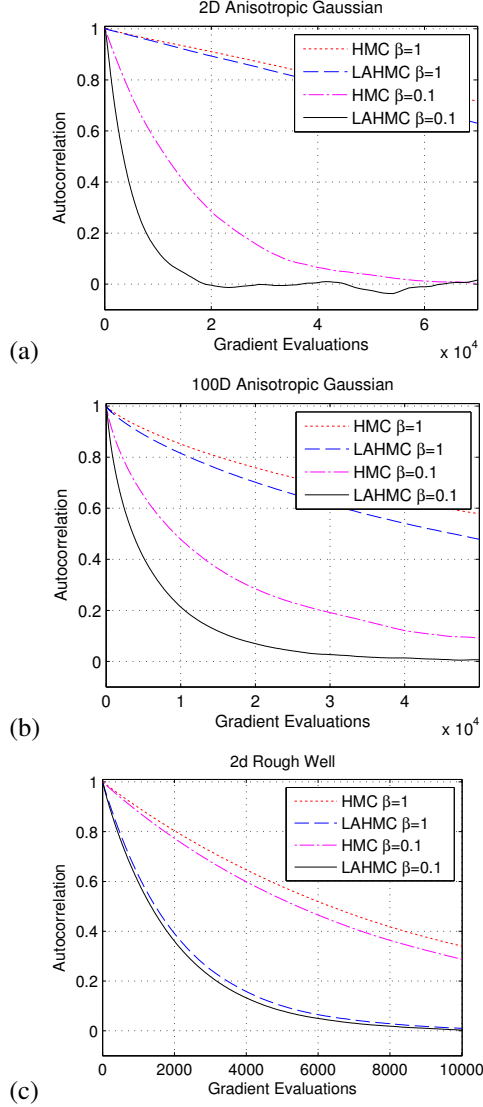


Figure 2. Autocorrelation vs. number of function evaluations for standard HMC (no momentum randomization, $\beta = 1$), LAHMC with $\beta = 1$, persistent HMC ($\beta = 0.1$), and persistent LAHMC ($\beta = 0.1$) for (a) a two dimensional ill-conditioned Gaussian, (b) a one hundred dimensional ill-conditioned Gaussian, and (c) a two dimensional well conditioned energy function with a “rough” surface. In all cases the LAHMC sampler demonstrates faster mixing.

still satisfy the fixed point equation. This algorithm eliminates much of the momentum flipping that occurs on rejection in HMC, and as a result greatly reduces random walk behavior. It also prevents the trajectory computations that would typically be discarded on proposal rejection from being wasted. We call our algorithm Look Ahead Hamiltonian Monte Carlo (LAHMC).

4.1. Intuition

In LAHMC, in situations that would correspond to a rejection in Step 1 of Section 3.5, we will instead attempt to travel even farther by applying the leapfrog operator \mathbf{L} additional times. This section provides intuition for how this update rule was discovered, and how it can be seen to connect to standard HMC. A more mathematically precise description will follow in the next several sections.

LAHMC can be understood in terms of a series of modifications of standard HMC. The net effect of Steps 1 and 2 in Section 3.5 is to transition from state ζ into either state $\mathbf{L}\zeta$ or state $\mathbf{F}\zeta$, depending on whether the update in Section 3.5 Step 1 was accepted or rejected.

We wish to minimize the transitions into state $\mathbf{F}\zeta$. In LAHMC we do this by replacing as many transitions from ζ to $\mathbf{F}\zeta$ as possible with transitions that instead go from ζ to $\mathbf{L}^2\zeta$. This would seem to change the number of transitions into both state $\mathbf{F}\zeta$ and state $\mathbf{L}^2\zeta$, violating the fixed point equation. However, the changes in incoming transitions from ζ are exactly counteracted because the state $\mathbf{F}\mathbf{L}^2\zeta$ is similarly modified, so that it makes fewer transitions into the state $\mathbf{L}^2\zeta = \mathbf{F}(\mathbf{F}\mathbf{L}^2\zeta)$, and more transitions into the state $\mathbf{F}\zeta = \mathbf{L}^2(\mathbf{F}\mathbf{L}^2\zeta)$.

