AMBIENT INTEGRATION RULES FOR RIEMANNIAN MANIFOLD HAMILTONIAN MONTE CARLO

Anonymous authors
Paper under double-blind review

ABSTRACT

Riemannian manifold Hamiltonian Monte Carlo (RMHMC) adapts the metric of Hamiltonian Monte Carlo to the local geometry of the posterior, but every integration step requires the solution of an implicit system that contains derivatives of the metric. We consider target posterior distributions $\text{pdf}(q \mid x)$ of hierarchical models, where the joint is given by a diffeomorphic map $\xi \to (q, x)$, where ξ is a standard multivariate normal distributed random variable. We use the fact that conditioning on x restricts ξ to a submanifold with chart q to derive integrators of RMHMC that take paths along secants through ambient space. To do this we consider variational integration methods and rephrase their kinetic term in ambient coordinates which makes them invariant under reparamterisation of the manifold. We show that each implicit integration method allows for an analogous secant version. We show that within our framework the solution of the implicit RATTLE integrator can be derived as a special case and is identical to the Generalized Leapfrog integrator in ambient space. We compare eight solvers for four implicit integrators, including the different solvers for RATTLE, on four test problems, including Neal's funnel, a supernova dust emission model and a hierarchical Gaussian process. Our experiments show that ambient secant formulations are superior in both robustness to the chosen step size as well as solver cost, where we observe that ambient formulations can allow for a 10x larger step-size. We then compare the resulting implicit MCMC schemes to the explicit LMC scheme and find that when matched for trajectory length, secant based methods can be competitive and can even improve over LMC especially for higher dimensional problems, even when using an efficient implementation of LMC that makes use of the geometric structure.

1 INTRODUCTION

Hamiltonian Monte Carlo (HMC, Neal (2011)) is known to perform poorly on distributions with varying curvature. This is a common problem when sampling from hierarchical models, where the curvature changes naturally with the position (Betancourt & Girolami, 2019). For example, when computing the latent posterior of a variational autoencoder (VAE, Kingma & Welling (2014)) that predicts $x|\theta \sim \mathcal{N}(\mu(\theta), \Sigma(\theta))$ the posterior $\theta|x$ is affected by the sensitivity of $D_\theta\Sigma(\theta)$ and $D_\theta\mu(\theta)$. In naive HMC and derived algorithms like the No-U-Turn sampler (NUTS,Hoffman & Gelman (2011)), a changing curvature forces the sampler to use a small step size that is compatible with the most difficult parts of the distribution. While there exist approaches to dynamically choose the step size (Bou-Rabee et al., 2026b;a; Modi et al., 2024) they typically come at additional costs of proposing multiple step sizes and acting on their rejection.

Riemannian Manifold HMC (RMHMC, Girolami & Calderhead (2011)) was designed to address the problem of varying curvature directly, by replacing the fixed Euclidean metric by a varying Riemannian metric $G(\theta)$. However, in practice RMHMC has a poor track record. Integration of the Hamiltonian flow requires implicit methods (Girolami & Calderhead, 2011; Brofos & Lederman, 2021a) which are only approximately reversible and volume-preserving (Brofos & Lederman, 2021b;a). While a backward solve can be used to check for violations of reversibility (Lelièvre et al., 2019; Au et al., 2023), it is expensive and there is no consensus on what degree of violation is acceptable as numerical noise. The cost of the implicit solves motivated the explicit Lagrangian Monte Carlo (LMC) integrator (Lan et al., 2015), which is no longer volume-preserving and requires

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

four additional determinants per step in the Metropolis-Hastings acceptance (Brofos & Lederman, 2022).

Furthermore, there is no accessible natural choice of $G(\theta)$, and a number of choices have been constructed from the Hessian (Betancourt, 2013), from gradient covariances (Kleppe, 2024) or from an embedding of the log density (Hartmann et al., 2022), none of which presents a well-performing default choice. The most natural metric for RMHMC is the pullback metric derived from the diffeomorphism mapping the target distribution to the standard normal, and RMHMC using this metric is equivalent to HMC applied to the normal distribution (Kailas et al., 2026). However, this metric is unknown, and even if it was known, one would be better off applying HMC to the normal distribution directly. Recently, an approach was suggested (Graham & Storkey, 2017; Au et al., 2023) that considers the case of a hierarchical model, of the joint distribution (θ, x), where the diffeomorphism to the standard normal is known, but we are interested in the conditional $\theta|x$. In this case, we can represent the conditional distribution as a potential over a manifold embedded in the transformed space, which allows sampling with the RATTLE integrator of constrained HMC (Brubaker et al., 2012; Lelièvre et al., 2019), which solves a constrained system of equations in the embedded space.

In this work, we make three contributions

- We take a novel point of view on RMHMC for hierarchical models by taking the Lagrangian formulation of the Hamiltonian and observe that the kinetic term can be rephrased naturally in terms of the coordinates of the embedding. This then allows us to derive implicit integrators in chart coordinates of the manifold, rather than solving a constrained problem in the embedded space. One of our solvers is equivalent to a chart-based reformulation of RATTLE.
- We perform a careful evaluation of 8 solvers for 4 implicit integrators on a difficult set of benchmark problems. We measure integrator performance and reversibility, and compare the best performing implicit methods with LMC for a full end-to-end sampling run. Our evaluation is designed to distinguish solver iterations and their varying costs from python-overhead dominated wall-clock times.
- As an aside, we show that LMC in this setting can be computed more efficiently, allowing for a fair comparison where all algorithms are compared using their best implementation.

2 Background

2.1 Riemannian manifold Hamiltonian Monte Carlo (RMHMC)

Let $\text{pdf}(q) \propto \exp(-U(q))$, $q \in \mathbb{R}^n$, be the target density and let $G(q) : \mathbb{R}^n \to \text{Sym}^+(n)$ be a smooth function. RMHMC (Girolami & Calderhead, 2011) augments q with a momentum $p \in \mathbb{R}^n$ and samples the joint density $\text{pdf}(q, p) \propto \exp(-H(q, p))$ with the Hamiltonian

$$H(q,p) = V(q) + \tfrac{1}{2}\, p^T G(q)^{-1} p \ , \tag{1}$$

$$V(q) := U(q) + \tfrac{1}{2} \log\det G(q) \ . \tag{2}$$

Since $p|q \sim \mathcal{N}(0, G(q))$, the marginal distribution of q is the target distribution. We write $z = (q, p)$ with $H(z) = H(q, p)$. Using $J = \left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$ and $\Gamma(q, u) := \frac{1}{2}\nabla_q \left[u^T G(q)\, u\right]$ the Hamiltonian flow is

$$\dot z = J\,\nabla H(z) = \begin{pmatrix} \nabla_p H(z) \\ -\nabla_q H(z) \end{pmatrix} = \begin{pmatrix} G(q)^{-1} p \\ \Gamma\big(q, G(q)^{-1}p\big) - \nabla V(q) \end{pmatrix} \ . \tag{3}$$

The standard approach in RMHMC is to sample $p_0 \sim \mathcal{N}(0, G(q_0))$ and then to integrate the flow from the starting point $z_0 = (q_0, p_0)$, for k time integration steps of length h using a reversible and volume preserving integrator. With this, the final point z_k can be accepted with probability $\min\{1, \exp(H(z_0) - H(z_k))\}$ and p_k is discarded (Girolami & Calderhead, 2011).

2.2 Integration rules for RMHMC

A simple way to construct volume-preserving integrators mapping $z_0 \to z_1$ uses the convex dual of H (Marsden & West, 2001). With the conjugate variable $\dot q = G(q)^{-1}p$, it is given by the Lagrangian

$$L(q, \dot q) = \sup_p \left[p^T \dot q - H(q, p)\right] = \tfrac{1}{2}\, \dot q^T G(q)\, \dot q - V(q) \ . \tag{4}$$

Starting at $q(0) = q_0$, the path that follows (3) for time h and passes through $q_1 = q(h)$ fulfills

$$\begin{aligned} S(q_0, q_1) &= \int_0^h L\big(q(t), \dot{q}(t)\big)\, dt \ , \\ p_0 &= -\nabla_{q_0} S(q_0, q_1) \\ p_1 &= \nabla_{q_1} S(q_0, q_1) \ . \end{aligned} \tag{5}$$

