Langevin Markov Chain Monte Carlo with stochastic gradients

Charles Matthews
Department of Statistics
The University of Chicago
Chicago, IL 60615
c.matthews@uchicago.edu

Jonathan Q. Weare
Department of Statistics
James Franck Institute
The University of Chicago
Chicago, IL 60615
weare@uchicago.edu

Abstract

Monte Carlo sampling techniques have broad applications in machine learning, Bayesian posterior inference, and parameter estimation. Typically the target distribution takes the form of a product distribution over a dataset with a large number of entries. For sampling schemes utilizing gradient information it is cheaper for the derivative to be approximated using a random small subset of the data, introducing extra noise into the system. We present a new discretization scheme for underdamped Langevin dynamics when utilizing a stochastic (noisy) gradient. This scheme is shown to bias computed averages to second order in the stepsize while giving exact results in the special case of sampling a Gaussian distribution with a normally distributed stochastic gradient.

1 Introduction

A commonly encountered problem in data science and machine learning applications is the sampling of parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ from a target probability distribution $\pi(\boldsymbol{\theta})$. Typically the target distribution is a product distribution over a dataset $\boldsymbol{y}$ containing N observations $\boldsymbol{y}_i$, along with some prior $\pi_0(\boldsymbol{\theta})$ incorporating knowledge of the overall distribution that acts as a regularizer

$$\pi(\boldsymbol{\theta}) := \pi_0(\boldsymbol{\theta}) \prod_{i=1}^{N} \pi(\boldsymbol{\theta} \,|\, \boldsymbol{y}_i). \tag{1}$$

Such a formulation is commonplace in Bayesian inverse applications, where π is referred to as a posterior distribution. Markov Chain Monte Carlo (MCMC) schemes are an effective method for sampling from such distributions [19, 4], however conventional schemes (see e.g. [15, 17]) require an evaluation of $\log \pi(\boldsymbol{\theta})$ or its gradient to propose new points. Because each evaluation can be prohibitively expensive for large N , there is considerable interest in MCMC schemes that sample the target distribution but do not require access to the entirety of $\boldsymbol{y}$.

In this article we are interested in sampling π using stochastic approximations to the gradient. For a distribution as in (1), the gradient vector (or the *force*) is a sum over the data

$$\boldsymbol{F}(\boldsymbol{\theta}) := \boldsymbol{\nabla} \log \pi_0(\boldsymbol{\theta}) + \sum_{i=1}^{N} \boldsymbol{\nabla} \log \pi(\boldsymbol{\theta} \,|\, \boldsymbol{y}_i). \tag{2}$$

This is used to drive proposals for an MCMC scheme in efficient directions in the D dimensional space. We will refer to $\boldsymbol{F}$ as the exact or 'true' force vector, using all N datapoints to compute the derivative. By contrast we may use a random subset of datapoints to stochastically approximating the

Preprint. Work in progress.

Algorithm 1 *NOGIN* : Noisy Gradient Integrator

Input: $\boldsymbol{\theta}_0$, $h > 0$, $\gamma > 0$, $T > 0$

Initialize: $\boldsymbol{p} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\lambda \leftarrow \sqrt{(1 - e^{-\gamma h})/(1 + e^{-\gamma h})}$
for $t = 1$ to T **do**
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + h\boldsymbol{p}/2$
  $\widetilde{\boldsymbol{F}} \leftarrow \widetilde{\boldsymbol{F}}(\boldsymbol{\theta})$, $\boldsymbol{\Sigma} \leftarrow \mathrm{Cov}(\widetilde{\boldsymbol{F}}(\boldsymbol{\theta}))$, $\boldsymbol{R} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{p} \leftarrow \boldsymbol{p} + h\widetilde{\boldsymbol{F}}/2 + \lambda \boldsymbol{R}$
  $\boldsymbol{p} \leftarrow \left((1-\lambda^2)\boldsymbol{I} - \frac{h^2}{4}\boldsymbol{\Sigma}\right)\left((1+\lambda^2)\boldsymbol{I} + \frac{h^2}{4}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{p}$
  $\boldsymbol{p} \leftarrow \boldsymbol{p} + h\widetilde{\boldsymbol{F}}/2 + \lambda \boldsymbol{R}$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + h\boldsymbol{p}/2$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}$
end for

$O(N)$ expensive true gradient. This idea is widely applicable and a number of schemes have recently been proposed to capitalize on the potential increased efficiency that it offers [7, 6, 24, 14].

The noisy gradient estimator $\widetilde{\boldsymbol{F}}(\boldsymbol{\theta})$ is a random vector estimate with

$$\mathbf{E}\left[\widetilde{\boldsymbol{F}}(\boldsymbol{\theta})\right] = \boldsymbol{F}(\theta), \qquad \mathrm{Cov}\left[\widetilde{\boldsymbol{F}}(\boldsymbol{\theta})\right] = \boldsymbol{\Sigma}(\theta) \tag{3}$$

for positive semidefinite $\boldsymbol{\Sigma}$. There are many such choices of estimator, with much recent work undertaken to increase the available accuracy (quantified by the size of $\boldsymbol{\Sigma}$) and storage requirements (see e.g. [3, 8]). Usually the primary cost of the estimator comes from an evaluation of force terms over some mini-batch $\widetilde{\boldsymbol{y}} \subseteq \boldsymbol{y}$ where $|\widetilde{\boldsymbol{y}}| = n$ for some fixed $n \leq N$. The mini-batch is redrawn uniformly from $\boldsymbol{y}$ at every estimation of $\widetilde{\boldsymbol{F}}$, with $\boldsymbol{\Sigma} \to \mathbf{0}$ as $n \to N$. The simplest choice is using a scaled sum over the mini-batch [18] :

$$\widetilde{\boldsymbol{F}}(\boldsymbol{\theta}) := \boldsymbol{\nabla} \log \pi_0(\boldsymbol{\theta}) + \frac{N}{n}\sum_{i=1}^{n} \boldsymbol{\nabla} \log \pi(\boldsymbol{\theta} \,|\, \tilde{\boldsymbol{y}}_i), \quad \boldsymbol{\Sigma}(\boldsymbol{\theta}) := \frac{N(N-n)}{n}\mathrm{Cov}\left[\{\boldsymbol{f}_i(\boldsymbol{\theta})\}\right] \tag{4}$$

where $\boldsymbol{f}_i(\theta) := \boldsymbol{\nabla} \log \pi(\boldsymbol{\theta} \,|\, \boldsymbol{y}_i)$. In practice $\boldsymbol{\Sigma}$ is estimated using a covariance taken over the mini-batch of n-many $\boldsymbol{f}_i$ terms.

The additional stochastic term introduced into the MCMC proposals can lead to a large bias if not accounted for. The size of this bias and the rate of decorrelation of samples largely differentiates noisy gradient methods, and is the subject of the present article. Given some estimator $\widetilde{\boldsymbol{F}}$ and an estimate for its covariance $\boldsymbol{\Sigma}$, we present a new sampling scheme, referred to as the Noisy Gradient Integrator (or *NOGIN*), in Algorithm 1. The structure of the article is as follows. In Section 2 we set notation for the noisy schemes and place our method in a proper context. Section 3 derives our proposed *NOGIN* scheme from established Langevin dynamics methods. Section 4 gives some analytical results for the expected error from the *NOGIN* scheme. In Section 5 we compare the proposed method against others on some classic machine learning applications. A discussion of outlook and ramifications concludes in Section 6.