For some states, after this modification there will still be transitions between the states ζ and $\mathbf{F}\zeta$. In order to further minimize these transitions, the process in the preceding paragraph is repeated for these remaining transitions and the state $\mathbf{L}^3\zeta$. This process is then repeated again for states $\mathbf{L}^4\zeta$, $\mathbf{L}^5\zeta$, etc, up to some maximum number of leapfrog applications K .

4.2. Algorithm

LAHMC consists of the following two steps,

1. Transition to a new state by applying the leapfrog operator \mathbf{L} between 1 and $K \in \mathcal{Z}^+$ times, or by applying the momentum flip operator \mathbf{F} ,

$$\zeta^{(t,1)} = \begin{cases} \mathbf{L}\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^1}(\zeta^{(t,0)}) \\ \mathbf{L}^2\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^2}(\zeta^{(t,0)}) \\ \dots & \dots \\ \mathbf{L}^K\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^K}(\zeta^{(t,0)}) \\ \mathbf{F}\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{F}}(\zeta^{(t,0)}) \end{cases} \quad (23)$$

Hamiltonian Monte Carlo Without Detailed Balance

Distribution	Sampler	$\mathbf{F}\zeta$	$\mathbf{L}\zeta$	$\mathbf{L}^2\zeta$	$\mathbf{L}^3\zeta$	$\mathbf{L}^4\zeta$
2d Gaussian	HMC $\beta = 1$	0.079	0.921	0	0	0
2d Gaussian	LAHMC $\beta = 1$	0.000	0.921	0.035	0.044	0.000
2d Gaussian	HMC $\beta = 0.1$	0.080	0.920	0	0	0
2d Gaussian	LAHMC $\beta = 0.1$	0.000	0.921	0.035	0.044	0.000
100d Gaussian	HMC $\beta = 1$	0.147	0.853	0	0	0
100d Gaussian	LAHMC $\beta = 1$	0.047	0.852	0.059	0.035	0.006
100d Gaussian	HMC $\beta = 0.1$	0.147	0.853	0	0	0
100d Gaussian	LAHMC $\beta = 0.1$	0.047	0.852	0.059	0.035	0.006
2d Rough Well	HMC $\beta = 1$	0.446	0.554	0	0	0
2d Rough Well	LAHMC $\beta = 1$	0.292	0.554	0.099	0.036	0.019
2d Rough Well	HMC $\beta = 0.1$	0.446	0.554	0	0	0
2d Rough Well	LAHMC $\beta = 0.1$	0.292	0.554	0.100	0.036	0.019

Table 1. A table showing the fraction of transitions which occurred to each target state for the conditions plotted in Figure 2. Note that LAHMC has far fewer momentum flips than standard HMC.

Note that there is no longer a Metropolis-Hastings accept/reject step. The state update in Equation 23 is a valid Markov transition for $p(\zeta)$ on its own.

- Corrupt the momentum with noise in an identical fashion as in Equation 21,

$$\zeta^{(t+1,0)} = \mathbf{R}(\beta) \zeta^{(t,1)}. \quad (24)$$

4.3. Transition Probabilities

We choose the probabilities $\pi_{\mathbf{L}^a}(\zeta)$ for the leapfrog transitions from state ζ to state $\mathbf{L}^a\zeta$ to be

$$\pi_{\mathbf{L}^a}(\zeta) = \min \left[1 - \sum_{b < a} \pi_{\mathbf{L}^b}(\zeta), \quad (25) \right. \\ \left. \frac{p(\mathbf{F}\mathbf{L}^a\zeta)}{p(\zeta)} \left(1 - \sum_{b < a} \pi_{\mathbf{L}^b}(\mathbf{F}\mathbf{L}^a\zeta) \right) \right].$$

