The Annals of Statistics
2019, Vol. 47, No. 3, 1288–1320
https://doi.org/10.1214/18-AOS1715
© Institute of Mathematical Statistics, 2019

THE ZIG-ZAG PROCESS AND SUPER-EFFICIENT SAMPLING FOR BAYESIAN ANALYSIS OF BIG DATA[1]

BY JORIS BIERKENS, PAUL FEARNHEAD AND GARETH ROBERTS

Delft University of Technology, Lancaster University and University of Warwick

Standard MCMC methods can scale poorly to big data settings due to the need to evaluate the likelihood at each iteration. There have been a number of approximate MCMC algorithms that use sub-sampling ideas to reduce this computational burden, but with the drawback that these algorithms no longer target the true posterior distribution. We introduce a new family of Monte Carlo methods based upon a multidimensional version of the Zig-Zag process of [*Ann. Appl. Probab.* **27** (2017) 846–882], a continuous-time piecewise deterministic Markov process. While traditional MCMC methods are reversible by construction (a property which is known to inhibit rapid convergence) the Zig-Zag process offers a flexible nonreversible alternative which we observe to often have favourable convergence properties. We show how the Zig-Zag process can be simulated without discretisation error, and give conditions for the process to be ergodic. Most importantly, we introduce a sub-sampling version of the Zig-Zag process that is an example of an *exact approximate scheme*, that is, the resulting approximate process still has the posterior as its stationary distribution. Furthermore, if we use a control-variate idea to reduce the variance of our unbiased estimator, then the Zig-Zag process can be super-efficient: after an initial preprocessing step, essentially independent samples from the posterior distribution are obtained at a computational cost which does not depend on the size of the data.

1. Introduction.

The importance of Markov chain Monte Carlo techniques in Bayesian inference shows no signs of diminishing. However, all commonly used methods are variants on the Metropolis–Hastings (MH) algorithm (Hastings (1970), Metropolis et al. (1953)) and rely on innovations which date back over 60 years. All MH algorithms simulate realisations from a discrete reversible ergodic Markov chain with invariant distribution π which is (or is closely related to) the *target* distribution, that is, the posterior distribution in a Bayesian context. The MH algorithm gives a beautifully simple though flexible recipe for constructing such Markov chains, requiring only local information about π (typically pointwise evaluations of π and, perhaps, its derivative at the current and proposed new locations) to complete each iteration.

Received July 2016; revised March 2018.

[1]Supported by EPSRC Grants EP/D002060/1 (CRiSM) and EP/K014463/1 (iLike).

MSC2010 subject classifications. Primary 65C60; secondary 65C05, 62F15, 60J25.

Key words and phrases. MCMC, nonreversible Markov process, piecewise deterministic Markov process, stochastic gradient Langevin dynamics, sub-sampling, exact sampling.

However, new complex modelling and data paradigms are seriously challenging these established methodologies. First, the restriction of traditional MCMC to reversible Markov chains is a serious limitation. It is now well understood both theoretically (Bierkens (2016), Chen and Hwang (2013), Duncan, Lelièvre and Pavliotis (2016), Hwang, Hwang-Ma and Sheu (1993), Rey-Bellet and Spiliopoulos (2015)) and heuristically (Neal (1998)) that nonreversible chains offer potentially massive advantages over reversible counterparts. The need to escape reversibility, and create momentum to aid mixing throughout the state space is certainly well known, and motivates a number of modern MCMC methods, including the popular Hamiltonian MCMC (Duane et al. (1987)).

A second major obstacle to the application of MCMC for Bayesian inference is the need to process potentially massive data-sets. Since MH algorithms in their pure form require a likelihood evaluation—and thus processing the full data-set—at every iteration, it can be impractical to carry out large numbers of MH iterations. This has led to a range of alternatives that use sub-samples of the data at each iteration (Ma, Chen and Fox (2015), Maclaurin and Adams (2014), Quiroz, Villani and Kohn (2015), Welling and Teh (2011)), or that partition the data into shards, run MCMC on each shard and then attempt to combine the information from these different MCMC runs (Li, Srivastava and Dunson (2017), Neiswanger, Wang and Xing (2013), Scott et al. (2016), Wang and Dunson (2013)). However, most of these methods introduce some form of approximation error, so that the final sample will be drawn from some approximation to the posterior, and the quality of the approximation can be impossible to evaluate. As an exception the Firefly algorithm (Maclaurin and Adams (2014)) samples from the exact distribution of interest (but see the comment below).

This paper introduces the multidimensional Zig-Zag sampling algorithm (ZZ) and its variants. These methods overcome the restrictions of the lifted Markov chain approach of Turitsyn, Chertkov and Vucelja (2011) as they do not depend on the introduction of momentum generating quantities. They are also amenable to the use of sub-sampling ideas. The dynamics of the Zig-Zag process depends on the target distribution through the gradient of the logarithm of the target. For Bayesian applications this is a sum, and is easy to estimate unbiasedly using sub-sampling. Moreover, Zig-Zag with Sub-Sampling (ZZ-SS) retains the exactness of the required invariant distribution. Furthermore, if we also use control variate ideas to reduce the variance of our sub-sampling estimator of the gradient, the resulting Zig-Zag with Control Variates (ZZ-CV) algorithm has remarkable *super-efficient* scaling properties for large data sets.

We will call an algorithm *super-efficient* if it is able to generate independent samples from the target distribution at a higher efficiency than if we would draw independently from the target distribution at the cost of evaluating all data. The only situation we are aware of where we can implement super-efficient sampling is with simple conjugate models, where the likelihood function has a low-dimensional summary statistic which can be evaluated at cost $O(n)$, where n is the number of observations, after which we can obtain independent samples from the posterior distribution at a cost of $O(1)$ by using the functional form of the posterior distribution. The ZZ-CV can replicate this computational efficiency: After a precomputation of $O(n)$, we are able to obtain independent samples at a cost of $O(1)$. In this sense it contrasts with the Firefly algorithm (Maclaurin and Adams (2014)) which has an ESS per datum which decreases approximately as $1/n$ where n is the size of the data, so that the gains of this algorithm do not increase with n; see Bouchard-Côté, Vollmer and Doucet (2017), Section 4.6.

This breakthrough is based upon the Zig-Zag process, a continuous time piecewise deterministic Markov process (PDMP). Given a d-dimensional differentiable target density π, Zig-Zag is a continuous-time nonreversible stochastic process with continuous, piecewise linear trajectories on $\mathbb{R}^d$. It moves with constant velocity, $\Theta \in \{-1, 1\}^d$, until one of the velocity components switches sign. The event time and choice of which direction to reverse is controlled by a collection of state-dependent switching rates, $(\lambda_i)_{i=1}^d$ which in turn are constrained via an identity (2) which ensures that π is a stationary distribution for the process. The process intrinsically is constructed in continuous-time, and it can be easily simulated using standard Poisson thinning arguments as we shall see in Section 3.

The use of PDMPs such as the Zig-Zag processes is an exciting and mostly unexplored area in MCMC. The first occurrence of a PDMP for sampling purposes is in the computational physics literature (Peters and De With (2012)), which in one dimension coincides with the Zig-Zag process. In Bouchard-Côté, Vollmer and Doucet (2017), this method is given the name *Bouncy Particle Sampler*. In multiple dimensions the Zig-Zag process and Bouncy Particle Sampler (BPS) are different processes: Both are PDMPs which move along straight line segments, but the Zig-Zag process changes direction in only a single component at each switch, whereas the Bouncy Particle Sampler reflects the full direction vector in the level curves of the density function. As we will see in Section 2.4, this difference has a beneficial effect on the ergodic properties of the Zig-Zag process. The one-dimensional Zig-Zag process is analysed in detail in, for example, Bierkens and Roberts (2017), Fontbona, Guérin and Malrieu (2012), Fontbona, Guérin and Malrieu (2016), Monmarché (2014).

Since the first version of this paper was conceived, already several other related theoretical and methodological papers have appeared. In particular, we mention here results on exponential ergodicity of the BPS (Deligiannidis, Bouchard-Côté and Doucet (2017)) and ergodicity of the multidimensional Zig-Zag process (Bierkens, Roberts and Zitt (2017)). The Zig-Zag process has the advantage that it is ergodic under very mild conditions, which in particular means that we are not required to choose a refreshment rate. At the same time, the BPS seems more "natural", in that it tries to minimise the bounce rate and the change in direction at bounces, and it may be more efficient for this reason. However, it is a challenge to make a direct comparison in efficiency of the two methods since the efficiency depends both on the computational effort per unit of continuous time of the respective algorithms, as well as the mixing time of the underlying processes. Therefore, we expect analysing the relative efficiency of PDMP based algorithms to be an important area of continued research for years to come.

A continuous-time sequential Monte Carlo algorithm for scalable Bayesian inference with big data, the SCALE algorithm, is given in Pollock et al. (2016). Advantages that Zig-Zag has over SCALE is that it avoids the issue of controlling the stability of importance weights, and it is simpler to implement. Whereas the SCALE algorithm is well adapted for the use of parallel architecture computing, and has particularly simple scaling properties for big data.

1.1. *Notation.* For a topological space X, let $\mathcal{B}(X)$ denote the Borel σ-algebra. We write $\mathbb{R}_+ := [0, \infty)$. If $h : \mathbb{R}^d \to \mathbb{R}$ is differentiable then $\partial_i h$ denotes the function $\xi \mapsto \frac{\partial h(\xi)}{\partial \xi_i}$. We equip $E := \mathbb{R}^d \times \{-1, +1\}^d$ with the product topology of the Euclidean topology on $\mathbb{R}^d$ and the discrete topology on $\{-1, +1\}^d$. Elements in E will often be denoted by (ξ, θ) with $\xi \in \mathbb{R}^d$ and $\theta \in \{-1, +1\}^d$. For $g : E \to \mathbb{R}$ differentiable in its first argument, we will use $\partial_i g$ to denote the function $(\xi, \theta) \mapsto \frac{\partial g(\xi,\theta)}{\partial \xi_i}$, $i = 1, \dots, d$.

2. The Zig-Zag process.

The Zig-Zag process is a continuous time Markov process whose trajectories lie in the space $E = \mathbb{R}^d \times \{-1, +1\}^d$ and will be denoted by $(\Xi(t), \Theta(t))_{t \geq 0}$. They can be described as follows: At random times, a single component of $\Theta(t)$ flips. In between these switches, $\Xi(t)$ is linear with $\frac{d}{dt}\Xi(t) = \Theta(t)$. The rates at which the flips in $\Theta(t)$ occur are time inhomogeneous: The ith component of Θ switches at rate $\lambda_i(\Xi(t), \Theta(t))$, where $\lambda_i : E \to \mathbb{R}_+$ for $i = 1, \dots, d$ are continuous functions.

2.1. *Construction.* For a given $(\xi, \theta) \in E$, we may construct a trajectory of (Ξ, Θ) of the Zig-Zag process with initial condition (ξ, θ) as follows:

- Let $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$.
- For $k = 1, 2, \dots$
  - Let $\xi^k(t) := \Xi^{k-1} + \Theta^{k-1} t$, $t \geq 0$.
  - For $i = 1, \dots, d$, let τ_i^k be distributed according to

$$\mathbb{P}(\tau_i^k \geq t) = \exp\left(-\int_0^t \lambda_i(\xi^k(s), \Theta^{k-1})\, ds\right).$$

  - Let $i_0 := \operatorname{argmin}_{i \in \{1,\dots,d\}} \tau_i^k$ and let $T^k := T^{k-1} + \tau_{i_0}^k$.
  - Let $\Xi^k := \xi^k(T^k)$.
  - Let

$$\Theta^k(i) := \begin{cases} \Theta^{k-1}(i) & \text{if } i \neq i_0, \\ -\Theta^{k-1}(i) & \text{if } i = i_0. \end{cases}$$

This procedure defines a sequence of *skeleton points* $(T^k, \Xi^k, \Theta^k)_{k=0}^{\infty}$ in $\mathbb{R}_+ \times E$, which correspond to the time and position at which the direction of the process changes. The trajectory $\xi^k(t)$ represents the position of the process at time $T^{k-1} + t$ until time T^k, for $0 \le t \le T^k - T^{k-1}$. The time until the next skeleton event is characterized as the smallest time of a set of events in d simultaneous point processes, where each point process corresponds to switching of a different component of the velocity. For the ith of these processes, events occur at rate $\lambda_i(\xi^k(s), \Theta^{k-1})$, and τ_i^k is defined to be the time to the first event for the ith component. The component for which the earliest event occurs is i_0. This defines $\tau_{i_0}^k$, the time between the $(k-1)$th and kth skeleton point, and the component, i_0, of the velocity that switches.

