Coordinate sampler: a non-reversible Gibbs-like MCMC sampler

Changye Wu1 · Christian P. Robert1,2,3

Received: 25 September 2018 / Accepted: 18 November 2019 / Published online: 26 December 2019 © Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
We derive a novel non-reversible, continuous-time Markov chain Monte Carlo sampler, called Coordinate Sampler, based on a piecewise deterministic Markov process, which is a variant of the Zigzag sampler of Bierkens et al. (Ann Stat 47(3):1288–1320, 2019). In addition to providing a theoretical validation for this new simulation algorithm, we show that the Markov chain it induces exhibits geometrical ergodicity convergence, for distributions whose tails decay at least as fast as an exponential distribution and at most as fast as a Gaussian distribution. Several numerical examples highlight that our coordinate sampler is more efficient than the Zigzag sampler, in terms of effective sample size.

Keywords
Markov chain Monte Carlo · Piecewise deterministic Markov processes · Zigzag sampling · Gibbs sampling

1 Introduction
A powerful and generic sampling technique, the Markov chain Monte Carlo (MCMC) method, (see, e.g., Robert and Casella, 2004) has been widely exploited in computational statistics to become a standard tool in Bayesian inference, where posterior distributions are often analytically intractable and at best known up to a normalizing constant. However, almost all existing MCMC algorithms, such as the Metropolis-Hastings algorithm (MH), the Hamiltonian Monte Carlo (HMC) (Neal et al. 2011) and Metropolis adjusted Langevin algorithm (MALA), satisfy detailed balance conditions, dating back to Metropolis et al. (1953) and Hastings (1970). Recently, a different technology of MCMC sampling—piecewise deterministic Markov process (PDMP)—was introduced in computational statistics, towards removing reversibility constraints. The basic theory of PDMP was developed in Davis (1984) and Davis (1993), while an application to computational statistics was implemented by, e.g., Peters et al. (2012), Bierkens et al. (2019), and Bouchard-Côté et al. (2018).

Since piecewise deterministic Markov processes for sampling from distributions was introduced by Peters et al. (2012), PDMP-based, continuous-time, non-reversible, MCMC algorithms have become relevant tools, from applied probability (Bierkens and Roberts 2017; Fontbona et al. 2016) to physics (Peters et al. 2012; Harland et al. 2017; Michel et al. 2014), to statistics (Bierkens et al. 2019; Fearnhead et al. 2018; Bierkens et al. 2018; Bouchard-Côté et al. 2018; Michel and Sénécal 2017; Vanetti et al. 2017; Pakman et al. 2017). However, almost all existing PDMP-based MCMC samplers are based on two original versions: the Bouncy Particle Sampler (BPS) of Bouchard-Côté et al. (2018) and the Zigzag sampling of Bierkens et al. (2019).

Bouchard-Côté et al. (2018) exhibit that BPS can provide state-of-the-art performance compared with the reference HMC for high dimensional distributions, while Bierkens et al. (2019) show that the PDMP-based sampler is easier to scale in big data settings, without introducing bias. Bierkens et al. (2018) considers the application of PDMP for distributions on restricted domains. Fearnhead et al. (2018) unify BPS and Zigzag samplers within the framework of PDMPs: they propose a choice of the process velocity, at event times, over the unit sphere, based on the angle between this velocity and the gradient of the potential function (this perspective...
relates to the transition dynamics used here). To overcome the main difficulty met by PDMP-based samplers, namely the simulation of time-inhomogeneous Poisson processes, Sherlock and Thiery (2017) and Vanetti et al. (2017) resort to a discretization of such continuous-time samplers. Furthermore, a pre-conditioning of the velocity set is shown to accelerate the algorithms, see Pakman et al. (2017).

In this article, we propose the Coordinate Sampler (CS), a novel PDMP-based MCMC sampler that is a variant of the Zigzag sampler (ZS) of Bierkens et al. (2019). However, it differs from ZS in three significant aspects. First, the velocity set considered in the coordinate sampler consists of an orthonormal basis of the Euclidean space $\mathbb{R}^d$, while the one in the Zigzag sampler is restricted to $[-1, 1]^d$, if $d$ denotes the dimension of the target distribution. Second, the event rate function in the Zigzag sampler is much larger than the one for the coordinate sampler, especially for high dimensional targets. This means that events occur more frequently in the Zigzag sampler and hence this lowers its efficiency compared with our approach. Thirdly, the coordinate sampler targets only one component at a time when exploring the target space, and it keeps the other components unchanged, while the Zigzag sampler modifies all components at the same time.

