A Condition Number for Hamiltonian Monte Carlo

Ian Langmore [*] **, Michael Dikovsky, Scott Geraedts, Peter Norgaard, and Rob von Behren**

Google LLC. 1600 Amphitheatre Parkway. Mountain View, CA. 94030, e-mail: [*]ianlangmore@gmail.com

Abstract: Hamiltonian Monte Carlo is a popular sampling technique for smooth target densities. The scale lengths of the target have long been known to influence integration error and sampling efficiency. However, quantitative measures intrinsic to the target have been lacking. In this paper, we restrict attention to the multivariate Gaussian and the leapfrog integrator, and obtain a condition number corresponding to sampling efficiency. This number, based on the spectral and Schatten norms, quantifies the number of leapfrog steps needed to efficiently sample. We demonstrate its utility by using this condition number to analyze HMC preconditioning techniques. We also find the condition number of large inverse Wishart matrices, from which we derive burn-in heuristics.

Keywords and phrases: Hamiltonian Monte Carlo, Markov Chain Monte Carlo, Condition Number, Preconditioning, Random Matrices.

1. Introduction

Hamiltonian Monte Carlo (HMC) is a technique for sampling random variables $X \in \mathbb{R}^N$, possessing smooth densities $p(x)$. A a core step is the numerical integration of Hamilton's equations for time T, in ℓ discrete steps, each of size h. Tools are available to adjust h and ℓ so as to maximize sampling efficiency for particular problems [20, 10, 2]. Lacking has been a measure of difficulty, intrinsic to the density $p(x)$, rather than sub-optimal choices of h and ℓ.

It has long been recognized that disparate covariance scales in X tend to make sampling difficult [6]. This motivates techniques to "flatten" X through transformations. These transformations can be as basic as scaling components of X by their standard deviation, or as complex as application of a diffeomorphism built with polynomials or a neural network [25, 16]. Despite some success, there is limited understanding as to exactly *how much* better or worse different covariance scales are.

Our main contribution is to show that, in the multivariate Gaussian case, one particular condition number governs the number of leapfrog integration steps needed to efficiently sample in every direction. This number, κ (see (3.6)), differs from the common spectral condition number (ratio of largest to smallest singular values) since it takes into account all eigenvalues of the covariance matrix. This is needed, since all eigenvalues contribute to integration error.

Using κ we are able to analyze and develop preconditioning techniques. We find the law of κ when preconditioning with the sample covariance. The law turns out to be the condition number of the inverse Wishart ensemble. An asymptotic expression for this law is then derived, leading to a set of preconditioning heuristics. We next show that the popular component-wise standardization can be better or worse than preconditioning with a diagonal transformation trained via variational inference. Each of these in turn can be better or worse than doing nothing at all. Just as importantly, insight is given into what sort of spectra are antithetical to efficient HMC. These "bad" spectra have only a few large eigenvalues, and many small ones. We demonstrate the virtue of a low rank update preconditioner for this situation.

We limit analysis to Gaussian targets, despite the fact that sampling from them does not even require HMC. Our hope is that, by providing *explicit* formulas and *precise* analysis, these results will be helpful in more general situations. For example, the original motivation for this work was to determine if reverse or forward KL preconditioning was strictly superior (the answer is "No"). Additionally, a pleasant consequence of a condition number formulation is the connection to random matrix theory. We hope to expand on this line of thinking in future work.

Our result relating step size to integration error (theorem 3.1) is similar to the analysis of [5, 7], which derive scaling laws for (non-Gaussian) densities with repeated components. Our restriction to Gaussian densities allows for an explicit relationship between scales of the random variable X and the necessary step size/number of integration steps, and removes the repeated components requirement. A condition number is used for selecting preconditioners in [4]. Their number shows some promise for non-Gaussian systems, but does not have a precise relation to sampling efficiency (our κ does).

In section 2 we briefly review the HMC method. Section 3 goes over our main results surrounding our condition number κ. Section 4 derives an asymptotic result on κ for the inverse Wishart ensemble. Section 5 demonstrates using κ to derive and analyze different preconditioning techniques. Proofs of the main results are in section 6.

2. The Hamiltonian Monte Carlo Method

Here we quickly review the basics of HMC for purposes of establishing notation. A comprehensive introduction can be found in [24].

The Hamiltonian Monte Carlo (HMC) method was introduced in 1987 as "Hybrid Monte Carlo" for use in lattice field theory simulations [12]. Since then, it has been recognized as an efficient alternative to random walk Metropolis, well suited for higher dimensional problems. Implementations are available for a variety of languages [10].

HMC defines a way to sample from smooth densities $p(x)$ for $X \in \mathbb{R}^N$ by

augmenting state space with a momentum $\xi \in \mathbb{R}^N$, and defining the joint density

$$p(x,\xi) = \exp\left\{-H(x,\xi)\right\}, \quad \text{where} \quad H(x,\xi) := -\log p(x) + \frac{\|\xi\|^2}{2},$$

where $\|\xi\|$ is the Euclidean norm. Alternative norms may be used, although these are less popular in practice [14]. Moreover, a fixed norm generated through the inner product $\langle LL^T\xi, \xi\rangle$ is shown in [24] to be equivalent to the linear preconditioning $X \mapsto LX$ (which we *do* consider here).

In the physics setting, the *Hamiltonian* H, is total energy, whereas $-\log p(x)$, $\|\xi\|^2/2$ are potential and kinetic energies. Sampling proceeds by (a numerical approximation to) the following iteration from point (x^j, ξ^j).

1. Draw $\tilde{\xi} \sim \mathcal{N}(0, I_N)$.
2. Let $(x(t), \xi(t))$ be the time t solution to the ODE $\dot{x} = \xi$, $\dot{\xi} = \nabla \log p(x)$, with initial condition $(x^j, \tilde{\xi})$.
3. Set $(x^{j+1}, \xi^{j+1}) = (x(T), -\xi(T))$, for integration time T.

In practice, the ODE must be solved numerically over ℓ steps with step-size h. Denote this solution by Ψ^ℓ. The integration error means we can no longer just accept the move in step 3, which is replaced by a Metropolis correction:

$$(x^{j+1}, \xi^{j+1}) = \Psi^\ell, \quad \text{with probability } a(x^j, \xi^j \to \Psi^\ell),$$

and

$$(x^{j+1}, \xi^{j+1}) = (x^j, \xi^j), \quad \text{with probability } 1 - a(x^j, \xi^j \to \Psi^\ell),$$

for acceptance probability

$$a(x^j, \xi^j \to \Psi^\ell) := \min\left(1, \exp\left\{H(x^j, \xi^j) - H(\Psi^\ell)\right\}\right). \tag{2.1}$$

Since Hamilton's equations of motion preserve the Hamiltonian, if numerical integration was perfect, $H(x^j, \xi^j) = H(\Psi^\ell)$ and every step would be accepted. In practice, finite step size leads to some rejections and wasted effort.

The numerical integration is usually done with ℓ steps of the *Störmer-Verlet* or *leapfrog integrator*, each step progressing (x, ξ) to (x_h, ξ_h) via

1. Set $\xi_{h/2} = \xi + \frac{h}{2}\nabla \log p(x)$
2. Set $x_h = x + h\xi_{h/2}$
3. Set $\xi_h = \xi_{h/2} + \frac{h}{2}\nabla \log p(x_h)$

Figure 1 shows that integration errors remain small, even if $h \approx \sigma$. Just as importantly, trajectories do not diverge, but instead follow paths of a modified Hamiltonian due to the fact that the leapfrog integrator is symplectic [24, 21].

The number of leapfrog steps ℓ is often chosen to be a fixed (but highly influential) constant. To avoid unlucky (or difficult to analyze) circumstances, we use a random integration time T, then set $\ell = \lceil T/h \rceil$. This randomness is often introduced to ensure ergodicity. See section 3.2 of [24] as well as [22]. Inspection of our proofs show (see e.g. (6.7)), without this regularizing effect the spectrum *could* conspire to make the leading term vanish. Additionally, with a fixed integration length and dense enough spectrum, near *resonances* can occur, whereby samples nearly repeat the same trajectory.

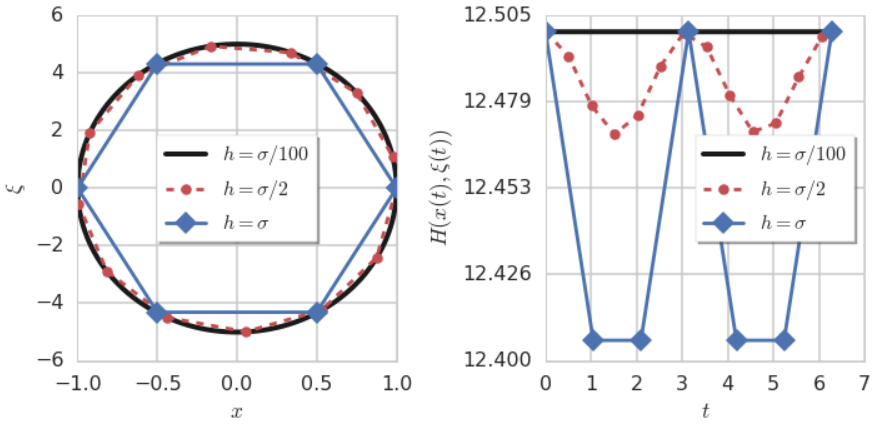

FIG 1. ***Leapfrog Integration Error. Left:*** *Integrating trajectories* $(x(t), \xi(t))$ *with different step size* h. $h = \sigma/100$ *is nearly perfect. Even with larger* h*, deviation of trajectories from the perfect line are barely perceptible.* ***Right:*** *Values of the Hamiltonian,* $H(x(t), \xi(t))$*, along the trajectories. For larger* h*, the error is greater, but does not diverge.*

3. κ and Computational Effort in HMC

Important results relating computational effort and step size has been established by previous work: Being a second order integrator, leapfrog results in error in the Hamiltonian, bounded at each step by $O(h^3)$. Thus, over $\ell = T/h$ steps, error is bounded by $O(Th^2)$ [24, 21]. In expectation the situation is better: [5, 7] show that asymptotically, the integration error is Normal with scale $O(h^2)$, with T contributing only to higher order terms.

In this section, we establish a relationship between the covariance spectrum, and the number of leapfrog steps needed to effectively sample. This is rigorously analyzed as dimension $N \to \infty$. Before submersing into the world of limits, consider fixed dimension and covariance spectrum $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_N^2 > 0$. Our results are motivated by a practitioner who adjusts the step size h and number of leapfrog steps ℓ to achieve the following

Desiderata 3.1. (i) The average acceptance probability $\mathbb{E}\left\{a(X, \xi \to \Psi^\ell)\right\} = \bar{a}$, for desired $\bar{a} \in (0, 1)$. (ii) The number of integration steps $\ell = \lceil \sigma_1 T/h \rceil$, for integration time $T \sim \pi$, where π is some probability density.

