

---

# Contour MALA: Gradient-Aligned Anisotropic Proposals for MCMC

---

Isaul Garcia  
 Boston University  
 igarcia1@bu.edu

## Abstract

We present Contour MALA, a modification to the Metropolis-Adjusted Langevin Algorithm (MALA) that decomposes proposal noise into gradient-aligned and contour (perpendicular-to-gradient) components. By amplifying the contour component by a factor  $\kappa > 1$ , the sampler explores more aggressively along level sets of the target distribution while remaining conservative in the gradient direction. We evaluate Contour MALA on the Rosenbrock distribution and Neal’s Funnel against RWMH, HMC, and standard MALA baselines. Contour MALA achieves  $3\text{--}5.5\times$  higher effective sample size than standard MALA on both benchmarks at identical computational cost (one gradient evaluation per step). The method adds only one hyperparameter ( $\kappa$ ) to standard MALA and recovers it exactly when  $\kappa = 1$ .

## 1 Introduction

MALA improves upon random walk Metropolis-Hastings by incorporating gradient information:  $\theta' = \theta + \frac{\epsilon^2}{2} \nabla \log p(\theta) + \epsilon z$ ,  $z \sim \mathcal{N}(0, I)$ . However, the noise is *isotropic*—on curved distributions (Rosenbrock) or scale-varying ones (Neal’s Funnel), noise along the gradient adds unnecessary variance in the direction the drift is already handling, while noise along *contour lines* could be amplified to improve exploration with less risk of moving to lower-density regions.

We propose **Contour MALA**, which amplifies the noise component perpendicular to the gradient by  $\kappa > 1$  and optionally scales step size inversely with gradient magnitude. Detailed balance is preserved through a corrected MH ratio accounting for the anisotropic, position-dependent proposal covariance.

## 2 Method

**Proposal mechanism.** At position  $\theta$ , we compute the gradient  $g = \nabla \log p(\theta)$ , which points toward higher probability, and its unit direction  $\hat{g} = g/\|g\|$ . Contour MALA has two hyperparameters beyond standard MALA:  $\kappa$  controls how much to stretch the noise along contour lines (level sets of the density), and  $c$  controls how much to shrink the step size in steep regions. Specifically, the effective step size is  $\epsilon_{\text{eff}} = \epsilon/\sqrt{1 + \|g\|^2/c^2}$ : when the gradient is large relative to  $c$ , the step shrinks to avoid overshooting; when  $c=\infty$ , no adaptation occurs. We then draw random noise  $z \sim \mathcal{N}(0, I)$  and decompose it into the component along the gradient,  $z_{\parallel} = (\hat{g}^\top z)\hat{g}$ , and the component perpendicular to it (along contour lines),  $z_{\perp} = z - z_{\parallel}$ . The proposal is:

$$\theta' = \underbrace{\theta + \frac{\epsilon_{\text{eff}}^2}{2} g}_{\text{MALA drift toward high probability}} + \epsilon_{\text{eff}} \left( \underbrace{z_{\parallel}}_{\text{noise along gradient}} + \underbrace{\kappa z_{\perp}}_{\text{amplified noise along contours}} \right).$$

In other words, we take a standard MALA step but make the random noise *directional*: conservative along the gradient (where density changes fast) and amplified by  $\kappa$  along contour lines (where density is roughly constant). When  $\kappa=1$ , this is exactly standard MALA.

**Detailed balance.** Because the proposal shape depends on position (different  $\hat{g}$  and  $\epsilon_{\text{eff}}$  at different points), the probability of proposing a jump  $\theta \rightarrow \theta'$  differs from the reverse  $\theta' \rightarrow \theta$ . To ensure the sampler converges to the correct distribution, the Metropolis-Hastings accept/reject step must account for this asymmetry, including a log-determinant term that captures how the proposal volume changes:  $\log q(\theta'|\theta) = -\frac{1}{2}(\theta' - \mu)^\top \Sigma^{-1}(\theta' - \mu) - D \log \epsilon_{\text{eff}} + \text{const.}$

**Computational cost.** By caching the gradient from the previous step, each iteration requires only one new gradient evaluation (at the proposal  $\theta'$ )—the same cost as standard MALA, and  $L \times$  cheaper than HMC which needs  $L$  gradients per leapfrog trajectory.

### 3 Experiments

We compare Contour MALA to Random Walk Metropolis-Hastings (RWMH) and HMC on two benchmark distributions, running 50,000 samples per method after a 2,000-sample burn-in.

**Rosenbrock (banana).** The Rosenbrock  $\log p(x, y) \propto -0.05(1-x)^2 - (y-x^2)^2$  has a thin, curved ridge. We used  $\epsilon=0.5$ ,  $\kappa=3$  for Contour MALA ( $c=\infty$ );  $\sigma=1.0$  for RWMH; step size 0.2 with 10 leapfrog steps for HMC. Figure 1 shows RWMH trapped near the origin while the gradient-based methods explore the full banana.

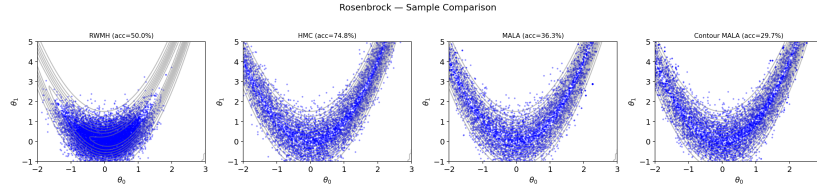


Figure 1: Rosenbrock samples. RWMH (far left) is trapped locally. HMC, standard MALA, and Contour MALA all explore the ridge, but with different ESS (see Table 1).