The outline of this article is as follows. Section 2 introduces the necessary background of PDMP-based MCMC samplers, the techniques used in its implementation, and two specified samplers, BPS and ZS. Section 3 describes the methodology behind the coordinate sampler, provides some theoretical validation along with a proof of geometrical ergodicity, obtained under quite mild conditions, and compares this proposal with the Zigzag sampler in an informal analysis. Section 4 further compares the efficiency of both approaches on banana-shaped distributions, multivariate Gaussian distributions and a Bayesian logistic model, when effective sample size is measuring efficiency. Section 5 concludes by pointing out further research directions about this special MCMC sampler.

### 2 Piecewise deterministic Markov process

In this section, we briefly introduce piecewise deterministic Markov processes (PDMP) and describe how to apply this methodology into statistical computing problems. We describe two specified PDMP-based MCMC samplers: the bouncy particle sampler (BPS) and the Zigzag sampler (ZS).

#### 2.1 PDMP-based sampler

Let $\pi$ be the continuous target distribution over $\mathbb{R}^d$ and for convenience sake, denote $\pi(x)$ for the probability density function of $\pi$, when $x \in \mathbb{R}^d$. We define $U(x)$ as the potential function of $\pi(x)$, that is, $\pi(x) \propto \exp\{-U(x)\}$, with $U$ positive. In the PDMP framework, an auxiliary variable, $V \in \mathcal{V}$, is introduced and a PDMP-based sampler explores the augmented state space $\mathbb{R}^d \times \mathcal{V}$, targeting a variable $Z = (X, V)$ with distribution $\rho(dx, dv)$ over $\mathbb{R}^d \times \mathcal{V}$ as its invariant distribution. By construction, the distribution $\rho$ enjoys $\pi$ as its marginal distribution in $x$. In practice, the existing PDMP-based samplers choose $\mathcal{V}$ to be the Euclidean space $\mathbb{R}^d$, the sphere $S^{d-1}$, or the discrete set $\mathcal{V} = \{v = (v_1, \ldots, v_d)|v_i \in \{-1, 1\}, i = 1, \ldots, d\}$. Following Fearnhead et al. (2018), a piecewise deterministic Markov process $Z_t = (X_t, V_t)$ consists of three distinct components: a deterministic dynamic between events, an event occurrence rate, and a transition dynamic at event times. Specifically,

1. **Deterministic dynamic** between two events, the Markov process evolves deterministically, according to some ordinary differential equation: $\frac{dZ_t}{dt} = \Psi(Z_t)$.
2. **Event occurrence rate** an event occurs at time $t$ with rate $\lambda(Z_t)$.
3. **Transition dynamic** At an event time, $\tau$, the state prior to $\tau$ is denoted by $Z_{\tau^-}$, with the new state being generated by $Z_\tau \sim Q(\cdot|Z_{\tau^-})$.

Here, an “event” refers to an occurrence of a time-inhomogeneous Poisson process with rate $\lambda(\cdot)$ (Kingman 1992). Following (Davis 1993, Theorem 26.14), this Markov process had an extended generator equal to

$$L f(x) = \nabla f(x) \cdot \Psi(x) + \lambda(x) \int_{\mathcal{V}} \left[ f(x') - f(x) \right] Q(dx' | x)$$

In order to guarantee invariance with respect to $\rho(dx)$, the extended generator need satisfy $\int L f(x) \rho(dx) = 0$ for all $f$ in an appropriate function class on $\mathbb{R}^d \times \mathcal{V}$ (Davis 1993, Theorem 34.7).

#### 2.2 Implementation of a PDMP-based sampler

In practice, choosing an appropriate deterministic dynamic, an event rate and a transition dynamic, produces a Markov chain with invariant distribution $\rho(dx)$. As for regular MCMC, generating such a Markov chain for a duration $T$, leads to an estimator $\frac{1}{T} \int_{t=0}^{T} h(X_t) dt$, converging to the integral of interest, $I = \int h(x) \pi(dx)$, by the Law of Large Numbers for Markov processes (Glynn and Haas 2006), under appropriate assumptions. More specifically,

$$\frac{1}{T} \int_{t=0}^{T} g(Z_t) dt \longrightarrow \int g(x) \rho(dx), \quad \text{as } T \to \infty$$

Springer
and defining \( g(z) = g(x, v) := h(x) \) induces, as \( T \to \infty \),
\[
\frac{1}{T} \int_{t=0}^{T} h(X_t) dt = \frac{1}{T} \int_{t=0}^{T} g(Z_t) dt \\
\to \int g(z) \rho(dz) = \int h(x) \pi(dx),
\]
where \( \rho(dv|x) \) is the conditional distribution of the variable \( V \), given \( X = x \). Algorithm 1 contains a pseudo-code reproducing the simulation of a PDMP in practice:

**Algorithm 1** General PDMP-based sampler

1: **Input**: start at position \( x_0 \), velocity \( v_0 \) and simulation time threshold \( T^{\text{total}} \).
2: Generate a set of event times of the PDMP \( \{t_0, t_1, \ldots, t_M\} \) and their associated states \( \{Z_{t_0}, Z_{t_1}, \ldots, Z_{t_M}\} \), where \( t_0 = 0, t_{M-1} < T^{\text{total}} \), \( t_M = T^{\text{total}} \). Set \( Z_0 = (X_0, V_0) \).
3: Set \( t \leftarrow 0, T \leftarrow 0, m \leftarrow 0, \tau_m \leftarrow 0 \).
4: **while** \( T < T^{\text{total}} \) **do**
5: \( m \leftarrow m + 1 \).
6: \( u \leftarrow \text{Uniform}(0, 1) \).
7: Solve the equation
\[
\int_{0}^{\eta_m} \lambda_m(t) dt = -\log(u), \tag{2}
\]

10: to obtain \( \eta_m \), where \( \lambda_m(t) = \lambda \left( \Phi_t(X_{\tau_m-1}, V_{\tau_m-1}) \right) \), and \( \Phi_t(\cdot, \cdot) \) is the flow of the deterministic dynamic.
8: \( \tau_m \leftarrow \tau_{m-1} + \eta_m, T \leftarrow t \leftarrow t, Z_{\tau_m} = \Phi_{\tau_m}(X_{\tau_m-1}, V_{\tau_m-1}), Z_{\tau_m} \sim Q(Z_{\tau_m}, \cdot) \).
9: **Output**: A trajectory of the Markov chain up to time \( \tau_M \), \( \{Z_t\}_{t=0}^{\tau_M} \).

where \( Z_t = \Phi_{\tau_{m+1}}(Z_{\tau_m}) \) for \( \tau_m \leq t < \tau_{m+1} \).

In many cases, evaluating the path integral \( \int_{t=0}^{T} h(Z_t) dt \) may however be expensive, or even impossible, and a discretization of the simulated trajectory is a feasible alternative. This means estimating the quantity of interest, \( I \), by the following estimator
\[
\hat{I} = \frac{1}{N} \sum_{n=1}^{N} h(X_{nT}/N).
\]

In practice, the main difficulty in implementing a PDMP-based sampler is the generation of the occurrence times of the associated time-inhomogeneous Poisson process with event rate \( \lambda(\cdot) \). Fortunately, the following two theorems alleviate this difficulty.

**Theorem 1** (Superposition Theorem) (Kingman 1992) Let \( \Pi_1, \Pi_2, \ldots, \) be a countable collection of independent Poisson processes on state space \( \mathbb{R}^+ \) and let \( \Pi_n \) have rate \( \lambda_n(\cdot) \) for each \( n \). If \( \sum_{n=1}^{\infty} \lambda_n(t) < \infty \) for all \( t \), then the superposition
\[
\Pi = \bigcup_{n=1}^{\infty} \Pi_n
\]
is a Poisson process with rate
\[
\lambda(t) = \sum_{n=1}^{\infty} \lambda_n(t)
\]

**Theorem 2** (Thinning Theorem) (Lewis and Shedler 1979) Let \( \lambda : \mathbb{R}^+ \to \mathbb{R}^+ \) and \( \Lambda : \mathbb{R}^+ \to \mathbb{R}^+ \) be continuous functions such that \( \lambda(t) \leq \Lambda(t) \) for all \( t \geq 0 \). Let \( \tau^1, \tau^2, \ldots, \) be the increasing finite or infinite sequence of a Poisson process with rate \( \Lambda(\cdot) \). For all \( i \), if the point \( \tau^i \) is removed from the sequence with probability \( 1 - \lambda(\tau^i)/\Lambda(\tau^i) \), then the remaining points \( \tau^1, \tau^2, \ldots \) form a non-homogeneous Poisson process with rate \( \lambda(\cdot) \).

In practice, according to Theorem 1, we can split the event rate function into the summation of several event sub-rate functions and take the minimum of the first arrival times of the Poisson processes, induced by these sub-rate functions, as the desired event time interval. In addition, in order to generate the first arrival times of the sub-Poisson processes, we can choose an upper bound function, whose induced Poisson process is easy to simulate, for each sub-rate function, and resort to Theorem 2.

### 2.3 Two reference PDMD-based samplers

Almost all existing PDMD-based samplers are based on two specific versions, both of which rely on linearly deterministic dynamics, a feature that facilitates the determination of the state of the Markov chain between Poisson events. Vanetti et al. (2017) uses Hamiltonian dynamics over an approximation of the target distribution to accelerate the bouncy particle sampler, but the efficiency of that modification depends on the quality of the approximation and it only transfers the difficulty from setting the deterministic dynamics to computing the event rate function.