To motivate (i), consider results in [5, 7], where computational cost is shown to be (asymptotically in dimension) optimal when the average acceptance probability approaches a limit (approximately 0.68). In addition to being asymptotically optimal, tuning h to achieve desired $\bar{a}$ is often convenient [2, 10]. Condition (ii) states that each trajectory travels a distance $h\ell$ that scales with the largest scale length, σ_1. This prevents HMC from reverting to a random walk in the direction corresponding to σ_1.

Condition (ii) does imply ℓ could be quite large, but this turns out not to be

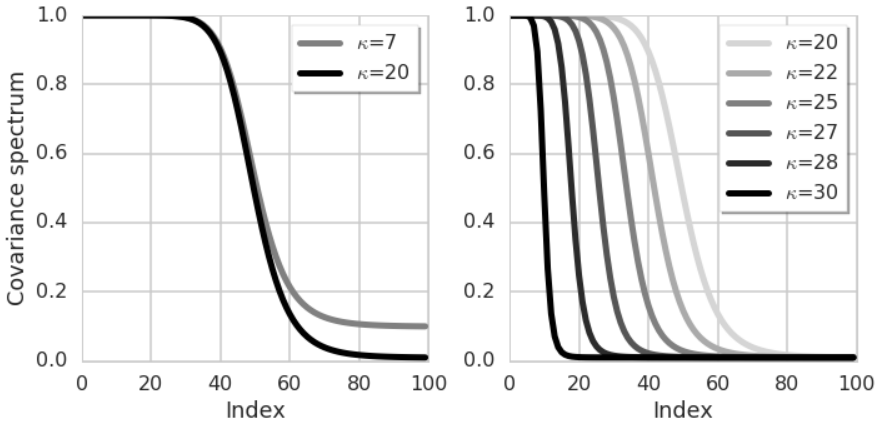

FIG 2. ***Spectra and*** κ. *Spectra were generated using* f *from* (3.4)*, and* κ *was computed.* ***Left:*** *Holding the maximal eigenvalue* (σ_1) *at one, smaller tails of the spectrum increase* κ. ***Right:*** *Holding* σ_1/σ_N *constant, the worst case is to have one large eigenvalue, and many small ones.*

an issue. The reason, is that the dominant error term depends on integration length only through an average of $\sin^2(\cdot)$, which is close to being constant (see the proof of theorem 3.1 in section 6.2). Therefore, the acceptance rate depends strongly on h but only weakly on ℓ. Thus, the user can often adjust ℓ *after* setting h, so that $h\ell \propto \sigma_1$ as desired.

Define

$$\kappa := \left(\sum_{n=1}^{N}\left(\frac{\sigma_1}{\sigma_n}\right)^4\right)^{1/4}, \qquad \nu := \left(\sum_{n=1}^{N}\left(\frac{1}{\sigma_n}\right)^4\right)^{1/4}. \tag{3.1}$$

Corollary 3.2 shows that if σ_n does not decay too fast, one may meet desiderata 3.1 (ii) using a step size

$$\bar{h} \approx \frac{1}{\nu}\, 2^{7/4}\sqrt{\Phi^{-1}\left(1-\frac{\bar{a}}{2}\right)}, \tag{3.2}$$

where Φ is the normal cumulative distribution function. The number of leapfrog steps is then proportional to $\sigma_1/h \propto \sigma_1\nu = \kappa$. Thus κ is a measure of work in a tuned HMC setup.

Since κ involves a ratio of eigenvalues, it is the shape (as opposed to overall scale) of the spectrum that determines the conditioning. As with the spectral condition number, σ_1/σ_N, κ is minimal when the spectrum is flat, i.e., $\sigma_1^2 = \cdots = \sigma_N^2$. Unlike the spectral condition number, κ is worst when there are many small eigenvalues and at least one large one (see figure 2).

3.1. Numerical estimation of κ and demonstration of main results.

A straightforward estimate of κ may be obtained by plugging the sample covariance into (3.1). We did this, and it led to a very inaccurate estimate (see figure 3 "Sample κ"). The reason being, accurate estimation of a covariance matrix requires many samples (see e.g., discussion of the Wishart case in section 5.1).

We obtained a more accurate estimate by re-writing (3.2) as

$$\kappa := \sigma_1 \nu \approx \frac{\sigma_1}{\bar{h}} 2^{7/4} \sqrt{\Phi^{-1}\left(1 - \frac{\bar{a}}{2}\right)}. \tag{3.3}$$

Thus, κ can be estimated by drawing samples with step size h, observing the acceptance probability $\hat{a}$, estimating $\hat{\sigma}_1 \approx \sigma_1$ from the sample covariance, then plugging into (3.3) (figure 3 "Inferred κ"). In experiments, where we know σ_1 ahead of time, this relation can be used to check theorem 3.1 (figure 3 "Inferred κ (known σ_1)").

To generate random spectra for these numerical tests, we use the set valued function

$$f(\mathcal{Y};\ m, M, c, \beta) := \left\{ \frac{g(y) - \min_{y\in\mathcal{Y}}\{g(y)\}}{\max_{y\in\mathcal{Y}}\{g(y)\} - \min_{y\in\mathcal{Y}}\{g(y)\}} \cdot (M - m) + m : \ y \in \mathcal{Y} \right\},$$
$$g(y) := 1/\left(1 + |y/c|^{\beta}\right). \tag{3.4}$$

We repeatedly drew $\{\sigma_n\} \sim f(\mathcal{Y};\ m, M, c, \beta)$, for random sets $\mathcal{Y}$ of size $N = 32, 64, 128, 256, 512$, each sampled from $\mathcal{U}(0, 1)$, with *minval* $m = 1$, *maxval* $M \in \{5, 20\}$, *cutoff* $c \in \{0.25, 0.75\}$, and *power* $\beta \in \{2, 6\}$. This means σ_n are values of the function $g(y)$ re-scaled to the interval $[m, M]$. See e.g., figure 2 for examples. Since the performance of HMC for Gaussian targets is invariant under isometries (see section 4 of [24]), it suffices to use these spectra in a Multivariate normal with covariance $C = \text{Diag}(\sigma_1^2, \ldots, \sigma_N^2)$. For each spectrum, we adjust step size h until the acceptance probability is close to either 0.8 or 0.95. HMC (implemented in `TensorFlow Probability` [20]) is then used to draw S samples, with the *oversampling ratio* $S/N \in \{4, 6, 8, 12, 16, 32\}$. A total of 4790 random spectra were generated.

3.2. κ is a condition number on scale matrices

Condition numbers provide worst-case bounds on solutions to linear systems. For HMC, κ also provides a worst-case result of sorts; the work needed to sample from the most difficult direction corresponding to σ_1. Our usage of κ is close to the *stiffness ratio* of a linear ODE, which is just the spectral condition number. The stiffness ratio can determine the number of time steps needed for convergence to steady state.

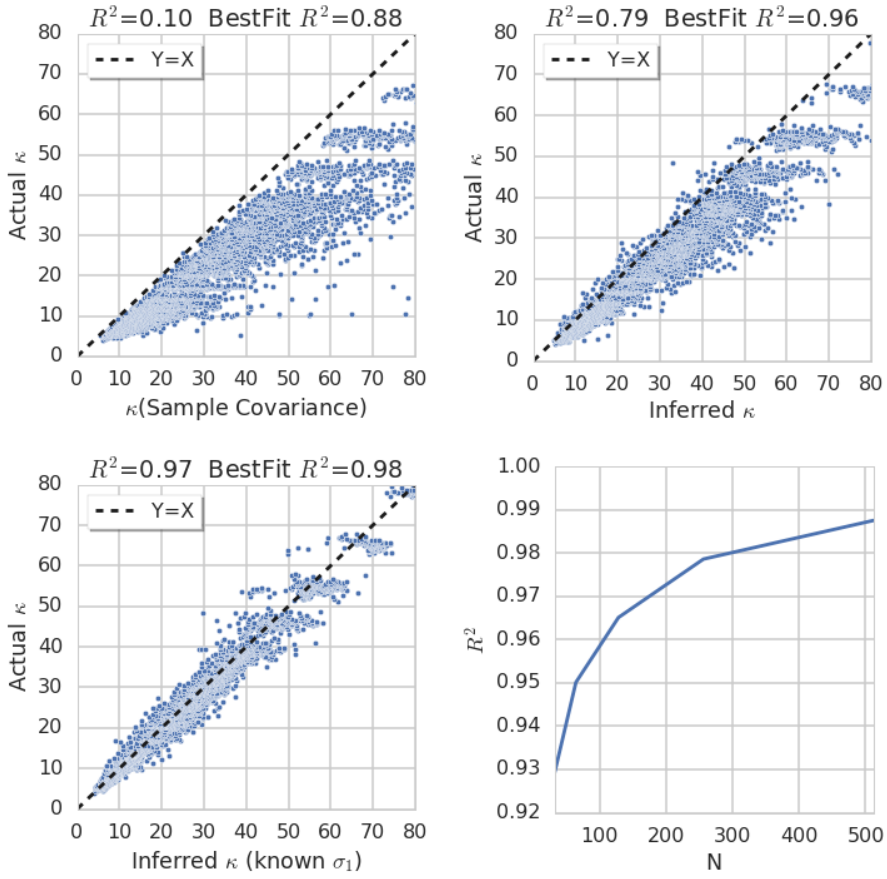

FIG 3. ***Estimation of κ and validation of main result.*** *Random spectra were generated using $f(\mathcal{Y}, m, M, \beta, c)$ from (3.4), and the relationship between estimated and actual κ is plotted. The coefficient of determination R^2 of estimated vs. actual is shown, as is R^2 of the best fit line.* ***Top Left:*** *Estimating κ directly from the sample covariance matrix resulted in over-estimation.* ***Top Right:*** *Using (3.3) with estimated $\hat{\sigma}_1$ gives a fairly accurate estimate of κ. It tends to over-estimate, due to over-estimation of σ_1.* ***Bottom Left:*** *When σ_1 is known, $R^2 \approx 1$. This validates theorem 3.1 in the finite N case.* ***Bottom Right:*** *R^2 vs. N when σ_1 is known. R^2 appears set to converge to 1, validating theorem 3.1.*

Recall the vector norm $\|\cdot\|_2$ and the induced matrix norm (the *spectral norm*):

$$\|x\|_2 := \left(\sum_{n=1}^{N} x_n^2\right)^{1/2}, \qquad \|A\|_2 := \sup_x \frac{\|Ax\|_2}{\|x\|_2}.$$

A condition number quantifies worst case sensitivity of solutions to $Ax = b$ with respect to perturbations of b. For example, consider the perturbed system $A(x + \delta x) = b + \delta b$ for nonsingular A. Elementary steps show that

$$\frac{\|\delta b\|_2}{\|b\|_2} \leq \|A\|_2 \|A^{-1}\|_2 \frac{\|\delta x\|_2}{\|x\|_2}.$$

Hence $A \mapsto \|A\|_2\|A^{-1}\|_2$ is a condition number.

