Bernoulli **22**(2), 2016, 794–856
DOI: 10.3150/14-BEJ676

On the exact and ε-strong simulation of (jump) diffusions

MURRAY POLLOCK*, ADAM M. JOHANSEN** and GARETH O. ROBERTS†

Department of Statistics, University of Warwick, Coventry, CV4 7AL, United Kingdom.
E-mail: *m.pollock@warwick.ac.uk; **a.m.johansen@warwick.ac.uk; †gareth.o.roberts@warwick.ac.uk

This paper introduces a framework for simulating finite dimensional representations of (jump) diffusion sample paths over finite intervals, without discretisation error (*exactly*), in such a way that the sample path can be restored at any finite collection of time points. Within this framework we extend existing exact algorithms and introduce novel adaptive approaches. We consider an application of the methodology developed within this paper which allows the simulation of upper and lower bounding processes which almost surely constrain (jump) diffusion sample paths to any specified tolerance. We demonstrate the efficacy of our approach by showing that with finite computation it is possible to determine whether or not sample paths cross various irregular barriers, simulate to any specified tolerance the first hitting time of the irregular barrier and simulate killed diffusion sample paths.

Keywords: adaptive exact algorithms; barrier crossing probabilities; Brownian path space probabilities; exact simulation; first hitting times; killed diffusions

1. Introduction

Diffusions and jump diffusions are widely used across a number of application areas. An extensive literature exists in economics and finance, spanning from the seminal Black–Scholes model [10,25,26] to the present [4,16]. Other applications can be easily found within the physical [29] and life sciences [18,19] to name but a few. A jump diffusion $V:\mathbb{R}\to\mathbb{R}$ is a Markov process. In this paper, we consider jump diffusions defined as the solution to a stochastic differential equation (SDE) of the form (denoting $V_{t-}:=\lim_{s\uparrow t}V_s$),

$$\mathrm{d}V_t=\beta(V_{t-})\,\mathrm{d}t+\sigma(V_{t-})\,\mathrm{d}W_t+\mathrm{d}J_t^{\lambda,\mu},\qquad V_0=v\in\mathbb{R},t\in[0,T],\tag{1}$$

where $\beta:\mathbb{R}\to\mathbb{R}$ and $\sigma:\mathbb{R}\to\mathbb{R}_+$ denote the (instantaneous) drift and diffusion coefficients, respectively, W_t is a standard Brownian motion and $J_t^{\lambda,\mu}$ denotes a compound Poisson process. $J_t^{\lambda,\mu}$ is parameterised with (finite) jump intensity $\lambda:\mathbb{R}\to\mathbb{R}_+$ and jump size coefficient $\mu:\mathbb{R}\to\mathbb{R}$ with jumps distributed with density f_μ. All coefficients are themselves (typically) dependent on V_t. Regularity conditions are assumed to hold to ensure the existence of a unique non-explosive weak solution (see, e.g., [28,30]). To apply the methodology developed within this paper we primarily restrict our attention to univariate diffusions and require a number of additional conditions on the coefficients of (1), details and a discussion of which can be found in Section 2.

1350-7265 © 2016 ISI/BS

Motivated by the wide range of possible applications we are typically interested (directly or indirectly) in the measure of V on the path space induced by (1), denoted $\mathbb{T}^v_{0,T}$. As $\mathbb{T}^v_{0,T}$ is typically not explicitly known, in order to compute expected values $\mathbb{E}_{\mathbb{T}^v_{0,T}}[h(V)]$ for various test functions $h:\mathbb{R}\to\mathbb{R}$, we can construct a Monte Carlo estimator. In particular, if it is possible to draw independently $V^{(1)}, V^{(2)}, \ldots, V^{(n)} \sim \mathbb{T}^v_{0,T}$ then by applying the strong law of large numbers we can construct a consistent estimator of the expectation (unbiasedness follows directly by linearity),