Replacing $S(q_0, q_1)$ by any integration rule $\hat{S}(q_0, q_1)$ with $\hat{S}(q_0, q_1) = \hat{S}(q_1, q_0)$ defines a symplectic (and thus volume-preserving) and reversible map $(q_0, p_0) \mapsto (q_1, p_1)$ wherever the equation for q_1 has a locally unique solution (Marsden & West, 2001). A well known integrator of (3) is the implicit midpoint integrator (Brofos & Lederman, 2021a):

$$z_1 = z_0 + h\, J\, \nabla H\big(\tfrac{1}{2}(z_0 + z_1)\big) \tag{6}$$

While conceptually simple, it requires the solution in both p_1 and q_1 simultaneously. Due to the simplicity of the equation for q_1, we can eliminate p_1 from it and obtain

$$\begin{aligned} hp_0 &= G\big(\tfrac{q_0+q_1}{2}\big)(q_1 - q_0) - \tfrac{1}{2}\Gamma\big(\tfrac{q_0+q_1}{2}, q_1 - q_0\big) + \tfrac{h^2}{2}\nabla V\big(\tfrac{q_0+q_1}{2}\big) \\ hp_1 &= G\big(\tfrac{q_0+q_1}{2}\big)(q_1 - q_0) + \tfrac{1}{2}\Gamma\big(\tfrac{q_0+q_1}{2}, q_1 - q_0\big) - \tfrac{h^2}{2}\nabla V\big(\tfrac{q_0+q_1}{2}\big) \ , \end{aligned} \tag{7}$$

where the equation of p_1 is explicit given q_1. Notably, this form can be derived from (5) directly, by setting $\hat{S}^{\text{mid}}(q_0, q_1) = h\, L\big(\tfrac{1}{2}(q_0 + q_1), (q_1 - q_0)/h\big)$. The first integrator proposed was the Generalized Leapfrog integrator (Girolami & Calderhead, 2011; Brofos & Lederman, 2021a)

$$\begin{aligned} p_{1/2} &= p_0 - \tfrac{h}{2}\nabla_q H(q_0, p_{1/2}) \\ q_1 &= q_0 + \tfrac{h}{2}\big[\nabla_p H(q_0, p_{1/2}) + \nabla_p H(q_1, p_{1/2})\big] \\ p_1 &= p_{1/2} - \tfrac{h}{2}\nabla_q H(q_1, p_{1/2}) \ , \end{aligned} \tag{8}$$

and the first two equations need to be solved one after the other. This integrator is generated by the trapezoidal rule with a momentum that is shared by both end points, see Appendix A.

Finally, the Lagrangian Monte Carlo (LMC) integrator of Lan et al. (2015) integrates $(q, \dot{q})$ instead of (q, p). With the directional derivative $\partial_u G = \sum_k u_k\, \partial_k G$ and the Jacobian $D_q(Gu)$ of $G(q)u$ with respect to q at fixed u, the matrix $\Omega(q, u) = \frac{1}{2}\, G(q)^{-1}\big[\partial_u G(q) + D_q(G(q)u) - D_q(G(q)u)^T\big]$ collects the Christoffel symbols, and one step is

$$\begin{aligned} \dot{q}_{1/2} &= \big[I + \tfrac{h}{2}\Omega(q_0, \dot{q}_0)\big]^{-1}\big[\dot{q}_0 - \tfrac{h}{2}\, G(q_0)^{-1}\nabla V(q_0)\big] \\ q_1 &= q_0 + h\, \dot{q}_{1/2} \\ \dot{q}_1 &= \big[I + \tfrac{h}{2}\Omega(q_1, \dot{q}_{1/2})\big]^{-1}\big[\dot{q}_{1/2} - \tfrac{h}{2}\, G(q_1)^{-1}\nabla V(q_1)\big] \ . \end{aligned} \tag{9}$$

The integrator is explicit, reversible and second order, but not volume preserving (Lan et al., 2015; Brofos & Lederman, 2022). The acceptance probability is corrected by the Jacobian determinant of the map, which is a product of four determinants of $I \pm \frac{h}{2}\Omega$ per step (Lan et al., 2015).

2.3 RATTLE INTEGRATORS

An alternative route to the implicit integration of (3) is the RATTLE integrator (Lelièvre et al., 2019; Au et al., 2023) which can be used when the target distribution can be written as posterior pdf$(\theta|x)$ with observed variable $x \in \mathbb{R}^m$ and latent $\theta \in \mathbb{R}^n$. We assume that the joint pdf(θ, x) is a hierarchical model generated by mapping a pair of multivariate standard normal random variables $(q, \varepsilon) \in \mathbb{R}^n \times \mathbb{R}^m$, via a diffeomorphism $\theta = \phi(q)$, $x = \psi_q^{-1}(\varepsilon)$. Since ϕ is a diffeomorphism, we can sample from the posterior distribution $q|x$ and map $q \to \theta$ afterwards. With this setup, $q|x$ is the target pdf$(q|x) \propto \exp(-U(q))$ of Section 2.1 with potential

$$U(q) = U^A\big(q, \psi_q(x)\big) \ , \qquad U^A(q, \varepsilon) = \tfrac{1}{2}\|q\|^2 + \tfrac{1}{2}\|\varepsilon\|^2 - \log\big|\det \partial_x \psi_q(x)\big| \ ,$$

i.e., U^A is the representation of U when restricted to the embedded manifold $\mathcal{M} = \{(q, \varepsilon) : c(q, \varepsilon) = 0\}$ with constraint $c(q, \varepsilon) = \psi_q^{-1}(\varepsilon) - x$. The manifold has chart $\Psi(q) = (q, \psi_q(x))$ and within the chart, the pullback metric $G(q)$ of the Euclidean metric on (q, ε) gives $G(q) =$

$D\Psi(q)^T D\Psi(q)$. As an example of this framework, Variational Autoencoders (VAE) with conditional distribution $x|q \sim \mathcal{N}(\mu(q), \sigma^2(q)I)$ can be rephrased via $\phi(q) = q$ and $\psi_q^{-1}(\varepsilon) = \mu(q) + \sigma(q)\varepsilon$, the reparameterisation trick of the VAE.

We can now rephrase sampling in ambient coordinates $q_1^A = (q_1, \varepsilon_1)$ as a constrained system of equations, which results in the RATTLE family of solvers using Lagrange multipliers $\lambda \in \mathbb{R}^m$:

$$\begin{aligned} q_1^A &= \Psi(q_0) + h\, D\Psi(q_0) G(q_0)^{-1} \big(p_0 - \tfrac{h}{2}\nabla V(q_0)\big) - h\, Dc\big(\Psi(q_0)\big)^T \lambda \\ q_1^A &\in \mathcal{M} \\ p_1 &= \tfrac{1}{h}\, D\Psi(q_1)^T \big(q_1^A - \Psi(q_0)\big) - \tfrac{h}{2}\nabla V(q_1) \ . \end{aligned} \tag{10}$$

3 Ambient Integrators for RMHMC

When considering the derivation of the implicit integrators in Section 2 using the Lagrangian $L(q, \dot{q})$, (4), it becomes apparent that, since q is the position variable on the manifold $\mathcal{M}$ with metric $G(q)$, we must have $\dot{q} \in T_q\mathcal{M}$. However, the discretization uses $\dot{q} \propto q_1 - q_0$ which are positional variables at different points and thus these quantities are incommensurate. The *consistent* way to define $\dot{q}$ on a manifold is via the inverse of the exponential map, which is not available for most manifolds of interest. Thus, $q_1 - q_0$ is only a sensible quantity as long as $G(q_0) \approx G(q_1)$, which we hypothesize limits the step length. Based on this intuition, we will generalize the RATTLE approach in Section 2.3 of phrasing the integration problem in ambient coordinates. Our approach can be understood as approximating the integration path $q(t)$ by a piecewise linear path through ambient space, where only the end points of a piece are required to lie on $\mathcal{M}$. This allows us to consistently define $\dot{q}$ as the slope of the secant between two points through ambient space.

