SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 7, No. 2, pp. 685–719

Generalizing Parallel Replica Dynamics: Trajectory Fragments, Asynchronous Computing, and PDMPs*

David Aristoff†

Abstract. We study parallel replica dynamics in a general setting. We introduce a trajectory fragment framework that can be used to design and prove consistency of parallel replica algorithms for generic Markov processes. We use our framework to formulate a novel condition that guarantees an asynchronous algorithm is consistent. Exploiting this condition and our trajectory fragment framework, we present new synchronous and asynchronous parallel replica algorithms for piecewise deterministic Markov processes.

Key words. parallel replica dynamics, long-time dynamics, stationary distributions, asynchronous computing, piecewise deterministic Markov processes

AMS subject classifications. 65C05, 65C20, 65C40, 65Y05, 82C80

DOI. 10.1137/18M1177792

1. Introduction.

Many problems in applied sciences require the sampling of complex probability distributions. In computational chemistry—which is the main setting of this article—such distributions can arise from stochastic models of molecular dynamics [34] or chemical reaction networks [2], while obstacles to efficient sampling include high dimensionality and metastability, the latter being the tendency to become stuck in certain subsets of state space [31]. Some attempts to surmount these difficulties have been based on importance sampling and stratification [51, 57, 58, 60], interacting particles [17, 18, 19, 20], coarse graining and preconditioning [4, 35, 56], accelerated dynamics [32, 53, 61, 62], and nonreversibility [21, 25, 33, 48, 68].

This article concerns parallel replica dynamics (ParRep) [62], an accelerated dynamics method designed to overcome metastability. ParRep has two distinct advantages over many other enhanced sampling methods. First, it computes correct *dynamical* [5, 30, 61] as well as *stationary* or *equilibrium* [3, 65] quantities associated with a stochastic process. Second, ParRep is very general: it only requires mild assumptions on the underlying process. Indeed, though originally intended for Langevin dynamics [30, 61, 63], straightforward extensions of ParRep to discrete and continuous time Markov chains have appeared in [3, 5, 65].

The goal of this article is as follows. First, we introduce a new mathematical framework that may be used to design, and prove the consistency of, ParRep algorithms for Markov processes satisfying a few mild assumptions. Second, we use our framework to obtain valuable

*Received by the editors March 28, 2018; accepted for publication (in revised form) April 8, 2019; published electronically June 25, 2019.
http://www.siam.org/journals/juq/7-2/M117779.html
Funding: This work was supported by the National Science Foundation via the awards NSF-DMS-1522398 and NSF-DMS-1818726.
†Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (aristoff@rams.colostate.edu).

insights into asynchronous computing. In particular, we present specific, novel conditions that ensure an asynchronous ParRep algorithm is consistent. Lastly, we construct ParRep algorithms for piecewise deterministic Markov processes (PDMPs) in both the synchronous and asynchronous setting, leaning on our new framework to demonstrate their consistency. Asynchronous ParRep algorithms must be carefully designed since, as we show below, inaccuracies can arise when the speed of computing paths of the underlying process is coupled to the process itself.

PDMPs are emerging as a useful tool in fields as diverse as applied probability [38], computational chemistry [1, 7, 26, 28, 37, 44, 49, 50, 66], machine learning [9, 45, 67], and big data [6]. As indicated by the name, PDMPs move along deterministic paths in between random jump times. In the context of chemical reaction networks, PDMPs called *hybrid models* can be obtained by approximating fast reactions by a deterministic flow, and representing slow reactions with an appropriate Poisson process [28, 66]. The resulting PDMPs can be metastable [10, 11, 12, 13, 39, 40, 41, 42, 43], making direct simulation unattractive. Several PDMP-based algorithms have also been proposed for sampling from distributions known up to normalization—like the Boltzmann distribution or the posterior distribution in Bayesian analysis—including event chain Monte Carlo [26, 37], the zig-zag process [6], and the bouncy particle sampler [9, 38]. Below, we give a general argument suggesting these PDMPs also become metastable under certain conditions.

This article is organized as follows. Section 2 defines notation that we use throughout. In section 3, we formally define metastability in terms of quasistationary distributions. We describe ParRep in more detail, and explain what we mean by a consistent ParRep algorithm, in section 4. In section 5 we outline a general mathematical framework for ParRep, and in section 6 we study synchronous and asynchronous computing. Section 7 serves as a brief introduction to PDMPs, while section 8 outlines several ParRep algorithms for PDMPs that are based on our framework from section 5. A numerical example is in section 9. All proofs are in section 10.

2. Notation.

Throughout, $X(t)_{t\geq 0}$ is a time homogeneous Markov process, either discrete or continuous in time, with values in a standard Borel state space; U is a subset of the state space; and g is a real-valued function defined on the state space. Without explicit mention we assume all sets are measurable and all functions are bounded and measurable. We write $X(t)$ to refer to the process $X(t)_{t\geq 0}$ at time t. We denote various expectations and probabilities by $\mathbb{E}$ and $\mathbb{P}$, with the precise meaning being clear from the context. We write $\mathcal{L}$ for the probability law of a random object, with $\mathcal{L}$ above an equals sign indicating equality in law. We say a random object is a *copy* of another random object if it has the same law as that object. When we say a collection of random objects is independent we mean these objects are *mutually* independent unless otherwise specified. We define $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$, and write $\lfloor s \rfloor$ for the greatest integer less than or equal to s.

3. Metastability.

Informally, U is a metastable set for $X(t)_{t\geq 0}$ if $X(t)_{t\geq 0}$ tends to reach a local equilibrium in U much faster than it escapes from U. Local equilibrium can be understood in terms of *quasistationary distributions* (QSDs).

Definition 3.1. *Fix a subset U of state space, and consider*

$$T = \inf\{t \geq 0 : X(t) \notin U\},$$

the first time $X(t)_{t\geq 0}$ escapes U. A QSD ρ of $X(t)_{t\geq 0}$ in U satisfies $\rho(U) = 1$ and

$$\rho(A) = \mathbb{P}(X(t) \in A | \mathcal{L}(X(0)) = \rho, T > t) \tag{3.1}$$

for every $t \geq 0$ and $A \subseteq U$.

Note that ρ is supported in U. Equation (3.1) states that if $X(0)$ is distributed as ρ and $X(t)_{t\geq 0}$ does not escape from U by time t, then $X(t)$ is distributed as ρ. Throughout, we will assume the QSD of $X(t)_{t\geq 0}$ in U exists, is unique, and is the long-time distribution of $X(t)$ conditioned to never escape U. That is, we assume that for any initial distribution of $X(0)$ supported in U,

$$\rho(A) = \lim_{t\to\infty} \mathbb{P}(X(t) \in A | X(s) \in U \text{ for } s \in [0, t]) \quad \forall\, A \subseteq U. \tag{3.2}$$

The QSD ρ can then be sampled as follows: Choose a time $T^{\rho}_{corr}(U)$ for relaxation to ρ. Start $X(t)_{t\geq 0}$ in U, and if it escapes from U before time $t = T^{\rho}_{corr}(U)$, restart it in U. Repeat this until a trajectory of $X(t)_{t\geq 0}$ remains in U for a consecutive time interval of length $T^{\rho}_{corr}(U)$. This trajectory's terminal position is then a sample of ρ. For more details on the QSD, see, for instance, [16]. For conditions ensuring existence of and convergence to the QSD for general Markov processes, see [14, 15, 16].

The following is a more formal definition of metastability: A set U is metastable for $X(t)_{t\geq 0}$ if the time scale to reach ρ is small compared to the mean time to escape from U starting at ρ. In some cases these times can be written in terms of the eigenvalues of the adjoint, L^*, of the generator L of $X(t)_{t\geq 0}$, with absorbing boundary conditions on the complement of U. See [30] and [65] for the corresponding spectral analysis for overdamped Langevin dynamics and finite state space discrete and continuous time Markov chains, and see [8] for an application of these ideas to choosing $T^{\rho}_{corr}(U)$.

4. Parallel replica dynamics.

ParRep can boost the efficiency of simulating metastable processes [3, 5, 30, 62, 63, 65]. Currently, implementations have been proposed only for Langevin or overdamped Langevin dynamics [30] and discrete or continuous time Markov chains [3, 5, 65]. However, the generality of ParRep allows for extensions to any metastable time homogeneous strong Markov process with càdlàg paths, in cases where the QSD exists and metastable sets can be identified. We make this precise in the next section.

ParRep algorithms are based on two basic steps:

- a step in which $X(t)_{t\geq 0}$ reaches the QSD in some metastable set U, using direct or serial simulation—called the *decorrelation step*;
- a step generating an escape event from U, starting from the QSD, using parallel simulation—called the *parallel step*.

By *escape event* we mean the random pair $(T, X(T))$, where T is the time for $X(t)_{t\geq 0}$ to escape from U when $X(0)$ is distributed as the QSD in U, and $X(T)$ is the corresponding

escape point. The parallel step efficiently computes an escape event starting from the QSD via a sort of time parallelization.

The decorrelation step, as it uses only serial simulation, is exact. By *exact* we mean there is *zero error*—except for the inevitable error in simulating $X(t)_{t\geq 0}$ arising from numerical discretizations, which we will ignore. Our analysis will therefore focus on the parallel step.

The parallel step is sometimes divided into two subroutines: first, a routine that generates independent samples of the QSD in U—called *dephasing*—and second, a routine that uses copies of $X(t)_{t\geq 0}$ starting from these QSD samples to generate an escape event of $X(t)_{t\geq 0}$ from U. Below, we will mostly omit discussion of the dephasing routine, and we will not discuss the error associated with imperfect convergence to the QSD in the dephasing and decorrelation steps, as these points have been previously studied in [8, 30, 52, 65].

We say the parallel step of a ParRep algorithm is *consistent* when

- the parallel step generates escape events from each metastable set U with the correct probability law—see Theorem 5.2 below;
- the parallel step produces correct mean contributions to time averages in each metastable set U—see Theorem 5.3 below.

By *correct* we mean exact, provided the QSD sampling has zero error. A consistent ParRep algorithm defines a coarse dynamics, that is, a dynamics that is correct on the quotient space obtained by considering each metastable set as a single point [3, 5, 65]. A consistent ParRep algorithm also defines stationary averages that are correct for functions defined on the original uncoarsened state space [3, 65]. ParRep produces only a coarse dynamics because the parallel step does not resolve the exact behavior of $X(t)_{t\geq 0}$. The parallel step is faithful enough to $X(t)_{t\geq 0}$, however, to produce correct stationary averages on the original uncoarsened state space [3].

Previous analyses of ParRep have relied on the structure of $X(t)_{t\geq 0}$ and the particular algorithms studied [3, 5, 30, 65]. We introduce a new framework below that allows us to study the consistency of any ParRep algorithm. Our analysis is inspired by *ParSplice*, a recent implementation of ParRep employing asynchronous computing [46, 47, 54]. Our framework provides explicit conditions that ensure an asynchronous ParRep algorithm is consistent. In particular, it shows a certain class of asynchronous ParRep algorithms is consistent, provided the wall-clock time to simulate a step of $X(t)_{t\geq 0}$ is not coupled to its position in state space; see section 6 below for precise statements.

5. Trajectory fragments.

We now formalize conditions which lead to consistency of ParRep. Our arguments are based on what we call *trajectory fragments*. The fragments are copies of the underlying process satisfying the dependency conditions of Assumption 5.1 below. In practice, the trajectory fragments may be computed asynchronously in parallel. We discuss this in the next section.

The $X_m(t)_{0\leq t\leq t_m}$ from Assumption 5.1 are the *trajectory fragments*. We will refer to $X_m(t)_{t>t_m}$ as a fragment's *irrelevant future*. The reason for this choice of words is that the output of a general parallel step, described in Algorithm 5.1, is the same no matter how the $X_m(t)_{t\geq 0}$ are defined for times $t > t_m$.

Assumption 5.1. *Let $X(t)_{t\geq 0}$ have càdlàg paths and the strong Markov property. Assume*

$X(t)_{t\geq 0}$ has a QSD ρ in an open set U and that $T = \inf\{t \geq 0 : X(t) \notin U\}$ is finite almost surely. Let $(X_m(t)_{t\geq 0}, T_m)_{m\geq 1}$ be copies of $(X(t)_{t\geq 0}, T)$ such that

$$\text{(5.1)} \qquad \text{conditional on } X_m(0),\ X_m(t)_{t\geq 0} \text{ is independent of } (X_k(t)_{t\geq 0})_{1\leq k<m};$$

$$\text{(5.2)} \qquad \mathcal{L}(X_m(0)|T_{m-1} > t_{m-1}, \dots, T_1 > t_1) = \rho \text{ for } m \geq 2, \quad \mathcal{L}(X_1(0)) = \rho,$$

where $t_m > 0$ are deterministic times satisfying $\sum_{m=1}^{\infty} t_m = \infty$.

Algorithm 5.1 A general parallel step in U.

Let Assumption 5.1 hold and adopt the notation therein.

1. Define $L = \inf\{m \geq 1 : T_m \leq t_m\}$.

2. In the discrete time case, set

$$g_{par} = \mathbb{E}\left(\sum_{m=1}^{L} \sum_{t=0}^{T_m \wedge t_m - 1} g(X_m(t))\right),$$

while in the continuous time case, set

$$g_{par} = \mathbb{E}\left(\sum_{m=1}^{L} \int_0^{T_m \wedge t_m} g(X_m(t))\, dt\right).$$

3. Let $T_{par} = t_1 + \cdots + t_{L-1} + T_L$ and $X_{par} = X_L(T_L)$.

Once g_{par}, T_{par}, and X_{par} can be computed, the parallel step is complete. This parallel step is consistent in the sense of Theorems 5.2 and 5.3 below.

Algorithm 5.1 outlines a general parallel step. As discussed above, this parallel step can be combined with a decorrelation step to compute a coarse dynamics or a time average of a function g. The idea behind Algorithm 5.1 is simple: We imagine concatenating fragments whose starting points and terminal points are distributed as the QSD ρ, thus obtaining an artificial long trajectory. See Figure 1 below. One must be careful, however, in treating dependencies of the fragments. The dependencies described in Assumption 5.1 lead to a consistent Algorithm 5.1 in the sense of Theorems 5.2 and 5.3. More general dependencies can violate consistency, as we will discuss in the next section.

Our next two results demonstrate the consistency of Algorithm 5.1 under the conditions in Assumption 5.1. Theorem 5.2 states that Algorithm 5.1 produces the correct escape law from U starting at the QSD in U, while Theorem 5.3 says that Algorithm 5.1 produces the correct mean contribution to time averages.

Theorem 5.2. *Suppose that Assumption 5.1 holds. Let $X(t)_{t\geq 0}$ be such that $\mathcal{L}(X(0)) = \rho$, and set $T = \inf\{t > 0 : X(t) \notin U\}$. Then in Algorithm 5.1,*

$$(T_{par}, X_{par}) \overset{\mathcal{L}}{=} (T, X(T)).$$

Theorem 5.3. *Suppose that Assumption 5.1 holds. Let $X(t)_{t\geq 0}$ be such that $\mathcal{L}(X(0)) = \rho$, and set $T = \inf\{t > 0 : X(t) \notin U\}$. Then in Algorithm 5.1, in the discrete time case,*

$$g_{par} := \mathbb{E}\left(\sum_{m=1}^{L}\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\right) = \mathbb{E}\left(\sum_{t=0}^{T-1} g(X(t))\right),$$

while in the continuous time case,

$$g_{par} := \mathbb{E}\left(\sum_{m=1}^{L}\int_0^{T_m\wedge t_m} g(X_m(t))\,dt\right) = \mathbb{E}\left(\int_0^T g(X(t))\,dt\right).$$

Recall that the gain in ParRep is from parallel computations in the parallel step. In our trajectory fragment framework, the basic idea is that the work to compute the fragments $X_m(t)_{0\leq t\leq t_m}$ can be spread over multiple processors. See [3, 5, 30, 65] for related results in special cases. The parallel step is more efficient than direct, or serial, simulation, provided the computational effort to sample the QSD is small relative to the effort to simulate an escape from U via serial simulation.