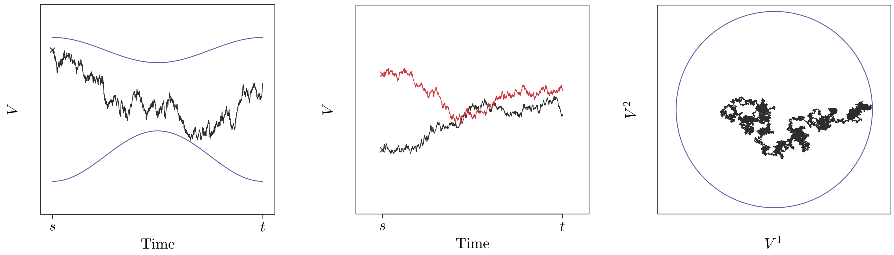
$$\text{w.p. 1:} \qquad \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} h\big(V^{(i)}\big) = \mathbb{E}_{\mathbb{T}^v_{0,T}}\big[h(V)\big]. \tag{2}$$

Furthermore, provided $\mathbb{V}_{\mathbb{T}^v_{0,T}}[h(V)] =: \sigma^2_h < \infty$, by application of the central limit theorem we have,

$$\lim_{n\to\infty} \sqrt{n}\left[\mathbb{E}_{\mathbb{T}^v_{0,T}}\big[h(V)\big] - \frac{1}{n}\sum_{i=1}^{n} h\big(V^{(i)}\big)\right] \overset{\mathcal{D}}{=} \xi, \qquad \text{where } \xi \sim \mathrm{N}\big(0,\sigma^2_h\big). \tag{3}$$

Unfortunately, as diffusion sample paths are infinite dimensional random variables it isn't possible to draw an entire sample path from $\mathbb{T}^v_{0,T}$ – at best we can hope to simulate some finite dimensional subset of the sample path, denoted V^{fin} (we further denote the remainder of the sample path by $V^{\text{rem}} := V \setminus V^{\text{fin}}$). Careful consideration has to be taken as to how to simulate V^{fin} as any numerical approximation impacts the unbiasedness and convergence of the resulting Monte Carlo estimator (2), (3). Equally, consideration has to be given to the form of the test function, h, to ensure it's possible to evaluate it given V^{fin}.

To illustrate this point we consider some possible applications. In Figure 1(a), (b) and (c) we are interested in whether a simulated sample path $V \sim \mathbb{T}^v_{0,T}$, crosses some barrier (i.e., for some set A we have $h(V) := \mathbb{1}(V \in A)$). Note that in all three cases in order to evaluate h we would require some characterisation of the entire sample path (or some further approximation) and even for diffusions with constant coefficients and simple barriers this is difficult. For instance, as illustrated in Figure 1(c), even in the case where $\mathbb{T}^v_{0,T}$ is known (e.g., when $\mathbb{T}^v_{0,T}$ is Wiener



(a) Non-linear two sided barrier (b) Diffusion barrier (c) 2-D circular barrier

Figure 1. Examples of test functions in which evaluation requires the characterisation of an entire sample path.

measure) and the barrier is known in advance and has a simple form, there may still not exist any exact approach to evaluate whether or not a sample path has crossed the barrier.

Diffusion sample paths can be simulated approximately at a finite collection of time points by *discretisation* [21,24,30], noting that as Brownian motion has a Gaussian transition density then over short intervals the transition density of (1) can be approximated by that of an SDE with fixed coefficients (by a continuity argument). This can be achieved by breaking the interval the sample path is to be simulated over into a fine mesh (e.g., of size Δt), then iteratively (at each mesh point) fixing the coefficients and simulating the sample path to the next mesh point. For instance, in an *Euler* discretisation [21] of (1), the sample path is propagated between mesh points as follows (where $\xi \sim \mathrm{N}(0, \Delta t)$ and $\mu_t \sim f_\mu(\cdot; V_t)$),

$$V_{t+\Delta t} = \begin{cases} V_t + \beta(V_t)\Delta t + \sigma(V_t)\xi, & \text{w.p. } \exp\{-\lambda(V_t)\Delta t\}, \\ V_t + \beta(V_t)\Delta t + \sigma(V_t)\xi + \mu_t, & \text{w.p. } 1 - \exp\{-\lambda(V_t)\Delta t\}. \end{cases} \tag{4}$$