2 Noisy Gradient Integration

An effective way to utilize gradient information in MCMC schemes is by proposing points from solution trajectories of ergodic dynamics that sample the target distribution π [4]. For appropriate test functions f, ergodicity implies that for a solution trajectory $\boldsymbol{\theta}(t)$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{\theta}(t)) = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} =: \langle f \rangle.$$

However exact solutions for dynamics involving gradients of complicated, nonlinear π are seldom known, so a discretization scheme is usually employed to advance the state through time, computing a sequence of $\boldsymbol{\theta}_k \approx \boldsymbol{\theta}(kh)$ for a discretization timestep $h > 0$. The discretization introduces

error into the computed trajectory at both finite and infinite time (see e.g. [12]). The infinite-time (sometimes called asymptotic or perfect) sampling bias introduced by the schemes is the difference in the respective averages of f,

$$\text{Bias } = |\langle f \rangle_h - \langle f \rangle|, \qquad \langle f \rangle_h := \lim_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} f(\boldsymbol{\theta}_k),$$

where $\langle f \rangle_h$ is the infinite-time observed average evaluated from numerical computation. The size of this bias can be tuned by decreasing the stepsize h, at the cost of sacrificing the rate of exploration for the scheme (see Section 4.3). Alternatively schemes such as MALA [20] employ an additional Metropolis-Hastings (MH) step to correct for any bias introduced from the discretization, at the cost of rejecting some moves.

In the case of a target distribution in the form of (1), the $O(N)$ cost of the force $\boldsymbol{F}$ or log-likelihood $\log \pi(\boldsymbol{\theta} \,|\, \boldsymbol{y})$ may be prohibitively expensive in terms of memory or computation. Naïvely exchanging $\boldsymbol{F}$ for a stochastic estimate $\widetilde{\boldsymbol{F}}$ in an MCMC algorithm usually leads to a large bias introduced due to the additional unchecked noise.

Our aim in this article is to give a numerical scheme that introduces a bias of order h^2 when using a fixed stepsize with a noisy gradient. Our strategy is to treat the stochasticity of the noisy gradient as an additional random noise term that acts to 'heat' the resulting dynamics. We can appropriately damp the resulting dynamics, assuming we have some knowledge of the estimator being used, specifically the covariance of the noise $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. We shall not consider correcting for the effects from moments higher than the covariance, as higher moments introduce bias at orders of h beyond the usual order of our integrators, and hence will be 'invisible' in error analysis.

3 The NOGIN Scheme

The *NOGIN* scheme given in Algorithm 1 discretizes an underdamped Langevin dynamics SDE

$$\mathrm{d}\boldsymbol{\theta} = \boldsymbol{p}\,\mathrm{d}t, \qquad \mathrm{d}\boldsymbol{p} = \boldsymbol{F}(\boldsymbol{\theta})\,\mathrm{d}t - \boldsymbol{\mu}(\boldsymbol{\theta})\boldsymbol{p}\,\mathrm{d}t + \sqrt{2\boldsymbol{\mu}(\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{W} \tag{5}$$

to second order, where $\boldsymbol{\mu}(\boldsymbol{\theta})$ is a particular symmetric positive definite matrix and $\boldsymbol{W}$ is a standard Wiener process. We include details for including a temperature parameter (used for example in tempering strategies) or a rescaling matrix (used to precondition the dynamics such as in [13]) in the Appendix. The dynamics uniquely preserve the augmented distribution [21]

$$\pi(\boldsymbol{\theta}, \boldsymbol{p}) := \pi(\boldsymbol{\theta})\mathrm{N}(\boldsymbol{p} \,|\, \boldsymbol{0}, \boldsymbol{I}), \tag{6}$$

whose marginal in $\boldsymbol{\theta}$ yields the required target distribution. We consider integration schemes built by additively decomposing the vector field of (17) into three pieces, following the procedure in [11]. The pieces, labeled A, B, and O, are defined as

$$\mathrm{d}\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{p} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{p}\,\mathrm{d}t \\ \boldsymbol{0} \end{bmatrix}}_{\mathrm{A}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{F}(\boldsymbol{\theta})\,\mathrm{d}t \end{bmatrix}}_{\mathrm{B}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{\mu}(\boldsymbol{\theta})\boldsymbol{p}\,\mathrm{d}t + \sqrt{2\boldsymbol{\mu}(\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{W} \end{bmatrix}}_{\mathrm{O}}. \tag{7}$$

The A and B parts are referred to as the 'drift' and 'kick' pieces respectively, while the O 'fluctuation' piece corresponds to an Ornstein-Uhlenbeck (OU) linear SDE process. Note that each of the pieces, when taken individually, can be solved exactly in expectation. If we define propagation functions

$$\Phi_h^{\mathrm{A}}(\boldsymbol{\theta}, \boldsymbol{p}) := (\boldsymbol{\theta} + h\boldsymbol{p}, \boldsymbol{p}), \qquad \Phi_h^{\mathrm{B}}(\boldsymbol{\theta}, \boldsymbol{p}) := (\boldsymbol{\theta}, \boldsymbol{p} + h\boldsymbol{F}(\boldsymbol{\theta})), \tag{8}$$

$$\Phi_{\boldsymbol{\Gamma}_h, \boldsymbol{R}}^{\mathrm{O}}(\boldsymbol{\theta}, \boldsymbol{p}) := \left(\boldsymbol{\theta}, \boldsymbol{\Gamma}_h\boldsymbol{p} + \sqrt{\boldsymbol{I} - \boldsymbol{\Gamma}_h^2}\boldsymbol{R}\right) \tag{9}$$

then for suitable test function f, denoting the backward Kolmogorov operator corresponding to piece X in (7) as $\mathcal{L}_X$,

$$\left(e^{h\mathcal{L}_{\mathrm{A}}} f\right)(\boldsymbol{\theta}, \boldsymbol{p}) = \mathbf{E}\left[f\left(\Phi_h^{\mathrm{A}}(\boldsymbol{\theta}, \boldsymbol{p})\right)\right], \qquad \left(e^{h\mathcal{L}_{\mathrm{B}}} f\right)(\boldsymbol{\theta}, \boldsymbol{p}) = \mathbf{E}\left[f\left(\Phi_h^{\mathrm{B}}(\boldsymbol{\theta}, \boldsymbol{p})\right)\right], \tag{10}$$

$$\left(e^{h\mathcal{L}_{\mathrm{O}}} f\right)(\boldsymbol{\theta}, \boldsymbol{p}) = \mathbf{E}\left[f\left(\Phi_{\boldsymbol{\Gamma}_h, \boldsymbol{R}}^{\mathrm{O}}(\boldsymbol{\theta}, \boldsymbol{p})\right)\right] \quad \text{if } \boldsymbol{R} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I}) \text{ and } \boldsymbol{\Gamma}_h = \exp(-\boldsymbol{\mu}(\boldsymbol{\theta})h). \tag{11}$$

Numerical schemes for (17) can then be proposed by composing these three mappings in a prescribed sequence, using the A-B-O alphabet. For example, a step of the BAOAB scheme in [11] with stepsize $h > 0$ and using $\boldsymbol{\mu}(\boldsymbol{\theta}) = \gamma \boldsymbol{I}$ (for friction constant $\gamma > 0$) can be written as

$$(\boldsymbol{\theta}_{k+1}, \boldsymbol{p}_{k+1}) = \Phi^{\mathrm{B}}_{h/2} \circ \Phi^{\mathrm{A}}_{h/2} \circ \Phi^{\mathrm{O}}_{e^{-\gamma h \boldsymbol{I}}, \boldsymbol{R}^k} \circ \Phi^{\mathrm{A}}_{h/2} \circ \Phi^{\mathrm{B}}_{h/2}(\boldsymbol{\theta}_k, \boldsymbol{p}_k).$$

This labeling convention provides a family of methods that integrate the dynamics (17) robustly, with a particular method encoded by its string of characters. By the Jacobi identity [9, 12], a method encoded by a symmetric string will give a second order $O(h^2)$ bias in observed averages.

