Exponential Integration for Hamiltonian Monte Carlo

Wei-Lun Chao[1] WEILUNC@USC.EDU
Justin Solomon[2] JUSTIN.SOLOMON@STANFORD.EDU
Dominik L. Michels[2] MICHELS@CS.STANFORD.EDU
Fei Sha[1] FEISHA@USC.EDU
[1]Department of Computer Science, University of Southern California, Los Angeles, CA 90089
[2]Department of Computer Science, Stanford University, 353 Serra Mall, Stanford, California 94305 USA

Abstract

We investigate numerical integration of ordinary differential equations (ODEs) for Hamiltonian Monte Carlo (HMC). High-quality integration is crucial for designing efficient and effective proposals for HMC. While the standard method is leapfrog (Störmer-Verlet) integration, we propose the use of an *exponential integrator*, which is robust to stiff ODEs with highly-oscillatory components. This oscillation is difficult to reproduce using leapfrog integration, even with carefully selected integration parameters and preconditioning. Concretely, we use a Gaussian distribution approximation to segregate stiff components of the ODE. We integrate this term *analytically* for stability and account for deviation from the approximation using variation of constants. We consider various ways to derive Gaussian approximations and conduct extensive empirical studies applying the proposed "exponential HMC" to several benchmarked learning problems. We compare to state-of-the-art methods for improving leapfrog HMC and demonstrate the advantages of our method in generating many effective samples with high acceptance rates in short running times.

1. Introduction

Markov chain Monte Carlo (MCMC) is at the core of many methods in computational statistics (Gelman et al., 2004; Robert & Casella, 2004) to sample from complex probability distributions for inference and learning.

Metropolis-Hastings MCMC obtains such samples by constructing a Markov chain with help from proposal distributions that generate random walks in sample space. A fraction of these walks aligned with the target distribution is kept, while the remainder is discarded. Sampling efficiency and quality depends critically on the proposal distribution. For this reason, designing good proposals has long been a subject of intensive research.

For distributions over continuous variables, Hamiltonian Monte Carlo (HMC) is a state-of-the-art sampler using classical mechanics to generate proposal distributions (Neal, 2011). The method starts by introducing auxiliary sampling variables. It then treats the negative log density of the variables' joint distribution as the Hamiltonian of particles moving in sample space. Final positions on the motion trajectories are used as proposals.

The equations of motion are ordinary differential equations (ODEs) requiring numerical integration. Several factors affect the choice of integrators. Time reversibility is needed for the Markov chain to converge to the target distribution, and volume preservation ensures that the sample acceptance test is tractable. The acceptance rate of HMC is determined by how well the integrator preserves the energy of the physical system, and hence high-fidelity numerical solutions are preferable. In the rare case that the ODE can be integrated exactly, no samples would be rejected. More realistically, however, when the ODE is integrated approximately, especially over a long time period to encourage exploration of the sample space, integration error can impede energy conservation, slowing convergence. Thus, the time period for integration usually is subdivided into many shorter steps.

Despite its relevance to sampling efficiency, investigation of sophisticated numerical integrators for HMC has been scarce in the current machine learning literature. The standard HMC integrator is the leapfrog (Störmer-Verlet) method. Leapfrog integration, however, is sensitive to *stiff* ODEs with highly-oscillatory dynamical components (Hairer et al., 2006). When the target density yields a stiff ODE, e.g. a multivariate Gaussian distribution with small variances in certain dimensions, the time step for leapfrog

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

is limited to the scale of the stiff components to avoid perturbing the energy of the system and consequently lowering MCMC acceptance rates. This results in limited motion in sample space, requiring many integration steps to explore the space. More generally, stiff ODEs occur in HMC whenever target distributions tightly peak around their modes, further motivating the need for advanced integrators. Although preconditioning can reduce stiffness and partially alleviate this problem, it is often insufficient, as demonstrated in our empirical studies.

In this paper, we address the challenge of simulating stiff ODEs in HMC using an explicit *exponential integrator*. Exponential integrators are known in simulation and numerics for their ability to take large time steps with enhanced stability. They decompose ODEs into two terms, a linearization solvable in closed form and a nonlinear remainder. For HMC, the linear term encodes a Gaussian component of the distribution; when the small variances in the distribution are adequately summarized by this part, exponential integration outperforms generic counterparts like leapfrog. Our exponential integrator ("expHMC") is easily-implemented and efficient, with high acceptance rates, broad exploration of sample space, and fast convergence due to the reduced restriction on the time step size. The flexibility to choose filter functions in exponential integrators further enhances the applicability of our method.

We validate the effectiveness of expHMC with extensive empirical studies on various types of distributions, including Bayesian logistic regression and independent component analysis. We also compare expHMC to alternatives, such as the conventional leapfrog method and the recently proposed Riemann manifold HMC, demonstrating desirable characteristics in scenarios where other methods suffer.

We describe basic HMC in §2 and our approach in §3. We then review related work in §4, and report several empirical studies in §5. We conclude in §7.

2. Hamiltonian Monte Carlo

Suppose we would like to sample a random variable $\boldsymbol{q} \in \mathbb{R}^d \sim p(\boldsymbol{q})$. Hamiltonian Monte Carlo (HMC) considers the joint distribution between $\boldsymbol{q}$ and an auxiliary random variable $\boldsymbol{p} \in \mathbb{R}^d \sim p(\boldsymbol{p})$ that is distributed independently of $\boldsymbol{q}$. For simplicity, $p(\boldsymbol{p})$ is often assumed to be a zero-mean Gaussian with covariance $\boldsymbol{M}$.

The joint density $p(\boldsymbol{q}, \boldsymbol{p})$ is used to define a Hamiltonian

$$\begin{aligned} H(\boldsymbol{q}, \boldsymbol{p}) &= -\log p(\boldsymbol{q}) - \log p(\boldsymbol{p}) \\ &= U(\boldsymbol{q}) + \frac{1}{2}\boldsymbol{p}^{\mathsf{T}}\boldsymbol{M}^{-1}\boldsymbol{p} + \text{const.}, \end{aligned} \tag{1}$$

where $\boldsymbol{M}$ is referred to as the (preconditioning) mass matrix.