We actually do not need to assume that U is open and that $X(t)_{t\geq 0}$ has the strong Markov property and càdlàg paths to prove consistency of the parallel step in Algorithm 5.1. Indeed, Algorithm 5.1 is consistent in the sense of Theorems 5.2 and 5.3 whenever $X(t)_{t\geq 0}$ is a time homogeneous Markov process and (5.1)–(5.2) hold. However, to combine the parallel step with a decorrelation step to obtain a coarse dynamics or stationary average, we want càdlàg paths to ensure that the escape time from an open set U is a stopping time, and we need the strong Markov property so that we can start afresh at these stopping times.

6. Synchronous and asynchronous computing.

Recall that the speedup in ParRep comes from computing the trajectory fragments $X_m(t)_{0\leq t\leq t_m}$ partly or fully in parallel. These fragments must be *ordered*, via the index $m \geq 1$, to obtain the long trajectory pictured in Figure 1. Below, we explore two possible ways to order the fragments, depending on whether we want to employ *synchronous* or *asynchronous computing*. In the former case, we have in mind a computing environment consisting of R processors that are nearly synchronous. In the latter case we consider an arbitrary number of processors that potentially have widely different performance.

Below, we will consider only fragments of constant time length, $t_m \equiv \Delta t$. For synchronous computing, following ideas from [3, 5, 65], we consider the ordering of trajectory fragments in Proposition 6.1 below.

Proposition 6.1 (synchronous computing). *Suppose $Y^r(t)_{t\geq 0}$, $r = 1, \ldots, R$, are independent copies of $X(t)_{t\geq 0}$ with $\mathcal{L}(X(0)) = \rho$. Let $m_k = \lfloor (k-1)/R \rfloor$ and $r_k = k - R\lfloor (k-1)/R \rfloor$, and for $k \geq 1$ define trajectory fragments $X_k(t)_{0\leq t\leq \Delta t}$ by*

$$X_k(t) = Y^{r_k}(m_k\Delta t + t), \qquad 0 \leq t \leq \Delta t. \tag{6.1}$$

Then Assumption 5.1 holds with an appropriate definition of the trajectory fragments' irrelevant futures. Thus the conclusions of Theorem 5.2 and 5.3 hold.

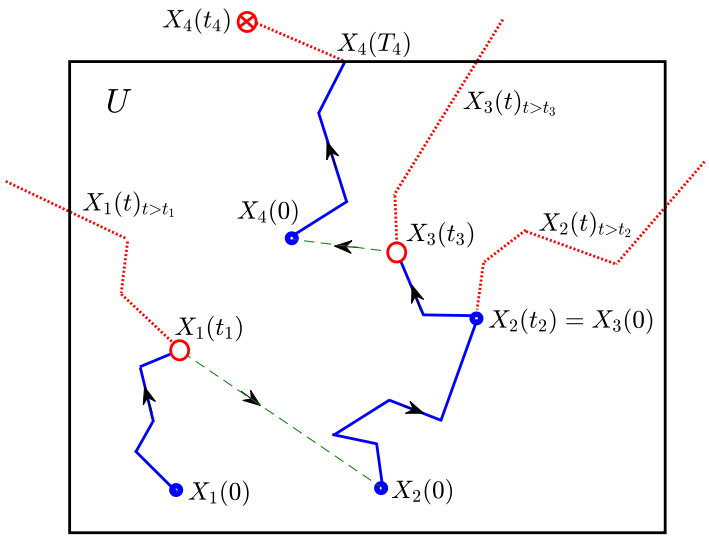

Figure 1. *Intuition behind the general parallel step of Algorithm 5.1. Pictured are $L = 4$ trajectory fragments $(X_m(t)_{0\le t\le t_m})_{m=1,2,3,4}$ combined to form one long trajectory, advancing in time in the direction indicated by the arrows. The terminal point of the first fragment that escapes from U is denoted by a cross. The dashed line parts of the long trajectory are artificial and do not contribute to g_{par} or T_{par}. Only the solid lines and solid dots along the long trajectory contribute to g_{par} and T_{par}. The dotted lines show the fragments' irrelevant futures. Note that one of the fragments' starting point, $X_3(0)$, is equal to another fragment's terminal point, $X_2(t_2)$. There can be other fragments, but they are not relevant to the parallel step in this example since $T_4 < t_4$.*

Figure 2 shows the trajectory fragments defined in (6.1).

In asynchronous computing, perhaps the most natural ordering is the *wall-clock ordering*: the fragments are ordered according to the wall-clock time that their starting points are computed. See Figure 3. When does the wall-clock ordering satisfy Assumption 5.1? Note that (5.1) simply says that each fragment evolves forward in time independently of the preceding fragments and their irrelevant futures. This condition is easy to establish with an appropriate choice of the irrelevant futures. Ensuring (5.2) holds is more subtle. We will show, however, that if the wall-clock time it takes to compute each fragment depends on processor variables, but not on the fragments themselves, then the wall-clock time ordering is independent of the fragments, and (5.2) holds.

We will distinguish between a wall-clock time and a physical time, where the former is self-explanatory and the latter refers to the time index t of a copy of $X(t)_{t\ge 0}$. Let $Y^r(t)_{t\ge 0}$ be independent copies of $X(t)_{t\ge 0}$ starting at independent samples of the QSD ρ. The wall-clock time ordering of fragments satisfies Assumption 5.1 above if (i) the wall-clock times are independent of the physical times, (ii) the wall-clock time to compute copy $Y^r(t)_{t\ge 0}$ is an increasing function of the physical time, and (iii) two processors never finish at exactly the same wall-clock time, so that the wall-clock times can be given a unique ordering. Write $t^r_{wall}(m)$ for the wall-clock time it takes to compute $Y^r(t)_{t\ge 0}$ up to physical time $t = m\Delta t$. Proposition 6.2 below makes the claims above precise.

Proposition 6.2 (asynchronous computing). *Suppose $Y^r(t)_{t\ge 0}$, $r = 1, \dots, R$, are indepen-*

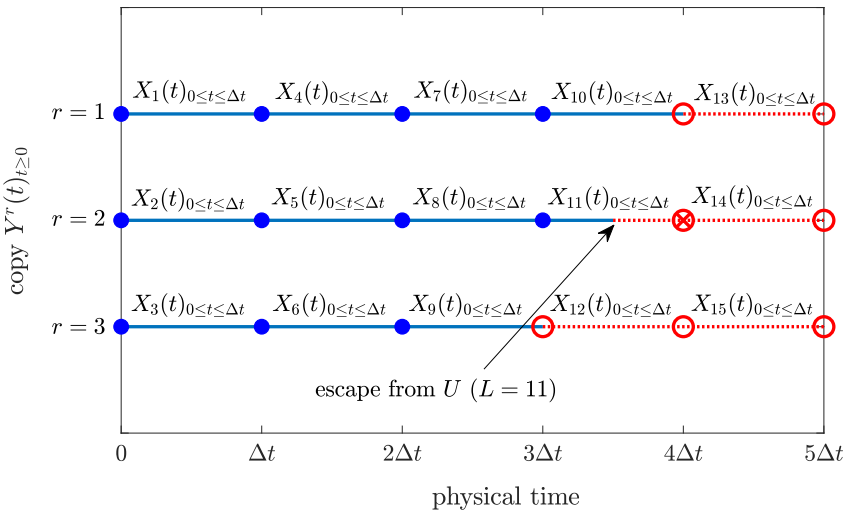

Figure 2. *An example of a synchronous ParRep algorithm based on the ordering of fragments in Proposition 6.1. Solid dots and hollow circles correspond to fragments' initial and terminal points. The cross corresponds to the terminal point of the first fragment, in terms of the ordering, to escape from U. Times of copies that contribute to g_{par} and T_{par} are pictured with solid line segments and solid dots, while times that do not contribute are pictured with dotted lines and hollow circles.*

dent copies of $X(t)_{t\ge0}$ with $\mathcal{L}(X(0))=\rho$. Assume $(t^r_{wall}(m)_{m\ge0})^{1\le r\le R}$ are nonnegative random numbers such that

(i) *$(t^r_{wall}(m)_{m\ge0})^{1\le r\le R}$ is independent of $(Y^r(t)_{t\ge0})^{1\le r\le R}$;*
(ii) *almost surely, $t^r_{wall}(m)\le t^r_{wall}(n)$ when $m\le n$ and $1\le r\le R$;*
(iii) *almost surely, there is a unique sequence $(r_k,m_k)_{k\ge1}$ such that*

$$(r_k,m_k)_{k\ge1} \text{ has range } \{1,\dots,R\}\times\{0,1,2,\dots\} \qquad \textit{(surjectivity)},$$
$$t^{r_1}_{wall}(m_1)<t^{r_2}_{wall}(m_2)<t^{r_3}_{wall}(m_3)<\cdots \qquad \textit{(monotonicity)}.$$

For $k\ge1$ define trajectory fragments $X_k(t)_{0\le t\le\Delta t}$ by

$$X_k(t)=Y^{r_k}(m_k\Delta t+t),\qquad 0\le t\le\Delta t. \tag{6.2}$$

Then Assumption 5.1 holds with an appropriate definition of the trajectory fragments' irrelevant futures. Thus the conclusions of Theorems 5.2 and 5.3 hold.

Figure 3 shows the trajectory fragments $X_m(t)_{0\le t\le t_m}$ defined in (6.2).

Assumptions (ii) and (iii) are quite natural, but assumption (i) can fail in many very ordinary settings. We sketch an example explaining how this could happen. Suppose $U=(0,1)\subseteq\mathbb{R}$ and say $X(t)_{t\ge0}$ obeys some one-dimensional stochastic differential equation. Suppose we use an integrator for $X(t)_{t\ge0}$ that is slow near 0 but fast near 1. Then the wall-clock ordering will likely put trajectory fragments that are near 1 ahead of those near 0. This bias in the ordering would in turn create a bias toward escaping through 1: in Algorithm 5.1, we would

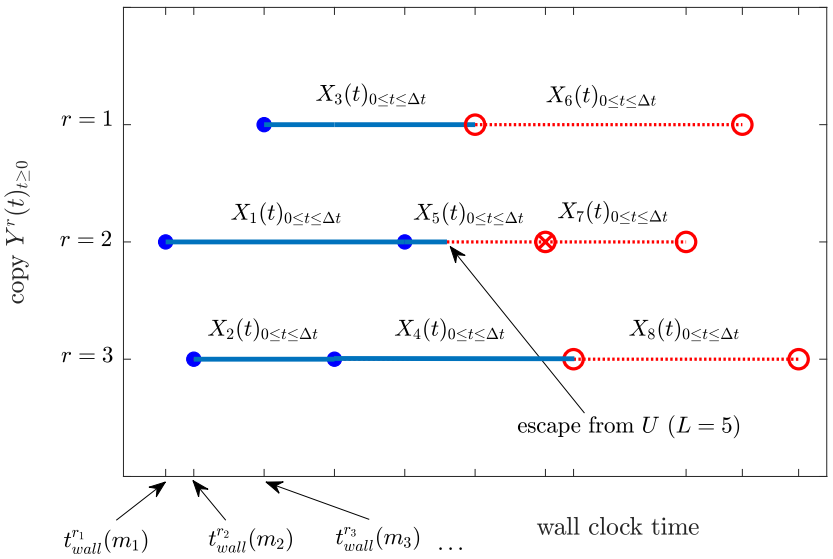

Figure 3. *An example of an asynchronous ParRep algorithm based on wall-clock time ordering of fragments in Proposition 6.2. Solid dots and hollow circles correspond to fragments' initial and terminal points. The cross corresponds to the terminal point of the first fragment, in terms of the ordering, to escape from U. Times of copies that contribute to g_{par} and T_{par} are pictured with solid line segments and solid dots, while times that do not contribute are pictured with dotted lines and hollow circles.*

expect that $\mathbb{P}(X_{par}=1) > \mathbb{P}(X_T=1)$, where X_T is the correct escape point. We construct a specific example demonstrating this bias in Remark 10.5 in section 10.1 below.

The speed of integrators does commonly depend on position in space, particularly when the time step varies to account for numerical stiffness [55]. This is an important caveat to keep in mind for asynchronous algorithms. This issue has not been explored much in the literature; see, however, the brief discussions in [30] and [46].

The setting of ParSplice [46, 47, 54] is slightly different from the above. In ParSplice, a *splicer* tells a *producer* to generate fragments among several metastable sets. The splicer distributes the fragments according to where it speculates they will be needed. These fragments are given a label as soon as they are assigned, and this label never changes. The labels are assigned in wall-clock time order. Thus label i is less than label j if and only if the splicer tells the producer to generate fragment i before it tells the producer to generate fragment j. When the splicer tells the producer to generate a fragment in a particular metastable set U, it takes as its starting point the terminal point of the fragment in U with the smallest label. Crucially, this label is smallest among *all* fragments in U and not just among fragments in U which have been fully computed at the current wall-clock time. Thus, the ordering of fragments in U, fixed by the splicer, can be seen as independent of the fragments themselves, and the arguments above demonstrate the consistency of ParSplice.

7. PDMPs.

The remainder of this article will focus on applying our ideas above to PDMPs. We begin with a brief informal description of PDMPs. A PDMP is a càdlàg process

consisting of a deterministic dynamics interrupted by jumps at random times; formally, a PDMP in $\mathbb{R}^d$ has a generator L of the form

$$Lf(z) = \partial_{\Gamma(z)} f(z) + \lambda(z) \int (f(z') - f(z)) Q(z, dz') \tag{7.1}$$

acting on suitable $f : \mathbb{R}^d \to \mathbb{R}$. Here $\lambda(z)$ is the jump rate at $z \in \mathbb{R}^d$, a Markov kernel $Q(z, dz')$ describes the jump distribution, and Γ defines the deterministic flow

$$\partial_t \psi(t, z) = \Gamma(\psi(t, z)), \qquad \psi(0, z) = z.$$

Write $\theta_0 + \cdots + \theta_{n-1}$ for the nth jump time, so that θ_{n-1} is the holding time before the nth jump, and write ξ_n for the position immediately after the nth jump, with ξ_0 the initial position. Then the PDMP generated by (7.1) is described by ψ together with $(\xi_n, \theta_n)_{n \geq 0}$; we call the latter the *skeleton chain* of $Z(t)_{t \geq 0}$. Note that the skeleton chain is a time homogeneous Markov chain.

For convenience we describe a way to simulate a PDMP described by (7.1) in Algorithm 7.1 below. In the algorithm, we abuse notation by writing θ_n, ξ_n, and $Z(t)$ for particular realizations of these random objects.

Algorithm 7.1 Simulating a PDMP.

Starting from an initial point ξ_0 and time $t = 0$, set $n = 0$, and iterate:

1. Sample θ_n according to the distribution

$$\mathbb{P}(\theta_n > r) = \exp\left(- \int_0^r \lambda(\psi(s, \xi_n))\, ds \right). \tag{7.2}$$

2. Set $Z(t + s) = \psi(s, \xi_n)$ for $s \in [0, \theta_n)$, and sample ξ_{n+1} from $Q(Z(t + \theta_n^-), dz)$.

3. Update $t \leftarrow t + \theta_n$ and then $n \leftarrow n + 1$. Then return to step 1.

Steps 1–3 above define a realization of $Z(t)_{t \geq 0}$ with skeleton chain $(\xi_n, \theta_n)_{n \geq 0}$.