When using a noisy gradient in the same setting, some consideration is required as the B 'kick' step is replaced by a noisy update $\widetilde{\mathrm{B}}$ where the force term is correct only in expectation. Our strategy is to use this noise term in place of the noise usually appearing in the O step, and so we inject an additional random term $\boldsymbol{R} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ scaled by a constant $\lambda_h > 0$ to properly approximate the stochastic driving term in (17). We can write the update function for this step as

$$\Phi^{\widetilde{\mathrm{B}}}_{h, \widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{p}) = (\boldsymbol{\theta}, \boldsymbol{p} + h\boldsymbol{F}(\boldsymbol{\theta}) + \widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})) = (\boldsymbol{\theta}, \boldsymbol{p} + h\widetilde{\boldsymbol{F}}(\boldsymbol{\theta}) + \lambda_h \boldsymbol{R})$$

where $\widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})$ is a random vector with $\mathbf{E}[\widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})] = \boldsymbol{0}$ and symmetric positive definite covariance $\mathrm{Cov}(\widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})) = \boldsymbol{\Sigma}_h(\boldsymbol{\theta}) = \lambda_h^2 \boldsymbol{I} + h^2 \boldsymbol{\Sigma}(\boldsymbol{\theta})$. We may encode methods using the same convention as in (7), e.g. the $\widetilde{\mathrm{B}}\mathrm{AOA}\widetilde{\mathrm{B}}$ method, however we no longer recover second-order sampling due to the noise term in the force update. As the $\widetilde{\mathrm{B}}$ step is atomic (i.e. we cannot decouple the true force term from the noise term $\widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta})$) we are not able to directly manipulate this nuisance term and recover higher order sampling.

For methods containing a '$\widetilde{\mathrm{B}}\mathrm{O}\widetilde{\mathrm{B}}$' string, we may move the noise term using the relation

$$\begin{aligned}
\Phi^{\widetilde{\mathrm{B}}}_{h/2, \widetilde{\boldsymbol{Z}}_{h/2}} \circ \Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h, \boldsymbol{0}} \circ \Phi^{\widetilde{\mathrm{B}}}_{h/2, \widetilde{\boldsymbol{Z}}_{h/2}}(\boldsymbol{\theta}, \boldsymbol{p}) &= \left(\boldsymbol{\theta}, \tilde{\boldsymbol{\Gamma}}_h \boldsymbol{p} + (\boldsymbol{I} + \tilde{\boldsymbol{\Gamma}}_h)(\tfrac{h}{2}\boldsymbol{F}(\boldsymbol{\theta}) + \widetilde{\boldsymbol{Z}}_{h/2}(\boldsymbol{\theta}))\right), \\
&= \left(\boldsymbol{\theta}, \boldsymbol{\Gamma}_h \boldsymbol{p} + \tfrac{h}{2}(\boldsymbol{I} + \tilde{\boldsymbol{\Gamma}}_h)\boldsymbol{F}(\boldsymbol{\theta}) + \sqrt{\boldsymbol{I} - \tilde{\boldsymbol{\Gamma}}_h^2}\widetilde{\boldsymbol{Y}}(\boldsymbol{\theta})\right), \\
&= \Phi^{\mathrm{B}}_{h/2} \circ \Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h, \widetilde{\boldsymbol{Y}}} \circ \Phi^{\mathrm{B}}_{h/2}(\boldsymbol{\theta}, \boldsymbol{p}) \qquad (12)
\end{aligned}$$

where $\widetilde{\boldsymbol{Y}}(\theta) := (\sqrt{\boldsymbol{\Sigma}_{h/2}(\boldsymbol{\theta})})^{-1}\widetilde{\boldsymbol{Z}}_{h/2}(\boldsymbol{\theta})$, with $\mathbf{E}[\widetilde{\boldsymbol{Y}}] = \boldsymbol{0}$ and $\mathrm{Cov}(\widetilde{\boldsymbol{Y}}) = \boldsymbol{I}$ and for the unique nontrivial, symmetric matrix

$$\tilde{\boldsymbol{\Gamma}}_h = \left(\boldsymbol{I} - \boldsymbol{\Sigma}_{h/2}(\boldsymbol{\theta})\right)\left(\boldsymbol{I} + \boldsymbol{\Sigma}_{h/2}(\boldsymbol{\theta})\right)^{-1} = \left(\boldsymbol{I} - \lambda_{h/2}^2 \boldsymbol{I} - \frac{h^2}{4}\boldsymbol{\Sigma}(\boldsymbol{\theta})\right)\left(\boldsymbol{I} + \lambda_{h/2}^2 \boldsymbol{I} + \frac{h^2}{4}\boldsymbol{\Sigma}(\boldsymbol{\theta})\right)^{-1}. \qquad (13)$$

The resulting O update appears in step 6 in Algorithm 1, where the positive-definite inverse term does not need to be explicitly evaluated to compute the resulting product, and a low rank approximation of $\boldsymbol{\Sigma}$ could be used in the case where this step is computationally expensive. For the choice of $\lambda_h^2 = (1 - \exp(-2h\gamma))/(1 + \exp(-2h\gamma))$ the damping has the desirable property that $\tilde{\boldsymbol{\Gamma}}_h \to \exp(-h\gamma \boldsymbol{I})$ as $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \to \boldsymbol{0}$.

As we require a '$\widetilde{\mathrm{B}}\mathrm{O}\widetilde{\mathrm{B}}$' string to use (12), we propose an integrator coded $\mathrm{A}\widetilde{\mathrm{B}}\mathrm{O}\widetilde{\mathrm{B}}\mathrm{A}$, which we refer to as the *NOGIN* (NOisy Gradient INtegrator) scheme given explicitly in Algorithm 1. The proposed scheme fundamentally differs from decreasing stepsize schemes such as SGLD [24], which reduce the bias at the cost of decreased computational efficiency as T gets large [16].Many similar strategies involve either decreasing h only up to a given value (e.g. [6]) or running with a fixed stepsize and accepting the error introduced in exchange for faster convergence (see [23, 16]). The SGNHT scheme [7] corrects for the gradient noise without specific evaluation of $\boldsymbol{\Sigma}$, however this requires the covariance to be independent of $\boldsymbol{\theta}$ which is too stringent an assumption for our considerations. The CCADL scheme [22] builds upon the thermostatting framework of the SGNHT scheme [7] to reduce the bias in systems with stochastic gradients with non-constant covariance, though offers only a first-order scheme. A modified version of the SGLD scheme (mSGLD) using fixed stepsize was introduced in [23] and varies the strength of the introduced white noise term to balance against the nuisance noise from the noisy gradient. Similarly SGHMC in [6] balances the injected noise term with the noisy gradient term, but uses a formulation including momentum. While alternatives to stochastic gradient methods exist for reducing the computational cost of the gradient over the data (see e.g. [5, 10, 14]) we shall keep our focus on a noisy gradient formulation.