Following RATTLE, our target is the posterior distribution $q|x$ on an embedded manifold $\mathcal{M}$ with chart map $\Psi(q) = (q, \psi_q(x))$ and a Euclidean ambient metric inducing $G(q) = D\Psi(q)^T D\Psi(q)$. Using this, an element $\dot{q} \in T_q\mathcal{M}$ can be represented in ambient coordinates $\dot{q}^A$ via

$$\dot{q}^A = D\Psi(q)\, \dot{q}, \quad \dot{q}^T G(q)\, \dot{q} = (\dot{q}^A)^T\, \dot{q}^A$$

As a result L rephrased with $\dot{q}$ in ambient coordinates simplifies to

$$L^A(q, \dot{q}^A) = -V(q) + \frac{1}{2}\,(\dot{q}^A)^T \dot{q}^A \ .$$

In ambient coordinates, the average velocity of any path from $\Psi(q_0)$ to $\Psi(q_1)$ is exactly $\dot{q}^A = \frac{1}{h}(\Psi(q_1) - \Psi(q_0))$, the slope of the secant. The advantage compared to the approximation $\dot{q} = \frac{1}{h}(q_1 - q_0)$ in chart coordinates is that $\dot{q}^A$ is independent of the chart. As first integrator, we consider the midpoint rule. We define

$$\begin{aligned} \hat{S}^{\text{mid},A}(q_0, q_1) &= h\, L^A\big(\tfrac{1}{2}(q_0 + q_1), \tfrac{1}{h}(\Psi(q_1) - \Psi(q_0))\big) \\ &= \tfrac{1}{2h}\big\|\Psi(q_1) - \Psi(q_0)\big\|^2 - h\, V\big(\tfrac{1}{2}(q_0 + q_1)\big) \ . \end{aligned} \tag{11}$$

Inserting $\hat{S}^{\text{mid},A}$ for S in (5) gives the system

$$\begin{aligned} hp_0 &= D\Psi(q_0)^T \big(\Psi(q_1) - \Psi(q_0)\big) + \tfrac{h^2}{2}\nabla V\big(\tfrac{1}{2}(q_0 + q_1)\big) \\ hp_1 &= D\Psi(q_1)^T \big(\Psi(q_1) - \Psi(q_0)\big) - \tfrac{h^2}{2}\nabla V\big(\tfrac{1}{2}(q_0 + q_1)\big) \ . \end{aligned} \tag{12}$$

Compared to (7), the expensive Γ term involving second derivatives of Ψ vanished and got replaced by a term involving the Jacobian. However, the resulting integrator is not reparameterisation invariant due to the presence of the midpoint $\frac{1}{2}(q_0 + q_1)$. Using the trapezoidal rule for $\hat{S}$ with the velocity $\dot{q}^A$ at both end points leads to

$$\begin{aligned} \hat{S}^{\text{trap},A}(q_0, q_1) &= \tfrac{h}{2}\Big[L^A\big(q_0, \tfrac{1}{h}(\Psi(q_1) - \Psi(q_0))\big) + L^A\big(q_1, \tfrac{1}{h}(\Psi(q_1) - \Psi(q_0))\big)\Big] \\ &= \tfrac{1}{2h}\big\|\Psi(q_1) - \Psi(q_0)\big\|^2 - \tfrac{h}{2}\big(V(q_0) + V(q_1)\big) \ , \end{aligned} \tag{13}$$

and gives the system

$$\begin{aligned} hp_0 &= D\Psi(q_0)^T \big(\Psi(q_1) - \Psi(q_0)\big) + \tfrac{h^2}{2}\nabla V(q_0) \\ hp_1 &= D\Psi(q_1)^T \big(\Psi(q_1) - \Psi(q_0)\big) - \tfrac{h^2}{2}\nabla V(q_1) \ . \end{aligned} \tag{14}$$

Table 1: Implicit solvers compared in Experiment 1.

Algorithm	Equation	Solver
Secant, Precond.	(14)	Picard, Preconditioner $G(q_0)$
Secant, Newton	(14)	Newton
RATTLE, Precond.	(10)	Picard, Preconditioner $Dc(\Psi(q_0))\, Dc(\Psi(q_0))^T$
RATTLE, Newton	(10)	Newton
Secant Midpoint	(12)	Picard, Preconditioner $G(q_0) + \frac{h^2}{4}\nabla^2 V(q_0)$
Midpoint	(7)	Picard, Preconditioner $G(q_0) + \frac{h^2}{4}\nabla^2 V(q_0)$
Midpoint joint	(6)	Picard
Leapfrog	(8)	Picard

Since $\hat{S}^{\text{trap},A}$ is invariant under reparametrization, the resulting integrator is also invariant. Note that only the first equation is implicit, and both the gradient of V and the Jacobian of Ψ are functions of q_0, which means that in contrast to (7), they are constant. Thus, the trapezoidal rule in ambient coordinates only requires the computation of $D\Psi$ at two points. We refer to the integrator generated by (13) as Secant and to the integrator generated by (11) as Secant Midpoint.

We can relate the system (14) to the systems defined in Section 2. First of all, the solution of the system defines the same map as RATTLE. RATTLE keeps q_1^A on $\mathcal{M}$ through the multiplier term, which is normal to $\mathcal{M}$ at $\Psi(q_0)$. When we parametrize $q_1^A = \Psi(q_1)$, $q_1^A \in \mathcal{M}$ holds by construction, and multiplying (10) by $D\Psi(q_0)^T$ removes the multiplier term, since $Dc(\Psi(q_0))\, D\Psi(q_0) = 0$, and leaves (14). Note that while Generalized Leapfrog (8) is a different map, since velocities and momenta coincide in ambient coordinates, using the same approach to derive Generalized Leapfrog leads to system (14).

As a final method, we observe that for LMC the Christoffel symbols of the first kind of the pullback metric are $\langle \partial_l \Psi, \partial_k \partial_j \Psi \rangle$, so the matrix Ω in (9) becomes

$$\Omega(q,u) = G(q)^{-1} D\Psi(q)^T\, \partial_u D\Psi(q) \ , \tag{15}$$

with the directional derivative $\partial_u D\Psi = \sum_k u_k\, \partial_k D\Psi$. This is a single forward mode derivative of $D\Psi$, whereas the general form requires all first derivatives of G.

4 Experiments

We compare the algorithms introduced in Section 2 against their counterparts reformulated in ambient coordinates in Section 3 in three experiments. We use four different benchmark problems, including common problems like Neal's funnel, or a hierarchical Gaussian Process regression (GP) task. Due to our setup, we need to extend some of the benchmark problems to have the shape of a conditional pdf$(q|x)$, rather than a joint distribution, see Section 4.1.

Experiment 1 We compare which implicit solvers are most stable across step sizes and take the least amount of iterations to solve the implicit system, ignoring computation time. We instantiate each benchmark problem in three settings, governing problem difficulty. For each problem instance, we sweep the step size h on a logarithmic grid with 21 points $h \in [10^{-2.5}, 1]$ and perform 1000 trials per choice of h. In each trial we sample q_0 from the problem prior as well as a random momentum $p_0 \sim \mathcal{N}(0, G(q_0))$ and apply the solver once. We then record the number of evaluated positions q as the number of iterations, normalized by step length, as well as the number of cases where the solver does not reach convergence within 200 iterations, or the algorithm was stopped due to divergent behaviour. The implicit solvers compared are given in Table 1. For preconditioning we use the exact Jacobian where computationally sensible for a real application. As the Jacobians of Midpoint joint and Leapfrog require derivatives of G, we replace them with the Jacobian computed at $h = 0$. Finally, we conduct a study on reversibility of the solvers at varying h, see Appendix B.3.

Experiment 2 To obtain a metric for the cost of a single step, we compare runtimes of Secant (Anderson, Newton), LMC (9) and LMC (ambient) (15) to perform a step. We integrate a batch of 100 chains for 5 steps and report the runtime per chain and step. To make results comparable to Experiment 1, we fix $h = \pi/60$, for which the implicit solvers take 2–4 iterations, and report

the average number of iterations together with the runtime. We evaluate the algorithms on the GP problem, where we vary the input dimension $D \in \{10, 20, 30, 50, 100\}$ of the kernel.

Experiment 3 We run the samplers on one instance of each of the four benchmark problems in a realistic implementation with solver code being part of *redacted*[1]. We compare Secant with the preconditioned Picard iteration accelerated by Anderson mixing (Walker & Ni, 2011) with Newton, and LMC (ambient). Every sampler uses the trajectory length $\pi/2$ and a fixed step size, and we run 10 batched chains for 5000 transitions after 500 transitions of warmup. The number of steps per trajectory is tuned per problem and sampler and we record ESS per second and integration step. As a general rule, for the implicit solvers we tune towards an acceptance rate of about 99% with less than 1% of solver failures, so that we tune in a safe region where irreversibility concerns are not an issue. This often aligned with maximizing ESS per solver-iteration. For LMC, we tune to maximize ESS per integration step, under the constraint that less than 1.5% of the transitions are divergent, where a transition is divergent when the change of the energy, including the log Jacobian exceeds 100 in absolute value. While LMC is always reversible, a large amount of rejected points indicates mixing issues in the most difficult parts of the distribution, which skews ESS metrics.

4.1 Benchmark posteriors

For every benchmark problem we state the forward model that defines $\text{pdf}(q \mid x)$.

Funnel. We adapt the funnel of Neal (2003) by adding a noisy observable variable,

$$\begin{aligned} v &= 3q_1 \\ y_i &= e^{v/2}\, q_{i+1} \;, \qquad i = 1, \ldots, 20 \\ x_i &= y_i + \sigma\, \varepsilon_i \;, \end{aligned} \tag{16}$$

with $\sigma = 0.1$ and the observation $x_i = s$ for all i. Experiment 1 uses the three cases $s \in \{0, 1, 5\}$, which target different regions of the funnel, and Experiment 3 uses $s = 1$.