#### 2.3.1 Bouncy particle sampler

For the Bouncy Particle sampler, as described by Bouchard-Côté et al. (2018), the velocity set \( \mathcal{V} \) is either the Euclidean space \( \mathbb{R}^d \), or the unit sphere \( S^{d-1} \). The associated augmented target distribution is either \( \rho(dx, dv) = \pi(dx) \mathcal{N}(dv|0, I_d) \), or \( \rho(dx, dv) = \pi(dx) \mathcal{U}_{S^{d-1}}(dv) \), where \( \mathcal{N}(\cdot|0, I_d) \) represents the standard \( d \)-dimensional Gaussian distribution and \( \mathcal{U}_{S^{d-1}}(dv) \) denotes the uniform distribution over \( S^{d-1} \), respectively. The corresponding deterministic dynamic is
\[
\frac{dX_t}{dt} = V_t, \quad \frac{dV_t}{dt} = 0.
\]
the event rate satisfies \( \lambda(\mathbf{x}, \mathbf{v}) = \langle \mathbf{v}, \nabla U(\mathbf{x}) \rangle_+ + \lambda^{\text{ref}} \), where \( \lambda^{\text{ref}} \) is a user-chosen non-negative constant and the transition dynamic is as

\[
Q((d\mathbf{x}', d\mathbf{v}')(\mathbf{x}, \mathbf{v})) = \frac{(\mathbf{v}, \nabla U(\mathbf{x}))+}{\lambda(\mathbf{x}, \mathbf{v})} \delta(\mathbf{x}) \delta_{R_{\mathbf{v}}(\mathbf{x})}(d\mathbf{v}') + \frac{\lambda^{\text{ref}}}{\lambda(\mathbf{x}, \mathbf{v})} \delta(\mathbf{x}) \varphi(d\mathbf{v}')
\]

where \( \varphi(d\mathbf{v}') = U_{\mathbf{v}}-1(d\mathbf{v}) \) or \( \varphi(d\mathbf{v}') = N(d\mathbf{v}|0, I_d) \), depending on the choice of the velocity set, and the operator \( R_{\mathbf{w}} \), for any non-zero vector \( \mathbf{w} \in \mathbb{R}^d \setminus \{0\} \), is \( R_{\mathbf{w}} \mathbf{v} = \mathbf{v} - 2 \langle \mathbf{w}, \mathbf{v} \rangle \mathbf{w} \).

### 2.3.2 Zigzag sampler

For the Zigzag sampler (Bierkens et al. 2019), the velocity set, \( \mathcal{V} \), is the discrete set \( \{v_i = (v_{i1}, \ldots, v_{id}) | v_i \in [-1, 1], i = 1, \ldots, d\} \) and \( \rho(d\mathbf{x}, d\mathbf{v}) = \pi(d\mathbf{x}) \varphi(d\mathbf{v}) \), where \( \varphi \) is the uniform distribution over \( \mathcal{V} \). ZS uses the same linear deterministic dynamics as BPS. Its event rate is \( \lambda(\mathbf{x}) = \sum_{i=1}^{d} \lambda_i(\mathbf{x}, \mathbf{v}) = \sum_{i=1}^{d} \left\{ v_i \nabla_i U(\mathbf{x}) \right\}_+ + \lambda^{\text{ref}}_i \), where the \( \lambda^{\text{ref}}_i \)'s are user-chosen non-negative constants. The transition dynamics is

\[
Q((d\mathbf{x}', d\mathbf{v}')(\mathbf{x}, \mathbf{v})) = \sum_{i=1}^{d} \frac{\lambda_i(\mathbf{x}, \mathbf{v})}{\lambda(\mathbf{x}, \mathbf{v})} \delta(\mathbf{x}) \delta_{F_i}(d\mathbf{v}'),
\]

where \( F_i \) denotes the operator that flips the \( i \)-th component of \( \mathbf{v} \) and keeps the others unchanged. In practice, ZS relies on the Superposition Theorem: At each event time, ZS simulates \( d \) Poisson processes, with rates \( \lambda_i(\mathbf{x}+i\mathbf{v}, \mathbf{v}) \), computes their first occurrence time, and takes their minimum, e.g., the \( i \)-th, for the duration between current and next events, and flips the \( i \)-th component of the velocity \( \mathbf{v} \).