The dynamics governed by this Hamiltonian are given by

function LEAPFROG($(\boldsymbol{q}_0, \boldsymbol{p}_0); h, L$)
  $\boldsymbol{p}_{1/2} \leftarrow \boldsymbol{p}_0 - \frac{1}{2}h\nabla_{\boldsymbol{q}}U(\boldsymbol{q}_0)$
  for $i \leftarrow 1, 2, \ldots, L$
    $\boldsymbol{q}_i \leftarrow \boldsymbol{q}_{i-1} + h\boldsymbol{M}^{-1}\boldsymbol{p}_{i-1/2}$
    if $i \neq L$ **then**
      $\boldsymbol{p}_{i+1/2} \leftarrow \boldsymbol{p}_{i-1/2} - h\nabla_{\boldsymbol{q}}U(\boldsymbol{q}_i)$
  $\boldsymbol{p}_L \leftarrow \boldsymbol{p}_{L-1/2} - \frac{1}{2}h\nabla_{\boldsymbol{q}}U(\boldsymbol{q}_L)$
  return $(\boldsymbol{q}_L, \boldsymbol{p}_L)$ as $(\boldsymbol{q}^*, \boldsymbol{p}^*)$

Figure 1. Leapfrog integrator.

the following coupled system of ODEs:

$$\begin{cases} \dot{\boldsymbol{q}} = \nabla_{\boldsymbol{p}}H = \boldsymbol{M}^{-1}\boldsymbol{p} \\ \dot{\boldsymbol{p}} = -\nabla_{\boldsymbol{q}}H = -\nabla_{\boldsymbol{q}}U(\boldsymbol{q}), \end{cases} \tag{2}$$

where dots denote derivatives in time.

The trajectories of $\boldsymbol{q}$ and $\boldsymbol{p}$ provide proposals to the Metropolis-Hastings MCMC procedure. Specifically, HMC applies the following steps: (i) starting from $k = 1$, draw $\boldsymbol{p}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$; (ii) compute the position $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ by simulating (2) for time t with initial conditions $(\boldsymbol{q}_{k-1}, \boldsymbol{p}_k)$; (iii) compute the change in Hamiltonian $\delta H = H(\boldsymbol{q}_{k-1}, \boldsymbol{p}_k) - H(\boldsymbol{q}^*, \boldsymbol{p}^*)$; (iv) output $\boldsymbol{q}_k = \boldsymbol{q}^*$ with probability $\min(e^{\delta H}, 1)$.

If $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ comes from solving (2) *exactly*, then by conservation of H, $\delta H = 0$ and the new sample is always accepted. Closed-form solutions to (2) rarely exist, however. In practice, t is broken into L discrete steps, and integration is carried out numerically in increments of $h = t/L$. This discretization creates a chance of rejected samples, as H is no longer conserved exactly. The constructed chain of $\{\boldsymbol{q}_k\}$ still converges to the target distribution, so long as the integrator is symplectic and time-reversible (Neal, 2011).

An integrator satisfying these criteria is the leapfrog (Störmer-Verlet) method in Figure 1. It is easily implemented and explicit—no linear or nonlinear solvers are needed to generate $(\boldsymbol{q}_L, \boldsymbol{p}_L)$ from $(\boldsymbol{q}_0, \boldsymbol{p}_0)$. For large h, however, it suffers from instability that can lead to inaccurate solutions. This restriction on h is problematic when the ODE (2) is stiff, requiring small h and correspondingly large L to resolve high-frequency oscillations of $(\boldsymbol{q}, \boldsymbol{p})$. We demonstrate this instability empirically in §5.

3. Exponential HMC

The idea behind exponential integration is to consider dynamics at different scales explicitly. This is achieved by decomposing the Hamiltonian into two terms. The first encodes high-frequency and numerically-unstable oscillation, while the second encodes slower dynamics that are robust to explicit integration. The insight is that the first part can be solved analytically, leaving numerical integration for the

more stable remainder. Assembling these two parts will provide a stable solution to the original dynamics.

Here, we introduce the basic idea first and then propose practical strategies for decomposition.

3.1. Exponential Integrator

We consider the following structure for U:

$$\nabla_{\boldsymbol{q}} U(\boldsymbol{q}) = \boldsymbol{\Sigma}^{-1}\boldsymbol{q} + \boldsymbol{f}(\boldsymbol{q}) \tag{3}$$

for some positive definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ and a possibly nonlinear function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$. This form can originate from a density with "slow" portion $\boldsymbol{f}(\boldsymbol{q})$ modulated by a possibly *stiff* Gaussian distribution from $\boldsymbol{\Sigma}^{-1}$, with stiffness from small variances.

The dynamical equations (2) are equivalent to

$$\ddot{\boldsymbol{q}} + \boldsymbol{M}^{-1}(\boldsymbol{\Sigma}^{-1}\boldsymbol{q} + \boldsymbol{f}(\boldsymbol{q})) = 0. \tag{4}$$

We substitute $\boldsymbol{r} = \boldsymbol{M}^{1/2}\boldsymbol{q}$ and rewrite (4) as

$$\ddot{\boldsymbol{r}} + \boldsymbol{\Omega}^2\boldsymbol{r} + \boldsymbol{F}(\boldsymbol{r}) = 0, \tag{5}$$

where $\boldsymbol{\Omega}^2 = \boldsymbol{M}^{-1/2}\boldsymbol{\Sigma}^{-1}\boldsymbol{M}^{-1/2}$ and $\boldsymbol{F}(\boldsymbol{r}) = \boldsymbol{M}^{-1/2}\boldsymbol{f}(\boldsymbol{M}^{-1/2}\boldsymbol{r})$.

When the nonlinear component $\boldsymbol{f}(\cdot)$ (thus $\boldsymbol{F}(\cdot)$) vanishes, (5) is analytically solvable; in one dimension, the solution is $r(t) = a\cos(\omega t + b)$ where a and b depend on the initial conditions. When the nonlinear component does not vanish, *variation of constants* shows

$$\begin{bmatrix} \boldsymbol{r}(t) \\ \dot{\boldsymbol{r}}(t) \end{bmatrix} = \begin{bmatrix} \cos t\boldsymbol{\Omega} & \boldsymbol{\Omega}^{-1}\sin t\boldsymbol{\Omega} \\ -\boldsymbol{\Omega}\sin t\boldsymbol{\Omega} & \cos t\boldsymbol{\Omega} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}(0) \\ \dot{\boldsymbol{r}}(0) \end{bmatrix} - \int_0^t \begin{bmatrix} \boldsymbol{\Omega}^{-1}\sin(t-s)\boldsymbol{\Omega} \\ \cos(t-s)\boldsymbol{\Omega} \end{bmatrix} \boldsymbol{F}(\boldsymbol{r}(s))\, ds. \tag{6}$$

Cosine and sine here are matrix functions evaluated on eigenvalues with eigenvectors intact.