4 Error Analysis

4.1 Weak Convergence Analysis

Given some initial conditions $(\boldsymbol{\theta}, \boldsymbol{p})$, the expected value of a test function f at time t is

$$u_f((\boldsymbol{\theta}, \boldsymbol{p}), t) := \mathbf{E}\left[f(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) \,|\, (\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}, \boldsymbol{p})\right],$$

where the expectation is over all dynamical paths of length t. If the state evolves with respect to the underdamped Langevin dynamics (17), then u solves the backward Kolmogorov equation

$$\frac{\partial u_f}{\partial t} = (\mathcal{L}_{\mathrm{A}} + \mathcal{L}_{\mathrm{B}} + \mathcal{L}_{\mathrm{O}})\, u_f, \qquad u_f((\boldsymbol{\theta}, \boldsymbol{p}), 0) = f(\boldsymbol{\theta}, \boldsymbol{p}). \tag{14}$$

If a numerical scheme integrating (17) has update Ψ_h then define the single-step expectation v_f as

$$(\boldsymbol{\theta}_k, \boldsymbol{p}_k) = \Psi_h((\boldsymbol{\theta}_{k-1}, \boldsymbol{p}_{k-1})), \qquad v_f((\boldsymbol{\theta}, \boldsymbol{p}), h) := \mathbf{E}\left[f(\Psi_h((\boldsymbol{\theta}, \boldsymbol{p})))\right].$$

for timestep h. A scheme is weakly consistent to order p if v_f and u_f match to order $p+1$:

$$u_f((\boldsymbol{\theta}, \boldsymbol{p}), h) - v_f((\boldsymbol{\theta}, \boldsymbol{p}), h) = h^{p+1}(\mathcal{A}f)(\boldsymbol{\theta}, \boldsymbol{p}) + O(h^{p+1+s}) \tag{15}$$

for some $s > 0$, and a nonzero linear differential operator $\mathcal{A}$ depending smoothly on $\log \pi$ and its derivatives. We now give the main result of this article.

Theorem 4.1. *For sufficiently smooth π the* NOGIN *scheme is second-order weakly consistent with the dynamics* (17) *with $\boldsymbol{\mu}(\boldsymbol{\theta}) = \gamma \boldsymbol{I} + h\boldsymbol{\Sigma}(\boldsymbol{\theta})/2$.*

This can be shown directly from the definition by plugging the expectations into (15), but a cleaner and more insightful proof is given in the Appendix by composing the maps in (10)-(11) using (12). We expect, but do not prove, that a second-order weak scheme gives an $O(h^2)$ infinite-time bias (see [1]).

4.2 Exactness For Gaussian Distributions

In the case of normally distributed gradient noise we recover an exactness result for *NOGIN* in terms of introduced bias into the target distribution. This comes from a stronger result giving the perturbed invariant distribution preserved by the numerical scheme.

Lemma 4.2. *If the target distribution is of the form $\pi(\boldsymbol{\theta}, \boldsymbol{p}) = N(\boldsymbol{\theta} \,|\, \boldsymbol{\eta}, \boldsymbol{\Omega}) \times N(\boldsymbol{p} \,|\, \mathbf{0}, \boldsymbol{I})$ then for $h^2 < 4\rho(\boldsymbol{\Omega})$ and gradient noise $\widetilde{\boldsymbol{Z}}_h(\boldsymbol{\theta}) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_h(\boldsymbol{\theta}))$ with positive definite $\boldsymbol{\Sigma}_h$ and $\boldsymbol{\Omega}$, the* NOGIN *scheme uniquely preserves the perturbed distribution*

$$\pi_h(\boldsymbol{\theta}, \boldsymbol{p}) = N(\boldsymbol{\theta} \,|\, \boldsymbol{\eta}, \boldsymbol{\Omega}) \times N\left(\boldsymbol{p} \,\middle|\, \mathbf{0}, \left(\boldsymbol{I} - \frac{h^2}{4}\boldsymbol{\Omega}^{-1}\right)^{-1}\right).$$

This is demonstrated directly in the Appendix using the update maps in (8-9). Taking the marginal of π_h over $\boldsymbol{\theta}$, we can see that the correct Gaussian target distribution $\mathrm{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\eta}, \boldsymbol{\Omega})$ is recovered without discretization bias. Given ergodicity (which can be proven rigorously using the machinery from e.g. [12]), this gives the unique long-time distribution for trajectories generated using the *NOGIN* scheme.

4.3 Diminishing returns on reducing n

Intuitively we may expect that Lemma 4.2 suggests that we suffer no consequences sampling using gradient noise with a large covariance (equivalently n as small as we wish), as the bias does not increase as a result. However this proves to be untrue as the rate-of-exploration suffers as the gradient noise increases. This is easy to see from the definition of *NOGIN* in step 6 in Algorithm 1. The scheme will appropriately damp the momentum to preserve the correct distribution, with a large gradient noise requiring a large damping that quashes exploration.

We can quantify the exploration rate using the integrated autocorrelation time (IAT) denoted τ. In most cases, for a suitable test function f we have that the observed error behaves like

$$\lim_{T\to\infty} T\,\mathbf{E}\left[\left(\bar{f}_T - \langle f \rangle\right)^2\right] = \sigma_f^2 \tau_f, \qquad \bar{f}_T := \frac{1}{T}\sum_{t=1}^{T} f(\theta(t))$$

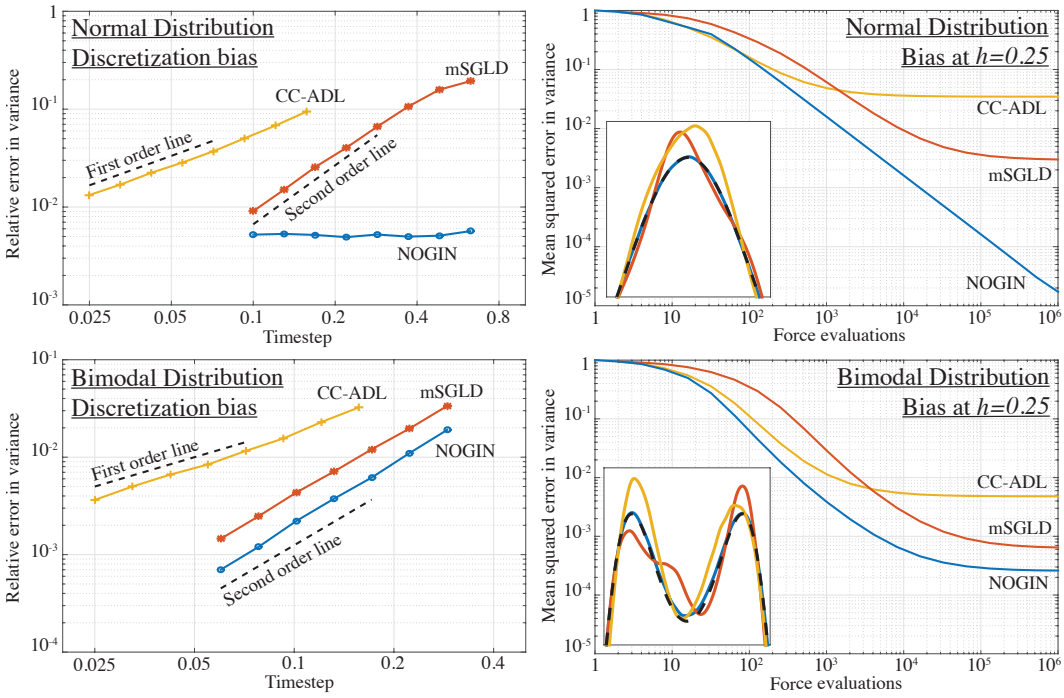

Figure 1: The left panels show the resulting long-time error in the computed variance as a function of changing the stepsize h in the algorithms. The right panels demonstrate how this error changes as the total number of iterations increases when run at $h = 0.25$. The inset on the right panels plots the final long-time distributions for each method, with the exact distribution given as a black dashed line.