Banana. The banana of Girolami & Calderhead (2011) in $D = 20$ dimensions, rephrased to use the mean of the observations as sufficient statistic,

$$x = 2q_1 + \gamma \sum_{j=2}^{D} 4 f(q_j) + \sigma_y\, \varepsilon \;, \tag{17}$$

with the ridge function $f(q) = q^2$ and two variants $f(q) = q^3$ and $f(q) = e^q - 1$, and $\gamma = 1/\sqrt{(D-1)\operatorname{Var} f/2}$, where $\operatorname{Var} f$ is the variance of f under the standard normal distribution. Experiment 1 uses the three cases $\sigma_y \in \{0.2, 0.05, 0.0125\}$, where smaller values make the posterior narrower and thus the banana shape sharper, and Experiment 3 uses $\sigma_y = 0.05$. The observation is one draw of (17) at $q^* = (0.25, 2^{-1.5}, \ldots, 2^{-1.5})$.

Dust. A real task based on the work that motivated this research. Obtaining the posterior of the dust model of Sarangi et al. (2025) for the supernova SN 2005af, with seven JWST/MIRI AB magnitudes as observation. We model the emission curve as two dust shells with parameters T_i, m_i, v_i representing the temperature, mass and velocity of the dust in the ith shell. The shells differ in their absorption coefficients as they represent carbon and silicate dust, respectively. The published photometric errors are multiplied by a factor s, so that $s = 1$ is the published error model. Experiment 1 uses $s \in \{10, 3, 1\}$ and Experiment 3 uses $s = 1$. For detail, see Appendix B.1.

Hierarchical GP. As posterior, we sample the hyperparameters of a Gaussian process regression with an ARD squared exponential kernel with offset evaluated at points X_i drawn uniformly from $[-1, 1]^D$, $D = 10$, $i = 1, \ldots, 50$, and known noise $\sigma = 0.1$,

$$x = (K + \sigma^2 I)^{1/2}\, \varepsilon \;, \qquad K_{ij} = \exp\Big(-\tfrac{1}{2} \sum_{d=1}^{D} \tfrac{(X_{id} - X_{jd})^2}{\ell_d^2} \Big) + b^2 \;, \tag{18}$$

where we denote with the matrix square root the Cholesky factor, $\ell_d = 3e^{\tau q_d}$, $\tau = \tau_0 \cdot e^{q_\tau}$ and $b = \exp q_b$, so that $n = D + 2$. The observation is $x_i = 1 + \sigma\, \zeta_i$ with $\zeta_i \sim \mathcal{N}(0, 1)$. Experiment 1 uses $\tau_0 \in \{0.25, 0.5, 1.5\}$, and Experiments 2 and 3 use $\tau_0 = 0.5$, where Experiment 2 varies D.

[1] the full code is provided in the supplementary material

Table 2: Experiment 2: runtime in microseconds per chain and step on the GP problem with $m = 50$ observations, for a batch of $N_{\text{batch}} = 100$ chains, $h = \pi/60$. LMC ran out of memory for $n \geq 52$. Timings on an NVIDIA Titan RTX.

	Secant, Anderson		Secant, Newton		LMC	LMC, ambient
n	time (μs)	steps	time (μs)	steps	time (μs)	time (μs)
12	646	3.6	854	2.5	6732	1766
22	762	3.1	1216	2.2	19636	2422
32	937	2.9	1666	2.2	40066	3150
52	1341	2.8	2692	2.1	–	5027
102	2630	2.7	5618	2.1	–	10301

4.2 Results

Experiment 1 Results are shown in Figure 1 for all problems except the three variants of the banana Problem, which are shown in Appendix B.3. On all problems except on the quadratic banana, we can observe that the Midpoint-based variants as well as Generalized Leapfrog perform worse than the algorithms based on the trapezoidal rule. While for small h some variants require a similar number of solver iterations, they all have a smaller region of stability: Secant or RATTLE-based approaches allowed an order of magnitude larger step-size on all problems except GP and the quadratic banana. On the quadratic banana, Midpoint and Secant Midpoint performed identically well and on par with the Secant solvers. This is because in this problem Ψ is quadratic and thus the kinetic terms in both variants are identical and moreover, the computed $\dot{q}^A$ is a valid tangent at the midpoint. In the variants that changed the quadratic term to be cubic or exponential, the performance of Midpoint decreased significantly, falling in line with its performance on the other problems.

The best performing algorithm throughout was Secant (Newton). Notably on the GP problem the solver was still reliable at $h = 1$. When comparing Secant and RATTLE, it becomes clear that both algorithms perform similarly, especially the Picard-based solvers. For the Newton-based solvers, Secant was more stable and also the overall fastest in number of iterations. This was mostly because RATTLE with Newton showed instabilities as the number of solved systems dropped rapidly on the funnel and partially on GP. On the same runs we tested the reversibility of the solves by integrating one step back from $(q_1, -p_1)$. In the region where a solver converges on more than 99% of the draws, ending in a different root is rare, and rarer for Secant than for RATTLE, see Appendix B.3. **Discussion:** The results indicate that among the implicit solvers considered, in our setting, the Secant solvers using the trapezoidal rule are the most robust and efficient and outperform the RATTLE solvers that have the same solution. However, this experiment does not consider the additional time it takes for Newton to compute $D\Psi(q)$ at every solver iteration. We therefore chose both solver variants of Secant for Experiments 2&3.

Experiment 2 In this experiment we compared runtime performance of the two best performing variants, Secant (Anderson, Newton), with LMC. Results indicate that on a per-step basis both implicit solvers compute a step faster than LMC, at all n, but with increasing gap as n increases. Moreover, our results indicate that computing LMC (ambient) using the fast Ω implementation greatly improves its performance in our setting. The difference in runtime between Secant and LMC is a result of requiring second derivatives in the computation of Ω, which is dominating the LMC runtime cost. Note that we have shown our results using batch computations of 100 parallel chains. This skews performance in favour of LMC which requires a relatively expensive graph traversal step through the graph of the first derivative.

Experiment 3 We show end-to-end performance results for the full sampling runs in Figure 2. On the left plot we show the ESS on a per-integration-step basis, where the number of integration steps was tuned as described in Section 4, keeping the integration length fixed to $\pi/2$ across all approaches; the tuned numbers are given above the bars. We found that LMC often needs more integration steps to achieve a low number of divergences. As a result, on a per-integration-step basis, we saw that the two Secant variants outperformed LMC on three out of four problems. Considering ESS/s, the Secant variants outperformed LMC even on GP, and Secant (Anderson) reached almost twice the ESS/s of the next best algorithm on dust and GP. **Discussion:** While the timing results are in-line with the results in **Experiment 2**, it is inherently difficult to interpret ESS/s in this setting,