The solution (6) can be seen as a perturbation of the analytic solution (the first term on the right hand side) due to the nonlinear component. Following Hairer & Lubich (2000), discretizing the integral leads to a class of explicit integrators advancing the pair $(\boldsymbol{r}_i, \dot{\boldsymbol{r}}_i)$ forward for time h:

$$\begin{cases} \boldsymbol{r}_{i+1} = \cos(h\boldsymbol{\Omega})\boldsymbol{r}_i + \boldsymbol{\Omega}^{-1}\sin(h\boldsymbol{\Omega})\dot{\boldsymbol{r}}_i \\ \qquad\qquad - \dfrac{1}{2}h^2\boldsymbol{\psi}(h\boldsymbol{\Omega})\boldsymbol{F}(\boldsymbol{\phi}(h\boldsymbol{\Omega})\boldsymbol{r}_i) \\ \dot{\boldsymbol{r}}_{i+1} = -\boldsymbol{\Omega}\sin(h\boldsymbol{\Omega})\boldsymbol{r}_i + \cos(h\boldsymbol{\Omega})\dot{\boldsymbol{r}}_i \\ \qquad\qquad - \dfrac{1}{2}h(\boldsymbol{\psi}_0(h\boldsymbol{\Omega})\boldsymbol{F}(\boldsymbol{\phi}(h\boldsymbol{\Omega})\boldsymbol{r}_i) \\ \qquad\qquad\quad + \boldsymbol{\psi}_1(h\boldsymbol{\Omega})\boldsymbol{F}(\boldsymbol{\phi}(h\boldsymbol{\Omega})\boldsymbol{r}_{i+1})) \end{cases} \tag{7}$$

These integrators are parameterized by the *filter functions* $\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1 : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$. For HMC to converge, we enforce the following criteria on the filters:

function EXPONENTIAL-STEP($(\boldsymbol{q}_0, \boldsymbol{p}_0); h, L$)
  ▷ Transform into the new variable
  $(\boldsymbol{r}_0, \dot{\boldsymbol{r}}_0) \leftarrow (\boldsymbol{M}^{1/2}\boldsymbol{q}_0, \boldsymbol{M}^{-1/2}\boldsymbol{p}_0)$
  ▷ Precompute the following matrices & the filters
  $(\boldsymbol{C}, \boldsymbol{S}) \leftarrow (\cos(h\boldsymbol{\Omega}), \sin(h\boldsymbol{\Omega}))$
  for $i \leftarrow 0, 1, \ldots, L-1$
    $\boldsymbol{r}_{i+1} \leftarrow \boldsymbol{C}\boldsymbol{r}_i + \boldsymbol{\Omega}^{-1}\boldsymbol{S}\dot{\boldsymbol{r}}_i - \frac{1}{2}h^2\boldsymbol{\psi}\boldsymbol{F}(\boldsymbol{\phi}\boldsymbol{r}_i)$
    $\dot{\boldsymbol{r}}_{i+1} \leftarrow \left| \begin{array}{l} -\boldsymbol{\Omega}\boldsymbol{S}\boldsymbol{r}_i + \boldsymbol{C}\dot{\boldsymbol{r}}_i \\ \quad -\frac{1}{2}h(\boldsymbol{\psi}_0\boldsymbol{F}(\boldsymbol{\phi}\boldsymbol{r}_i) + \boldsymbol{\psi}_1\boldsymbol{F}(\boldsymbol{\phi}\boldsymbol{r}_{i+1})) \end{array} \right.$
  return $(\boldsymbol{M}^{-1/2}\boldsymbol{r}_L, \boldsymbol{M}^{1/2}\dot{\boldsymbol{r}}_L)$ as $(\boldsymbol{q}^*, \boldsymbol{p}^*)$

Figure 2. Exponential integrator.

- Consistency and accuracy to second order: $\boldsymbol{\phi}(0) = \boldsymbol{\psi}(0) = \boldsymbol{\psi}_0(0) = \boldsymbol{\psi}_1(0) = 1$
- Reversibility: $\boldsymbol{\psi}(\cdot) = \mathrm{sinc}(\cdot)\boldsymbol{\psi}_1(\cdot)$ and $\boldsymbol{\psi}_0(\cdot) = \cos(\cdot)\boldsymbol{\psi}_1(\cdot)$
- Symplecticity: $\boldsymbol{\psi}(\cdot) = \mathrm{sinc}(\cdot)\boldsymbol{\phi}(\cdot)$

These conditions permit several choices of filters. We adopt two basic options:

Simple filters (Deuflhard, 1979)**:**

$$\boldsymbol{\phi} = 1, \quad \boldsymbol{\psi} = \mathrm{sinc}(\cdot), \quad \boldsymbol{\psi}_0 = \cos(\cdot), \quad \boldsymbol{\psi}_1 = 1$$

Mollified filters (García-Archilla et al., 1998)**:**

$$\boldsymbol{\phi} = \mathrm{sinc}(\cdot), \quad \boldsymbol{\psi} = \mathrm{sinc}^2(\cdot), \quad \boldsymbol{\psi}_0 = \cos(\cdot)\mathrm{sinc}(\cdot), \quad \boldsymbol{\psi}_1 = \mathrm{sinc}(\cdot)$$

The **mollified filters** promote smooth evolution of the Hamiltonian. In contrast, the **simple filters** do not filter the argument of the nonlinearity. In this case, numerical error can oscillate with an amplitude growing with the step size, increasing the likelihood for rejected HMC samples.

Figure 2 lists the steps of our proposed exponential integrator for HMC—it is a drop-in replacement for leapfrog in Figure 1. The integrator is explicit and with many operations precomputable, so iterations are comparable in speed to those from leapfrog. See the supplementary material for more details. This method is stable, however, when $\boldsymbol{F}(\boldsymbol{r})$ is small relative to $\boldsymbol{\Omega}^2\boldsymbol{r}$. Correctness is given by the following proposition, proved in the supplementary material:

Proposition 1. *HMC using the integrator in Figure 2 is convergent with equilibrium distribution* $p(\boldsymbol{q})$.

3.2. Strategies for Decomposing

$U(\boldsymbol{q}) = -\log p(\boldsymbol{q})$ may not be given in form (3) naturally. In this case, we separate out the $\boldsymbol{\Sigma}^{-1}\boldsymbol{q}$ term by approximating $p(\boldsymbol{q})$ with a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Without loss of generality, we assume $\boldsymbol{\mu} = \boldsymbol{0}$ by shifting the distribution by $\boldsymbol{\mu}$ to achieve so. Our decomposition then becomes

$$\nabla_{\boldsymbol{q}} U(\boldsymbol{q}) = \boldsymbol{\Sigma}^{-1}\boldsymbol{q} + \underbrace{(-\nabla_{\boldsymbol{q}}\log p(\boldsymbol{q}) - \boldsymbol{\Sigma}^{-1}\boldsymbol{q})}_{\boldsymbol{f}(\boldsymbol{q})}. \tag{8}$$

Gaussian approximation provides a natural decomposition for our integrator. The integrator is exact when $\boldsymbol{f}(\boldsymbol{q}) = 0$, implying that HMC for Gaussian distributions using this integrator *never* rejects a sample. Even if $\boldsymbol{\Sigma}^{-1}\boldsymbol{q}$ invokes high-frequency oscillatory behavior, e.g. when $p(\boldsymbol{q})$ is sharply peaked, the numerical drift of this integrator is only a function of its behavior on $\boldsymbol{f}(\boldsymbol{q})$.