where $\bar{f}_T$ is the observed average after T steps and σ_f^2 is the variance of $f(\theta)$ under π, and not related to the numerical method. The IAT τ_f can be thought of as the marginal number of timesteps required to generate an additional independent sample. In the Appendix we show that for a one-dimensional Gaussian distribution $\pi(\theta) = \mathrm{N}(\theta \,|\, 0, 1)$, using *NOGIN* with $\widetilde{F}(\theta) \sim \mathrm{N}(F(\theta), C^2)$ for constant C, the IAT for $f(\theta, p) = v_1\theta + v_2 p$ (with v_1, v_2 given in the Appendix) is

$$\tau_f = \frac{4}{h^2} + C^2 - 1 + O(C^{-2})$$

when $h^2 < 4$ and C is sufficiency large. If $C^2 = O(1/n)$ as in (4), then for sufficiently small n we do not gain an efficiency boost by reducing n further as we correspondingly increase the IAT, keeping the observed error constant for a fixed amount of computation Tn.

5 Experiments

We compare the sampling bias and efficiency between our proposed *NOGIN* method, the modified-SGLD method (mSGLD) [23] and the CC-ADL scheme [22]. Algorithms are implemented in a python code and run on a single Intel E5-2670 node.

5.1 One-Dimensional Toy Models

We consider sampling $\theta \in \mathbb{R}$ distributed according to $\pi(\theta) \propto \exp(-V(\theta))$. We run two experiments using different *potential energy* functions V:

1. $V(\theta) = \theta^2/2$, giving normally distributed θ with unit variance and zero mean,
2. $V(\theta) = (\theta^2 - 1)^2/4$, a double-well potential for bimodally distributed θ.

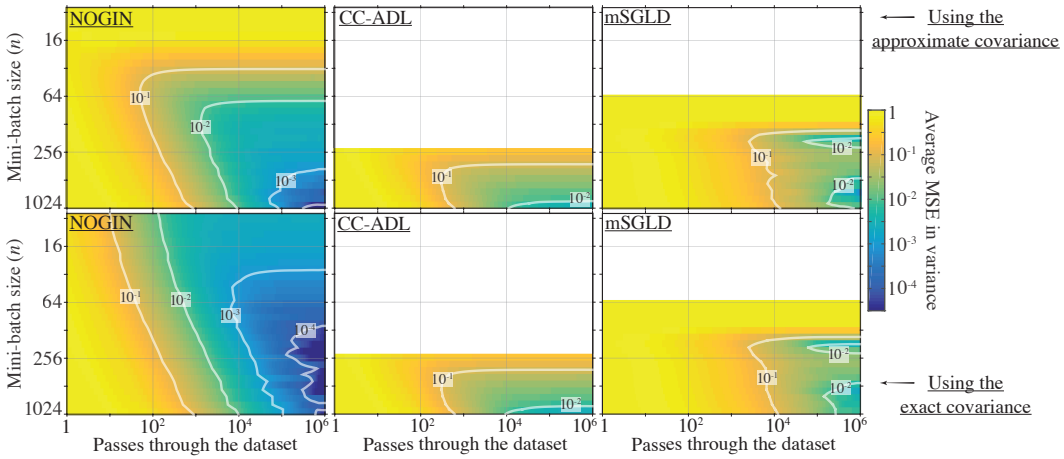

Figure 2: The error as a function of dataset passes in the Gaussian mixture model, using either the $O(N)$ exact or the $O(n)$ estimated covariance used in the schemes. The error is shown by color, with white indicating instability. It is clear that the additional bias in *NOGIN* comes from the poor estimation of the covariance, and not from the integrator itself.

In both experiments we use a noisy force $\widetilde{F}(\theta) \sim \mathrm{N}\left(F(\theta)\,,\,\sigma^2(\theta)\right)$ where $F(\theta) = -\nabla V(\theta)$ and standard deviation is chosen to be $\sigma(\theta) = 1 - \cos(1 + 5\theta)$ for both examples. In all of the algorithms we evaluate σ exactly for the required variance correction.

The computed infinite-time biases for the two experiments, estimated over several long runs where Th is constant for each algorithm, are shown in the left panels of Figure 1. The average error as a function of total force evaluations is shown in the right panels at $h = 0.25$. As expected, in long-time limit the CC-ADL gives a first-order error with respect to h while the mSGLD scheme gives a second order error. From Lemma 4.2 we expect the *NOGIN* scheme to be exact for Gaussian distributions, while Theorem 4.1 suggests that it should be second-order otherwise. This is demonstrated in Figure 1 where the error is agnostic to h in the case of quadratic V and second-order in the double-well example.

While the first column in Figure 1 shows the superiority of *NOGIN* in the infinite-time limit, the second column shows that even in the finite time regime *NOGIN* still provides more efficient convergence compared to the other methods. Qualitatively the distributions produced by *NOGIN* in the inset show an improvement with a smaller bias and with minimal breaking of symmetry.

5.2 Gaussian Mixture Model

We fit $N = 1024$ one-dimensional data points to a Gaussian mixture model using three equally weighted component Gaussian distributions. The component centers μ_i and component precisions λ_i are inferred, and thus $\pi(\boldsymbol{\theta}|y_j) = \sum_{i=1}^{3} \mathrm{N}(y_j|\mu_i, \lambda_i^{-1})/3$. The problem's fungability (symmetry under label permutation) is overcome by enforcing an ordering $\mu_i < \mu_{i+1}$ in the prior distribution. We use a normal prior on the μ_i and gamma distributed prior on λ_i, to give

$$\pi_0(\boldsymbol{\theta}) \propto \mathrm{H}(\mu_2 - \mu_1)\mathrm{H}(\mu_3 - \mu_2) \prod_{i=1}^{3} \mathrm{N}(\mu_i \,|\, 0, 100)\, \mathrm{Gamma}(\lambda_i \,|\, 2, 1),$$

where $\mathrm{H}(x)$ is the Heaviside function. We generate a synthetic dataset for the experiment, drawing points from a mixture with $\boldsymbol{\lambda} = (10, 1, 2)$ and $\boldsymbol{\mu} = (-0.8, -0.2, 1)$. We compare results for the three schemes used in Section 5.1, using the estimator (4) and mini-batch size n.