A class of matrix norms of interest to us are the Schatten Norms [18]. For $r \in [1, \infty]$, the r^{th} Schatten norm of matrix A, $\|A\|_{S^r}$, is the vector r norm applied to the singular values of A. For example, with $\{\sigma_1, \ldots, \sigma_N\}$ the N singular values of $A \in \mathbb{R}^{N\times N}$,

$$\|A\|_{S^r} := \left(\sum_{n=1}^{N} \sigma_n^r\right)^{1/r}.$$

Since $\|A\|_2 = \max_n \{\sigma_n\}$, we have $\|A\|_2 \leq \|A\|_{S^r}$. Therefore

$$\frac{\|\delta b\|_2}{\|b\|_2} \leq \|A\|_2 \|A^{-1}\|_{S^4} \frac{\|\delta x\|_2}{\|x\|_2}, \tag{3.5}$$

and $A \mapsto \|A\|_2\|A^{-1}\|_{S^4}$ is a condition number w.r.t. $\|\cdot\|_2$. As compared with $\|A\|_2\|A^{-1}\|_2$, it is inferior since it provides a looser bound. The interest for us is that it is equal to κ. Indeed, suppose the covariance matrix is AA^T. Then it's eigenvalues $\sigma_1^2 \geq \cdots \geq \sigma_N^2 > 0$ are by definition the squared singular values of A, and

$$\kappa := \left(\sum_{n=1}^{N} \left(\frac{\sigma_1}{\sigma_n}\right)^4\right)^{1/4} = \|A\|_2\|A^{-1}\|_{S^4} = \sqrt{\|AA^T\|_2\|(AA^T)^{-1}\|_{S^2}}. \tag{3.6}$$

The first equality shows κ is a condition number on the *scale matrix* A. The second writes κ in terms of the *covariance matrix* AA^T, which is often more convenient.

3.3. Additional properties of κ

By definition, the set of covariance matrices for multivariate normals is the set of symmetric positive definite (SPD) matrices. Viewing κ as a function of covariance, we have

$$\kappa(C) := \sqrt{\|C\|_2\|C^{-1}\|_{S^2}}. \tag{3.7}$$

Lemma 3.1. *$\kappa : SPD \to (0, \infty)$ satisfies*

(i) $\kappa(C)^2 = \lim_{k\to\infty} \sqrt[k]{Trace\{C^k\}} \cdot \sqrt{Trace\{C^{-2}\}}$
(ii) Suppose A, B are non-singular, then $\kappa(ABB^TA^T) = \kappa(B^TA^TAB)$
(iii) Suppose U is orthogonal and C is SPD, then $\kappa(UCU^T) = \kappa(C)$

Proof. If C is SPD, its eigenvalues are its singular values $\sigma_1^2 \geq \cdots \geq \sigma_N^2 > 0$, and then Trace $\{C^k\} = \sum_{n=1}^N \sigma_n^{2k}$. Taking the limit, we have (i). To show (ii), use (i) along with the cyclic permutation property of the trace. (iii) follows from (ii) with $C = BB^T$. □

3.4. Sequences of spectra

To study convergence, we must establish a way of taking dimension to infinity. We draw inspiration from the discretization of a continuous linear operator. Here, we expect the N point discretization to have singular values $(\sigma_{N1}, \ldots, \sigma_{NN})$ that are close to singular values of the continuous operator [15]. This arises e.g., in linear inverse problems [19]. Another example is the discretization of a linear filter in signal processing. By contrast, [5, 7, 24] consider a *fixed* set of (possibly non-Gaussian, correlated) random variables, then let $p(x)$ be the law of N i.i.d. groups of these fixed variables. This simplification allows them to ignore problems associated with vanishing eigenvalues, but would be unnatural in our setting.

3.5. Acceptance probabilities for sequences of spectra

Here we use a random integration time $T_N := \sigma_{N1}T$, where $T \sim \pi$. Let

$$\hat{\pi}(\omega) := \int e^{-i\omega t}\pi(t)\,\mathrm{d}t.$$

The bound $|\hat{\pi}| \leq \hat{\pi}(0) = 1$ is trivial. In addition, we impose the regularity condition

$$|\hat{\pi}(\omega)| \leq C_\pi < 1, \quad \text{for all} \quad |\omega| \geq 2. \tag{3.8}$$

This condition is satisfied e.g., if π is a uniform density on any interval. This ensures each integral appearing in (3.9) is uniformly (in n) bounded below, and allows derivation of (3.10).

For $\alpha > 0$, $N \in \mathbb{N}$, define the step sizes

$$\begin{aligned} h_N &:= \left(\frac{1}{\alpha}\sum_{n=1}^N \frac{1}{(2\sigma_{N,n})^4}\int \sin^2\left(\frac{t}{\sigma_{Nn}}\right)\frac{\pi(t/\sigma_{N1})}{\sigma_{N1}}\,\mathrm{d}t\right)^{-1/4}, \\ \bar{h}_N &:= \left(\frac{1}{\alpha}\sum_{n=1}^N \frac{1}{(2\sigma_{N,n})^4}\frac{1}{2}\right)^{-1/4}. \end{aligned} \tag{3.9}$$

Note that $\bar{h}_N$ is, up to a constant, the inverse of ν from (3.1).

One can show (see section 6.1) that

$$(1 + C_\pi)^{-1/4} \bar{h}_N \leq h_N \leq (1 - C_\pi)^{-1/4} \bar{h}_N, \tag{3.10}$$

so the step sizes differ by at most a constant, and may sometimes be equal (corollary 3.1). Moreover, the proof of theorem 3.1 shows the chain is stable as soon as $h_N < 2\sigma_{NN}$. Then, since (see (6.2) in section 6.1) $h_N/\sigma_{NN} \to 0$, the chain is stable for large enough N.

We assume the spectra do not decay too rapidly in the sense that

$$\lim_{N\to\infty} \sigma_{N1} \left(\sum_{n=1}^{N} \frac{1}{\sigma_{Nn}^7} \right) \left(\sum_{n=1}^{N} \frac{1}{\sigma_{Nn}^4} \right)^{-3/2} = 0. \tag{3.11}$$

One can check that (3.11) holds for any polynomial decrease $\sigma_{N,n} \sim n^{-k}$, but not for exponential $\sigma_{N,n} \sim e^{-n}$. Note also that (3.11) provides *uniform* control over the spectra, e.g., it implies $h_N/\sigma_{Nn} \to 0$ uniformly in n (see (6.2)). This allows convergence despite $\{\sigma_{Nn}\}$ otherwise being unrelated at different N.

Our (standard) choice of momentum term, $\|\xi\|^2/2$, means leapfrog integration is invariant under isometry (see section 4 of [24]). Applying a rotation aligning the axis with eigenvectors of the covariance matrix, the Hamiltonian is diagonalized, and integration error may be studied one component at a time. Integration error in component n, after ℓ leapfrog steps is

$$\delta_{N,n} := H(\Psi_{N,n}^{\ell}) - H(\Psi_{N,n}^{0}).$$

The total integration error is then

$$\Delta_N := \sum_{n=1}^{N} \delta_{N,n}.$$

Theorem 3.1. *Given step size h_N from (3.9), integration time $T_N = \sigma_{N1} T$ with $T \sim \pi$ satisfying (3.8), and sequences of spectra satisfying (3.11), we have convergence in distribution for the HMC (leapfrog) integration error*

$$\Delta_N \to \mathcal{N}\left(\frac{\alpha}{2}, \alpha \right),$$

for chains in equilibrium.

Theorem 3.1 is not surprising. Indeed, since Δ_N is a sum of N independent random variables, one expects a central limit theorem to hold, provided the scaling given by h_N is correct, and many terms contribute to the sum (as opposed to it being dominated by a few terms). See e.g. the CLT for triangular arrays in [13]. The meat of the proof is establishing this scaling (see section 6.2). Once that is done, assumption (3.11) ensures many terms contribute to the sum.

The inclusion of the integral $\int \sin^2(\cdot)\, dt$ in h_N is ugly. Unfortunately, it is necessary to handle the case where the spectrum contains significant terms close to σ_{N1}^2, for which the averaging of $\sin^2(\cdot)$ does not happen. One simple case where it does is

Corollary 3.1. *Assume there exists $\delta, C > 0$ such that $|\hat{\pi}(\omega)| \leq C|\omega|^{-\delta}$. Suppose further $\sigma_{NK}/\sigma_{N1} < r_K$, where r_K is a sequence $\to 0$ as $K \to \infty$. Then as $N \to \infty$,*

$$\frac{\bar{h}_N}{h_N} \to 1.$$

Corollary 3.2 shows the free parameter α may be chosen to achieve desired acceptance rate $\bar{a} \in (0, 1)$.

Corollary 3.2. *Given the hypothesis of theorem 3.1, choose (with Φ the normal distribution function)*

$$\alpha := 4\left(\Phi^{-1}\left(1 - \frac{\bar{a}}{2}\right)\right)^2,$$

for use in h_N. We then have

$$\lim_{N\to\infty} \mathbb{E}\left\{a_N(x_j, \xi_j \to \Psi^{\ell}_{Nn})\right\} = \bar{a}.$$

Given hypothesis of corollary 3.1, the same result holds with $\bar{h}_N$ in place of h_N.

Proof. Designate the distributional limit of Δ_N by $\Delta_\infty \sim \mathcal{N}(\alpha/2, \alpha)$. As in the proof of theorem 3.6 in [5], the boundedness of $u \mapsto 1 \wedge e^u$ implies

$$\mathbb{E}\left\{a_N(x_j, \xi_j \to \Psi^{\ell}_{Nn})\right\} \to \mathbb{E}\left\{1 \wedge e^{-\Delta_\infty}\right\}.$$

This expectation can be found analytically, and is $2\Phi\left(-\sqrt{\alpha}/2\right)$. The result then follows by inverting the relation and applying the continuous mapping theorem [13]. □

4. κ for Large Inverse Wishart Matrices

The $Wishart(N, S)$ ensemble is that of $N \times N$ random matrices $(1/S)\sum_{s=1}^{S}(X^s)(X^s)^T$, where each $X^s \sim \mathcal{N}(0, I_N)$. $C \sim InverseWishart(N, S)$ if $C^{-1} \sim Wishart(N, S)$. Wishart matrices arise naturally as the S-sample covariance matrix of N-variate random normals. Inverse Wishart matrices can result from preconditioning (lemma 5.1).

If $N \to \infty$ with the *oversampling ratio* $S/N \to \omega \in (1, \infty)$, then the smallest and largest eigenvalues of a Wishart matrix approach $a := (1 - \omega^{-1/2})^2$ and $b := (1 + \omega^{-1/2})^2$ almost surely [28]. The limiting spectral density is given by the Marčenko-Pastur law[23].

$$f(x) := \begin{cases} \frac{\omega}{2\pi x}\sqrt{(b-x)(x-a)}, & a \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

See also figure 5.

The following proposition gives an asymptotic expression for κ in the inverse Wishart case. Figure 4 shows good agreement between this expression and sampled κ.

