JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2018, VOL. 113, NO. 522, 855–867, Theory and Methods
https://doi.org/10.1080/01621459.2017.1294075

Taylor & Francis
Taylor & Francis Group

The Bouncy Particle Sampler: A Nonreversible Rejection-Free Markov Chain Monte Carlo Method

Alexandre Bouchard-Côté[a], Sebastian J. Vollmer[b,c], and Arnaud Doucet[d]

[a]Department of Statistics, University of British Columbia, Vancouver, Canada; [b]Mathematics Institute and Department of Statistics, University of Warwick, Coventry, United Kingdom; [c]Alan Turing Institute, London, United Kingdom; [d]Department of Statistics, University of Oxford, Oxford, United Kingdom

ABSTRACT

Many Markov chain Monte Carlo techniques currently available rely on discrete-time reversible Markov processes whose transition kernels are variations of the Metropolis–Hastings algorithm. We explore and generalize an alternative scheme recently introduced in the physics literature (Peters and de With 2012) where the target distribution is explored using a continuous-time nonreversible piecewise-deterministic Markov process. In the Metropolis–Hastings algorithm, a trial move to a region of lower target density, equivalently of higher "energy," than the current state can be rejected with positive probability. In this alternative approach, a particle moves along straight lines around the space and, when facing a high energy barrier, it is not rejected but its path is modified by bouncing against this barrier. By reformulating this algorithm using inhomogeneous Poisson processes, we exploit standard sampling techniques to simulate exactly this Markov process in a wide range of scenarios of interest. Additionally, when the target distribution is given by a product of factors dependent only on subsets of the state variables, such as the posterior distribution associated with a probabilistic graphical model, this method can be modified to take advantage of this structure by allowing computationally cheaper "local" bounces, which only involve the state variables associated with a factor, while the other state variables keep on evolving. In this context, by leveraging techniques from chemical kinetics, we propose several computationally efficient implementations. Experimentally, this new class of Markov chain Monte Carlo schemes compares favorably to state-of-the-art methods on various Bayesian inference tasks, including for high-dimensional models and large datasets. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received February 2016
Revised October 2016

KEYWORDS
Inhomogeneous Poisson process; Markov chain Monte Carlo; Piecewise deterministic Markov process; Probabilistic graphical models; Rejection-free simulation

1. Introduction

Markov chain Monte Carlo (MCMC) methods are standard tools to sample from complex high-dimensional probability measures. Many MCMC schemes available at present are based on the Metropolis-Hastings (MH) algorithm and their efficiency is strongly dependent on the ability of the user to design proposal distributions capturing the main features of the target distribution; see Liu (2008) for a comprehensive review. We examine, analyze, and generalize here a different approach to sample from distributions on $\mathbb{R}^d$ that has been recently proposed in the physics literature (Peters and de With 2012). Let the energy be defined as the negative logarithm of an unnormalized version of the target density. In this methodology, a particle explores the space by moving along straight lines and, when it faces a high energy barrier, it bounces against the contour lines of this energy. This nonreversible rejection-free MCMC method will be henceforth referred to as the bouncy particle sampler (BPS). This algorithm and closely related schemes have already been adopted to simulate complex physical systems such as hard spheres, polymers, and spin models (Michel, Kapfer, and Krauth 2014; Kampmann, Boltz, and Kierfeld 2015; Michel, Mayer, and Krauth 2015; Nishikawa, Michel, and Krauth 2015). For these models, it has been demonstrated experimentally that such methods can outperform state-of-the-art MCMC methods by up to several orders of magnitude.

However, the implementation of the BPS proposed in Peters and de With (2012) is not applicable to most target distributions arising in statistics. In this article, we make the following contributions:

Simulation schemes based on inhomogeneous Poisson processes: By reformulating explicitly the bounces times of the BPS as the first arrival times of inhomogeneous Poisson processes (PP), we leverage standard sampling techniques (Devroye 1986, chap. 6) and methods from chemical kinetics (Thanh and Priami 2015) to obtain new computationally efficient ways to simulate the BPS process for a large class of target distributions.

Factor graphs: When the target distribution can be expressed as a factor graph (Wainwright and Jordan 2008), a representation generalizing graphical models where the target is given by a product of factors and each factor can be a function of only a subset of variables, we adapt a physical multi-particle system

CONTACT Sebastian J. Vollmer svollmer@turing.ac.uk Mathematics Institute and Department of Statistics, University of Warwick, Coventry CV4 7AL, United Kingdom.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JASA.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

method discussed in Peters and de With (2012, sec. III) to achieve additional computational efficiency. This local version of the BPS only manipulates a restricted subset of the state components at each bounce but results in a change of all state components, not just the one being updated contrary to the Gibbs sampler.

Ergodicity analysis: We present a proof of the ergodicity of BPS when the velocity of the particle is additionally refreshed at the arrival times of a homogeneous PP. When this refreshment step is not carried out, we exhibit a counter-example where ergodicity does not hold.

Efficient refreshment: We propose alternative refreshment schemes and compare their computational efficiency experimentally.

Empirically, these new MCMC schemes compare favorably to state-of-the-art MCMC methods on various Bayesian inference problems, including for high-dimensional scenarios and large datasets. Several additional original extensions of the BPS including versions of the algorithm and distributions restricted to a compact support can be found in Bouchard-Côté, Vollmer, and Doucet (2015). For brevity, these are not discussed here.

The rest of this article is organized as follows. In Section 2, we introduce the basic version of the BPS, propose original ways to implement it, and prove its ergodicity under weak assumptions. Section 3 presents a modification of the basic BPS, which exploits a factor graph representation of the target distribution and develops computationally efficient implementations of this scheme. In Section 4, we demonstrate this methodology on various Bayesian models. The proofs are given in the the supplementary material available online.

2. The Bouncy Particle Sampler

2.1. Problem Statement and Notation

Consider a probability distribution π on $\mathbb{R}^d$, equipped with the Borel σ-algebra $\mathcal{B}(\mathbb{R}^d)$. We assume that π admits a probability density with respect to the Lebesgue measure dx and slightly abuse notation by denoting also this density by π. In most practical scenarios, we only have access to an unnormalized version of this density, that is,

$$\pi(x) = \frac{\gamma(x)}{\mathcal{Z}},$$

where $\gamma : \mathbb{R}^d \to (0, \infty)$ can be evaluated pointwise but the normalizing constant $\mathcal{Z} = \int_{\mathbb{R}^d} \gamma(x)dx$ is unknown. We call

$$U(x) = -\log \gamma(x)$$

the associated energy, which is assumed continuously differentiable, and we denote by $\nabla U(x) = (\frac{\partial U(x)}{\partial x_1}, \ldots, \frac{\partial U(x)}{\partial x_d})^\top$ the gradient of U evaluated at x. We are interested in approximating numerically the expectation of arbitrary test functions $\varphi : \mathbb{R}^d \to \mathbb{R}$ with respect to π.

2.2. Algorithm Description

The BPS methodology introduced in Peters and de With (2012) simulates a continuous piecewise linear trajectory $\{x(t)\}_{t\geq 0}$ in $\mathbb{R}^d$. It has been informally derived as a continuous-time limit of the Metropolis algorithm in Peters and de With (2012). Each segment in the trajectory is specified by an initial position $x^{(i)} \in \mathbb{R}^d$, a length $\tau_{i+1} \in \mathbb{R}^+$, and a velocity $v^{(i)} \in \mathbb{R}^d$ (example shown in Figure 1, left). We denote the times where the velocity changes by $t_i = \sum_{j=1}^{i} \tau_j$ for $i \geq 1$, and set $t_0 = 0$ for convenience. The position at time $t \in [t_i, t_{i+1})$ is thus interpolated linearly, $x(t) = x^{(i)} + v^{(i)}(t - t_i)$, and each segment is connected to the next, $x^{(i+1)} = x^{(i)} + v^{(i)}\tau_{i+1}$. The length of these segments is governed by an inhomogeneous PP of intensity function $\lambda : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$

$$\lambda(x, v) = \max\{0, \langle \nabla U(x), v \rangle\}. \tag{1}$$

When the particle bounces, its velocity is updated in the same way as a Newtonian elastic collision on the hyperplane tangential to the gradient of the energy. Formally, the velocity after bouncing is given by

$$\begin{aligned} R(x)v &= \left(I_d - 2\frac{\nabla U(x)\{\nabla U(x)\}^\top}{\|\nabla U(x)\|^2}\right)v \\ &= v - 2\frac{\langle \nabla U(x), v\rangle}{\|\nabla U(x)\|^2}\nabla U(x), \end{aligned} \tag{2}$$

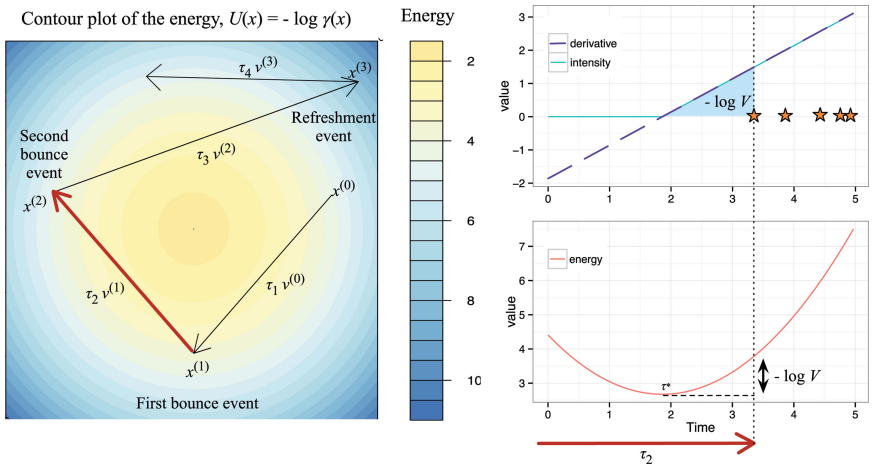

Figure 1. Illustration of BPS on a standard bivariate Gaussian distribution. Left and top right: see Section 2.2; bottom right: see Example 1. The stars represent a realization of an inhomogeneous Poisson process with intensity given by the solid line.

where I_d denotes the $d \times d$ identity matrix, $\|\cdot\|$ the Euclidean norm, and $\langle w, z\rangle = w^t z$ the scalar product between column vectors w, z. (Computations of the form $R(x)v$ are implemented via the right-hand side of Equation (2), which takes time $O(d)$ rather than the left-hand side, which would take time $O(d^2)$.) Peters and de With (2012) also refreshed the velocity at periodic times. We slightly modify their approach by performing a velocity refreshment at the arrival times of a homogeneous PP of intensity $\lambda^{\text{ref}} \geq 0$, λ^{ref} being a parameter of the algorithm. A similar refreshment scheme was used for a related process in Michel, Kapfer, and Krauth (2014). Throughout the article, we use the terminology "event" for a time at which either a bounce or a refreshment occurs. The basic version of the BPS algorithm proceeds as follows:

Algorithm 1 Basic BPS algorithm

1. Initialize $\left(x^{(0)}, v^{(0)}\right)$ arbitrarily on $\mathbb{R}^d \times \mathbb{R}^d$ and let T denote the requested trajectory length.
2. For $i = 1, 2, \ldots$
   a. Simulate the first arrival time $\tau_{\text{bounce}} \in (0, \infty)$ of a PP of intensity
   $$\chi(t) = \lambda(x^{(i-1)} + v^{(i-1)}t, v^{(i-1)}).$$
   b. Simulate $\tau_{\text{ref}} \sim \text{Exp}\left(\lambda^{\text{ref}}\right)$.
   c. Set $\tau_i \leftarrow \min(\tau_{\text{bounce}}, \tau_{\text{ref}})$ and compute the next position using
   $$x^{(i)} \leftarrow x^{(i-1)} + v^{(i-1)}\tau_i. \quad (3)$$
   d. If $\tau_i = \tau_{\text{ref}}$, sample the next velocity $v^{(i)} \sim \mathcal{N}(0_d, I_d)$.
   e. If $\tau_i = \tau_{\text{bounce}}$, compute the next velocity $v^{(i)}$ using
   $$v^{(i)} \leftarrow R\left(x^{(i)}\right) v^{(i-1)}. \quad (4)$$
   f. If $t_i = \sum_{j=1}^{i} \tau_j \geq T$ exit For Loop (line 2).

In the algorithm above, $\exp(\delta)$ denotes the exponential distribution of rate δ and $\mathcal{N}(0_d, I_d)$ the standard normal on $\mathbb{R}^d$. Refer to Figure 1 for an example of a trajectory generated by BPS on a standard bivariate Gaussian target distribution. An example of a bounce time simulation is shown in Figure 1, top right, for the segment between the first and second events—the intensity $\chi(t)$ (turquoise) is obtained by thresholding $\left\langle \nabla U(x^{(1)} + v^{(1)}t), v^{(1)} \right\rangle$ (purple, dashed); arrival times of the PP of intensity $\chi(t)$ are denoted by stars.

We will show further that the transition kernel of the BPS process admits π as invariant distribution for any $\lambda^{\text{ref}} \geq 0$ but it can fail to be irreducible when $\lambda^{\text{ref}} = 0$ as demonstrated in Section 4.1.

It is thus critical to use $\lambda^{\text{ref}} > 0$. Our proof of invariance and ergodicity can accommodate some alternative refreshment steps 2d. One such variant, which we call restricted refreshment, samples $v^{(i)}$ uniformly on the unit hypersphere $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$. We compare experimentally these two variants and others in Section 4.3.

2.3. Algorithms for Bounce Time Simulation

Implementing BPS requires sampling the first arrival time τ of a one-dimensional inhomogeneous PP Π of intensity $\chi(t) = \lambda(x + vt, v)$ given by (1). Simulating such a process is a well-studied problem; see Devroye (1986, chap. 6, sec. 1.3). We review here three methods and illustrate how they can be used to implement BPS for examples from Bayesian statistics. The first method described in Section 2.3.1 will be particularly useful when the target is log-concave, while the two others described in Sections 2.3.2 and 2.3.3 can be applied to more general scenarios.

2.3.1. Simulation Using a Time-Scale Transformation

If we let $\Xi(t) = \int_0^t \chi(s)ds$, then the PP Π satisfies

$$\mathbb{P}(\tau > u) = \mathbb{P}(\Pi \cap [0, u) = \emptyset) = \exp(-\Xi(u)),$$

and therefore τ can be simulated from a uniform variate $V \sim \mathcal{U}(0, 1)$ via the identity

$$\tau = \Xi^{-1}(-\log(V)), \quad (5)$$

where Ξ^{-1} denotes the quantile function of Ξ, $\Xi^{-1}(p) = \inf\{t : p \leq \Xi(t)\}$. Refer to Figure 1, top right for a graphical illustration. This identity corresponds to the method proposed in Peters and de With (2012) to determine the bounce times and is also used in Michel, Kapfer, and Krauth (2014); Michel, Mayer, and Krauth (2015); Nishikawa, Michel, and Krauth (2015) to simulate related processes.

In general, it is not possible to obtain an analytical expression for τ. However, when the target distribution is strictly log-concave and differentiable, it is possible to solve Equation (5) numerically (see Example 1).

Example 1. Log-concave densities. If the energy is strictly convex (see Figure 1, bottom right), we can minimize it along the line specified by (x, v)

$$\tau_* = \text{argmin}_{t:t\geq 0} U(x + vt),$$

where τ_* is well defined and unique by strict convexity. On the interval $[0, \tau_*)$, which might be empty, we have $dU(x + vt)/dt < 0$ and $dU(x + vt)/dt \geq 0$ on $[\tau_*, \infty)$. The solution τ of (5) is thus necessarily such that $\tau \geq \tau_*$ and (5) can be rewritten using the gradient theorem as

$$\int_{\tau_*}^{\tau} \frac{dU(x + vt)}{dt} dt = U(x + v\tau) - U(x + v\tau_*) = -\log V. \quad (6)$$

Even if we only compute U pointwise through a black box, we can solve (6) through line search within machine precision.

We note that (6) also provides an informal connection between the BPS and MH algorithms. Exponentiating this equation, we get indeed

$$\frac{\pi(x + v\tau)}{\pi(x + v\tau_*)} = V.$$

Hence, in the log-concave case, and when the particle is climbing the energy ladder (i.e., $\tau_* = 0$), BPS can be viewed as "swapping" the order of the steps taken by the MH algorithm. In the latter, we first sample a proposal and second sample a uniform V to perform an accept-reject decision. With BPS, V is first drawn then the maximum distance allowed by the same MH ratio is traveled. As for the case of a particle going down the energy ladder,

the behavior of BPS is simpler to understand: bouncing never occurs. We illustrate this method for Gaussian distributions.

Multivariate Gaussian distributions. Let $U(x) = \|x\|^2$, then simple calculations yield

$$\tau = \frac{1}{\|v\|^2} \begin{cases} -\langle x, v\rangle + \sqrt{-\|v\|^2 \log V} & \text{if } \langle x, v\rangle \leq 0, \\ -\langle x, v\rangle + \sqrt{\langle x, v\rangle^2 - \|v\|^2 \log V} & \text{otherwise.} \end{cases} \tag{7}$$

2.3.2. Simulation Using Adaptive Thinning

When it is difficult to solve (5), the use of an adaptive thinning procedure provides an alternative. Assume we have access to local-in-time upper bounds $\bar{\chi}_s(t)$ on $\chi(t)$, that is,

$$\begin{aligned} \bar{\chi}_s(t) &= 0 \qquad \text{for all } t < s, \\ \bar{\chi}_s(t) &\geq \chi(t) \quad \text{for all } s \leq t \leq s + \Delta(s), \end{aligned}$$

where Δ is a positive function (standard thinning corresponds to $\Delta = +\infty$). Assume additionally that we can simulate the first arrival time of the PP $\bar{\Pi}_s$ with intensity $\bar{\chi}_s(t)$. Such bounds can be constructed based on upper bounds on directional derivatives of U provided the remainder of the Taylor expansion can be controlled . Algorithm 2 shows the pseudocode for the adaptive thinning procedure.

Algorithm 2 Simulation of the first arrival time of a PP through thinning

1. Set $s \leftarrow 0, \tau \leftarrow 0$.
2. Do
   a. Set $s \leftarrow \tau$.
   b. Sample τ as the first arrival point of the PP $\bar{\Pi}_s$ of intensity $\bar{\chi}_s$.
   c. If $\bar{\Pi}_s = \{\emptyset\}$ then set $\tau \leftarrow s + \Delta(s)$.
   d. If $s + \Delta(s) \leq \tau$ set $s \leftarrow s + \Delta(s)$ and go to (b).
   e. While $V > \{\chi(\tau)/\bar{\chi}_s(\tau)\}$ where $V \sim \mathcal{U}(0, 1)$.
3. Return τ.

The case $V > \{\chi(\tau)/\bar{\chi}_s(\tau)\}$ corresponds to a rejection step in the thinning algorithm but, in contrast to rejection steps that occur in standard MCMC samplers, in the BPS algorithm this means that the particle does not bounce and just coasts. Practically, we would like ideally Δ and the ratio $\chi(\tau)/\bar{\chi}_s(\tau)$ to be large. Indeed this would avoid having to simulate too many candidate events from $\bar{\Pi}_s$, which would be rejected as these rejection steps incur a computational cost.