We consider a few Gaussian approximation strategies:

Laplace approximation Let $\boldsymbol{\theta}^*$ be a local maximum (i.e., a mode) of $p(\boldsymbol{\theta})$. Then we can write

$$-\log p(\boldsymbol{\theta}) \approx -\log p(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^{\mathrm{T}} \boldsymbol{H}_{\boldsymbol{\theta}^*} (\boldsymbol{\theta} - \boldsymbol{\theta}^*)$$

where $\boldsymbol{H}$ is the Hessian of $-\log p(\boldsymbol{\theta})$. Hence, we can approximate $p(\boldsymbol{\theta})$ with a Gaussian $\mathcal{N}(\boldsymbol{\theta}^*, \boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1})$. The inverse Hessian can then be used in our decomposition.

Empirical statistics We run a probing MCMC sampler, such as HMC with leapfrog, to generate preliminary samples $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m$. We then use the sample mean and covariance to form a Gaussian approximation to kickstart exponential HMC. We then accumulate a few more samples to refine our estimate of the mean and covariance and form a new approximating Gaussian. The process can be iterated and converges asymptotically to the true distribution.

Manifold structures Inspired by Riemann Manifold HMC (RMHMC) in (Girolami & Calderhead, 2011), we can leverage geometric properties of the sample space. Similar to the strategy above using empirical statistics, we update our mean and covariance estimates after every m samples. We take the sample mean of the m samples as the updated mean and exploit the location-specific Fisher-Rao metric defined in (Girolami & Calderhead, 2011) to update the covariance. Namely, we set the covariance to be the inverse of the average of the metrics corresponding to the m samples. Note that if the distribution is Gaussian, this average simplifies to the inverse covariance matrix.

In contrast to the Laplace approximation, which provides a global approximation centered at a mode $\boldsymbol{\theta}^*$ of $p(\boldsymbol{\theta})$, manifold structures incorporate local information: Each Fisher-Rao metric defines a local Gaussian approximation.

In §5, we extensively study the empirical effectiveness of our exponential integrator with these strategies.

4. Related Work

Obtaining a high number of effective samples in a short amount of computation time is a core research problem in Hamiltonian Monte Carlo. Girolami & Calderhead (2011) adapts the mass matrix $\boldsymbol{M}$ to the manifold structure of the sample space, performing location-specific preconditioning. Hoffman & Gelman (2014) and Wang et al. (2013) automatically choose steps h and iteration counts L. Sohl-Dickstein et al. (2014) extend trajectories to accepting states to avoid rejection. For large datasets, Chen et al. (2014) formulate HMC using stochastic gradients to reduce computational cost. Many those approaches treat the numerical integrator as a black box, often using the simple leapfrog method. Our aim is to establish the benefits of more sophisticated and computationally efficient integrators. As such, the integrator we propose can be substituted into any of those approaches.

Splitting is a common alternative that bears some similarity to our work (Hairer et al., 2006). It separates Hamiltonians into multiple terms and alternating between integrating each independently. In statistical physics, splitting is applied in lattice field theory (Kennedy & Clark, 2007; Clark et al., 2010) and in quantum chromodynamics (Takaishi, 2002). In statistics, Neal (2011) and Shahbaba et al. (2014) split over data or into Gaussian/non-Gaussian parts. The latter strategy, also used by Beskos et al. (2011), has similar structure to ours, never rejecting samples from Gaussians. *Our approximations (§3.2) can be used in such splittings. But while their integrator for the non-Gaussian part has no knowledge of the Gaussian part, we use a stable, geometrically-motivated combination.* Blanes et al. (2014) also split using Gaussian model problems; while they provide theoretical groundwork for stability and accuracy of HMC, they study variations of leapfrog. Pakman & Paninski (2014) report exact HMC integration for Gaussians and truncated Gaussians but do not consider the general case.

Our proposed exponential integrator represents a larger departure from leapfrog and splitting integrators. Exponential integration was motivated by the observation that traditional integrators do not exploit analytical solutions to linear ODEs (Hersch, 1958; Hochbruck et al., 1998). We employ a trigonometric formula, introduced by Gautschi (1961). This was combined with the trapezoidal rule (Deuflhard, 1979) and extended using filters (García-Archilla et al., 1999). There exist many variations of the proposed exponential integrator; the one in this paper is chosen for its simplicity and effectiveness in HMC.

5. Experiments

We validate the effectiveness of exponential HMC on both synthetic and real data and compare to alternative methods. All the algorithms are implemented in Matlab to ensure a fair comparison[1], especially when we evaluate different approaches for their computational cost.

We present major results in what follows. Additional empirical studies are described in the supplementary material.

[1]Code: https://github.com/pujols/Exponential-HMC

5.1. Setup

Variants of exponential integrators We use expHMC to denote the exponential integrator where the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ are computed from parameters of the distribution, $\widehat{\text{expHMC}}$ for integration with empirically estimated $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and rmExpHMC for those obtained from manifold structures; see §3.2 for details. We only consider mollified filters (§3.1) in this section and defer comparison to the simple filters to the supplementary material. We take $\boldsymbol{M} = \boldsymbol{I}$ for the mass matrix in (1), unless stated otherwise.

For expHMC, we directly use the parameters of Gaussian distributions on synthetic data, and apply the Laplace approximation on real data. For $\widehat{\text{expHMC}}$, we run HMC with leapfrog for burn-in and use the last N_1 burn-in samples to estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Both are updated after sampling every N_2 samples. rmExpHMC follows a similar procedure to $\widehat{\text{expHMC}}$, yet in updating, it uses information only from the N_2 new samples.

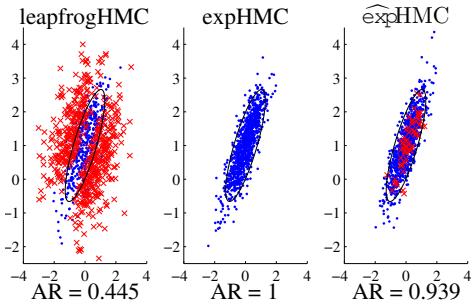
Evaluation criteria We evaluate performance by computing acceptance rates (AR) and the minimum effective sample sizes (min ESS). The min ESS, suggested by Girolami & Calderhead (2011), measures sampling efficiency. To take computation time into consideration, we report min ESS/sec and the relative speed (RS) to the leapfrog method.