The piecewise deterministic trajectories $(\Xi(t), \Theta(t))$ are now obtained as

$$\bigl(\Xi(t), \Theta(t)\bigr) := \bigl(\Xi^k + \Theta^k(t - T^k), \Theta^k\bigr) \qquad \text{for } t \in [T^k, T^{k+1}), k = 0, 1, 2, \ldots.$$

Since the switching rates are continuous, and hence bounded on compact sets, and Ξ will travel a finite distance within any finite time interval, within any bounded time interval there will be finitely many switches almost surely.

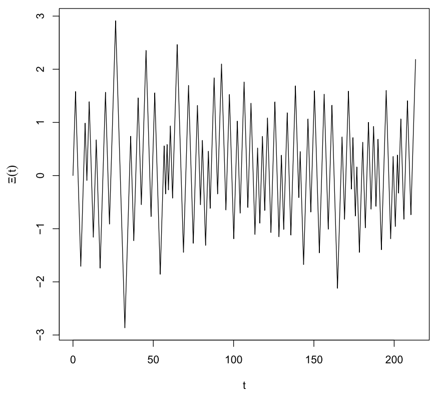
The above procedure provides a mathematical construction of a Markov process as well as the outline of an algorithm which simulates this process. The only step in this procedure which presents a computational challenge is the simulation of the random times (T_i^k) and a significant part of this paper will consider obtaining these in a numerically efficient way.

Figure 1 displays trajectories of the Zig-Zag process for several examples of invariant distributions. The name of the process is derived by the *zig-zag* nature of paths that the process produces. Figure 1 shows an important difference in the output of the Zig-Zag process, as compared to a discrete-time MCMC algorithm: The output of is a continuous-time sample path. The bottom row of plots in Figure 1 also gives a comparison to a reversible MCMC algorithm, Metropolis Adjusted Langevin (MALA, Roberts and Tweedie (1996)), and demonstrates an advantage of a nonreversible sampler: it can cope better with a heavy tailed target. This is most easily seen if we start the process out in the tail, as in the figure. The velocity component of the Zig-Zag process enables it to quickly return to the mode of the distribution, whereas the reversible algorithm behaves like a random walk in the tails, and takes much longer to return to the mode.

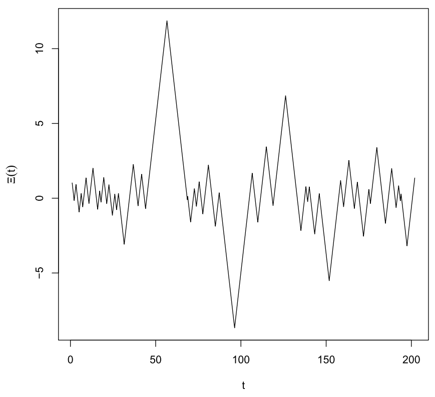
2.2. *Invariant distribution.* The most important aspect of the Zig-Zag process is that in many cases the switching rates are directly related to an easily identifiable invariant distribution. Let $C^1(\mathbb{R}^d)$ denote the space of continuously differentiable functions on $\mathbb{R}^d$. For $\theta \in \{-1, +1\}^d$ and $i \in \{1, \ldots, d\}$, let $F_i[\theta] \in \{-1, +1\}^d$ denote the binary vector obtained by flipping the i-th component of θ, that is,

$$\bigl(F_i[\theta]\bigr)_j = \begin{cases} \theta_j & \text{if } i \neq j, \\ -\theta_j & \text{if } i = j. \end{cases}$$

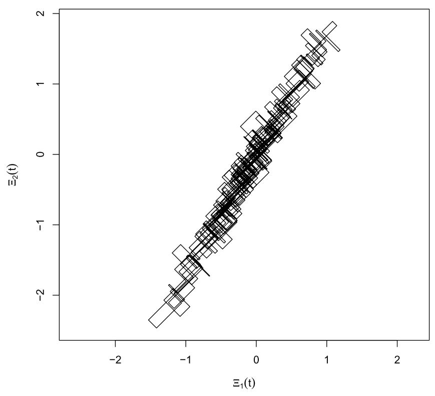
We introduce the following assumption.



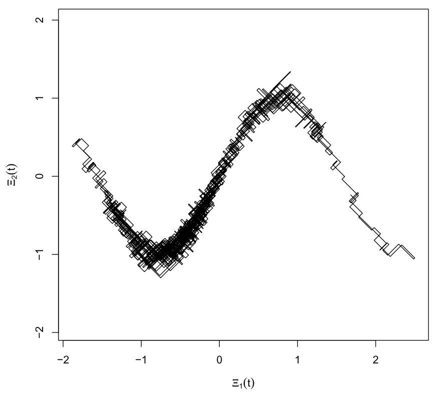
(a) 1D Gaussian



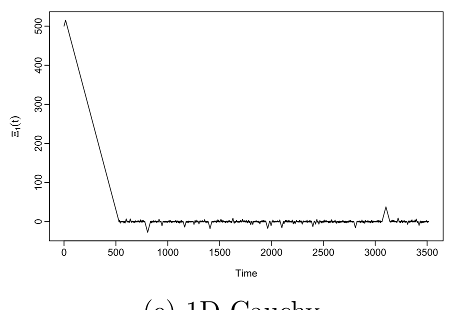
(b) 1D Cauchy



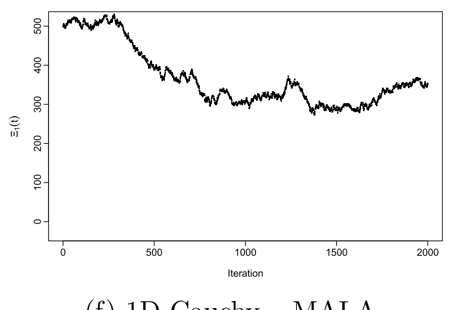
(c) 2D anisotropic Gaussian



(d) 2D S-shaped density



(e) 1D Cauchy



(f) 1D Cauchy – MALA

FIG. 1. *Top two rows*: *example trajectories of the canonical Zig-Zag process. In* (a) *and* (b) *the horizontal axis shows time and the vertical axis the* Ξ*-coordinate of the* 1*D process. In* (c) *and* (d), *the trajectories in* $\mathbb{R}^2$ *of* (Ξ_1, Ξ_2) *are plotted. Bottom row*: *Zig-Zag process* (e) *and MALA* (f) *for a Cauchy target with both processes started in the tail.*

ASSUMPTION 2.1. For some function $\Psi \in C^1(\mathbb{R}^d)$ satisfying

$$\int_{\mathbb{R}^d} \exp\bigl(-\Psi(\xi)\bigr)\, d\xi < \infty \tag{1}$$

we have

$$\lambda_i(\xi, \theta) - \lambda_i\bigl(\xi, F_i[\theta]\bigr) = \theta_i \partial_i \Psi(\xi) \qquad \text{for all } (\xi, \theta) \in E, i = 1, \dots, d. \tag{2}$$

Throughout this paper, we will refer to Ψ as the *negative log density*. Let μ_0 denote the measure on $\mathcal{B}(E)$ such that, for $A \in \mathcal{B}(\mathbb{R}^d)$ and $\theta \in \{-1, +1\}^d$, $\mu_0(A \times \{\theta\}) = \text{Leb}(A)$, with Leb denoting Lebesgue measure on $\mathbb{R}^d$.

THEOREM 2.2. *Suppose Assumption 2.1 holds. Let μ denote the probability distribution on E such that μ has Radon–Nikodym derivative*

$$\frac{d\mu}{d\mu_0}(\xi, \theta) = \frac{\exp(-\Psi(\xi))}{Z}, \qquad (\xi, \theta) \in E, \tag{3}$$

where $Z = \int_E \exp(-\Psi)\, d\mu_0$. Then the Zig-Zag process (Ξ, Θ) with switching rates $(\lambda_i)_{i=1}^d$ has invariant distribution μ.

The proof is located in Section 1 of the Supplementary Material (Bierkens, Fearnhead and Roberts (2019)). We see that under the invariant distribution of the Zig-Zag process, ξ and θ are independent, with ξ having density proportional to $\exp(-\Psi(\xi))$ and θ having a uniform distribution on the points in $\{-1, +1\}^d$.

For $a \in \mathbb{R}$, let $(a)^+ := \max(0, a)$ and $(a)^- := \max(0, -a)$ denote the positive and negative parts of a, respectively. We will often use the trivial identity $a = (a)^+ - (a)^-$ without comment. The following result characterizes the switching rates for which (2) holds.

PROPOSITION 2.3. *Suppose $\lambda : E \to \mathbb{R}_+^d$ is continuous. Then Assumption 2.1 is satisfied if and only if there exists a continuous function $\gamma : E \to \mathbb{R}_+^d$ such that for all $i = 1, \dots, d$ and $(\xi, \theta) \in E$, $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$ and, for $\Psi \in C^1(\mathbb{R}^d)$ satisfying (1),*

$$\lambda_i(\xi, \theta) = \bigl(\theta_i \partial_i \Psi(\xi)\bigr)^+ + \gamma_i(\xi, \theta). \tag{4}$$

The proof is located in Section 1 of the Supplementary Material (Bierkens, Fearnhead and Roberts (2019).

2.3. *Zig-Zag process for Bayesian inference.* One application of the Zig-Zag process is as an alternative to MCMC for sampling from posterior distributions in Bayesian statistics. We show here that it is straightforward to derive a class of Zig-Zag processes that have a given posterior distribution as their invariant distribution.

The dynamics of the Zig-Zag process only depend on knowing the posterior density up to a constant of proportionality.

To keep notation consistent with that used for the Zig-Zag process, let $\xi \in \mathbb{R}^d$ denote a vector of continuous parameters. We are given a prior density function for ξ, which we denote by $\pi_0(\xi)$, and observations $x^{1:n} = (x^1, \dots, x^n)$. Our model for the data defines a likelihood function $L(x^{1:n}|\xi)$. Thus the posterior density function is

$$\pi(\xi) \propto \pi_0(\xi) L(x^{1:n}|\xi).$$

We can write $\pi(\xi)$ in the form of the previous section,

$$\pi(\xi) = \frac{1}{Z} \exp(-\Psi(\xi)), \qquad \xi \in \mathbb{R}^d,$$

where $\Psi(\xi) = -\log \pi_0(\xi) - \log L(x^{1:n}|\xi)$, and $Z = \int_{\mathbb{R}^d} \exp(-\Psi(\xi))\, d\xi$ is the unknown normalising constant. Now assuming that $\log \pi_0(\xi)$ and $\log L(x^{1:n}|\xi)$ are both continuously differentiable with respect to ξ, from (4) a Zig-Zag process with rates

$$\lambda_i(\xi, \theta) = \bigl(\theta_i \partial_i \Psi(\xi)\bigr)^+$$

will have the posterior density $\pi(\xi)$ as the marginal of its invariant distribution. We call the process with these rates the *Canonical Zig-Zag process* for the negative log density Ψ. As explained in Proposition 2.3, we can construct a family of Zig-Zag processes with the same invariant distribution by choosing any set of functions $\gamma_i(\xi, \theta)$, for $i = 1, \dots, d$, which take nonnegative values and for which $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$, and setting

$$\lambda_i(\xi, \theta) = \bigl(\theta_i \partial_i \Psi(\xi)\bigr)^+ + \gamma_i(\xi, \theta) \qquad \text{for } i = 1, \dots, d.$$

The intuition here is that $\lambda_i(\xi, \theta)$ is the rate at which we transition from θ to $F_i[\theta]$. The condition $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$ means that we increase by the same amount both the rate at which we will transition from θ to $F_i[\theta]$ and vice versa. As our invariant distribution places the same probability of being in a state with velocity θ as that of being in state $F_i[\theta]$, these two changes in rate cancel out in terms of their effect on the invariant distribution. Changing the rates in this way does impact the dynamics of the process, with larger γ_i values corresponding to more frequent changes in the velocity of the Zig-Zag process, and we would expect the resulting process to mix more slowly.

Under the assumption that the Zig-Zag process has the desired invariant distribution and is ergodic, it follows from the Birkhoff ergodic theorem that for any bounded continuous function $f : E \to \mathbb{R}$,

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t f\bigl(\Xi(s), \Theta(s)\bigr)\, ds = \int_E f\, d\mu,$$

for any initial condition $(\xi, \theta) \in E$. Sufficient conditions for ergodicity will be discussed in the following section. Taking γ to be positive and bounded everywhere ensures ergodicity, as will be established in Theorem 2.10.

2.4. *Ergodicity of the Zig-Zag process.* We have established in Section 2.2 that for any continuously differentiable, positive density π on $\mathbb{R}^d$ a Zig-Zag process can be constructed that has π as its marginal stationary density. In order for ergodic averages $\frac{1}{T}\int_0^T f(\Xi(s))\,ds$ of the Zig-Zag process to converge asymptotically to $\pi(f)$, we further require $(\Xi(t), \Theta(t))$ to be ergodic, that is, to admit a *unique* invariant distribution.