2.3.3. Simulation Using Superposition and Thinning

Assume that the energy can be decomposed as

$$U(x) = \sum_{j=1}^{m} U^{[j]}(x), \tag{8}$$

then

$$\chi(t) \leq \sum_{j=1}^{m} \chi^{[j]}(t),$$

where $\chi^{[j]}(t) = \max(0, \langle \nabla U^{[j]}(x + tv), v\rangle)$ for $j = 1, \ldots, m$. It is therefore possible to use the thinning algorithm of Section 2.3.2 with $\bar{\chi}_0(t) = \sum_{j=1}^{m} \chi^{[j]}(t)$ for $t \geq 0$ (and $\Delta = +\infty$), as we can simulate from $\bar{\Pi}_0$ via superposition by simulating the first arrival time $\tau^{[j]}$ of each PP with intensity $\chi^{[j]}(t) \geq 0$ then returning

$$\tau = \min_{j=1,\ldots,m} \tau^{[j]}.$$

Example 2. Exponential families. Consider a univariate exponential family with parameter x, observation y, sufficient statistic $\phi(y)$, and log-normalizing constant $A(x)$. If we assume a Gaussian prior on x, we obtain

$$U(x) = \underbrace{x^2/2}_{U^{[1]}(x)} + \underbrace{-x\phi(y)}_{U^{[2]}(x)} + \underbrace{A(x)}_{U^{[3]}(x)}.$$

The time $\tau^{[1]}$ is computed analytically in Example 1 whereas the times $\tau^{[2]}$ and $\tau^{[3]}$ are given by

$$\tau^{[2]} = \begin{cases} \frac{\log V^{[2]}}{v\phi(y)} & \text{if } v\phi(y) < 0, \\ +\infty & \text{otherwise,} \end{cases}$$

and

$$\tau^{[3]} = \begin{cases} \tilde{\tau}^{[3]} & \text{if } \tilde{\tau}^{[3]} > 0, \\ +\infty & \text{otherwise,} \end{cases}$$

with $\tilde{\tau}^{[3]} = (A^{-1}(-\log V^{[3]} + A(x)) - x)/v$ and $V^{[2]}, V^{[3]} \sim \mathcal{U}(0, 1)$. For example, with a Poisson distribution with natural parameter x, we obtain

$$\tilde{\tau}^{[3]} = \frac{\log(-\log V^{[3]} + \exp(x)) - x}{v}.$$

Example 3. Logistic regression. The class label of the data point $r \in \{1, 2, \ldots, R\}$ is denoted by $y_r \in \{0, 1\}$ and its covariate $k \in \{1, 2, \ldots, d\}$ by $\iota_{r,k}$ where we assume that $\iota_{r,k} \geq 0$ (this assumption can be easily relaxed as demonstrated but would make the notation more complicated; see Galbraith (2016) for details). The parameter $x \in \mathbb{R}^d$ is assigned a standard Gaussian prior density denoted by ψ, yielding the posterior density

$$\pi(x) \propto \psi(x) \prod_{r=1}^{R} \frac{\exp(y_r \langle \iota_r, x\rangle)}{1 + \exp\langle \iota_r, x\rangle}. \tag{9}$$

Using the superposition and thinning method (Section 2.3.3), simulation of the bounce times can be broken into subproblems corresponding to $R + 1$ factors: one factor coming from the prior, with corresponding energy

$$U^{[R+1]}(x) = -\log \psi(x) = \|x\|^2/2 + \text{constant}, \tag{10}$$

and R factors coming from the likelihood of each data point, with corresponding energy

$$U^{[r]}(x) = \log(1 + \exp\langle \iota_r, x\rangle) - y_r \langle \iota_r, x\rangle. \tag{11}$$

Simulation of $\tau^{[R+1]}$ is covered in Example 1. Simulation of $\tau^{[r]}$ for $r \in \{1, 2, \ldots, R\}$ can be approached using thinning. In Section 3.1 of the supplementary material, we show that

$$\chi^{[r]}(t) \leq \bar{\chi}^{[r]} = \sum_{k=1}^{d} \mathbf{1}[v_k(-1)^{y_r} \geq 0] \iota_{r,k} |v_k|. \tag{12}$$

Since the bound is constant for a given v, we sample $\tau^{[r]}$ by simulating an exponential random variable.

2.4. Estimating Expectations

Given a realization of $x(t)$ over the interval $[0, T]$, where T is the total trajectory length, the expectation $\int_{\mathbb{R}^d} \varphi(x)\pi(dx)$ of a function $\varphi : \mathbb{R}^d \to \mathbb{R}$ with respect to π can be estimated using

$$\frac{1}{T}\int_0^T \varphi(x(t))\,dt = \frac{1}{T}\left(\sum_{i=1}^{n-1}\int_0^{\tau_i} \varphi\left(x^{(i-1)} + v^{(i-1)}s\right)ds + \int_0^{t_n - T} \varphi\left(x^{(n-1)} + v^{(n-1)}s\right)ds\right);$$

see, for example, Davis (1993). When $\varphi(x) = x_k$, $k \in \{1, 2, \ldots, d\}$, we have

$$\int_0^{\tau_i} \varphi\left(x^{(i-1)} + v^{(i-1)}s\right)ds = x_k^{(i-1)}\tau_i + v_k^{(i-1)}\frac{\tau_i^2}{2}.$$

When the above integral is intractable, we may just discretize $x(t)$ at regular time intervals to obtain an estimator

$$\frac{1}{L}\sum_{l=0}^{L-1}\varphi(x(l\delta)),$$

where $\delta > 0$ is the mesh size and $L = 1 + \lfloor T/\delta \rfloor$. Alternatively, we could approximate these univariate integrals through quadrature.

2.5. Theoretical Results

An informal proof establishing that the BPS with $\lambda^{\text{ref}} = 0$ admits π as invariant distribution is given in Peters and de With (2012). As the BPS process $z(t) = (x(t), v(t))$ is a piecewise deterministic Markov process, the expression of its infinitesimal generator can be established rigorously using Davis (1993). We show here that this generator has invariant distribution π whenever $\lambda^{\text{ref}} \geq 0$ and prove that the resulting process is additionally ergodic when $\lambda^{\text{ref}} > 0$. We denote by $\mathbb{E}_z[h(z(t))]$ the expectation of $h(z(t))$ under the law of the BPS process initialized at $z(0) = z$.

Proposition 1. For any $\lambda^{\text{ref}} \geq 0$, the infinitesimal generator $\mathcal{L}$ of the BPS is defined for any sufficiently regular bounded function $h : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ by

$$\begin{aligned}\mathcal{L}h(z) &= \lim_{t\downarrow 0}\frac{\mathbb{E}_z[h(z(t))] - h(z)}{t} \\ &= \langle \nabla_x h(x, v), v\rangle + \lambda(x, v)\{h(x, R(x)v) - h(z)\} \\ &\quad + \lambda^{\text{ref}}\int \left(h(x, v') - h(x, v)\right)\psi(v')\,dv', \end{aligned} \tag{13}$$

where we recall that $\psi(v)$ denotes the standard multivariate Gaussian density on $\mathbb{R}^d$.

This transition kernel of the BPS is nonreversible and admits ρ as invariant probability measure, where the density of ρ w.r.t. Lebesgue measure on $\mathbb{R}^d \times \mathbb{R}^d$ is given by

$$\rho(z) = \pi(x)\psi(v). \tag{14}$$

If we add the condition $\lambda^{\text{ref}} > 0$, we get the following stronger result.

Theorem 1. If $\lambda^{\text{ref}} > 0$ then ρ is the unique invariant probability measure of the transition kernel of the BPS and for ρ-almost every $z(0)$ and h integrable with respect to ρ

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T h(z(t))dt = \int h(z)\rho(z)dz \quad \text{a.s.}$$

In fact, Lemma A.3 establishes a minorization so it is only left to establish a Lyapunov function to establish polynomial or geometric ergodicity in total variation. We exhibit in Section 4.1 a simple example where P_t is not ergodic for $\lambda^{\text{ref}} = 0$.

3. The Local Bouncy Particle Sampler

3.1. Structured Target Distribution and Factor Graph Representation

In numerous applications, the target distribution admits some structural properties that can be exploited by sampling algorithms. For example, the Gibbs sampler takes advantage of conditional independence properties. We present here a "local" version of the BPS introduced in Peters and de With (2012, sec. III), which can similarly exploit these properties and, more generally, any representation of the target density as a product of positive factors

$$\pi(x) \propto \prod_{f\in F}\gamma_f(x_f), \tag{15}$$

where x_f is a restriction of x to a subset of the components of x, and F is an index set called the set of factors. Hence, the energy associated with π is of the form

$$U(x) = \sum_{f\in F} U_f(x) \tag{16}$$

with $\partial U_f(x)/\partial x_k = 0$ for any variable absent from factor f, that is, for any $k \in \{1, 2, \ldots, d\}\backslash N_f$.

Such a factorization of the target density can be formalized using factor graphs (Figure 2, top). A factor graph is a bipartite graph, with one set of vertices N called the variables, each corresponding to a component of x ($|N| = d$), and a set of vertices F corresponding to the local factors $(\gamma_f)_{f\in F}$. There is an edge between $k \in N$ and $f \in F$ if and only if $k \in N_f$. This representation generalizes undirected graphical models (Wainwright and Jordan 2008, chap. 2, sec. 2.1.3) as, for example, factor graphs can have distinct factors connected to the same set of components (i.e., $f \neq f'$ with $N_f = N_{f'}$) as in the example of Section 4.6.

3.2. Local BPS: Algorithm Description

Similarly to the Gibbs sampler, each step of the local BPS manipulates only a subset of the d components of x. Contrary to the Gibbs sampler, the local BPS does not require sampling from any full conditional distribution and each local calculation results in a change of *all* state components, not just the one being updated—how this can be done implicitly without manipulating the full state at each iteration is described below. Related processes exhibiting similar characteristics have been proposed in Kampmann, Boltz, and Kierfeld (2015), Michel,

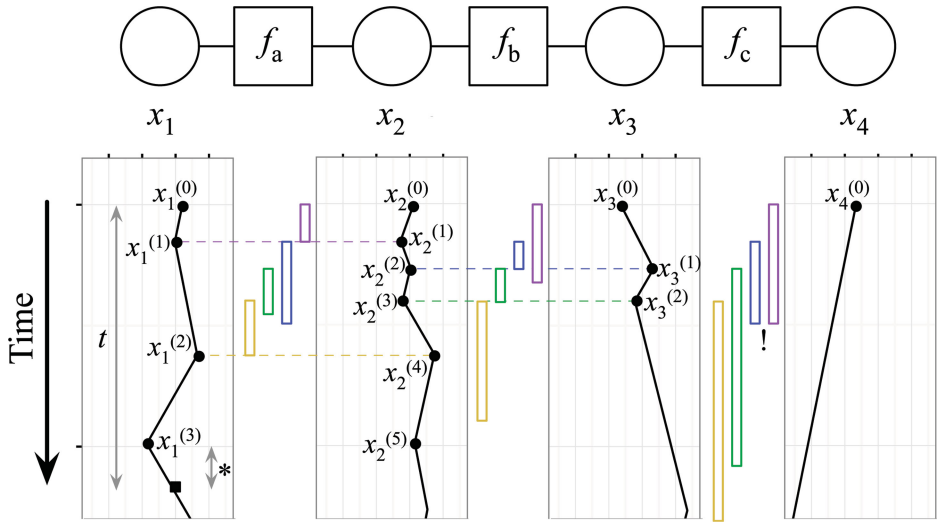

Figure 2. Top: A factor graph with $d = 4$ variables and 3 binary factors, $F = \{f_a, f_b, f_c\}$. Bottom: Sample paths of $(x_i(t))_{t\geq 0}$ for $i = 1, \ldots, 4$ for the local BPS. Rectangles represent candidate collision times stored in a priority queue. The exclamation mark shows an example where computation of a candidate collision time can be skipped. The star shows the quantity t t(3)1 used at the end of Section 3.2.

Kapfer, and Krauth (2014), Michel, Mayer, and Krauth (2015), and Nishikawa, Michel, and Krauth (2015).

For each factor $f \in F$, we define a local intensity function $\lambda_f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ and a local bouncing matrix $R_f : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ by

$$\lambda_f(x, v) = \max\left\{0, \left\langle \nabla U_f(x), v\right\rangle\right\}, \tag{17}$$

$$R_f(x)v = v - 2\frac{\left\langle \nabla U_f(x), v\right\rangle \nabla U_f(x)}{\left\|\nabla U_f(x)\right\|^2}. \tag{18}$$

We can check that $R_f(x)$ satisfies

$$k \in \{1, 2, \ldots, d\} \setminus N_f \Longrightarrow \{R_f(x)v\}_k = v_k. \tag{19}$$

When suitable, we will slightly abuse notation and write $R_f(x_f)$ for $R_f(x)$ as $R_f(x_f, x_{-f}) = R_f(x_f, x'_{-f})$ for any $x_{-f}, x'_{-f} \in \mathbb{R}^{d-|N_f|}$, where $|S|$ denotes the cardinality of a set S. Similarly, we will use $\lambda_f(x_f, v_f)$ for $\lambda_f(x, v)$.

We define a collection of PP intensities based on the previous event position $x^{(i-1)}$ and velocity $v^{(i-1)}$: $\chi_f(t) = \lambda_f(x^{(i-1)} + v^{(i-1)}t, v^{(i-1)})$. In the local BPS, the next bounce time τ is the first arrival of a PP with intensity $\chi(t) = \sum_{f\in F}\chi_f(t)$. However, instead of modifying all velocity variables at a bounce as in the basic BPS, we sample a factor f with probability $\chi_f(\tau)/\chi(\tau)$ and modify only the variables connected to the sampled factor. More precisely, the velocity v_f is updated using $R_f(x_f)$ defined in (18). A generalization of the proof of Proposition 1 given in the supplementary material shows that the local BPS algorithm results in a π-invariant kernel. In the next subsection, we describe various computationally efficient procedures to simulate this process.

For all these implementations, it is useful to encode trajectories in a sparse fashion: each variable $k \in N$ only records information at the times $t_k^{(1)}, t_k^{(2)}, \ldots$ where an event (a bounce or refreshment) affected it. By (17), this represents a sublist of the list of all event times. At each of those times $t_k^{(i)}$, the component's position $x_k^{(i)}$ and velocity $v_k^{(i)}$ right after the event is stored. Let L_k denote a list of triplets $(x_k^{(i)}, v_k^{(i)}, t_k^{(i)})_{i\geq 0}$, where $x_k^{(0)}$ and $v_k^{(0)}$ denote the initial position and velocity and $t_k^{(0)} = 0$ (see Figure 2, where the black dots denote the set of recorded triplets). This list is sufficient to compute $x_k(t)$ for $t \leq t_k^{(|L_k|+1)}$. This procedure is detailed in Algorithm 3 and an example is shown in Figure 2, where the black square on the first variable's trajectory shows how Algorithm 3 reconstructs $x_1(t)$ at a fixed time t: it identifies $i(t, 1) = 3$ as the index associated with the largest event time $t_1^{(3)}$ before time t affecting x_1 and return $x_1(t) = x_1^{(3)} + v_1^{(3)}(t - t_1^{(3)})$.

Algorithm 3 Computation of $x_k(t)$ from a list of events.

1. Find the index $i = i(t, k)$ associated with the largest time $t_k^{(i)}$ verifying $t_k^{(i)} \leq t$.
2. Set $x_k(t) \leftarrow x_k^{(i(t,k))} + (t - t_k^{(i(t,k))})v_k^{(i(t,k))}$.

3.3. Local BPS: Efficient Implementations

3.3.1. Implementation via Priority Queue

We can sample arrivals from a PP with intensity $\chi(t) = \sum_{f\in F}\chi_f(t)$ using the superposition method of Section 2.3.3, the thinning step therein being omitted. To implement this technique efficiently, we store potential future bounce times (called "candidates") t_f, one for each factor, in a priority queue Q: only a subset of these candidates will join the lists L_k, which store past, "confirmed" events. We pick the smallest time in Q to determine the next bounce time and the next factor f to modify. The priority queue structure ensures that finding the minimum element of Q or inserting/updating an element of Q can be performed with computational complexity $O(\log|F|)$. When a bounce occurs, a key observation behind efficient implementation of the local BPS is that not all the other candidate bounce times need to be resimulated. Suppose that the bounce was associated with factor f. In this case, only the candidate bounce times $t_{f'}$ corresponding to factors f' with $N_{f'} \cap N_f \neq \emptyset$ need to be resimulated. For example, consider the first bounce in Figure 2 (shown in purple), which is triggered by factor f_a

(rectangles represent candidate bounce times t_f; dashed lines connect bouncing factors to the variables that undergo an associated velocity change). Then only the velocities for the variables x_1 and x_2 need to be updated. Therefore, only the candidate bounce times for factors f_a and f_b need to be resimulated while the candidate bounce time for f_c stays constant (this is shown by an exclamation mark in Figure 2).

The method is detailed in Algorithm 4. Several operations of the BPS such as Step 4, 6.iii, 6.iv, and 7.ii can be easily parallelized.

Algorithm 4 Local BPS algorithm (priority queue implementation)

1. Initialize $\left(x^{(0)}, v^{(0)}\right)$ arbitrarily on $\mathbb{R}^d \times \mathbb{R}^d$.
2. Initialize the global clock $T \leftarrow 0$.
3. For $k \in N$ do
   a. Initialize the list $L_k \leftarrow \left(x_k^{(0)}, v_k^{(0)}, T\right)$.
4. Set $Q \leftarrow$ **new queue** $\left(x^{(0)}, v^{(0)}, T\right)$.
5. Sample $t_{\text{ref}} \sim \text{Exp}\left(\lambda^{\text{ref}}\right)$.
6. While more events $i = 1, 2, \ldots$ requested do
   a. $(t, f) \leftarrow$ **smallest candidate bounce time and associated factor in** Q.
   b. Remove (t, f) from Q.
   c. Update the global clock, $T \leftarrow t$.
   d. If $T < t_{\text{ref}}$ then
      i. $\left(v_f\right)_k \leftarrow v_k^{(|L_k|-1)}$ for all $k \in N_f$.
      ii. $x_f \leftarrow x_f(T)$ (computed using Algorithm 3).
      iii. For $k \in N_f$ do
         A. $x_k^{(|L_k|)} \leftarrow x_k^{(|L_k|-1)} + (T - t_k^{(|L_k|-1)}) v_k^{(|L_k|-1)}$, where $t_k^{(|L_k|-1)}$ and $v_k^{(|L_k|-1)}$ are retrieved from L_k.
         B. $v_k^{(|L_k|)} \leftarrow \left\{R_f\left(x_f\right) v_f\right\}_k$.
         C. $L_k \leftarrow \left\{L_k, \left(x_k^{(|L_k|)}, v_k^{(|L_k|)}, T\right)\right\}$ (add the new sample to the list).
      iv. For $f' \in F : N_{f'} \cap N_f \neq \emptyset$ (note: this includes the update of f) do
         A. for all $k \in N_{f'}$.
         B. $x_{f'} \leftarrow x_{f'}(T)$ (computed using Algorithm 3).
         C. Simulate the first arrival time $\tau_{f'}$ of a PP of intensity $\lambda_{f'}\left(x_{f'} + t v_{f'}, v_{f'}\right)$ on $[0, +\infty)$.
         D. Set in Q the candidate bounce time associated with f' to the value $t_{f'} = T + \tau_{f'}$.
   e. Else
      i. Sample $v' \sim \mathcal{N}(0_d, I_d)$.
      ii. $Q \leftarrow$ **new queue** $\left(x\left(t_{\text{ref}}\right), v', t_{\text{ref}}\right)$ where $x\left(t_{\text{ref}}\right)$ is computed using Algorithm 3.
      iii. Set $t_{\text{ref}} \leftarrow t_{\text{ref}} + \tau_{\text{ref}}$ where $\tau_{\text{ref}} \sim \text{Exp}\left(\lambda^{\text{ref}}\right)$.
7. Return the samples encoded as the lists L_k, $k \in N$.