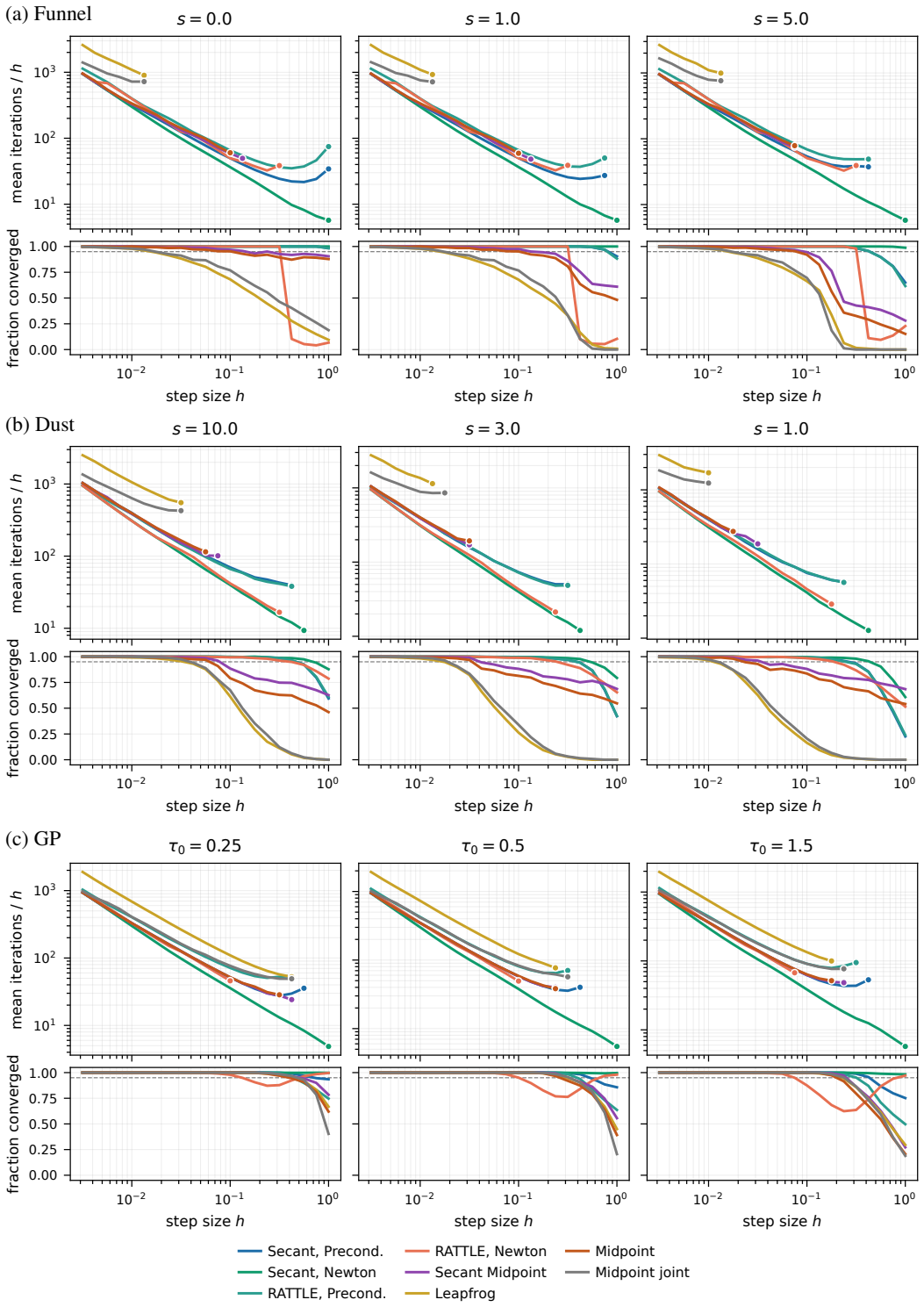


Figure 1: Experiment 1 on (a) the funnel, (b) the dust and (c) the GP: mean number of iterations per unit step length over the converged draws (upper panels) and fraction of converged draws (lower panels) as a function of the step size h, for three values of the difficulty parameter. A curve ends at the last h before the fraction of converged draws first drops below 95 percent.

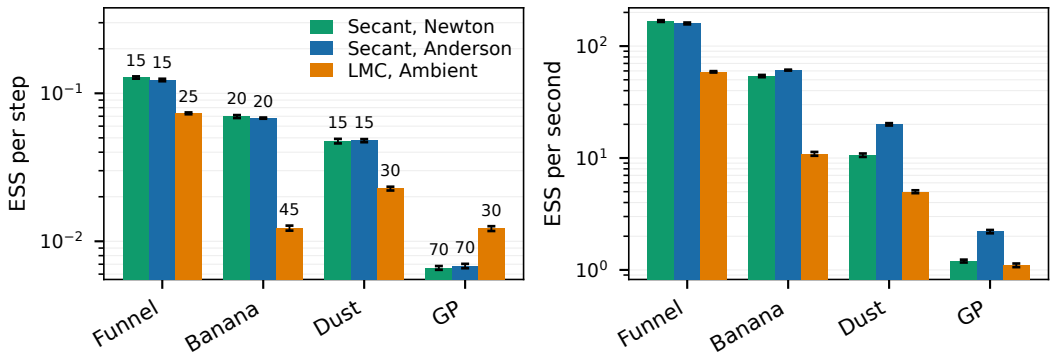


Figure 2: Experiment 3: effective sample size per chain and integration step (left), with the steps per transition above the bars, and effective sample size per second of wall time summed over the 10 chains (right) for the four problems. Error bars are the standard error over chains.

since low dimensionality and high overhead in the python runtime dominate the cost, more so than the actually numerical computations. However, since RMHMC is often used in small dimensional settings using python, these results still indicate real-world performance. Further note, that tuning these algorithms results in different regimes. LMC can be tuned to a far lower acceptance rate, which introduces an additional cost in the ESS metric due to longer correlation times, See Appendix B.5, Table 5 for individual ESS numbers and iteration counts.

5 Conclusion

This article introduced a novel way to derive implicit integrators for RMHMC by rephrasing the kinetic term in ambient coordinates. This construction is useful for a wide variety of hierarchical latent variable models, including variational autoencoders, and other models in which the reparameterisation trick is a bijective transformation. We then conducted a careful set of experiments, comparing the different integrators and solvers in a setting where we control the integrators for trajectory length. Our experiments show that, while our best proposed integrator is equivalent to RATTLE, the derived solver outperforms not only RATTLE, but all previously used implicit integrators for RMHMC, as well as the explicit LMC in effective samples per second. The ambient form of the LMC matrix Ω reduced the runtime of LMC by a factor of thirteen on the GP regression task with $d = 32$ latent variables and reduced the memory requirements significantly.

Our results suggest that implicit midpoint methods, while theoretically interesting, might not allow for large step sizes. We hypothesize this is because the computed slope is not consistent with the geodesic path across the manifold. This is supported by the fact that the only experiment where Midpoint methods performed well was the quadratic banana function, where the slope was consistent. Finally, we suggest tuning implicit integrators for RMHMC with step sizes where the solver is reliable, as we found that in this region irreversibility of the solver was not an issue, and increasing the number of integration steps instead to obtain the desired trajectory length, as proved successful in our experiments.

Reproducibility Statement

The supplementary material contains the code used for the experiments.

REFERENCES

Khai Xiang Au, Matthew M Graham, and Alexandre H Thiery. Manifold lifting: scaling markov chain monte carlo to the vanishing noise regime. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(3):757–782, 2023. doi: 10.1093/jrsssb/qkad023.

Michael Betancourt. A general metric for Riemannian Manifold Hamiltonian Monte Carlo. In *Geometric Science of Information*, volume 8085 of *Lecture Notes in Computer Science*, pp. 327–334. Springer, 2013.

Michael Betancourt and Mark Girolami. Hamiltonian monte carlo for hierarchical models. In *Current trends in Bayesian methodology with applications*, pp. 119–142. Chapman and Hall/CRC, 2019.

Nawaf Bou-Rabee, Bob Carpenter, Tore Selland Kleppe, and Sifan Liu. The within-orbit adaptive leapfrog no-u-turn sampler. *Journal of Machine Learning Research*, 27(113):1–64, 2026a. URL `http://jmlr.org/papers/v27/25-1452.html`.

Nawaf Bou-Rabee, Bob Carpenter, and Milo Marsden. GIST: Gibbs self-tuning for locally adaptive Hamiltonian Monte Carlo. *Statistics Surveys*, 20:135–179, 2026b. doi: 10.1214/26-SS156.

James Brofos and Roy R Lederman. Evaluating the implicit midpoint integrator for Riemannian Hamiltonian Monte Carlo. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1072–1081. PMLR, 2021a. URL `https://proceedings.mlr.press/v139/brofos21a.html`.

James A Brofos and Roy R Lederman. On numerical considerations for Riemannian Manifold Hamiltonian Monte Carlo. *arXiv preprint arXiv:2111.09995*, 2021b.

James A Brofos and Roy R Lederman. Several remarks on the numerical integrator in Lagrangian Monte Carlo. *arXiv preprint arXiv:2202.13888*, 2022.

Marcus Brubaker, Mathieu Salzmann, and Raquel Urtasun. A family of MCMC methods on implicitly defined manifolds. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 161–172. PMLR, 2012.

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

Matthew M Graham and Amos J Storkey. Asymptotically exact inference in differentiable generative models. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 499–508. PMLR, 2017.

Marcelo Hartmann, Mark Girolami, and Arto Klami. Lagrangian manifold Monte Carlo on Monge patches. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4764–4781. PMLR, 2022.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo, 2011.

Miika Kailas, Matti Vihola, and Jonas Wallin. Adaptive Riemannian Manifold Hamiltonian Monte Carlo with hierarchical metric. *arXiv preprint arXiv:2604.09832*, 2026.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Tore Selland Kleppe. Log-density gradient covariance and automatic metric tensors for Riemann manifold Monte Carlo methods. *Scandinavian Journal of Statistics*, 51(3), 2024. doi: 10.1111/sjos.12705.

Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, 4(33): 1143, 2019. doi: 10.21105/joss.01143.

Shiwei Lan, Vasileios Stathopoulos, Babak Shahbaba, and Mark Girolami. Markov Chain Monte Carlo from Lagrangian dynamics. *Journal of Computational and Graphical Statistics*, 24(2): 357–378, 2015.

Tony Lelièvre, Mathias Rousset, and Gabriel Stoltz. Hybrid Monte Carlo methods for sampling probability measures on submanifolds. *Numerische Mathematik*, 143(2):379–421, 2019.

Jerrold E Marsden and Matthew West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.

Chirag Modi, Alex Barnett, and Bob Carpenter. Delayed rejection hamiltonian monte carlo for sampling multiscale distributions. *Bayesian Analysis*, 19(3):815–842, 2024.