Ergodicity is directly related to the requirement that $(\Xi(t), \Theta(t))$ is irreducible, that is, the state space is not reducible into components which are each invariant for the process $(\Xi(t), \Theta(t))$. For the one-dimensional Zig-Zag process, (exponential) ergodicity has already been established under mild conditions (Bierkens and Roberts (2017)). As we discuss below, irreducibility, and thus ergodicity, can be established for large classes of multidimensional target distributions, such as i.i.d. Gaussian distributions, and also if the switching rates $\lambda_i(\xi, \theta)$ are positive for all $i = 1, \dots, d$, and $(\xi, \theta) \in E$.

Let $P^t((\xi, \theta), \cdot)$ be the transition kernel of the Zig-Zag process with initial condition (ξ, θ). A function $f : E \to \mathbb{R}$ is called *norm-like* if $\lim_{\|\xi\|\to\infty} f(\xi, \theta) = \infty$ for all $\theta \in \{-1, +1\}^d$. Let $\|\cdot\|_{\mathrm{TV}}$ denote the total variation norm on the space of signed measures. First, we consider the one-dimensional case.

ASSUMPTION 2.4. Suppose $d = 1$ and there exists $\xi_0 > 0$ such that:

(i) $\inf_{\xi \geq \xi_0} \lambda(\xi, +1) > \sup_{\xi \geq \xi_0} \lambda(\xi, -1)$ and
(ii) $\inf_{\xi \leq -\xi_0} \lambda(\xi, -1) > \sup_{\xi \leq -\xi_0} \lambda(\xi, +1)$.

PROPOSITION 2.5 (Bierkens and Roberts (2017), Theorem 5). *Suppose Assumption 2.4 holds. Then there exists a function $f : E \to [1, \infty)$ which is norm-like such that the Zig-Zag process is f-exponentially ergodic, that is, there exists a constant $\kappa > 0$ and $0 < \rho < 1$ such that*

$$\|P^t((\xi, \theta), \cdot) - \pi\|_{\mathrm{TV}} \leq \kappa f(\xi, \theta)\rho^t \qquad \text{for all } (\xi, \theta) \in E \text{ and } t \geq 0.$$

EXAMPLE 2.6. As an example of fundamental importance, which will also be used in the proof of Theorem 2.10, consider a one-dimensional Gaussian distribution. For simplicity, let $\pi(\xi)$ be centred, $\pi(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\xi^2}{2\sigma^2})$ for some $\sigma > 0$. According to (4) the switching rates take the form

$$\lambda(\xi, \theta) = (\theta\xi/\sigma^2)^+ + \gamma(\xi), \qquad (\xi, \theta) \in E.$$

As long as γ is bounded from above, Assumption 2.4 is satisfied. In particular, this holds if γ is equal to a nonnegative constant.

REMARK 2.7. We say a probability density function π is of *product form* if $\pi(\xi) = \prod_{i=1}^d \pi_i(\xi_i)$, where $\pi_i : \mathbb{R}^d \to (0, \infty)$ are one-dimensional probability density functions. When its target density is of product form the Zig-Zag process is the concatenation of independent Zig-Zag processes. In this case, the negative log density is of the form $\Psi(\xi) = \sum_{i=1}^d \Psi_i(\xi_i)$ and the switching rate for the ith component of θ is

$$\lambda_i(\xi, \theta) = \left(\theta_i \Psi_i'(\xi_i)\right)^+ + \gamma_i(\xi). \tag{5}$$

As long as $\gamma_i(\xi) = \gamma_i(\xi_i)$, so if $\gamma_i(\xi)$ only depends on the ith coordinate of ξ, the switching rate of coordinate i is independent of the other coordinates ξ_j, $j \neq i$. It follows that the switches of the ith coordinate can be generated by a one-dimensional time inhomogeneous Poisson process, which is independent of the switches in the other coordinates. As a consequence, the d-dimensional Zig-Zag process $(\Xi(t), \Theta(t)) = (\Xi^1(t), \dots, \Xi_d(t), \Theta^1(t), \dots, \Theta^d(t))$ consists of a combination of d independent Zig-Zag processes $(\Xi^i(t), \Theta^i(t))$, $i = 1, \dots, d$.

Suppose $P(x, dy)$ is the transition kernel of a Markov chain on a state space E. We say that the Markov chain associated to P is *mixing* if there exists a probability distribution π on E such that

$$\lim_{k \to \infty} \| P^k(x, \cdot) - \pi \|_{\mathrm{TV}} = 0 \qquad \text{for all } x \in E.$$

For any continuous time Markov process with family of transition kernels $P^t(x, dy)$, we can consider the associated *time-discretized process*, which is a Markov chain with transition kernel $Q(x, dy) := P^\delta(x, dy)$ for a fixed $\delta > 0$. The value of δ will be of no significance in our use of this construction.

PROPOSITION 2.8. *Suppose π is of product form and $\lambda : E \to \mathbb{R}_+^d$ admits the representation* (5) *with $\gamma_i(\xi)$ only depending on $\{\xi_i, i = 1, \dots, d\}$. Furthermore suppose that for every $i = 1, \dots, d$, the one-dimensional time-discretized Zig-Zag process corresponding to switching rate λ_i is mixing in $\mathbb{R} \times \{-1, +1\}$. Then the time-discretized d-dimensional Zig-Zag process with switching rates (λ_i) is mixing. In particular, the multidimensional Zig-Zag process admits a unique invariant distribution.*

PROOF. This follows from the decomposition of the d-dimensional Zig-Zag process as d one-dimensional Zig-Zag processes and Lemma 1.1 in the Supplementary Material. □

EXAMPLE 2.9. Continuing Example 2.6, consider the simple case in which π is of product form with each π_i a centered Gaussian density function with variance σ_i^2. It follows from Proposition 2.8 and Example 2.6 that the multidimensional canonical Zig-Zag process (i.e., the Zig-Zag process with $\gamma_i \equiv 0$) is mixing. This is different from the Bouncy Particle Sampler (Bouchard-Côté, Vollmer and Doucet (2017)), which is not ergodic for an i.i.d. Gaussian without "refreshments" of the momentum variable.

We now show that strict positivity of the rates ensures ergodicity.

THEOREM 2.10. *Suppose $\lambda : E \to (0,\infty)^d$, in particular $\lambda_i(\xi,\theta)$ is positive for all $i = 1,\dots,d$ and $(\xi,\theta) \in E$. Then there exists at most a single invariant measure for the Zig-Zag process with switching rate λ.*

The proof of this result consists of a Girsanov change of measure with respect to a Zig-Zag process targeting an i.i.d. standard normal distribution, which we know to be irreducible. The irreducibility then carries over to the Zig-Zag process with the stated switching rates. A detailed proof can be found in the Supplementary Material.

REMARK 2.11. Based on numerous experiments, we conjecture that the canonical multidimensional Zig-Zag process is ergodic in general under only mild conditions. A detailed investigation of ergodicity will be the subject of a forthcoming paper (Bierkens, Roberts and Zitt (2017)).

3. Implementation.

As mentioned earlier, the main computational challenge is an efficient simulation of the random times T_i^k introduced in Section 2.1. We will focus on simulation by means of Poisson thinning.

PROPOSITION 3.1 (Poisson thinning, Lewis and Shedler (1979)). *Let $m : \mathbb{R}_+ \to \mathbb{R}_+$ and $M : \mathbb{R}_+ \to \mathbb{R}_+$ be continuous such that $m(t) \le M(t)$ for $t \ge 0$. Let $\tau^1, \tau^2, \dots$ be the increasing finite or infinite sequence of points of a Poisson process with rate function $(M(t))_{t\ge 0}$. For all i, delete the point τ^i with probability $1 - m(\tau^i)/M(\tau^i)$. Then the remaining points, $\tilde{\tau}^1, \tilde{\tau}^2, \dots$ say, form a nonhomogeneous Poisson process with rate function $(m(t))_{t\ge 0}$.*

Now for a given initial point $(\xi,\theta) \in E$, let $m_i(t) := \lambda_i(\xi + \theta t, \theta)$, for $i = 1,\dots,d$, and suppose we have available continuous functions $M_i(t)$ such that $m_i(t) \le M_i(t)$ for $i = 1,\dots,d$ and $t \ge 0$. We call these $(M_i)_{i=1}^d$ *computational bounds* for $(m_i)_{i=1}^d$. We can use Proposition 3.1 to obtain the first switching times $(\tilde{\tau}_i^1)_{i=1}^d$ from a (theoretically infinite) collection of *proposed switching times* $(\tau_i^1, \tau_i^2, \dots)_{i=1}^d$ given the initial point (ξ,θ), and use the obtained skeleton point at time $\tilde{\tau}^1 := \min_{i\in\{1,\dots,d\}} \tilde{\tau}_i^1$ as a new initial point (which is allowed by the strong Markov property) with the component $i_0 = \operatorname{argmin}_{i\in\{1,\dots,d\}} \tilde{\tau}_i^1$ of θ switched.

The strong Markov property of the Zig-Zag process simplifies the computational procedure further: We can draw for each component $i = 1,\dots,d$ the first proposed switching time $\tau_i := \tau_i^1$, determine $i_0 := \operatorname{argmin}_{i\in\{1,\dots,d\}} \tau_i$ and decide whether the appropriate component of θ is switched at this time with probability $m_{i_0}(\tau)/M_{i_0}(\tau)$, where $\tau := \tau_{i_0}$. Then since τ is a stopping time for the Markov process, we can use the obtained point of the Zig-Zag process at time τ as new

```
Algorithm 1: Zig-Zag sampling (ZZ)
  Input: initial condition $(\xi, \theta) \in E$.
  Output: a sequence of skeleton points $(T^k, \Xi^k, \Theta^k)_{k=0}^{\infty}$.
  1. $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$.
  2. for $k = 1, 2, \dots$
     (a) Define $m_i(t) := \lambda_i(\Xi^{k-1} + \Theta^{k-1}t, \Theta^{k-1})$ for $t \ge 0$ and $i = 1, \dots, d$.
     (b) For $i = 1, \dots, d$, let $(M_i)$ denote computational bounds for $(m_i)$.
     (c) Draw $\tau_1, \dots, \tau_d$ such that $\mathbb{P}(\tau_i \ge t) = \exp(-\int_0^t M_i(s)\,ds)$.
     (d) $i_0 := \operatorname{argmin}_{i=1,\dots,d}\{\tau_i\}$ and $\tau := \tau_{i_0}$.
     (e) $(T^k, \Xi^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1}\tau)$
     (f) With probability $m_{i_0}(\tau)/M_{i_0}(\tau)$,
         • $\Theta^k := F_{i_0}[\Theta^{k-1}]$,
         otherwise
         • $\Theta^k := \Theta^{k-1}$.
```

starting point, regardless of whether we switch a component of θ at the obtained skeleton point. A full computational procedure for simulating the Zig-Zag process is given by Algorithm 1.

3.1. *Computational bounds.* We now come to the important issue of obtaining computational bounds for the Zig-Zag Process, that is, useful upper bounds for the switching rates (m_i). If we can compute the inverse function $G_i(y) := \inf\{t \ge 0 : H_i(t) \ge y\}$ of $H_i : t \mapsto \int_0^t M_i(s)\,ds$, we can simulate $\tau_1, \dots, \tau_d$ using the CDF inversion technique, which involves drawing i.i.d. uniform random variables $U_1, \dots, U_d$ and setting $\tau_i := G_i(-\log U_i)$, $i = 1, \dots, d$.

Let us ignore the subscript i for a moment. Examples of computational bounds are piecewise affine bounds of the form $M : t \mapsto (a + bt)^+$, with $a, b \in \mathbb{R}$, and the constant bounds $M : t \mapsto c$ for $c \ge 0$. It is also possible to simulate using the combined rate $M : t \mapsto \min(c, (a + bt)^+)$. In these cases, $H(t) = \int_0^t M(s)\,ds$ is piecewise linear or quadratic and nondecreasing, so we can obtain an explicit expression for the inverse function, G.

The computational bounds are directly related to the algorithmic efficiency of Zig-Zag sampling. From Algorithm 1, it is clear that for every simulated time τ a single component of λ needs to be evaluated, which corresponds by (4) to the evaluation of a single component of the gradient of the negative log density Ψ. The magnitude of the computational bounds, (M_i), will determine how far the Zig-Zag process will have moved in the state space before a new evaluation of a component of λ is required, and we will pay close attention to the scaling of M_i with respect to the number of available observations in a Bayesian inference setting.