Algorithm 5 New Queue (x, v, T)

1. For $f \in F$ do
   a. $\left(v_f\right)_k \leftarrow v_k^{(|L_k|-1)}$ for all $k \in N_f$.
   b. $x_f \leftarrow x_f(T)$ (computed using Algorithm 3).
   c. Simulate the first arrival time τ_f of a PP of intensity $\lambda_f\left(x_f + t v_f, v_f\right)$ on $[0, +\infty)$.
   d. Set in Q the time associated to f to the value $T + \tau_f$.
2. Return Q.

3.3.2. Implementation via Thinning

When the number of factors involved in Step 6(d)iv is large, the previous queue-based implementation can be computationally expensive. Implementing the local BPS in this setup is closely related to the problem of simulating stochastic chemical kinetics and innovative solutions have been proposed in this area. We adapt here the algorithm proposed in Thanh and Priami (2015) to the local BPS context. For ease of presentation, we present the algorithm without refreshment and only detail the simulation of the bounce times. This algorithm relies on the ability to compute local-in-time upper bounds on λ_f for all $f \in F$. More precisely, we assume that given a current position x and velocity v, and $\Delta \in (0, \infty]$, we can find a positive number $\bar{\chi}_f$, such that for any $t \in [0, \Delta)$, we have $\bar{\chi}_f \geq \lambda_f(x + vt, v)$. We can also use this method on a subset G of F and combine it with the previously discussed techniques to sample candidate bounce times for factors in $F \backslash G$ but we restrict ourselves to $G = F$ to simplify the presentation.

Algorithm 6 Local BPS algorithm (thinning implementation)

1. Initialize $\left(x^{(0)}, v^{(0)}\right)$ arbitrarily on $\mathbb{R}^d \times \mathbb{R}^d$.
2. Initialize the global clock $T \leftarrow 0$.
3. Initialize $\bar{T} \leftarrow \Delta$ (time until which local upperbounds are valid).
4. Compute local-in-time upper bounds $\bar{\chi}_f$ for $f \in F$ such that $\bar{\chi}_f \geq \lambda_f(x^{(0)} + v^{(0)}t, v^{(0)})$ for all $t \in [0, \Delta)$.
5. While more events $i = 1, 2, \ldots$ requested do
   a. Sample $\tau \sim \text{Exp}(\bar{\chi})$ where $\bar{\chi} = \sum_{f \in F} \bar{\chi}_f$.
   b. If $\left(T + \tau > \bar{T}\right)$ then
      i. $x^{(i)} \leftarrow x^{(i-1)} + v^{(i-1)}(\bar{T} - T)$.
      ii. $v^{(i)} \leftarrow v^{(i-1)}$.
      iii. For all $f \in F$, update $\bar{\chi}_f$ to ensure that $\bar{\chi}_f \geq \lambda_f(x^{(i)} + v^{(i)}t, v^{(i)})$ for $t \in [0, \Delta)$.
      iv. Set $T \leftarrow \bar{T}$, $\bar{T} \leftarrow \bar{T} + \Delta$.
   c. Else
      i. $x^{(i)} \leftarrow x^{(i-1)} + v^{(i-1)}\tau$.
      ii. Sample $\mathcal{F} \in F$ where $\mathbb{P}\left(\mathcal{F} = f\right) = \bar{\chi}_f / \bar{\chi}$.
      iii. If $V < \lambda_{\mathcal{F}}\left(x^{(i)}, v^{(i-1)}\right) / \bar{\chi}_{\mathcal{F}}$ where $V \sim \mathcal{U}(0, 1)$ then a bounce for factor $\mathcal{F}$ occurs at time T.
         A. $v^{(i)} \leftarrow R_{\mathcal{F}}\left(x^{(i)}\right) v^{(i-1)}$.
         B. For all $f' \in F : N_{f'} \cap N_{\mathcal{F}} \neq \emptyset$, update $\bar{\chi}_{f'}$ to ensure that $\bar{\chi}_{f'} \geq \lambda_{f'}(x^{(i)} + v^{(i)}t, v^{(i)})$ for $t \in [0, \bar{T} - T - \tau)$.
      iv. Else
         A. $v^{(i)} \leftarrow v^{(i-1)}$.
      v. Set $T \leftarrow T + \tau$.

Algorithm 6 will be particularly useful in scenarios where summing over the bounds (Step 5a) and sampling a factor (Step 5(c)ii) can be performed efficiently. A scenario where it is possible to implement these two operations in constant time is detailed in Section 4.6. Another scenario where sampling quickly from $\mathcal{F}$ is feasible is if the number of distinct upper bounds is much smaller than the number of factors. For example, we only need to sample a factor $\mathcal{F}$ uniformly at random if $\Lambda = \bar{\chi}_f = \bar{\chi}_{f'}$ for all f, f' in F and $\bar{\chi} = |F| \cdot \Lambda$ (i.e., no factor

needs to be inspected to execute Algorithm 5(c)ii) and the thinning procedure in Step 5(c)iii boils down to

$$V \leq \frac{|F| \max\left(0, \left\langle \nabla U_f(x^{(i)}), v^{(i-1)} \right\rangle\right)}{\bar{\chi}}. \tag{20}$$

A related approach has been adopted in Bierkens, Fearnhead, and Roberts (2016) for the analysis of big data. In this particular scenario, an alternative local BPS can also be implemented where $s > 1$ factors $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_s)$ are sampled uniformly at random without replacement from F, the thinning occurs with probability

$$\frac{|F|}{s\bar{\chi}} \max\left(0, \sum_{j=1}^{s} \left\langle \nabla U_{\mathcal{F}_j}(x^{(i)}), v^{(i-1)} \right\rangle\right) \tag{21}$$

and the components of x belonging to $N_{\mathcal{F}}$ bounce based on $\sum_{j=1}^{s} \nabla U_{\mathcal{F}_j}(x)$. One can check that the resulting dynamics preserves π as an invariant distribution. In contrast to $s = 1$, this is not an implementation of local BPS described in Algorithm 6, but instead this corresponds to a local BPS update for a random partition of the factors.

4. Numerical Results

4.1. Gaussian Distributions and the Need for Refreshment

We consider an isotropic multivariate Gaussian target distribution, $U(x) = \|x\|^2$, to illustrate the need for refreshment. Without refreshment, we obtain from Equation (7)

$$\left\langle x^{(i)}, v^{(i)} \right\rangle = \begin{cases} -\sqrt{-\log V_i} & \text{if } \left\langle x^{(i-1)}, v^{(i-1)} \right\rangle \leq 0, \\ -\sqrt{\left\langle x^{(i-1)}, v^{(i-1)} \right\rangle^2 - \log V_i} & \text{otherwise,} \end{cases}$$

and

$$\left\|x^{(i)}\right\|^2 = \begin{cases} \left\|x^{(i-1)}\right\|^2 - \left\langle x^{(i-1)}, v^{(i-1)} \right\rangle^2 - \log V_i & \text{if } \left\langle x^{(i-1)}, v^{(i-1)} \right\rangle \leq 0, \\ \left\|x^{(i-1)}\right\|^2 - \log V_i & \text{otherwise,} \end{cases}$$

see the supplementary material for details. In particular, these calculations show that if $\left\langle x^{(i)}, v^{(i)} \right\rangle \leq 0$ then $\left\langle x^{(j)}, v^{(j)} \right\rangle \leq 0$ for $j > i$ so that $\|x^{(i)}\|^2 = \left\|x^{(1)}\right\|^2 - \left\langle x^{(1)}, v^{(1)} \right\rangle^2 - \log V_i$ for $i \geq 2$. In particular for $x^{(0)} = e_1$ and $v^{(0)} = e_2$ with e_i being elements of standard basis of $\mathbb{R}^d$, the norm of the position at all points along the trajectory can never be smaller than 1 as illustrated in Figure 3.

In this scenario, we show that BPS without refreshment admits a countably infinite collection of invariant distributions. Let us define $r(t) = \|x(t)\|$ and $m(t) = \langle x(t), v(t) \rangle / \|x(t)\|$ and denote by χ_k the probability density of the chi distribution with k degrees of freedom.

Proposition 2. For any dimension $d \geq 2$, the process $(r(t), m(t))_{t \geq 0}$ is Markov and its transition kernel is invariant with respect to the probability densities $\{f_k(r, m) \propto \chi_k(\sqrt{2}r) \cdot (1 - m^2)^{(k-3)/2}; k \in \{2, 3, \ldots\}\}$.

By Theorem 1, we have a unique invariant measure as soon as $\lambda^{\text{ref}} > 0$.

Next, we look at the scaling of the effective sample size (ESS) per CPU second of the basic BPS algorithm for $\varphi(x) = x_1$ when $\lambda_{\text{ref}} = 1$ as the dimension d of the isotropic normal target increases. The ESS is estimated using the R package *mcmcse* (Flegal, Hughes, and Vats 2015) by evaluating the trajectory on a fine discretization of the sampled trajectory. The results in log-log scale are displayed in Figure 3. The curve suggests a decay of roughly $d^{-1.47}$, slightly inferior to the $d^{-1.25}$ scaling for an optimally tuned Hamiltonian Monte Carlo (HMC) algorithm (Creutz 1988, sec. III; Neal 2011, sec. 5.4.4). It should be noted that BPS achieves this scaling without varying any tuning parameter, whereas HMC's performance critically depends on tuning two parameters (leap-frog stepsize and number of leap-frog steps). Both BPS and HMC compare favorably to the d^{-2} scaling of the optimally tuned random walk MH (Roberts and Rosenthal 2001).

4.2. Comparison of the Global and Local Schemes

We compare the basic "global" BPS of Section 2 to the local BPS of Section 3 on a sparse Gaussian field. We use a chain-shaped undirected graphical model of length $d = 1000$ and perform separate experiments for various pairwise precision parameters for the interaction between neighbors in the chain. Both methods are run for 60 sec. We compare the Monte Carlo estimate of the variance of x_{500} to its true value. The results are shown in Figure 4. The smaller computational complexity per

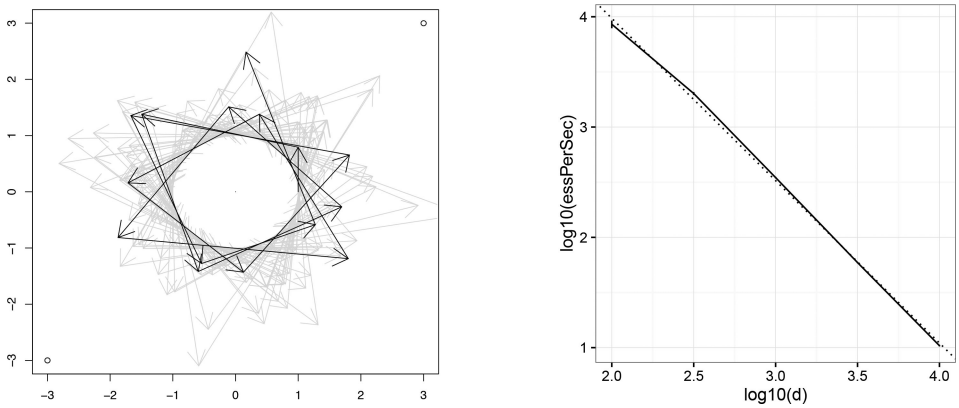

Figure 3. Left: The 200 first segments/bounces of a BPS path for $\lambda^{\text{ref}} = 0$ (for clarity the first 15 segments are in black, the following ones in light gray): the center of the space is never explored. Right, solid line: ESS per CPU second as a function of d (log-log scale), along with 95% confidence intervals based on 40 runs (the intervals are small and difficult to see). Dashed line: Linear regression curve. See Section 4.1 for details.

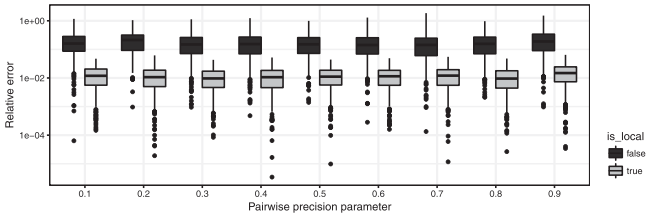

Figure 4. Boxplots of relative errors over 100 local BPS runs for Gaussian chain-shaped fields of pairwise precisions 0.1–0.9.

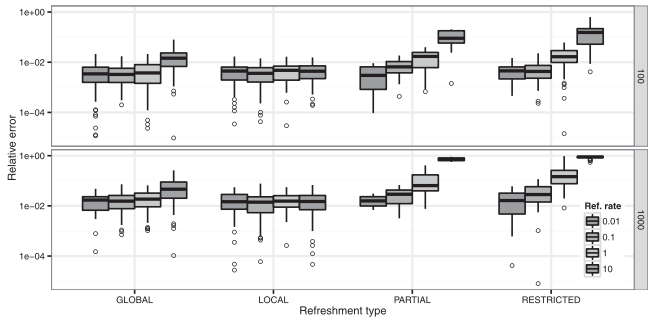

Figure 5. Comparison of refreshment schemes for $d = 100$ (top) and $d = 1000$ (bottom). Each boxplot summarizes the relative error for the variance estimates (in log scale) of x_{50} over 100 runs of BPS.

local bounce of the local BPS offsets significantly the associated decrease in expected trajectory segment length. Moreover, both versions appear insensitive to the pairwise precision used in this sparse Gaussian field.

4.3. Comparisons of Alternative Refreshment Schemes

In Section 2, the velocity was refreshed using a Gaussian distribution. We compare here this global refreshment scheme to three alternatives:

Local refreshment: If the local BPS is used, the factor graph structure can be exploited to design computationally cheaper refreshment operators. We pick one factor $f \in F$ uniformly at random and resample only the components of v with indices in N_f. By the same argument used in Section 3, each refreshment requires bounce time recomputation only for the factors f' with $N_f \cap N_{f'} \neq \emptyset$.

Restricted refreshment: The velocities are refreshed according to $\phi(v)$, the uniform distribution on $\mathcal{S}^{d-1}$, and the BPS admits now $\rho(z) = \pi(x)\phi(v)$ as invariant distribution.

Restricted partial refreshment: A variant of restricted refreshment where we sample an angle θ by multiplying a Beta(α, β)-distributed random variable by 2π. We then select a vector uniformly at random from the unit length vectors that have an angle θ from v. We used $\alpha = 1, \beta = 4$ to favor small angles.

We compare these methods for different values of λ^{ref}, the trade-off being that too small a value can lead to a failure to visit certain regions of the space, while too large a value leads to a random walk behavior.

The rationale behind the partial refreshment procedure is to suppress the random walk behavior of the particle path arising from a refreshment step independent from the current velocity. Refreshment is needed to ensure ergodicity but a "good" direction should only be altered slightly. This strategy is akin to the partial momentum refreshment strategy for HMC methods (Horowitz 1991; Neal 2011, sec. 4.3) and could be similarly implemented for global refreshment. It is easy to check that all of the above schemes preserve π as invariant distribution. We tested these schemes on the chain-shaped factor graph described in the previous section (with the pairwise precision parameter set to 0.5). All methods are provided with a computational budget of 30 sec. The results are shown in Figure 5. The results show that local refreshment is less sensitive to λ^{ref}, performing as well or better than global refreshment. The performance of the restricted and partial methods appears more sensitive to λ^{ref} and generally inferior to the other two schemes.

One limitation of the results in this section is that the optimal refreshment scheme and refreshment rate will in general be problem dependent. Adaptation methods used in the HMC literature could potentially be adapted to this scenario (Wang, Mohamed, and de Freitas 2013; Hoffman and Gelman 2014), but we leave these extensions to future work.

4.4. Comparisons with HMC Methods on High-Dimensional Gaussian Distributions

We compare the local BPS with no partial refreshment and $\lambda^{\text{ref}} = 1$ to advanced adaptive versions of HMC implemented using Stan Hoffman and Gelman (2014) on a 100-dimensional Gaussian example from Neal (2011, sec. 5.3.3.4). For each method, we compute the relative error on the estimated marginal variances after a wall clock time of 30 sec, excluding from this time the time taken to compile the Stan program. The adaptive HMC methods use 1000 iterations of adaptation. When only the leap-frog stepsize is adapted ("adapt=true"), HMC provides several poor estimates of marginal variances. These deviations disappear when adapting a diagonal metric (denoted "fit-metric") and/or using advanced auxiliary variable methods to select the number of leap-frog steps (denoted "nuts"). Given that adaptation is critical to HMC in this scenario, it is encouraging that BPS without adaptation is competitive (Figure 6).

Next, we compare the local BPS to NUTS ("adapt=true,fit_metric=true,nuts=true") as the dimension d increases. Experiments are performed on the chain-shaped Gaussian random field of Section 4.2 with the pairwise precision parameter set to 0.5. We vary the length of the chain (10, 100, 1000), and run Stan's implementation of NUTS for 1000 iterations + 1000 iterations of adaptation. We measure the wall-clock time (excluding the time taken to compile the Stan program) and then run our method for the same wall-clock time 40 times for each chain size. The absolute value of the relative error averaged on 10 equally spaced marginal variances is measured as a function of the percentage of the total computational budget used; see Figure 7. The gap between the two methods widens as d increases. To visualize the different behavior of the two algorithms, three marginals of the Stan and BPS paths for $d = 100$ are shown in Figure 8. Contrary to Section 4.1, BPS outperforms here HMC as its local version is able to exploit the sparsity of the random field.

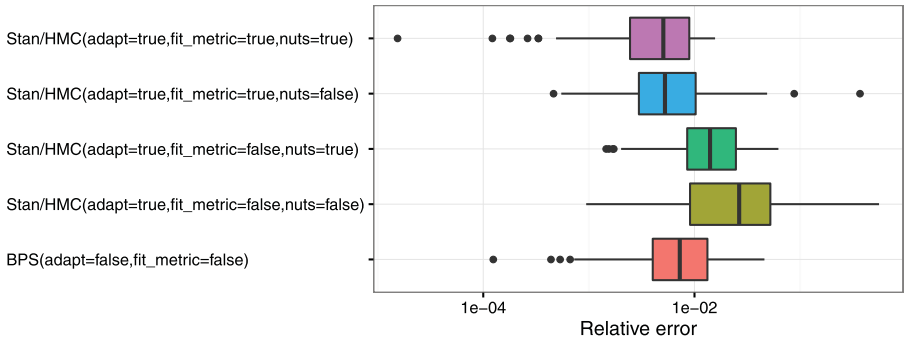

Figure 6. Boxplots showing the relative absolute error of variance estimates for a fixed computational budget.