5.2. Synthetic Example: Gaussian Distribution

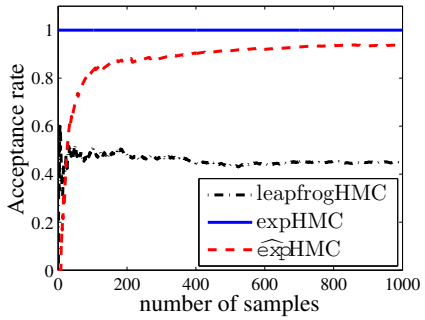
As a sanity check and to illustrate the basic properties of exponential integration, we first compare the proposed exponential integrator to the leapfrog (leapfrogHMC) integrator on the 2D Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, where $\boldsymbol{\Sigma}_g$ has eigenvalues $\{1, 0.1\}$. We run 200 iterations for burn-in and collect 1000 samples, with $(h, L) = (0.6, 8)$ for all methods, a fairly large step to encourage exploration of sample space. For $\widehat{\text{expHMC}}$, we set $(N_1, N_2) = (50, 20)$.

Samples and acceptance rates are shown in Figure 3. As expected, expHMC accepts samples unconditionally. The acceptance rate of $\widehat{\text{expHMC}}$ increases as it updates the Gaussian approximation, since the sample statistics begin to estimate the mean and covariance accurately.

Next, we examine how the eigenvalues of $\boldsymbol{\Sigma}_g$ affect acceptance. We construct $\boldsymbol{\Sigma}_g$ with eigenvalues $\{1, \lambda\}$, using identical parameters as above except $(h, L) = (0.12, 10)$; smaller h increases leapfrog acceptance rates. We vary $\lambda \in [2^{-8}, 2^0]$, showing the acceptance rate of each method in Figure 4a. Again, expHMC maintains acceptance rate 1, while the leapfrog acceptance degrades as λ shrinks. The acceptance rate of $\widehat{\text{expHMC}}$ is also affected by λ. As the sample size increases, however, the acceptance rate of $\widehat{\text{expHMC}}$ increases and converges to 1, as in Figure 4b.



(a) Accepted (blue) and rejected (red) proposals of each method



(b) The accumulated acceptance rate

Figure 3. On a 2D Gaussian, expHMC always accepts proposals, outperforming leapfrog under identical (h, L). The acceptance rate (AR) of $\widehat{\text{expHMC}}$ increases as samples are collected.

Preconditioning One way to mitigate stiffness in ODEs of HMC (e.g., small λ in the previous example) is to adjust the mass matrix $\boldsymbol{M}$ (Neal, 2011). HMC with $\boldsymbol{M} = \boldsymbol{\Sigma}_g^{-1}$ is equivalent to sampling from a homogeneous Gaussian. However, even with preconditioning, leapfrogHMC has time step restrictions on Gaussian distributions. We further demonstrate in §5.3 that preconditioning alone is insufficient to deal with stiff ODEs on more complicated distributions.

We include more experiments on synthetic examples (e.g., mixtures of Gaussians) in the supplementary material.

5.3. Real Data: Bayesian Logistic Regression

We apply the proposed methods to sampling from the posterior of Bayesian logistic regression (BLR), comparing to leapfrogHMC and Riemann manifold HMC (RMHMC) by Girolami & Calderhead (2011). We also compare to an accelerated version of RMHMC using Lagrangian dynamics (e-RMLMC), by Lan et al. (2012).

Model Given a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ with N instances, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{-1, 1\}$ is a binary label, the posterior distribution $p(\boldsymbol{\theta})$ of Bayesian logistic regression is defined as

$$p(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta}) \prod_{i=1}^{N} \sigma(y_i(\boldsymbol{\theta}^T \boldsymbol{x}_i)),$$

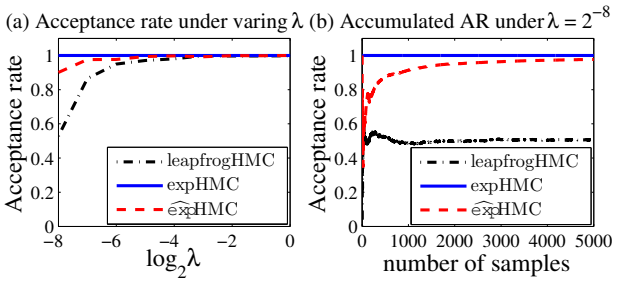


Figure 4. (a) On a 2D Gaussian, expHMC has acceptance rate 1 under varying λ (with h fixed). (b) The acceptance rate of $\widehat{\text{exp}}$HMC (at $\lambda = 2^{-8}$) continuously increases as more samples are collected.

where $\pi(\boldsymbol{\theta})$ is a prior on $\boldsymbol{\theta}$ and $\sigma(z) = (1 + e^{-z})^{-1}$.

Dataset We consider five datasets from the UCI repository (Bache & Lichman, 2013): Ripley, Heart, Pima Indian, German credit, and Australian credit. They have 250 to 1000 instances with feature dimensions between 2 and 24. Each dimension is normalized to have zero mean and unit variance to ensure that $\boldsymbol{M} = \boldsymbol{I}$ is a fair choice for leapfrog and exponential integration.

Experimental setting We consider the homogeneous Gaussian prior $\mathcal{N}(\mathbf{0}, \sigma\boldsymbol{I})$ with $\sigma \in \{0.01, 1, 100\}$. We take $L = 100$ and manually set h in leapfrogHMC to achieve acceptance rates in $[0.6, 0.9]$ for each combination of dataset and σ, as suggested by Betancourt et al. (2014a). The same (h, L) is used for expHMC, $\widehat{\text{exp}}$HMC, and rmExpHMC, along with $(2h, L/2)$ and $(4h, L/4)$ to test larger steps. We use parameters for RMHMC and e-RMLMC from their corresponding references. We set $(N_1, N_2) = (500, 250)$ for $\widehat{\text{exp}}$HMC and apply the same location-specific metric defined by Girolami & Calderhead (2011) along with $(N_1, N_2) = (500, 500)$ for rmExpHMC. For expHMC, $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ come from the Laplace approximation. We collect 5000 samples after 5000 iterations of burn-in as suggested in previous work, repeating for 10 trials. To ensure high effective sample size (ESS), for all the compared methods we uniformly sample L from $\{1, \ldots, L\}$ in each iteration, as suggested by Neal (2011).

Experimental results We summarize results on the Pima Indian dataset in Table 1; remaining results are in the supplementary material. Our three methods expHMC, $\widehat{\text{exp}}$HMC, and rmExpHMC outperform leapfrogHMC in terms of min ESS and acceptance rate, with similar speed as leapfrogHMC. RMHMC and e-RMLMC have the highest min ESS but execute much slower than leapfrogHMC, leading to poor relative speeds. Besides, expHMC, $\widehat{\text{exp}}$HMC, and rmExpHMC achieve acceptance rate > 0.8 under the cases $(2h, L/2)$ and $(4h, L/4)$. The high acceptance rates under larger step sizes allow them to take fewer steps during sampling, raising the relative speed. Enlarging the steps in leapfrogHMC, RMHMC, and e-RMLMC, however, leads to a significant drop in acceptance and thus is not presented.