3.2. *Example*: *Globally bounded log density gradient.* If there are constants $c_i > 0$ such that $\sup_{\xi \in \mathbb{R}^d} |\partial_i \Psi(\xi)| \le c_i$, $i = 1, \dots, d$, then we can use the global upper bounds $M_i(t) = c_i$ for $t \ge 0$. Indeed, for $(\xi, \theta) \in E$,

$$\lambda_i(\xi, \theta) = \big(\theta_i \partial_i \Psi(\xi)\big)^+ \le \big|\partial_i \Psi(\xi)\big| \le c_i.$$

Algorithm 1 may be used with $M_i \equiv c_i$ for $i = 1, \dots, d$ at every iteration.

This situation arises with heavy-tailed distributions, for example, if π is Cauchy, then $\Psi(\xi) = \log(1 + \xi^2)$, and consequently $\lambda(\xi, \theta) = (\frac{2\theta\xi}{1+\xi^2})^+ \le 1$.

3.3. *Example*: *Negative log density with dominated Hessian.* Another important case is when there exists a positive definite matrix $Q \in \mathbb{R}^{d\times d}$ which dominates the Hessian $H_\Psi(\xi)$ in the positive definite ordering of matrices for every $\xi \in \mathbb{R}^d$. Here, $H_\Psi(\xi) = (\partial_i \partial_j \Psi(\xi))_{i,j=1}^d$ denotes the Hessian of Ψ.

Denote the Euclidean inner product in $\mathbb{R}^d$ by $\langle \cdot, \cdot \rangle$. For $p \in [1, \infty]$, the ℓ^p-norm on $\mathbb{R}^d$ and the induced matrix norms are both denoted by $\|\cdot\|_p$. For symmetric matrices $S, T \in \mathbb{R}^{d\times d}$, we write $S \preceq T$ if $\langle v, Sv\rangle \le \langle v, Tv\rangle$ for every $v \in \mathbb{R}^d$, or in words, if T dominates S in the positive definite ordering. The key assumption is that $H_\Psi(\xi) \preceq Q$ for all $\xi \in \mathbb{R}^d$, where $Q \in R^{d\times d}$ is positive definite. In particular, if $\|H_\Psi(\xi)\|_2 \le c$ for all ξ, then this holds for $Q = cI$. We let $(e_i)_{i=1}^d$ denote the canonical basis vectors in $\mathbb{R}^d$.

For an initial value $(\xi, \theta) \in E$, we move along the trajectory $t \mapsto \xi(t) := \xi + \theta t$. Let a_i denote an upper bound for $\theta_i \partial_i \Psi(\xi)$, $i = 1, \dots, d$ and let $b_i := \sqrt{d}\|Qe_i\|_2$. For general symmetric matrices S, T with $S \preceq T$, we have for any $v, w \in \mathbb{R}^d$ that

$$\langle v, Sw\rangle \le \|v\|_2 \|Sw\|_2 \le \|v\|_2 \|Tw\|_2. \tag{6}$$

Applying this inequality, we obtain for $i = 1, \dots, d$

$$\begin{aligned}\theta_i \partial_i \Psi\big(\xi(t)\big) &= \theta_i \partial_i \Psi(\xi) + \int_0^t \sum_{j=1}^d \partial_i \partial_j \Psi\big(\xi(s)\big)\theta_j \, ds \le a_i + \int_0^t \big\langle H_\Psi\big(\xi(s)\big) e_i, \theta\big\rangle \, ds \\ &\le a_i + \int_0^t \|Qe_i\|_2 \|\theta\|_2 \, ds = a_i + b_i t.\end{aligned}$$

It thus follows that

$$\lambda_i\big(\xi(t), \theta\big) = \big(\theta_i \partial_i \Psi\big(\xi(t)\big)\big)^+ \le (a_i + b_i t)^+.$$

Hence the general Zig-Zag algorithm may be applied taking

$$M_i(t) := (a_i + b_i t)^+, \qquad t \ge 0, i = 1, \dots, d,$$

with a_i and b_i as specified above. A complete procedure for Zig-Zag sampling for a log density with dominated Hessian is provided in Algorithm 2.

Algorithm 2: Zig-Zag sampling for log density with dominated Hessian

```
Input: initial condition $(\xi, \theta) \in E$.
Output: a sequence of skeleton points $(T^k, \Xi^k, \Theta^k)_{k=0}^{\infty}$.

1. $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$.
2. $a_i := \theta_i \partial_i \Psi(\xi)$, $i = 1, \dots, d$.
3. $b_i := Q e_i \sqrt{d}$, $i = 1, \dots, d$.
4. For $k = 1, 2, \dots$
   (a) Draw $\tau_i$ such that $\mathbb{P}(\tau_i \geq t) = \exp(-\int_0^t (a_i + b_i s)^+ \, ds)$, $i = 1, \dots, d$.
   (b) $i_0 := \operatorname{argmin}_{i \in \{1,\dots,d\}} \tau_i$ and $\tau := \tau_{i_0}$.
   (c) $(T^k, \Xi^k, \Theta^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1}\tau, \Theta^{k-1})$.
   (d) $a_i := a_i + b_i \tau$, $i = 1, \dots, d$.
   (e) with probability $\frac{(\Theta_{i_0}^{k-1} \partial_{i_0} \Psi(\Xi^k))^+}{(a_{i_0})^+}$,
       • $\Theta^k := F_{i_0}[\Theta^{k-1}]$,
       otherwise
       • $\Theta^k := \Theta^{k-1}$.
   (f) $a_{i_0} := \Theta_{i_0}^{k-1} \partial_{i_0} \Psi(\Xi^k)$ (re-using the earlier computation).
```

REMARK 3.2. It is also possibly to apply inequality (6) in such a way as to obtain the estimate

$$\langle H_\Psi(\xi(s)) e_i, \theta \rangle = \langle e_i, H_\Psi(\xi(s))\theta \rangle \leq \|e_i\|_2 \|Q\theta\|_2 = \|Q\theta\|_2.$$

This requires us to compute $Q\theta$ whenever θ changes [a computation of $O(d)$].

4. Big data Bayesian inference by means of error-free sub-sampling.

Throughout this section, we assume the derivatives of Ψ admit the representation

$$\partial_i \Psi(\xi) = \frac{1}{n} \sum_{j=1}^{n} E_i^j(\xi), \qquad i = 1, \dots, d, \xi \in \mathbb{R}^d, \tag{7}$$

with $(E^j)_{j=1}^n$ continuous functions mapping $\mathbb{R}^d$ into $\mathbb{R}^d$. The motivation for considering such a class of density functions is the problem of sampling from a posterior distribution for big data. The key feature of such posteriors is that they can be written as the product of a large number of terms. For example, consider the simplest example of this, where we have n independent data points $(x^j)_{j=1}^n$ and for which the likelihood function is $L(\xi) = \prod_{j=1}^n f(x^j | \xi)$, for some probability density or probability mass function f. In this case, we can write the negative log density Ψ associated with the posterior distribution as an average

$$\Psi(\xi) = \frac{1}{n}\sum_{j=1}^{n} \Psi^j(\xi), \qquad \xi \in \mathbb{R}^d, \tag{8}$$

where $\Psi^j(\xi) = -\log \pi_0(\xi) - n \log f(x^j|\xi)$, and we could choose $E_i^j(\xi) = \partial_i \Psi^j(\xi)$. It is crucial that every E_i^j is a factor $O(n)$ cheaper to evaluate than the full derivative $\partial_i \Psi(\xi)$.

We will describe two successive improvements over the basic Zig-Zag sampling (ZZ) algorithm specifically tailored to the situation in which (7) is satisfied. The first improvement consists of a sub-sampling approach where we need calculate only one of the $E_i^j s$ at each simulated time, rather than sum of all n of them. This sub-sampling approach (referred to as Zig-Zag with Sub-Sampling, ZZ-SS) comes at the cost of an increased computational bound. Our second improvement is to use control variates to reduce this bound, resulting in the Zig-Zag with Control Variates (ZZ-CV) algorithm.

4.1. *Main idea.* Let $(\xi(t))_{t\geq 0}$ denote a linear trajectory originating in $(\xi, \theta) \in E$, so $\xi(t) = \xi + \theta t$. Define a collection of switching rates along the trajectory $(\xi(t))$ by

$$m_i^j(t) := \left(\theta_i E_i^j(\xi(t))\right)^+, \qquad i = 1, \dots, d, j = 1, \dots, n, t \geq 0.$$

We will make use of computational bounds (M_i) as before, which this time bound (m_i^j) uniformly. Let $M_i : \mathbb{R}_+ \to \mathbb{R}_+$ be continuous and satisfy

$$m_i^j(t) \leq M_i(t) \qquad \text{for all } i = 1, \dots, d, j = 1, \dots, n, \text{ and } t \geq 0. \tag{9}$$

We will generate random times according to the computational upper bounds (M_i) as before. However, we now use a two-step approach to deciding whether to switch or not at the generated times. As before, for $i = 1, \dots, d$ let $(\tau_i)_{i=1}^d$ be simulated random times for which $\mathbb{P}(\tau \geq t) = \exp(-\int_0^t M_i(s)\, ds)$ and let $i_0 := \operatorname{argmin}_{i \in \{1,\dots,d\}} \tau_i$, and $\tau := \tau_{i_0}$. Then switch component i_0 of θ with probability $m_{i_0}^J(\tau)/M_{i_0}(\tau)$, where $J \in \{1, \dots, n\}$ is drawn uniformly at random, independent of τ. This "sub-sampling" procedure is detailed in Algorithm 3. Depending on the choice of E_i^j, we will refer to this algorithm as Zig-Zag with Sub-Sampling (ZZ-SS, Section 4.2) or ZZ-CV (Section 4.3).

THEOREM 4.1. *Algorithm 3 generates a skeleton of a Zig-Zag process with switching rates given by*

$$\lambda_i(\xi, \theta) = \frac{1}{n}\sum_{j=1}^{n} \left(\theta_i E_i^j(\xi)\right)^+, \qquad i = 1, \dots, d, (\xi, \theta) \in E, \tag{10}$$

and invariant distribution μ given by (3).

Algorithm 3: Zig-Zag with Sub-Sampling (ZZ-SS)/Zig-Zag with Control Variates (ZZ-CV)

```
Input: initial condition $(\xi, \theta) \in E$.
Output: a sequence of skeleton points $(T^k, \Xi^k, \Theta^k)_{k=0}^{\infty}$.

1. $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$.
2. for $k = 1, 2, \ldots$
   (a) Define $m_i^j(t) := (\Theta^{k-1} E_i^j(\Xi^{k-1} + \Theta^{k-1} t))^+$ for $t \geq 0$, $i = 1, \ldots, d$
         and $j = 1, \ldots, n$.
   (b) For $i = 1, \ldots, d$, let $(M_i)$ denote computational bounds for $(m_i^j)$,
         satisfying (9).
   (c) Draw $\tau_1, \ldots, \tau_d$ such that $\mathbb{P}(\tau_i \geq t) = \exp(-\int_0^t M_i(s)\, ds)$.
   (d) $i_0 := \operatorname{argmin}_{i=1,\ldots,d} \tau_i$ and $\tau := \tau_{i_0}$.
   (e) $(T^k, \Xi^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1}\tau)$.
   (f) Draw $J \sim \text{Uniform}(\{1, \ldots, n\})$.
   (g) With probability $m_{i_0}^J(\tau)/M_{i_0}(\tau)$,
         • $\Theta^k := F_{i_0}[\Theta^{k-1}]$,
       otherwise
         • $\Theta^k := \Theta^{k-1}$.
```

PROOF. Conditional on τ, the probability that component i_0 of θ is switched at time τ is seen to be

$$\mathbb{E}_J[m_{i_0}^J(\tau)/M_{i_0}(\tau)] = \frac{\frac{1}{n}\sum_{j=1}^n m_{i_0}^j(\tau)}{M_{i_0}(T)} = \frac{m_{i_0}(\tau)}{M_{i_0}(\tau)},$$

where

$$m_i(t) := \frac{1}{n}\sum_{j=1}^n m_i^j(t) = \frac{1}{n}\sum_{j=1}^n (\theta_i E_i^j(\xi(t)))^+, \qquad i = 1, \ldots, d, t \geq 0.$$

By Proposition 3.1, we thus have an effective switching rate λ_i for switching the ith component of θ given by (10). Finally, we verify that the switching rates (λ_i) given by (10) satisfy (2). Indeed,

$$\begin{aligned}\lambda_i(\xi, \theta) - \lambda_i(\xi, F_i[\theta]) &= \frac{1}{n}\sum_{j=1}^n \{(\theta_i E_i^j(\xi))^+ - (\theta_i E_i^j(\xi))^-\} \\ &= \frac{1}{n}\sum_{j=1}^n \theta_i E_i^j(\xi) = \theta_i \partial_i \Psi(\xi).\end{aligned}$$

By Theorem 2.2, the Zig-Zag process has the stated invariant distribution. □

The important advantage of using Zig-Zag in combination with sub-sampling is that at every iteration of the algorithm we only have to evaluate a single component of E_i^j, which reduces algorithmic complexity by a factor $O(n)$. However, this may come at a cost. First, the computational bounds (M_i) may have to be increased which in turn will increase the algorithmic complexity of simulating the Zig-Zag sampler. Also, the dynamics of the Zig-Zag process will change, because the actual switching rates of the process are increased. This increases the diffusivity of the continuous time Markov process, and affects the mixing properties in a negative way.