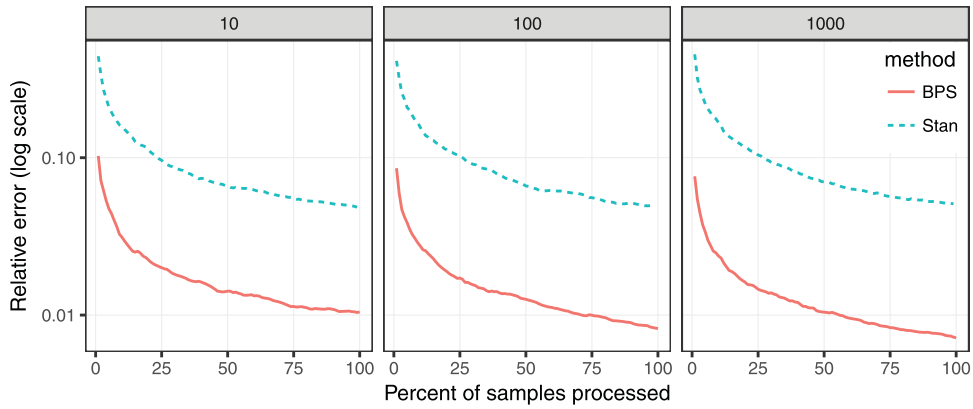

Figure 7. Relative error for $d = 10$ (left), $d = 100$ (middle), and $d = 1000$ (right), averaged over 10 of the dimensions and 40 runs. Each d uses a fixed computational budget.

4.5. Poisson–Gaussian Markov Random Field

Let $x = (x_{i,j} : i, j \in \{1, 2, \ldots, 10\})$ denote a grid-shaped Gaussian Markov random field with pairwise interactions of the same form as those used in the previous chain examples (pairwise precision set to 0.5) and let $y_{i,j}$ be Poisson distributed, independent over i, j given x, with rate $\exp(x_{i,j})$. We generate a synthetic dataset $y = (y_{i,j} : i, j \in \{1, 2, \ldots, 10\})$ from this model and approximate the resulting posterior distribution of x. We first run Stan with default settings ("adapt=true,fit_metric=true,nuts=true") for 16, 32, 64, ..., 4096 iterations. For each number of Stan iterations, we run local BPS for the same wall-clock time as Stan, using a local refreshment ($\lambda^{\text{ref}} = 1$) and the method from Example 2 for the bouncing time computations. We repeat these experiments 10 times with different random seeds. Figure 9 displays the boxplots of the estimates of the posterior variances of the variables $x_{0,0}$ and $x_{5,5}$ summarizing the 10 replications. As expected, both methods converge to the same value, but BPS requires markedly less computing time to achieve any given level of accuracy.

4.6. Bayesian Logistic Regression for Large Datasets

Consider the logistic regression model introduced in Example 3 when the number of data R is large. In this context, standard MCMC schemes such as the MH algorithm are computationally expensive as they require evaluating the likelihood associated

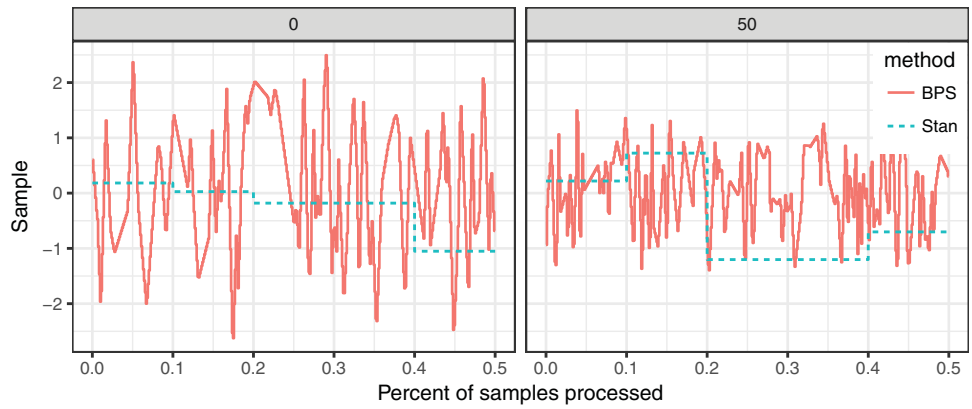

Figure 8. Simulated paths for x_0 and x_{50} for $d = 100$. Each state of the HMC trajectory is obtained by leap-frog steps (not displayed), these latter cannot be used to estimate Monte Carlo averages as HMC relies on an MH step. In contrast, BPS exploits the full path.

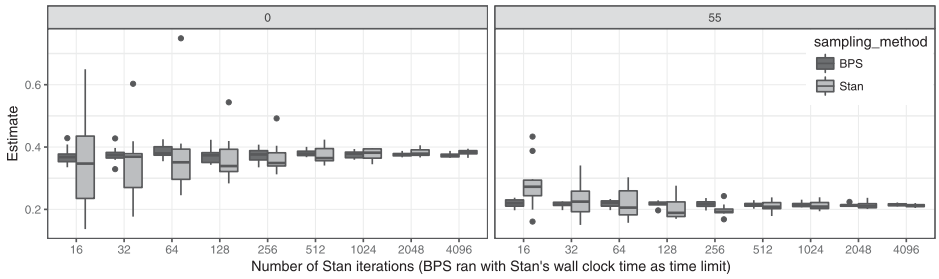

Figure 9. Boxplots of estimates of the posterior variance of $x_{0,0}$ (left) and $x_{5,5}$ (right) using Stan implementation of HMC and local BPS.

with the R observations at each iteration. This has motivated the development of techniques that only evaluate the likelihood of a subset of the data at each iteration. However, most of the methods currently available introduce either some non-vanishing asymptotic bias, for example, the subsampling MH scheme proposed in Bardenet, Doucet, and Holmes (2014), or provide consistent estimates converging at a slower rate than regular MCMC algorithms, for example, the stochastic gradient Langevin dynamics introduced in Welling and Teh (2011) and Teh, Thiéry, and Vollmer (2016). The only available algorithm that only requires evaluating the likelihood of a subset of data at each iteration yet provides consistent estimates converging at the standard Monte Carlo rate is the Firefly algorithm (Maclaurin and Adams 2014).

In this context, we associate $R + 1$ factors with the target posterior distribution: one for the prior and one for each data point with $x_f = x$ for all $f \in F$. As a uniform upper bound on the intensities of these local factors is available for restricted refreshment, see Section 3.1 of the supplementary material, we could use (21) in conjunction with Algorithm 6 to provide an alternative to the Firefly algorithm, which selects at each bounce a subset of s data points uniformly at random without replacement. For $s = 1$, a related algorithm has been recently explored in Bierkens, Fearnhead, and Roberts (2016). In presence of outliers, this strategy can be inefficient as the uniform upper bound becomes very large, resulting in a computationally expensive implementation. After a precomputation step of complexity $O(R \log R)$ only executed once, we show here that Algorithm 6 can be implemented using data-dependent bounds mitigating the sensitivity to outliers while maintaining the computational cost of each bounce independent of R. We first precompute the sum of covariates over the data points, $\iota_k^c = \sum_{r=1}^{R} \iota_{r,k} \mathbf{1}[y_r = c]$,

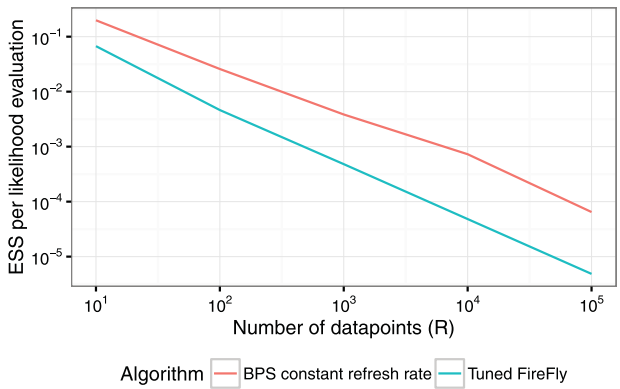

Figure 10. ESS per-datum likelihood evaluation for local BPS and firefly.

for $k \in \{1, \ldots, d\}$ and class label $c \in \{0, 1\}$. Using these quantities, it is possible to compute

$$\bar{\chi} = \sum_{r=1}^{R} \bar{\chi}^{[r]} = \sum_{k=1}^{d} |v_k| \, \iota_k^{\mathbf{1}[v_k < 0]},$$

with $\bar{\chi}^{[r]}$ given in (12). If d is large, we can keep the sum $\bar{\chi}$ in memory and add-and-subtract any local updates to it. The implementation of Step 5(c)ii relies on the alias method (Devroye 1986, sec. 3.4). A detailed description of these derivations and of the algorithm is presented in Section 3 of the supplementary material.

We compare the local BPS with thinning to the MAP-tuned Firefly algorithm implementation provided by the authors. This version of Firefly outperforms experimentally significantly the standard MH in terms of ESS per-datum likelihood (Maclaurin and Adams 2014). The two algorithms are here compared in terms of this criterion, where the ESS is averaged over the $d = 5$ components of x. We generate covariates $\iota_{rk} \overset{\text{iid}}{\sim} \mathcal{U}(0.1, 1.1)$ and data $y_r \in \{0, 1\}$ for $r = 1, \ldots, R$ according to (9) and set a zero-mean normal prior of covariance $\sigma^2 I_d$ for x. For the algorithm, we set $\lambda_{\text{ref}} = 0.5$, $\sigma^2 = 1$, and $\Delta = 0.5$, which is the length of the time interval for which a constant upper bound for the rate associated with the prior is used, see Algorithm C.1. As illustrated in Figure 10, local BPS always outperforms experimentally Firefly, by about an order of magnitude for large datasets. However, we also observe that both Firefly and local BPS have an ESS per datum likelihood evaluation decreasing in approximately $1/R$ so that the gains brought by these algorithms over a correctly scaled random walk MH algorithm do not appear to increase with R. The rate for local BPS is slightly superior in the regime of up to 10^4 data points, but then returns to the approximate $1/R$ rate. To improve this rate, one can adapt the control variate ideas introduced in Bardenet, Doucet, and Holmes (2017) for the MH algorithm to these schemes. This has been proposed in Bierkens, Fearnhead, and Roberts (2016) for a related algorithm and in Galbraith (2016) for the local BPS.

4.7. Bayesian Inference of Evolutionary Parameters

We analyze a dataset of primate mitochondrial DNA (Hayasaka, Takashi, and Satoshi 1988) at the leaves of a fixed reference tree (Huelsenbeck and Ronquist 2001) containing 898 sites and 12 species. We want to approximate the posterior evolutionary parameters x encoded into a rate matrix Q. A detailed description of this phylogenetic model can be found in the supplementary materials. The global BPS is used with restricted

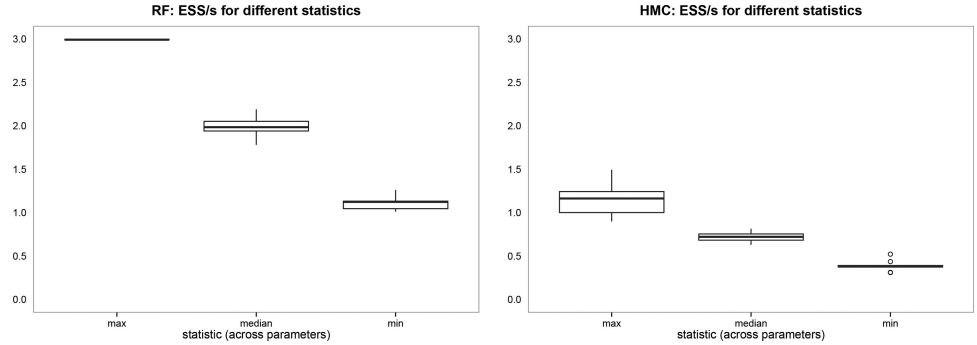

Figure 11. Boxplots of maximum, median, and minimum ESS per second for BPS (left) and HMC (right).

refreshment and $\lambda^{\text{ref}} = 1$ in conjunction with an auxiliary variable-based method similar to the one described in Zhao et al. (2016), alternating between two moves: (1) sampling continuous-time Markov chain paths along the tree given x using uniformization, (2) sampling x given the path (in which case the derivation of the gradient is simple and efficient). The only difference compared to Zhao et al. (2016) is that we substitute the HMC kernel by the kernel induced by running BPS for a fixed trajectory length. This auxiliary variable method is convenient because, conditioned on the paths, the energy function is convex and hence we can simulate the bouncing times using the method described in Example 1.

We compare against a state-of-the-art HMC sampler (Wang, Mohamed, and de Freitas 2013) that uses Bayesian optimization to adapt the key parameters of HMC, the leap-frog stepsize, and the number of leap-frog steps, while preserving convergence to the target distribution. Both our method and this HMC method are implemented in Java and share the same gradient computation code. Refer to the supplementary materials for additional background and motivation behind this adaptation method.

We first perform various checks to ensure that both BPS and HMC chains are in close agreement given a sufficiently large number of iterations. After 20 million HMC iterations, the credible intervals estimates from the HMC method are in close agreement with those obtained from BPS (result not shown) and both methods pass the Geweke diagnostic (Geweke 1992).

To compare the effectiveness of the two samplers, we look at the ESS per second of the model parameters. We show the maximum, median, and maximum over the 10 parameter components for 10 runs, for both BPS and HMC in Figure 11. We observe a speed-up by a factor two for all statistics considered (maximum, median, minimum). In the supplement, we show that the HMC chain displays much larger autocorrelations than the BPS chain.

5. Discussion

Most MCMC methods currently available, such as the MH and HMC algorithms, are discrete-time reversible processes. There is a wealth of theoretical results showing that nonreversible Markov processes mix faster and provide lower variance estimates of ergodic averages (Ottobre 2016). However, most of the nonreversible processes studied in the literature are diffusions and cannot be simulated exactly. The BPS is an alternative continuous-time Markov process that, thanks to its piecewise deterministic paths, can be simulated exactly for many problems of interest in statistics.

As any MCMC method, the BPS can struggle in multimodal scenarios and when the target exhibits very strong correlations. However, for a range of applications including sparse factor graphs, large datasets, and high-dimensional settings, we have observed empirically that BPS is on par or outperforms state-of-the art methods such as HMC and Firefly. The main practical limitation of the BPS compared to HMC is that its implementation is model-specific and requires more than knowing ∇U pointwise. An important open problem is therefore whether its implementation, and in particular the simulation of bouncing times, can be fully automated. However, the techniques described in Section 2.3 are already sufficient to handle many interesting models. There are also numerous potential methodological extensions of the method to study. In particular, it has been shown in Girolami and Calderhead (2011) how one can exploit the local geometric structure of the target to improve HMC and it would be interesting to investigate how this could be achieved for BPS. More generally, the BPS is a specific continuous-time piecewise deterministic Markov process (Davis 1993). This class of processes deserves further exploration as it might provide a whole new class of efficient MCMC methods.

Supplementary Materials

The supplementary material includes the proofs of Proposition 1, Proposition 2 and Theorem 1 and a proof of the validity of the local BPS algorithm. It also includes a detailed description of the algorithm used in Section 4.6 and additional information on the experiments performed in Section 4.7 is also provided.

Acknowledgments

The authors thank Markus Upmeier for helpful discussions of differential geometry as well as Nicholas Galbraith, Fan Wu, and Tingting Zhao for their comments.

Funding

Alexandre Bouchard-Côté's research is partially supported by a Discovery Grant from the National Science and Engineering Research Council. Arnaud Doucet's research is partially supported by the Engineering and Physical Sciences Research Council (EPSRC), Grants EP/K000276/1, EP/K009850/1 and by the Air Force Office of Scientific Research/Asian

Office of Aerospace Research and Development, Grant AFOSRA/AOARD-144042. Sebastian Vollmer's research is partially supported by the EPSRC Grants EP/K009850/1 and EP/N000188/1.

References

Bardenet, R., Doucet, A., and Holmes, C. C. (2014), "Towards Scaling up MCMC: An Adaptive Subsampling Approach," in *Proceedings of the 31st International Conference on Machine Learning*, pp. 405–413. [865]

——— (2017), "On Markov Chain Monte Carlo for Tall Data," *Journal of Machine Learning Research*, 18, 1515–1557. [865]

Bierkens, J., Fearnhead, P., and Roberts, G. O. (2016), "The Zig-Zag Process and Super-Efficient Monte Carlo for Bayesian Analysis of Big Data," Technical Report. arXiv preprint:1607.03188. [862,865]

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2015), "The Bouncy Particle Sampler: A Non-Reversible Rejection-Free Markov Chain Monte Carlo Method," Technical Report. arxiv preprint:1510.02451v1. [856]

Creutz, M. (1988), "Global Monte Carlo Algorithms for Many-Fermion Systems," *Physical Review D*, 38, 1228–1237. [862]

Davis, M. H. A. (1993), *Markov Models & Optimization* (Vol. 49), Boca Raton, FL: CRC Press. [859,866]

Devroye, L. (1986), *Non-uniform Random Variate Generation*, New York: Springer-Verlag. [857,865]

Flegal, J. M., Hughes, J., and Vats, D. (2015), *mcmcse: Monte Carlo Standard Errors for MCMC, R Package Version 1.1-2*. Available at *https://cran.r-project.org/web/packages/mcmcse/index.html* [862]

Galbraith, N. (2016), "On Event-Chain Monte Carlo Methods," Master's thesis, Department of Statistics, Oxford University. [858,865]

Geweke, J. (1992), "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments," *Bayesian Statistics*, 4, 169–193. [866]

Girolami, M., and Calderhead, B. (2011), "Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods," *Journal of the Royal Statistical Society, Series B*, 73, 123–214. [866]

Hayasaka, K., Takashi, G., and Satoshi, H., (1988), "Molecular Phylogeny and Evolution of Primate Mitochondrial DNA," *Molecular Biology and Evolution*, 5, 626–644. [865]

Hoffman, M. D., and Gelman, A. (2014), "The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, 15, 1593–1623. [863]

Horowitz, A. M. (1991), "A Generalized Guided Monte Carlo Algorithm," *Physics Letters B*, 268, 247–252. [863]

Huelsenbeck, J. P., and Ronquist, F. (2001), "MRBAYES: Bayesian Inference of Phylogenetic Trees," *Bioinformatics*, 17, 754–755. [865]

Kampmann, T. A., Boltz, H. H., and Kierfeld, J. (2015), "Monte Carlo Simulation of Dense Polymer Melts using Event Chain Algorithms," *Journal of Chemical Physics*, 143, 044105. [855,859]

Liu, J. S. (2008), *Monte Carlo Strategies in Scientific Computing*, New York: Springer. [855]

Maclaurin, D., and Adams, R. P. (2014), "Firefly Monte Carlo: Exact MCMC with Subsets of Data," in *Uncertainty in Artificial Intelligence* (Vol. 30), pp. 543–552. [865]

Michel, M., Kapfer, S. C., and Krauth, W. (2014), "Generalized Event-Chain Monte Carlo: Constructing Rejection-Free Global-Balance Algorithms from Infinitesimal Steps," *Journal of Chemical Physics*, 140, 054116. [855,857,860]

Michel, M., Mayer, J., and Krauth, W. (2015), "Event-Chain Monte Carlo for Classical Continuous Spin Models," *Europhysics Letters*, 112, 20003. [855,857,860]

Neal, R. M. (2011), "Markov Chain Monte Carlo using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*, eds. S. Brooks, A. Gelman, G. L. Jones, and X. L. Meng, pp. 113–162, Boca Raton, FL: Chapman & Hall/CRC. [862,863]

Nishikawa, Y., Michel, M., Krauth, W., and Hukushima, K. (2015), "Event-Chain Monte Carlo Algorithm for the Heisenberg Model," *Physical Review E*, 92, 063306. [855,857,860]

Ottobre, M. (2016), "Markov Chain Monte Carlo and Irreversibility," *Reports on Mathematical Physics*, 77, 267–292. [866]

Peters, E. A. J. F., and de With, G. (2012), "Rejection-Free Monte Carlo Sampling for General Potentials," *Physical Review E*, 85, 026703. [855,856,857,859]

Roberts, G. O., and Rosenthal, J. S. (2001), "Optimal Scaling for Various Metropolis-Hastings Algorithms," *Statistical Science*, 16, 351–367. [862]

Teh, Y. W., Thiéry, A. H., and Vollmer, S. J. (2016), "Consistency and Fluctuations for Stochastic Gradient Langevin Dynamics," *Journal of Machine Learning Research*, 17, 1–33. [865]

Thanh, V. H., and Priami, C. (2015), "Simulation of Biochemical Reactions with Time-Dependent Rates by the Rejection-Based Algorithm," *Journal of Chemical Physics*, 143, 054104. [861]

Wainwright, M. J., and Jordan, M. I. (2008), "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends in Machine Learning*, 1, 1–305. [859]

Wang, Z., Mohamed, S., and de Freitas, N. (2013), "Adaptive Hamiltonian and Riemann manifold Monte Carlo," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 1462–1470. [863,866]

Welling, M., and Teh, Y. W. (2011), "Bayesian Learning via Stochastic Gradient Langevin Dynamics," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688. [865]

Zhao, T., Wang, Z., Cumberworth, A., Gsponer, J., de Freitas, N., and Bouchard-Côté, A. (2016), "Bayesian Analysis of Continuous Time Markov Chains with Application to Phylogenetic Modelling," *Bayesian Analysis*, 11, 1203–1237. [866]

Supplementary Material

The Bouncy Particle Sampler: A Non-Reversible Rejection-Free Markov Chain Monte Carlo Method

Alexandre Bouchard-Côté*, Sebastian J. Vollmer† and Arnaud Doucet‡

3rd May 2018

*Department of Statistics, University of British Columbia, Canada.