### 3 Coordinate sampler

We now describe the coordinate sampler (CS), in which only one component of \( \mathbf{x} \) evolves and the others remain inactive between event times. For CS, the velocity set \( \mathcal{V} \) is chosen to be \( \{ \pm e_i, i = 1, \ldots, d \} \), where \( e_i \) is the vector with \( i \)-th component equal to one and the others set to zero. The augmented target distribution is \( \rho(d\mathbf{x}, d\mathbf{v}) = \pi(d\mathbf{x}) \varphi(d\mathbf{v}) \), with \( \varphi(d\mathbf{v}) \) the uniform distribution over \( \mathcal{V} \). The PDMP characteristics of CS are thus

1. **Deterministic dynamic**

\[
\frac{d\mathbf{X}_i}{dt} = \mathbf{v}_i, \quad \frac{d\mathbf{V}_i}{dt} = 0.
\]

### 3.1 Theoretical properties of the coordinate sampler

We now establish that CS is associated with the augmented target distribution, \( \rho(d\mathbf{x}, d\mathbf{v}) \), as its invariant distribution under the condition that \( U : \mathbb{R}^d \rightarrow \mathbb{R}^+ \) is \( C^1 \). Furthermore, under the following assumptions, the Markov process induced by CS is \( \mathcal{V} \)-uniformly ergodic for the Lyapunov function

\[
V(\mathbf{x}, \mathbf{v}) = e^{U(\mathbf{x})/2} / \sqrt{\lambda^{\text{ref}} + \langle \nabla U(\mathbf{x}), -\mathbf{v} \rangle_+},
\]

which was also used in Deligiannidis et al. (2019).

**Theorem 3** For any positive \( \lambda^{\text{ref}} > 0 \), the PDMP produced by CS enjoys \( \rho(d\mathbf{x}, d\mathbf{v}) \) as its unique invariant distribution, provided the potential \( U \) is \( C^1 \).

It is easy to check that the generator of CS, \( \mathcal{L} \), satisfies

\[
\int \mathcal{L} f(\mathbf{z}) \rho(d\mathbf{z}) = 0.
\]
for all functions $f$ in its extended generator, which means
that $\rho$ is an invariant distribution of CS (Davis 1993, Theorem 34.7). Uniqueness follows from the positivity of $\lambda^{\text{ref}}$, which enables the Markov process to reach any state $(x^*, v^*)$ from any starting state $(x_0, v_0)$, in finite time (details of the proof are provided as supplementary material). In practice, it appears that the constraint $\lambda^{\text{ref}} > 0$ is unnecessary for convergence in many examples.

**Assumption** Assume $U : \mathbb{R}^d \to \mathbb{R}^+$ satisfy the following conditions, reproduced from Deligiannidis et al. (2019),

A.1 $\frac{\partial^2 U(x)}{\partial x_i \partial x_j}$ is locally Lipschitz continuous for all $i, j$,
A.2 $\int |\nabla U(x)| \pi(dx) < \infty$,
A.3 $\lim_{|x| \to \infty} e^{U(x)/2} / \sqrt{|\nabla U(x)|} > 0$,
A.4 $V \geq c_0$ for some positive constant $c_0$.

**Conditions:** We set conditions

C.1 $\lim_{|x| \to \infty} |\nabla U(x)| = \infty$, $\lim_{|x| \to \infty} \|\Delta U(x)\| \leq \alpha_1 < \infty$ and $\lambda^{\text{ref}} > \sqrt{8\alpha_1}$.
C.2 $\lim_{|x| \to \infty} \|\nabla U(x)\| = 2\alpha_2 > 0$, $\lim_{|x| \to \infty} \|\Delta U(x)\| = 0$ and $\lambda^{\text{ref}} < \frac{\alpha_2}{16\alpha_1}$.

where C.1 corresponds to distributions whose tails decay at rate $O(|x|^\beta)$, where $1 < \beta \leq 2$, and C.2 to distributions with tails of order $O(|x|)$.

**Theorem 4** Suppose assumptions A.1 – A.4 hold, $\lambda^{\text{ref}} > 0$, and one of the conditions C.1 or C.2 hold, then CS is $V$-uniformly ergodic: There exist constants $\Gamma < \infty$ and $0 < \gamma < 1$, such that

$$\|P^t(z, \cdot) - \rho\|_V \leq V(z) \Gamma^t \gamma^t,$$

where $P^t(z, \cdot)$ is the distribution of the Markov chain with starting state $z$ at time $t$, and the norm $\| \cdot \|_V$ is defined by

$$\|\mu\|_V = \sup_{|f| < V} \int f(z) \mu(dz).$$

The proof appears in the supplementary material, based on techniques quite similar to those in Deligiannidis et al. (2019).