Sampling the times θ_n is a nontrivial task, but there are efficient methods based on Poisson thinning [6, 59] and identifying certain critical points along the flow direction [26]. See also [59] for other methods to simulate a PDMP, including some based on time discretization. We will always assume our initial points (ξ_0, θ_0) are chosen so that θ_0 satisfies (7.2) for $n = 0$, so that we can skip the first step in Algorithm 7.1.

7.1. Example: Linear flow. Consider a PDMP with deterministic paths that are lines in $\mathbb{R}^{d-1}$ corresponding to a finite collection of velocity vectors $d_i \in \mathbb{R}^{d-1}$, $i \in \mathcal{I} \subseteq \mathbb{N}$. Its generator L is defined on suitable functions $f : \mathbb{R}^{d-1} \times \mathcal{I} \to \mathbb{R}$ by

$$Lf(x, i) = d_i \cdot \nabla f(x, i) + \sum_{j \neq i} \lambda_j(x, i)(f(x, j) - f(x, i)), \tag{7.3}$$

where $\lambda_j(x, i) \geq 0$ for $j \neq i$. Suppose we want to sample the probability density

$$Z^{-1} e^{-V(x)}, \qquad Z = \int e^{-V(x)}\, dx, \tag{7.4}$$

where $V : \mathbb{R}^{d-1} \to \mathbb{R}$ is smooth and grows sufficiently fast at ∞ so that $Z < \infty$. For the PDMP generated by (7.3) to have an invariant probability density independent of i and proportional to (7.4), the jump rates must satisfy

$$\sum_{j \neq i} (\lambda_i(x, j) - \lambda_j(x, i)) = -d_i \cdot \nabla V(x). \tag{7.5}$$

See Remark 10.6 in section 10.1 below, and [6] for a similar calculation.

Event Chain Monte Carlo and the zig-zag process fit into this framework, and while these methods were designed for efficient sampling, we argue that they may be limited in certain situations. To see why, note that (7.5) says that at x, the rate into d_i minus the rate out of d_i equals minus the gradient of V in direction d_i. Thus the PDMP is likely to change directions when it moves up a steep slope of V. This suggests the PDMP can struggle to escape from a basin of attraction of V, defined as the set of initial conditions $x(0)$ for which $dx(t)/dt = -\nabla V(x(t))$ has a unique long-time limit.

8. ParRep for PDMPs.

In this section $Z(t)_{t\geq 0}$ is a PDMP with stationary distribution π, and f is a real-valued function defined on the state space of $Z(t)_{t\geq 0}$. Below we outline some ParRep algorithms for estimating coarse dynamics as well as stationary averages, with a focus on the latter. The stationary average of f is

$$\langle f \rangle = \int f(x)\pi(dx). \tag{8.1}$$

Algorithms 8.3 and 8.6 below are ParRep algorithms based on the skeleton chain and the continuous time PDMP, respectively. Algorithms 8.1 and 8.4 are parallel steps for synchronous computing, while Algorithms 8.2 and 8.5 are for asynchronous computing. Algorithms 8.1 and 8.4, which are essentially extensions to PDMPs of algorithms recently proposed for continuous time Markov chains [64, 65], use the ordering of trajectory fragments defined in Proposition 6.1. Algorithms 8.2 and 8.5 employ the wall-clock time ordering of fragments from Proposition 6.2. We prove consistency of all of our parallel steps via our trajectory fragment framework.

We do not attempt to prove existence, uniqueness, or convergence to the QSD for general PDMPs. Instead we refer the reader to recent articles [14, 15] for conditions which ensure convergence to a unique QSD. From those works, under appropriate assumptions, one can establish convergence to a unique QSD in $D \times \mathcal{I} \subseteq \mathbb{R}^d$ for a PDMP generated by (7.3). For instance, exponential convergence is guaranteed if $D \subseteq \mathbb{R}^{d-1}$ is an open connected bounded domain and there exist m, M so that $0 < m \leq \lambda_j(x, i) \leq M$ for all $x \in D$ and $i, j \in \mathcal{I}$; see [14, p. 261]. Similar arguments can be made for the QSD of the skeleton chain. Even without theoretical guarantees, in practice, one can empirically validate convergence to the QSD using certain diagnostics; see, for instance, [8].

8.1. Skeleton chain-based ParRep algorithm.

Let $\mathcal{W}$ be the collection of metastable sets for the skeleton chain $(\xi_n, \theta_n)_{n\geq 0}$. For instance, if $Z(t)_{t\geq 0}$ has generator similar to the form (7.3) and we want to sample from the distribution (7.4), it is natural to define $\mathcal{W}$ in terms of basins of attraction of V, in which case elements of $\mathcal{W}$ may be identified on the fly by gradient descent [62, 63]. See section 9 for an example of metastable sets defined this way.

Assumption 8.1. *$(\xi_n, \theta_n)_{n\geq 0}$ has a QSD $\nu = \nu_W$ in each $W \in \mathcal{W}$ satisfying*

$$\nu(A) = \lim_{n\to\infty} \mathbb{P}((\xi_n, \theta_n) \in A | (\xi_m, \theta_m) \in W,\, 0 \leq m \leq n) \quad \forall\, A \subseteq W.$$

For simpler notation, we do not explicitly indicate the dependence of ν on W.

Algorithm 8.1 Synchronous skeleton chain parallel step in W.

1. Generate i.i.d. samples $(\xi_0^r, \theta_0^r)^{r=1,\dots,R}$ from the QSD ν in W. Using these as starting points, independently evolve R copies $((\xi_n^r, \theta_n^r)_{n\geq 0})^{r=1,\dots,R}$ of the skeleton chain.
2. Let $N = \inf\{n : \exists\, r \text{ s.t. } (\xi_n^r, \theta_n^r) \notin W\}$, $J = \min\{r : (\xi_N^r, \theta_N^r) \notin W\}$, and define

$$f_{par} = \sum_{n=0}^{N-2} \sum_{r=1}^{R} \int_0^{\theta_n^r} f(\psi(t, \xi_n^r))\, dt + \sum_{r=1}^{J} \int_0^{\theta_{N-1}^r} f(\psi(t, \xi_{N-1}^r))\, dt$$

and $T_{par} = \mathbb{1}_{par}$ by using the same formula but with $\mathbb{1}(z) \equiv 1$ in place of f. Set

$$(\xi_{par}, \theta_{par}) = (\xi_N^J, \theta_N^J).$$

Once f_{par}, T_{par}, and $(\xi_{par}, \theta_{par})$ can be computed, the parallel step is complete.

Algorithm 8.2 Asynchronous skeleton chain parallel step in W.

1. Generate i.i.d. samples $(\xi_0^r, \theta_0^r)^{r=1,\dots,R}$ from the QSD ν in W. Using these as starting points, independently evolve R copies $((\xi_n^r, \theta_n^r)_{n\geq 0})^{r=1,\dots,R}$ of the skeleton chain.
2. Reorder these skeleton chain points in the order they are computed in wall-clock time, i.e., as $(\xi_{m_k}^{r_k}, \theta_{m_k}^{r_k})_{k\geq 1}$, where $t_{wall}^{r_1}(m_1) \leq t_{wall}^{r_2}(m_2) \leq \cdots$ and $t_{wall}^r(n)$ is the wall-clock time it takes to compute the skeleton chain $(\xi_m^r, \theta_m^r)_{m\geq 0}$ up to physical time $m = n$. Set $\sigma^r = \inf\{m : (\xi_m^r, \theta_m^r) \notin W\}$, $K = \inf\{k : \sigma^{r_k} \leq m_k + 1\}$, and

$$f_{par} = \sum_{k=1}^{K} \int_0^{\theta_{m_k}^{r_k}} f(\psi(t, \xi_{m_k}^{r_k}))\, dt.$$

Define $T_{par} = \mathbb{1}_{par}$ by using the same formula but with $\mathbb{1}(z) \equiv 1$ in place of f. Let

$$(\xi_{par}, \theta_{par}) = (\xi_{\sigma^{r_K}}^{r_K}, \theta_{\sigma^{r_K}}^{r_K}).$$

Once f_{par}, T_{par}, and $(\xi_{par}, \theta_{par})$ can be computed, the parallel step is complete.

Algorithms 8.1 and 8.2 are parallel steps designed for synchronous and asynchronous computing, respectively. See Figure 4 for a diagram of both parallel steps. The first step in both Algorithm 8.1 and Algorithm 8.2—called *dephasing* in the literature [3, 5, 62, 65]—involves generating R independent samples from the QSD ν in W. These QSD samples may be obtained in a variety of ways. One option is to do rejection sampling using independent copies of the skeleton chain: whenever a copy escapes from W, start it afresh in W until each

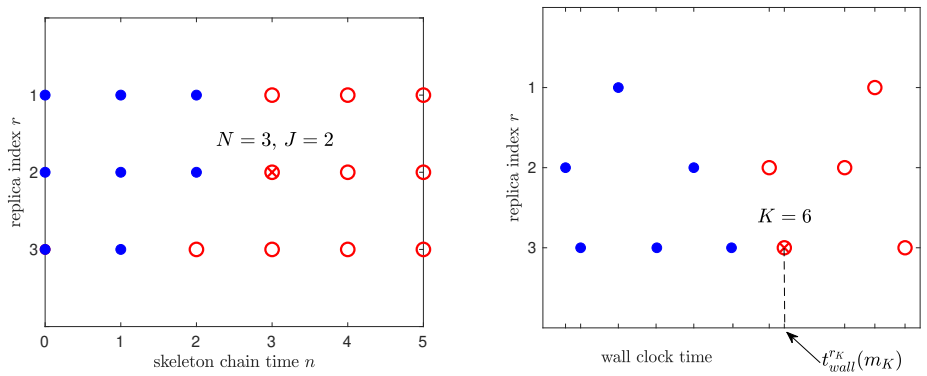

Figure 4. *Illustration of the parallel steps used in Algorithm 8.3. The number of copies, or parallel replicas, is $R = 3$. The cross indicates an escape from W. Time steps of copies of the skeleton chain that contribute to f_{par} and T_{par} are pictured with solid dots, while time steps that do not contribute are pictured with hollow circles. Left: The synchronous parallel step, Algorithm 8.1. Copy $r = 2$ escapes from W at skeleton chain time 3. In this example, $N = 3$ and $J = 2$. Right: The asynchronous parallel step, Algorithm 8.2. Copy $r = 3$ is the first to escape from W in terms of the wall-clock time ordering. It escapes at wall-clock time $t^{r_K}_{wall}(m_K)$. In this example, $K = 6$.*

copy has remained in W for a long enough consecutive time. Another possibility is based on the Fleming–Viot branching process [8, 24]: when a copy escapes from W, restart it at the current position of a copy still in W chosen at random. For more discussion see [8, 52, 62].

Recall that the speedup from ParRep comes from the parallel step. The speedup—the factor by which ParRep reduces the wall-clock computation time, compared to serial simulation of a trajectory of the same physical time—can be a factor of up to R, the number of copies or replicas [3, 5, 30, 62, 65], when Algorithm 7.1 is used to simulate the skeleton chain. See Figure 7. The parallel step is consistent no matter the choice of W, but if W is not metastable, there may be no gain in efficiency, as too much computation time will be spent sampling the QSD.

Theorem 8.2 gives conditions that establish consistency of Algorithm 8.1 and 8.2. For Algorithm 8.2, the crucial condition is that the wall-clock times it takes for the processors to compute steps of the skeleton chains are independent of those chains. Whether this holds true will depend on the algorithm used to simulate the PDMP. If it is a time discretization–based algorithm, or an implementation of Algorithm 7.1 based on Poisson thinning, then the computational effort to obtain one step of the skeleton chain can be larger in regions in state space with lower jump rates. For CTMCs simulated via the SSA/Gillespie algorithm [2], the effort to simulate one step of the skeleton chain may be essentially independent of the position of the chain.

Theorem 8.2 (consistency of the parallel steps, Algorithm 8.1 and 8.2).
(i) *Let ν be the QSD of $(\xi_n, \theta_n)_{n\geq 0}$ in some $W \in \mathcal{W}$, suppose that $\mathcal{L}(\xi_0, \theta_0) = \nu$, and define $M = \inf\{n \geq 0 : (\xi_n, \theta_n) \notin W\}$. Then in Algorithm 8.1,*

$$\mathbb{E}(f_{par}) = \mathbb{E}\left(\sum_{n=0}^{M-1} \int_0^{\theta_n} f(\psi(t, \xi_n))\, dt\right) \tag{8.2}$$

and

$$(\xi_{par}, \theta_{par}) \stackrel{\mathcal{L}}{=} (\xi_M, \theta_M). \tag{8.3}$$

(ii) *Suppose* $(t_{wall}(m)_{m\geq 0})^{1\leq r\leq R}$*, the wall-clock times from Algorithm* 8.2*, satisfy the assumptions of Proposition* 6.2 *when* $(Y^r(t)_{t\geq 0})^{r=1,\ldots,R}$ *equals* $((\xi_n^r, \theta_n^r)_{n\geq 0})^{r=1,\ldots,R}$*. Adopt the assumptions in* (i) *above. Then* (8.2)–(8.3) *hold.*

The times $T_{corr}^{\nu}(W)$ in Algorithm 8.3 may be chosen on the fly, or they may be set at the beginning of simulations. Choosing an appropriate value may be done using various convergence diagnostics or a priori information; see [8, 52, 62] for details.

Algorithm 8.3 Skeleton chain computation of stationary averages.

Choose an initial point (ξ_0, θ_0), set $f_{sim} = 0$, $T_{sim} = 0$, and iterate:

1. Starting at (ξ_0, θ_0), evolve $(\xi_n, \theta_n)_{n\geq 0}$ forward in time, stopping at time

$$L = \inf\{n \geq T_{corr}^{\nu}(W) - 1 : \exists\, W \in \mathcal{W} \text{ s.t. } (\xi_{n-k}, \theta_{n-k}) \in W,\ k = 0, \ldots, T_{corr}^{\nu}(W) - 1\},$$

the first time it remains in some $W \in \mathcal{W}$ for $T_{corr}^{\nu}(W)$ consecutive time steps. Set

$$f_{decorr} = \sum_{n=0}^{L-1} \int_0^{\theta_n} f(\psi(t, \xi_n))\, dt$$

and $T_{decorr} = \mathbb{1}_{decorr}$ using the same formula. Store this W for step 2 and update

$$f_{sim} \leftarrow f_{sim} + f_{decorr}, \qquad T_{sim} \leftarrow T_{sim} + T_{decorr}.$$

2. Run the parallel step (Algorithm 8.1 or 8.2) in the set W from step 1. Update $f_{sim} \leftarrow f_{sim} + f_{par}$, $T_{sim} \leftarrow T_{sim} + T_{par}$, $(\xi_0, \theta_0) = (\xi_{par}, \theta_{par})$, and return to step 1.

The algorithm stops when T_{sim} exceeds a user-chosen threshold T_{stop}. At this time,

$$\langle f \rangle \approx \frac{f_{sim}}{T_{sim}}$$

is our estimate of the stationary average (8.1).

Consistency of the parallel steps, together with exactness of the decorrelation step, shows that Algorithm 8.3 produces correct stationary averages, provided some mild recurrence assumptions hold [3]. The reason is essentially the law of large numbers: for computations of stationary averages, due to repeated visits to each metastable set, in the parallel steps it is enough to get contributions f_{par} to f_{sim} with the correct *average* value along with escape events with the correct law.

We do not attempt here to prove ergodicity using this argument, but mention it has been studied previously in [3, 65]. Our numerical simulations in section 9 below also support its validity. One interesting aspect of the parallel step is that the averaging over independent copies or replicas can be considered a bonus, as it likely lowers the variance of the estimate

Algorithm 8.4 Synchronous continuous time parallel step in W.