†Mathematics Institute and Department of Statistics, University of Warwick, Coventry, UK

‡Department of Statistics, University of Oxford, UK.

1 Proofs of Section 2

1.1 Proof of Proposition 1

The BPS process is a specific piecewise-deterministic Markov process so the expression of its generator follows from [2, Theorem 26.14]. Its adjoint is given in [6] and a derivation of this expression from first principles can be found in Section 4. To establish the invariance with respect to ρ, we first show that $\int \mathcal{L}h(z)\rho(z)\,\mathrm{d}z = 0$. We have

$$\begin{aligned}\int\int \mathcal{L}h(z)\rho(z)\,\mathrm{d}z &= \int\int \langle \nabla_x h(x,v), v\rangle \rho(z)\,\mathrm{d}z && (22)\\ &+ \int\int \lambda(x,v)[h(x,R(x)v) - h(x,v)]\rho(z)\,\mathrm{d}z && (23)\\ &+ \lambda^{\mathrm{ref}} \int\int\int (h(x,v') - h(x,v))\,\psi(\mathrm{d}v')\,\rho(z)\,\mathrm{d}z && (24)\end{aligned}$$

As $\rho(z) = \pi(x)\psi(v)$, the term (24) is trivially equal to zero, while a change-of-variables shows that

$$\int\int \lambda(x,v)\, h(x, R(x)\, v)\rho(z)\, \mathrm{d}z = \int\int \lambda(x, R(x)\, v)\, h(x,v)\rho(z)\, \mathrm{d}z \tag{25}$$

as $R^{-1}(x)\, v = R(x)\, v$ and$\|R(x)\, v\| = \|v\|$ implies $\psi(R(x)\, v) = \psi(v)$. Additionally, by integration by parts, we obtain as h is bounded

$$\int\int \langle \nabla_x h(x,v), v\rangle \rho(z)\, \mathrm{d}z = \int\int \langle \nabla U(x), v\rangle\, h(x,v)\, \rho(z)\, \mathrm{d}z. \tag{26}$$

Substituting (25) and (26) into (22)-(23)-(24), we obtain

$$\int\int \mathcal{L}h(z)\rho(z)\, \mathrm{d}z = \int\int [\langle \nabla U(x), v\rangle + \lambda(x, R(x)\, v) - \lambda(x,v)]h(x,v)\rho(z)\, \mathrm{d}z.$$

Now we have

$$\begin{aligned}
\int\int \mathcal{L}h(z)\rho(z)\, \mathrm{d}z &= \int\int [\langle \nabla U(x), v\rangle + \lambda(x, R(x)\, v) - \lambda(x,v)]h(x,v)\rho(z)\, \mathrm{d}z. \\
&= \int\int [\langle \nabla U(x), v\rangle + \max\{0, \langle \nabla U(x), R(x)\, v\rangle\} - \max\{0, \langle \nabla U(x), v\rangle\}]h(x,v)\rho(z)\, \mathrm{d}z \\
&= \int\int [\langle \nabla U(x), v\rangle + \max\{0, -\langle \nabla U(x), v\rangle\} - \max\{0, \langle \nabla U(x), v\rangle\}]h(x,v)\rho(z)\, \mathrm{d}z \\
&= 0,
\end{aligned}$$

where we have used $\langle \nabla U(x), R(x)\, v\rangle = -\langle \nabla U(x), v\rangle$ and $\max\{0, -f\} - \max\{0, f\} = -f$ for any f. The result now follows by [2, Proposition 34.7].

1.2 Proof of Theorem 1

Informally, our proof of ergodicity relies on showing that refreshments allow the process to explore the entire space of velocities and positions, $\mathbb{R}^d \times \mathbb{R}^d$. To do so, it will be useful to condition on an event $\mathscr{E}$ on which paths are "tractable." Since two refreshment events are sufficient to reach any given destination point, we would like to focus on such paths (only one refreshment is not sufficient since a destination point specifies both a final position and a final velocity). In particular, refreshments are simpler to analyze than bouncing events, so we would like to condition on an event on which no bouncing occurs in a time interval of interest.

To formalize this idea, we make use of the time-scale implementation of the BPS algorithm (Section 2.3.1). This allows us to express $\mathscr{E}$ in terms of simple independent events. We start by introducing some notation for the time-scale implementation of the BPS algorithm. Let $i \geq 1$ denote the index of the current event being simulated, and assume without loss of generality that $\lambda^{\text{ref}} = 1$. Let x_0, v_0 denote the initial position and velocity, while the positions and velocities at the event times, $z_i = (x_i, v_i)$, $i \in \{1, 2, \dots\}$ are defined as in Algorithm 1, we use the notation (x_i, v_i) instead of $(x^{(i)}, v^{(i)})$ to slightly simplify notation. At event time i, we simulate three independent random variables: two exponentially distributed, $e_i^{(\text{bounce})}, \tau_i^{(\text{ref})} \sim \text{Exp}(1)$, and one d-dimensional normal, $n_i \sim \mathcal{N}(0, I)$. The candidate time to a refreshment is given by $\tau_i^{(\text{ref})}$ and the candidate time to a bounce event is defined as $\tau_i^{(\text{bounce})} = \Xi_{z_{i-1}}^{-1}(e_i^{(\text{bounce})})$, where Ξ_z^{-1} is the quantile function of $\Xi_z(t) = \int_0^t \chi_z(s)\,\mathrm{d}s$, and $\chi_{x,v}(s) = \lambda(x + vs, v)$. The time to the next event is $\tau_i = \min\left\{\tau_i^{(\text{bounce})}, \tau_i^{(\text{ref})}\right\}$. The random variable n_i is only used if $\tau_i = \tau_i^{(\text{ref})}$, otherwise the bouncing operator is used to update the velocity in a deterministic fashion.

We can now define our tractable set and establish its key properties.

Lemma 1. *Let $t > 0$, and assume the initial point of the BPS, satisfies $\|x_0\| \leq t$, $\|v_0\| \leq 1$. If $\|\nabla U\|^* = \sup\{\|\nabla U(x)\| : \|x\| \leq 3t\}$ then the event*

$$\mathscr{E} = \underbrace{\left(\tau_1^{(ref)} + \tau_2^{(ref)} \leq t < \tau_1^{(ref)} + \tau_2^{(ref)} + \tau_3^{(ref)}\right)}_{\mathscr{E}_1} \cap \underbrace{\bigcap_{i \in \{1,2\}} (\|n_i\| \leq 1)}_{\mathscr{E}_2} \cap \underbrace{\bigcap_{i \in 1,2,3} \left(e_i^{(bounce)} \geq t\|\nabla U\|^*\right)}_{\mathscr{E}_3}$$

has the following properties:

1. *On the event $\mathscr{E}$, we have $\|v(t')\| \leq 1$ and $\|x(t')\| \leq 2t$ for all $t' \in [0, t]$,*
2. *On the event $\mathscr{E}$, there are exactly two refreshments and no bouncing in the interval $(0, t)$, i.e. $\tau_1 = \tau_1^{(ref)}$, $\tau_2 = \tau_2^{(ref)}$, and $\tau_1 + \tau_2 \leq t \leq \tau_1 + \tau_2 + \tau_3$,*
3. *$\mathbb{P}(\mathscr{E})$ is a strictly positive constant that does not depend on $z(0) = (x_0, v_0)$,*
4. *$v_i | \mathscr{E} \overset{i.i.d.}{\sim} \psi_{\leq 1}(0, I)$ for $i \in \{1, 2\}$, where $\psi_{\leq 1}$ denotes the truncated Gaussian distribution, with $\|v_i\| \leq 1$,*
5. *$\left(\tau_1^{(ref)}, \tau_2^{(ref)}\right) | \mathscr{E} \sim \mathcal{U}\left(\{(\tau_1, \tau_2) \in (0, t)^2 : \tau_1 + \tau_2 \leq t\}\right)$.*

Proof. To prove Part 1 and 2, we will make use of this preliminary result: on $\mathscr{E}_3$, $\|v_{i-1}\| \leq 1, \|x_{i-1}\| \leq 2t$ implies $\tau_i^{(\text{bounce})} \geq t$ for $i \in \{1,2,3\}$. Indeed, $\|v_{i-1}\| \leq 1$ and $\|x_{i-1}\| \leq 2t$ imply that $\chi_{z_{i-1}}(t') \leq \|\nabla U\|^*$ for all $t' \in [0,t]$. It follows that $\Xi_{z_{i-1}}(t) \leq \|\nabla U\|^* t$. Hence, by the continuity of $\Xi_{z_{i-1}}$ and standard properties of the quantile function, $\tau_i^{(\text{bounce})} = \Xi_{z_{i-1}}^{-1}(e_i^{(\text{bounce})}) \geq t$.

Part 1 and 2: by the assumption on x_0 and v_0 and our preliminary result, $\tau_1^{(\text{bounce})} \geq t$, and hence, combining with $\mathscr{E}_1$ and $\mathscr{E}_2$ we have $\tau_1 = \tau_1^{(\text{ref})} \leq t$ and $\|v_1\| = \|n_1\| \leq 1$. Also, by the triangle inequality, $\|x_1\| \leq \|x_0\| + \|x_1 - x_0\| \leq t + \tau_1^{(\text{ref})} \leq 2t$. We can therefore apply our preliminary result again and obtain $\tau_2^{(\text{bounce})} \geq t$, and hence, combining again with $\mathscr{E}_1$ and $\mathscr{E}_2$, we have $\tau_2 = \tau_2^{(\text{ref})}$, $\tau_1 + \tau_2 \leq t$, $\|v_2\| = \|n_2\| \leq 1$. Applying the triangle inequality a second time yields $\|x_2\| \leq t + \tau_1^{(\text{ref})} + \tau_2^{(\text{ref})} \leq 2t$. We apply our preliminary result one last time to obtain $\tau_3^{(\text{bounce})} \geq t$. Hence, if $\tau_3^{(\text{ref})} > \tau_3^{(\text{bounce})}$, $\tau_1 + \tau_2 + \tau_3 = \tau_1^{(\text{ref})} + \tau_2^{(\text{ref})} + \tau_3^{(\text{bounce})} \geq t$, while if $\tau_3^{(\text{ref})} \leq \tau_3^{(\text{bounce})}$, we can use $\mathscr{E}_1$ to conclude that $\tau_1 + \tau_2 + \tau_3 = \tau_1^{(\text{ref})} + \tau_2^{(\text{ref})} + \tau_3^{(\text{ref})} \geq t$. It follows from the triangle inequality that $\|x(t')\| \leq 2t$ for all $t' \in [0,t]$.

Part 3, 4 and 5: these follow straightforwardly from the construction of $\mathscr{E}$. □

Note that the statement and proof of Part 4 is simple because $\mathscr{E}_3 \in \sigma(e_i^{(\text{bounce})} : i \in \{1,2,3\})$. In contrast, conditioning on conceptually simpler events of the form $(\tau_i^{(\text{bounce})} > t) \in \sigma\left(e_i^{(\text{bounce})}, z_{i-1}\right)$ leads to conditional distributions on v_k which are harder to characterize.

In the following, $B_R(x)$ denotes the d-dimensional Euclidean ball of radius R centered at x.

Lemma 2. *For all $\epsilon, t > 0$ such that $\epsilon \leq t/6$, and $v, v_2 \in B_1(0)$, $x, x' \in B_\epsilon(0)$, $0 \leq \tau_1 \leq \frac{t}{6}$, and $\frac{2t}{3} \leq \tau_2 \leq \frac{5t}{6}$, we have $\|v_1\| \leq 1$, where v_1 is defined by:*

$$v_1 = \frac{(x' - (t - \tau_1 - \tau_2)v_2) - (x + \tau_1 v)}{\tau_2}. \tag{27}$$

Proof. By the triangle inequality:

$$\begin{aligned} \|v_1\| &\leq \frac{\|x'\| + (t - \tau_1 - \tau_2)\|v_2\| + \|x\| + \tau_1\|v\|}{\tau_2} \\ &\leq \frac{\epsilon + (t - \tau_1 - \tau_2) + \epsilon + \tau_1}{\tau_2} \leq 1. \end{aligned}$$

□

The next lemma formalizes the idea that refreshments allow the BPS to explore the whole space. We use $\mathrm{vol}(A)$ to denote the Lebesgue volume of a measurable set $A \in \mathbb{R}^d \times \mathbb{R}^d$ and $P_t(z, A)$ is the probability $z(t) \in A$ given $z(0) = z$ under the BPS dynamics.

Lemma 3. *1. For all $\epsilon, t > 0$ such that $\epsilon \leq t/6$, there exist $\delta > 0$ such that for all $z = (x, v) \in B = B_\epsilon(0) \times B_1(0)$ and measurable set $A \subset \mathbb{R}^d \times \mathbb{R}^d$,*

$$P_t(z, A) \geq \delta \ \mathrm{vol}(A \cap B). \tag{28}$$

2. For all $t_0 > 0$, $z = (x, v) \in \mathbb{R}^d \times \mathbb{R}^d$ and open set $W \subset \mathbb{R}^d \times \mathbb{R}^d$ there exists an integer $n \geq 1$ such that $P_{nt_0}(z, W) > 0$.

Proof. Part 1: We have

$$\begin{aligned} P_t(z, A) &= \mathbb{E}_z[\mathbf{1}_A(z(t))] \\ &\geq \mathbb{P}(\mathscr{E})\mathbb{E}_z[\mathbf{1}_A(z(t))|\mathscr{E}] \\ &= \mathbb{P}(\mathscr{E})\mathbb{E}_z[\mathbf{1}_A(x + \tau_1^{(\mathrm{ref})} v + \tau_2^{(\mathrm{ref})} v_1 + (t - \tau_1^{(\mathrm{ref})} - \tau_2^{(\mathrm{ref})}) v_2|\mathscr{E}], \end{aligned}$$

where we used Part 2 of Lemma 1, which holds since $\|x\| \leq t/6 \leq t$. By Lemma 1, Part 3, it is enough to show $\mathbb{E}_z[\mathbf{1}_A(z(t))|\mathscr{E}] \geq \delta' \mathrm{vol}(A \cap B)$ for some $\delta' > 0$.

Using Lemma 1, Parts 4 and 5, we can rewrite the above conditional expectation as:

$$\mathbb{E}_z[\mathbf{1}_A(z(t))|\mathscr{E}] = \iiiint \mathbf{1}_{A\cap B}\left(x + \tau_1 v + \tau_2 v_1 + (t - \tau_1 - \tau_2)\, v_2\right) \psi_{\leq 1}(v_1)\psi_{\leq 1}(v_2) p(\tau_1, \tau_2)\, \mathrm{d}v_2\, \mathrm{d}v_1\, \mathrm{d}\tau_2\, \mathrm{d}\tau_1.$$

We will use the coarea formula to reorganize the order of integration, see e.g. Section 3.2 of [1]. We start by introducing some notation.

For C^1 Riemannian manifolds M and N of dimension m and n, a differentiable map $F : M \to N$ and h a measurable test function, the coarea formula can be written as:

$$\int_M h(F(y)) \mathrm{d}\mathcal{H}_m(y) = \int_N \mathrm{d}\mathcal{H}_n(u) h(u) \int_{F^{-1}(u)} \mathrm{d}\mathcal{H}_{m-n}(x) \frac{1}{JF(x)}.$$

Here $\mathcal{H}_m$, $\mathcal{H}_n$ and $\mathcal{H}_{m-n}$ denote the volume measures associated with the Riemannian metric on M, N and $F^{-1}(u)$ (with the induced metric of M). In the above equations, JF is a generalization of the determinant of the Jacobian $JF := \sqrt{\det g(\nabla f_i, \nabla f_j)}$ where g is the corresponding Riemannian metric and f is the representation of F in local coordinates, see [1]. Here $JF{=}\sqrt{\det DF\, DF^\top}$ where DF is defined in equation (30) below.