In Figure (2) we plot the convergence in the variance over all components of $\boldsymbol{\theta}$ as a function of the number of passes through the data, at a stepsize of $h = 0.02$. Each horizontal slice represents the results from one experiment using the prescribed mini-batch size n. We show the results for using two different choices of $\boldsymbol{\Sigma}$: either the estimate $\tilde{\boldsymbol{\Sigma}}$ taken from the covariance of the n mini-batch terms,

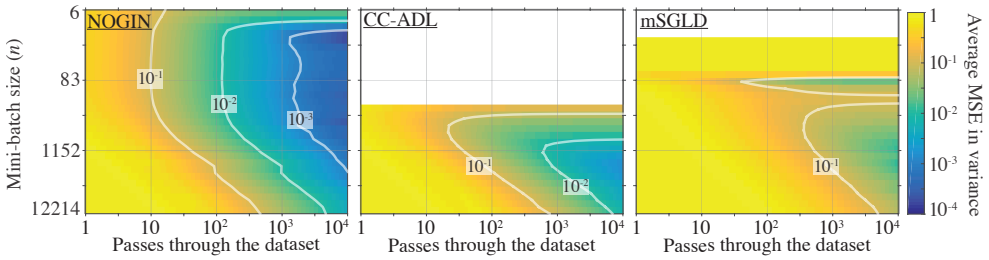

Figure 3: Results for the BLR model using the MNIST 7 and 9 dataset of 12214 data points.

or the true value of $\boldsymbol{\Sigma}$ computed from the N force terms. In the latter case we continue to use n force terms for $\widetilde{\boldsymbol{F}}$ making it a hypothetical case allowing us to pinpoint the source of error in the schemes.

It is clear in the Figure that the *NOGIN* scheme vastly outperforms the other methods, even when using the covariance estimated from only the mini-batch terms, both in stability and in the size of the introduced bias. For example, the *NOGIN* scheme requires around a factor 50 fewer passes through the data to produce a 1% error than the CC-ADL scheme. Additionally, a more accurate estimate for the covariance matrix significantly improves the accuracy of *NOGIN* , whereas the other two schemes yield no differences in the overall behavior. This suggests that the main source of bias in *NOGIN* comes from a poor estimate of $\boldsymbol{\Sigma}$, while the discretization method itself is of high-quality. While we may improve our estimate of the covariance cheaply (e.g. an online moving-average approach [2]) this will only reduce bias if the integrator is sufficiently accurate.

5.3 Bayesian Logistic Regression

We use Bayesian Logistic Regression (BLR) for classification by fitting feature parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ to the dataset $\boldsymbol{y} = \{\boldsymbol{y}_i\}$ containing N records $\boldsymbol{y_i} = \{\boldsymbol{x}_i, y_i\}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label indicating item i's classification. We then use the standard BLR likelihood

$$\pi(\boldsymbol{\theta} \,|\, \boldsymbol{y}_i) = \left(\frac{1}{1 + e^{-\boldsymbol{\theta}\cdot\boldsymbol{x}_i}}\right)^{y_i} \left(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}\cdot\boldsymbol{x}_i}}\right)^{(1-y_i)},$$

with a prior $\pi_0(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta} \,|\, \mathbf{0}, \sigma^2 \boldsymbol{I})$ and $\sigma^2 = 100$. We run a BLR experiment for classifying the number 7 or 9 in the MNIST dataset. Using principal component analysis on the dataset, we project the data $\boldsymbol{x}$ onto the top 128 produced eigenvectors. Including the constant term this gives $D = 129$, with a total dataset size of $N = 12214$. We use the estimator (4) to compute a noisy force.

The relative error in the the variance as a function of total passes through the dataset is plotted in Figure 3 for a fixed $h = 0.005$ for the three different methods. It is clear that the *NOGIN* scheme provides superior sampling efficiency compared to the other schemes, giving around a 1% error from using one hundred passes through the dataset. Additionally, the method remains stable (though with some bias) when using only $n = 6$ samples per evaluation of $\widetilde{\boldsymbol{F}}$, which the other schemes cannot deliver at the chosen stepsize.

The *NOGIN* scheme plateaus in efficiency for $n < 300$ in this problem. The vertical contour suggests that though n decreases the rate of sampling slows to match it. This matches the analysis in Section 4.3, where the damping increases to slow the dynamics and balances against the increase in efficiency. Using a more accurate estimator, reducing the size of the required damping, would prevent this effect. Although there may be other benefits for using a smaller n beyond computational efficiency, for example memory restrictions, that make the enhanced stability of *NOGIN* in this regime beneficial.

6 Discussion and Conclusion

In this paper we have presented a novel MCMC sampling algorithm for using noisy gradient information to sample from a prescribed target distribution. We demonstrate that the *NOGIN* scheme biases averages to second-order in the stepsize h in the general case, while remaining stable even when using a small mini-batch size. If the gradient noise is normally distributed, then the scheme

preserves the exact distribution when sampling from a quadratic log-posterior with sufficiently small timestep. In numerical tests the scheme provides significant improvements over other stochastic gradient schemes, both in stability and in accuracy.

We give analysis demonstrating that efficiency plateaus when the gradient noise is large, due to a correspondingly large damping required to balance the fluctuation-dissipation relation, which slows exploration. Surprisingly this slowdown is exactly countered by the reduced cost of the force and so sampling efficiency remains constant in this regime. Applying *NOGIN* with a more accurate estimator than (4) would reduce the variance of the gradient and alleviate this issue, which we leave to further work.

References

[1] A. Abdulle, G. Vilmart, and K. C. Zygalakis. High order numerical approximation of the invariant measure of ergodic SDEs. *SIAM Journal on Numerical Analysis*, 52(4):1600–1622, 2014.

[2] S. Ahn, A. Korattikara, and M. Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1771–1778. Omnipress, 2012.

[3] J. Baker, P. Fearnhead, E. B. Fox, and C. Nemeth. Control variates for stochastic gradient MCMC. *arXiv preprint arXiv:1706.05439*, 2017.

[4] S. Brooks, A. Gelman, G. Jones, and X. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2011.

[5] C. Chen, N. Ding, and L. Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2278–2286, 2015.

[6] T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

[7] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pages 3203–3211, 2014.

[8] K. A. Dubey, S. J. Reddi, S. A. Williamson, B. Poczos, A. J. Smola, and E. P. Xing. Variance reduction in stochastic gradient Langevin dynamics. In *Advances in neural information processing systems*, pages 1154–1162, 2016.

[9] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2013.

[10] A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning*, pages 181–189, 2014.

[11] B. Leimkuhler and C. Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013(1):34–56, 2012.

[12] B. Leimkuhler, C. Matthews, and G. Stoltz. The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. *IMA Journal of Numerical Analysis*, 36(1):13–79, 2015.

[13] B. Leimkuhler, C. Matthews, and J. Weare. Ensemble preconditioning for Markov chain Monte Carlo simulation. *Statistics and Computing*, 28(2):277–290, 2018.

[14] D. Maclaurin and R. P. Adams. Firefly Monte Carlo: Exact MCMC with subsets of data. 2014.

[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[16] T. Nagapetyan, A. B. Duncan, L. Hasenclever, S. J. Vollmer, L. Szpruch, and K. Zygalakis. The true cost of stochastic gradient Langevin dynamics. *arXiv preprint arXiv:1706.02692*, 2017.

[17] R. M. Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

[18] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[19] C. P. Robert. *Monte Carlo methods*. Wiley Online Library, 2004.