Proposition 4.1. *If* $C \sim InverseWishart(N, S)$*, and* $N \to \infty$ *with* $S/N \to \omega \in (1, \infty)$*, then*

$$\frac{\kappa(C)}{N^{1/4}} \to \frac{(1+\omega^{-1})^{1/4}}{1-\omega^{-1/2}}$$

almost surely.

Proof. Let $\lambda_1^2 \geq \cdots \geq \lambda_N^2 > 0$ be the eigenvalues of $C^{-1} \sim Wishart(N, S)$. Then,

$$\frac{\kappa(C)^4}{N} = \frac{1}{N}\sum_{n=1}^{N} \frac{\lambda_N^{-4}}{\lambda_n^{-4}} = \lambda_N^{-4}\frac{1}{N}\sum_{n=1}^{N}\lambda_n^4.$$

Now, as shown in [28] (main result) and [9] (remark 5) we have almost sure convergence,

$$\lambda_N^2 \to (1-\omega^{-1/2})^2 \quad \text{and} \quad \frac{1}{N}\sum_{n=1}^{N}\lambda_n^4 \to \mathbb{E}_f\left\{X^2\right\}.$$

Therefore, almost surely,

$$\frac{\kappa(C)^4}{N} \to \frac{1}{(1-\omega^{-1/2})^4}\mathbb{E}_f\left\{X^2\right\}.$$

The result then follows from

$$\mathbb{E}_f\left\{X^2\right\} = \int_a^b \frac{\omega}{2\pi x}\sqrt{(b-x)(x-a)}x^2\,\mathrm{d}x = 1+\omega^{-1},$$

which is a straightforward computation involving integrals of the Beta function. □

Convergence of the normalized trace to the moment is a result of remark 5 in [9], which gives *a.s.* convergence of the empirical spectral density function. Convergence of this type holds for broad class of matrices [3], [8].

5. Preconditioning HMC

The results of section 3 show a clear sampling advantage for problems where the spectrum is as close to "flat" as possible. Here we consider techniques where a diffeomorphism F transforms the random variable X into $Z := F^{-1}(X)$, with a hope that Z is easier to sample from. This is not a new idea. Linear transformations have been considered as far back as [24]. Trainable nonlinear mappings seem to have been introduced by [25], where variational inference is used to find F. It has since been developed further in [26], which considers mappings based on low-fidelity approximations to the posterior, and [16], where the preconditioner is a neural network.

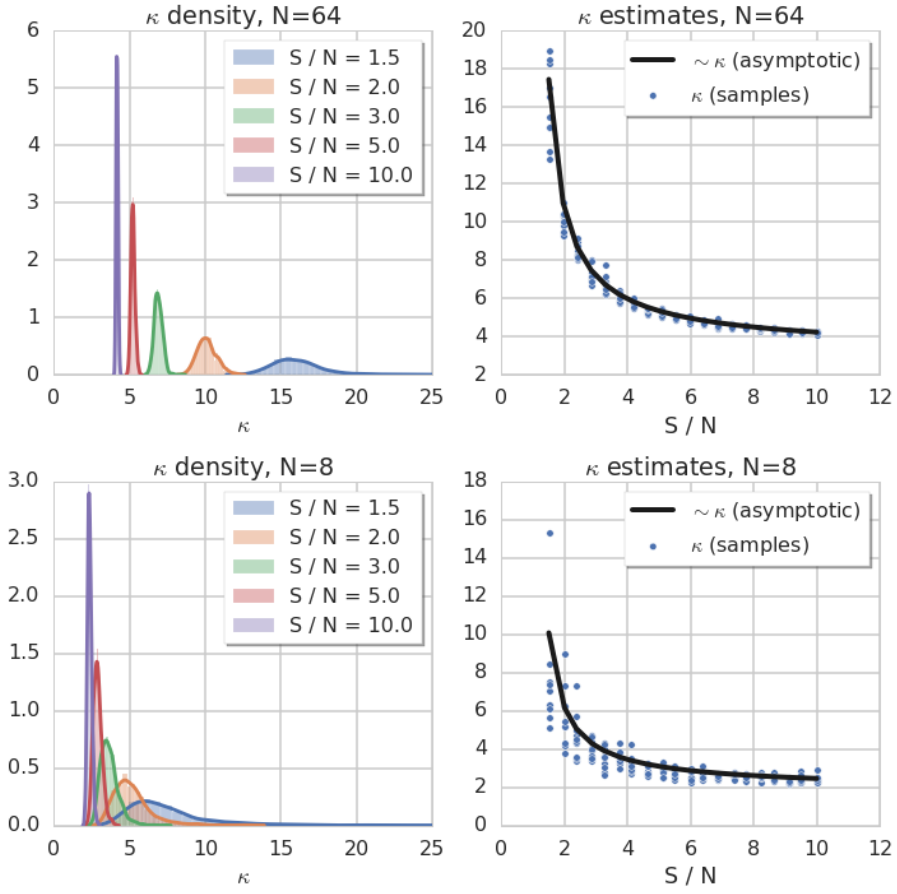

FIG 4. ***Density and asymptotic*** $\kappa(C)$, $C \sim InverseWishart(N, S)$. ***Top Left:*** *Density plots when dimension* $N = 64$. *Large* S/N *means* κ *is closer to* $N^{1/4}$, *and for* $S/N \gtrsim 3$, κ *concentrates near its mean, so point estimates will be useful.* ***Top Right:*** *Estimate vs. samples of* κ *when dimension* $N = 64$ *and* S *varies. Here "asymptotic" means the large* N *formula implied by proposition 4.1. Agreement is very good, except for small* S/N. ***Bottom Left:*** *When* $N = 8$, *the densities overlap quite a bit, so point estimates will be misleading.* ***Bottom Right:*** *When* $N = 8$, *samples of* κ *vary significantly from the asymptotic* κ.

To formalize this procedure, let us start with $X \sim p_X(x)$, and a diffeomorphism F, which transforms $X \mapsto Z = F^{-1}(X)$. Equivalently, the density p_X is transformed by the *pushforward*

$$p_Z(z) = F^{-1}_{\#} p_X(z) := p_X(F(z))|\det(DF(z))|. \tag{5.1}$$

Above, DF is the matrix of partial derivatives, $(DF)_{ij} = \partial F_j/\partial z_i$. Using HMC, we sample from the density p_Z, producing $Z^1, \ldots, Z^K$. Transforming back, $X^k := F(Z^k)$, and we have samples from p_X as desired.

In the Gaussian case, $p_X \sim \mathcal{N}(0, AA^T)$, the linear preconditioner induced by a matrix F transforms the covariance and κ as follows:

$$\begin{aligned} AA^T &\mapsto (F^{-1}A)(F^{-1}A)^T = (F^{-1})AA^T(F^{-1})^T \\ \kappa(AA^T) &\mapsto \|F^{-1}A\|_2 \|(F^{-1}A)^{-1}\|_{S^4}. \end{aligned} \tag{5.2}$$

5.1. How Many Samples are Enough?

The preconditioning techniques we will consider have an upfront cost: Either in obtaining some number of preliminary samples (which should be thrown away), or in solving an optimization problem. It's fair to ask whether the subsequent speed-up is worth it. That depends on how many final samples are needed. Here we present two cases where the ratio of samples to dimension, or *oversampling ratio*, $\omega := S/N$, is required to be 10 or more.

First, the estimation of component means and variance is done with summations of the form $(1/S)\sum_{s=1}^{S} Y^s$, resulting in relative error on each component of $O(1/\sqrt{S})$. Thus, relative error of size ε requires $S \sim O(\varepsilon^{-2})$. This is independent of dimension N, but for moderate $N \approx 100$, and reasonable $\varepsilon \approx 0.025$, we will need $S \approx 1600$, which implies $\omega := S/N \approx 16$ is required.

Second, consider the error in reconstructing the covariance spectrum from the sample covariance matrix $\hat{C} := (1/S)\sum_{s=1}^{S}(X^s)(X^s)^T$. If $X^s \sim_{\text{i.i.d.}} \mathcal{N}(0, I_N)$, we have $\hat{C} \sim Wishart(S, N)$, and for the bulk of the spectrum to be within ε of the correct values (which are all one), we must have $(1 + \omega^{-1/2}) < 1 + \varepsilon$, which implies $\omega > 4/\varepsilon^2$ (see section 4).

5.2. Preconditioning with Sample Covariance

Here we consider a choice to be made by a practitioner who has gathered $S \geq N$ burn-in samples $(X^1, \ldots, X^S)$. They could continue gathering samples until a goal of $S_f > S$ "final" are obtained. Alternatively, they could form the sample covariance $\hat{C} := (1/S)\sum_{s=1}^{S}(X^s)(X^s)^T$, precondition with its Cholesky factor $\hat{L}$, throw away the first S samples, then gather S_f final samples.

Abusing notation, we let κ_0 be the initial condition number, and $\kappa(S)$ be the condition number after preconditioning with S samples. Assuming the sampling rate is proportional to $1/\kappa$, then the total time τ to obtain S_f samples is

$$\tau \propto S\kappa_0 + S_f\kappa(S),$$

which has extremal points $\mathrm{d}\tau/\,\mathrm{d}S = 0$ at S^* whenever

$$\frac{d\kappa}{\mathrm{d}S}(S^*) = -\frac{\kappa_0}{S_f}. \tag{5.3}$$

This suggests continuing to draw burn-in samples until $\mathrm{d}\kappa/\,\mathrm{d}S \leq -\kappa_0/S_f$, at which time updates may stop and S_f samples can be drawn. The speedup is

$$\frac{S_f\kappa_0}{S\kappa_0 + S_f\kappa(S)}. \tag{5.4}$$

Assuming the burn-in samples are i.i.d., we will obtain an approximate expression for the value of S/N at which (5.3) is met. First however, we must establish

Lemma 5.1. *Suppose $(X^1, \ldots, X^S)$ are i.i.d., and HMC sampling of $X \sim \mathcal{N}(0, C)$ is preconditioned with the S-sample Cholesky factor $\hat{L}$. Then the preconditioned κ follows the law of $\kappa(C)$, for $C \sim InverseWishart(S, N)$.*

Proof. The preconditioned covariance is $\hat{L}^{-1}C\hat{L}^{-T}$. Due to lemma 3.1, $\kappa(\hat{L}^{-1}C\hat{L}^{-T}) = \kappa(L^T\hat{C}^{-1}L)$. Since

$$L^T\hat{C}^{-1}L = \left(\frac{1}{S}\sum_{s=1}^{S}(L^{-1}X^s)(L^{-1}X^s)^T\right)^{-1},$$

and $L^{-1}X^s \sim \mathcal{N}(0, I_N)$, we see that $L^T\hat{C}^{-1}L \sim InverseWishart(S, N)$, which completes the proof. □

To check (5.3) we will use lemma 5.1, and proposition 4.1. That is, we start with the approximation

$$\kappa(S) \approx g_N(S) := N^{1/4}\,\frac{\left(1 + \frac{N}{S}\right)^{1/4}}{1 - \sqrt{\frac{N}{S}}}. \tag{5.5}$$

Then, (5.3) is approximately satisfied when $\kappa'(S) \approx g'_N(S) = -\kappa_0/S_f$, which is equivalent to finding S/N such that

$$U\left(\frac{S}{N}\right) = \frac{N^{1/4}}{\kappa_0}\,\frac{S_f}{N}, \quad \text{where} \quad U(\omega) := \frac{4(\omega^{1/2} - 1)^2(\omega^2 + \omega)^{3/4}}{2\omega + \omega^{1/2} + 1}. \tag{5.6}$$

This expresses an optimality condition on the burn-in oversampling ratio S/N in terms of the final oversampling ratio S_f/N and the rescaled initial condition number $\kappa_0/N^{1/4}$.