1. Generate i.i.d. samples $Z^r(0)^{r=1,\dots,R}$ from the QSD μ in W. Using these as starting points, independently evolve R copies $(Z^r(t)_{t\ge 0})^{r=1,\dots,R}$ of the PDMP.
2. Let $\tau^r = \inf\{t : Z^r(t) \notin W\}$, set

$$N = \inf\{n \in \mathbb{N} : \exists\, r\ s.t.\ \tau^r \le n\Delta t\}, \qquad J = \min\{r : \tau^r \le N\Delta t\},$$

and define

$$f_{par} = \sum_{n=1}^{N-1}\sum_{r=1}^{R}\int_{(n-1)\Delta t}^{n\Delta t} f(Z^r(t))\,dt + \sum_{r=1}^{J-1}\int_{(N-1)\Delta t}^{N\Delta t} f(Z^r(t))\,dt + \int_{(N-1)\Delta t}^{\tau^J} f(Z^J(t))\,dt,$$

and $T_{par} = \mathbb{1}_{par}$ by using the same formula but with $\mathbb{1}(z) \equiv 1$ in place of f. Set

$$Z_{par} = Z^J(\tau^J).$$

Once f_{par}, T_{par}, and Z_{par} can be computed, the parallel step is complete.

$f_{sim}/T_{sim} \approx \langle f\rangle$ of the stationary average, compared to an estimate from a serial trajectory of physical time length T_{sim}.

8.2. Continuous time PDMP-based algorithm. Let $\mathcal{V}$ be the collection of metastable sets for $Z(t)_{t\ge 0}$. As above, if $Z(t)_{t\ge 0}$ has a generator similar to (7.3) and we want to sample from the distribution (7.4), the elements of $\mathcal{V}$ can be defined in terms of the basins of attraction of V. We will require a time interval $\Delta t > 0$, which is not necessarily a time step for discretizing the PDMP. For instance, Δt could be a polling time for resynchronizing parallel processors.

We will adopt the following assumption.

Assumption 8.3. *$Z(t)_{t\ge 0}$ has a QSD $\mu = \mu_W$ in each $W \in \mathcal{V}$ satisfying*

$$\mu(A) = \lim_{t\to\infty} \mathbb{P}(Z(t) \in A | Z(s) \in W,\ 0 \le s \le t) \quad \forall\, A \subseteq W.$$

We do not explicitly indicate the dependence of μ on W. Notice that the QSD of the PDMP is different from that of its skeleton chain in general.

Algorithms 8.4 and 8.5 are parallel steps designed for synchronous and asynchronous computing, respectively; see Figure 5. The first step in both Algorithm 8.4 and Algorithm 8.5—the dephasing step—involves generating R independent samples from the QSD μ in W. Note that this is the QSD of the PDMP in W, not the QSD of its skeleton chain. The QSD samples can be obtained exactly as described in the previous section, but with the PDMP taking the place of the skeleton chain.

The speedup from the parallel step can be up to a factor of R, the number of copies or replicas [5, 3, 30, 62, 65], provided the underlying PDMP simulation algorithm is based on

Algorithm 8.5 Asynchronous continuous time parallel step in W.

1. Generate i.i.d. samples $Z^r(0)^{r=1,\dots,R}$ from the QSD μ in W. Using these as starting points, independently evolve R copies $(Z^r(t)_{t\geq 0})^{r=1,\dots,R}$ of the PDMP.

2. Reorder the Δt time intervals of these copies in the order they are computed in wall-clock time, i.e., as $Z^{r_k}(m_k\Delta t)_{k\geq 1}$, where $t^{r_1}_{wall}(m_1) \leq t^{r_2}_{wall}(m_2) \leq \cdots$ and $t^r_{wall}(n)$ is the wall-clock time it takes to compute the PDMP $Z^r(t)_{t\geq 0}$ up to physical time $t = n\Delta t$. Set $\tau^r = \inf\{t : Z^r(t) \notin W\}$, $K = \inf\{k : \tau^{r_k} \leq (m_k+1)\Delta t\}$, and

$$f_{par} = \sum_{k=1}^{K-1} \int_{m_k\Delta t}^{(m_k+1)\Delta t} f(Z^{r_k}(t))\,dt + \int_{m_K\Delta t}^{\tau^{r_K}} f(Z^{r_K}(t))\,dt,$$

and $T_{par} = \mathbb{1}_{par}$ by using the same formula but with $\mathbb{1}(z) \equiv 1$ in place of f. Let

$$Z_{par} = Z^{r_K}(\tau^{r_K}).$$

Once f_{par}, T_{par}, and ξ_{par} can be computed, the parallel step is complete.

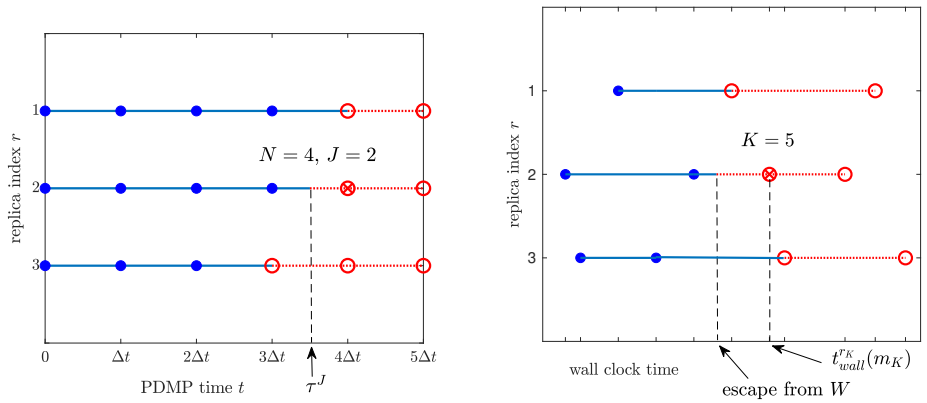

Figure 5. *Illustration of the parallel steps used in Algorithm 8.6. The number of copies, or parallel replicas, is $R = 3$. Solid dots and hollow circles correspond to trajectories at PDMP times $n\Delta t$. The cross indicates the terminal point of a Δt-time interval corresponding to an escape from W. Times of copies that contribute to g_{par} and T_{par} are pictured with solid line segments and solid dots, while times that do not contribute are pictured with dotted lines and hollow circles. Left: The synchronous parallel step, Algorithm 8.4. Copy $r = 2$ escapes at PDMP time τ^2. In this example, $N = 4$ and $J = 2$. Right: The asynchronous parallel step, Algorithm 8.5. Copy $r = 2$ is the first to escape in terms of the wall-clock time ordering. The notches on the wall-clock time axis are the values of $t^{r_k}_{wall}(m_k)$, $k = 1, \dots, 11$. In this example $K = 5$.*

time discretization. If the PDMP simulation algorithm is based on computing the skeleton chain, then the speedup in Algorithm 8.4 may be reduced. This can be mitigated, however, by using Algorithm 8.5 instead. The parallel step is consistent for any W, with a speedup if W is metastable for the PDMP.

Theorem 8.4 gives conditions that establish consistency of the parallel steps, Algorithm 8.4 and 8.5. For the asynchronous parallel step, the crucial condition essentially says that the

wall-clock time it takes to compute a Δt time interval of $Z(t)_{t\geq 0}$ is independent of its position. This is reasonable if $Z(t)_{t\geq 0}$ is simulated via a time discretization technique with a fixed time step. It may not be reasonable if a skeleton chain-based technique, like Algorithm 7.1, is used instead.

Theorem 8.4 (consistency of the parallel steps, Algorithm 8.4 and 8.5).

(i) *Let μ be the QSD of $Z(t)_{t\geq 0}$ in some $W \in \mathcal{V}$, suppose that $\mathcal{L}(Z(0)) = \mu$, and define $\tau = \inf\{t \geq 0 : Z(t) \notin W\}$. Then in Algorithm 8.4,*

$$\mathbb{E}(f_{par}) = \mathbb{E}\left(\int_0^\tau f(Z(t))\, dt\right) \tag{8.4}$$

and

$$(T_{par}, Z_{par}) \stackrel{\mathcal{L}}{=} (\tau, Z(\tau)). \tag{8.5}$$

(ii) *Suppose $(t_{wall}(m)_{m\geq 0})^{1\leq r\leq R}$, the wall-clock times from Algorithm 8.5, satisfy the assumptions of Proposition 6.2 when $(Y^r(t)_{t\geq 0})^{r=1,\dots,R}$ equals $(Z^r(t)_{t\geq 0})^{r=1,\dots,R}$. Adopt the assumptions in* (i) *above. Then* (8.4)–(8.5) *hold.*

Algorithm 8.6 Continuous time computation of stationary averages.

Choose an initial point $Z(0)$, set $f_{sim} = 0$, $T_{sim} = 0$, and iterate:

1. Starting at $Z(0)$, evolve $Z(t)_{t\geq 0}$ forward in time, stopping at time

$$S = \inf\{t \geq T^\mu_{corr}(W) : \exists W \in \mathcal{V} \text{ s.t. } Z(s) \in W,\ s \in [t - T^\mu_{corr}(W), t]\},$$

the first time it remains in some $W \in \mathcal{V}$ for consecutive time $T^\mu_{corr}(W)$. Set

$$f_{decorr} = \int_0^S f(Z(t))\, dt$$

and $T_{decorr} = \mathbb{1}_{decorr}$ using the same formula. Store this W for step 2 and update

$$f_{sim} \leftarrow f_{sim} + f_{decorr}, \qquad T_{sim} \leftarrow T_{sim} + T_{decorr}.$$

2. Run the parallel step (Algorithm 8.4 or 8.5) in the set W from step 1. Update $f_{sim} \leftarrow f_{sim} + f_{par}$, $T_{sim} \leftarrow T_{sim} + T_{par}$, set $Z(0) = Z_{par}$, and then go to step 1.

The algorithm stops when T_{sim} exceeds a user-chosen threshold T_{stop}. At this time,

$$\langle f \rangle \approx \frac{f_{sim}}{T_{sim}}$$

is our estimate of the stationary average (8.1).

Algorithm 8.6 generates correct stationary averages by the same argument as in the previous section. It is worth mentioning that Algorithms 8.4 and 8.5 have a property not shared by Algorithms 8.1 and 8.2: the escape events (T_{par}, Z_{par}) in these parallel steps have the correct

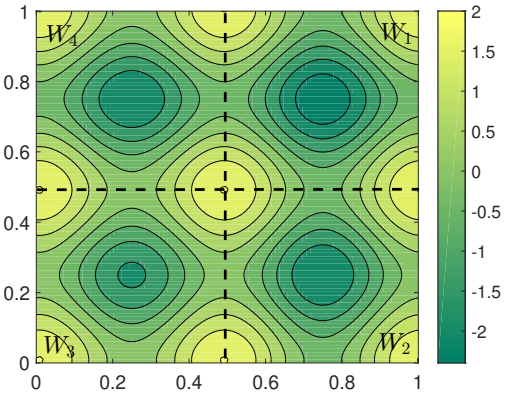

Figure 6. *Contour plot of the potential V in (9.2). V has 4 basins of attraction, W_1, W_2, W_3, W_4, of different depths, with W_1 the deepest. Recall that a basin of attraction for V is a set of initial conditions $x(0)$ for which the differential equation $dx(t)/dt = -\nabla V(x(t))$ has a unique long-time limit.*

law for the PDMP. This allows us to use Algorithm 8.6 to compute the dynamics of $Z(t)_{t\geq 0}$. More precisely, Algorithm 8.6 leads to a PDMP dynamics that is correct on the quotient space obtained by considering each $W \in \mathcal{V}$ as a single point. Note that Algorithm 8.3 cannot be used in this way, as it generates dynamics of the skeleton chain and not the PDMP.

9. Numerics.

Here we test our algorithms above on a toy PDMP model, our aim being to illustrate Algorithms 8.3 and 8.6. We will use these algorithms to sample the stationary average of a function f with respect to the Boltzmann density $\pi = Z^{-1}e^{-\beta V}$, where f and V are defined below and $\beta > 0$ is the inverse temperature. The toy model is a two-dimensional version of a PDMP that may be defined in an arbitrary dimension $d-1$, as follows. Let $d_0, \dots, d_{N-1} \in \mathbb{R}^{d-1}$ be direction vectors such that $d_0 + \cdots + d_{N-1} = 0$. Let $\mathbb{Z}_N$ denote the integers modulo N, consider the indices of the d_k's as elements of $\mathbb{Z}_N$, and for $k, \ell \in \mathbb{Z}_N$ define

$$F_{k,\ell}(x) = \beta(d_k + \cdots + d_{k+\ell}) \cdot \nabla V(x).$$

Consider the PDMP with generator defined by

$$Lg(x,k) = d_k \cdot \nabla g(x,k) + [g(x,k-1) - g(x,k)] \max_{0\leq \ell \leq N-1} F_{k,\ell}(x) \tag{9.1}$$

for suitable $g : \Omega \times \mathbb{Z}_N \to \mathbb{R}$, where either $\Omega = \mathbb{R}^{d-1}$ or Ω is a cube in $\mathbb{R}^{d-1}$ with periodic boundaries. This is the generator for a PDMP that, when moving in direction d_k at point x, switches to direction d_{k-1} with rate $\max_{0\leq \ell \leq N-1} F_{k,\ell}(x)$. The resulting process can be seen as a rejection-free or "lifted" version of the sequential Metropolis algorithm [29, 36], historically the first nonreversible sampling algorithm [27, 36] for sampling the Boltzmann distribution. Straightforward calculations show this PDMP has invariant density proportional to π; see Remark 10.7 in section 10.1.

We consider the case where the state space is $\Omega = [0,1]^2$ with periodic boundaries, $N = 4$, $d_0 = (1,0)$, $d_1 = (-1,0)$, $d_2 = (0,1)$, $d_3 = (0,-1)$, and the potential energy V is pictured in

Figure 6. Specifically

$$V(x,y) = \cos(4\pi x) + \cos(4\pi y) + \frac{1}{5}\sin(2\pi x) + \frac{1}{5}\sin(2\pi y). \tag{9.2}$$

We define $\mathcal{W}$ and $\mathcal{V}$ using the basins of attraction W_i, $i = 1,\dots,4$, defined as the four squares of equal side length $1/2$ inside $[0,1]^2$. See Figure 6. Thus with x and k the position and direction variables, respectively, of the skeleton chain and PDMP, and θ the jump time variable of the skeleton chain,

$$\mathcal{W} = \{\{(x,k,\theta) : x \in W_i,\, k \in \{0,1,2,3\},\, \theta > 0\} : i = 1,2,3,4\},$$
$$\mathcal{V} = \{\{(x,k) : x \in W_i,\, k \in \{0,1,2,3\}\} : i = 1,2,3,4\}.$$

That is, the skeleton chain or PDMP is in a given set in $\mathcal{W}$ or $\mathcal{V}$ at a particular time if and only if its position variable belongs to a given W_i at that time.

Algorithm 9.1 Time discretization of the PDMP (9.1).

Choose an initial point $Z(0) \in \mathbb{R}^{d-1} \times \{0,\dots,N-1\}$ and pick $d_0,\dots,d_{N-1} \in \mathbb{R}^{d-1}$ with $\sum_{k=0}^{N-1} d_k = 0$. Choose a time step $\delta t > 0$. Then set $t = 0$ and iterate:

1. If $Z(t) = (x,k)$, define an acceptance probability

$$p = \min_{0\le \ell \le N-1} \exp\left(\beta V(x) - \beta V(x + d_k\delta t + \cdots + d_{k+\ell}\delta t)\right).$$

2. With probability p, set $Z(t+\delta t) = (x + d_k\delta t, k)$; otherwise set $Z(t+\delta t) = (x, k-1)$.

3. Update $t \leftarrow t + \delta t$ and return to step 1.

Here, $Z(n\delta t)_{n\ge 0}$ has invariant measure proportional to $e^{-\beta V}$; see Remark 10.8.

