
Curvature-Adaptive Langevin Monte Carlo

Qurui Zhang
Boston University
zhangqr@bu.edu

Abstract

We propose Curvature-Adaptive Langevin Monte Carlo (CALM), a Metropolis-adjusted Langevin algorithm that preconditions proposals using the local Hessian of the log-density via automatic differentiation. By eigendecomposing the negative Hessian and clamping eigenvalues, CALM constructs a position-dependent preconditioner that adapts proposal scale and orientation to local geometry. On the Rosenbrock distribution, CALM achieves ESS competitive with HMC. On Neal’s Funnel, where optimal scale varies by orders of magnitude, CALM’s local curvature adaptation automatically adjusts proposal scale without requiring a hand-specified mass matrix—a structural advantage over fixed-step-size methods that either under-cover the tails or sacrifice efficiency through conservative tuning. Ablation studies confirm curvature adaptation is the key ingredient.

1 Introduction

MCMC efficiency depends critically on how well proposals match the target geometry. RWMH uses isotropic proposals and ignores the target’s shape. HMC [3] uses gradients for coherent moves but its fixed mass matrix cannot adapt to varying-scale regions. Standard MALA adds gradient drift but still uses isotropic noise.

We propose **CALM**, which extends MALA by preconditioning with the local Hessian of $\log p(\theta)$. Related to Riemannian MALA [2], our approach uses JAX’s `jax.hessian` for exact Hessian computation, eigendecomposition for curvature extraction, and eigenvalue clamping for robustness in non-log-concave regions. The MH correction ensures exact targeting despite asymmetric, position-dependent proposals.

2 Method

Given target $\pi(\theta) \propto \exp(\ell(\theta))$, we compute gradient $g = \nabla \ell(\theta)$ and Hessian $H = \nabla^2 \ell(\theta)$. We eigendecompose $-H = Q \text{diag}(\lambda) Q^\top$ and clamp: $\tilde{\lambda}_i = \max(\lambda_i, \lambda_{\min})$. The preconditioner is $M = Q \text{diag}(1/\tilde{\lambda}) Q^\top$.

Proposal: $\theta' = \theta + \frac{\varepsilon^2}{2} Mg + \varepsilon M^{1/2} z$, $z \sim \mathcal{N}(0, I)$, so $\theta' \sim \mathcal{N}(\theta + \frac{\varepsilon^2}{2} Mg, \varepsilon^2 M)$.

MH correction: Since $M(\theta)$ is position-dependent, $q(\theta'| \theta) \neq q(\theta | \theta')$. We accept with probability $\min(1, \alpha)$ where $\log \alpha = [\ell(\theta') - \ell(\theta)] + [\log q(\theta | \theta') - \log q(\theta' | \theta)]$, with:

$$\log q(\theta_{\text{to}} | \theta_{\text{from}}) = -\frac{D}{2} \log(2\pi) - D \log \varepsilon - \frac{1}{2} \log \det M - \frac{1}{2\varepsilon^2} (\theta_{\text{to}} - \mu)^\top M^{-1} (\theta_{\text{to}} - \mu). \quad (1)$$

Each step costs $O(D^3)$ for eigendecomposition (done twice: current and proposed positions).

Algorithm 1 CALM: Curvature-Adaptive Langevin Monte Carlo

Require: Log-density ℓ , step size ε , eigenvalue floor λ_{\min} , initial θ_0

- 1: **for** $t = 1, \dots, T$ **do**
- 2: $g \leftarrow \nabla \ell(\theta_{t-1})$; eigendecompose $-\nabla^2 \ell(\theta_{t-1}) = Q \text{diag}(\lambda) Q^\top$
- 3: $\tilde{\lambda}_i \leftarrow \max(\lambda_i, \lambda_{\min})$; $M \leftarrow Q \text{diag}(1/\tilde{\lambda}) Q^\top$
- 4: Propose $\theta' \leftarrow \theta_{t-1} + \frac{\varepsilon^2}{2} Mg + \varepsilon M^{1/2} z$, $z \sim \mathcal{N}(0, I)$
- 5: Recompute g' , M' at θ' ; accept/reject via MH ratio
- 6: **end for**

3 Experiments

We draw 50,000 samples from each method on two benchmarks. RWMH and HMC use BlackJAX; CALM uses `jax.grad`, `jax.hessian`, and `jnp.linalg.eigh`. All chains are run as a single sequence; bulk ESS is computed via ArviZ [1], which issues a shape warning for single-chain inputs (rendering \hat{R} unavailable). Bulk ESS values remain informative for comparing relative mixing efficiency across methods, though absolute magnitudes should be interpreted conservatively.