**3.2 An informal comparison between Zigzag and coordinate samplers**

For CS, each event time sees a change of a single component of $X$, in contrast with ZS, which modifies all components at the same time. At first glance, this gives the impression that CS is less efficient than ZS in its exploration of the target space, because of this restriction. However, this intuition is misleading: Suppose that the $\lambda_i$’s, $i = 1, \ldots, d$ in ZS and $\lambda$ in CS are of a similar scale, for instance taking the expected duration between two Poisson events to be the same value $\ell$. Assume further that computing an occurrence time has the same computation cost, $c_1$, for all Poisson processes and evaluating each partial derivative of $U$ also has the same cost, $c_2$. In ZS, the event rate is the summation of the rates $\lambda_1, \ldots, \lambda_d$. Therefore, the time duration between two events is $O(\ell^c)$ and the induced computation cost is $O(dc_1)$. Thus, when each component of $X$ evolves for a time duration $\ell$ this costs $O(d^2c_1 + d c_2)$ for ZS. By contrast, in CS, a $O(dc_1 + d^2c_2)$ computation cost will result from the Markov chain moving for a duration time $O(d\ell)$. Hence, the computation cost for monitoring each component for a time duration $\ell$ is now $O(dc_1 + d^2c_2)$. Since $c_1$ is the computation cost of sampling an event time from a Poisson process, it should be larger than $c_2$ in most cases. Hence, resulting in the informal conclusion that CS should often be more efficient than ZS and that such an improvement, whenever it happens, should be more noticeable for cases where $c_1$ is much larger than $c_2$. The following numerical experiments further confirm this point.

**4 Numerical experiments**

In this section, we compare the efficiency of both samplers over benchmarks (a banana-shaped distribution, two multivariate Gaussian distributions, and a Bayesian logistic
Fig. 2 The upper plot shows the ESS comparison for the MVN1 model and the lower plot for the MVN2 model. The $x$-axis is indexed by the dimension $d$ of the distribution, while the $y$-axis measures the efficiency ratios of CS over ZS in terms of minimum, mean, median and maximum of ESS across the components and over the number of event time sampled from the corresponding Poisson processes.

In each model, we run both samplers for the same computer time or the same number of calls of the event rate functions and we compare their efficiency in terms of effective sample size (ESS) (Liu 2008) per second or per call of an event rate function. The models are reproduced forty times to produce an averaged efficiency ratio, namely the ratio of ESS per second for CS over the one for ZS. We use the function `ess` of package `mcmcse` in R to compute ESS of samples. The code for this paper is available online at https://github.com/wcythh/CoordinateSampler.

**Banana-Shaped Distribution** The target distribution is a 2-dimensional banana-shaped distribution with density

$$
\pi(x) \propto \exp\left\{- (x_1 - 1)^2 - \kappa(x_2 - x_1^2)^2\right\}
$$

where $\kappa$ controls the similarity between $x_2$ and $x_1^2$. A high $\kappa$ enforces the approximate constraint $x_2 \simeq x_1^2$. The comparison between Zigzag and coordinate samplers runs over the configurations $2^{-2} \leq \kappa \leq 2^5$. With an increase in $\kappa$, the distribution becomes more difficult to simulate and the event rate functions in CS and ZS make the generations of time durations more costly. Figure 1 shows that CS is more...
efficient than ZS across a large range of $\kappa$ in this model.

**Strongly Correlated Multivariate Gaussian Distribution (MVN1)** Here, the target is a multivariate Gaussian distribution with zero mean and covariance matrix equal to $A \in \mathbb{R}^{d \times d}$, where $A_{ii} = 1$ and $A_{ij} = 0.9, i \neq j$. We consider the values $d = 10, 20, \ldots, 100$ in our comparison of the sampling methods.

**Correlated Multivariate Gaussian Distribution (MVN2)** In this scenario, the target distribution is again a multivariate Gaussian distribution with zero mean and covariance matrix such that $A_{ii} = 1$ and $A_{ij} = 0.9^{|i-j|}$. Once again, the comparison runs for $d = 10, \ldots, 100$.

Figure 2 presents the comparison between CS and ZS for both models MVN1 and MVN2 in terms of the minimal ESS, mean ESS, median ESS and maximal ESS taken across all $d$ components per generation of occurrence time induced by event (sub-) rate function. Figure 3 shows the comparison in terms of time-consuming of each program. Figure 2 confirms that when the sub-rate functions of ZS and rate function of CS enjoy the same scale, the required numbers of event times are $O(d)$ for CS and $O(d^2)$ for ZS, in order to make each component of the target move a length $O(\ell)$. Figure 3 shows that when the dimension of the target distribution increases, the improvement of CS over ZS will be capped by the $c_1/c_2$ ratio (as explained in Sect. 3.2).