[20] G. O. Roberts and J. S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

[21] M. Sachs, B. Leimkuhler, and V. Danos. Langevin dynamics with variable coefficients and nonconservative forces: From stationary states to numerical methods. *Entropy*, 19(12):647, 2017.

[22] X. Shang, Z. Zhu, B. Leimkuhler, and A. J. Storkey. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems*, pages 37–45, 2015.

[23] S. J. Vollmer, K. C. Zygalakis, and Y. W. Teh. Exploration of the (non-) asymptotic bias and variance of stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 159(17), 2016.

[24] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Appendix

A Extending *NOGIN* to include mass and temperature information

For enhanced sampling schemes such as tempering it is highly desirable to include information about temperature into the distribution. Similarly we may wish to rescale the momentum by a symmetric positive definite mass matrix $\boldsymbol{M}$ to precondition the dynamics to aid exploration in certain directions. Define a new target distribution

$$\pi_\beta(\boldsymbol{\theta}, \boldsymbol{p}) \propto \pi(\boldsymbol{\theta})^\beta \mathrm{N}(\boldsymbol{p} \,|\, \mathbf{0}, \boldsymbol{M}/\beta), \tag{16}$$

for inverse temperature parameter $\beta > 0$. Choosing $\beta = 1$ and $\boldsymbol{M} = \boldsymbol{I}$ recovers the *NOGIN* scheme given in the main article. The dynamics to be integrated are

$$\mathrm{d}\boldsymbol{\theta} = \boldsymbol{p}\,\mathrm{d}t, \qquad \mathrm{d}\boldsymbol{p} = \boldsymbol{F}(\boldsymbol{\theta})\,\mathrm{d}t - \boldsymbol{\mu}(\boldsymbol{\theta})\boldsymbol{p}\,\mathrm{d}t + \sqrt{(2/\beta)\boldsymbol{\mu}(\boldsymbol{\theta})\boldsymbol{M}}\,\mathrm{d}\boldsymbol{W}, \tag{17}$$

with $\boldsymbol{\mu}(\boldsymbol{\theta}) = \gamma \boldsymbol{I} + h\boldsymbol{\Sigma}(\boldsymbol{\theta})/2$ as before. The algorithm is given in Algorithm 2.

B Proof of second-order

We assume that the target distribution π is a normalized probability distribution such that $\nabla \log \pi$ is C^∞ with bounded derivatives at all orders, and the gradient noise has bounded moments at all orders for all $\boldsymbol{\theta}$.

Algorithm 2 *NOGIN* : Noisy Gradient Integrator

Input: $\boldsymbol{\theta}_0$, $h > 0$, $\gamma > 0$, $T > 0$, $\boldsymbol{M}$, $\beta > 0$

Initialize: $\boldsymbol{p} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{M}/\beta)$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\lambda \leftarrow \sqrt{(1 - e^{-\gamma h})/(1 + e^{-\gamma h})}$

for $t = 1$ to T **do**

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + h\boldsymbol{M}^{-1}\boldsymbol{p}/2$

  $\widetilde{\boldsymbol{F}} \leftarrow \widetilde{\boldsymbol{F}}(\boldsymbol{\theta})$, $\boldsymbol{\Sigma} \leftarrow \mathrm{Cov}(\widetilde{\boldsymbol{F}}(\boldsymbol{\theta}))$, $\boldsymbol{R} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{M}/\beta)$

  $\boldsymbol{p} \leftarrow \boldsymbol{p} + h\widetilde{\boldsymbol{F}}/2 + \lambda \boldsymbol{R}$

  $\boldsymbol{p} \leftarrow \left((1 - \lambda^2)\boldsymbol{I} - \frac{h^2}{4\beta}\boldsymbol{M}\boldsymbol{\Sigma}\right)\left((1 + \lambda^2)\boldsymbol{I} + \frac{h^2}{4\beta}\boldsymbol{M}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{p}$

  $\boldsymbol{p} \leftarrow \boldsymbol{p} + h\widetilde{\boldsymbol{F}}/2 + \lambda \boldsymbol{R}$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + h\boldsymbol{M}^{-1}\boldsymbol{p}/2$

  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}$

end for

In the case of the *NOGIN* scheme we have a single-step update as the composition of maps

$$\Psi_h((\boldsymbol{\theta},\boldsymbol{p})\,|\,\widetilde{\boldsymbol{Z}}) = \Phi^{\mathrm{A}}_{h/2} \circ \Phi^{\widetilde{\mathrm{B}}}_{h/2,\widetilde{\boldsymbol{Z}}} \circ \Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h,\mathbf{0}} \circ \Phi^{\widetilde{\mathrm{B}}}_{h/2,\widetilde{\boldsymbol{Z}}} \circ \Phi^{\mathrm{A}}_{h/2}(\boldsymbol{\theta},\boldsymbol{p})$$

where $\widetilde{\boldsymbol{Z}}$ represents the injected and gradient noise and where $\tilde{\boldsymbol{\Gamma}}_h$ is chosen as in (13) . We can extricate the force updates using (12) to rewrite the step as

$$\Psi_h((\boldsymbol{\theta},\boldsymbol{p})\,|\,\widetilde{\boldsymbol{Z}}) = \Phi^{\mathrm{A}}_{h/2} \circ \Phi^{\mathrm{B}}_{h/2} \circ \Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h,\widetilde{\boldsymbol{Y}}} \circ \Phi^{\mathrm{B}}_{h/2} \circ \Phi^{\mathrm{A}}_{h/2}(\boldsymbol{\theta},\boldsymbol{p})$$

where $\widetilde{\boldsymbol{Y}}(\boldsymbol{\theta}) = \mathrm{Cov}(\widetilde{\boldsymbol{Z}}(\boldsymbol{\theta}))^{-1}\widetilde{\boldsymbol{Z}}(\boldsymbol{\theta})$ so that $\mathbf{E}[\widetilde{\boldsymbol{Y}}] = \mathbf{0}$ and $\mathrm{Cov}[\widetilde{\boldsymbol{Y}}] = \boldsymbol{I}$. Comparing to the exact update for the O piece in (9) and taking expectations we obtain

$$\begin{aligned}\mathbf{E}\left[f\left(\Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h,\widetilde{\boldsymbol{Y}}}(\boldsymbol{\theta},\boldsymbol{p})\right)\right] &= \mathbf{E}\left[f\left(\Phi^{\mathrm{O}}_{\boldsymbol{\Gamma}_h,\boldsymbol{R}}(\boldsymbol{\theta},\boldsymbol{p})\right)\right] + O(h^3)\\ &= \left(e^{h\mathcal{L}_{\mathrm{O}}}f\right)(\boldsymbol{\theta},\boldsymbol{p}) + h^3(\mathcal{A}f)(\boldsymbol{\theta},\boldsymbol{p}) + O(h^{7/2})\end{aligned}$$

for operator $\mathcal{A}$ depending upon $\log\pi$ and its derivatives, with $\boldsymbol{\Gamma}_h = \exp(-h\boldsymbol{\mu}(\boldsymbol{\theta}))$ and $\boldsymbol{R}\sim\mathrm{N}(\mathbf{0},\boldsymbol{I})$. Thus the single-step expectation is

$$v_f((\boldsymbol{\theta},\boldsymbol{p}),h) = \left(e^{h\mathcal{L}_{\mathrm{A}}/2}e^{h\mathcal{L}_{\mathrm{B}}/2}e^{h\mathcal{L}_{\mathrm{O}}}e^{h\mathcal{L}_{\mathrm{B}}/2}e^{h\mathcal{L}_{\mathrm{A}}/2}f\right)(\boldsymbol{\theta},\boldsymbol{p}) + O(h^3).$$

This can be written as

$$v_f((\boldsymbol{\theta},\boldsymbol{p}),h) = \left(e^{h(\mathcal{L}+h^2\mathcal{X})}f\right)(\boldsymbol{\theta},\boldsymbol{p}) + O(h^3) = u_f((\boldsymbol{\theta},\boldsymbol{p}),h) + O(h^3)$$

through the Jacobi identity, where the operator $\mathcal{X}$ is explicitly given through the Baker-Campbell-Hausdorff (BCH) formula [9].