These steps are put together in figure 5. For example, suppose $N = 50$, $\kappa_0/N^{1/4} = 10$, and uncertainty in sample covariance needs to be less than 25%. Then, the "Marcenko-Pastur Density" plot shows $S_f/N \approx 40$ is required, the "Optimal Burn-In Size S" plot shows $S/N \approx 4$ should be used, and finally, the "Speedup" plot shows our expected speedup is 3.

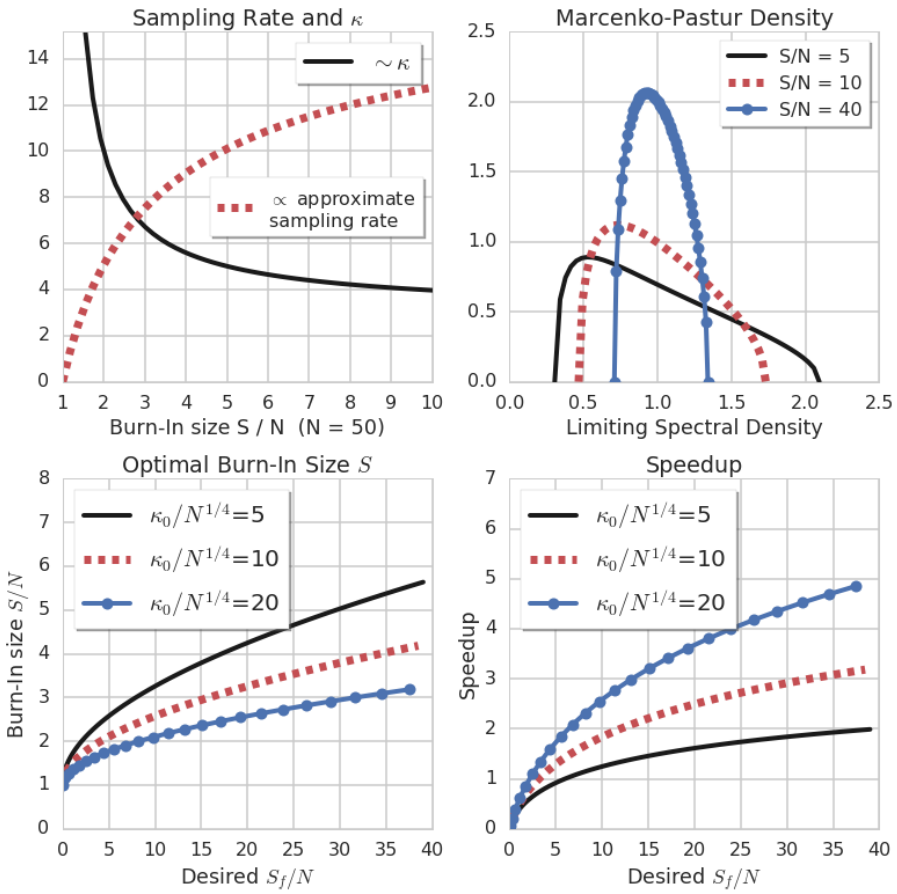

FIG 5. ***Sample Covariance Preconditioning:*** *Charts to help decide how many samples S to use for burn-in.* **Top Left:** *Approximation of $\kappa(S)$, from proposition 4.1. As always, κ is (approximately) inversely proportional to the sampling rate for HMC. We conclude larger burn-in size S leads to faster sampling rate.* **Top Right:** *Limiting spectral density for Wishart ensemble, (4.1), for three different S/N values. The spectrum has a fair bit of spread, even with 40x oversampling.* **Bottom Left:** *The optimal burn-in oversampling ratio S/N, as a function of the final desired oversampling ratio S_f/N for different values of $\kappa_0/N^{1/4}$. This is the graph $\{(U(\omega)\kappa_0/N^{1/4}, \omega) : \omega \in (0,5)\}$, with $U(\omega)$ defined in (5.6).* **Bottom Right:** *Speedup, as defined by (5.4), with $\kappa(S)$ approximated by (5.5).*

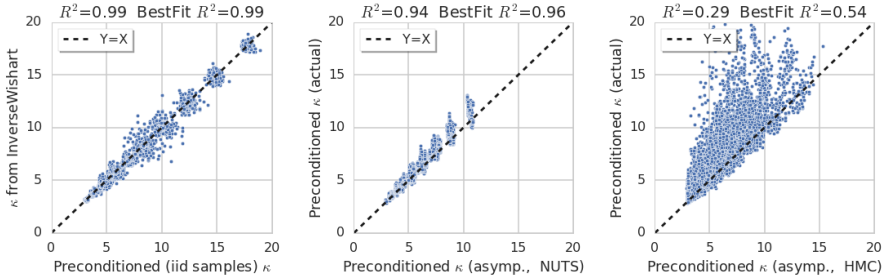

FIG 6. ***Sample Covariance Preconditioning : Theory vs. Numerics.*** *Random spectra were generated using $f(\mathcal{Y}; m, M, \beta, c)$ from (3.4), and the relationship between theoretical and actual κ is plotted. The coefficient of determination R^2 of estimated vs. actual is shown, as is R^2 of the best fit line.* **Left:** *κ after preconditioning with S i.i.d. samples in dimension N (various S and N) is close to κ obtained via samples from $InverseWishart(N, S)$, as lemma 5.1 dictates. The scatter plots of course don't match perfectly, since samples are random.* **Center:** *The function $g_N(S)$ (with S equal to the mean effective sample size) from (5.5), which approximates κ, is compared with the actual κ obtained by preconditioning with S effective samples from a NUTS chain. Agreement is good.* **Right:** *Same as Center, but using an HMC chain* without *NUTS. Agreement is not so good.*

To enact these steps in practice, one needs both an estimate of κ_0, and S i.i.d. samples. The estimate of κ_0 can be obtained using the steps in section 3.1. However, i.i.d. samples are presumably impossible to come by, else we would just use them. As a supplement, we suggest gathering S HMC samples, then using the effective sample size (ESS) in place of S in (5.6) [27]. Our experiments found ESS to be a good substitute, but *only* if the original samples were obtained using the No-U-Turn Sampler (NUTS) [17]. See figure 6. This is unfortunate, since the nice estimate of κ_0 (as per section 3.1) falls apart with NUTS, due to its non-trivial acceptance criteria. Our suggested remedy is to obtain a small number of additional non-NUTS samples, and use these to estimate κ_0. This is possible, since estimation of κ_0 only requires an estimate of the acceptance probability.

5.3. Preconditioning by way of Variational Inference

In variational inference, parameters θ are tuned to minimize a loss function involving a parameterized distribution $q(\cdot; \theta)$ and the target p. For example, the *reverse KL divergence* is

$$\mathrm{KL}[q \,||\, p] = \int \log \left[\frac{q(x;\theta)}{p(x)} \right] q(x;\theta)\, \mathrm{d}x. \tag{5.7}$$

We will always choose q to be a pushforward (see (5.1)) of the standard normal. That is, $q = F_{\#}\phi$, for $\phi \sim \mathcal{N}(0, I_N)$. It follows that

$$\mathrm{KL}[q \,||\, p] = \mathrm{KL}[F_{\#}\phi \,||\, p] = \mathrm{KL}[\phi \,||\, (F_{\#})^{-1}(p)], \tag{5.8}$$

which leads to the approximation

$$\mathrm{KL}[q\,||\,p] \approx \frac{1}{K}\sum_{k=1}^{K} \log\left[\frac{\phi(z^k)}{p(F(z^k))|DF(z)|}\right] = \frac{1}{K}\sum_{k=1}^{K}\log\left[\frac{q(F(z^k))}{p(F(z^k))}\right], \qquad z^k \sim \phi. \tag{5.9}$$

If F is smooth and $\theta \mapsto \mathrm{KL}[q\,||\,p]$ is convex, (5.7) may be minimized by stochastic gradient descent using (5.9). This is an example of *sample path optimization* [1]. Since the summands are log probabilities, (5.9) is usually found to be stable.

The "reverse" moniker is attached to (5.7) to differentiate it from *forward KL divergence*,

$$\mathrm{KL}[p\,||\,q] = \int \log\left[\frac{p(x)}{q(x;\theta)}\right] p(x)\,\mathrm{d}x. \tag{5.10}$$

Since presumably $p(x)$ is *not* easy to sample from, a stable "log space" formula analogous to (5.9) cannot be used to approximate (5.10).

Equation (5.8) shows that, if our minimization results in small KL divergence, then $(F_{\#})^{-1}p$ is close to the well-conditioned unit Gaussian, in the sense of KL divergence. Unfortunately, this does not imply $(F_{\#})^{-1}p$ has well-conditioned covariance. Indeed, if p is Gaussian, then, with λ_n^2 the eigenvalues of the preconditioned covariance WW^T, one can check that, up to additive and multiplicative constants,

$$\begin{aligned}
\mathrm{KL}[p\,||\,q] &\propto \|W\|_F^2 - \log|\det(WW^T)| = \sum_{n=1}^{N}\left(\lambda_n^2 - \log(\lambda_n^2)\right), \\
\mathrm{KL}[q\,||\,p] &\propto \|W^{-1}\|_F^2 - \log|\det((WW^T)^{-1})| = \sum_{n=1}^{N}\left(\lambda_n^{-2} - \log(\lambda_n^{-2})\right).
\end{aligned} \tag{5.11}$$

Clearly, minimizing forward or reverse KL is different than minimizing κ.

5.3.1. Diagonal preconditioning

Here we consider preconditioning a Gaussian $\mathcal{N}(0, C)$ with a diagonal matrix D. The covariance and κ transform as in (5.2).

Minimizing reverse KL over the set of diagonal matrices (see (5.11)) gives us

$$D_{ii}^2 = 1/(C^{-1})_{ii}, \tag{5.12}$$

while minimizing forward KL gives us

$$D_{ii}^2 = C_{ii}, \tag{5.13}$$

which is just a diagonal matrix with the component-wise variances. In this case, preconditioning is equivalent to re-scaling the axis so that X has unit standard deviation.

As diagonal preconditioners, both forward and reverse KL exhibit a scale invariance, the proof of which follows directly from (5.12) and (5.13).