Table 1 further compares CS with several standard MCMC algorithms for a 20-dimensional MVN2 model in terms of their Kolmogorov-Smirnov (KS) distance to the target. Since such quantities are too difficult to compute for multivariate
distributions, we computed instead marginal KS distances between samples from every algorithm and from the target, across coordinates, and take the minimum, mean, median and maximum of these distances as a summary of the efficiency of each algorithm, for identical computation times about 155 seconds. In this experiment, HMC performs best in terms of Kolmogorov-Smirnov statistic. However, among the PDMP-based MCMC algorithms, CS outperforms both ZS and BPS.

**Bayesian Logistic Model** In this example, the target is the posterior of a Bayesian logistic model under a flat prior, with no intercept. The simulated dataset contains \( N \) observations \( \{(x_{ni}, r_{ni})\}_{n=1}^{N} \), where each \( r_{ni}, n = 1, \ldots, N, i = 1, \ldots, d \), is drawn from a standard normal distribution and \( t_{ni} \) is drawn from \([0, 1]\) uniformly. The targeted density function is thus

\[
\pi(x) \propto \prod_{n=1}^{N} \frac{\exp(t_{ni}x^{T}r_{ni})}{1 + \exp(x^{T}r_{ni})}
\]

In the simulations, we set \( N = 40, d = 10 \), and \( \lambda_{i}^{\text{ref}} = 1 \) for CS, and \( \lambda_{i}^{\text{ref}} = 1, i = 1, \ldots, d \) for ZS. Figure 4 presents the comparison between the two samplers, with a massive improvement brought by our proposal.

**Log-Gaussian Cox Point Process** In this example, already implemented by Galbraith (2016), the observations \( Y = \{y_{ij}\} \) are Poisson distributed and conditionally independent given a latent intensity process \( = \{\lambda_{ij}\} \) with means \( s\lambda_{ij} = s \exp(x_{ij}) \), where \( s = 1/d^2 \). The underlying process \( X = \{x_{ij}\} \) is a Gaussian process with mean function \( m(x_{ij}) = \mu 1 \) and covariance function \( \Sigma(x_{ij}, x'_{ij}, j') = \sigma^2 \exp(-\delta(i, i', j, j'/\beta d)) \), where \( \delta(i, i', j, j') = (i - i')^2 + (j - j')^2 \). In our experiment, we set \( d = 20 \) and choose \( \sigma^2 = 1.91, \mu = \log(126) - \sigma^2/2 \) and \( \beta = 1/6 \). The target is conditional on the observations \( Y \),

\[
\pi(X|Y, \mu, \sigma, \beta) \propto \exp \left\{ \sum_{i,j=1}^{d} (y_{ij}x_{ij} - s \exp(x_{ij})) \right\}
- \frac{1}{2} (X - \mu 1)^T \Sigma^{-1} (X - \mu 1)
\]

The smaller the numerical value, the better the algorithm performs.

| Sampler | Min KS | Mean KS | Median KS | Max KS |
|---------|--------|---------|-----------|--------|
| CS      | 4.02 \times 10^{-3} | 7.00 \times 10^{-3} | 7.04 \times 10^{-3} | 10.02 \times 10^{-3} |
| ZS      | 9.17 \times 10^{-3} | 16.53 \times 10^{-3} | 16.07 \times 10^{-3} | 24.34 \times 10^{-3} |
| BPS     | 4.94 \times 10^{-3} | 8.98 \times 10^{-3} | 9.04 \times 10^{-3} | 12.58 \times 10^{-3} |
| HMC     | 1.26 \times 10^{-3} | 2.12 \times 10^{-3} | 2.04 \times 10^{-3} | 3.31 \times 10^{-3} |

**5 Conclusion**

We have introduced and studied the coordinate sampler as an alternative to the Zigzag sampler of Bierkens et al. (2019) and compared the efficiencies of the two samplers in terms of effective sample size over several simulation experiments. In all examples, CS exhibits a higher efficiency, while enjoying the same ergodicity guarantees. While our intuition about the advantage of a component-wise implementation led to our proposal, exhibiting a theoretical reason for this improvement requires further investigation.

We also stress that, among PDMP-based MCMC samplers, CS is quite easy to scale for big data problems, albeit this is also the case for the Zigzag sampler. In addition, taking advantage of the techniques exposed in Bierkens et al.
CS and ZS samplers for the same computation time, when targeting a log-Gaussian Cox point process. Only the first two components are represented here.