Rosenbrock: $\log p(x, y) \propto -0.05(1-x)^2 - (y-x^2)^2$. A narrow curved ridge. RWMH ($\sigma=1.0$, acc. 50%) attains a high bulk ESS (16,997 / 18,094 for x/y), yet its posterior s.d. for x (0.64) is far below HMC’s (2.20), revealing that rapid local decorrelation within a narrow neighbourhood does not imply global coverage of the banana. HMC ($\varepsilon=0.2$, $L=10$, acc. 75%) follows the curvature via gradient-based dynamics and recovers the correct marginal variances, despite lower bulk ESS (42 / 93). CALM ($\varepsilon=0.5$, $\lambda_{\min}=10^{-3}$, acc. 45%, ESS 29 / 138) uses the Hessian to align proposals with the ridge; its ESS in the y -dimension is comparable to HMC, and the single-step proposal structure—rather than curvature adaptation per se—is the principal bottleneck for the x -dimension.

Neal’s Funnel: $v \sim \mathcal{N}(0, 9)$, $x \sim \mathcal{N}(0, e^v)$. The optimal scale for x varies exponentially with v , making any fixed step size suboptimal. We tuned both baselines toward their respective recommended acceptance ranges. Despite tuning, each exhibits a characteristic failure mode. RWMH ($\sigma=2.0$, acc. 38%) achieves a deceptively high bulk ESS (6,864 / 16,125), but its posterior s.d. for v (1.54) is substantially below the true value of 3.00, indicating that the chain mixes efficiently within an under-covered region rather than exploring the full distribution. HMC ($\varepsilon=0.15$, $L=10$) converged to 93.6% acceptance—above the 70–85% target—indicating an overly conservative step size; while HMC recovers the correct marginal variance of v ($\hat{\sigma}_v \approx 3.17$), the reduced displacement per step limits bulk ESS to 168 / 206. CALM ($\varepsilon=0.3$, $\lambda_{\min}=10^{-3}$, acc. 26%, ESS 110 / 77) attains lower raw ESS than either baseline, yet automatically scales proposals via the local Hessian without requiring a hand-specified mass matrix.

We report acceptance rates and bulk ESS (ArviZ, single chain; see Section 3). On the Funnel, raw ESS values require careful interpretation: RWMH’s high ESS reflects efficient local mixing in an under-covered region, whereas CALM and HMC both recover the correct marginal of v at the cost of lower per-sample throughput. The key distinction is that CALM achieves this coverage through automatic curvature adaptation, while HMC requires a correctly specified mass matrix and well-chosen step size.

Ablation: step size. Sweeping $\varepsilon \in \{0.05, \dots, 2.0\}$: acceptance decreases monotonically with ε ; optimal ESS at $\varepsilon \approx 0.2\text{--}0.5$.

Ablation: eigenvalue floor. Sweeping $\lambda_{\min} \in \{10^{-5}, \dots, 10^2\}$: as $\lambda_{\min} \rightarrow \infty$, $M \rightarrow (1/\lambda_{\min})I$ and CALM degrades to MALA. The benefit is most pronounced on the Funnel, where Hessian eigenvalues vary by orders of magnitude.

4 Discussion

Strengths. CALM excels on distributions with varying local geometry. The Hessian provides direct curvature information, eigenvalue clamping ensures robustness, and MH correction guarantees exactness.

Weaknesses. The $O(D^3)$ eigendecomposition (done twice per step) limits scalability beyond $D \approx 100$. Unlike HMC’s multi-step leapfrog, CALM makes single-step proposals, limiting long-range moves; this is the primary reason for its comparatively modest bulk ESS on both benchmarks, rather than the quality of curvature adaptation itself. Like all local methods, CALM cannot jump between separated modes. Additionally, the single-chain experimental setup precludes \hat{R} diagnostics; multi-chain runs would provide stronger convergence guarantees.

Future work. Diagonal/low-rank Hessian approximations for scalability; combining curvature preconditioning with leapfrog dynamics (Riemannian HMC); adaptive step size tuning.

5 AI Collaboration

I used Claude (Anthropic) as a coding agent via the Claude Code CLI throughout this project.

Algorithm design. I described the goal (curvature-adaptive sampling) and Claude helped formalize the connection to Riemannian MALA, suggested `jax.hessian` for exact computation, and proposed eigenvalue clamping. The insight that eigendecomposition naturally provides $M^{1/2}$ came from the discussion.

Implementation. Claude wrote the CALM implementation including preconditioner construction, proposal density, and MH correction. I verified the asymmetric proposal density calculation—ensuring the log-determinant and Mahalanobis distance were correct for position-dependent covariance $\varepsilon^2 M(\theta)$.

What worked well. Providing mathematical formulations for Claude to implement was highly effective. The agent excelled at translating math to JAX code and generating plotting/diagnostics boilerplate.

What needed correction. I carefully verified the MH acceptance ratio, as errors produce silently biased samples. I also guided hyperparameter choices based on domain knowledge.

Lesson. AI assistants are effective as mathematical translators and code generators, but humans must own correctness—especially for MCMC acceptance ratios where bugs are invisible without verification.

References

- [1] R. Kumar, C. Carroll, A. Hartikainen, and O. Martin. ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, 4(33):1143, 2019.
- [2] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *JRSS-B*, 73(2):123–214, 2011.
- [3] R. M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of MCMC*, ch. 5. Chapman and Hall/CRC, 2011.