We tested Algorithms 8.3 and 8.6 with the synchronous parallel steps, Algorithm 8.1 and Algorithm 8.4, respectively. We used both algorithms to estimate the stationary average $\langle f \rangle$ where $f(x,k) = \mathbb{1}_{x\in W_1}$, the characteristic function of the deepest basin of V. We used up to $R = 100$ replicas and decorrelation times that were the same in each basin, $T^{\nu}_{corr} \equiv T^{\nu}_{corr}(W_i)$ and $T^{\mu}_{corr} \equiv T^{\mu}_{corr}(W_i)$, $i = 1,\dots,4$. We used Algorithm 9.1 with time step $\delta t = 10^{-2}$ to simulate the PDMP. In Algorithm 8.6 we took $\Delta t = \delta t = 10^{-2}$. The results are in Figures 7, 8, and 9.

To analyze our results, we defined an idealized speedup factor as follows. Let T^R be an idealized wall-clock time for a simulation of Algorithm 8.3 or 8.6 using the parallel steps, Algorithms 8.1 and 8.4, respectively, up to a fixed time T_{stop}. The idealized wall-clock time T^R is obtained by assuming that we use R parallel processors with zero communication cost, such that on each processor, one step of the skeleton chain is computed in wall-clock time 1. Writing T^1 for the wall-clock time corresponding to 1 processor or direct serial simulation, we define

$$\text{time speedup} = \frac{T^R}{T^1}.$$

We include computation time from the dephasing step—i.e., the QSD sampling step in Algorithms 8.1 and 8.4—as part of T^R. We assume this dephasing is done using a Fleming–

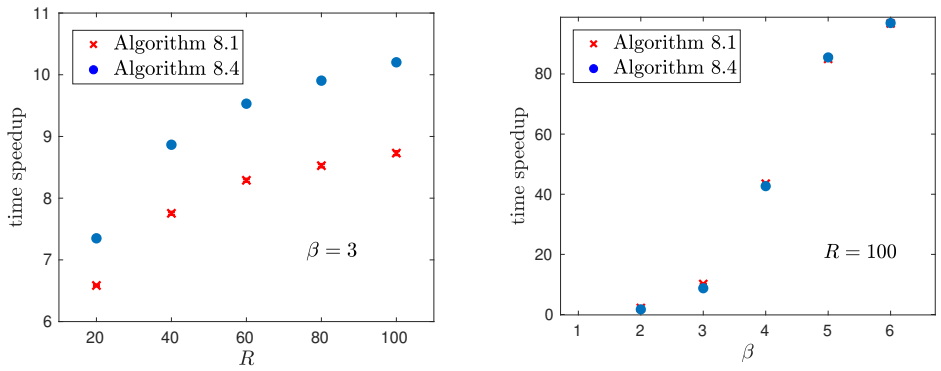

Figure 7. *Left: Time speedup vs. number of replicas, R, when $\beta = 3$. Right: Time speedup vs. β when $R = 100$. In Algorithm 8.1 we used $T^{\nu}_{corr} = 100$, while in Algorithm 8.4 we used $T^{\mu}_{corr} = 6$. The decorrelation times were chosen so that Algorithms 8.1 and 8.4 would have similar values for the time speedup. Error bars for each data point were obtained from 50 independent simulations, but they are smaller than the data markers. In the limit $\beta \to \infty$, the computational effort to sample the QSD vanishes compared to the effort to generate an escape event using serial simulation. In this limit, at left, we expect scaling like R for finite R. On the other hand, at right, we expect the time speedup to level off at $R = 100$ as $\beta \to \infty$. This figure is based on simulations performed by Peter Christman.*

Viot-based technique as described above, with R copies of the underlying skeleton chain or PDMP.

Processor communication, which we do not account for, of course takes a toll on the time speedup. However, the processor communication cost is small compared to the rest of the computational effort if the sets in $\mathcal{W}$ and $\mathcal{V}$ are significantly metastable. Thus, our time speedup gives a reasonable picture of the gain that can be expected.

The time speedup depends on the parameters in Algorithms 8.3 and 8.6. If all other parameters are held constant, the time speedup increases with R or β, due to increasing parallelization or metastability, respectively (Figure 7), while the time speedup decreases with T^{ν}_{corr} and T^{μ}_{corr}, due to increased effort to sample the QSD (Figure 8). With increasing metastability, the relative computational effort to sample the QSD decreases in comparison with the effort to simulate an escape from a metastable set. Since the latter is done in parallel, increasing the metastability leads to a larger time speedup.

Figure 9 shows that the approximation f_{sim}/T_{sim} approaches the stationary average $\langle f \rangle$ as the decorrelation times T^{ν}_{corr} and T^{μ}_{corr} increase, as expected. As discussed above, appropriate values of these QSD sampling times depend on the degree of metastability. Note that the approximations f_{sim}/T_{sim} are quite good even for small QSD sampling times. This was true not just for $f(x,k) = \mathbb{1}_{x \in W_1}$ but also for a variety of other functions. This feature, of reasonable accuracy in ParRep even for relatively small decorrelation times, was observed before in [64, 65]. Here this may be a result of the momentum-like direction variables, which can make the PDMP unlikely to immediately escape from a metastable set just after entering.

10. Proofs. Our first two results below, Propositions 10.1 and 10.2, establish the memoryless distribution of the escape time starting from the QSD, and the independence of the

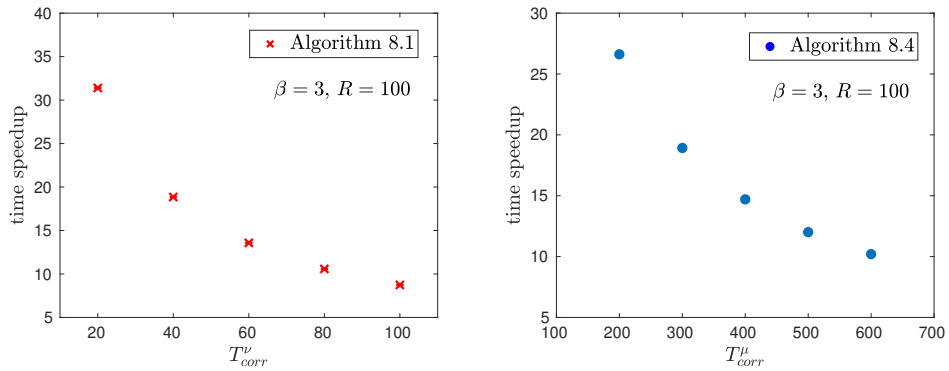

Figure 8. *Left: Time speedup vs. T^{ν}_{corr} in Algorithm 8.1. Right: Time speedup vs. T^{μ}_{corr} in Algorithm 8.4. In both plots $\beta = 3$ and $R = 100$. Error bars for each data point were obtained from 50 independent simulations, but they are smaller than the data markers. This figure is based on simulations performed by Peter Christman.*

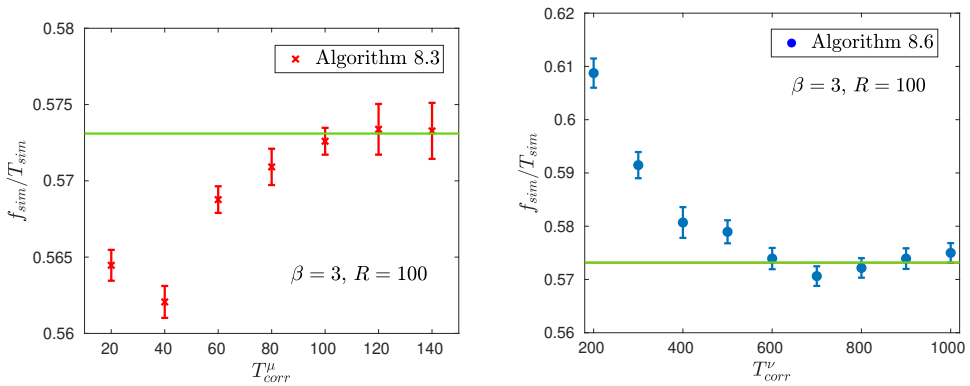

Figure 9. *Left: Approximation f_{sim}/T_{sim} of the stationary average $\langle f \rangle$ using Algorithm 8.3. Right: Approximation f_{sim}/T_{sim} of the stationary average $\langle f \rangle$ using Algorithm 8.6. In both plots , $\beta = 3$ and $R = 100$, and simulations ran for T_{sim} exceeding 10^6. Error bars are empirical standard deviations obtained from 50 independent simulations for each data point. The exact value $\langle f \rangle$ is indicated with a solid line. This figure is based on simulations performed by Peter Christman.*

escape time and escape point, for general Markov processes. See, for instance, [16] for details. We include proofs for completeness.

Proposition 10.1. *Let $X(t)_{t\geq 0}$ have a QSD ρ in U, and suppose $\mathcal{L}(X(0)) = \rho$. Suppose $T = \inf\{t \geq 0 : X(t) \notin U\}$ is finite almost surely. Then T has a memoryless distribution; that is, T is either exponentially or geometrically distributed.*

Proof. The definition (3.1) of the QSD together with the Markov property shows that $\mathcal{L}(X(t+s)_{s\geq 0}|T > t) = \mathcal{L}(X(s)_{s\geq 0})$. Thus, $\mathbb{P}(T > t+s|T > t) = \mathbb{P}(T > s)$ for any $s, t \geq 0$. When T is finite-valued, the only distribution satisfying this is

$$\mathbb{P}(T > t) = e^{-\lambda t}. \tag{10.1}$$

We use (10.1) to indicate either the (continuous) exponential distribution with parameter

$\lambda > 0$ or the (discrete) geometric distribution with parameter $p = 1 - e^{-\lambda}$. ∎

Proposition 10.2. *Let $X(t)_{t\geq 0}$ have a QSD ρ in U, and suppose $\mathcal{L}(X(0)) = \rho$. Suppose $T = \inf\{t \geq 0 : X(t) \notin U\}$ is finite almost surely. Then T and $X(T)$ are independent.*

Proof. By Proposition 10.1, $\mathbb{P}(T > t) = e^{-\lambda t}$. For A in the complement of U,

$$\begin{aligned}\mathbb{P}(T \in ((n-1)t, nt], X(T) \in A) &= \mathbb{P}(T > (n-1)t, X(nt \wedge T) \in A)\\ &= \mathbb{P}(X(nt \wedge T) \in A | T > (n-1)t)\mathbb{P}(T > (n-1)t)\\ &= \mathbb{P}(X(T) \in A, T \leq nt | T > (n-1)t)\mathbb{P}(T > (n-1)t)\\ &= \mathbb{P}(X(T) \in A, T \leq t)e^{-\lambda(n-1)t},\end{aligned}$$

where the last step uses (3.1). Summing over $n \geq 1$ establishes the result:

$$\begin{aligned}\mathbb{P}(X(T) \in A) &= \mathbb{P}(X(T) \in A, T \leq t)\frac{1}{1 - e^{-\lambda t}}\\ &= \frac{\mathbb{P}(X(T) \in A, T \leq t)}{\mathbb{P}(T \leq t)}.\end{aligned}$$

∎

Below, we write MGF for the moment generating function of a random variable. The results in Proposition 10.3 below hold in both continuous and discrete time. To connect the discrete and continuous time cases, we write $1 - p = e^{-\lambda}$, where $p \in (0, 1)$ is the geometric parameter and $\lambda > 0$ is the exponential rate.

Proposition 10.3. *Let $t_1, t_2, \ldots$ be nonnegative deterministic times such that $\sum_{m=1}^{\infty} t_m = \infty$. Let $\tau_1, \tau_2, \ldots$ be random variables such that $\mathbb{P}(\tau_1 > t) = e^{-\lambda t}$ and*

$$\mathbb{P}(\tau_m > t | \tau_{m-1} > t_{m-1}, \ldots, \tau_1 > t_1) = e^{-\lambda t}, \quad m \geq 2. \tag{10.2}$$

Let $L = \inf\{m \geq 1 : \tau_m \leq t_m\}$. Then

$$\mathbb{P}(t_1 + \cdots + t_{L-1} + \tau_L > t) = e^{-\lambda t}.$$

Proof. Let $s_m = t_1 + \cdots + t_m$ and $s_0 = 0$. By (10.2) and induction,

$$\mathbb{P}(\tau_m > t_m, \ldots, \tau_1 > t_1) = e^{-\lambda(t_1 + \ldots + t_m)} = e^{-\lambda s_m}. \tag{10.3}$$

Using (10.2) again, in the continuous case,

$$\begin{aligned}&\mathbb{E}\left(e^{u\tau_m} \mathbb{1}_{\tau_m \leq t_m} | \tau_{m-1} > t_{m-1}, \ldots, \tau_1 > t_1\right)\\ &= \int_0^{t_m} e^{us} \lambda e^{-\lambda s}\, ds = \frac{\lambda}{u - \lambda}\left(e^{(u-\lambda)t_m} - 1\right),\end{aligned} \tag{10.4}$$

while in the discrete case, where $e^{-\lambda} = 1 - p$,

$$\begin{aligned}&\mathbb{E}\left(e^{u\tau_m} \mathbb{1}_{\tau_m \leq t_m} | \tau_{m-1} > t_{m-1}, \ldots, \tau_1 > t_1\right)\\ &= \sum_{s=1}^{t_m} e^{us} e^{-\lambda(s-1)}(1 - e^{-\lambda}) = \frac{e^u - e^{u-\lambda}}{1 - e^{u-\lambda}}\left(1 - e^{(u-\lambda)t_m}\right).\end{aligned} \tag{10.5}$$

Note also that

$$\{L = m\} = \{\tau_m \le t_m, \tau_{m-1} > t_{m-1}, \dots, \tau_1 > t_1\}. \tag{10.6}$$

Consider the continuous case. Combining (10.3), (10.4), and (10.6) gives

$$\begin{aligned}
&\mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{L=m}\right)\\
&= \mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{\tau_m\le t_m}\mathbb{1}_{\tau_{m-1}>t_{m-1},\dots,\tau_1>t_1}\right)\\
&= \mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{\tau_m\le t_m}|\tau_{m-1} > t_{m-1},\dots,\tau_1 > t_1\right)\mathbb{P}(\tau_{m-1} > t_{m-1},\dots,\tau_1 > t_1)\\
&= \frac{\lambda}{u-\lambda}\left(e^{(u-\lambda)t_m} - 1\right)e^{-\lambda s_{m-1}}.
\end{aligned} \tag{10.7}$$

We now see that $s_{L-1} + \tau_L$ has the MGF of an exponential(λ) random variable:

$$\begin{aligned}
\mathbb{E}\left(e^{u(s_{L-1}+\tau_L)}\right) &= \sum_{m=1}^{\infty}\mathbb{E}\left(e^{u(s_{L-1}+\tau_L)}\mathbb{1}_{L=m}\right)\\
&= \sum_{m=1}^{\infty} e^{us_{m-1}}\mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{L=m}\right)\\
&= \frac{\lambda}{u-\lambda}\sum_{m=1}^{\infty}\left(e^{(u-\lambda)s_m} - e^{(u-\lambda)s_{m-1}}\right)\\
&= \frac{\lambda}{\lambda-u} \qquad \text{if } u < \lambda.
\end{aligned}$$

Similarly, in the discrete case, combining (10.3), (10.5), and (10.6) gives

$$\begin{aligned}
&\mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{L=m}\right)\\
&= \mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{\tau_m\le t_m}\mathbb{1}_{\tau_{m-1}>t_{m-1},\dots,\tau_1>t_1}\right)\\
&= \mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{\tau_m\le t_m}|\tau_{m-1} > t_{m-1},\dots,\tau_1 > t_1\right)\mathbb{P}(\tau_{m-1} > t_{m-1},\dots,\tau_1 > t_1)\\
&= \frac{e^u - e^{u-\lambda}}{1-e^{u-\lambda}}\left(1 - e^{(u-\lambda)t_m}\right)e^{-\lambda s_{m-1}}.
\end{aligned} \tag{10.8}$$