The plots of log-density and the final component of the samples generated by CS (red) and ZS (blue) for the same computation time, when targeting a log-Gaussian Cox point process. (Color figure online)

(2018), CS can also be implemented for distributions defined on restricted domains. In such settings, since only one component of the target distribution is active between Poisson events, the efficiency of CS relatively to ZS may suffer, especially in cases when the variances across the components are of different magnitudes. An appropriate re-parametrization of the target distribution should however alleviate this problem, and accelerate CS, which amounts to a pre-conditioning of the velocity set. An interesting extension that needs further investigation is to build CS that take advantage of the curvature of the target by Riemann manifold techniques as in Girolami and Calderhead (2011).

References

Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A.B., Fearnhead, P., Lienart, T., Roberts, G., Vollmer, S.J.: Piecewise deterministic Markov processes for scalable Monte Carlo on restricted domains. Stat. Probab. Lett. 136, 148–154 (2018)

Bierkens, J., Fearnhead, P., Roberts, G.: The Zig-zag process and super-efficient sampling for Bayesian analysis of big data. Ann. Stat. 47(3), 1288–1320 (2019)

Bierkens, J., Roberts, G., et al.: A piecewise deterministic scaling limit of lifted Metropolis-Hastings in the Curie–Weiss model. Ann. Appl. Probab. 27(2), 846–882 (2017)

Bouchard-Côté, A., Vollmer, S.J., Doucet, A.: The bouncy particle sampler: a nonreversible rejection-free Markov chain Monte Carlo method. J. Am. Stat. Assoc. 13, 1–13 (2018)

Davis, M.H.: Piecewise-deterministic Markov processes: a general class of non-diffusion stochastic models. J. R. Stat. Soc. Ser. B Methodol. 46(3), 353–388 (1984)

Davis, M.H.: Markov Models and Optimization, vol. 49. CRC Press, Boca Raton (1993)

Deligiannidis, G., Bouchard-Côté, A., Doucet, A.: Exponential ergodicity of the bouncy particle sampler. Ann. Stat. 47(3), 1268–1287 (2019)

Fearnhead, P., Bierkens, J., Pollock, M., Roberts, G.O., et al.: Piecewise deterministic Markov processes for continuous-time Monte Carlo. Stat. Sci. 33(3), 386–412 (2018)

Fontbona, J., Guérin, H., Malrieu, F.: Long time behavior of telegraph processes under convex potentials. Stoch. Process. Appl. 126(10), 3077–3101 (2016)

Galbraith, N.: On event-chain Monte Carlo methods. Master’s thesis, Department of Statistics, Oxford University, p. 9 (2016)

Girolami, M., Calderhead, B.: Riemann manifold Langevin and Hamiltonian Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat. Methodol. 73(2), 123–214 (2011)
Glynn, P.W., Haas, P.J.: Laws of large numbers and functional central limit theorems for generalized semi-Markov processes. Stoch. Models 22(2), 201–231 (2006)

Harland, J., Michel, M., Kampmann, T.A., Kierfeld, J.: Event-chain Monte Carlo algorithms for three- and many-particle interactions. EPL Europhys. Lett. 117(3), 30001 (2017)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1), 97–109 (1970)

Kingman, J.F.C.: Poisson Processes, vol. 3. Clarendon Press, Oxford (1992)

Lewis, P.A., Shedler, G.S.: Simulation of nonhomogeneous Poisson processes by thinning. Naval Res. Logist. 26(3), 403–413 (1979)

Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, Berlin (2008)

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. 21(6), 1087–1092 (1953)

Michel, M., Kapfer, S.C., Krauth, W.: Generalized event-chain Monte Carlo: constructing rejection-free global-balance algorithms from infinitesimal steps. J. Chem. Phys. 140(5), 054116 (2014)

Michel, M., Sénécal, S.: Forward event-chain Monte Carlo: a general rejection-free and irreversible Markov chain simulation method. arXiv preprint arXiv:1702.08397 (2017)

Neal, R.M., et al.: MCMC using Hamiltonian dynamics. In: Brooks, S., Gelman, A., Jones, G., Meng, X.-L. (eds.) Handbook of Markov Chain Monte Carlo, vol. 2(11). CRC, New York (2011)

Pakman, A., Gilboa, D., Carlson, D., Paninski, L.: Stochastic bouncy particle sampler. In: Proceedings of the 34th International Conference on Machine Learning, PMLR, vol. 70, pp. 2741–2750 (2017)

Peters, E.A., et al.: Rejection-free Monte Carlo sampling for general potentials. Phys. Rev. E 85(2), 026703 (2012)

Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2004)

Sherlock, C., Thiery, A.H.: A Discrete Bouncy Particle Sampler. arXiv preprint arXiv:1707.05200 (2017)

Vanetti, P., Bouchard-Côté, A., Deligiannidis, G., Doucet, A.: Piecewise Deterministic Markov Chain Monte Carlo. arXiv preprint arXiv:1707.05296 (2017)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.