We apply the coarea formula to $M = \left\{(\tau_1, \tau_2) \in (0,t)^2 : \tau_1 + \tau_2 \leq t\right\} \times B_1(0) \times B_1(0)$, $N = B_t(x) \times B_1(0)$, $F_z(\tau_1, v_1, \tau_2, v_2) = (x + \tau_1 v + \tau_2 v_1 + (t - \tau_1 - \tau_2)v_2, v_2)$, $m = 2d + 2$, $n = 2d$ and obtain:

$$\mathbb{E}_z[\mathbf{1}_A(z(t))|\mathscr{E}] \;=\; \int_N \mathrm{d}z' \mathbf{1}_{A\cap B}(z') \underbrace{\int_{F_z^{-1}(z')} \frac{\mathrm{d}\mathcal{H}_2(\tau_1,\tau_2,v_1,v_2)\psi_{\leq 1}(v_1)\psi_{\leq 1}(v_2)p(\tau_1,\tau_2)}{JF_z(\tau_1,\tau_2,v_1,v_2)}}_{I_z(z')}, \quad (29)$$

where $p(\tau_1, \tau_2)$ denotes the joint conditional density of $\tau_1, \tau_2|\mathscr{E}$ described in Part 5 of Lemma 1, and:

$$DF_z = \begin{pmatrix} v - v_2 & \tau_2 I & v_1 - v_2 & (t - \tau_1 - \tau_2)\, I \\ 0 & 0 & 0 & I \end{pmatrix}, \quad (30)$$

$$DF_z\; DF_z^\top = \begin{pmatrix} (v - v_2)(v - v_2)^\top + (v_1 - v_2)\,(v_1 - v_2)^\top + \left(\tau_2^2 + (t - \tau_1 - \tau_2)^2\right) I & (t - \tau_1 - \tau_2)\, I \\ (t - \tau_1 - \tau_2)\, I & I \end{pmatrix},$$

$$JF_z = \sqrt{\det DF_z\; DF_z^\top}.$$

We define $\delta' = \inf\{I_z(z') : z, z' \in B\}$, and obtain the following inequality

$$\begin{aligned} \int_N \mathrm{d}z' \mathbf{1}_{A\cap B}(z') I_z(z') \;&\geq\; \delta' \int_N \mathrm{d}z' \mathbf{1}_{A\cap B}(z') \\ &=\; \delta'\ \mathrm{vol}(A \cap B). \end{aligned}$$

It is therefore enough to show that $\delta' > 0$. To do so, we will derive the following bounds related to the integral in $I_z(z')$:

1. Its domain of integration $F_z^{-1}(z')$ is guaranteed to contain a set of positive $\mathcal{H}_2$ measure.

2. Its integrand is bounded below by a strictly positive constant.

To establish 1, we let $z' = (x', v') = (x', v_2)$, and notice that rearranging

$$x' = x + \tau_1 v + \tau_2 v_1 + (t - \tau_1 - \tau_2) v_2$$

yields an expression for v_1 given in (27). From Lemma 2, it follows that

$$C_z(z') := \left\{ \left(\tau_1, \frac{(x' - (t - \tau_1 - \tau_2)v') - (x + \tau_1 v)}{\tau_2}, \tau_2, v' \right) : 0 \leq \tau_1 \leq \frac{t}{6}, \frac{2t}{3} \leq \tau_2 \leq \frac{5t}{6} \right\} \subseteq F_z^{-1}(z'),$$

and $\mathcal{H}_2(C_z(z')) \geq (t/6)^2$ since the surface of the graph of a function is larger than the surface of the domain.

To establish 2, we start by analyzing JF_z. Exploiting its block structure, we obtain:

$$\begin{aligned} JF_z^2 &= \det DF_z \; DF_z^\top \\ &= \det((v - v_2)(v - v_2)^\top + (v_1 - v_2)(v_1 - v_2)^\top + \tau_2^2 I) \\ &\leq \left(\frac{\|v - v_2\|^2 + \|v_1 - v_2\|^2}{d} + \tau_2^2 \right)^d \leq \left(8 + t^2\right)^d \end{aligned}$$

Moreover, it follows from basic properties of the truncated Gaussian distribution $\psi_{\leq 1}$ and of $p(\tau_1, \tau_2)$ that

$$\inf \{\psi_{\leq 1}(v_1)\psi_{\leq 1}(v_2)p(\tau_1, \tau_2) : (\tau_1, v_1, \tau_2, v_2) \in M\} \quad = \quad K > 0.$$

Combining (29) to 1 and 2, we obtain,

$$
\begin{aligned}
I_z(z') &\geq \frac{K}{(8+t^2)^d} \int_{F_z^{-1}(z')} \mathrm{d}\mathcal{H}_2(\tau_1, \tau_2, v_1, v_2) \\
&\geq \frac{K}{(8+t^2)^d} \mathcal{H}_2(C_z(z')) \\
&\geq \frac{K}{(8+t^2)^d} \left(\frac{t}{6}\right)^2,
\end{aligned}
$$

and hence, $\delta' > 0$.

To prove Part 2 of the lemma, we divide the trajectory of length nt_0 into three "phases" namely a deceleration, travel, and acceleration phases, or respective lengths $t_\mathrm{d} + t_\mathrm{t} + t_\mathrm{a} = nt_0$ defined below (see also Supplement for a figure illustrating the notation used in this part of the lemma). This allows us to use Part 1 of the present lemma which requires velocities bounded in norm by one. We require an acceleration phase since W may not necessarily include velocities of norms bounded by one.

First, we show that we decelerate with positive probability by time $t_\mathrm{d} = t_0$. Let $\mathscr{R}$ denote the event that there is exactly one refreshment in the interval $(0, t_0)$, and that the refreshed velocity has norm bounded by one. Define also $r_0 = t_0 \max\{1, \|v\|\}$, which bounds the distance travelled in $(0, t_0)$ for outcomes in $\mathscr{R}$, since bouncing does not change the norm of the velocity. We have:

$$
P_{t_0}(z, B_{r_0}(x) \times B_1(0)) \geq \mathbb{P}_z(\mathscr{R}) = (t_0 \lambda^{\mathrm{ref}}) \exp(-t_0 \lambda^{\mathrm{ref}}) \psi(B_1(0)) =: K' > 0.
$$

Next, to prepare applying the first part of the lemma, set

$$
\begin{aligned}
\epsilon &= 1 + \max\{\|x\| + r_0, r'\}, \\
r' &= \inf\left\{\|x'\| : \exists\, v' \in \mathbb{R}^d \text{ with } (x', v') \in W\right\}.
\end{aligned}
$$

Informally, ϵ is selected so that the ball of radius ϵ around the origin contains both any position attained after deceleration, as well as ball around a point $x^\star$ in W. Indeed, since W is open and that $\epsilon > r'$, there exists some $r > 0, (x^\star, v^\star) \in W$ such that $B_r(x^\star) \times B_r(v^\star) \subseteq W$ and $B_\epsilon(0) \supseteq B_{r_0}(x) \cup B_r(x^\star)$. Let also $n = 2 + \left\lceil \frac{6\epsilon}{t_0} \right\rceil$.

Of the total time nt_0, we reserve time $t_\mathrm{a} = \min\left\{t_0, (2(\|v^\star\|/r + 1))^{-1}, r/2\right\}$ to accelerate. This time is selected so that (a) $t_0 - t_\mathrm{a} \geq 0$, and (b), if we start with a position in $B_{r/2}(x^\star)$, move with a

velocity bounded in norm by $\bar{v} = \max\{1, \|v^\star\| + r\}$ for a time $\Delta t \leq \min\left\{(2(\|v^\star\|/r + 1))^{-1}, r/2\right\}$, we have that the final position is in $B_r(x^\star)$. This holds since the distance travelled is bounded by $\bar{v}\Delta t \leq r/2$. Hence, by a similar argument as used for deceleration, we have, for all $z'' \in B_{r/2}(x^\star) \times B_1(0)$,

$$P_{t_\mathrm{a}}(z'', B_r(x^\star) \times B_r(v^\star)) \geq (t_\mathrm{a}\lambda^{\mathrm{ref}}) \exp(-t_\mathrm{a}\lambda^{\mathrm{ref}})\psi(B_r(v^\star)) =: K'' > 0.$$

With these definitions, we can apply the first part of the present lemma with $t_\mathrm{t} = (n-2)t_0 + (t_0 - t_\mathrm{a}) \geq 6\epsilon$ and obtain a constant $\delta > 0$ such that $P_{t_\mathrm{t}}(z', A) \geq \delta \,\mathrm{vol}(A \cap (B_\epsilon(0) \times B_1(0)))$ for all $z' \in B_\epsilon(0) \times B_1(0)$ and measurable set $A \subseteq \mathbb{R}^{2d}$. We thus obtain:

$$\begin{aligned}
P_{nt_0}(z, W) &\geq P_{nt_0}(z, B_r(x^\star) \times B_r(v^\star)) \\
&= \int\int P_{t_\mathrm{d}}(z, \mathrm{d}z') P_{t_\mathrm{t}}(z', \mathrm{d}z'') P_{t_\mathrm{a}}(z'', B_r(x^\star) \times B_r(v^\star)) \\
&\geq \int_{z' \in B_{r_0}(x) \times B_1(0)} \int_{z'' \in B_{r/2}(x^\star) \times B_1(0)} P_{t_\mathrm{d}}(z, \mathrm{d}z') P_{t_\mathrm{t}}(z', \mathrm{d}z'') P_{t_\mathrm{a}}(z'', B_r(x^\star) \times B_r(v^\star)) \\
&\geq K' K'' \delta \,\mathrm{vol}(B_{r/2}(x^\star) \times B_1(0)) > 0.
\end{aligned}$$

□

We can now exploit this Lemma to prove Theorem 1.

Proof of Theorem 1. Suppose BPS is not ergodic, then it follows from standard results in ergodic theory that there are two measures μ_1 and μ_2 such that $\mu_1 \perp \mu_2$ and $\mu_i P_{t_0} = \mu_i$; see e.g. [5, Theorem 1.7]. Thus there is a measurable set $A \subset \mathbb{R}^d \times \mathbb{R}^d$ such that

$$\mu_1(A) = \mu_2(A^c) = 0. \tag{31}$$

Let $A_1 = A$, $A_2 = A^c$, and $B = B_1(0) \times B_1(0)$. Because of Lemma 3 Part 2 and Lemma 2.2 of [5] the support of the μ_i is $\mathbb{R}^d \times \mathbb{R}^d$. Thus, $\mu_i(B) > 0$ for $i \in \{1, 2\}$. At least one of $A_1 \cap B$ or $A_2 \cap B$ has a positive Lebesgue volume, hence, we can denote by $i^\star \in \{1, 2\}$ an index satisfying $\mathrm{vol}(A_{i^\star} \cap B) > 0$. Now, we pick $t = 6$ and obtain from Lemma 3 Part 1 that there is some $\delta > 0$

such that $P_t(z, A_{i^\star}) \geq \delta \ \mathrm{vol}(A_{i^\star} \cap B)$ for all $z \in B$. By invariance we have

$$\begin{aligned}
\mu_{i^\star}(A_{i^\star}) &= \int \mu_{i^\star}(\mathrm{d}z)\, P_t(z, A_{i^\star}) \\
&\geq \int_B \mu_{i^\star}(\mathrm{d}z)\, P_t(z, A_{i^\star}) \\
&\geq \int_B \mu_{i^\star}(\mathrm{d}z)\, \delta \ \mathrm{vol}(A_{i^\star} \cap B) \\
&= \mu_{i^\star}(B)\delta \ \mathrm{vol}(A_{i^\star} \cap B) > 0.
\end{aligned}$$

This contradicts that $\mu_i(A_i) = 0$ for $i \in \{1, 2\}$.

The law of large numbers then follows by Birkhoff's pointwise ergodic theorem; see e.g. [4, Theorem 2.30, Section 2.6.4]. □

2 Proof of Proposition 2

The dynamics of the BPS can be lumped into a two-dimensional Markov process involving only the radius $r(t) = \|x(t)\|$ and $m(t) = \langle x(t), v(t)\rangle / \|x(t)\|$ for any dimensionality $d \geq 2$. The variable $m(t)$ can be interpreted (via $\arccos(m(t))$) as the angle between the particle position $x(t)$ and velocity $v(t)$. Because of the strong Markov property we can take $\tau_1 = 0$ without loss of generality and let t be some time between the current event and the next, yielding:

$$\begin{aligned}
r(t) &= \sqrt{\langle x(0) + v(0)\cdot t, x(0) + v(0)\cdot t\rangle} = \sqrt{r_0^2 + 2m(0)\, r(0)\, t + t^2} \\
m(t) &= \frac{\langle x(0) + v(0)\cdot t, v(0)\rangle}{\|x(0) + v(0)\cdot t\|} = \frac{m(0)\, r(0) + t}{r(t)}
\end{aligned} \tag{32}$$

If there is a bounce at time t, then $r(t)$ is not modified but $m(t) = -m(t)$.The bounce happens with intensity $\lambda(x + tv, v) = \max(0, \langle x + vt, v\rangle)$. These processes can also be written as an Stochastic Differential Equation (SDE) driven by a jump process whose intensity is coupled to its position. This is achieved by writing the deterministic dynamics given in (32) between events as the following Ordinary Differential Equation (ODE):

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} r(t) &= \frac{2m(t)\, r(t)}{2r(t)} = m(t) \\
\frac{\mathrm{d}}{\mathrm{d}t} m(t) &= \frac{r(t) - (r(t)\, m(t))\, m(t)}{(r(t))^2} = \frac{1 - (m(t))^2}{r(t)}.
\end{aligned}$$

Taking the bounces into account turns this ODE into an SDE with

$$\begin{aligned} \mathrm{d}r\left(t\right) &= m\left(t\right)\mathrm{d}t \\ \mathrm{d}m\left(t\right) &= \frac{1-\left(m\left(t\right)\right)^{2}}{r\left(t\right)}\mathrm{d}t - 2m\left(t\right)dN_{t}. \end{aligned} \tag{33}$$

where N_t is the counting process associated with a PP with intensity $\max\left(0, r\left(t\right)m\left(t\right)\right)$.

Now consider the push forward measure of $\mathcal{N}\left(0, \frac{1}{2}I_k\right)\otimes\mathcal{U}(\mathcal{S}^{k-1})$ under the map $(x,v)\mapsto\left(\|x\|, \langle x,v\rangle / \|x\|\right)$ where $\mathcal{U}(\mathcal{S}^{k-1})$ is the uniform distribution on $\mathcal{S}^{k-1}$. This yields the collection of measures with densities $f_k(r,m)$. One can check that $f_k(r,m)$ is invariant for (33) for all $k \geq 2$.

3 Bayesian logistic regression for large datasets

3.1 Bounds on the intensity

We derive here a datapoint-specific upper bound $\bar{\chi}^{[r]}$ to $\chi^{[r]}(t)$. First, we need to compute the gradient for one datapoint:

$$\nabla U^{[r]}(x) = \iota_r(\mathrm{logistic}\left\langle \iota_r, x\right\rangle - y_r),$$

where:

$$\mathrm{logistic}(a) = \frac{e^a}{1+e^a}.$$

We then consider two sub-cases depending on $y_r = 0$ or $y_r = 1$. Suppose first $y_r = 0$, and let $x(t) = x + tv$

$$\begin{aligned} \chi^{[r]}(t) &= \max\{0, \langle \nabla U_r(x(t)), v\rangle\} \\ &= \max\left\{0, \sum_{k=1}^{d}\iota_{r,k}v_k\mathrm{logistic}\left\langle \iota_r, x(t)\right\rangle\right\} \\ &\leq \sum_{k=1}^{d}\mathbf{1}[v_k > 0]\iota_{r,k}v_k\mathrm{logistic}\left\langle \iota_r, x(t)\right\rangle \\ &\leq \underbrace{\sum_{k=1}^{d}\mathbf{1}[v_k > 0]\iota_{r,k}v_k}_{\bar{\chi}^{[r]}} \end{aligned}$$

Similarly, we have for $y_r = 1$

$$\begin{aligned}
\chi^{[r]}(t) &= \max\left\{0, \sum_{k=1}^{d} \iota_{r,k} v_k (\text{logistic}\,\langle \iota_r, x(t)\rangle - 1)\right\} \\
&= \max\left\{0, \sum_{k=1}^{d} \iota_{r,k} (-v_k)(1 - \text{logistic}\,\langle \iota_r, x(t)\rangle)\right\} \\
&\le \sum_{k=1}^{d} \mathbf{1}[v_k < 0] \iota_{r,k} (-v_k)(1 - \text{logistic}\,\langle \iota_r, x(t)\rangle) \\
&= \sum_{k=1}^{d} \mathbf{1}[v_k < 0] \iota_{r,k} |v_k| (1 - \text{logistic}\,\langle \iota_r, x(t)\rangle) \\
&\le \underbrace{\sum_{k=1}^{d} \mathbf{1}[v_k < 0] \iota_{r,k} |v_k|}_{\bar{\chi}^{[r]}}.
\end{aligned}$$

Combining these terms we obtain

$$\bar{\chi}^{[r]} = \sum_{k=1}^{d} \mathbf{1}[v_k (-1)^{y_r} \ge 0] \iota_{r,k} |v_k|.$$

When implementing Algorithm 6, we need to bound $\sum_{r=1}^{R} \chi^{[r]}(t)$. We have

$$\begin{aligned}
\bar{\chi} &= \sum_{r=1}^{R} \bar{\chi}^{[r]} \ge \sum_{r=1}^{R} \chi^{[r]}(t) \\
&= \sum_{r=1}^{R} \sum_{k=1}^{d} \mathbf{1}[v_k (-1)^{y_r} \ge 0] \iota_{r,k}\ |v_k| \\
&= \sum_{k=1}^{d} |v_k| \sum_{r=1}^{R} \mathbf{1}[v_k (-1)^{y_r} \ge 0] \iota_{r,k} \\
&= \sum_{k=1}^{d} |v_k|\ \iota_k^{(\mathbf{1}[v_k < 0])},
\end{aligned}$$

where

$$\iota_k^{(c)} := \sum_{r=1}^{R} \mathbf{1}[(-1)^{c+y_r} \ge 0] \iota_{r,k}.$$

The bound $\bar{\chi}$ is constant between bounce events and only depends on the magnitude of v. If we further assume that we use restricted refreshment then this bound is valid for any $t > 0$ allowing us to implement Algorithm 6 using (20) or (21).

3.2 Sampling the thinned factor

We show here how to implement Step 5(c)ii of Algorithm 6 without enumerating over the R datapoints. We begin by introducing some required pre-computed data structures. The pre-computation is executed only once at the beginning of the algorithm, so its running time, $O(R \log R)$ is considered negligible (the number of bouncing events is assumed to be greater than R). For each dimensionality k and class label c, consider the categorical distribution with the following probability mass function over the datapoints:

$$\mu_k^{(c)}(r) = \frac{\iota_{r,k} \mathbf{1}[y_r = c]}{\iota_k^{(c)}}.$$

This is just the distribution over the datapoints that have the given label, weighted by the covariate k. An alias sampling data-structure [3, Section 3.4] is computed for each k and c. This pre-computation takes total time $O(R \log R)$. This allows subsequently to sample in time $O(1)$ from the distributions $\mu_k^{(c)}$.

We now show how this pre-computation is used to to implement Step 5(c)ii of Algorithm 6. We denote the probability mass function we want to sample from by

$$q(r) = \frac{\bar{\chi}^{[r]}}{\bar{\chi}}.$$

To sample this distribution efficiently, we construct an artificial joint distribution over both datapoints and covariate dimension indices

$$q(r,k) = \frac{\mathbf{1}[v_k(-1)^{y_r} \geq 0] \iota_{r,k} \ |v_k|}{\bar{\chi}}.$$

We denote by $q_{\mathrm{k}}(k)$, respectively $q_{\mathrm{r|k}}(r|k)$, the associated marginal, respectively conditional distribution. By construction, we have

$$\sum_{k=1}^{d} q(r,k) = q(r).$$

It is therefore enough to sample (r,k) and to return r. To do so, we first sample (a) $k \sim q_{\mathrm{k}}(\cdot)$ and then (b) sample $r|k \sim q_{\mathrm{r|k}}(\cdot|k)$.