Radford M Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

Radford M Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L Jones, and Xiao-Li Meng (eds.), *Handbook of Markov Chain Monte Carlo*, chapter 5, pp. 113–162. Chapman and Hall/CRC, 2011.

Arkaprabha Sarangi, Szanna Zsíros, Tamás Szalai, Laureano Martinez, Melissa Shahbandeh, Ori D. Fox, Schuyler D. Van Dyk, Alexei V. Filippenko, Melina Cecilia Bersten, Ilse De Looze, Chris Ashall, Tea Temim, Jacob E. Jencson, Armin Rest, Dan Milisavljevic, Luc Dessart, Eli Dwek, Nathan Smith, Samaporn Tinyanont, Thomas G. Brink, WeiKang Zheng, Geoffrey C. Clayton, and Jennifer Andrews. Two decades of dust evolution in SN 2005af through JWST, Spitzer, and chemical modeling. *The Astrophysical Journal*, 993(1):94, November 2025. doi: 10.3847/1538-4357/ae0645.

Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and to assist in the derivation of mathematical statements. Additionally, we used generative AI tools to create and edit software code and to draft and edit parts of the paper. The core idea of rephrasing the Lagrangian in ambient coordinates is the authors alone as is the design of the experiments and the choice of benchmark problems. We have reviewed all AI-assisted work. All derivations were checked by the authors, the code was tested against independent implementations of the same integrators, and every cited claim was verified against the cited source. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

A DISCRETE ACTION OF THE GENERALIZED LEAPFROG

The Generalized Leapfrog (8) is generated by the trapezoidal rule applied to the Lagrangian (4) with a momentum that is shared by both end points,

$$\hat{S}^{\mathrm{GL}}(q_0, q_1) = \tfrac{h}{2}\Big[L\big(q_0, G(q_0)^{-1}p_{1/2}\big) + L\big(q_1, G(q_1)^{-1}p_{1/2}\big)\Big] \ , \tag{19}$$

where $p_{1/2}$ is the solution of the equation for q_1 in (8) for given q_0 and q_1. With

$$M := G(q_0)^{-1} + G(q_1)^{-1} \tag{20}$$

and $\nabla_p H(q,p) = G(q)^{-1}p$ from (3), the equation for q_1 in (8) reads

$$q_1 - q_0 = \tfrac{h}{2}\, M\, p_{1/2} \ , \tag{21}$$

so that

$$p_{1/2} = \tfrac{2}{h}\, M^{-1}(q_1 - q_0) \ . \tag{22}$$

We show that $\hat{S}^{\mathrm{GL}}$ generates (8) through the relations $p_0 = -\nabla_{q_0}\hat{S}^{\mathrm{GL}}$ and $p_1 = \nabla_{q_1}\hat{S}^{\mathrm{GL}}$ of (5).

Closed form. Inserting $\dot{q} = G(q)^{-1}p_{1/2}$ into (4) gives for each end point

$$L\big(q, G(q)^{-1}p_{1/2}\big) = \tfrac{1}{2}\, p_{1/2}^T G(q)^{-1} p_{1/2} - V(q) \ . \tag{23}$$

The two kinetic terms of (19) therefore sum to

$$\tfrac{h}{2} \cdot \tfrac{1}{2}\, p_{1/2}^T \big[G(q_0)^{-1} + G(q_1)^{-1}\big] p_{1/2} = \tfrac{h}{4}\, p_{1/2}^T M\, p_{1/2} \ , \tag{24}$$

and with (22)

$$\tfrac{h}{4}\, p_{1/2}^T M\, p_{1/2} = \tfrac{h}{4} \cdot \tfrac{4}{h^2}\, (q_1 - q_0)^T M^{-1} M M^{-1} (q_1 - q_0) = \tfrac{1}{h}\, (q_1 - q_0)^T M^{-1} (q_1 - q_0) \ . \tag{25}$$

Hence

$$\hat{S}^{\mathrm{GL}}(q_0, q_1) = \tfrac{1}{h}\, (q_1 - q_0)^T M^{-1} (q_1 - q_0) - \tfrac{h}{2}\big(V(q_0) + V(q_1)\big) \ . \tag{26}$$

Gradient with respect to q_0. The kinetic term of (26) depends on q_0 through the difference $q_1 - q_0$ and through M. Holding M fixed, the derivative through the difference is

$$\nabla_{q_0} \Big[\tfrac{1}{h}\, (q_1 - q_0)^T M^{-1} (q_1 - q_0) \Big]_{M \text{ fixed}} = -\tfrac{2}{h}\, M^{-1} (q_1 - q_0) = -p_{1/2} \ . \tag{27}$$

For the derivative through M, the k-th component is

$$\tfrac{1}{h}\, (q_1 - q_0)^T \big(\partial_k M^{-1}\big)(q_1 - q_0) = -\tfrac{1}{h}\, (q_1 - q_0)^T M^{-1} \big(\partial_k M\big) M^{-1} (q_1 - q_0) = -\tfrac{h}{4}\, p_{1/2}^T \big(\partial_k M\big) p_{1/2} \ , \tag{28}$$

where ∂_k denotes the derivative with respect to the k-th component of q_0, the first equality is the derivative of the matrix inverse and the second uses (22). Only the first summand of M depends on q_0, and by the derivative of the inverse again

$$\partial_k M = \partial_k G(q_0)^{-1} = -G(q_0)^{-1} \big(\partial_k G(q_0)\big) G(q_0)^{-1} \ . \tag{29}$$

Inserting (29) into (28) gives

$$\tfrac{h}{4}\, \big(G(q_0)^{-1} p_{1/2}\big)^T \big(\partial_k G(q_0)\big) \big(G(q_0)^{-1} p_{1/2}\big) = \tfrac{h}{2}\, \Gamma\big(q_0, G(q_0)^{-1} p_{1/2}\big)_k \ , \tag{30}$$

by the definition $\Gamma(q, u) = \frac{1}{2} \nabla_q [u^T G(q) u]$ of Section 2.1. The potential term of (26) contributes $-\frac{h}{2} \nabla V(q_0)$. Collecting (27), (30) and the potential term,

$$\nabla_{q_0} \hat{S}^{\mathrm{GL}}(q_0, q_1) = -p_{1/2} + \tfrac{h}{2}\, \Gamma\big(q_0, G(q_0)^{-1} p_{1/2}\big) - \tfrac{h}{2}\, \nabla V(q_0) \ . \tag{31}$$

Gradient with respect to q_1. The same computation for q_1 differs only in the sign of the derivative through the difference, which is $+p_{1/2}$, and in the second summand of M taking the place of the first,

$$\nabla_{q_1} \hat{S}^{\mathrm{GL}}(q_0, q_1) = p_{1/2} + \tfrac{h}{2}\, \Gamma\big(q_1, G(q_1)^{-1} p_{1/2}\big) - \tfrac{h}{2}\, \nabla V(q_1) \ . \tag{32}$$

The momentum equations. From (3), the position gradient of the Hamiltonian is

$$\nabla_q H(q, p) = \nabla V(q) - \Gamma\big(q, G(q)^{-1} p\big) \ . \tag{33}$$

Applying the relations of (5) to (31) and (32) and using (33) gives

$$p_0 = -\nabla_{q_0} \hat{S}^{\mathrm{GL}}(q_0, q_1) = p_{1/2} + \tfrac{h}{2}\, \nabla_q H(q_0, p_{1/2}) \ , \tag{34}$$

$$p_1 = \nabla_{q_1} \hat{S}^{\mathrm{GL}}(q_0, q_1) = p_{1/2} - \tfrac{h}{2}\, \nabla_q H(q_1, p_{1/2}) \ . \tag{35}$$

Equation (34) is the first equation of (8), (21) is the second and (35) is the third. In (34), q_1 enters only through $p_{1/2}$. The Generalized Leapfrog therefore first solves (34) for $p_{1/2}$ and then (21) for q_1, after which (35) is explicit.