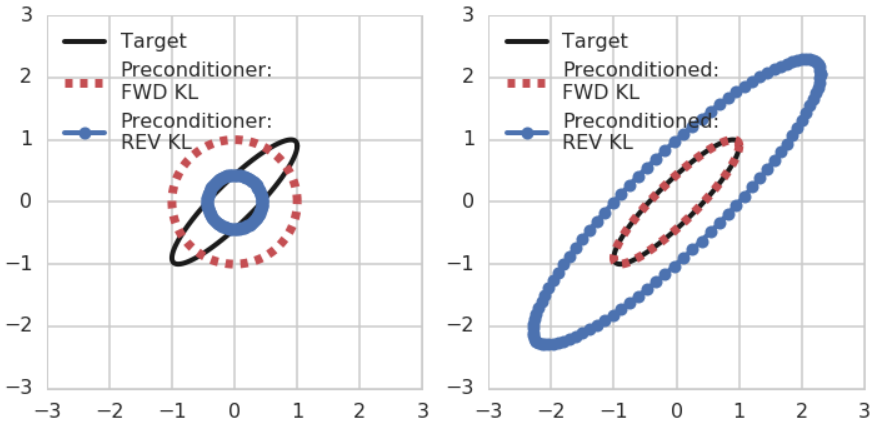

FIG 7. ***Forward and reverse KL, preconditioner and preconditioned.*** *The target* $p(x) \sim \mathcal{N}(0, C_n)$*, with* C_n *from* (5.14) *and* ($\rho = 0.9$)*. The variational distribution* $q(z) \sim \mathcal{N}(0, D_n^2)$*, where* D_n *is diagonal. Both forward and reverse KL were minimized to find* D_n*.* **Left:** *The one standard deviation iso-line of* $q(z)$ *is plotted. Both are circular due to symmetry, but forward KL chooses a larger sphere (of radius 1).* **Right:** *Iso-lines after preconditioning by* D_n *(which yields* $\sim \mathcal{N}(0, D_n^{-1} C_n D_n^{-1})$*). Reverse KL results in a much larger preconditioned spectrum.*

Lemma 5.2. *Suppose forward/reverse KL preconditioning of* $X \sim \mathcal{N}(0, C)$ *results in preconditioned covariance* Ω*. Then, for any positive diagonal matrix* $\tilde{D}$*, forward/reverse KL diagonal preconditioning of* $X \sim \mathcal{N}(0, \tilde{D}C\tilde{D})$ *also leads to preconditioned covariance* Ω*.*

Consider the following choices:

(i) Do not precondition, and sample directly from the target $p(x)$.
(ii) Precondition with D^{-1}, where D is obtained by minimizing $\mathrm{KL}[q \,||\, p]$, (i.e., reverse KL), for $q(x) \sim \mathcal{N}(\mu, D^2)$.
(iii) Precondition with D^{-1}, where D is obtained by minimizing $\mathrm{KL}[p \,||\, q]$, (i.e., forward KL), for $q(x) \sim \mathcal{N}(\mu, D^2)$.

In the proceeding sections, we will show realistic scenarios where each method is better than the other two. Before proceeding, we point out some practical considerations. Since forward KL is often unstable and cannot be minimized directly, (iii) is done by estimation of the component-wise standard deviation. If this must be done by sampling, then it somewhat defeats the purpose of preconditioning. Regarding (ii), setting up a variational problem is not too hard once the target $p(x)$ is built, and software packages exist to make this easier [11]. This does however incur a one time development cost that may be too great for the problem at hand.

5.3.2. Diagonal preconditioning of correlated diagonal blocks

A simple covariance comprised of 2x2 blocks provides a demonstration of cases where forward KL preconditions better than reverse KL, which, depending on the blocks, performs better or worse than doing nothing. We also see that neither forward nor reverse KL is optimal.

For $\rho_n \in (0,1)$, let covariance be given by the block-diagonal matrix

$$C = C_1 \oplus C_2 \oplus \cdots \oplus C_N, \quad C_n := \begin{pmatrix} 1 & \rho_n \\ \rho_n & 1 \end{pmatrix}, \tag{5.14}$$

which has eigenvalues $\{1 \pm \rho_n : n = 1, \ldots, N\}$. By symmetry, the optimal preconditioner D, for forward or reverse KL, will be partitioned as

$$D = D_1 \oplus \cdots \oplus D_N, \quad D_n := \begin{pmatrix} d_n & 0 \\ 0 & d_n \end{pmatrix}.$$

This leads to the preconditioned covariance

$$D^{-1}CD^{-1} = d_1^{-2}C_1 \oplus \cdots \oplus d_N^{-2}C_N,$$

with spectrum

$$\Lambda := \left\{ \frac{1+\rho_1}{d_1^2}, \frac{1-\rho_1}{d_1^2}, \cdots, \frac{1+\rho_N}{d_N^2}, \frac{1-\rho_N}{d_N^2}, \right\}.$$

Thus, each pair of eigenvalues, $\{1 \pm \rho_n\}$ is moved up and down together by the preconditioner. As seen below and in in figure 8, the optimal preconditioner moves the larger of the two from every block to the same level. Reverse KL to some extent does the opposite, moving the smaller of each pair to a similar level. This is expected, since as illustrated in figure 7, the scale of the reverse KL variational solution is mostly determined by the smallest scale of the target.

In the forward KL, reverse KL, and the optimal choices of d_n that follow, $(1+\rho_1)/d_1^2 \geq (1 \pm \rho_n)/d_n^2$, for $n = 2, \ldots, N$. Therefore, all three choices will have

$$\kappa(\Lambda)^4 = \sum_{n=1}^{N} \left(\frac{d_n^2}{d_1^2}\right)^2 \left[\left(\frac{1+\rho_1}{1+\rho_n}\right)^2 + \left(\frac{1+\rho_1}{1-\rho_n}\right)^2 \right]. \tag{5.15}$$

Referring to (5.12), (5.13), the minimizing d_n^2 for forward KL will be identically 1, and for reverse KL will be $1-\rho_n^2$. Thus

$$\kappa(\Lambda_{rev})^4 = \sum_{n=1}^{N} \left(\frac{1-\rho_n^2}{1-\rho_1^2}\right)^2 \left[\left(\frac{1+\rho_1}{1+\rho_n}\right)^2 + \left(\frac{1+\rho_1}{1-\rho_n}\right)^2 \right]$$

$$\kappa(\Lambda_{fwd})^4 = \sum_{n=1}^{N} \left[\left(\frac{1+\rho_1}{1+\rho_n}\right)^2 + \left(\frac{1+\rho_1}{1-\rho_n}\right)^2 \right],$$

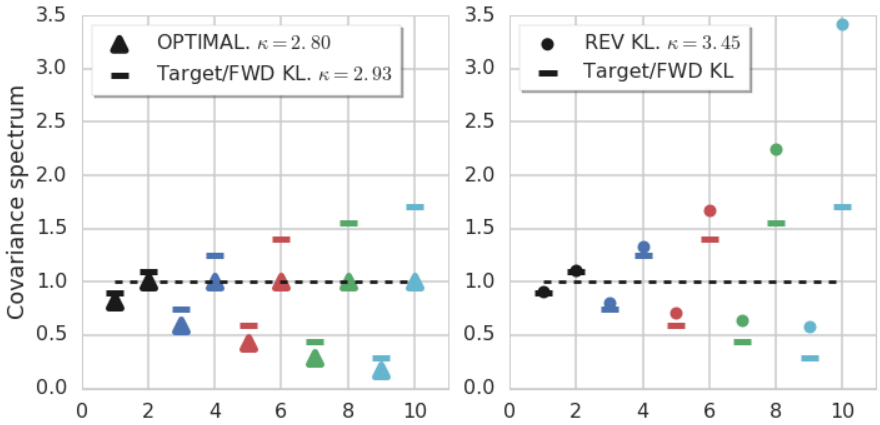

FIG 8. ***Preconditioned spectra in the case of 5 correlated 2x2 blocks.*** *Preconditioner choices (OPTIMAL vs. REV KL.) are plotted against the target, which is the same as FWD KL preconditioning. The spectrum comes in pairs of eigenvalues, which are scaled together by the preconditioners.* **Left:** *The optimal preconditioner pushes the larger member of each pair to 1.* **Right:** *Reverse KL (almost) pushes the smaller of each pair to 1.*

and therefore $\kappa(\Lambda_{fwd}) \leq \kappa(\Lambda_{rev})$.

Minimizing κ over all d_n we find $d_n^2 \propto 1 + \rho_n$, so that

$$\kappa(\Lambda_{OPT})^4 = \sum_{n=1}^{N} \left[1 + \left(\frac{1+\rho_n}{1-\rho_n} \right)^2 \right].$$

Since reverse KL is a practical method, it is disappointing to see that doing nothing (forward KL had $D = I$) performs better. In practice however, the situation is often closer to

$$C_n := \gamma_n^2 \begin{pmatrix} 1 & \rho_n \\ \rho_n & 1 \end{pmatrix},$$

for $\gamma_1^2 \geq \cdots \geq \gamma_N^2 > 0$. The results for forward and reverse KL will be the same as if $\gamma_n \equiv 1$ due to the scale invariance lemma 5.2, but "doing nothing" yields a baseline of (with $\beta := \max_n \left\{ \gamma_n^2 (1+\rho_n) \right\}$)

$$\kappa^4 := \left[\frac{\beta^2}{\gamma_n^4 (1+\rho_n)^2} + \frac{\beta^2}{\gamma_n^4 (1-\rho_n)^2} \right] \geq \sum_{n=1}^{N} \frac{\gamma_1^4}{\gamma_n^4} \left[\left(\frac{1+\rho_1}{1+\rho_n} \right)^2 + \left(\frac{1+\rho_1}{1-\rho_n} \right)^2 \right].$$

So if $\gamma_1 \gg \gamma_n$ is large enough, preconditioning with reverse KL *does* improve upon doing nothing. Forward KL would still be superior.

5.3.3. Diagonal preconditioning of random matrices

Here we compare preconditioner options (Forward or Reverse KL, or "Do Nothing") applied to 100x100 random matrices of different types. Each option is best

some fraction of the time.

The *Wishart* random matrices are constructed by (i) letting $A \in \mathbb{R}^{100\times 200}$ be composed of i.i.d. unit normal entries, then (ii) setting $C := AA^T$. The *inverse Wishart* matrices are constructed by inverting a Wishart matrix. The *rotated scale* matrices are constructed by (i) making the scale matrix $\Lambda := \text{Diag}(\sigma_1^2, \ldots, \sigma_{100}^2)$, where $\{\sigma_1, \ldots, \sigma_{100}\} = f(\{1, \ldots, 100\}, m = 1, M = 5, \beta = 4, c = \texttt{cutoff})$ from (3.4), then (ii) rotating with a random orthogonal matrix generated with `scipy.stats.ortho_group`, i.e. $C := U\Lambda U^T$ [29]. Three varieties of rotated scale matrix were constructed by choosing $\texttt{cutoff} = 0.05, 0.1, 0.2$, which means the scales have approximately 5%, 10%, or 20% of the values near the maximum (of 5), and the remainder near the minimum (of 1). For each of the 5 matrix types, 1000 different random matrices were generated, and the fraction of the time each preconditioner type leads to lower κ is shown in table 1.