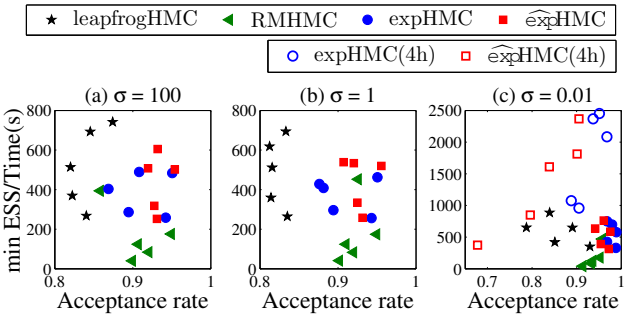


Figure 5. Acceptance rate and min ESS/sec of each method on the five datasets, with $\sigma \in \{0.01, 1, 100\}$ (the variance of the prior in BLR). As illustrated, expHMC and $\widehat{\text{exp}}$HMC achieve both higher acceptance rate and/or higher min ESS/sec. Best viewed in color.

Table 1. Pima Indian dataset ($d = 7, N = 532$, and 8 regression coefficients). "AR" denotes acceptance rate, and "RS" denotes relative speed.

METHOD	TIME(S)	MIN ESS	S/MIN ESS	RS	AR
$\sigma = 100$					
leapfrogHMC	6.3	3213	0.0019	1	0.82
RMHMC	28.3	4983	0.0057	0.34	0.95
e-RMLMC	19.8	4948	0.0040	0.48	0.95
expHMC (h, L)	7.8	3758	0.0021	0.94	0.95
expHMC (2h, L/2)	4.1	2694	0.0015	1.29	0.88
expHMC (4h, L/4)	2.2	2555	0.0008	2.30	0.88
$\widehat{\text{exp}}$HMC (h, L)	7.7	3876	0.0020	0.98	0.95
$\widehat{\text{exp}}$HMC (2h, L/2)	4.0	3025	0.0013	1.47	0.89
$\widehat{\text{exp}}$HMC (4h, L/4)	2.1	2845	0.0008	2.58	0.85
rmExpHMC (h, L)	7.8	4143	0.0019	1.00	0.97
rmExpHMC (2h, L/2)	4.1	3229	0.0013	1.49	0.91
rmExpHMC (4h, L/4)	2.2	3232	0.0007	**2.78**	0.90
$\sigma = 0.01$					
leapfrogHMC	6.0	3865	0.0015	1	0.89
RMHMC	28.3	4987	0.0057	0.27	0.95
e-RMLMC	20.2	4999	0.0040	0.38	0.95
expHMC (h, L)	7.3	4239	0.0017	0.89	0.99
expHMC (2h, L/2)	3.8	4164	0.0009	1.69	0.97
expHMC (4h, L/4)	2.0	4226	0.0005	**3.21**	0.97
$\widehat{\text{exp}}$HMC (h, L)	7.1	4141	0.0017	0.90	0.98
$\widehat{\text{exp}}$HMC (2h, L/2)	3.7	3771	0.0010	1.57	0.93
$\widehat{\text{exp}}$HMC (4h, L/4)	2.0	3540	0.0006	2.79	0.90
rmExpHMC (h, L)	7.3	4316	0.0017	0.89	0.99
rmExpHMC (2h, L/2)	3.8	4284	0.0009	1.68	0.97
rmExpHMC (4h, L/4)	2.0	3758	0.0005	2.80	0.93

As σ shrinks, the ODE becomes stiff and the relative speeds of all our methods improve, demonstrating the effectiveness of exponential integration on stiff problems. This is highlighted in Figure 5, where we plot each method by acceptance rate and min ESS/sec on the five datasets; each point corresponds to a method-dataset pair. expHMC and $\widehat{\text{exp}}$HMC achieve high acceptance rates and high min ESS/sec (top right) compared to leapfrogHMC and RMHMC, especially for small σ. For clarity, rmExpHMC and e-RMLMC are not shown; they perform similarly to $\widehat{\text{exp}}$HMC and RMHMC, respectively.

Table 2. MNIST dataset with digit 7 and 9 ($d = 100, N = 12214$). "AR" denotes acceptance rate, and "RS" denotes relative speed.

METHOD	TIME(S)	MIN ESS	S/MIN ESS	RS	AR
$\boldsymbol{\sigma = 0.01}$					
leapfrogHMC	1215	22164	0.055	1	0.88
expHMC (h, L)	1413	23720	0.060	0.92	0.96
expHMC (2h, L/2)	750	16643	0.045	1.22	0.87
$\widehat{\text{exp}}$HMC (h, L)	1437	23080	0.062	0.88	0.94
$\widehat{\text{exp}}$HMC (2h, L/2)	779	11919	0.065	0.84	0.73
rmExpHMC (h, L)	1450	24347	0.060	0.92	0.97
rmExpHMC (2h, L/2)	781	19623	0.040	**1.38**	0.90

Scalability Sophisticated integrators, like that for RMHMC, lose speed when dimensionality increases. To investigate this effect, we sample from the BLR posterior on digits 7 and 9 from the MNIST dataset (12214 training instances); each instance is reduced to 100 dimensions by PCA. We set $(h, L) = (0.013, 30)$, with $(N_1, N_2) = (2000, 500)$ for $\widehat{\text{exp}}$HMC and $(N_1, N_2) = (2000, 2000)$ for rmExpHMC. The prior $\sigma = 0.01$. Performance averaged over 10 trials is summarized in Table 2, for 30000 samples after 10000 iterations of burn-in.

expHMC, $\widehat{\text{exp}}$HMC, and rmExpHMC have runtimes comparable to leapfrogHMC, scaling up well. High acceptance rates and better relative speed under larger step sizes also suggest a more efficient sampling by our methods. RMHMC and e-RMLMC are impractical on this task as they take around 1 day to process 30000 samples. Specifically, the speed improvement of e-RMLMC over RMHMC diminishes for large-scale problems like this one, as shown by Lan et al. (2012) and analyzed by Betancourt et al. (2014b).

Preconditioning In BLR, a suitable choice of (constant) $\boldsymbol{M}$ is the Hessian of $-\log p(\boldsymbol{\theta})$ at MAP, i.e., $\boldsymbol{\Sigma}^{-1}$ by Laplace approximation. We precondition both our method and leapfrog for fair comparison.[2] With step sizes so that preconditioned leapfrog PLHMC has > 0.85 acceptance, expHMC still always results in a higher acceptance rate. Further increasing the step sizes, acceptances of PLHMC drop significantly while ours does not suffer as strongly. More importantly, as σ shrinks (the ODE becomes stiffer), we observe the same performance gain as in Figure 5. These demonstrate the advantage of our method even with preconditioning. Details are in the supplementary material.