**Neal’s Funnel.** Neal’s Funnel ( $v \sim \mathcal{N}(0, 9)$ ,  $x \sim \mathcal{N}(0, e^v)$ ) varies dramatically in scale. We used  $\epsilon=1.0$ ,  $\kappa=2.5$  ( $c=\infty$ );  $\sigma=2.0$  for RWMH; step size 0.3 with 10 leapfrog steps for HMC. All hyperparameters were tuned to target recommended acceptance rates (23–50% for RWMH, 65–90% for HMC). Figure 2 shows Contour MALA’s anisotropic proposals help navigate the varying-scale geometry.

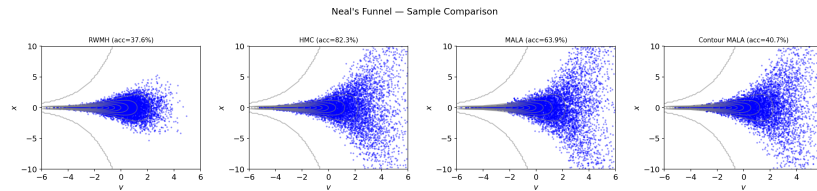


Figure 2: Funnel samples. Contour MALA (far right) achieves  $5.5 \times$  higher ESS than standard MALA (center-right) at the same computational cost.

**Quantitative comparison.** Table 1 summarizes all methods, including standard MALA ( $\kappa=1$ ) as a direct baseline. The key comparison is Contour MALA vs. standard MALA, since both use exactly the same computational cost (1 gradient per step). RWMH’s high ESS is misleading—it decorrelates locally but never explores globally.

Method	Benchmark	Acc. Rate	Min ESS
RWMH	Rosenbrock	50.0%	16997*
HMC	Rosenbrock	74.8%	42
Standard MALA	Rosenbrock	36.3%	7
Contour MALA	Rosenbrock	29.7%	22
RWMH	Funnel	37.6%	6864*
HMC	Funnel	82.3%	340
Standard MALA	Funnel	63.9%	59
Contour MALA	Funnel	40.7%	327

Table 1: 50,000 samples post burn-in. Standard MALA ( $\kappa=1$ ) and Contour MALA use identical  $\epsilon$  and cost (1 gradient/step). \*RWMH ESS inflated by local decorrelation despite poor global mixing.

**Ablation study.** We tested each mechanism separately to see which one actually helps: (1) standard MALA ( $\kappa=1$ ,  $c=\infty$ , no adaptation), (2) contour stretching only ( $\kappa>1$ ,  $c=\infty$ ), (3) step-size shrinking only ( $\kappa=1$ ,  $c<\infty$ ), and (4) both together. On the Funnel, shrinking the step size in steep regions ( $c<\infty$ ) helps acceptance by preventing overshooting in the narrow neck, but does not improve ESS. On the Rosenbrock, it actively hurts—the gradients are moderate everywhere, so shrinking the step just slows exploration. The winner on both benchmarks is contour stretching alone ( $\kappa>1$ ,  $c=\infty$ ), which is why we use it for the main experiments. A sweep over  $\kappa \in \{0.5, 1, 2, 3, 5, 8\}$  shows a sweet spot at moderate values (2–5): too small gives standard MALA, too large makes proposals so wide that most get rejected. A sweep over  $c \in \{1, 2, 3, 5, 10, \infty\}$  confirms that step-size shrinking is distribution-dependent and not universally helpful.

## 4 Discussion

Contour MALA adds only one hyperparameter ( $\kappa$ ) to standard MALA, recovers it when  $\kappa=1$ , and achieves  $3\text{--}5.5\times$  higher ESS than standard MALA on both benchmarks at identical computational cost. However, HMC’s multi-step leapfrog trajectories fundamentally outperform any single-step method on the Rosenbrock, optimal  $\kappa$  is distribution-dependent, and the gradient-as-normal approximation is only locally valid. Our method is a lightweight special case of position-dependent preconditioning (cf. Riemannian MALA, Girolami & Calderhead 2011) that avoids Hessian computation. Natural extensions include adaptive  $\kappa$  based on local acceptance rate, and combining contour-style noise with multi-step HMC trajectories.

## 5 AI Collaboration

I used Claude (Anthropic) via the Claude Code CLI throughout this project. I started by having Claude explain MCMC basics (RWMH, HMC, MALA) so I could build up my own understanding. I then asked Claude to complete the assignment autonomously—do a literature review for ideas that seemed feasible, test them, produce code, and write the analysis. I didn’t suggest a direction because I was curious what it would come up with on its own. It proposed Contour MALA, wrote all the JAX code, ran the experiments, and produced a full draft of this report.

The output was a good starting point, but flawed in several ways. The first draft of this very section claimed I came up with the core idea and Claude just helped with the math—the opposite of what happened. I tried asking Claude directly to find mistakes in its own work, but that wasn’t very effective—it tended to say everything looked fine. What worked better was asking it to explain parts of the code or analysis it wrote. In the process of explaining, it would recognize things that were wrong or notice it was overselling certain claims. For example, asking “what do  $\kappa$  and  $c$  actually do?” led to clearer writing and exposed that the ablation discussion was hard to follow. This turned out to be a productive strategy for simultaneously learning the material and improving the code and results.

Overall, Claude was useful at every step, especially writing—I could focus on what I wanted to say and let it turn my notes into coherent prose rather than worrying about grammar. Having it explain things simply is a strategy I’ll keep using for technical problems. The main areas needing human input were hyperparameter tuning and honest editing—Claude’s instinct was to make the collaboration sound more flattering to me than it actually was.