The Wishart/inverse Wishart matrices have a fairly regular structure, and forward KL is the "winner" more and more often as N increases. The results for rotated scale matrices varied considerably as the dimension or the parameters (M, β) were changed. We warn the reader *not* to draw broad conclusions or trends from table 1.

	Wishart	InvWishart	RS (5%)	RS (10%)	RS (20%)
Do Nothing	3%	0%	0%	20%	100%
Fwd KL	91%	100%	88%	1%	0%
Rev KL	6%	0%	12%	78%	0%

TABLE 1

For Wishart, inverse Wishart, and rotated scale (RS) random 100x100 matrices, the percentage of the time each preconditioning method had the lowest κ is tabulated. The magnitude of the differences was small, about 10% at most.

5.3.4. Diagonal plus low-rank preconditioning

As demonstrated in figure 2, having a few large eigenvalues and many small ones is especially bad for κ. To mitigate these situations, we consider a low-rank update to a diagonal preconditioner. Specifically, we choose variational distribution $q \sim \mathcal{N}(0, FF^T)$, where $F = D + UU^T$, $D \in \mathbb{R}^{N\times N}$ is diagonal, and $U \in \mathbb{R}^{N\times K}$. Both D and U were trained to minimize $\text{KL}[q \,||\, p]$, where $p \sim \mathcal{N}(0, LL^T)$ and L is circulant. This provides some correlation that cannot be matched with a diagonal preconditioner. The spectrum of L was chosen using (3.4) so that it was a low pass filter with some cutoff. As expected, when the rank of U was larger than the cutoff, preconditioning worked. When the rank of U was less than the cutoff, large eigenvalues remained and κ was barely reduced by preconditioning. See figure 9.

A word of caution: We also found that when the circulant matrix L had very small eigenvalues, the corresponding extreme correlations in $X \sim p(x)$ were too much for the optimization procedure to handle, and instabilities arose.

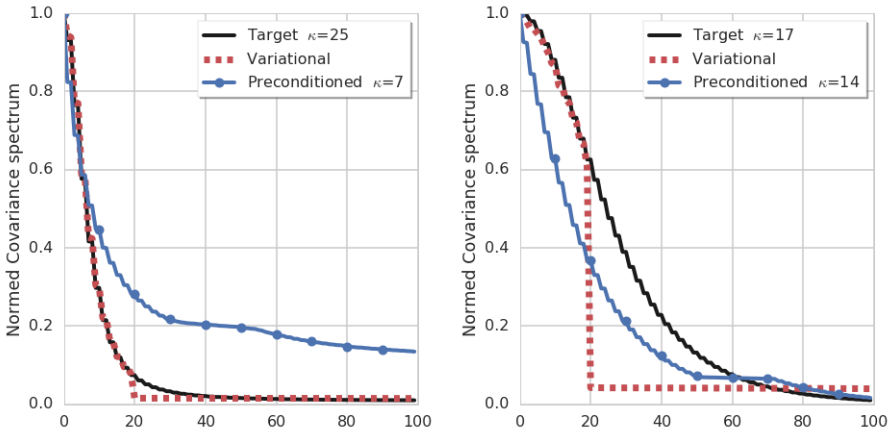

FIG 9. ***Preconditioning with a low-rank update.*** *Normalized covariance spectra of the Target $p \sim \mathcal{N}(0, LL^T)$, variational model $q \sim \mathcal{N}(0, D + UU^T)$, and the preconditioned target.* **Left:** *LL^T had 10 large eigenvalues, and U was a rank 20 update. The trained preconditioner $D + UU^T$ had about 10 large eigenvalues, which were used to reduce the largest eigenvalues in the preconditioned matrix.* **Right:** *Here LL^T had about 40 large eigenvalues, and the rank 20 update could not reduce the size of them all.*

6. Proof of Convergence Results

In this section we prove theorem 3.1 and corollary 3.1.

Denote by π_N the density of $T_N := \sigma_{N1} T$ (recall $T \sim \pi$). That is,

$$\pi_N(t) := \pi\left(\frac{t}{\sigma_{N1}}\right) \cdot \frac{1}{\sigma_{N1}}.$$

6.1. Two relations involving step size and scales

First, we show (3.10). To that end, the distinction between h_N and $\bar{h}_N$ is the replacement of $\int \sin^2(\cdot)\, \mathrm{d}t$ by its average, $1/2$. Let

$$R_{Nn} := \frac{1}{2} - \int \sin^2\left(\frac{t}{\sigma_{Nn}}\right) \pi_N(t)\, \mathrm{d}t = \frac{1}{4}\left[\hat{\pi}\left(2\frac{\sigma_{N1}}{\sigma_{Nn}}\right) + \hat{\pi}\left(-2\frac{\sigma_{N1}}{\sigma_{Nn}}\right)\right],$$

which, due to (3.8), satisfies

$$|R_{Nn}| \leq \frac{C_\pi}{2} < \frac{1}{2}. \tag{6.1}$$

This means

$$\left|\bar{h}_N^{-4} - h_N^{-4}\right| \leq C_\pi \bar{h}_N^{-4},$$

and therefore

$$(1+C_\pi)^{-1/4}\,\bar{h}_N \le h_N \le (1-C_\pi)^{-1/4}\bar{h}_N,$$

which is exactly (3.10).

Second, we show the uniform limit

$$\lim_{N\to\infty}\frac{h_N}{\sigma_{Nn}} \le \lim_{N\to\infty}\frac{h_N}{\sigma_{NN}} = 0. \tag{6.2}$$

To that end, since $\sigma_{NN} \le \sigma_{Nn}$, we have

$$\left(\frac{h_N}{\sigma_{Nn}}\right)^6 \le \left(\frac{h_N}{\sigma_{NN}}\right)^6 \le \frac{\sigma_{N1}}{\sigma_{NN}}\frac{h_N^6}{\sigma_{NN}^6} = \sigma_{N1}h_N^6\frac{1}{\sigma_{NN}^7} \le \sigma_{N1}h_N^6\sum_{n=1}^N\frac{1}{\sigma_{Nn}^7}.$$

Due to (3.10), we may replace h_N by $\bar{h}_N$, and suffer only a constant depending on α and C_π. From the definition of $\bar{h}_N$, we may replace $\bar{h}_N$ with $\left(\sum_{n=1}^N \sigma_{Nn}^{-4}\right)^{-1/4}$ and suffer only a constant depending on α. We therefore have (with C' depending only on α and C_π)

$$\sigma_{N1}h_N^6\sum_{n=1}^N\frac{1}{\sigma_{Nn}^7} \le C'\sigma_{N1}\left(\sum_{n=1}^N\frac{1}{\sigma_{Nn}^7}\right)\left(\sum_{n=1}^N\frac{1}{\sigma_{Nn}^4}\right)^{-3/2},$$

which tends to zero due to our assumption (3.11), and so too then does h_N/σ_{Nn}.

6.2. The normal limit

Proof of theorem 3.1. Since, in equilibrium, the performance of HMC with momentum term $\|\xi\|^2/2$ is invariant under isometries (see section 4 of [24]), we may assume without loss of generality that our distribution is a centered diagonal Gaussian with covariance $\mathrm{Diag}(\sigma_{N1}^2,\ldots,\sigma_{NN}^2)$. Leapfrog integration will act independently on each component. Moreover, in equilibrium, position and momentum samples from each component are independent.

We first consider the case of one component with variance σ^2, and hide the dependence on N. Each leapfrog step is an iteration of the matrix

$$U_h := \begin{pmatrix} 1-\frac{h^2}{2\sigma^2} & h \\ -\left(\frac{h}{\sigma^2}-\frac{h^3}{4\sigma^4}\right) & 1-\frac{h^2}{2\sigma^2}\end{pmatrix},$$

which has eigenvalues and eigenvectors

$$\lambda_\pm := 1-\frac{h^2}{2\sigma^2} \pm i\frac{h}{\sigma}\sqrt{1-\frac{h^2}{4\sigma^2}}, \qquad v_\pm := \left(1, \pm\frac{i}{\sigma}\sqrt{1-\frac{h^2}{4}}\right).$$

If $h/(2\sigma) < 1$, the eigenvalues have modulus 1 and the iteration is stable. Then, by diagonalizing, one can show

$$U_h^\ell = \begin{pmatrix} \cos(\ell\theta) & \gamma^{-1}\sin(\ell\theta) \\ -\gamma\sin(\ell\theta) & \cos(\ell\theta) \end{pmatrix},$$

where

$$\gamma := \sqrt{\frac{1}{\sigma^2} - \frac{h^2}{4\sigma^2}}, \qquad \theta := \cos^{-1}\left(1 - \frac{h^2}{2\sigma^2}\right).$$

To compute the Hamiltonian after ℓ steps, we apply U_h^ℓ to the starting point (x_0, ξ_0), then plug into $H(x,\xi) = x^2/(2\sigma^2) + \xi^2/2$ to get

$$\begin{aligned} H(U_h^\ell(x_0,\xi_0)) = \cos^2(\ell\theta)\left(\frac{x_0^2}{2\sigma^2} + \frac{\xi_0^2}{2}\right) + \sin^2(\ell\theta)\left(\gamma^2\sigma^2\frac{x_0^2}{2\sigma^2} + \frac{1}{\gamma^2\sigma^2}\frac{\xi_0^2}{2}\right) \\ + \cos(\ell\theta)\sin(\ell\theta)\left(\frac{1}{\gamma\sigma} - \gamma\sigma\right)\frac{x_0\xi_0}{\sigma}. \end{aligned} \tag{6.3}$$

Define

$$\chi := \left(\frac{h}{2\sigma}\right)^4 \cdot \frac{1}{1 - \left(\frac{h}{2\sigma}\right)^2},$$

then using the relations

$$\gamma^2\sigma^2 + \frac{1}{\gamma^2\sigma^2} = 2 + \chi, \qquad \gamma\sigma - \frac{1}{\gamma\sigma} = \sqrt{\chi},$$

and the fact that the initial Hamiltonian is $x_0^2/(2\sigma^2) + \xi_0^2/2$, we find

$$\delta^\ell := \frac{\sin^2(\ell\theta)}{2}\left(\frac{h}{2\sigma}\right)^2\left(\xi_0^2 - \frac{x_0^2}{\sigma^2}\right) + \sin^2(\ell\theta)\chi\frac{\xi_0^2}{2} + \cos(\ell\theta)\sin(\ell\theta)\sqrt{\chi}\frac{x_0\xi_0}{\sigma}. \tag{6.4}$$

This has the bound

$$|\delta^\ell| \le \frac{h^2}{8\sigma^2}\left(\xi_0^2 - \frac{x_0^2}{\sigma^2}\right) + \chi\frac{\xi_0^2}{2} + \sqrt{\chi}\frac{|x_0\xi_0|}{\sigma}. \tag{6.5}$$

To compute moments, use the fact that, in equilibrium, $x_0 \sim \mathcal{N}(0,\sigma^2)$, and $\xi_0 \sim \mathcal{N}(0,1)$ are independent.

$$\begin{aligned} \mathbb{E}\left\{\delta^\ell\right\} &= \frac{\sin^2(\ell\theta)}{2}\left(\frac{h}{2\sigma}\right)^4 + R_1(\delta^\ell)\left(\frac{h}{2\sigma}\right)^6, \\ \mathrm{Var}\left\{\delta^\ell\right\} &= 2\mathbb{E}\left\{\delta^\ell\right\} + R_2(\delta^\ell)\left(\frac{h}{2\sigma}\right)^6, \end{aligned}$$

where there exists a constant $C < \infty$, uniform in (σ, ℓ, θ) (so long as $h < 2\sigma$), such that $|R_j| \leq C$.