5.4. Real Data: Independent Component Analysis

Given N d-dimensional observations $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^N$, independent component analysis (ICA) aims to find the demixing matrix $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ so as to recover the original d mutually independent sources $\{\boldsymbol{y}_i \in \mathbb{R}^d\}_{i=1}^N$. We take the formulation introduced by Hyvärinen & Oja (2000) that defines a posterior distribution $p(\boldsymbol{W}|\boldsymbol{X})$. We then compare our methods to leapfrogHMC in sampling from the posterior.

[2]Both solve the same ODE (2), using different integrators.

Table 3. MEG dataset ($d = 5, N = 17730$, and 25 sampling dimensions). "AR" denotes acceptance rate, and "RS" denotes relative speed.

METHOD	TIME(S)	MIN ESS	S/MIN ESS	RS	AR
$\boldsymbol{\sigma = 100}$					
leapfrogHMC (h, L)	196	2680	0.073	1	0.85
leapfrogHMC (2h, L/2)	109	498	0.220	0.33	0.26
$\widehat{\text{exp}}$HMC (h, L)	207	3256	0.063	1.15	0.95
$\widehat{\text{exp}}$HMC (2h, L/2)	109	2579	0.042	**1.73**	0.82
$\widehat{\text{exp}}$HMC (4h, L/4)	62	1143	0.054	1.34	0.64

We do not compare to RMHMC and e-RMLMC as these methods do not scale well to this problem.

Model The joint probability of $(\boldsymbol{X}, \boldsymbol{W})$ is

$$p(\boldsymbol{X}, \boldsymbol{W}) = |\det(\boldsymbol{W})|^N \prod_{i=1}^N \{\prod_{j=1}^d p_j(\boldsymbol{w}_j^T \boldsymbol{x}_i)\} \times \prod_{kl} \mathcal{N}(W_{kl}; 0, \sigma), \tag{9}$$

where we use a Gaussian prior over $\boldsymbol{W}$, with $\boldsymbol{w}_j^T$ the jth row of $\boldsymbol{W}$. We set $p_j(y_{i_j}) = \{4\cosh^2(\frac{1}{2}y_{i_j})\}^{-1}$ with $\boldsymbol{y}_i = \boldsymbol{W}\boldsymbol{x}_i$, as suggested by Korattikara et al. (2014).

Dataset We experiment on the MEG dataset (Vigário et al., 1997), which contains 122 channels and 17730 timepoints. We extract the first 5 channels for our experiment (i.e, $\boldsymbol{W} \in \mathbb{R}^{5\times 5}$), leading to samples with 25 dimensions.

Experimental setting We set $\sigma = 100$ for the Gaussian prior (9). We set $L = 50$ and manually adjust h in leapfrogHMC to achieve acceptance rates in $[0.6, 0.9]$. The same (h, L) is used for our methods, along with $(2h, L/2)$ and $(4h, L/4)$ to test larger steps. We only examine $\widehat{\text{exp}}$HMC, given its tractable strategy for Gaussian approximation. The parameters are set to $(N_1, N_2) = (500, 250)$. We collect 5000 samples after 5000 iterations of burn-in, repeating for 5 trials. In each iteration, we sample L uniformly from $\{1, ..., L\}$ to ensure high ESS, as in §5.3.

Experimental results We summarize the results in Table 3. With comparable computational times, $\widehat{\text{exp}}$HMC produces a much higher min ESS and acceptance rate than leapfrogHMC. As the step size increases, acceptances of leapfrogHMC drop drastically, yet $\widehat{\text{exp}}$HMC can still maintain high acceptance and ESS, leading to better relative speed for efficient sampling.

6. Comparison to Splitting

We contrast our approach with the Gaussian splitting by Shahbaba et al. (2014). After applying the Laplace approximation, they integrate by alternating between the

nonlinear and linear terms. The linear part is integrated in closed form, while the nonlinear part is integrated using a "forward Euler" step. In the notation of Figure 2, we can write an iteration of their integrator as:

$$\begin{bmatrix} \dot{\boldsymbol{r}}^0_{i+1} \\ \boldsymbol{r}_{i+1} \\ \dot{\boldsymbol{r}}^1_{i+1} \\ \dot{\boldsymbol{r}}_{i+1} \end{bmatrix} \begin{array}{l} \leftarrow \dot{\boldsymbol{r}}_i - \frac{1}{2}h\boldsymbol{F}(\boldsymbol{r}_i) \\ \leftarrow \begin{bmatrix} \cos h\boldsymbol{\Omega} & \boldsymbol{\Omega}^{-1}\sin h\boldsymbol{\Omega} \\ -\boldsymbol{\Omega}\sin h\boldsymbol{\Omega} & \cos h\boldsymbol{\Omega} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_i \\ \dot{\boldsymbol{r}}^0_{i+1} \end{bmatrix} \\ \leftarrow \dot{\boldsymbol{r}}^1_{i+1} - \frac{1}{2}h\boldsymbol{F}(\boldsymbol{r}_{i+1}) \end{array}$$

The nonlinearity vanishes when $\boldsymbol{f}(\cdot) = 0$, providing exact integration for Gaussians.

The key difference between our integrator and theirs is in the treatment of $\boldsymbol{f}(\cdot)$. Their integrator treats $\boldsymbol{f}(\cdot)$ independently from the Gaussian part, stepping variables using a simple explicit integrator. Variation of constants (6), on the other hand, models the deviation from Gaussian behavior directly and steps all variables simultaneously.

Figure 6 illustrates with a simple setting where this difference leads to significant outcomes. Here, we show paths corresponding to a single iteration of HMC with $L = 1$ for a Gaussian. For leapfrog integration these paths would be straight lines, but both splitting and exponential integration follow elliptical paths. We estimate the mean incorrectly, however, so $\boldsymbol{f}(\cdot)$ is a constant but nonzero function. Splitting provides a velocity impulse and then follows an ellipse centered about the incorrect mean. Our integrator handles constant functions $\boldsymbol{f}(\cdot)$ exactly, recovering the correct path.

This example illustrates the robustness we expect from exponential integration. For nonlinear distributions, we can use the "empirical" approach in §3.2 to determine a Gaussian approximation. Empirical means and covariances likely are inexact, implying that even for Gaussian distributions, splitting with this approximation may reject samples. Our integrator better handles the coupling between the Gaussian and non-Gaussian parts of $p(\boldsymbol{q})$ and hence can be more robust to approximation error.

A more extensive empirical comparison to splitHMC is given in the supplementary material. In general, splitHMC performs similarly to our methods with simple filters (cf. §3.1), as discussed in §7. Our methods with mollified filters, however, handle departure from Gaussian approximation more stably, outperforming splitHMC.