This shows again that $s_{L-1} + \tau_L$ has the MGF of a geometric(p) random variable, via

$$\begin{aligned}
\mathbb{E}\left(e^{u(s_{L-1}+\tau_L)}\right) &= \sum_{m=1}^{\infty}\mathbb{E}\left(e^{u(s_{L-1}+\tau_L)}\mathbb{1}_{L=m}\right)\\
&= \sum_{m=1}^{\infty} e^{us_{m-1}}\mathbb{E}\left(e^{u\tau_m}\mathbb{1}_{L=m}\right)\\
&= \frac{e^u - e^{u-\lambda}}{1-e^{u-\lambda}}\sum_{m=1}^{\infty}\left(e^{(u-\lambda)s_{m-1}} - e^{(u-\lambda)s_m}\right)\\
&= \frac{e^u - e^{u-\lambda}}{1-e^{u-\lambda}} \qquad \text{if } u < \lambda\\
&= \frac{pe^u}{1-(1-p)e^u} \qquad \text{if } u < -\log(1-p).
\end{aligned}$$

■

Proof of Theorem 5.2. Let A be a subset of the complement of U. Note that, due to (5.1), the events $\{X_m(T_m) \in A, T_m \leq t\}$ and $\{T_{m-1} > t_{m-1}, \ldots, T_1 > t_1\}$ are independent conditional on $\{X_m(0) = x\}$. Using this, (5.2), and Proposition 10.2, we obtain

$$\begin{aligned}
&\mathbb{P}(X_m(T_m) \in A, T_m \leq t | T_{m-1} > t_{m-1}, \ldots, T_1 > t_1) \\
&= \int \mathbb{P}(X_m(T_m) \in A, T_m \leq t | X_m(0) = x, T_{m-1} > t_{m-1}, \ldots, T_1 > t_1) \\
&\qquad \times \mathbb{P}(X_m(0) \in dx | T_{m-1} > t_{m-1}, \ldots, T_1 > t_1) \\
&= \int \mathbb{P}(X_m(T_m) \in A, T_m \leq t | X_m(0) = x) \rho(dx) \\
&= \mathbb{P}(X(T) \in A, T \leq t) \\
&= \mathbb{P}(X(T) \in A) \mathbb{P}(T \leq t).
\end{aligned} \tag{10.9}$$

Taking A as the complement of U in (10.9), and using Proposition 10.1,

$$\mathbb{P}(T_m > t | T_{m-1} > t_{m-1}, \ldots, T_1 > t_1) = \mathbb{P}(T > t) = e^{-\lambda t} \quad \text{for some } \lambda > 0. \tag{10.10}$$

By (5.2), $\mathbb{P}(T_1 > t) = e^{-\lambda t}$. Thus by Proposition 10.3, $\mathcal{L}(T_{par}) = \mathcal{L}(T)$. Notice that

$$\{L = m\} = \{T_m \leq t_m, T_{m-1} > t_{m-1}, \ldots, T_1 > t_1\}. \tag{10.11}$$

From (10.9), (10.10), and (10.11) we have

$$\begin{aligned}
&\mathbb{P}(X_{par} \in A, T_{par} > t | L = m) \\
&= \mathbb{P}(X_m(T_m) \in A, s_{m-1} + T_m > t | L = m) \\
&= \mathbb{P}\left(X_m(T_m) \in A, T_m > t - s_{m-1} \middle| T_m \leq t_m, T_{m-1} > t_{m-1}, \ldots, T_1 > t_1\right) \\
&= \mathbb{P}\left(X_m(T_m) \in A, T_m \in (t - s_{m-1}, t_m] \middle| T_{m-1} > t_{m-1}, \ldots, T_1 > t_1\right) \\
&\qquad \times \mathbb{P}(T_m \leq t_m | T_{m-1} > t_{m-1}, \ldots, T_1 > t_1)^{-1} \\
&= \mathbb{P}(X(T) \in A) \mathbb{P}(T \in (t - s_{m-1}, t_m]) \mathbb{P}(T \leq t_m)^{-1} \\
&= \mathbb{P}(X(T) \in A) \mathbb{P}(T > t - s_{m-1} | T \leq t_m).
\end{aligned} \tag{10.12}$$

From (10.12) we conclude that

$$\begin{aligned}
\mathbb{P}(X_{par} \in A, T_{par} > t) &= \sum_{m=1}^{\infty} \mathbb{P}(X_{par} \in A, T_{par} > t | L = m) \mathbb{P}(L = m) \\
&= \mathbb{P}(X(T) \in A) \sum_{m=1}^{\infty} \mathbb{P}(T > t - s_{m-1} | T \leq t_m) \mathbb{P}(L = m).
\end{aligned} \tag{10.13}$$

In the last display, taking $t = 0$ shows $\mathbb{P}(X_{par} \in A) = \mathbb{P}(X(T) \in A)$, while taking A as the complement of U shows $\mathbb{P}(T_{par} > t) = \sum_{m=1}^{\infty} \mathbb{P}(T > t - s_{m-1} | T \leq t_m) \mathbb{P}(L = m)$. Thus, (10.13) shows that $\mathcal{L}(X_{par}) = \mathcal{L}(X(T))$ and X_{par}, T_{par} are independent. ∎

Proof of Theorem 5.3. We consider only the discrete time case, since the arguments in the continuous time case are analogous. Let $r \in \mathbb{N}$ be fixed. Observe that

$$\sum_{t=0}^{r-1} \mathbb{P}(T > t) = \sum_{t=0}^{r-1} \mathbb{E}(\mathbb{1}_{T>t}) = \sum_{t=0}^{\infty} \mathbb{E}(\mathbb{1}_{T\wedge r>t}) = \mathbb{E}(T \wedge r).$$

By the preceding display and the definition (3.1) of the QSD ρ,

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=0}^{T\wedge r-1} g(X(t))\right) &= \sum_{s=1}^{\infty}\sum_{t=0}^{s-1} \mathbb{E}\left(g(X(t))\mathbb{1}_{T\wedge r=s}\right) \\
&= \sum_{t=0}^{\infty} \mathbb{E}\left(g(X(t))\mathbb{1}_{T\wedge r>t}\right) \\
&= \sum_{t=0}^{r-1} \mathbb{E}\left(g(X(t))\mathbb{1}_{T>t}\right) \\
&= \sum_{t=0}^{r-1} \mathbb{E}(g(X(t))|T>t)\mathbb{P}(T>t) \\
&= \left(\int g\, d\rho\right) \mathbb{E}(T \wedge r).
\end{aligned} \tag{10.14}$$

Since $\{L \geq m\} = \{T_{m-1} > t_{m-1}, \ldots, T_1 > t_1\}$, using (5.1), (5.2), and (10.14) we get

$$\begin{aligned}
&\mathbb{E}\left(\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\,\middle|\, L \geq m\right) \\
&= \int \mathbb{E}\left(\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\,\middle|\, X_m(0) = x, L \geq m\right) \mathbb{P}(X_m(0) \in dx | L \geq m) \\
&= \int \mathbb{E}\left(\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\,\middle|\, X_m(0) = x\right) \rho(dx) \\
&= \mathbb{E}\left(\sum_{t=0}^{T\wedge t_m-1} g(X(t))\right) = \left(\int g\, d\rho\right) \mathbb{E}(T \wedge t_m) \\
&= \left(\int g\, d\rho\right) \mathbb{E}(T_m \wedge t_m | L \geq m),
\end{aligned} \tag{10.15}$$

where the last line of (10.15) follows by taking $g \equiv \mathbb{1}$ in the first four lines of (10.15). By

Theorem 5.2, $\mathcal{L}(T) = \mathcal{L}(T_{par}) = \mathcal{L}\big(\sum_{m=1}^{L} T_m \wedge t_m\big)$ and thus

$$\begin{aligned}\mathbb{E}(T) &= \mathbb{E}\left(\sum_{m=1}^{L} T_m \wedge t_m\right)\\ &= \sum_{n=1}^{\infty}\sum_{m=1}^{n} \mathbb{E}\left(T_m \wedge t_m \mathbb{1}_{L=n}\right)\\ &= \sum_{m=1}^{\infty} \mathbb{E}\left(T_m \wedge t_m \mathbb{1}_{L\geq m}\right)\\ &= \sum_{m=1}^{\infty} \mathbb{E}(T_m \wedge t_m | L \geq m)\mathbb{P}(L \geq m).\end{aligned} \tag{10.16}$$

Now by (10.15) and (10.16),

$$\begin{aligned}\mathbb{E}\left(\sum_{m=1}^{L}\sum_{t=0}^{T_m\wedge t_m - 1} g(X_m(t))\right) &= \sum_{n=1}^{\infty}\sum_{m=1}^{n}\mathbb{E}\left(\mathbb{1}_{L=n}\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\right)\\ &= \sum_{m=1}^{\infty}\mathbb{E}\left(\mathbb{1}_{L\geq m}\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\right)\\ &= \sum_{m=1}^{\infty}\mathbb{E}\left(\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\,\middle|\, L\geq m\right)\mathbb{P}(L\geq m)\\ &= \left(\int g\,d\rho\right)\sum_{m=1}^{\infty}\mathbb{E}(T_m\wedge t_m | L\geq m)\mathbb{P}(L\geq m)\\ &= \left(\int g\,d\rho\right)\mathbb{E}(T).\end{aligned} \tag{10.17}$$

Letting $r \to \infty$ in (10.14), using dominated convergence, and comparing with (10.17),

$$\mathbb{E}\left(\sum_{m=1}^{L}\sum_{t=0}^{T_m\wedge t_m-1} g(X_m(t))\right) = \mathbb{E}\left(\sum_{t=0}^{T-1} g(X(t))\right),$$

as desired. ∎

Below we will need the following basic facts.

Lemma 10.4. *Let $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{K}$ be σ-algebras.*

(i) *Let $\sigma(\mathcal{G},\mathcal{H})$ be the σ-algebra generated by $\mathcal{G}$ and $\mathcal{H}$. Suppose that $\mathcal{F}$, $\sigma(\mathcal{G},\mathcal{H})$ are independent conditional on $\mathcal{K}$, and that $\mathcal{G}$, $\mathcal{H}$ are independent conditional on $\mathcal{K}$. Then $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$ are mutually independent conditional on $\mathcal{K}$.*

(ii) *Suppose $\mathcal{H} \subseteq \mathcal{G}$. If $\mathcal{F}$ and $\mathcal{G}$ are independent, then $\mathcal{F}$ and $\mathcal{G}$ are independent conditional on $\mathcal{H}$.*

Proof. Throughout let $A \in \mathcal{F}$, $B \in \mathcal{G}$, and $C \in \mathcal{H}$, and write $\mathbb{1}_S$ for the characteristic or indicator function of a set S. Consider (i). Since $A \in \mathcal{F}$, $B \cap C \in \sigma(\mathcal{G},\mathcal{H})$, and $\mathcal{F}$, $\sigma(\mathcal{G},\mathcal{H})$ are

independent conditional on $\mathcal{K}$, $\mathbb{E}(\mathbb{1}_A\mathbb{1}_{B\cap C}|\mathcal{K}) = \mathbb{E}(\mathbb{1}_A|\mathcal{K})\mathbb{E}(\mathbb{1}_{B\cap C}|\mathcal{K})$ almost surely. Similarly $\mathbb{E}(\mathbb{1}_B\mathbb{1}_C|\mathcal{K}) = \mathbb{E}(\mathbb{1}_B|\mathcal{K})\mathbb{E}(\mathbb{1}_C|\mathcal{K})$ almost surely. Thus,

$$\begin{aligned}\mathbb{E}(\mathbb{1}_A\mathbb{1}_B\mathbb{1}_C|\mathcal{K}) = \mathbb{E}(\mathbb{1}_A\mathbb{1}_{B\cap C}|\mathcal{K}) &= \mathbb{E}(\mathbb{1}_A|\mathcal{K})\mathbb{E}(\mathbb{1}_{B\cap C}|\mathcal{K}) \\ &= \mathbb{E}(\mathbb{1}_A|\mathcal{K})\mathbb{E}(\mathbb{1}_B\mathbb{1}_C|\mathcal{K}) = \mathbb{E}(\mathbb{1}_A|\mathcal{K})\mathbb{E}(\mathbb{1}_B|\mathcal{K})\mathbb{E}(\mathbb{1}_C|\mathcal{K})\end{aligned}$$

almost surely, which proves (i).

Consider now (ii). Define $P = \mathbb{E}(\mathbb{1}_A\mathbb{1}_B|\mathcal{H})$ and $Q = \mathbb{E}(\mathbb{1}_A|\mathcal{H})\mathbb{E}(\mathbb{1}_B|\mathcal{H})$. As P and Q are $\mathcal{H}$-measurable and $C \in \mathcal{H}$ is arbitrary, if $\mathbb{E}(P\mathbb{1}_C) = \mathbb{E}(Q\mathbb{1}_C)$, then we can use uniqueness of conditional expectation to conclude $P = Q$ almost surely, so that (ii) holds. Since $\mathcal{H} \subseteq \mathcal{G}$, $B \cap C \in \mathcal{G}$. Moreover $A \in \mathcal{F}$ and $\mathcal{F}, \mathcal{G}$ are independent, so

$$\mathbb{E}(P\mathbb{1}_C) = \mathbb{E}(\mathbb{1}_A\mathbb{1}_B\mathbb{1}_C) = \mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B \cap C). \tag{10.18}$$

The first equality in (10.18) comes from the definition of conditional expectation. Since $A \in \mathcal{F}$ and $\mathcal{F}, \mathcal{G}$ are independent, $\mathbb{E}(\mathbb{1}_A|\mathcal{G}) = \mathbb{E}(\mathbb{1}_A)$. So by the tower property,

$$\mathbb{E}(\mathbb{1}_A|\mathcal{H}) = \mathbb{E}(\mathbb{E}(\mathbb{1}_A|\mathcal{G})|\mathcal{H}) = \mathbb{E}(\mathbb{E}(\mathbb{1}_A)|\mathcal{H}) = \mathbb{E}(\mathbb{1}_A) = \mathbb{P}(A).$$

Moreover since $\mathbb{1}_C$ is $\mathcal{H}$-measurable, $\mathbb{E}(\mathbb{1}_B|\mathcal{H})\mathbb{1}_C = \mathbb{E}(\mathbb{1}_B\mathbb{1}_C|\mathcal{H})$. Thus,

$$\begin{aligned}\mathbb{E}(Q\mathbb{1}_C) &= \mathbb{E}(\mathbb{E}(\mathbb{1}_A|\mathcal{H})\mathbb{E}(\mathbb{1}_B|\mathcal{H})\mathbb{1}_C) \\ &= \mathbb{E}(\mathbb{P}(A)\mathbb{E}(\mathbb{1}_B\mathbb{1}_C|\mathcal{H})) \\ &= \mathbb{P}(A)\mathbb{E}(\mathbb{E}(\mathbb{1}_{B\cap C}|\mathcal{H})) \\ &= \mathbb{P}(A)\mathbb{E}(\mathbb{1}_{B\cap C}) = \mathbb{P}(A)\mathbb{P}(B \cap C),\end{aligned} \tag{10.19}$$

with the last line using the tower property. Now (ii) follows from (10.18)–(10.19). ∎