4.2. *Zig-Zag with Sub-Sampling* (*ZZ-SS*) *for globally bounded log density gradient.* A straightforward application of sub-sampling is possible if we have (8) with $\nabla \Psi^j$ globally bounded, so that there exist positive constants (c_i) such that

$$|\partial_i \Psi^j(\xi)| \le c_i, \qquad i = 1, \dots, d, j = 1, \dots, n, \xi \in \mathbb{R}^d. \tag{11}$$

In this case, we may take

$$E_i^j := \partial_i \Psi^j \quad \text{and} \quad M_i(t) := c_i, \qquad i = 1, \dots, d, j = 1, \dots, n, t \ge 0,$$

so that (9) is satisfied. The corresponding version of Algorithm 3 will be called Zig-Zag with Sub-Sampling (ZZ-SS).

4.3. *Zig-Zag with Control Variates* (*ZZ-CV*). Suppose again that Ψ admits the representation (8), and further suppose that the derivatives $(\partial_i \Psi^j)$ are globally and uniformly Lipschitz, that is, there exist constants $(C_i)_{i=1}^n$ such that for some $p \in [1, \infty]$ and all $i = 1, \dots, d$, $j = 1, \dots, n$ and $\xi_1, \xi_2 \in \mathbb{R}^d$,

$$|\partial_i \Psi^j(\xi_1) - \partial_i \Psi^j(\xi_2)| \le C_i \|\xi_1 - \xi_2\|_p. \tag{12}$$

To use these Lipschitz bounds, we need to choose a reference point $\xi^\star$ in ξ-space, so that we can bound the derivative of the log density based on how close we are to this reference point. Now if we choose any fixed reference point, $\xi^\star \in \mathbb{R}^d$, we can use a control variate idea to write

$$\partial_i \Psi(\xi) = \partial_i \Psi(\xi^\star) + \frac{1}{n} \sum_{i=1}^n [\partial_i \Psi^j(\xi) - \partial_i \Psi^j(\xi^\star)], \qquad \xi \in \mathbb{R}^d, i = 1, \dots, d.$$

This suggests using

$$E_i^j(\xi) := \partial_i \Psi(\xi^\star) + \partial_i \Psi^j(\xi) - \partial_i \Psi^j(\xi^\star),$$
$$\xi \in \mathbb{R}^d, i = 1, \dots, d, j = 1, \dots, n.$$

The reason for defining $E_i^j(\xi)$ in this manner is to try and reduce its variability as we vary j. By the Lipschitz condition, we have $E_i^j(\xi) \le |\partial_i \Psi(\xi^\star)| + C_i \|\xi -$

$\xi^\star\|_p$, and thus the variability of the $E_i^j(\xi)$s will be small if (1) the reference point $\xi^\star$ is close to the mode of the posterior and (2) ξ is close to $\xi^\star$. Under standard asymptotics, we expect a draw from the posterior for ξ to be $O_p(n^{-1/2})$ from the posterior mode. Thus if we have a procedure for finding a reference point $\xi^\star$ which is within $O(n^{-1/2})$ of the posterior mode then this would ensure $\|\xi - \xi^\star\|_2$ is $O_p(n^{-1/2})$ if ξ is drawn from the posterior. For such a choice of $\xi^\star$, we would have $\partial_i \Psi(\xi^\star)$ of $O_p(n^{1/2})$.

Using the Lipschitz condition, we can now obtain computational bounds of (m_i) for a trajectory $\xi(t) := \xi + \theta t$ originating in (ξ, θ). Define

$$M_i(t) := a_i + b_i t, \qquad t \geq 0, i = 1, \dots, d,$$

where $a_i := (\theta_i \partial_i \Psi(\xi^\star))^+ + C_i \|\xi - \xi^\star\|_p$ and $b_i := C_i d^{1/p}$. Then (9) is satisfied. Indeed, using Lipschitz continuity of $y \mapsto (y)^+$,

$$\begin{aligned} m_i^j(t) &= \left(\theta_i E_i^j(\xi + \theta t)\right)^+ = \left(\theta_i \partial_i \Psi(\xi^\star) + \theta_i \partial_i \Psi^j(\xi + \theta t) - \theta_i \partial_i \Psi^j(\xi^\star)\right)^+ \\ &\leq \left(\theta_i \partial_i \Psi(\xi^\star)\right)^+ + \left|\partial_i \Psi^j(\xi) - \partial_i \Psi^j(\xi^\star)\right| + \left|\partial_i \Psi^j(\xi + \theta t) - \partial_i \Psi^j(\xi)\right| \\ &\leq \left(\theta_i \partial_i \Psi(\xi^\star)\right)^+ + C_i\left(\left\|\xi - \xi^\star\right\|_p + t\|\theta\|_p\right) = M_i(t). \end{aligned}$$

Implementing this scheme requires some preprocessing of the data. First, we need a way of choosing a suitable reference point $\xi^\star$ to find a value close to the mode using an approximate or exact numerical optimization routine. The complexity of this operation will be $O(n)$. Once we have found such a reference point, we have an one-off $O(n)$ cost of calculating $\partial_i \Psi(\xi^\star)$ for each $i = 1, \dots, d$. However, once we have paid this upfront computational cost, the resulting Zig-Zag sampler can be super-efficient. This is discussed in more detail in Section 5, and demonstrated empirically in Section 6. The version of Algorithm 3 resulting from this choice of E_i^j and M_i will be called Zig-Zag with Control Variates (ZZ-CV).

REMARK 4.2. When choosing $p \geq 1$, there will be a trade-off between the magnitude of C_i and of $\|\xi - \xi^\star\|_p$, which may influence the scaling of Zig-Zag sampling with dimension. We will see in Section 6.3 that for i.i.d. Gaussian components, the choice $p = \infty$ is optimal. When the situation is less clear, choosing the Euclidean norm ($p = 2$) is a reasonable choice.

5. Scaling analysis.

In this section, we provide an informal scaling argument for canonical Zig-Zag, and Zig-Zag with control variates and sub-sampling. For the moment fix $n \in \mathbb{N}$ and consider a posterior with negative log density,

$$\Psi(\xi) = -\sum_{j=1}^n \log f(x^j \mid \xi),$$

where x^j are i.i.d. drawn from $f(x^j \mid \xi_0)$. Let $\widehat{\xi}$ denote the maximum likelihood estimator (MLE) for ξ based on data $x^1, \dots, x^n$. Introduce the coordinate transformation

$$\phi(\xi) = \sqrt{n}(\xi - \widehat{\xi}), \qquad \xi(\phi) = \frac{1}{\sqrt{n}}\phi + \widehat{\xi}.$$

As $n \to \infty$, the posterior distribution in terms of ϕ will converge to a multivariate Gaussian distribution with mean 0 and covariance matrix given by the inverse of the expected information $i(\theta_0)$; see, for example, Johnson (1970).

5.1. *Scaling of Zig-Zag sampling* (*ZZ*). First, let us obtain a Taylor expansion of the switching rate for ξ close to $\widehat{\xi}$. We have

$$\begin{aligned}\partial_{\xi_i}\Psi(\xi) &= -\partial_{\xi_i}\sum_{j=1}^n \log f(x^j \mid \xi) \\ &= \underbrace{-\partial_{\xi_i}\sum_{j=1}^n \log f(x^j \mid \widehat{\xi})}_{=0} \\ &\quad - \sum_{j=1}^n\sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \widehat{\xi})(\xi_k - \widehat{\xi}_k) + O(\|\xi - \widehat{\xi}\|^2).\end{aligned}$$

The first term vanishes by the definition of the MLE. Expressed in terms of ϕ, the switching rates are

$$(\theta_i \partial_{\xi_i}\Psi(\xi(\phi)))^+ = \underbrace{\frac{1}{\sqrt{n}}\left(-\sum_{j=1}^n\sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \widehat{\xi})\phi_k\right)^+}_{O(\sqrt{n})} + O\left(\frac{\|\phi\|^2}{n}\right).$$

With respect to the coordinate ϕ, the canonical Zig-Zag process has constant speed $\sqrt{n}$ in each coordinate, and by the above computation, a switching rate of $O(\sqrt{n})$. After a rescaling of the time parameter by a factor $\sqrt{n}$, the process in the ϕ-coordinate becomes a Zig-Zag process with unit speed in every direction and switching rates

$$\left(-\frac{1}{n}\sum_{j=1}^n\sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \xi)\phi_k\right)^+ + O(n^{-1/2}).$$

If we let $n \to \infty$, the switching rates converge almost surely to those of a Zig-Zag process with switching rates

$$\widetilde{\lambda}_i(\phi, \theta) = (\theta_i(i(\theta_0)\phi)_i)^+,$$

where $i(\theta_0)$ denotes the expected information. These switching rates correspond to the limiting Gaussian distribution with covariance matrix $i(\theta_0)^{-1}$.

In this limiting Zig-Zag process, all dependence on n has vanished. Starting from equilibrium, we require a time interval of $O(1)$ (in the rescaled time) to obtain an essentially independent sample. In the original time scale, this corresponds to a time interval of $O(n^{-1/2})$. As long as the computational bound in the Zig-Zag algorithm is $O(n^{1/2})$, this can be achieved using $O(1)$ proposed switches. The computational cost for every proposed switch is $O(n)$, because the full data $(x^i)_{i=1}^n$ needs to be processed in the computation of the true switching rate at the proposed switching time.

We conclude that the computational complexity of the Zig-Zag (ZZ) algorithm per independent sample is $O(n)$, provided that the computational bound is $O(n^{1/2})$. This is the best we can expect for any standard Monte Carlo algorithm [where we will have a $O(1)$ number of iterations, but each iteration is $O(n)$ in computational cost].

To compare, if the computational bound is $O(n^\alpha)$ for some $\alpha > 1/2$, then we require $O(n^{\alpha-1/2})$ proposed switches before we have simulated a total time interval of length $O(n^{-1/2})$, so that, with a complexity of $O(n)$ per proposed switching time, the Zig-Zag algorithm has total computational complexity $O(n^{\alpha+1/2})$. So, for example, with global bounds we have that the computational bound is $O(n)$ [as each term in the log density is $O(1)$], and hence ZZ will have total computational complexity of $O(n^{3/2})$.

EXAMPLE 5.1 (Dominated Hessian). Consider Algorithm 2 in the one-dimensional case, with the second derivative of Ψ bounded from above by $Q > 0$. We have $Q = O(n)$ as Ψ'' is the sum of n terms of $O(1)$. The value of b is kept fixed at the value $b = Q = O(n)$. Next, a is given initially as

$$a = \theta\Psi'(\xi) \le \theta \underbrace{\Psi'(\widehat{\xi})}_{=0} + \underbrace{Q}_{O(n)} \underbrace{(\xi - \widehat{\xi})}_{O(n^{-1/2})} = O(n^{1/2}),$$

and increased by $b\tau$ until a switch happens and a is reset to $\theta\Psi'(\xi)$. Because of the initial value for a, switches will occur at rate $O(n^{1/2})$ so that τ will be $O(n^{-1/2})$, and the value of a will remain $O(n^{1/2})$. Hence the magnitude of the computational bound $M(t) = (a + bt)^+$ is $O(n^{1/2})$.

5.2. *Scaling of Zig-Zag with Control Variates* (*ZZ-CV*). Now we will study the limiting behaviour as $n \to \infty$ of ZZ-CV introduced in Section 4.3. In determining the computational bounds, we take $p = 2$ for simplicity, for example, in (12). Also for simplicity assume that $\xi \mapsto \partial_{\xi_i} \log f(x^j \mid \xi)$ has Lipschitz constant k_i (independent of $j = 1, \dots, n$) and write $C_i = nk_i$, so that (12) is satisfied. In practice, there may be a logarithmic increase with n in the Lipschitz constants k_i as we have to take a global bound in n. For the present discussion, we ignore such logarithmic factors. We assume reference points $\xi^\star$ for growing n are determined in such a way that $\|\xi^\star - \widehat{\xi}\|_2$ is $O(n^{-1/2})$. For definiteness, suppose there exists a d-dimensional random variable Z such that $n^{1/2}(\xi^\star - \widehat{\xi}) \to Z$ in distribution, with the randomness in Z independent of $(x^j)_{j=1}^\infty$.

We can look at ZZ-CV with respect to the scaled coordinate ϕ as $n \to \infty$. Denote the reference point for the rescaled parameter as $\phi^\star := \sqrt{n}(\xi^\star - \widehat{\xi})$.