7. Discussion and Conclusion

The integrator for HMC has a strong effect on acceptance rates, ESS, and other metrics of convergence. While large integration times encourage exploration, numerically solving the underlying ODE using leapfrog requires many substeps to avoid high rejection rates. Although RMHMC deals with this issue by locally adjusting the mass matrix $\boldsymbol{M}$, it suffers from significantly increased computation costs and poor scalability. In contrast, our method is based on an efficient explicit integrator that is robust to stiffness, even without preconditioning.

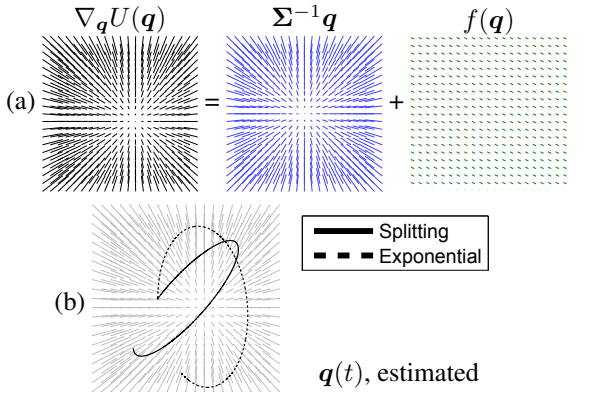


Figure 6. Comparison between splitting and exponential integration. (a) The acceleration vector field corresponding to a Gaussian distribution is shifted, leading to a nonzero $\boldsymbol{f}(\boldsymbol{q})$; (b) this simple but nonzero $\boldsymbol{f}(\boldsymbol{q})$ is enough to perturb the initial velocity $(0, 1)$ of the elliptical path for splitting, shearing to the right.

The choice of numerical integrators is a "meta" parameter for HMC, since each integrator discretizes the same dynamical equations. Our extensive studies have revealed,

- When the step size is small, leapfrog suffices.
- When the step size is large enough that acceptance for leapfrog declines, splitting and exponential integration with simple filters maintain stability. In this case, mollification sacrifices too much accuracy for stability. Details are in the supplementary material.
- For large step sizes, exponential HMC with mollified filters avoids degrading performance.
- Empirical means and covariances yield stable Gaussian approximations for exponential integrators.

While our experiments focus on basic HMC, future studies can assess all options for HMC, accounting e.g. for extensions like (Hoffman & Gelman, 2014) and (Sohl-Dickstein et al., 2014). Additional research also may reveal better integrators *specifically* for HMC. For instance, the exponential integrators introduced by Rosenbrock (1963) use local rather than global approximations that may be suitable for distributions like the mixture of Gaussians, whose Gaussian approximation should change depending on locations; the challenge is to maintain symplecticity and reversibility. Exponential integration also can be performed on Lie groups, providing avenues for non-Euclidean sampling. Regardless, exponential integrators remain practical, straightforward, and computationally inexpensive alternatives for HMC with the potential to improve sampling.

Acknowledgments

F. S. and W. C. are supported by ARO Award # W911NF-12-1-0241, ONR # N000141210066, and Alfred P. Sloan Fellowship. D. M. is supported by the Max Planck Center for Visual Computing and Communication.

References

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Beskos, A., Pinski, F. J., Sanz-Serna, J. M., and Stuart, A. M. Hybrid Monte Carlo on Hilbert spaces. *Stochastic Processes and their Applications*, 121(10):2201–2230, 2011.

Betancourt, M. J., Byrne, S., and Girolami, M. Optimizing the integrator step size for Hamiltonian Monte Carlo. *arXiv preprint arXiv:1411.6669*, 2014a.

Betancourt, M. J., Byrne, S., Livingstone, S., and Girolami, M. The geometric foundations of Hamiltonian Monte Carlo. *arXiv preprint arXiv:1410.5110*, 2014b.

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

Chen, T., Fox, E. B., and Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *Proc. ICML*, June 2014.

Clark, M. A., Joó, B., Kennedy, A. D., and Silva, P. J. Better HMC integrators for dynamical simulations. In *Proc. International Symposium on Lattice Field Theory*, 2010.

Deuflhard, P. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Journal of Applied Mathematics and Physics*, 30:177–189, 1979.

García-Archilla, B., Sanz-Serna, J. M., and Skeel, R. D. Long-time-step methods for oscillatory differential equations. *SIAM Journal on Scientific Computing*, 20(3):930–963, 1998.

García-Archilla, B., Sanz-Serna, J. M., and Skeel, R. D. Long-time-step methods for oscillatory differential equations. *SIAM Journal on Scientific Computing*, 20:930–963, 1999.

Gautschi, W. Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numerische Mathematik*, 3:381–397, 1961.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. R. *Bayesian Data Analysis*. Chapman and Hall, CRC, 2004.

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

Hairer, E. and Lubich, C. Long-time energy conservation of numerical methods for oscillatory differential equations. *SIAM J. Numer. Anal.*, 38(2):414–441, July 2000.

Hairer, E., Lubich, C., and Wanner, G. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

Hersch, J. Contribution à la méthode des équations aux différences. *Zeitschrift für Angewandte Mathematik und Physik*, 9:129–180, 1958.

Hochbruck, M., Lubich, C., and Selfhofer, H. Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing*, 19(5):1552–1574, 1998.

Hoffman, M. D. and Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, January 2014.

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

Kennedy, A. D. and Clark, M. A. Speeding up HMC with better integrators. In *Proc. International Symposium on Lattice Field Theory*, 2007.

Korattikara, A., Chen, Y., and Welling, M. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proc. ICML*, pp. 181–189, 2014.

Lan, S., Stathopoulos, V., Shahbaba, B., and Girolami, M. Lagrangian dynamical Monte Carlo. *arXiv preprint arXiv:1211.3759*, 2012.

Neal, R. M. MCMC using Hamiltonian dynamics. In Brooks, Steve, Gelman, Andrew, Jones, Galin, and Meng, Xiao-Li (eds.), *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

Pakman, A. and Paninski, L. Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014.

Robert, C. and Casella, G. *Monte Carlo Statistical Methods*. Springer, 2004.

Rosenbrock, H. H. Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, 5(4):329–330, 4 1963.

Shahbaba, B., Lan, S., Johnson, W. O., and Neal, R. M. Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24 (3):339–349, 2014.

Sohl-Dickstein, J., Mudigonda, M., and Deweese, M. Hamiltonian Monte Carlo without detailed balance. In *Proc. ICML*, pp. 719–726, 2014.

Takaishi, T. Higher order hybrid Monte Carlo at finite temperature. *Physics Letters B*, 540(1–2):159–165, 2002.

Vigário, R., Särelä, J., and Oja, E. MEG data for studies using independent component analysis, 1997. URL http://research.ics.aalto.fi/ica/eegmeg/MEG_data.html.

Wang, Z., Mohamed, S., and Nando, D. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *Proc. ICML*, volume 28, pp. 1462–1470, May 2013.