It is hoped the simulated sample path (generated approximately at a finite collection of mesh points) can be used as a proxy for an entire sample path drawn exactly from $\mathbb{T}^v_{0,T}$. More complex discretisation schemes exist (e.g., by exploiting Itô's lemma to make higher order approximations or by local linearisation of the coefficients [24,30]), but all suffer from common problems. In particular, minimising the approximation error (by increasing the mesh density) comes at the expense of increased computational cost, and further approximation or interpolation is needed to obtain the sample path at non-mesh points (which can be non-trivial). As illustrated in Figure 2, even when our test function h only requires the simulation of sample paths at a single time point, discretisation introduces approximation error resulting in the loss of unbiasedness of our Monte Carlo estimator (2). If $\mathbb{T}^v_{0,T}$ has a highly non-linear drift or includes a compound jump process or h requires simulation of sample paths at a collection of time points then this problem is exacerbated. In the case of the examples in Figure 1, mesh based discretisation schemes don't sufficiently characterise simulated sample paths for the evaluation of h.

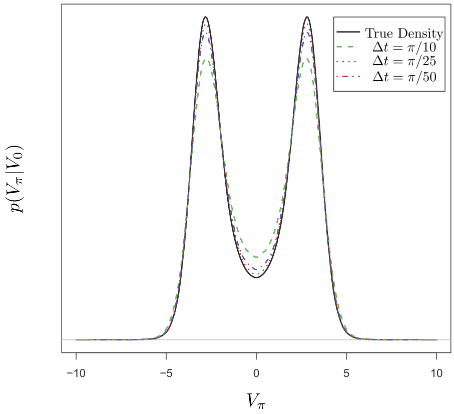


Figure 2. Density of V_π (obtained using 1 000 000 exact samples) and approximations given by an Euler discretisation with various mesh sizes, given $V_0 = 0$ where $\mathrm{d}V_t = \sin(V_t)\,\mathrm{d}t + \mathrm{d}W_t$.

Recently, a new class of *Exact Algorithms* for simulating sample paths at finite collections of time points without approximation error have been developed for both diffusions [5,6,9,14] and jump diffusions [13,17,20]. These algorithms are based on rejection sampling, noting that sample paths can be drawn from the (target) measure $\mathbb{T}^v_{0,T}$ by instead drawing sample paths from an equivalent proposal measure $\mathbb{P}^v_{0,T}$, and accepting or rejecting them with probability proportional to the Radon–Nikodým derivative of $\mathbb{T}^v_{0,T}$ with respect to $\mathbb{P}^v_{0,T}$. However, as with discretisation schemes, given a simulated sample path at a finite collection of time points subsequent simulation of the sample path at any other intermediate point may require approximation or interpolation and may not be exact. Furthermore, we are again unable to evaluate test functions of the type illustrated in Figure 1.

The key contribution of this paper is the introduction of a novel mathematical framework for constructing exact algorithms which addresses this problem. In particular, instead of exactly simulating sample paths at finite collections of time points, we focus on the extended notion of simulating *skeletons* which in addition characterise the entire sample path.