Equation 25 greedily sets the transition probability $\pi_{\mathbf{L}^a}(\zeta)$ as large as possible, subject to the restrictions that the total transition probability out of state ζ not exceed 1, and that the transition rate in the forward direction ($\zeta \rightarrow \mathbf{L}^a\zeta$) not exceed the transition rate in the reverse direction ($\mathbf{F}\mathbf{L}^a\zeta \rightarrow \mathbf{F}\zeta$). Some algebra shows that under these transition probabilities

$$p(\zeta) \pi_{\mathbf{L}^a}(\zeta) = p(\mathbf{F}\mathbf{L}^a\zeta) \pi_{\mathbf{L}^a}(\mathbf{F}\mathbf{L}^a\zeta). \quad (26)$$

Any remaining unassigned probability is assigned to the momentum flip transition,

$$\pi_{\mathbf{F}}(\zeta) = 1 - \sum_a \pi_{\mathbf{L}^a}(\zeta). \quad (27)$$

Note that transitions will be performed in a greedy fashion. It is only necessary to compute the state $\mathbf{L}^a\zeta$ and the transition probability $\pi_{\mathbf{L}^a}(\zeta)$ if none of the transitions to states $\mathbf{L}^b\zeta$, for $b < a$, have been taken.

4.4. Fixed Point Equation

We can substitute the transition rates from Section 4.3 into the left side of Equation 2, and verify that they satisfy the fixed point equation. Note that the integral over all states is transformed into a sum over all source states from which transitions into state ζ might be initiated.

$$\int d\zeta' p(\zeta') T(\zeta|\zeta') \\ = \int d\zeta' p(\zeta') \left(\sum_a \pi_{\mathbf{L}^a}(\zeta') \delta(\zeta - \mathbf{L}^a\zeta') \quad (28) \right. \\ \left. + \pi_{\mathbf{F}}(\zeta') \delta(\zeta - \mathbf{F}\zeta') \right), \\ = \sum_a p(\mathbf{L}^{-a}\zeta) \pi_{\mathbf{L}^a}(\mathbf{L}^{-a}\zeta) + p(\mathbf{F}^{-1}\zeta) \pi_{\mathbf{F}}(\mathbf{F}^{-1}\zeta), \quad (29)$$

$$= \sum_a p(\mathbf{F}\mathbf{L}^a\mathbf{F}\zeta) \pi_{\mathbf{L}^a}(\mathbf{F}\mathbf{L}^a\mathbf{F}\zeta) + p(\mathbf{F}\zeta) \pi_{\mathbf{F}}(\mathbf{F}\zeta), \quad (30)$$

$$= \sum_a p(\mathbf{F}\zeta) \pi_{\mathbf{L}^a}(\mathbf{F}\zeta) + p(\mathbf{F}\zeta) \pi_{\mathbf{F}}(\mathbf{F}\zeta), \quad (31)$$

$$= p(\mathbf{F}\zeta) \left[\sum_a \pi_{\mathbf{L}^a}(\mathbf{F}\zeta) + \pi_{\mathbf{F}}(\mathbf{F}\zeta) \right], \quad (32)$$

$$= p(\zeta). \quad (33)$$

5. Experimental Results

As illustrated in Figure 2, we compare the mixing time for our technique and standard HMC on three distributions. HMC and LAHMC both had step length and number of leapfrog steps set to $\epsilon = 1$, and $M = 10$. Values of β were set to 1 or 0.1 as stated in the legend. For LAHMC the maximum number of leapfrog applications was set to $K = 4$. In all cases, LAHMC outperformed standard HMC for the