Proof of Proposition 6.1. Adopt the notation of Assumption 5.1. Below let j, k, ℓ denote positive integers. It is easy to check that $r_j = r_k$ if and only if $k - j$ is an integer multiple of R, while $m_{k-\ell R} = m_k - \ell$ when $\ell \le m_k$. Thus,

$$\begin{aligned}\{m_j : j < k,\, r_j = r_k\} &= \{m_{k-\ell R} : \ell \le m_k\} \\ &= \{0, 1, \ldots, m_k - 1\},\end{aligned} \tag{10.20}$$

with both sides empty if $m_k = 0$. See Figure 2. Define

$$\zeta^r = \inf\{t \ge 0 : Y^r(t) \notin U\}.$$

Fix $k \ge 2$ and note that, by definition of the fragments,

$$\{\zeta^{r_j} > (m_j + 1)\Delta t \ \ \forall\, j < k\} \subseteq \{T_{k-1} > \Delta t, \ldots, T_1 > \Delta t\}, \tag{10.21}$$

where $E \subseteq E'$ indicates that event E' occurs whenever E occurs. By (10.20),

$$\begin{aligned}&\{Y^{r_j}(t) \in U \ \ \forall\, t \in [m_j\Delta t, (m_j + 1)\Delta t], j \le k\} \\ &\subseteq \{Y^{r_k}(t) \in U \ \ \forall\, t \in [0, (m_k + 1)\Delta t]\}.\end{aligned} \tag{10.22}$$

By definition of the fragments and (10.22),

$$\begin{aligned}\{T_{k-1} > \Delta t, \dots, T_1 > \Delta t\} &= \{Y^{r_j}(t) \in U \ \ \forall\, t \in [m_j \Delta t, (m_j+1)\Delta t], j < k\} \\ &\subseteq \{Y^{r_j}(t) \in U \ \ \forall\, t \in [0, (m_j+1)\Delta t], j < k\} \\ &= \{\zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k\}.\end{aligned} \tag{10.23}$$

Combining (10.21) and (10.23),

$$\{T_{k-1} > \Delta t, \dots, T_1 > \Delta t\} = \{\zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k\}. \tag{10.24}$$

Due to independence of $Y^r(t)_{t\geq 0}$ over r and Lemma 10.4(ii),

$$\begin{aligned}&\text{conditional on } \{\zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k \ \ s.t. \ \ r_j = r_k\}, \\ &\text{the event } \{\zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k \ \ s.t. \ \ r_j \neq r_k\} \\ &\text{is independent of } Y^{r_k}(m_k \Delta t).\end{aligned} \tag{10.25}$$

Again using (10.20),

$$\{\zeta^{r_k} > m_k \Delta t\} = \{\xi^{r_j} > (m_j+1)\Delta t \ \ \forall j < k \ \text{ s.t. } \ r_j = r_k\}. \tag{10.26}$$

Combining (10.24), (10.25), and (10.26) and using (3.1),

$$\begin{aligned}&\mathcal{L}(X_k(0) | T_{k-1} > \Delta t, \dots, T_1 > \Delta t) \\ &= \mathcal{L}(Y^{r_k}(m_k \Delta t) | \zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k) \\ &= \mathcal{L}\left(Y^{r_k}(m_k \Delta t) \middle| \, \zeta^{r_j} > (m_j+1)\Delta t \ \ \forall\, j < k \ \text{ s.t. } \ r_j = r_k\right) \\ &= \mathcal{L}\left(Y^{r_k}(m_k \Delta t) \middle| \, \zeta^{r_k} > m_k \Delta t\right) = \rho.\end{aligned} \tag{10.27}$$

As $Y^1(t)_{t\geq 0}$ is a copy of $X(t)_{t\geq 0}$ with $\mathcal{L}(X(0)) = \rho$, in particular $\mathcal{L}(Y^1(0)) = \rho$. Thus,

$$\mathcal{L}(X_1(0)) = \mathcal{L}(Y^{r_1}(m_1)) = \mathcal{L}(Y^1(0)) = \rho.$$

We have now established (5.2) of Assumption 5.1.

Consider now (5.1). Let $k \geq 1$. Due to the independence of $Y^r(t)_{t\geq 0}$ over r and Lemma 10.4(ii), we see that, conditional on $X_k(0)$, $(X_\ell(t)_{0\leq t\leq \Delta t})_{r_\ell = r_k}$ is independent of $(X_\ell(t)_{0\leq t\leq \Delta t})_{r_\ell \neq r_k}$. For $k \geq 2$, the Markov property of $Y^{r_k}(t)_{t\geq 0}$ and (10.20) show that, conditional on $X_k(0)$, $X_k(t)_{0\leq t\leq \Delta t}$ is independent of $(X_\ell(t)_{0\leq t\leq \Delta t})_{\ell<k, r_\ell = r_k}$. By Lemma 10.4(i) with $\mathcal{F} = \sigma((X_\ell(t)_{0\leq t\leq \Delta t})_{\ell<k, r_\ell \neq r_k})$, $\mathcal{G} = \sigma((X_\ell(t)_{0\leq t\leq \Delta t})_{\ell<k, r_\ell = r_k})$, $\mathcal{H} = \sigma(X_k(t)_{0\leq t\leq \Delta t})$, and $\mathcal{K} = \sigma(X_k(0))$, we have, for $k \geq 2$,

$$\text{conditional on } X_k(0),\ X_k(t)_{0\leq t\leq \Delta t} \text{ is independent of } (X_\ell(t)_{0\leq t\leq \Delta t})_{\ell<k}. \tag{10.28}$$

Now define the fragments' irrelevant futures as follows. Let $X_k(t)_{t\geq \Delta t}$ be copies of $X(t)_{t\geq 0}$ that evolve forward in time independently of everything else. That is, for each $k \geq 1$, conditional on $X_k(\Delta t)$, $X_k(t)_{t\geq \Delta t}$, $X_k(t)_{0\leq t\leq \Delta t}$, and $(X_\ell(t)_{t\geq 0})_{\ell<k}$ are mutually independent. From (10.28) it is easy to see this is possible, as the irrelevant futures have no bearing on the definitions of the fragments. Now by construction of the irrelevant futures and (10.28), it is easy to see that for $k \geq 2$, conditional on $X_k(0)$, $X_k(t)_{t\geq 0}$ is independent of $(X_\ell(t)_{t\geq 0})_{\ell<k}$. This proves (5.1) in Assumption 5.1. ∎

Proof of Proposition 6.2. Adopt the notation of Assumption 5.1. This proof will follow the same basic steps as the proof of Proposition 6.1, but the justifications will be different. Let $\zeta^r = \inf\{t \geq 0 : Y^r(t) \notin U\}$ be as above.

Fix $k \geq 2$. We first claim that (10.20) still holds. Let $n \in \{0, 1, \ldots, m_k - 1\}$. By the surjectivity assumption in (iii) there is j such that $r_j = r_k$ and $m_j = n$. Since $m_j = n < m_k$ and $r_j = r_k$, from (ii) we have $t^{r_j}_{wall}(m_j) \leq t^{r_k}_{wall}(m_k)$. Since $j \neq k$, using monotonicity in (iii) we conclude that $j < k$. Thus $\{0, 1, \ldots, m_k - 1\} \subseteq \{m_j : j < k, r_j = r_k\}$. Now consider m_j such that $j < k$ and $r_j = r_k$. By monotonicity in (iii) we must have $t^{r_j}_{wall}(m_j) < t^{r_k}_{wall}(m_k)$. Then by (ii) we can conclude that $m_j < m_k$. Thus $\{m_j : j < k, r_j = r_k\} = \{0, 1, \ldots, m_k - 1\}$.

Next we establish (5.2). Equipped with (10.20), we see that (10.26) holds. Moreover, since (6.1) agrees with (6.2), the same steps as in the proof of Proposition 6.1 show that (10.24) holds. On the other hand, (10.25) holds because of (i), Lemma 10.4(ii), and independence of $Y^r(t)_{t\geq 0}$ over r. The sequence of equalities in (10.27) then holds, with the last equality using (i) again. It remains to show that $\mathcal{L}(X_1(0)) = \rho$. Suppose $m_1 > 0$. By surjectivity in (iii) there is $j > 1$ such that $m_j = 0$ and $r_j = r_1$. But then (ii) implies $t^{r_j}_{wall}(m_j) \leq t^{r_1}_{wall}(m_1)$, which contradicts monotonicity in (iii). Thus $m_1 = 0$, so we can apply (i) to conclude that $\mathcal{L}(X_1(0)) = \mathcal{L}(Y^{r_1}(m_1)) = \rho$. Thus (5.2) in Assumption 5.1 holds.

Consider now (5.1). By (i) and the independence of $Y^r(t)_{t\geq 0}$ over r, conditional on $X_k(0)$, $(X_\ell(t)_{0\leq t\leq \Delta t})_{r_\ell = r_k}$ is independent of $(X_\ell(t)_{0\leq t\leq \Delta t})_{r_\ell \neq r_k}$. Recall that (10.20) still holds. Thus for $k \geq 2$, by the Markov property of $Y^{r_k}(t)_{t\geq 0}$ and (10.20), conditional on $X_k(0)$, $X_k(t)_{0\leq t\leq \Delta t}$ is independent of $(X_\ell(t)_{0\leq t\leq \Delta t})_{\ell<k, r_\ell = r_k}$. By Lemma 10.4(i) we conclude that (10.28) holds for $k \geq 2$. Let the trajectory fragments' irrelevant futures be independent of everything else as in the proof of Proposition 6.1. Following the reasoning in that proof, we see that (5.1) in Assumption 5.1 holds. ∎

Proof of Theorem 8.2. The statements (i) and (ii) follow from Propositions 6.1 and 6.2, respectively, with $((\xi^r_n, \theta^r_n)_{n\geq 0})^{r=1,\ldots,R}$ taking the place of $(Y^r(t)_{t\geq 0})^{r=1,\ldots,R}$, and with $t_m \equiv \Delta t = 1$ and $g(\xi, \theta) = \int_0^\theta f(\psi(t, \xi))\, dt$. ∎

Proof of Theorem 8.4. The statements (i) and (ii) follow from Propositions 6.1 and 6.2, respectively, with $(Z^r(t)_{t\geq 0})^{r=1,\ldots,R}$ taking the place of $(Y^r(t)_{t\geq 0})^{r=1,\ldots,R}$, and with $t_m \equiv \Delta t$ and $g = f$. ∎

10.1. Supplementary results. We first show that the decoupling of wall-clock times from the speed of computing $X(t)_{t\geq 0}$ is a necessary condition for consistency. Below we break assumption (i) in Proposition 6.2 by assuming the wall-clock times to obtain that the initial QSD samples $Y^r(0)$, $r = 1, \ldots, R$, in the parallel step are correlated with the positions of those samples.

Remark 10.5. In Proposition 6.2, if (i) does not hold, then the conclusions of Theorems 5.2 and 5.3 may not hold.

Example. Let $X(t)_{t\geq 0}$ be a simple random walk on $\mathbb{Z}$, meaning $X(t+1) - X(t) = 1$ or -1, each with probability $1/2$. Let $U = \{0, 1\}$. The QSD ρ of $X(t)_{t\geq 0}$ in U is simply the uniform distribution on U. Assume $X(t)_{t\geq 0}$ has initial distribution $\mathcal{L}(X(0)) = \rho$, and let

$(Y^r(t)_{t\geq 0})^{r=1,\dots,R}$ be independent copies of $X(t)_{t\geq 0}$. Suppose

$$t^r_{wall}(0) < t^s_{wall}(0) \qquad \text{whenever } Y^r(0) = 0 \text{ and } Y^s(0) = 1. \tag{10.29}$$

Notice that (10.29) violates (i) of Proposition 6.2. Assume, however, that (ii) and (iii) in Proposition 6.2 hold. Then arguments similar to those in the proof of Proposition 6.2 show that $\{Y^{r_1}(m_1) = 1\} = \{Y^r(0) = 1 \ \forall r\}$. Adopt the notation of Algorithm 5.1. Then by the above and the definition (6.2) of the fragments,

$$\begin{aligned} \mathbb{P}(T_{par} = 1, X_{par} = 2) &= \mathbb{P}(Y^{r_1}(m_1) = 1, Y^{r_1}(m_1 + 1) = 2) \\ &= \frac{1}{2}\mathbb{P}(Y^{r_1}(m_1) = 1) \\ &= \frac{1}{2}\mathbb{P}(Y^r(0) = 1 \ \forall r) = \frac{1}{2}\left(\frac{1}{2}\right)^R. \end{aligned} \tag{10.30}$$

Similarly,

$$\mathbb{P}(T = 1, X(T) = 2) = \mathbb{P}(X(0) = 1, X(1) = 2) = \frac{1}{2}\mathbb{P}(X(0) = 1) = \frac{1}{4}. \tag{10.31}$$

Notice when $R > 1$, (10.30) and (10.31) show the conclusion of Theorem 5.2 does not hold, as $\mathbb{P}(T_{par} = 1, X_{par} = 2) \neq \mathbb{P}(T = 1, X(T) = 2)$. A similar construction shows the conclusion of Theorem 5.3 can fail when (i) does not hold. ∎

The next two results below are formal calculations related to claims made in the text above. These results could be made precise using results in [22, 23]. However, we stick to formal computations for brevity.

Remark 10.6. Suppose (7.5) holds for all $x \in \mathbb{R}^{d-1}$ and $i \in \mathcal{I}$. Then $e^{-V(x)}$ is formally invariant for a PDMP generated by (7.3).

Formal proof. Let L be defined as in (7.3). We will show that

$$\sum_{i\in\mathcal{I}} \int_{\mathbb{R}^{d-1}} Lf(x,i)\pi(x,i)\,dx = 0,$$

provided (7.5) holds and $\pi(x,i) \propto e^{-V(x)}$, where $\propto$ indicates "proportional to." Write

$$\lambda_i(x,i) = -\sum_{j\neq i} \lambda_j(x,i).$$

With sufficient regularity we can integrate by parts to get

$$\begin{aligned}\sum_i \int Lf(x,i)\pi(x,i)\,dx &\propto \sum_i \int \left(d_i \cdot \nabla f(x,i) + \sum_j \lambda_j(x,i) f(x,j) \right) e^{-V(x)}\,dx \\ &= -\sum_i \int f(x,i) \nabla \cdot \left(d_i e^{-V(x)} \right) dx \\ &\quad + \sum_i \sum_j \int \lambda_i(x,j) f(x,i) e^{-V(x)}\,dx \\ &= \sum_i \int f(x,i) \left(d_i \cdot \nabla V(x) + \sum_j \lambda_i(x,j) \right) e^{-V(x)}\,dx. \end{aligned} \tag{10.32}$$

Above, all the sums are over $\mathcal{I}$ and integrals are over $\mathbb{R}^{d-1}$. If (7.5) holds,

$$d_i \cdot \nabla V(x) + \sum_j \lambda_i(x,j) = d_i \cdot \nabla V(x) + \sum_{j \neq i} (\lambda_i(x,j) - \lambda_j(x,i)) = 0.$$

Comparing with (10.32) gives the result. ∎

Note that the calculation in Remark 10.6 shows (7.5) is a necessary condition for (7.3) to define a PDMP with an invariant distribution of the form $\pi(x,i) \propto e^{-V(x)}$.

Remark 10.7. $e^{-\beta V(x)}$ is formally invariant for a PDMP generated by (9.1).