B EXPERIMENTAL DETAILS

B.1 DUST POSTERIOR PROBLEM

$q \in \mathbb{R}^6$. For the two dust shells $i \in \{1, 2\}$, carbon and silicate, the mass, the temperature and the expansion velocity are

$$m_i = 1.98892 \cdot 10^{33} \cdot 10^{-1.5+0.75\, q_{3i-2}} \ , \tag{36}$$

$$T_i = 10^{2+0.4\, q_{3i-1}} \ , \tag{37}$$

$$v_i = 10^5 \cdot 10^{3.05+0.2\, q_{3i}} \ . \tag{38}$$

The observation $x \in \mathbb{R}^7$ holds the AB magnitudes in the seven bands with centre wavelengths λ_j, photometric errors e_j and absorption coefficients $\kappa_{1,j}$, $\kappa_{2,j}$ of Table 3. The forward model is

$$x_j = \mathrm{mag}_j(q) + \sigma_j\, \varepsilon_j \ , \qquad \sigma_j^2 = (s\, e_j)^2 + 0.02^2 \ , \tag{39}$$

$$\mathrm{mag}_j(q) = -2.5 \log_{10}\big(F_{1,j}(m_1, T_1, v_1) + F_{2,j}(m_2, T_2, v_2)\big) + 76.80 \ , \tag{40}$$

$$F_{i,j}(m, T, v) = m\, \kappa_{i,j}\, B(\lambda_j, T)\, P\Big(\frac{3\, m\, \kappa_{i,j}}{4\pi\, (vt)^2}\Big) \ , \tag{41}$$

$$B(\lambda, T) = \frac{2 h_{\mathrm{P}} c_0}{\lambda^3}\, \frac{1}{\exp\big(h_{\mathrm{P}} c_0/(\lambda k_B T)\big) - 1} \ , \tag{42}$$

$$P(\tau) = \frac{3}{4\tau}\Big[1 - \frac{1}{2\tau^2} + \Big(\frac{1}{\tau} + \frac{1}{2\tau^2}\Big) e^{-2\tau}\Big] \ , \tag{43}$$

with $h_{\mathrm{P}} = 6.62607015 \cdot 10^{-27}$, $k_B = 1.380649 \cdot 10^{-16}$, $c_0 = 2.99792458 \cdot 10^{10}$ and the age $t = 5.8977 \cdot 10^8$ of the supernova, all in cgs units, and λ_j in cm. The additive constant in (40) collects the distance of the supernova and the zero point of the AB magnitude. Equations (41) to (43) are Equations 1 to 3 of Sarangi et al. (2025), and the observation and its errors are from their Table 3. Experiment 1 uses $s \in \{10, 3, 1\}$, Experiment 3 $s = 1$.

Table 3: Bands, observation, photometric errors and absorption coefficients of the dust problem.

λ_j (10^{-4} cm)	10.0	11.3	12.8	15.0	18.0	21.0	25.5
x_j	22.01	20.80	19.80	18.89	18.32	17.95	16.93
e_j	0.10	0.10	0.03	0.01	0.02	0.02	0.03
$\kappa_{1,j}$	989.67	862.96	733.44	576.36	452.86	371.51	279.01
$\kappa_{2,j}$	2827.45	1715.34	944.28	724.17	1096.13	870.35	572.64

B.2 SOLVERS

Update rules. The Picard solvers iterate $q \leftarrow q - P^{-1}F(q)$ on the residual F of their equation with the preconditioner P of Table 1 held fixed over the iterations, and Midpoint joint and Leapfrog with $P = I$. Newton's method uses the Jacobian at the current iterate. RATTLE iterates on the multiplier λ with the residual $c(q_1^A)$.

Stopping conditions. When solving for q_1, solvers stop when the residual F_q of their equation satisfies $\|L_0^{-1}F\|_\infty < 10^{-7}$ with $G(q_0) = L_0 L_0^T$. The momentum solve of the Generalized Leapfrog uses the dual norm $\|L_0^{-1}F_p\|_\infty$, and its position solve and the Midpoint joint use $\|L_0^T F_q\|_\infty$. RATTLE measures its constraint violation c as the minimum norm ambient correction $\|Dc(\Psi(q_0))^T(Dc(\Psi(q_0))\, Dc(\Psi(q_0))^T)^{-1}c\|_\infty$, so that the tolerance has the same units as for the chart solvers. The iteration cap is 200 in Experiment 1 and 100 in Experiment 3. A solve counts as diverged when its residual exceeds 1000 times its reference value, which is the residual at the start point for the chart solvers and the length of the free step for RATTLE.

Iteration count. Across solvers we count as iterations the number of q_1 points evaluated, including the initialization $q_1 = q_0$. We do not count the final momentum evaluation p_1 that evaluates the final $V(p_1)$, since in real-world implementations this endpoint is reused as the starting point q_0 in the following iteration.

B.3 Experiment 1 and the reversibility test

Figure 3 shows Experiment 1 on the banana with the ridge functions q^2, q^3 and $e^q - 1$. Table 4 gives the results of the reversibility test for every problem at its hardest setting.

The reversibility test is run on the draws of Experiment 1 at the grid points $h = 10^{-2}$, $10^{-1.5}$ and 10^{-1}. For every draw whose forward solve converged, the same algorithm integrates one step from $(q_1, -p_1)$ to (q_2, p_2). The draw is irreversible if the backward solve does not converge or if $\|\Psi(q_2) - \Psi(q_0)\|_2 > 0.1\,\|\Psi(q_1) - \Psi(q_0)\|_2$ in the ambient coordinates $\Psi(q) = (q, \psi_q(x))$, a criterion that depends on neither the chart nor the base point. The criterion involves only the position, since for a symmetric discrete action $p_2 = -p_0$ follows from $q_2 = q_0$. Over the remaining draws the 95th percentile of $\max(\|L_0^T(q_2 - q_0)\|_\infty, \|L_0^{-1}(p_2 + p_0)\|_\infty)$ is reported, which measures the error in position and momentum that the tolerance itself produces.

Results of the reversibility test. The most important observation is that for all solvers considered, regions in which solver stability is higher, i.e., more than 99% successful solves, irreversibility through selecting the wrong root is a rare occurrence. This holds especially for the Secant (Precond., Newton) variants which are more robust than RATTLE in this regard. **Discussion:** The reduced robustness of RATTLE is likely due to it working in ambient space. Since we initialize $\lambda = 0$ the initial step can lead the solver far away from the manifold, which makes it more likely to enter regions converging to different roots. The Secant-based solver in contrast never allows q_1 to leave the manifold. Moreover, the general robustness suggests that the strategy proposed by Lelièvre et al. (2019) to perform a forward and a backward solve to ensure reversibility might not be necessary for all applications. This is because since choosing step sizes beyond the high acceptance region, not only the number of rejected points increases, but the number of iterations per solve increases as well. This makes these tuning regimes less interesting in terms of maximizing sampling speed.

B.4 Experiment 2

The chains start from independent prior draws. The time of one transition of the batch, after one untimed transition for allocations, is divided by the number of steps and by the batch size. The generic Ω of (9) is built from n reverse passes through G per end point. Timings were taken on an NVIDIA Titan RTX with torch 2.7.1 and CUDA 11.8, with all chains in one batch on one device.

B.5 Experiment 3

The chains start from independent prior draws. A transition with a non-finite energy change is divergent as well, and for the implicit algorithms a step whose solver residual exceeds the tolerance gives an infinite energy change, so that a failed solve is rejected and counted as divergent.

The effective sample size is the bulk effective sample size of ArviZ (Kumar et al., 2019), and the tail effective sample size is that of the 5% and 95% quantiles, both computed per chain and per coordinate with the minimum over coordinates taken per chain. Tables report the mean over chains plus or minus the standard error of the mean. The effective sample size per iteration divides a chain's effective sample size by its total number of solver iterations over the 5000 transitions, counted as in Appendix B.2, and the iterations column is the mean number of solver iterations per transition. The effective sample size per second is the sum over chains divided by the wall time of the warmup and the 5000 transitions. The funnel, the banana and the dust were sampled on the CPU and the GP on the NVIDIA Titan RTX. Table 5 gives all columns for every problem.