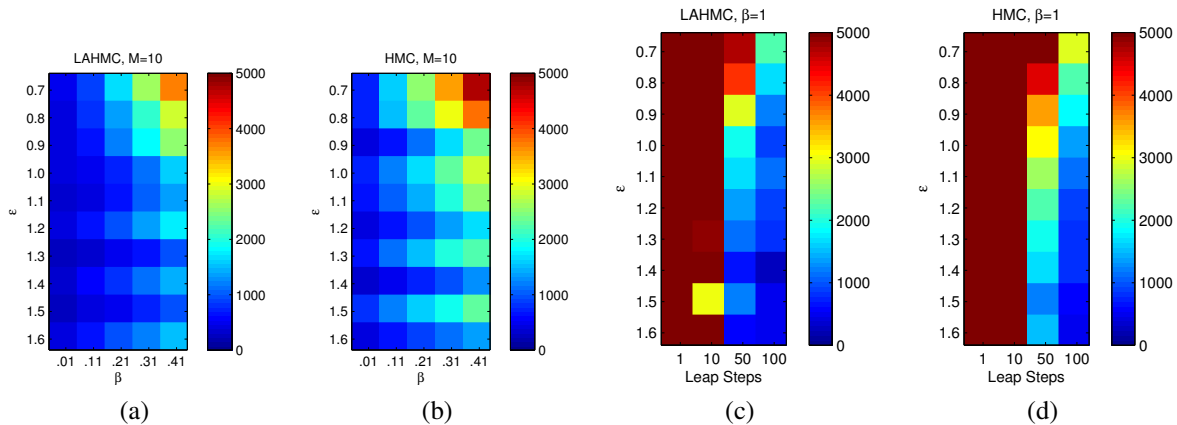


Figure 3. Images illustrating mixing time as a function of HMC hyperparameters for a two dimensional ill-conditioned Gaussian distribution. Pixel intensity indicates the number of gradient evaluations required to reach an autocorrelation of 0.5. LAHMC always outperforms HMC for the same hyperparameter settings. (a) LAHMC as a function of ϵ and β , for fixed $M = 10$, (b) HMC as a function of ϵ and β , for fixed $M = 10$, (c) LAHMC as a function of ϵ and M , for fixed $\beta = 1$, (d) HMC as a function of ϵ and M , for fixed $\beta = 1$.

same setting of hyperparameters, often by more than a factor of 2.

The first two target distributions are 2 and 100 dimensional ill-conditioned Gaussian distributions. In both Gaussians, the eigenvalues of the covariance matrix are log-linearly distributed between 1 and 10^6 .

The final target distribution was chosen to demonstrate that LAHMC is useful even for well conditioned distributions. The energy function used was the sum of an isotropic quadratic and sinusoids in each of two dimensions,

$$E(\mathbf{x}) = \frac{1}{2\sigma_1^2} (x_1^2 + x_2^2) + \cos\left(\frac{\pi x_1}{\sigma_2}\right) + \cos\left(\frac{\pi x_2}{\sigma_2}\right), \quad (34)$$

where $\sigma_1 = 100$ and $\sigma_2 = 2$. Although this distribution is well conditioned the sinusoids cause it to have a “rough” surface, such that traversing the quadratic well while maintaining a reasonable discretization error requires many leapfrog steps.

The fraction of the sampling steps resulting in each possible update for the samplers and energy functions in Figure 2 is illustrated in Table 1. The majority of momentum flips in standard HMC were eliminated by LAHMC. Note that the acceptance rate for HMC with these hyperparameter values is reasonably close to its optimal value of 65% (Neal, 2010).

Figure 3 shows several grid searches over hyperparameters for a two dimensional ill-conditioned Gaussian, and demonstrates that our technique outperforms standard HMC for all explored hyperparameter settings. Due to computational constraints, the eigenvalues of the covari-

ance of the Gaussian are 1 and 10^5 in Figure 3, rather than 1 and 10^6 as in Figure 2a.

MATLAB and Python implementations of LAHMC are available at <http://github.com/Sohl-Dickstein/LAHMC>. Figure 2 and Table 1 can be reproduced by running `generate_figure_2.m` or `generate_figure_2.py`.

6. Future Directions

There are many powerful variations on standard HMC that are complementary to and could be combined naturally with the present work. These include Riemann manifold HMC (Girolami et al., 2011), quasi-Newton HMC (Zhang & Sutton, 2011), Hilbert space HMC (Beskos & Pinski, 2011), shadow Hamiltonian methods (Izaguirre & Hampton, 2004), parameter adaptation techniques (Wang et al., 2013), Hamiltonian annealed importance sampling (Sohl-Dickstein & Culpepper, 2012), split HMC (Shahbaba et al., 2011), and tempered trajectories (Neal, 2010).