Formal proof. Let L be defined as in (9.1). We will show that

$$\sum_{k=0}^{N-1} \int_\Omega Lf(x,k)\pi(x,k)\,dx = 0,$$

provided $\pi(x,k) \propto e^{-\beta V(x)}$. Recall $d_0, \ldots, d_{N-1} \in \mathbb{R}^{d-1}$ sum to 0, and we consider the indices of the d_k's as elements of $\mathbb{Z}_N$, the integers modulo N. Write

$$F_{k,\ell}(x) = \beta(d_k + \cdots + d_{k+\ell}) \cdot \nabla V(x).$$

With sufficient regularity we can integrate by parts to get

$$\begin{aligned} &\sum_{k=0}^{N-1} \int_\Omega Lg(x,k)\pi(x,k)\,dx \\ &\propto \sum_{k=0}^{N-1} \int_\Omega \left(d_k \cdot \nabla g(x,k) + [g(x,k-1) - g(x,k)] \max_{0 \le \ell \le N-1} F_{k,\ell}(x) \right) e^{-\beta V(x)}\,dx \\ &= -\sum_{k=0}^{N-1} \int_\Omega g(x,k) \nabla \cdot \left(d_k e^{-\beta V(x)} \right) dx \\ &\quad + \sum_{k=0}^{N-1} \int_\Omega g(x,k) \left(\max_{0 \le \ell \le N-1} F_{k+1,\ell}(x) - \max_{0 \le \ell \le N-1} F_{k,\ell}(x) \right) e^{-\beta V(x)}\,dx \\ &= \sum_{k=0}^{N-1} \int_\Omega g(x,k) \left(\beta d_k \cdot \nabla V(x) + \max_{0 \le \ell \le N-1} F_{k+1,\ell}(x) - \max_{0 \le \ell \le N-1} F_{k,\ell}(x) \right) e^{-\beta V(x)}\,dx, \end{aligned}$$

where, when $\Omega = \mathbb{R}^{d-1}$, we assume V grows sufficiently fast at ∞ so that we can neglect the boundary term from the integration by parts. Observe that, because $\sum_{\ell=0}^{N-1} d_{k+\ell} = 0$ and $d_{k+N} = d_k$, we have

$$\begin{aligned}&\{d_k + d_{k+1}, d_k + d_{k+1} + d_{k+2}, \dots, d_k + \cdots + d_{k+N-1}, d_k + \cdots + d_{k+N}\}\\ &= \{d_k + d_{k+1}, d_k + d_{k+1} + d_{k+2}, \dots, d_k + \cdots + d_{k+N-1}, d_k\}\\ &= \{d_k, d_k + d_{k+1}, \dots, d_{k+N-1}\}.\end{aligned}$$

It follows that

$$\begin{aligned}&\beta d_k \cdot \nabla V(x) + \max_{0\le\ell\le N-1} F_{k+1,\ell}(x) - \max_{0\le\ell\le N-1} F_{k,\ell}(x)\\ &= \beta \max_{0\le\ell\le N-1} (d_k + \cdots + d_{k+\ell+1}) \cdot \nabla V(x) - \beta \max_{0\le\ell\le N-1} (d_k + \cdots + d_{k+\ell}) \cdot \nabla V(x)\\ &= 0.\end{aligned}$$

This gives the desired result. ∎

Remark 10.8. $e^{-\beta V}$ is invariant for $Z(n\delta t)_{n\ge 0}$ defined in Algorithm 9.1.

Proof. Write the acceptance probability in Algorithm 9.1 as

$$A_k(x) = \min_{0\le\ell\le N-1} \exp\left(\beta V(x) - \beta V(x + d_k\delta t + \cdots + d_{k+\ell}\delta t)\right).$$

Arguing similarly as in the formal proof of Remark 10.7, since $\sum_{k=0}^{N-1} d_k = 0$ we have

$$\frac{A_k(x)}{A_{k+1}(x + d_k\delta t)} = \exp(\beta V(x) - \beta V(x + d_k\delta t)).$$

Now let $\pi(x,k) \propto e^{-\beta V(x)}$. Then the last display shows that

$$\pi(x + d_k\delta t, k) = \pi(x,k)A_k(x) + \pi(x + d_k\delta t, k+1)(1 - A_{k+1}(x + d_k\delta t)). \tag{10.33}$$

Inspecting Algorithm 9.1, we see that (10.33) demonstrates the required result. ∎

Acknowledgments. The author gratefully thanks Peter Christman for producing the numerical results leading to Figures 7, 8, and 9 in section 9, as well as Petr Plecháč, Gideon Simpson, and Ting Wang for helpful conversations.

REFERENCES

[1] A. Alfonsi, E. Cances, G. Turinici, B. Di Ventura, and W. Huisinga, *Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems*, in CEMRACS 2004—Mathematics and Applications to Biology and Medicine, ESAIM Proc. 14, EDP Sciences, 2005, pp. 1–13.

[2] D. F. Anderson and T. G. Kurtz, *Continuous Time Markov Chain Models for Chemical Reaction Networks*, in Design and Analysis of Biomolecular Circuits, Springer, 2011, pp. 3–42.

[3] D. Aristoff, *The parallel replica method for computing equilibrium averages of Markov chains*, Monte Carlo Methods Appl., 21 (2015), pp. 255–273.

[4] D. Aristoff, *Analysis and optimization of weighted ensemble sampling*, ESAIM Math. Model. Numer. Anal., 52 (2018), pp. 1219–1238.
[5] D. Aristoff, T. Lelièvre, and G. Simpson, *The parallel replica method for simulating long trajectories of Markov chains*, Appl. Math. Res. Express. AMRX, 2014 (2014), pp. 332–352.
[6] J. Bierkens, P. Fearnhead, and G. Roberts, *The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data*, Ann. Statist., 47 (2019), pp. 1288–1320.
[7] J. Bierkens, G. Roberts, et al., *A piecewise deterministic scaling limit of lifted Metropolis–Hastings in the Curie–Weiss model*, Ann. Appl. Probab., 27 (2017), pp. 846–882.
[8] A. Binder, T. Lelièvre, and G. Simpson, *A generalized parallel replica dynamics*, J. Comput. Phys., 284 (2015), pp. 595–616.
[9] A. Bouchard-Côté, S. J. Vollmer, and A. Doucet, *The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method*, J. Amer. Statist. Assoc., 113 (2018), pp. 855–867.
[10] P. C. Bressloff, *Stochastic switching in biology: From genotype to phenotype*, J. Phys. A, 50 (2017), 133001.
[11] P. C. Bressloff and J. N. Maclaurin, *Stochastic hybrid systems in cellular neuroscience*, J. Math. Neurosci., 8 (2018), 12.
[12] P. C. Bressloff and J. M. Newby, *Metastability in a stochastic neural network modeled as a velocity jump Markov process*, SIAM J. Appl. Dyn. Syst., 12 (2013), pp. 1394–1435.
[13] P. C. Bressloff and J. M. Newby, *Path integrals and large deviations in stochastic hybrid systems*, Phys. Rev. E, 89 (2014), 042701.
[14] N. Champagnat and D. Villemonais, *Exponential convergence to quasi-stationary distribution and q-process*, Probab. Theory Related Fields, 164 (2016), pp. 243–283.
[15] N. Champagnat and D. Villemonais, *General Criteria for the Study of Quasi-Stationarity*, preprint, https://arxiv.org/abs/1712.08092, 2017.
[16] P. Collet, S. Martínez, and J. San Martín, *Quasi-Stationary Distributions: Markov Chains, Diffusions and Dynamical Systems*, Springer Science & Business Media, 2012.
[17] N. De Freitas, C. Andrieu, P. Højen-Sørensen, M. Niranjan, and A. Gee, *Sequential Monte Carlo methods for neural networks*, in Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 359–379.
[18] P. Del Moral, *Feynman–KAC Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer, 2004.
[19] P. Del Moral and A. Doucet, *Particle methods: An introduction with applications*, ESAIM Proc., 44 (2014), pp. 1–46.
[20] P. Del Moral and J. Garnier, *Genealogical particle analysis of rare events*, Ann. Appl. Probab., 15 (2005), pp. 2496–2534.
[21] A. B. Duncan, T. Lelievre, and G. Pavliotis, *Variance reduction using nonreversible Langevin samplers*, J. Statist. Phys., 163 (2016), pp. 457–491.
[22] A. Durmus, A. Guillin, and P. Monmarché, *Geometric Ergodicity of the Bouncy Particle Sampler*, preprint, https://arxiv.org/abs/1807.05401, 2018.
[23] A. Durmus, A. Guillin, and P. Monmarché, *Piecewise Deterministic Markov Processes and Their Invariant Measure*, preprint, https://arxiv.org/abs/1807.05421, 2018.
[24] P. Ferrari and N. Marić, *Quasi stationary distributions and Fleming-Viot processes in countable spaces*, Electronic J. Probab., 12 (2007), pp. 684–702.
[25] J. Goodman and J. Weare, *Ensemble samplers with affine invariance*, Commun. Appl. Math. Comput. Sci., 5 (2010), pp. 65–80.
[26] J. Harland, M. Michel, T. A. Kampmann, and J. Kierfeld, *Event-chain Monte Carlo algorithms for three- and many-particle interactions*, Europhys. Lett., 117 (2017), 30001.
[27] W. K. Hastings, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Biometrika, 57 (1970), pp. 97–109.
[28] H.-W. Kang and T. G. Kurtz, *Separation of time-scales and model reduction for stochastic reaction networks*, Ann. Appl. Probab., 23 (2013), pp. 529–583.
[29] S. C. Kapfer and W. Krauth, *Irreversible local Markov chains with rapid convergence towards equilibrium*, Phys. Rev. Lett., 119 (2017), 240603.
[30] C. Le Bris, T. Lelievre, M. Luskin, and D. Perez, *A mathematical formalization of the parallel*

replica dynamics, Monte Carlo Methods Appl., 18 (2012), pp. 119–146.
[31] T. Lelievre, *Two mathematical tools to analyze metastable stochastic processes*, in Numerical Mathematics and Advanced Applications 2011, Springer, 2013, pp. 791–810.
[32] T. Lelièvre, *Accelerated dynamics: Mathematical foundations and algorithmic improvements*, European Phys. J. Special Topics, 224 (2015), pp. 2429–2444.
[33] T. Lelièvre, F. Nier, and G. A. Pavliotis, *Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion*, J. Statist. Phys., 152 (2013), pp. 237–274.
[34] T. Lelièvre, M. Rousset, and G. Stoltz, *Free Energy Computations: A Mathematical Perspective*, World Scientific, 2010.
[35] E. Lyman, F. M. Ytreberg, and D. M. Zuckerman, *Resolution exchange simulation*, Phys. Rev. Lett., 96 (2006), 028105.
[36] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys., 21 (1953), pp. 1087–1092.
[37] M. Michel, S. C. Kapfer, and W. Krauth, *Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps*, J. Chem. Phys., 140 (2014), 054116.
[38] P. Monmarché, *Piecewise Deterministic Simulated Annealing*, preprint, https://arxiv.org/abs/1410.1656, 2014.
[39] J. Newby, *Bistable switching asymptotics for the self regulating gene*, J. Phys. A, 48 (2015), 185001.
[40] J. M. Newby, *Isolating intrinsic noise sources in a stochastic genetic switch*, Phys. Biol., 9 (2012), 026002.
[41] J. M. Newby, *Spontaneous excitability in the Morris–Lecar model with ion channel noise*, SIAM J. Appl. Dyn. Syst., 13 (2014), pp. 1756–1791.
[42] J. M. Newby, P. C. Bressloff, and J. P. Keener, *Breakdown of fast-slow analysis in an excitable system with channel noise*, Phys. Rev. Lett., 111 (2013), 128101.
[43] J. M. Newby and J. P. Keener, *An asymptotic analysis of the spatially inhomogeneous velocity-jump process*, Multiscale Model. Simul., 9 (2011), pp. 735–765.
[44] Y. Nishikawa and K. Hukushima, *Event-chain Monte Carlo algorithm for continuous spin systems and its application*, J. Phys. Conf. Ser., 750 (2016), 012014.
[45] A. Pakman, D. Gilboa, D. Carlson, and L. Paninski, *Stochastic bouncy particle sampler*, Proc. Mach. Learn. Res., 70 (2017), pp. 2741–2750.
[46] D. Perez, E. D. Cubuk, A. Waterland, E. Kaxiras, and A. F. Voter, *Long-time dynamics through parallel trajectory splicing*, J. Chem. Theory Comput., 12 (2015), pp. 18–28.
[47] D. Perez, R. Huang, and A. F. Voter, *Long-time molecular dynamics simulations on massively parallel platforms: A comparison of parallel replica dynamics and parallel trajectory splicing*, J. Materials Res., 33 (2018), pp. 813–822.
[48] L. Rey-Bellet and K. Spiliopoulos, *Irreversible Langevin samplers and variance reduction: A large deviations approach*, Nonlinearity, 28 (2015), pp. 2081–2103.
[49] R. Rudnicki and M. Tyran-Kamińska, *Piecewise Deterministic Processes in Biological Models*, Springer, 2017.
[50] H. Salis and Y. Kaznessis, *Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions*, J. Chem. Phys., 122 (2005), 054103.
[51] M. R. Shirts and J. D. Chodera, *Statistically optimal analysis of samples from multiple equilibrium states*, J. Chem. Phys., 129 (2008), 124105.
[52] G. Simpson and M. Luskin, *Numerical analysis of parallel replica dynamics*, ESAIM Math. Model. Numer. Anal., 47 (2013), pp. 1287–1314.
[53] M. R. Sørensen and A. F. Voter, *Temperature-accelerated dynamics for simulation of infrequent events*, J. Chem. Phys., 112 (2000), pp. 9599–9606.
[54] T. D. Swinburne and D. Perez, *Self-optimized construction of transition rate matrices from accelerated atomistic simulations with Bayesian uncertainty quantification*, Phys. Rev. Materials, 2 (2018), 053802.
[55] D. Talay, *Numerical Solution of Stochastic Differential Equations*, Stoch. Stoch. Rep., 47 (1994), pp. 121–126.
[56] J. O. Tempkin, B. Qi, M. G. Saunders, B. Roux, A. R. Dinner, and J. Weare, *Using multiscale preconditioning to accelerate the convergence of iterative molecular calculations*, J. Chem. Phys., 140

(2014), 05B6141.
[57] E. H. Thiede, B. Van Koten, J. Weare, and A. R. Dinner, *Eigenvector method for umbrella sampling enables error analysis*, J. Chem. Phys., 145 (2016), 084115.
[58] G. M. Torrie and J. P. Valleau, *Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling*, J. Comput. Phys., 23 (1977), pp. 187–199.
[59] P. Vanetti, A. Bouchard-Côté, G. Deligiannidis, and A. Doucet, *Piecewise-Deterministic Markov Chain Monte Carlo*, preprint, https://arxiv.org/abs/1707.05296, 2017.
[60] Y. Vardi, *Empirical distributions in selection bias models*, Ann. Statist., 13 (1985), pp. 178–203.
[61] A. F. Voter, *Hyperdynamics: Accelerated molecular dynamics of infrequent events*, Phys. Rev. Lett., 78 (1997), 3908.
[62] A. F. Voter, *Parallel replica method for dynamics of infrequent events*, Phys. Rev. B, 57 (1998), R13985.
[63] A. F. Voter, *Accelerated Molecular Dynamics Methods*, Tech. report, Los Alamos National Laboratory (LANL), 2012.
[64] T. Wang and P. Plecháč, *Parallel replica dynamics method for bistable stochastic reaction networks: Simulation and sensitivity analysis*, J. Chem. Phys., 147 (2017), 234110.
[65] T. Wang, P. Plecháč, and D. Aristoff, *Stationary averaging for multiscale continuous time Markov chains using parallel replica dynamics*, Multiscale Model. Simul., 16 (2018), pp. 1–27.
[66] S. Winkelmann and C. Schütte, *Hybrid models for chemical reaction networks: Multiscale theory and application to gene regulatory systems*, J. Chem. Phys., 147 (2017), 114115.
[67] C. Wu and C. P. Robert, *Generalized Bouncy Particle Sampler*, preprint, https://arxiv.org/abs/1706.04781, 2017.
[68] S.-J. Wu, C.-R. Hwang, and M. T. Chu, *Attaining the optimal Gaussian diffusion acceleration*, J. Statist. Phys., 155 (2014), pp. 571–590.