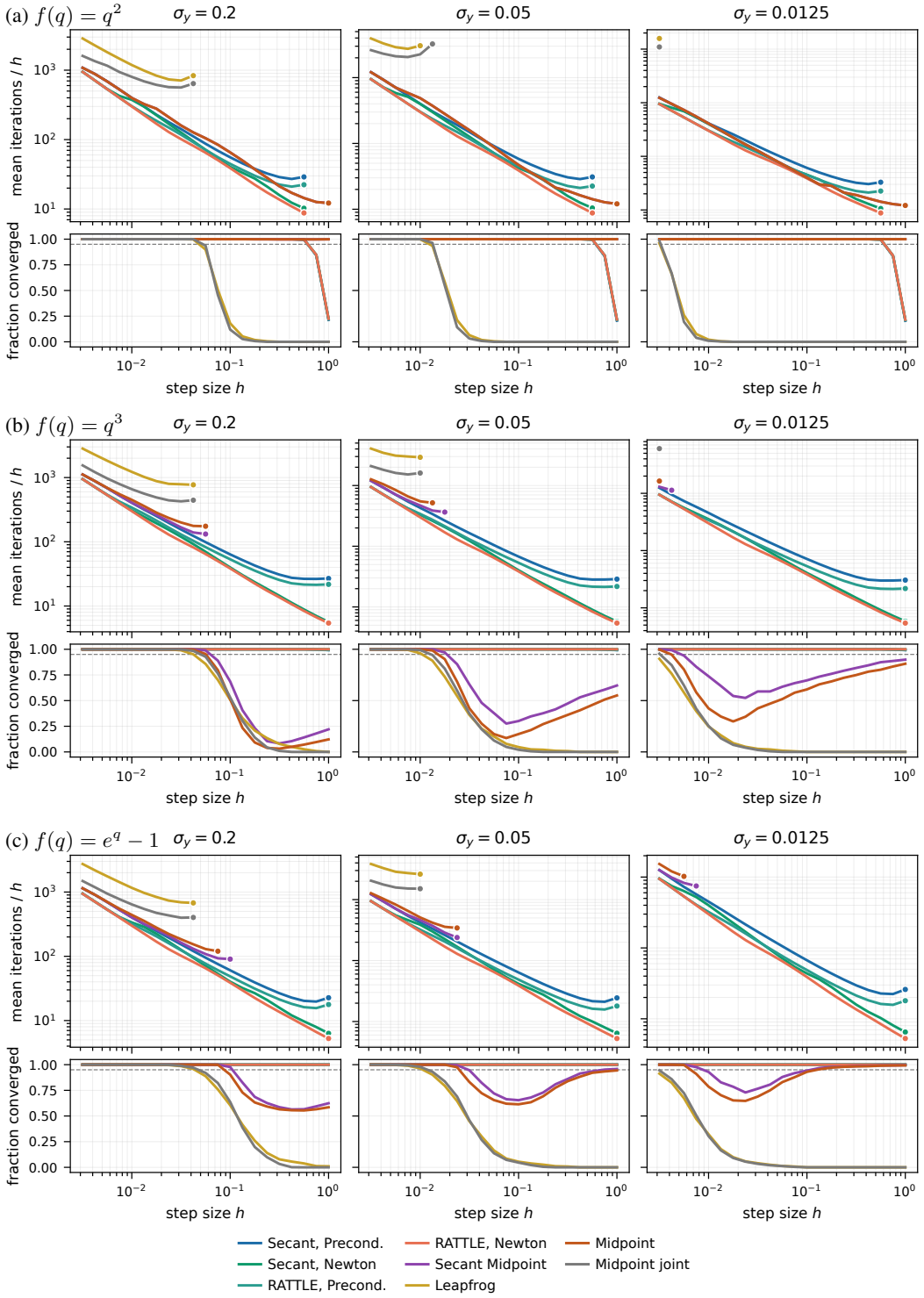


Figure 3: Experiment 1 on the banana with the ridge functions (a) q^2, (b) q^3 and (c) $e^q - 1$, as Figure 1.

Table 4: Reversibility of one forward–backward step in Experiment 1 over $N = 1000$ prior draws, for every problem at its hardest setting. conv. is the number of draws whose forward solve converges, irrev. the number of those whose backward solve fails or returns to a point farther than a tenth of the step from the start in ambient Euclidean distance, and 95% the 95th percentile of the error $\max(\|L_0^T(q_2 - q_0)\|_\infty, \|L_0^{-1}(p_2 + p_0)\|_\infty)$ over the remaining draws, at solver tolerance 10^{-7}.

	$h = 0.01$			$h = 0.0316$			$h = 0.1$		
Algorithm	conv.	irrev.	95%	conv.	irrev.	95%	conv.	irrev.	95%
Funnel, $s = 5$									
Secant, Precond.	1000	0	1e-09	1000	0	4e-08	1000	0	5e-08
Secant, Newton	1000	0	1e-11	1000	0	4e-12	1000	0	2e-12
RATTLE, Precond.	1000	0	1e-09	1000	0	4e-08	1000	0	1e-07
RATTLE, Newton	1000	0	2e-12	1000	0	2e-12	1000	0	7e-12
Secant Midpoint	1000	1	2e-09	991	0	5e-08	945	14	1e-07
Leapfrog	978	0	3e-07	887	1	6e-07	663	1	8e-07
Midpoint	998	0	1e-07	980	2	1e-07	920	11	2e-07
Midpoint joint	980	0	2e-05	915	0	1e-05	695	0	8e-06
Banana, $\sigma_y = 0.0125$									
Secant, Precond.	1000	0	9e-09	1000	0	5e-09	1000	0	9e-09
Secant, Newton	1000	0	7e-11	1000	0	2e-11	1000	0	1e-10
RATTLE, Precond.	1000	0	1e-07	1000	0	2e-07	1000	0	1e-06
RATTLE, Newton	1000	0	6e-11	1000	0	2e-10	1000	0	5e-10
Secant Midpoint	1000	0	1e-08	1000	0	1e-09	1000	0	6e-10
Leapfrog	21	0	2e-06	0	–	–	0	–	–
Midpoint	1000	0	1e-08	1000	0	1e-09	1000	0	6e-10
Midpoint joint	10	0	5e-06	0	–	–	0	–	–
Dust, $s = 1$									
Secant, Precond.	1000	0	2e-07	996	1	2e-07	991	2	3e-07
Secant, Newton	1000	0	9e-10	997	2	3e-10	996	0	3e-10
RATTLE, Precond.	1000	0	3e-08	997	2	2e-07	990	3	7e-07
RATTLE, Newton	1000	0	4e-11	996	1	4e-11	985	3	5e-11
Secant Midpoint	994	3	1e-07	962	11	5e-07	881	26	4e-07
Leapfrog	964	1	2e-06	627	7	4e-06	164	3	3e-06
Midpoint	988	0	3e-07	929	8	9e-07	836	13	5e-07
Midpoint joint	972	0	3e-05	671	2	2e-05	205	4	7e-06
GP, $\tau_0 = 1.5$									
Secant, Precond.	1000	0	1e-07	1000	0	1e-07	1000	0	2e-07
Secant, Newton	1000	0	2e-11	1000	0	7e-12	1000	0	2e-12
RATTLE, Precond.	1000	0	5e-09	1000	0	2e-08	1000	2	9e-08
RATTLE, Newton	1000	0	4e-13	998	0	4e-13	875	16	4e-13
Secant Midpoint	1000	0	1e-07	1000	0	1e-07	999	0	2e-07
Leapfrog	1000	0	7e-10	1000	0	5e-09	998	0	3e-08
Midpoint	1000	0	1e-07	1000	0	1e-07	998	0	2e-07
Midpoint joint	1000	0	3e-07	1000	0	4e-07	999	0	8e-07

Table 5: Experiment 3, all columns: steps per transition, bulk and tail effective sample size per chain, effective sample size per solver iteration, solver iterations per transition, acceptance rate, number of divergent transitions and effective sample size per second, as defined in Appendix B.5. Mean $\pm$ standard error over the 10 chains where applicable.

Algorithm	N_{steps}	ESS	tail ESS	ESS/iter	iters	acc	div	ESS/s
Funnel, $s = 1$, $n = 21$								
Secant, Newton	15	9590 ± 170	3976 ± 84	0.0479 ± 0.0008	40.0 ± 0.0	0.995 ± 0.000	0	167.5
Secant, Anderson	15	9228 ± 176	3992 ± 64	0.0345 ± 0.0007	53.5 ± 0.0	0.995 ± 0.001	0	159.4
LMC, ambient	25	9155 ± 130	3989 ± 52	0.0732 ± 0.0010	25.0 ± 0.0	0.996 ± 0.000	0	58.9
Banana, $\sigma_y = 0.05$, $n = 20$								
Secant, Newton	20	6967 ± 159	2328 ± 68	0.0211 ± 0.0005	66.1 ± 0.0	0.994 ± 0.000	0	54.0
Secant, Anderson	20	6798 ± 63	2408 ± 57	0.0149 ± 0.0001	91.1 ± 0.1	0.994 ± 0.000	0	61.0
LMC, ambient	45	2764 ± 108	1090 ± 39	0.0123 ± 0.0005	45.0 ± 0.0	0.678 ± 0.003	0	10.9
Dust, $s = 1$, $n = 6$								
Secant, Newton	15	3566 ± 125	3564 ± 69	0.0163 ± 0.0006	43.9 ± 0.1	0.990 ± 0.000	0	10.6
Secant, Anderson	15	3592 ± 86	3662 ± 122	0.0116 ± 0.0003	62.1 ± 0.2	0.989 ± 0.001	0	20.0
LMC, ambient	30	3406 ± 100	3134 ± 108	0.0227 ± 0.0007	30.0 ± 0.0	0.958 ± 0.002	0	5.0
GP, $\tau_0 = 0.5$, $n = 12$								
Secant, Newton	70	2314 ± 68	1223 ± 41	0.0024 ± 0.0001	192.3 ± 0.1	0.990 ± 0.001	1	1.2
Secant, Anderson	70	2383 ± 81	1108 ± 37	0.0018 ± 0.0001	267.3 ± 0.1	0.990 ± 0.000	2	2.2
LMC, ambient	30	1829 ± 68	1000 ± 37	0.0122 ± 0.0005	30.0 ± 0.0	0.865 ± 0.002	585	1.1