C Exactness for Gaussian distributions

This can be demonstrated directly using the update maps in (8-9). Defining

$$\pi_h(\boldsymbol{\theta},\boldsymbol{p}) = \mathrm{N}(\boldsymbol{\theta}\,|\,\boldsymbol{\eta},\boldsymbol{\Omega}) \times \mathrm{N}\left(\boldsymbol{p}\,\middle|\,\mathbf{0},\left(\boldsymbol{I}-\frac{h^2}{4}\boldsymbol{\Omega}^{-1}\right)^{-1}\right)$$

we have

$$\begin{aligned}\Phi^{\mathrm{B}}_{h/2}\circ\Phi^{\mathrm{A}}_{h/2}(\boldsymbol{\theta},\boldsymbol{p}) &\sim \pi'_h(\boldsymbol{\theta},\boldsymbol{p}) \quad \text{if} \quad (\boldsymbol{\theta},\boldsymbol{p})\sim\pi_h(\boldsymbol{\theta},\boldsymbol{p})\\ \Phi^{\mathrm{A}}_{h/2}\circ\Phi^{\mathrm{B}}_{h/2}(\boldsymbol{\theta},\boldsymbol{p}) &\sim \pi_h(\boldsymbol{\theta},\boldsymbol{p}) \quad \text{if} \quad (\boldsymbol{\theta},\boldsymbol{p})\sim\pi'_h(\boldsymbol{\theta},\boldsymbol{p})\end{aligned}$$

and

$$\Phi^{\mathrm{O}}_{\tilde{\boldsymbol{\Gamma}}_h,\widetilde{\boldsymbol{Y}}}(\boldsymbol{\theta},\boldsymbol{p}) \sim \pi'_h(\boldsymbol{\theta},\boldsymbol{p}) \quad \text{if} \quad (\boldsymbol{\theta},\boldsymbol{p})\sim\pi'_h(\boldsymbol{\theta},\boldsymbol{p})$$

for the distribution

$$\pi'_h(\boldsymbol{\theta},\boldsymbol{p}) := \mathrm{N}\left(\boldsymbol{\theta}\,\middle|\,\boldsymbol{\eta},\boldsymbol{\Omega}\left(\boldsymbol{I}-\frac{h^2}{4}\boldsymbol{\Omega}^{-1}\right)^{-1}\right)\times\mathrm{N}\left(\boldsymbol{p}\,|\,\mathbf{0},\boldsymbol{I}\right).$$

Thus we have

$$\Phi^{\mathrm{A}}_{h/2}\circ\Phi^{\mathrm{B}}_{h/2}\circ\Phi^{\mathrm{O}}_{h,\widetilde{\boldsymbol{Y}}}\circ\Phi^{\mathrm{B}}_{h/2}\circ\Phi^{\mathrm{A}}_{h/2}(\boldsymbol{\theta},\boldsymbol{p}) \sim \pi_h(\boldsymbol{\theta},\boldsymbol{p}) \quad \text{if} \quad (\boldsymbol{\theta},\boldsymbol{p})\sim\pi_h(\boldsymbol{\theta},\boldsymbol{p})$$

as required.

D Computing the integrated autocorrelation time

We consider applying the *NOGIN* scheme to a one-dimensional standard normal distribution $\pi(\theta) = \mathrm{N}(\theta\,|\,0,1)$ with constant gradient noise variance $\Sigma(\theta)\equiv C^2$, for constant C^2. We may write an update of the *NOGIN* scheme as

$$\begin{bmatrix}\theta_{k+1}\\ p_{k+1}\end{bmatrix} = \boldsymbol{A}\begin{bmatrix}\theta_k\\ p_k\end{bmatrix} + \frac{1}{2}(1+\widetilde{\Gamma})hCR_k\begin{bmatrix}h/2\\ 1\end{bmatrix},$$

for $R_k \sim N(0,1)$ and

$$\boldsymbol{A} = \frac{1}{8}\begin{bmatrix} 8 - 2h^2(1+\widetilde{\Gamma}) & (4h - h^3)(1+\widetilde{\Gamma}) \\ -4(1+\widetilde{\Gamma})h & 8\widetilde{\Gamma} - 2h^2(1-\widetilde{\Gamma}) \end{bmatrix}.$$

We shall examine the rate of exploration of the state $\boldsymbol{z}_k = [\theta_k, p_k]^{\mathrm{T}}$ in a direction $\boldsymbol{v} = [v_1, v_2]^{\mathrm{T}}$ by looking at the integrated autocorrelation time of $f(\boldsymbol{z}) = \boldsymbol{z} \cdot \boldsymbol{v}$. If we assume that C is large enough so that all the eigenvalues of $\boldsymbol{A}$ are real, then choosing $\boldsymbol{v}$ to be the eigenvector of $\boldsymbol{A}^{\mathrm{T}}$ with largest associated eigenvalue λ, the autocorrelation function for f is

$$\mathrm{acf}_f(k) := \frac{\mathbf{E}[(\boldsymbol{z}_0 \cdot \boldsymbol{v})(\boldsymbol{z}_k \cdot \boldsymbol{v})]}{\mathbf{E}[(\boldsymbol{z}_0 \cdot \boldsymbol{v})(\boldsymbol{z}_0 \cdot \boldsymbol{v})]} = \frac{\mathbf{E}\left[(\boldsymbol{z}_0 \cdot \boldsymbol{v})(\boldsymbol{z}_0 \cdot (\boldsymbol{A}^{\mathrm{T}})^k \boldsymbol{v})\right]}{\mathbf{E}[(\boldsymbol{z}_0 \cdot \boldsymbol{v})(\boldsymbol{z}_0 \cdot \boldsymbol{v})]} = \lambda^k,$$

where the expectation is over all initial conditions weighted according to the known invariant distribution given. The IAT τ_f, is

$$\tau_f := 1 + 2\sum_{k=1}^{\infty} \mathrm{acf}_f(k) = 1 + \frac{2\lambda}{1-\lambda}.$$

Plugging in the value of $\tilde{\Gamma} = (1 - C^2h^2/2)/(1 + C^2h^2/2)$ with the explicit eigenvalue λ we obtain

$$\tau_f = \frac{8 + (C^2-2)h^2 + h\sqrt{h^2(C^4-4) - 16}}{(C^2+2)h^2 - h\sqrt{h^2(C^4-4) - 16}}.$$

For a fixed h and sufficiently large C, we have

$$\tau_f \approx \frac{4}{h^2} + C^2 - 1$$

and hence we expect that the autocorrelation time increases like C^2, with changing the stepsize h having a negligible impact on τ_f when C is large.