The essential quantities to consider are the switching rate estimators E_i^j. We estimate

$$\begin{aligned}
|E_i^j(\xi)| &= |\partial_{\xi_i}\Psi(\xi^\star) + \partial_{\xi_i}\Psi^j(\xi) - \partial_{\xi_i}\Psi^j(\xi^\star)| \\
&= |\partial_{\xi_i}\Psi(\xi^\star) - \partial_{\xi_i}\Psi(\widehat{\xi}) + \partial_{\xi_i}\Psi^j(\xi) - \partial_{\xi_i}\Psi^j(\xi^\star)| \\
&\le \underbrace{C_i}_{O(n)} \underbrace{\|\xi^\star - \widehat{\xi}\|}_{O(n^{-1/2})} + \underbrace{C_i}_{O(n)} \underbrace{\|\xi - \xi^\star\|}_{O(n^{-1/2})}.
\end{aligned}$$

We find that $|E_i^j(\xi)| = O(n^{1/2})$ under the stationary distribution.

By slowing down the Zig-Zag process in ϕ space by $\sqrt{n}$, the continuous time process generated by ZZ-CV will approach a limiting Zig-Zag process with a certain switching rate of $O(1)$. In general, this switching rate will depend on the way that $\xi^\star$ is obtained. To simplify the exposition, in the following computation we assume $\xi^\star = \widehat{\xi}$. Rescaling by $n^{-1/2}$, and developing a Taylor approximation around $\widehat{\xi}$,

$$\begin{aligned}
n^{-1/2}E_i^j(\xi) &= n^{-1/2}\big(\partial_{\xi_i}\Psi^j(\xi) - \partial_{\xi_i}\Psi^j(\widehat{\xi})\big) \\
&= n^{-1/2}\big(-n\partial_{\xi_i}\log f(x^j \mid \xi) + n\partial_{\xi_i}\log f(x^j \mid \widehat{\xi})\big) \\
&= -n^{1/2}\left(\sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \widehat{\xi})(\xi_k - \widehat{\xi}_k)\right) + O\big(n^{1/2}\|\xi - \widehat{\xi}\|^2\big) \\
&= -\sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \widehat{\xi})\phi_k + O\big(n^{-1/2}\big).
\end{aligned}$$

By Theorem 4.1, the rescaled effective switching rate for ZZ-CV is given by

$$\begin{aligned}
\widetilde{\lambda}_i(\phi,\theta) := n^{-1/2}\lambda_i\big(\xi(\phi),\theta\big) &= \frac{1}{n^{3/2}}\sum_{j=1}^n \big(\theta_i E_i^j(\xi(\phi))\big)^+ \\
&= \frac{1}{n}\sum_{j=1}^n \left(-\theta_i \sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(x^j \mid \widehat{\xi})\phi_k\right)^+ + O\big(n^{-1/2}\big) \\
&\to \mathbb{E}\left(-\theta_i \sum_{k=1}^d \partial_{\xi_i}\partial_{\xi_k}\log f(X \mid \xi_0)\phi_k\right)^+,
\end{aligned}$$

where $\mathbb{E}$ denotes expectation with respect to X, with density $f(\cdot \mid \xi_0)$ and the convergence is a consequence of the law of large numbers. If $\xi^{\star}$ is not exactly equal to $\widehat{\xi}$, the limiting form of $\widetilde{\lambda}_i(\phi, \theta)$ will be different, but the important point is that it will be $O(1)$, which follows from the bound on $|E_i^j|$ above.

Just as with ZZ, the rescaled Zig-Zag process underlying ZZ-CV converges to a limiting Zig-Zag process with switching rate $\widetilde{\lambda}_i(\phi, \theta)$. Since the computational bounds of ZZ-CV are $O(n^{1/2})$, a completely analogous reasoning to the one for ZZ algorithm above (Section 5.1) leads to the conclusion that $O(1)$ proposed switches are required to obtain an independent sample. However, in contrast with the ZZ-algorithm, the ZZ-CV algorithm is designed in such a way that the computational cost per proposed switch is $O(1)$.

We conclude that the computational complexity of the ZZ-CV algorithm is $O(1)$ per independent sample. This provides a factor n increase in efficiency over standard MCMC algorithms, resulting in an *asymptotically unbiased* algorithm for which *the computational cost of obtaining an independent sample does not depend on the size of the data.*

5.3. *Remarks.* The arguments above assume we are at stationarity—and how quickly the two algorithms converge is not immediately clear. Note, however, that for sub-sampling Zig-Zag it is possible to choose the reference point $\xi^{\star}$ as the starting point, thus avoiding much of the issues about convergence.

In some sense, the good computational scaling of ZZ-CV is leveraging the asymptotic normality of the posterior, but in such a way that ZZ-CV always samples from the true posterior. Thus when the posterior is close to Gaussian it will be quick; when it is far from Gaussian it may well be slower but will still be "correct". This is fundamentally different from other algorithms (e.g., Bardenet, Doucet and Holmes (2017), Neiswanger, Wang and Xing (2013), Scott et al. (2016)) that utilise the asymptotic normality in terms of justifying their approximation to the posterior. Such algorithms are accurate if the posterior is close to Gaussian, but may be inaccurate otherwise, and it is often impossible to quantify the size of the approximation in practice.

6. Examples and experiments.

6.1. *Sampling and integration along Zig-Zag trajectories.* There are essentially two different ways of using the Zig-Zag skeleton points which we obtain by using, for example, Algorithms 1, 2 or 3.

The first possible approach is to collect a number of samples along the trajectories. Suppose we have simulated the Zig-Zag process up to time $\tau > 0$, and we wish to collect m samples. This can be achieved by setting $t_i = i\tau/m$, and setting $\Xi_i := \Xi(t_i)$ for $i = 1, \dots, m$, with the continuous time trajectory $(\Xi(t))$ defined as in Section 2.1. In order to approximate $\pi(f)$ numerically for some function $f : \mathbb{R}^d \to \mathbb{R}$ of interest, we can use the usual ergodic average

$$\widehat{\pi(f)} := \frac{1}{m} \sum_{i=1}^{m} f(\Xi_i).$$

We can also estimate posterior quantiles by using the quantiles of the sample $\Xi_1, \dots, \Xi_m$, as with standard MCMC output. An issue with this approach is that we have to decide on the number, m, of samples we wish to use. Whilst the more samples we use the greater the accuracy of our approximation to $\pi(f)$, this comes at an increased computational and storage cost. The trade-off in choosing an appropriate value for m is equivalent to the choice of how much to thin output from a standard MCMC algorithm.

It is important that one does not make the mistake of using the switching points of the Zig-Zag process as samples, as these points are not distributed according to π. In particular, the switching points are biased towards the tails of the target distribution.

An alternative approach is intrinsically related to the continuous time and piecewise linear nature of the Zig-Zag trajectories. This approach consists of continuous time integration of the Zig-Zag process. By the continuous time ergodic theorem, for f as above, $\pi(f)$ can be estimated as

$$\widehat{\pi(f)} = \frac{1}{\tau} \int_0^{\tau} f(\Xi(s))\, ds.$$

Since the output of the Zig-Zag algorithms consists of a finite number of skeleton points $(T^i, \Xi^i, \Theta^i)_{i=0}^k$, we can express this as

$$\widehat{\pi(f)} = \frac{1}{T^k} \sum_{i=1}^{k} \int_{T^{i-1}}^{T^i} f(\Xi^{i-1} + \Theta^{i-1}(s - T^{i-1}))\, ds.$$

Due to the piecewise linearity of $\Xi(t)$, in many cases these integrals can be computed exactly, for example, for the moments, $f(x) = x^p$, $p \in \mathbb{R}$. In cases where the integral cannot be computed exactly, numerical quadrature rules can be applied. An advantage of this method is that we do not have to make an arbitrary decision on the number of samples to extract from the trajectory.

6.2. *Beating one ESS per epoch.*

We use the term *epoch* as a unit of computational cost, corresponding to the number of iterations required to evaluate the complete gradient of $\log \pi$. This means that for the basic Zig-Zag algorithm (without sub-sampling), an epoch consists of exactly one iteration, and for the sub-sampled variants of the Zig-Zag algorithm, an epoch consists of n iterations. The CPU running times per epoch of the various algorithms we consider are equal up to a constant factor. To assess the scaling of various algorithms, we use *ESS per epoch*. The notion of ESS is discussed in the Supplementary Material [Bierkens, Fearnhead and Roberts (2019), Section 2]. Consider any classical MCMC algorithm based upon the Metropolis–Hastings acceptance rule. Since every iteration requires an evaluation of the full density function to compute the acceptance probability, we have that the ESS per epoch for such an algorithm is bounded from above by one. Similar observations apply to all other known MCMC algorithms capable of sampling asymptotically from the exact target distribution.

There do exist several conceptual innovations based on the idea of subsampling, which have some theoretical potential to overcome the fundamental limitation of one ESS per epoch sketched above.

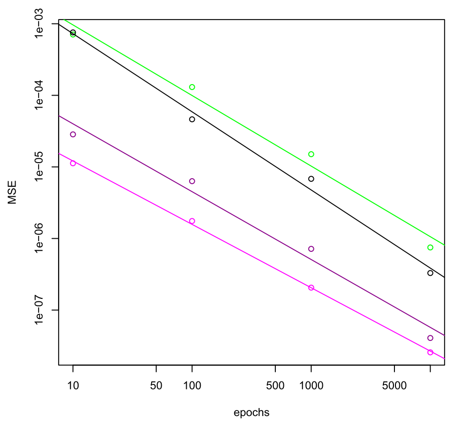
The Pseudo-Marginal Method (PMM, Andrieu and Roberts (2009)) is based upon using a positive unbiased estimator for a possibly unnormalized density. Obtaining an unbiased estimator of a product is much more difficult than obtaining one for a sum. Furthermore, it has been shown to be impossible to construct an estimator that is guaranteed to be positive without other information about the product, such as a bound on the terms in the product (Jacob and Thiery (2015)). Therefore, the PMM does not apply in a straightforward way to vanilla MCMC in Bayesian inference.

In the Supplementary Material (Bierkens, Fearnhead and Roberts (2019), Section 3) we analyse the scaling of Stochastic Gradient Langevin Dynamics (SGLD, Welling and Teh (2011)) in an analogous fashion to the analysis of ZZ and ZZ-CV in Section 5. From this analysis, we conclude that it is in general not possible to implement SGLD in such a way that the ESSpE has a larger order of magnitude than $O(1)$. We compare SGLD to Zig-Zag in experiments of Sections 6.3 and 6.5.

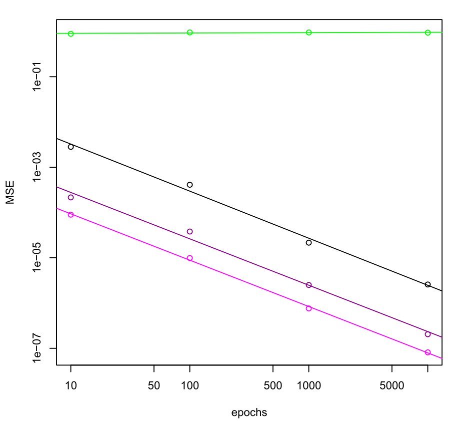
6.3. *Mean of a Gaussian distribution.* Consider the illustrative problem of estimating the mean of a Gaussian distribution. This problem has the advantage that it allows for an analytical solution which can be compared with the numerical solutions obtained by Zig-Zag sampling and other methods. Conditional on a one-dimensional parameter ξ, the data is assumed to be i.i.d. from $N(\xi, \sigma^2)$. Furthermore a $N(0, 1/\rho^2)$ prior on ξ is specified. Data are generated from the true distribution $N(\xi_0, \sigma^2)$ for some fixed ξ_0. For a detailed description of the experiment and computational bounds, see Section 4 of the Supplementary Material.

In this experiment, we compare the mean square error (MSE) for several algorithms, namely basic Zig-Zag (ZZ), Zig-Zag with Control Variates (ZZ-CV), Zig-Zag with Control Variates with a "sub-optimal" reference point (ZZ-soCV) and Stochastic Gradient Langevin Dynamics (SGLD). SGLD is implemented with fixed step size, as is usually done in practice; see, for example, Vollmer, Zygalakis and Teh (2016), with the added benefit that it makes the comparison with the Zig-Zag algorithms more straightforward. Here, in basic Zig-Zag we pretend that every iteration requires the evaluation of n observations (whereas in practice, we can precompute ξ^{MAP}).