It should be possible to further reduce random walk behavior by exploring new topologies and allowed state transitions. Two other schemes have already been explored, though with only marginal benefit. In one scheme as many flips as possible are replaced by identity transitions. This is described in the note (Sohl-Dickstein, 2012). In a second scheme, a state space is constructed with two sets of auxiliary momentum variables, and an additional momentum-swap operator which switches the two momenta with each other is included in the allowed transitions. In this scenario, in situations that would typically lead to momentum flipping, with high probability the two sets of momenta can

instead be exchanged with each other. This leads to momentum randomization on rejection, rather than momentum reversal. Unfortunately, though this slightly improves mixing time, it still amounts to a random walk on a similar length scale. The exploration of other topologies and allowed transitions will likely prove fruitful.

Any deterministic, reversible, discrete stepped trajectory through a state space can be mapped onto the ladder structure in Figure 1. The Markov transition rules presented in this paper could therefore be applied to a wide range of problems. All that is required in addition to the mapping is an auxiliary variable indicating direction along that trajectory. In HMC, the momentum variable doubles as a direction indicator, but there could just as easily be an additional variable $d \in \{-1, 1\}$, $p(d = 1) = \frac{1}{2}$, which indicates whether transitions are occurring up or down the ladder. The efficiency of the exploration then depends only on choosing a sensible, approximately energy conserving, trajectory.

References

- Beskos, A and Pinski, FJ. Hybrid monte carlo on hilbert spaces. *Stochastic Processes and their Applications*, 2011.
- Cadieu, CF and Koepsell, K. Phase coupling estimation from multivariate phase statistics. *Neural computation*, 2010.
- Culpepper, Benjamin J, Sohl-Dickstein, Jascha, and Olshausen, Bruno A. Building a better probabilistic model of images by factorization. *International Conference on Computer Vision*, 2011.
- Duane, S, Kennedy, AD, Pendleton, BJ, and Roweth, D. Hybrid monte carlo. *Physics letters B*, 1987.
- Girolami, Mark, Calderhead, Ben, and Chin, Siu A. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, March 2011. ISSN 13697412. doi: 10.1111/j.1467-9868.2010.00765.x.
- Hairer, E, Lubich, C, and Wanner, G. Geometric numerical integration illustrated by the Stormer-Verlet method. *Acta Numerica*, 2003.
- Hastings, W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, January 1970.
- Hoffman, MD and Gelman, A. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Arxiv preprint arXiv:1111.4246*, pp. 1–30, 2011.
- Horowitz, AM. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 1991.
- Izaguirre, JA and Hampton, SS. Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules. *Journal of Computational Physics*, 2004.
- Kennedy, AD and Pendleton, B. Acceptances and auto-correlations in hybrid Monte Carlo. *Nuclear Physics B-Proceedings Supplements*, 1991.
- MacKay, DJC. *Information theory, inference and learning algorithms*. 2003.
- Neal, Radford M. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, January 2010.
- Neal, RM. Probabilistic inference using Markov chain Monte Carlo methods. *Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto*, 1993.
- Neal, RM. An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 1994.
- Schütte, C and Fischer, A. A direct approach to conformational dynamics based on hybrid Monte Carlo. *Journal of Computational Physics*, 1999.
- Shahbaba, B, Lan, S, Johnson, WO, and Neal, RM. Split hamiltonian monte carlo. *Statistics and Computing*, 2011.
- Sohl-Dickstein, Jascha. Hamiltonian Monte Carlo with Reduced Momentum Flips. *arXiv:1205.1939v1*, May 2012.
- Sohl-Dickstein, Jascha and Culpepper, Benjamin J. Hamiltonian Annealed Importance Sampling for partition function estimation. *arXiv:1205.1925v1*, May 2012.
- Wang, Z, Mohamed, S, and Nando, D. Adaptive Hamiltonian and Riemann Manifold Monte Carlo. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.
- Zhang, Y and Sutton, C. Quasi-Newton Markov chain Monte Carlo. 2011.