For (a), we have

$$q_{\mathrm{k}}(k) = \frac{|v_k| \ \iota_k^{(\mathbf{1}[v_k<0])}}{\bar{\chi}},$$

so this sampling step again does not require looping over the datapoints thanks the pre-computations described earlier.

For (b), we have

$$q_{\mathrm{r|k}}(r|k) = \mu_k^{(\mathbf{1}[v_k<0])}(r),$$

and therefore this sampling step can be computed in $O(1)$ thanks to the pre-computed alias sampling data structure.

3.3 Algorithm description

Algorithm 1 contains a detailed implementation of the local BPS with thinning for the logistic regression example (Example 4.6).

Algorithm 1 Local BPS algorithm for Logistic Regression with Large Datasets

1. Precompute the alias tables $\mu_k^{(c)}(r)$ for $k = 1, \ldots, d$, $c \in \{0, 1\}$ in order to sample from $q_{\mathrm{r|k}}(\cdot|k)$.
2. Initialize $\left(x^{(0)}, v^{(0)}\right)$ arbitrarily on $\mathbb{R}^d \times \mathbb{R}^d$.
3. Initialize the global clock $T \leftarrow 0$.
4. Initialize $\bar{T} \leftarrow \triangle$. (time until which local upper bounds are valid)
5. Compute local-in-time upper bound on the prior factor as $\bar{\chi}_{\mathrm{prior}} = \sigma^{-2} \max\left(0, \left\langle x^{(0)} + v^{(0)}\Delta, v^{(0)} \right\rangle\right)$ (notice the rate associated with the prior is monotonically increasing).
6. While more events $i = 1, 2, \ldots$ requested do
   (a) Compute the local-in-time upper bound on the data factors in $\mathcal{O}(d)$

$$\bar{\chi} = \sum_{k=1}^{d} |v_k^{(i-1)}| \ \iota_k^{(\mathbf{1}[v_k^{(i-1)}<0])}.$$

   (b) Sample $\tau \sim \mathrm{Exp}\left(\bar{\chi}_{\mathrm{prior}} + \bar{\chi} + \lambda^{\mathrm{ref}}\right)$.

(c) If $\left(T+\tau > \bar{T}\right)$ then

i. $x^{(i)} \leftarrow x^{(i-1)} + v^{(i-1)}(\bar{T} - T)$.

ii. $v^{(i)} \leftarrow v^{(i-1)}$.

iii. Compute the local-in-time upper bound $\bar{\chi}_{\text{prior}} = \sigma^{-2} \max\left(0, \left\langle x^{(i)} + v^{(i)}\Delta, v^{(i)} \right\rangle\right)$.

iv. Set $T \leftarrow \bar{T}$, $\bar{T} \leftarrow \bar{T} + \triangle$.

(d) Else

i. $x^{(i)} \leftarrow x^{(i-1)} + v^{(i-1)}\tau$.

ii. Sample j from Discrete$(\frac{\bar{\chi}}{\bar{\chi}_{\text{prior}} + \bar{\chi} + \lambda^{\text{ref}}}, \frac{\lambda^{\text{ref}}}{\bar{\chi}_{\text{prior}} + \bar{\chi} + \lambda^{\text{ref}}}, \frac{\bar{\chi}_{\text{prior}}}{\bar{\chi}_{\text{prior}} + \bar{\chi} + \lambda^{\text{ref}}})$.

iii. If $j = 1$

A. Sample k according to $q_{\text{k}}(k) = |v_k^{(i-1)}| \, \iota_k^{(\mathbf{1}[v_k^{(i-1)} < 0])} / \bar{\chi}$.

B. Sample $r \sim q_{\text{r|k}}(\cdot|k)$ using the precomputed alias table.

C. If $V < \frac{\max\left(0, \left\langle \nabla U^{[r]}(x^{(i)}), v^{(i-1)} \right\rangle\right)}{\bar{\chi}^{[r]}}$ where $V \sim \mathcal{U}(0,1)$.

$v^{(i)} \leftarrow R_r(x^{(i)}) v^{(i-1)}$ where R_r is the bouncing operator associated with the r-th data item .

D. Else $v^{(i)} \leftarrow v^{(i-1)}$.

iv. If $j = 2$

A. $v^{(i)} \sim \mathcal{N}(0_d, I_d)$.

v. If $j = 3$

A. If $V < \frac{\sigma^{-2} \max\left(0, \left\langle x^{(i)}, v^{(i-1)} \right\rangle\right)}{\chi_{\text{prior}}}$ where $V \sim \mathcal{U}(0,1)$.

$v^{(i)} \leftarrow R_{\text{prior}}(x^{(i)}) v^{(i-1)}$ where R_{prior} is the bouncing operator associated with the prior.

B. Else $v^{(i)} \leftarrow v^{(i-1)}$.

vi. Compute the local-in-time upper bound

$$\bar{\chi}_{\text{prior}} = \sigma^{-2} \max\left(0, \left\langle x^{(i)} + v^{(i)}\left(\bar{T} - T - \tau\right), v^{(i)} \right\rangle\right).$$

vii. Set $T \leftarrow T + \tau$.

4 Direct proof of invariance

Let μ_t be the law of $z(t)$. In the following, we prove invariance by explicitly verifying that the time evolution of the density $\frac{\mathrm{d}\mu_t}{\mathrm{d}t} = 0$ is zero if the initial distribution μ_0 is given by $\rho(z) = \pi(x)\psi(v)$ in Proposition 1. This is achieved by deriving the forward Kolmogorov equation describing the evolution of the marginal density of the stochastic process. For simplicity, we start by presenting the invariance argument when $\lambda^{\mathrm{ref}} = 0$.

Notation and description of the algorithm. We denote a pair of position and velocity by $z = (x, v) \in \mathbb{R}^d \times \mathbb{R}^d$ and we denote translations by $\Phi_t(z) = (\Phi_t^{\mathrm{pos}}(z), \Phi_t^{\mathrm{dir}}(z)) = (x + vt, v)$. The time of the first bounce coincides with the first arrival T_1 of a PP with intensity $\chi(t) = \lambda(\Phi_t(z))$ where:

$$\lambda(z) = \max\{0, \langle \nabla U(x), v\rangle\}. \tag{34}$$

It follows that the probability of having no bounce in the interval $[0, t]$ is given by:

$$\mathrm{No}_t(z) = \exp\left(-\int_0^t \lambda(\Phi_s(z))\mathrm{d}s\right), \tag{35}$$

and the density of the random variable T_1 is given by:

$$q(t_1; z) = \quad \mathbf{1}[t_1 > 0]\tfrac{\mathrm{d}}{\mathrm{d}t_1}(1 - \mathrm{No}_{t_1}(z)) \tag{36}$$

$$= \quad \mathbf{1}[t_1 > 0]\mathrm{No}_{t_1}(z)\lambda(\Phi_{t_1}(z)). \tag{37}$$

If a bounce occurs, then the algorithm follows a translation path for time T_1, at which point the velocity is updated using a bounce operation $C(z)$, defined as:

$$C(z) = (x, R(x)v) \tag{38}$$

where

$$R(x)v = v - 2\frac{\langle \nabla U(x), v\rangle \nabla U(x)}{\|\nabla U(x)\|^2}. \tag{39}$$

The algorithm then continues recursively for time $t - T_1$, in the following sense: a second bounce time T_2 is simulated by adding to T_1 a random increment with density $q(\cdot; C \circ \Phi_{t_1}(z))$. If $T_2 > t$, then the output of the algorithm is $\Phi_{t-t_1} \circ C \circ \Phi_{t_1}(z)$, otherwise an additional bounce is simulated,

etc. More generally, given an initial point z and a sequence $\mathbf{t} = (t_1, t_2, \dots)$ of bounce times, the output of the algorithm at time t is given by:

$$\Psi_{\mathbf{t},t}(z) = \begin{cases} \Phi_t(z) & \text{if } t_1 > 0 \text{ or } \mathbf{t} = (\,), \\ \Psi_{\mathbf{t}',t-t_1}(z) \circ C \circ \Phi_{t_1}(z) & \text{otherwise,} \end{cases} \tag{40}$$

where $(\,)$ denotes the empty list and $\mathbf{t}'$ the suffix of $\mathbf{t}$: $\mathbf{t}' = (t_2, t_3, \dots)$. As for the bounce times, they are distributed as follows:

$$T_1 \sim q(\,\cdot\,; z) \tag{41}$$

$$T_i - T_{i-1} | T_{1:i-1} \sim q\Big(\,\cdot\,;\ \underbrace{\Psi_{T_{1:i-1},T_{i-1}}(z)}_{\text{Pos. after collision } i-1}\ \Big), \quad i \in \{2, 3, 4, \dots\} \tag{42}$$

where $T_{1:i-1} = (T_1, T_2, \dots, T_{i-1})$.

Decomposition by the number of bounces. Let h denote an arbitrary non-negative measurable test function. We show how to decompose expectations of the form $\mathbb{E}[h(\Psi_{\mathbf{T},t}(z))]$ by the number of bounces in the interval $(0, t)$. To do so, we introduce a function $\#\mathrm{Col}_t(\mathbf{t})$, which returns the number of bounces in the interval $(0, t)$:

$$\#\mathrm{Col}_t(\mathbf{t}) = \min\{n \geq 1 : t_n > t\} - 1. \tag{43}$$

From this, we get the following decomposition:

$$\mathbb{E}[h(\Psi_{\mathbf{T},t}(z))] = \mathbb{E}[h(\Psi_{\mathbf{T},t}(z)) \sum_{n=0}^{\infty} \mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = n]] \tag{44}$$

$$= \sum_{n=0}^{\infty} \mathbb{E}[h(\Psi_{\mathbf{T},t}(z)) \mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = n]]. \tag{45}$$

On the event that no bounce occurs in the interval $[0, t)$, i.e. $\#\mathrm{Col}_t(\mathbf{T}) = 0$, the function $\Psi_{\mathbf{T},t}(z)$ is equal to $\Phi_t(z)$, therefore:

$$\mathbb{E}[h(\Psi_{\mathbf{T},t}(z)) \mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = 0]] = h(\Phi_t(z)) \mathbb{P}(\#\mathrm{Col}_t(\mathbf{T}) = 0) \tag{46}$$

$$= h(\Phi_t(z)) \mathrm{No}_t(z). \tag{47}$$

Indeed, on the event that $n \geq 1$ bounces occur, the random variable $h(\Phi_t(z))$ only depends on a

finite dimensional random vector, $(T_1, T_2, \dots, T_n)$, so we can write the expectation as an integral with respect to the density $\widetilde{q}(t_{1:n}; t, z)$ of these variables:

$$\begin{aligned}
&\mathbb{E}[h(\Psi_{\mathbf{T},t}(z))\mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = n]] && (48)\\
&= \mathbb{E}\left[h(\Psi_{\mathbf{T},t}(z))\mathbf{1}[0 < T_1 < \cdots < T_n < t < T_{n+1}]\right]\\
&= \int \cdots \int_{0<t_1<\cdots<t_n<t<t_{n+1}} h(\Psi_{t_{1:n},t}(z)) q(t_1; z) \prod_{i=2}^{n+1} q(t - t_{i-1}; \Psi_{t_{1:i-1},t_{i-1}}(z)) \mathrm{d}t_{1:n+1}\\
&= \int \cdots \int_{0<t_1<\cdots<t_n<t} h(\Psi_{t_{1:n},t}(z)) \widetilde{q}(t_{1:n}; t, z) \mathrm{d}t_{1:n}, && (49)
\end{aligned}$$

where:

$$\widetilde{q}(t_{1:n}; t, z) = q(t_1; z) \times \begin{cases} \mathrm{No}_{t-t_1}(\Phi_{t_1}(z)) & \text{if } n = 1 \\ \mathrm{No}_{t-t_n}(\Phi_{t_{1:n},t_n}(z)) \quad \prod_{i=2}^{n} q\left(t_i - t_{i-1}; \Psi_{t_{1:i-1},t_{i-1}}(z)\right) & \text{if } n \geq 2. \end{cases}$$

To include Equations (47) and (49) under the same notation, we define $t_{1:0}$ to the empty list, $(\,)$, $\widetilde{q}(\,(\,); t, z) = \mathrm{No}_t(z)$, and abuse the integral notation so that for all $n \in \{0, 1, 2, \dots\}$:

$$\mathbb{E}[h(\Psi_{\mathbf{T},t}(z))\mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = n]] = \int \cdots \int_{0<t_1<\cdots<t_n<t} h(\Psi_{t_{1:n},t}(z)) \widetilde{q}(t_{1:n}; t, z) \mathrm{d}t_{1:n}. \qquad (50)$$

Marginal density. Let us fix some arbitrary time $t > 0$. We seek a convenient expression for the marginal density at time t, $\mu_t(z)$, given an initial vector $Z \sim \rho$, where ρ is the hypothesized stationary density $\rho(z) = \pi(x)\,\psi(v)$ on Z. To do so, we look at the expectation of an arbitrary non-negative measurable test function h:

$$\begin{aligned}
\mathbb{E}[h(\Psi_{\mathbf{T},t}(Z))] &= \mathbb{E}\Big[\mathbb{E}[h(\Psi_{\mathbf{T},t}(Z))|Z]\Big] && (51)\\
&= \sum_{n=0}^{\infty} \mathbb{E}\Big[\mathbb{E}[h(\Psi_{\mathbf{T},t}(Z))\mathbf{1}[\#\mathrm{Col}_t(\mathbf{T}) = n]|Z]\Big] && (52)\\
&= \sum_{n=0}^{\infty} \int_{\mathcal{Z}} \rho(z) \int \cdots \int_{0<t_1<\cdots<t_n<t} h(\Psi_{t_{1:n},t}(z)) \widetilde{q}(t_{1:n}; t, z) \mathrm{d}t_{1:n} \mathrm{d}z && (53)\\
&= \sum_{n=0}^{\infty} \int \cdots \int_{0<t_1<\cdots<t_n<t} \int_{\mathcal{Z}} \rho(z) h(\Psi_{t_{1:n},t}(z)) \widetilde{q}(t_{1:n}; t, z) \mathrm{d}z \mathrm{d}t_{1:n} && (54)\\
&= \sum_{n=0}^{\infty} \int \cdots \int_{0<t_1<\cdots<t_n<t} \int_{\mathcal{Z}} \rho(\Psi^{-1}_{t_{1:n},t}(z')) h(z') \widetilde{q}(t_{1:n}; t, \Psi^{-1}_{t_{1:n},t}(z')) \left|\det D\Psi^{-1}_{t_{1:n},t}\right| \mathrm{d}z' \mathrm{d}t_{1:n} && (55)\\
&= \int_{\mathcal{Z}} h(z') \underbrace{\sum_{n=0}^{\infty} \int \cdots \int_{0<t_1<\cdots<t_n<t} \rho(\Psi^{-1}_{t_{1:n},t}(z')) \widetilde{q}(t_{1:n}; t, \Psi^{-1}_{t_{1:n},t}(z')) \mathrm{d}t_{1:n}}_{\mu_t(z')} \mathrm{d}z'.
\end{aligned}$$

We used the following in the above derivation successively the law of total expectation, equation (45), equation (51), Tonelli's theorem and the change of variables, $z' = \Psi_{t_{1:n},t}(z)$, justified since for any fixed $0 < t_1 < t_2 < \cdots < t_n < t < t_{n+1}$, $\Psi_{t_{1:n},t}(\cdot)$ is a bijection (being a composition

of bijections). Now the absolute value of the determinant is one since $\Psi_{\mathbf{t},t}(z)$ is a composition of unit-Jacobian mappings and, by using Tonelli's theorem again, we obtain that the expression above the brace is necessarily equal to $\mu_t(z')$ since h is arbitrary.

Derivative. Our goal is to show that for all $z' \in \mathcal{Z}$

$$\frac{\mathrm{d}\mu_t(z')}{\mathrm{d}t} = 0.$$

Since the process is time homogeneous, once we have computed the derivative, it is enough to show that it is equal to zero at $t = 0$. To do so, we decompose the computation according to the terms I_n in Equation (55):

$$\mu_t(z') = \sum_{n=0}^{\infty} I_n(z', t) \tag{56}$$

$$I_n(z', t) = \int \cdots \int_{0<t_1<\cdots<t_n<t} \rho(\Psi^{-1}_{t_{1:n},t}(z'))\widetilde{q}(t_{1:n}; t, \Psi^{-1}_{t_{1:n},t}(z'))\mathrm{d}t_{1:n}. \tag{57}$$

The categories of terms in Equation (56) to consider are:

No bounce: $n = 0$, $\Psi_{t_{1:n},t}(z) = \Phi_t(z)$, or,

Exactly one bounce: $n = 1$, $\Psi_{t_{1:n},t}(z) = F_{t,t_1} := \Phi_{t-t_1} \circ C \circ \Phi_{t_1}(z)$ for some $t_1 \in (0, t)$, or,

Two or more bounces: $n \geq 2$, $\Psi_{t_{1:n},t}(z) = \Psi_{t-t_2} \circ C \circ F_{t_2,t_1}(z)$ for some $0 < t_1 < t_2 < t$

In the following, we show that the derivative of the terms in the third category, $n \geq 2$, are all equal to zero, while the derivative of the first two categories cancel each other.