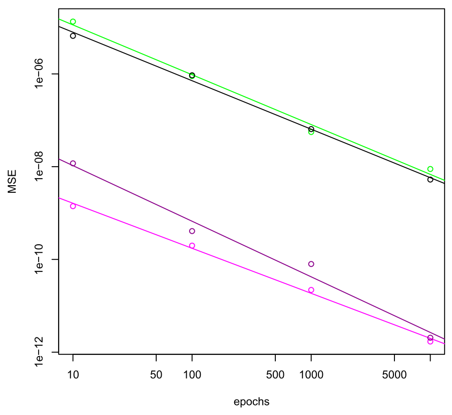
Results for this experiment are displayed in Figure 2. The MSE for the second moment using SGLD does not decrease beyond a fixed value, indicating the



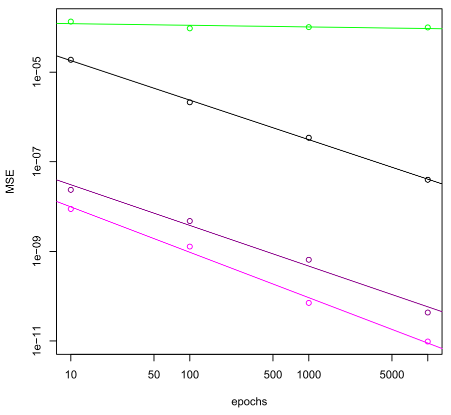
(a) First moment, 100 observations



(b) Second moment, 100 observations



(c) First moment, 10^4 observations



(d) Second moment, 10^4 observations

FIG. 2. *Mean square error (MSE) in the first and second moment as a function of the number of epochs, based on $n = 100$ or $n = 10{,}000$ observations, for a one-dimensional Gaussian posterior distribution (Section 6.3). Displayed are SGLD (green), ZZ-CV (magenta), ZZ-soCV (dark magenta), ZZ (black). The displayed dots represent averages over experiments based on randomly generated data from the true posterior distribution. Parameter values (see Bierkens, Fearnhead and Roberts (2019), Section 4) are $\xi_0 = 1$ (the true value of the mean parameter), $\rho = 1$, $\sigma = 1$ and $c_1 = 1$, $c_2 = 1/100$ (for the SGLD parameters; see the Supplementary Material, Bierkens, Fearnhead and Roberts (2019), Section 3). The value of $\xi^\star$ for ZZ-soCV is based on a sub-sample of size $m = n/10$ so that it will not be equal to the exact maximizer of the posterior. For an honest comparison, trajectories of all algorithms have initial condition equal to ξ^{MAP}.*

presence of bias in SGLD. This bias does not appear in the different versions of Zig-Zag sampling, agreeing with the theoretical result that ergodic averages over Zig-Zag trajectories are consistent. Furthermore, we see a significant relative increase in efficiency for ZZ-(so)CV over basic ZZ when the number of observations is increased, agreeing with the scaling results of Section 5. A poor choice of reference point (as in ZZ-soCV) is seen to have only a small effect on the efficiency.

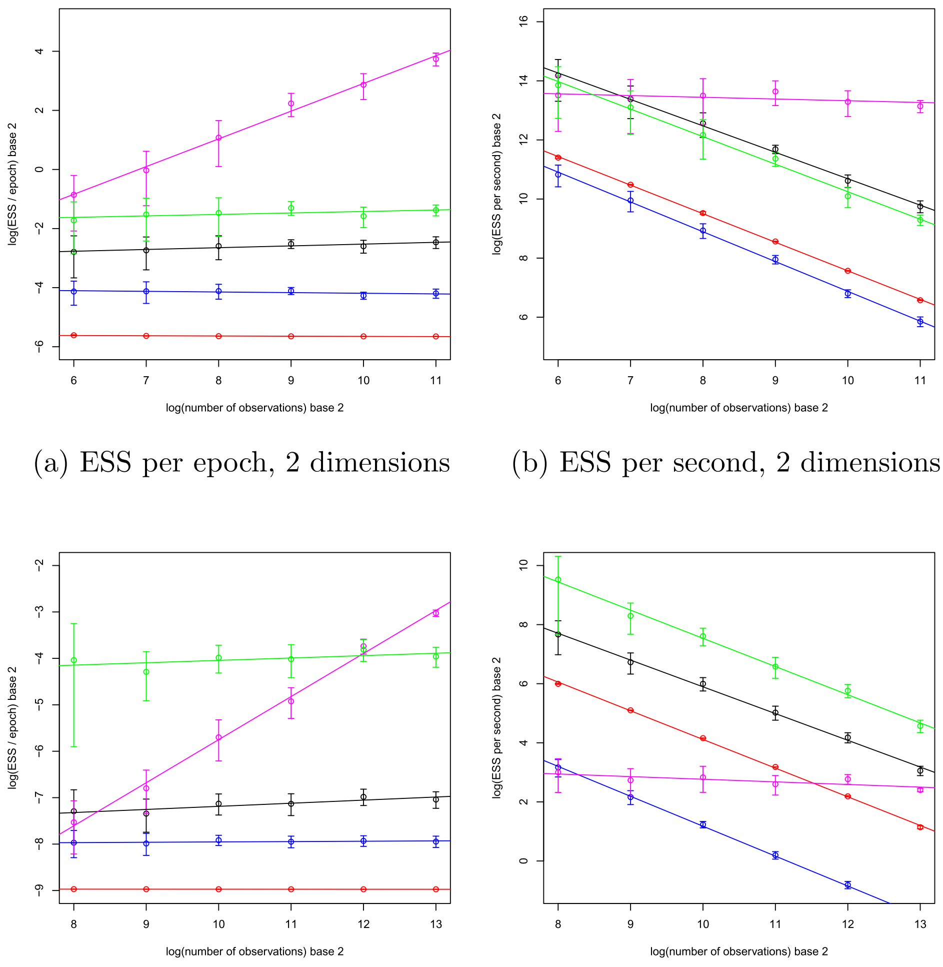
6.4. *Logistic regression.* In this numerical experiment, we compare how the ESS per epoch (ESSpE) and ESS per second grow with the number of observations n for several Zig-Zag algorithms and the MALA algorithm when applied to a logistic regression problem. Conditional on a d-dimensional parameter ξ and given d-dimensional covariates $x^j \in \mathbb{R}^d$, where $j = 1, \dots, n$, and with $x_1^j = 1$ for all j, the binary variable $y^j \in \{0, 1\}$ has distribution

$$\mathbb{P}(y^j \mid x_1^j, \dots, x_d^j, \xi_1, \dots, \xi_d) = \frac{1}{1 + \exp(-\sum_{i=1}^d \xi_i x_i)}.$$

Combined with a flat prior distribution, this induces a posterior distribution ξ given observations of (x^j, y^j) for $j = 1, \dots, n$; see the Supplementary Material for implementational details (Bierkens, Fearnhead and Roberts (2019), Section 5).

The results of this experiment are shown in Figure 3. In both of the plots of ESS per epoch (see (a) and (c)), the best linear fit for ZZ-CV has slope approximately 0.95, which is in close agreement with the scaling analysis of Section 5. The other algorithms have roughly a horizontal slope, corresponding to a linear scaling with the size of the data. We conclude that, among the algorithms tested, ZZ-CV is the only algorithm for which the ESS per CPU second is approximately constant as a function of the size of the data (see Figure 3(b) and (d)). Furthermore ZZ-CV obtains an ESSpE which is roughly linearly increasing with the number of observations n (see Figure 3(a) and (c)), whereas the other versions of the Zig-Zag algorithms, and MALA, have an ESSpE which is approximately constant with respect to n. These statements apply regardless of the dimensionality of the problem.

6.5. *A nonidentifiable logistic regression example with unbounded Hessian.* In a further experiment, we consider one-dimensional data (x^j, y^j), for $j = 1, \dots, n$, $x^j \in \mathbb{R}$, $y^j \in \{0, 1\}$, which we assume for illustrational purposes to be generated from a logistic model where $\mathbb{P}(y^j = +1 \mid x^j, \xi_1, \xi_2) = \frac{1}{1+\exp(-(\xi_1+\xi_2^2)x^j)}$. The model is nonidentifiable since two parameters ξ, η correspond to the same model as long as $\xi_1 + \xi_2^2 = \eta_1 + \eta_2^2$. This leads to a sharply rigged probability density function reminiscent of density functions concentrated along lower dimensional submanifolds which often arise in Bayesian inference problems. In this case, the Hessian of the log density is unbounded so that we cannot use the standard framework for the Zig-Zag algorithms. It is discussed in the Supplementary



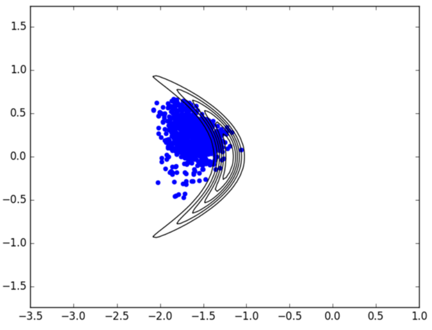
(a) ESS per epoch, 2 dimensions (b) ESS per second, 2 dimensions

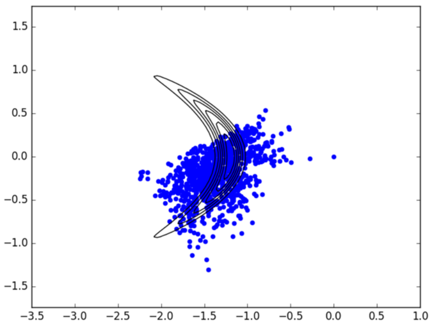
(c) ESS per epoch, 16 dimensions (d) ESS per second, 16 dimensions

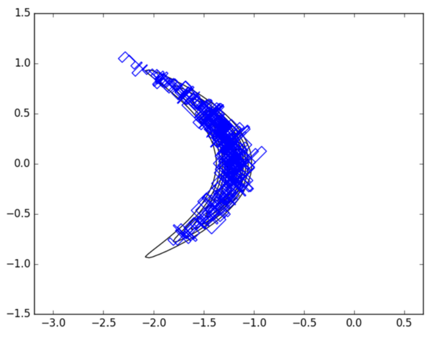
FIG. 3. *Log-log plots of the experimentally observed dependence of ESS per epoch* (*ESSpE*) *and ESS per second* (*ESSpS*) *with respect to the first coordinate* Ξ^1, *as a function of the number of observations* n *in the case of* (2-*D and* 16-*D*) *Bayesian logistic regression* (*Section* 6.4). *Data is randomly generated based on true parameter values* $\xi_0 = (1, 2)$ (2-*D*) *and* $\xi_0 = (1, \dots, 1)$ (16-*D*). *Trajectories all start in the true parameter value* ξ_0. *Plotted are mean and standard deviation over* 10 *experiments, along with the best linear fit. Displayed are MALA* (*tuned to have optimal acceptance ratio, green*), *Zig-Zag with global bound* (*red*), *Zig-Zag with Lipschitz bound* (*black*), *ZZ-SS using global bound* (*blue*) *and ZZ-CV* (*magenta*), *all run for* 10^5 *epochs. As reference point for ZZ-CV we compute the posterior mode numerically, the cost of which is negligible compared to the MCMC. The experiments are carried out in R with C++ implementations of all algorithms.*

Material (Bierkens, Fearnhead and Roberts (2019), Section 6) how to obtain computational bounds for the Zig-Zag and ZZ-CV algorithms, which may serve as an illustration on how to obtain such bounds in settings beyond those described in Sections 3.3 and 4.3.

In Figure 4, we compare trace plots for the Zig-Zag algorithms (ZZ, ZZ-CV) to trace plots for Stochastic Gradient Langevin Dynamics (SGLD) and the Consensus Algorithm (Scott et al. (2016)). SGLD and Consensus are seen to be strongly biased, whereas ZZ and ZZ-CV target the correct distribution. However, this comes at a cost: ZZ-CV loses much of its efficiency in this situation (due to the combination of lack of posterior contraction and unbounded Hessian); in particular it is not


(a) Consensus algorithm, 10 batches


(b) SGLD, 100 batches


(c) Zig-Zag

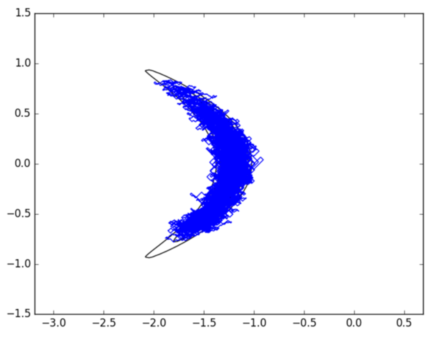

(d) ZZ-CV

FIG. 4. *Trace plots of several algorithms (blue) and density contour plots for the nonidentifiable logistic regression example of Section 6.5. In this example we have for the number of observations* $n = 1000$. *Data is randomly generated from the model with true parameter satisfying* $\xi_1 + \xi_2^2 = -1$. *The prior is a two-dimensional standard normal distribution. Due to the unbounded Hessian and because SGLD is not corrected by a Metropolis–Hastings accept/reject, the stepsize of SGLD needs to be set to a very small value (compared, e.g., to what would be required for MALA) in order to prevent explosion of the trajectory; still the algorithm exhibits a significant asymptotic bias.*

super-efficient. The use of multiple reference points may alleviate this problem, see also the discussion in Section 7.