Re-introducing dependence on N, n, and setting $\ell = T/h_N$ for random integration time $T \sim \pi_N$ (we assume ℓ is an integer, if not minor adjustments are needed), we have

$$\begin{aligned}\mathbb{E}\left\{\Delta_N\right\} = \frac{1}{2}\sum_{n=1}^{N}\left(\frac{h_N}{2\sigma_{Nn}}\right)^4 \int \sin^2\left(\frac{t}{h_N}\cos^{-1}\left(1-\frac{h_N^2}{2\sigma_{N,n}^2}\right)\right)\pi_N(t)\,\mathrm{d}t \\ + \frac{1}{2}\sum_{n=1}^{N}\left(\frac{h_N}{2\sigma_{Nn}}\right)^6 \int R_1(\delta_{N,n}^{\ell})\pi_N(t)\,\mathrm{d}t.\end{aligned} \tag{6.6}$$

The second term is bounded in absolute value by a constant times

$$\left(\frac{h_N}{\sigma_{NN}}\right)^2 h_N^4 \sum_{n=1}^{N}\frac{1}{\sigma_{Nn}^4},$$

which tends to zero due to (6.2) and (3.10). As for the first term, upon solving $1-\varepsilon = \cos(y)$ we have the relation $\cos^{-1}(1-\varepsilon) = \sqrt{2\varepsilon} + O(\varepsilon^{3/2})$. This means

$$\begin{aligned}\sin^2\left(\frac{t}{h_N}\cos^{-1}\left(1-\frac{h_N^2}{2\sigma_{N,n}^2}\right)\right) &= \sin^2\left(\frac{t}{h_N}\left[\frac{h_N}{\sigma_{Nn}} + O\left(\left(\frac{h_N}{\sigma_{Nn}}\right)^3\right)\right]\right) \\ &= \sin^2\left(\frac{t}{\sigma_{Nn}}\right) + t\cdot O\left(\frac{h_N^2}{\sigma_{Nn}^3}\right),\end{aligned}$$

where $O\left(h_N^2/\sigma_{Nn}^3\right)$ is a term bounded (uniformly in N, and n) by a constant times h_N^2/σ_{Nn}^3. Since $\int t\cdot\pi_N(t)\,\mathrm{d}t = \sigma_{N1}\mathbb{E}\left\{T\right\}$, we have

$$\begin{aligned}\lim_{N\to\infty}\mathbb{E}\left\{\Delta_N\right\} &= \lim_{N\to\infty}\frac{1}{2}\sum_{n=1}^{N}\left(\frac{h_N}{2\sigma_{Nn}}\right)^4 \int \sin^2\left(\frac{t}{h_N}\cos^{-1}\left(1-\frac{h_N^2}{2\sigma_{N,n}^2}\right)\right)\pi_N(t)\,\mathrm{d}t \\ &= \lim_{N\to\infty}\frac{1}{2}\sum_{n=1}^{N}\left(\frac{h_N}{2\sigma_{Nn}}\right)^4 \left\{\int \sin^2\left(\frac{t}{h_N}\right)\pi_N(t)\,\mathrm{d}t + O\left(\frac{\sigma_{N1}h_N^2}{\sigma_{Nn}^3}\right).\right\} \\ &= \frac{\alpha}{2} + \sum_{n=1}^{N} O\left(\frac{\sigma_{N1}h_N^6}{\sigma_{Nn}^7}\right).\end{aligned} \tag{6.7}$$

The first term is the desired limit. The second term tends to zero upon replacement of h_N by $\sum_{n=1}^{N}\sigma_{Nn}^{-4}$ (as in section 6.1) and the use of assumption (3.11).

The steps for variance are similar, and yield

$$\lim_{N\to\infty}\mathrm{Var}\left\{\Delta_N\right\} = \alpha.$$

The normal limit follows after verifying the Lindeberg condition [13]: For all $\varepsilon > 0$,

$$\lim_{N\to\infty} \sum_{n=1}^{N} \mathrm{Var}\left\{\delta_{Nn}^{\ell} : |\delta_{Nn}^{\ell} - \mathbb{E}\left\{\delta_{Nn}^{\ell}\right\}| > \varepsilon\right\} = 0. \tag{6.8}$$

One can check that $\delta_{Nn}^{\ell} - \mathbb{E}\left\{\delta_{Nn}^{\ell}\right\}$ is bounded by a term similar to (6.5) which tends uniformly to zero, so (6.8) follows. □

6.3. The simple step size

Proof of corollary 3.1. The simpler step size amounts to replacing the integrals $\int \sin^2(t/\sigma_{Nn})\pi_N(t)\,\mathrm{d}t$ in (6.7) with 1/2. This will be implied by sufficient decay of the remainder R_{Nn} in (6.1). Indeed, $|\hat{\pi}(\omega)| \le C|\omega|^{-\delta}$ implies

$$|R_{Nn}| \le C\left(\frac{\sigma_{Nn}}{\sigma_{N1}}\right)^{\delta},$$

This means (with $\lesssim$ denoting $\le$ up to a constant depending only on α, C_π),

$$\left|\frac{\bar{h}_N^4}{h_N^4} - 1\right| = \bar{h}_N^4\left|\bar{h}_N^{-4} - h_N^{-4}\right| \lesssim \bar{h}_N^4 \sum_{n=1}^{N} \frac{1}{\sigma_{Nn}^4}\left(\frac{\sigma_{Nn}}{\sigma_{N1}}\right)^{\delta}.$$

For any given K,

$$\begin{aligned}
\bar{h}_N^4 \sum_{n=1}^{N} \frac{1}{\sigma_{Nn}^4}\left(\frac{\sigma_{Nn}}{\sigma_{N1}}\right)^{\delta} &= \bar{h}_N^4 \sum_{n=1}^{K} \frac{1}{\sigma_{Nn}^4}\left(\frac{\sigma_{Nn}}{\sigma_{N1}}\right)^{\delta} + \bar{h}_N^4 \sum_{n=K+1}^{N} \frac{1}{\sigma_{Nn}^4}\left(\frac{\sigma_{Nn}}{\sigma_{N1}}\right)^{\delta} \\
&\le \bar{h}_N^4 \sum_{n=1}^{K} \frac{1}{\sigma_{Nn}^4} + \left(\frac{\sigma_{NK}}{\sigma_{N1}}\right)^{\delta} \bar{h}_N^4 \sum_{n=K+1}^{N} \frac{1}{\sigma_{Nn}^4} \\
&\lesssim K\frac{\bar{h}_N^4}{\sigma_{NK}^4} + \left(\frac{\sigma_{NK}}{\sigma_{N1}}\right)^{\delta}.
\end{aligned}$$

As $N \to \infty$, the first term tends to zero due to (6.2). The second is bounded by r_K by our hypothesis. Therefore,

$$\lim_{N\to\infty}\left|\frac{\bar{h}_N^4}{h_N^4} - 1\right| \lesssim r_K.$$

Since K was arbitrary, and $r_K \to 0$, we have shown $\bar{h}_N^4/h_N \to 0$. □

Acknowledgements

The authors would like to acknowledge the value of many conversations with members of the TensorFlow probability team. In particular (in alphabetical order!) Josh Dillon, Matt Hoffman, Pavel Sountsov, and Srinivas Vasudevan.

References

[1] Amaran, S., Sahinidis, N. V., Sharda, B. and Bury, S. J. (2016). Simulation optimization: a review of algorithms and applications. *Ann. Oper. Res.* **240** 351–380.

[2] Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Stat. Comput.* **18** 343–373.

[3] Bai, Z. D. (1999). Methodologies in spectral analysis of large dimensional random matrices, a review. *Statistica Sinica* **9** 611–662.

[4] Bales, B., Pourzanjani, A., Vehtari, A. and Petzold, L. (2019). Selecting the Metric in Hamiltonian Monte Carlo. *arXiv preprints*.

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

[6] Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo.

[7] Betancourt, M. J., Byrne, S. and Girolami, M. (2014). Optimizing The Integrator Step Size for Hamiltonian Monte Carlo. *arXiv preprints*.

[8] Bose, A., Gangopadhyay, S. and Sen, A. (2010). Limiting spectral distribution of XX' matrices. *Ann. Inst. Henri Poincare Probab. Stat.* **46** 677–707.

[9] Bose, A. and Sen, A. (2008). Another look at the moment method for large dimensional random matrices. *Electron. J. Probab.* **13** 588–628.

[10] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. and Riddell, A. (2017). Stan: A Probabilistic Programming Language.

[11] Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. and Saurous, R. A. (2017). TensorFlow Distributions.

[12] Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987). Hybrid Monte Carlo.

[13] Durrett, R. (2010). *Probability: Theory and Examples*. Cambridge University Press.

[14] Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

[15] Hansen, P. C. (1988). Hansen – Computing 1988 – Computation of the Singular Value Expansion.pdf. *Computing* **40** 185–199.

[16] Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D. and Vasudevan, S. (2019). NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. *ArXiv preprints*.

[17] Hoffman, M. D. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* **15**.

[18] Horn, R. A., Horn, R. A. and Johnson, C. R. (1990). *Matrix Analysis*. Cambridge University Press.

[19] Kaipio, J. and Somersalo, E. (2006). *Statistical and Computational Inverse Problems*. Springer Science & Business Media.
[20] Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D. and Dillon, J. V. (2020). Modern Markov Chain Monte Carlo Tools Built for Modern Hardware.
[21] Leimkuhler, B. and Reich, S. (2005). *Simulating Hamiltonian Dynamics*. Cambridge University Press.
[22] Mackenze, P. B. (1989). An improved hybrid Monte Carlo method.
[23] Marčenko, V. A. and Pastur, L. A. (1967). Distribution of Eigenvalues for some Sets of Random Matrices.
[24] Neal, R. M. (2012). MCMC using Hamiltonian dynamics. *arXiv preprints*.
[25] Parno, M. and Marzouk, Y. (2018). Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification* **6** 645–682.
[26] Peherstorfer, B. and Marzouk, Y. (2019). A transport-based multifidelity preconditioner for Markov chain Monte Carlo. *Adv. Comput. Math.* **45** 2321–2348.
[27] Robert, C. P. and Casella, G. (2004). Monte Carlo Statistical Methods.
[28] Silverstein, J. W. (1985). The Smallest Eigenvalue of a Large Dimensional Wishart Matrix. *Ann. Probab.* **13** 1364–1368.
[29] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1. 0 Contributors (2019). SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python.