No bounce in the interval. From Equation (47):

$$I_0(z', t) = \rho(\Phi_{-t}(z'))\mathrm{No}_t(\Phi_{-t}(z')). \tag{58}$$

We now compute the derivative at zero of the above expression:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} I_0(z', t)\Big|_{t=0} = \mathrm{No}_0 \quad & (\Phi_0(z'))\ \frac{\mathrm{d}\rho(\Phi_{-t}(z'))}{\mathrm{d}t}\Big|_{t=0} + \\ & \rho(\Phi_0(z'))\ \frac{\mathrm{dNo}_t(\Phi_{-t}(z'))}{\mathrm{d}t}\Big|_{t=0}\end{aligned} \tag{59}$$

The first term in the above equation can be simplified as follows:

$$\begin{aligned}
\mathrm{No}_0(\Phi_0(z'))\frac{\mathrm{d}\rho(\Phi_{-t}(z'))}{\mathrm{d}t} &= \frac{\mathrm{d}\rho(\Phi_{-t}(z'))}{\mathrm{d}t} && (60)\\
&= \left\langle \frac{\partial\rho(\Phi_{-t}(z'))}{\partial\Phi^{\mathrm{pos}}_{-t}(z')}, \frac{\mathrm{d}\Phi^{\mathrm{pos}}_{-t}(z')}{\mathrm{d}t} \right\rangle + \\
&\quad \left\langle \frac{\partial\rho(\Phi_{-t}(z'))}{\partial\Phi^{\mathrm{dir}}_{-t}(z')}, \underbrace{\frac{\mathrm{d}\Phi^{\mathrm{dir}}_{-t}(z')}{\mathrm{d}t}}_{=\mathbf{0}} \right\rangle && (61)\\
&= \left\langle \frac{\partial\rho(z)}{\partial x}, -v' \right\rangle && (62)\\
&= \left\langle \frac{\partial}{\partial x}\frac{1}{Z}\exp\left(-U(x)\right)\psi\left(v\right), -v' \right\rangle \\
&= \rho(\Phi_{-t}(z'))\left\langle \nabla U(x), v' \right\rangle, && (63)
\end{aligned}$$

where $x = \Phi^{\mathrm{pos}}_{-t}(z')$. The second term in Equation (59) is equal to:

$$\begin{aligned}
\rho(\Phi_0(z'))\left.\frac{\mathrm{dNo}_t(\Phi_{-t}(z'))}{\mathrm{d}t}\right|_{t=0} &= -\rho(\Phi_0(z'))\mathrm{No}_0(z')\lambda(\Phi_0(z')) && (64)\\
&= -\rho(z')\lambda(z'), && (65)
\end{aligned}$$

using Equation (37). In summary, we have:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}I_0(z',t)\right|_{t=0} = \rho(z')\left\langle \nabla U(x'), v' \right\rangle - \rho(z')\lambda(z').$$

Exactly one bounce in the interval. From Equation (49), the trajectory consists in a bounce at a time T_1, occurring with density (expressed as before as a function of the final point z') $q(t_1; F^{-1}_{t,t_1}(z'))$, followed by no bounce in the interval $(T_1, t]$, an event of probability:

$$\begin{aligned}
\mathrm{No}_{t-t_1}(C\circ\Phi_{t_1}(z)) &= \mathrm{No}_{t-t_1}(C\circ\Phi_{t_1}\circ F^{-1}_{t,t_1}(z')) && (66)\\
&= \mathrm{No}_{t-t_1}(\Phi_{t_1-t}(z')), && (67)
\end{aligned}$$

where we used that $C^{-1} = C$. This yields:

$$I_1(z',t) = \int_0^t q(t_1; F^{-1}_{t,t_1}(z'))\rho(\Psi^{-1}_{t_{1:1},t}(z'))\mathrm{No}_{t-t_1}(\Phi_{t_1-t}(z'))\mathrm{d}t_1. \quad (68)$$

To compute the derivative of the above equation at zero, we use again Leibniz's rule:

$$\frac{\mathrm{d}}{\mathrm{d}t} I_1(z', t)\Big|_{t=0} = \rho(C(z'))\lambda(C(z')).$$

Two or more bounces in the interval. For a number of bounce, we get:

$$I_n(z', t) = \int_0^t \Bigg[\underbrace{\int \cdots \int_{t_{2:n}: t_1 < t_2 \cdots < t_n < t} \rho(\Psi^{-1}_{t_{1:n},t}(z'))\widetilde{q}(t_{1:n}; t, \Psi^{-1}_{t_{1:n},t}(z'))\mathrm{d}t_{2:n}}_{\tilde{I}(t_1, t, z')} \Bigg] \mathrm{d}t_1, \tag{69}$$

and hence, using Leibniz's rule on the integral over t_1:

$$\frac{\mathrm{d}}{\mathrm{d}t} I_n(z', t)\Big|_{t=0} = \tilde{I}(0, 0, z') = 0. \tag{70}$$

Putting all terms together. Putting everything together, we obtain:

$$\frac{\mathrm{d}\mu_t(z')}{\mathrm{d}t}\Big|_{t=0} = \rho(z') \left\langle \nabla U(x'), v' \right\rangle \underbrace{-\rho(z')\lambda(z') + \rho(C(z'))\lambda(C(z'))}. \tag{71}$$

From the expression of $\lambda(\cdot)$, we can rewrite the two terms above the brace as follows:

$$\begin{aligned}
&- \rho(z')\lambda(z') + \rho(C(z'))\lambda(C(z')) \\
=& - \rho(z')\lambda(z') + \rho(z')\lambda(C(z')) \\
=& - \rho(z') \max\{0, \langle \nabla U(x'), v' \rangle\} + \rho(z') \max\{0, \langle \nabla U(x'), R\left(x'\right) v' \rangle\} \\
=& - \rho(z') \max\{0, \langle \nabla U(x'), v' \rangle\} + \rho(z') \max\{0, \langle \nabla U(x'), R\left(x'\right) v' \rangle\} \\
=& - \rho(z') \max\{0, \langle \nabla U(x'), v' \rangle\} + \rho(z') \max\{0, - \langle \nabla U(x'), v' \rangle\} \\
=& - \rho(z') \left\langle \nabla U(x'), v' \right\rangle,
\end{aligned}$$

where we used that $\rho(z') = \rho(C(z'))$, $\langle \nabla U(x'), R\left(x'\right) v' \rangle = - \langle \nabla U(x'), v' \rangle$ and $-\max\{0, f\} + \max\{0, -f\} = -f$ for any function f. Hence we have $\frac{\mathrm{d}\mu_t(z')}{\mathrm{d}t}\Big|_{t=0} = 0$, establishing that that the bouncy particle sampler $\lambda^{\mathrm{ref}} = 0$ admits ρ as invariant distribution. The invariance for $\lambda^{\mathrm{ref}} > 0$ then follows from Lemma 4 given below.

Lemma 4. *Suppose P_t is a continuous time Markov kernel and Q is a discrete time Markov kernel which are both invariant with respect to μ. Suppose we construct for $\lambda^{ref} > 0$ a Markov process $\hat{P}_t$ as follows: at the jump times of an independent PP with intensity λ^{ref} we make a transition with*

Q and then continue according to P_t, then $\hat{P}_t$ is also μ-invariant.

Proof. The transition kernel is given by

$$\begin{aligned}\hat{P}_t &= e^{-\lambda t}P_t + \int_0^t \mathrm{d}t_1 \lambda e^{\lambda t_1} e^{-\lambda(t-t_1)} P_{t-t_1} Q P_{t_1} \\ &\quad + \int_0^t \mathrm{d}t_1 \int_{t_1}^{t_2} \mathrm{d}t_2 \lambda^2 e^{\lambda t_1} e^{\lambda(t_2-t_1)} e^{-\lambda(t-t_2)} P_{t-t_2} Q P_{t_2-t_1} Q P_{t_1} + \ldots\end{aligned}$$

Therefore

$$\begin{aligned}\mu\hat{P}_t &= \mu\left(e^{-\lambda t} + \lambda t e^{-\lambda t} + \frac{(\lambda t)^2}{2} e^{-\lambda t} \ldots\right) \\ &= \mu.\end{aligned}$$

Hence $\hat{P}_t$ is μ-invariant. □

5 Invariance of the local sampler

The generator of the local BPS is given by

$$\begin{aligned}\mathcal{L}h(z) &= \langle \nabla_x h(x,v), v\rangle \\ &\quad + \sum_{f\in F} \lambda_f(x,v)\{h(x, R_f(x)v) - h(x,v)\} \\ &\quad + \lambda^{\mathrm{ref}} \int (h(x,v') - h(x,v))\,\psi(\mathrm{d}v').\end{aligned} \tag{72}$$

The proof of invariance of the local BPS is very similar to the proof of Propostion 1. We have

$$\begin{aligned}\int \mathcal{L}h(z)\rho(z)\,\mathrm{d}z &= \int\int \langle \nabla_x h(x,v), v\rangle \rho(z)\,\mathrm{d}z &(73)\\ &\quad + \int\int \sum_{f\in F} \lambda_f(x,v)\{h(x,R_f(x)v) - h(x,v)\}]\rho(z)\,\mathrm{d}z &(74)\\ &\quad + \lambda^{\mathrm{ref}} \int\int\int (h(x,v') - h(x,v))\,\psi(\mathrm{d}v')\,\rho(z)\,\mathrm{d}z &(75)\end{aligned}$$

where the term (75) is straightforwardly equal to 0 while, by integration by parts, the term (73) satisfies

$$\int\int \langle \nabla_x h(x,v), v\rangle \rho(z)\,\mathrm{d}z = \int\int \langle \nabla U(x), v\rangle h(x,v)\rho(z)\,\mathrm{d}z. \tag{76}$$

as h is bounded. Now a change-of-variables shows that for any $f \in F$

$$\int\int \lambda_f(x,v)\, h(x, R_f(x)\, v)\rho(z)\,\mathrm{d}z = \int\int \lambda(x, R_f(x)\, v)\, h(x,v)\rho(z)\,\mathrm{d}z \tag{77}$$

as $R_f^{-1}(x)\, v) = R(x)\, v$ and $\|R_f(x)\, v\| = \|v\|$ implies $\psi(R_f(x)\, v) = \psi(v)$. So the term (74) satisfies

$$\begin{aligned}
& \int\int \sum_{f\in F} \lambda_f(x,v)\,\{h(x, R_f(x)\, v) - h(z)\}]\rho(z)\,\mathrm{d}z \\
= & \int\int \sum_{f\in F} [\lambda(x, R_f(x)\, v) - \lambda(x,v)]h(x,v)\rho(z)\,\mathrm{d}z \\
= & \int\int \sum_{f\in F} [\max\{0, \langle \nabla U_f(x), R(x)\, v\rangle\} - \max\{0, \langle \nabla U_f(x), v\rangle\}]h(x,v)\rho(z)\,\mathrm{d}z \\
= & \int\int \sum_{f\in F} [\max\{0, -\langle \nabla U_f(x), v\rangle\} - \max\{0, \langle \nabla U_f(x), v\rangle\}]h(x,v)\rho(z)\,\mathrm{d}z \\
= & -\int\int \sum_{f\in F} [\langle \nabla U_f(x), v\rangle]h(x,v)\rho(z)\,\mathrm{d}z \\
= & -\int\int \langle \nabla U(x), v\rangle]h(x,v)\rho(z)\,\mathrm{d}z,
\end{aligned} \tag{78}$$

where we have used $\langle \nabla U_f(x), R_f(x)\, v\rangle = -\langle \nabla U_f(x), v\rangle$ and $\max\{0,-f\} - \max\{0,f\} = -f$ for any f. Hence, summing (76)-(78)-(75), we obtain $\int \mathcal{L}h(z)\rho(z)\,\mathrm{d}z = 0$ and the result follows by [2, Proposition 34.7].

6 Calculations in the isotropic normal case

As we do not use refreshment, it follows from the definition of the collision operator that

$$\begin{aligned}
\left\langle x^{(i)}, v^{(i)} \right\rangle &= \left\langle x^{(i)}, v^{(i-1)} - \frac{2\left\langle x^{(i)}, v^{(i-1)}\right\rangle}{\left\|x^{(i)}\right\|^2} x^{(i)} \right\rangle \\
&= -\left\langle x^{(i)}, v^{(i-1)} \right\rangle = -\left\langle x^{(i-1)}, v^{(i-1)} \right\rangle - \tau_i \\
&= \begin{cases} -\sqrt{-\log V_i} & \text{if } \left\langle x^{(i-1)}, v^{(i-1)}\right\rangle \le 0 \\ -\sqrt{\left\langle x^{(i-1)}, v^{(i-1)}\right\rangle^2 - \log V_i} & \text{otherwise} \end{cases},
\end{aligned}$$

and therefore

$$\left\|x^{(i)}\right\|^2 = \begin{cases} \left\|x^{(i-1)}\right\|^2 - \left\langle x^{(i-1)}, v^{(i-1)}\right\rangle^2 - \log V_i & \text{if } \left\langle x^{(i-1)}, v^{(i-1)}\right\rangle \leq 0 \\ \left\|x^{(i-1)}\right\|^2 - \log V_i & \text{otherwise.} \end{cases}.$$

It follows that $\left\langle x^{(j)}, v^{(j)}\right\rangle \leq 0$ for $j > 0$ if $\left\langle x^{(0)}, v^{(0)}\right\rangle \leq 0$ so, in this case, we have

$$\begin{aligned} \left\|x^{(i)}\right\|^2 &= \left\|x^{(i-1)}\right\|^2 - \left\langle x^{(i-1)}, v^{(i-1)}\right\rangle^2 - \log V_i \\ &= \left\|x^{(i-1)}\right\|^2 + \log V_{i-1} - \log V_i \\ &= \left\|x^{(i-2)}\right\|^2 - \left\langle x^{(i-1)}, v^{(i-1)}\right\rangle^2 - \log V_{i-1} + \log V_{i-1} - \log V_i \\ &\vdots \\ &= \left\|x^{(1)}\right\|^2 - \left\langle x^{(1)}, v^{(1)}\right\rangle^2 - \log V_i \end{aligned}$$

In particular for $x^{(0)} = e_1$ and $v^{(0)} = e_2$ with e_i being elements of standard basis of $\mathbb{R}^d$, the norm of the position at all points along the trajectory can never be smaller than 1.

7 Supplementary information on the evolutionary parameters inference experiments

7.1 Model

We consider an over-parameterized generalized time reversible rate matrix [7] with $d = 10$ corresponding to 4 unnormalized stationary parameters $x_1, \ldots, x_4$, and 6 unconstrained substitution parameters $x_{\{i,j\}}$, which are indexed by sets of size 2, i.e. where $i, j \in \{1, 2, 3, 4\}$, $i \neq j$. Off-diagonal entries of Q are obtained via $q_{i,j} = \pi_j \exp\left(x_{\{i,j\}}\right)$, where

$$\pi_j = \frac{\exp\left(x_j\right)}{\sum_{k=1}^{4} \exp\left(x_k\right)}.$$

We assign independent standard Gaussian priors on the parameters x_i. We assume that a matrix of aligned nucleotides is provided, where rows are species and columns contains nucleotides believed to come from a shared ancestral nucleotide. Given $x = \left(x_1, \ldots, x_4, x_{\{1,2\}}, \ldots, x_{\{3,4\}}\right)$, and hence Q,

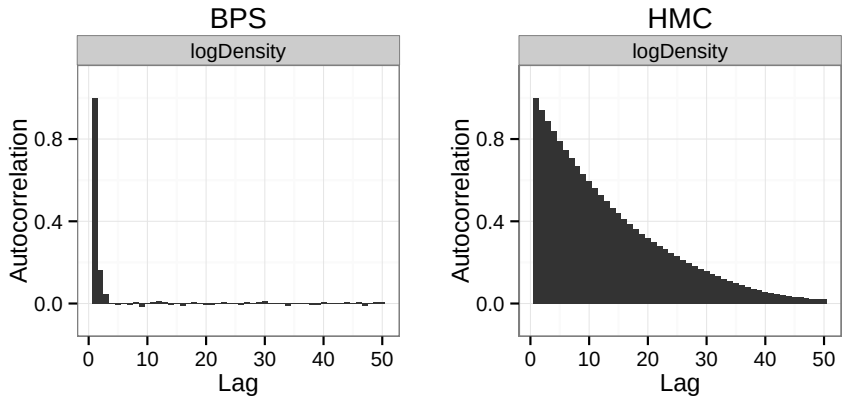

Figure 1: Estimate of the ACF of the log-likelihood statistic for BPS (left) and HMC (right). A similar behavior is observed for the ACF of the other statistics.

the likelihood is a product of conditionally independent continuous time Markov chains over {A, C, G, T}, with "time" replaced by a branching process specified by the phylogenetic tree's topology and branch lengths. The parameter x is unidentifiable, and while this can be addressed by bounded or curved parameterizations, the over-parameterization provides an interesting challenge for sampling methods, which need to cope with the strong induced correlations.

7.2 Baseline

We compare the BPS against a state-of-the-art HMC sampler [8] that uses Bayesian optimization to adapt the the leap-frog stepsize ϵ and trajectory length L of HMC. This sampler was shown in [9] to be comparable or better to other state-of-the-art HMC methods such as NUTS. It also has the advantage of having efficient implementations in several languages. We use the author's Java implementation to compare to our Java implementation of the BPS. Both methods view the objective function as a black box (concretely, a Java interface supporting pointwise evaluation and gradient calculation). In all experiments, we initialize at the mode and use a burn-in of 100 iterations and no thinning. The HMC auto-tuner yielded $\epsilon = 0.39$ and $L = 100$. For our method, we use the global sampler and the global refreshment scheme.

7.3 Additional experimental results

To ensure that BPS outperforming HMC does not come from a faulty auto-tuning of HMC parameters, we look at the ESS/s for the log-likelihood statistic when varying the stepsize ϵ. The results

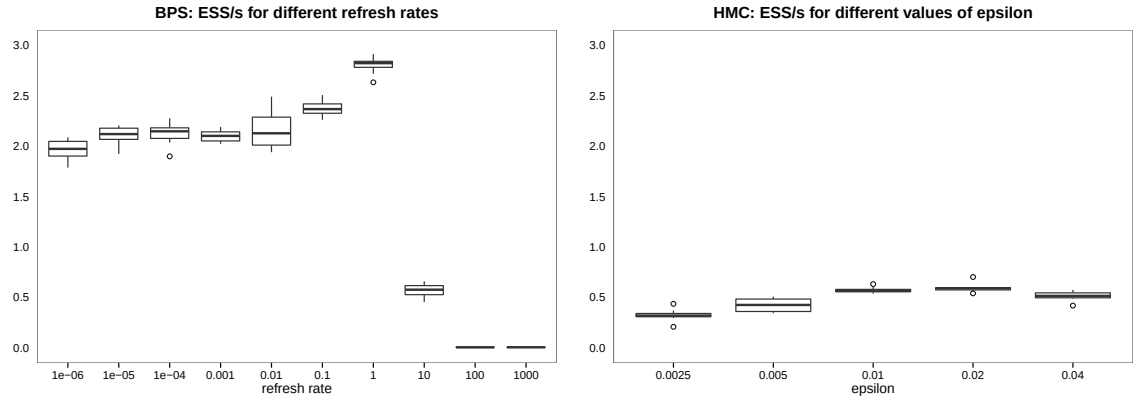

Figure 2: Left: sensitivity of BPS's ESS/s on the log likelihood statistic. Right: sensitivity of HMC's ESS/s on the log likelihood statistic. Each setting is replicated 10 times with different algorithmic random seeds.

in Figure 2 on page 26(right) show that the value selected by the auto-tuner is indeed reasonable, close to the value 0.02 found by brute force maximization. We repeat the experiments with $\epsilon = 0.02$ and obtain the same conclusions. This shows that the problem is genuinely challenging for HMC.

The BPS algorithm also exhibits sensitivity to λ^{ref}. We analyze this dependency in Figure Figure 2 on page 26(left). We observe an asymmetric dependency, where values higher than 1 result in a significant drop in performance, as they bring the sampler closer to random walk behavior. Values one or more orders of magnitudes lower than 1 have a lower detrimental effect. However for a range of values of λ^{ref} covering six orders of magnitudes, BPS outperforms HMC at its optimal parameters.

References

[1] R.J. Adler and J.E. Taylor. *Topological Complexity of Smooth Random Functions*, volume 2019 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2011. École d'Été de Probabilités de Saint-Flour XXXIX.

[2] M.H.A. Davis. *Markov Models & Optimization*, volume 49. CRC Press, 1993.

[3] L. Devroye. *Non-uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

[4] M. Einsiedler and T. Ward. *Ergodic Theory: with a view towards Number Theory*, volume 259 of *Graduate Texts in Mathematics*. Springer-Verlag, London, 2011.

[5] M. Hairer. Convergence of Markov processes. http://www.hairer.org/notes/Convergence.pdf, 2010. Lecture notes, University of Warwick.

[6] E.A. J. F. Peters and G. de With. Rejection-free Monte Carlo sampling for general potentials. *Physical Review E*, 85:026703, 2012.

[7] S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences*, 17:56–86, 1986.

[8] Z. Wang, S. Mohamed, and N. De Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1462–1470, 2013.

[9] T Zhao, Z. Wang, A. Cumberworth, J. Gsponer, N. De Freitas, and A. Bouchard-Côté. Bayesian analysis of continuous time Markov chains with application to phylogenetic modelling. *Bayesian Analysis*, 2015.