7. Discussion.

We have introduced the multidimensional Zig-Zag process and shown that it can be used as an alternative to standard MCMC algorithms. The advantages of the Zig-Zag process are that it is a nonreversible process, and thus has the potential to mix better than standard reversible MCMC algorithms, and that we can use sub-sampling ideas when simulating the process and still be guaranteed to sample from the true target distribution of interest. We have shown that it is possible to implement sub-sampling with control-variates in a way that we can have super-efficient sampling from a posterior: the cost per effective sample size is sub-linear in the number of data points. We believe the latter aspect will be particularly useful for applications where the computational cost of calculating the likelihood for a single data point is high.

As such, the Zig-Zag process holds substantial promise. However, being a completely new method, there are still substantial challenges in implementation which will need to be overcome for Zig-Zag to reach the levels of popularity of standard discrete-time MCMC. The key challenges to implementing the Zig-Zag efficiently are:

1. to simulate from the relevant time-inhomogeneous Poisson process; and
2. in order to realise the advantages of Zig-Zag for large datasets, reasonable centering points need to be found before commencing the MCMC algorithm itself.

For the first of these challenges, we have shown how this can be achieved through bounding the rate of the Poisson process, but the overall efficiency of the simulation algorithm then depends on how tight these bounds are. In Section 3.1, we describe efficient ways to carry this out. Moreover, as pointed out by a reviewer, there is a substantial literature on simulating stochastic processes that involve simulating such time-inhomogeneous Poisson processes (Anderson (2007), Gibson and Bruck (2000)). Ideas from this literature could be leveraged both to extend the class of models for which we can simulate the Zig-Zag process, and also to make implementation of simulation algorithms more efficient.

The second challenge applies when using the ZZ-CV algorithm to obtain super-efficiency for big data as discussed in Section 4.3. Although in our experience finding appropriate centering points is rarely a serious problem, it is difficult to give a prescriptive recipe for this step.

On the face of it, these challenges may limit the practical applicability of Zig-Zag, at least in the short term. With that in mind, we have released a R/Rcpp package for logistic regression, as well as the code which reproduces the experiments of Section 6 (Bierkens (2017)).

In addition, while Zig-Zag is an *exact approximate* simulation method, there are various short-cuts to speed it up at the expense of the introduction of an approximation. For instance, there are already ideas of approximately simulating the continuous-time dynamics, through approximate bounds on the Poisson rate (Pakman et al. (2016)). These ideas can lead to efficient simulation of the Zig-Zag process for a wide class of models, albeit with the loss of exactness. Understanding the errors introduced by such an approach is an open area.

The most exciting aspect of the Zig-Zag process is the super-efficiency we observe when using sub-sampling with control variates. Already this idea has been adapted and shown to apply to other recent continuous-time MCMC algorithms (Fearnhead (2002), Pakman et al. (2016)). We have shown in Section 6.5 that Zig-Zag can be applied effectively within highly non-Gaussian examples where rival approximate methods such as SGLD and the Consensus Algorithm are seriously biased. So there is no intrinsic reason to expect Zig-Zag to rely on the target distribution being close to Gaussian, although posterior contraction and the ability to find tight Poisson process rate bounds play important roles as we saw in our examples. There is much to learn about how the efficiency of Zig-Zag depends on the statistical properties of the posterior distribution. However, unlike its approximate competitors, Zig-Zag will still remain an *exact approximate* method whatever the structure of the target distribution.

In truly "big data" settings, in principle we still need to process all the data once, although a suitable reference point can be determined using a subset of the data, we do need to evaluate the full gradient of the log density once at this reference point, and this computation is $O(n)$. This operation, however, is much easier to parallelize than MCMC is, and after this approximately independent samples can be obtained at a cost of $O(1)$ each. Thus if we wish to obtain k approximately independent samples, the computational efficiency of ZZ-CV is $O(k + n)$ while the complexity of traditional MCMC algorithms is $O(kn)$. This is confirmed by the experiment in Section 6.4.

The idea for control variates we present in this paper is just one, possibly the simplest, implementation of this idea. There are natural extensions to deal with, for example, multimodal posteriors or situations where we do not have posterior concentration for all parameters. The simplest of these involve using multiple reference points and monitoring the computational bound we get within the CV-ZZ algorithm and switching to a different algorithm when we stray so far from a reference point that this bound becomes too large. More sophisticated approaches include using the ideas from (Dubey et al. (2016)), where we introduce a reference point for each data point and update the reference points for data within the subsample at each iteration of the algorithm. This would lead to the estimate of the gradient that we center our control variate estimator around to depend on the recent history of the Zig-Zag process, and thus could be accurate even if we explore multiple modes or the tails of the target distribution.

Acknowledgments. The authors are grateful for helpful comments from referees, the Editor and the Associate Editor which have improved the paper. Furthermore, the authors acknowledge Matthew Moores (University of Warwick) for helpful advice on implementing the Zig-Zag algorithms a s an R package using Rcpp.

SUPPLEMENTARY MATERIAL

Supplement to "The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data" (DOI: 10.1214/18-AOS1715SUPP; .pdf). Mathematics of the Zig-Zag process, scaling of SGLD, details on the experiments including how to obtain computational bounds.

REFERENCES

ANDERSON, D. F. (2007). A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *J. Chem. Phys.* **127** 214107.

ANDRIEU, C. and ROBERTS, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist.* **37** 697–725. MR2502648

BARDENET, R., DOUCET, A. and HOLMES, C. (2017). On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.* **18** 1515–1557. MR3670492

BIERKENS, J. (2016). Non-reversible Metropolis–Hastings. *Stat. Comput.* **26** 1213–1228. MR3538633

BIERKENS, J. (2017). Computer experiments accompanying J. Bierkens, P. Fearnhead and G. Roberts, the Zig-Zag process and super-efficient sampling for Bayesian analysis of big data. Available at https://github.com/jbierkens/zigzag-experiments. Date accessed: 20-10-2017.

BIERKENS, J., FEARNHEAD, P. and ROBERTS, G. (2018). Supplement to "The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data." DOI:10.1214/18-AOS1715SUPP.

BIERKENS, J. and ROBERTS, G. (2017). A piecewise deterministic scaling limit of lifted Metropolis–Hastings in the Curie–Weiss model. *Ann. Appl. Probab.* **27** 846–882. MR3655855

BIERKENS, J., ROBERTS, G. O. and ZITT, P.-A. (2017). Ergodicity of the zigzag process. Preprint. Available at arXiv:1712.09875.

BOUCHARD-CÔTÉ, A., VOLLMER, S. J. and DOUCET, A. (2017). The bouncy particle sampler: A non-reversible rejection-free Markov chain Monte Carlo method. *J. Amer. Statist. Assoc.* To appear. Available at arXiv:1510.02451.

CHEN, T.-L. and HWANG, C.-R. (2013). Accelerating reversible Markov chains. *Statist. Probab. Lett.* **83** 1956–1962. MR3079029

DELIGIANNIDIS, G., BOUCHARD-CÔTÉ, A. and DOUCET, A. (2017). Exponential ergodicity of the bouncy particle sampler. Preprint. Available at arXiv:1705.04579.

DUANE, S., KENNEDY, A. D., PENDLETON, B. J. and ROWETH, D. (1987). Hybrid Monte Carlo. *Phys. Lett. B* **195** 216–222.

DUBEY, K. A., REDDI, S. J., WILLIAMSON, S. A., POCZOS, B., SMOLA, A. J. and XING, E. P. (2016). Variance reduction in stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems* 1154–1162.

DUNCAN, A. B., LELIÈVRE, T. and PAVLIOTIS, G. A. (2016). Variance reduction using nonreversible Langevin samplers. *J. Stat. Phys.* **163** 457–491. MR3483241

FEARNHEAD, P., BIERKENS, J., POLLOCK, M. and ROBERTS, G. O. (2018). Piecewise deterministic Markov processes for continuous-time Monte Carlo. *Statist. Sci.* **33** 386–412. MR3843382

FONTBONA, J., GUÉRIN, H. and MALRIEU, F. (2012). Quantitative estimates for the long-time behavior of an ergodic variant of the telegraph process. *Adv. in Appl. Probab.* **44** 977–994. MR3052846

FONTBONA, J., GUÉRIN, H. and MALRIEU, F. (2016). Long time behavior of telegraph processes under convex potentials. *Stochastic Process. Appl.* **126** 3077–3101. MR3542627

GIBSON, M. A. and BRUCK, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A* **104** 1876–1889.

HASTINGS, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109. MR3363437

HWANG, C.-R., HWANG-MA, S.-Y. and SHEU, S. J. (1993). Accelerating Gaussian diffusions. *Ann. Appl. Probab.* **3** 897–913. MR1233633

JACOB, P. E. and THIERY, A. H. (2015). On nonnegative unbiased estimators. *Ann. Statist.* **43** 769–784. MR3319143

JOHNSON, R. A. (1970). Asymptotic expansions associated with posterior distributions. *Ann. Math. Stat.* **41** 851–864. MR0263198

LEWIS, P. A. W. and SHEDLER, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Nav. Res. Logist. Q.* **26** 403–413. MR0546120

LI, C., SRIVASTAVA, S. and DUNSON, D. B. (2017). Simple, scalable and accurate posterior interval estimation. *Biometrika* **104** 665–680. MR3694589

MA, Y.-A., CHEN, T. and FOX, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* 2917–2925.

MACLAURIN, D. and ADAMS, R. P. (2014). Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proceedings of the* 30*th Conference on Uncertainty in Artificial Intelligence* AUAI Press, Arlington, VA.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21** 1087.

MONMARCHÉ, P. (2014). Hypocoercive relaxation to equilibrium for some kinetic models via a third order differential inequality. Available at arXiv:1306.4548.

NEAL, R. M. (1998). Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In *Learning in Graphical Models* 205–228. Springer, Berlin.

NEISWANGER, W., WANG, C. and XING, E. (2013). Asymptotically exact, embarrassingly parallel MCMC. Available at arXiv:1311.4780.

PAKMAN, A., GILBOA, D., CARLSON, D. and PANINSKI, L. (2016). Stochastic bouncy particle sampler. Preprint. Available at arXiv:1609.00770.

PETERS, E. A. J. F. and DE WITH, G. (2012). Rejection-free Monte Carlo sampling for general potentials. *Phys. Rev. E* (3) **85** 1–5.

POLLOCK, M., FEARNHEAD, P., JOHANSEN, A. M. and ROBERTS, G. O. (2016). The scalable Langevin exact algorithm: Bayesian inference for big data. Available at arXiv:1609.03436.

QUIROZ, M., VILLANI, M. and KOHN, R. (2015). Speeding up MCMC by efficient data subsampling. Riksbank Research Paper Series 121.

REY-BELLET, L. and SPILIOPOULOS, K. (2015). Irreversible Langevin samplers and variance reduction: A large deviations approach. *Nonlinearity* **28** 2081–2103. MR3366637

ROBERTS, G. O. and TWEEDIE, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2** 341–363. MR1440273

SCOTT, S. L., BLOCKER, A. W., BONASSI, F. V., CHIPMAN, H. A., GEORGE, E. I. and MCCULLOGH, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *Int. J. Manag. Sci. Eng. Manag.* **11** 78–88.

TURITSYN, K. S., CHERTKOV, M. and VUCELJA, M. (2011). Irreversible Monte Carlo algorithms for efficient sampling. *Phys. D* **240** 410–414.

VOLLMER, S. J., ZYGALAKIS, K. C. and TEH, Y. W. (2016). Exploration of the (non-)asymptotic bias and variance of stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.* **17** 1–48. MR3555050

WANG, X. and DUNSON, D. B. (2013). Parallelizing MCMC via Weierstrass sampler. Available at arXiv:1312.4605.

WELLING, M. and TEH, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the* 28*th International Conference on Machine Learning* (*ICML*-11) 681–688.

J. BIERKENS
DELFT INSTITUTE OF APPLIED MATHEMATICS
VAN MOURIK BROEKMANWEG 6
2628 XE DELFT
THE NETHERLANDS
E-MAIL: joris.bierkens@tudelft.nl

P. FEARNHEAD
DEPARTMENT OF MATHEMATICS AND STATISTICS
FYLDE COLLEGE
LANCASTER UNIVERSITY
LANCASTER, LA1 4YF
UNITED KINGDOM
E-MAIL: p.fearnhead@lancs.ac.uk

G. ROBERTS
DEPARTMENT OF STATISTICS
UNIVERSITY OF WARWICK
COVENTRY CV4 7AL
UNITED KINGDOM
E-MAIL: gareth.o.roberts@warwick.ac.uk