Definition 1 (Skeleton). *A skeleton* ($\mathcal{S}$) *is a finite dimensional representation of a diffusion sample path* ($V \sim \mathbb{T}^v_{0,T}$), *that can be simulated without any approximation error by means of a proposal sample path drawn from an equivalent proposal measure* ($\mathbb{P}^v_{0,T}$) *and accepted with probability proportional to* $\frac{\mathrm{d}\mathbb{T}^v_{0,T}}{\mathrm{d}\mathbb{P}^v_{0,T}}(V)$, *which is sufficient to restore the sample path at any finite collection of time points exactly with finite computation where* $V|\mathcal{S} \sim \mathbb{P}^v_{0,T}|\mathcal{S}$. *A skeleton typically comprises information regarding the sample path at a finite collection of time points and path space information which ensures the sample path is almost surely constrained to some compact interval.*

Methodology for simulating skeletons (the size and structure of which is dependent on exogenous randomness) is driven by both computational and mathematical considerations (i.e., we need to ensure the required computation is finite and the skeleton is exact). Central to both notions is that the path space of the proposal measure $\mathbb{P}^v_{0,T}$ can be partitioned (into a set of *layers*), and that the layer to which any sample path belongs to can be simulated.

Definition 2 (Layer). *A layer* $R(V)$, *is a function of a diffusion sample path* $V \sim \mathbb{P}^v_{0,T}$ *which determines the compact interval to which any particular sample path* $V(\omega)$ *is constrained.*

To illustrate the concept of a layer and skeleton, we could, for instance, have $R(V) = \inf\{i \in \mathbb{N}\colon \forall u \in [0,T], V_u \in [v-i, v+i]\}$ and $\mathcal{S} = \{V_0 = v, V_T = w, R(V) = 1\}$.

We show that a valid exact algorithm can be constructed if it is possible to partition the proposal path space into layers, simulate unbiasedly to which layer a proposal sample path belongs and then, conditional on that layer, simulate a skeleton. Our exact algorithm framework for simulating skeletons is based on three principles for choosing a proposal measure and simulating a path space layer:

Principle 1 (Layer construction). *The path space of the process of interest, can be partitioned and the layer to which a proposal sample path belongs can be unbiasedly simulated,* $R(V) \sim \mathcal{R} := \mathbb{P}^v_{0,T} \circ R^{-1}$.

Principle 2 (Proposal exactness). *Conditional on* V_0, V_T *and* $R(V)$, *we can simulate any finite collection of intermediate points of the trajectory of the proposal diffusion exactly*, $V \sim \mathbb{P}^v_{0,T}|_{R^{-1}(R(V))}$.

Together Principles 1 and 2 ensure it is possible to simulate a skeleton. However, in addition we want to characterise the entire sample path and so we construct exact algorithms with the following additional principle.

Principle 3 (Path restoration). *Any finite collection of intermediate* (*inference*) *points*, *conditional on the skeleton*, *can be simulated exactly*, $V_{t_1}, \dots, V_{t_n} \sim \mathbb{P}^v_{0,T}|\mathcal{S}$.

In developing a methodological framework for simulating exact skeletons of (jump) diffusion sample paths we are able to present a number of significant extensions to the existing literature on exact algorithms [5,6,13,20]. In particular, we present novel exact algorithm constructions requiring the simulation of fewer points of the proposal sample path in order to evaluate whether to accept or reject (in effect a Rao–Blackwellisation of EA3 for diffusions [6], which we term the Unbounded Exact Algorithm (UEA), and a Rao–Blackwellisation of the Jump Exact Algorithms (JEAs) for jump diffusions [13,20], which we term the Bounded Jump Exact Algorithm (BJEA) and Unbounded Jump Exact Algorithm (UJEA), resp.) – all of which we recommend are adopted in practice instead of the existing equivalent exact algorithm. Furthermore, we extend both existing and novel exact algorithms to satisfy Principle 3, enabling the further simulation of a proposed sample path *after* it has been accepted, which hitherto has not been possible except under the limiting conditions imposed in EA1 [5].

Although in the context of a particular application we do not necessarily require the ability to further simulate a proposal sample path after acceptance (e.g., particle filtering for partially observed jump diffusions [31], in which only the end point of each sample path is required), to tackle the type of problem in which we do require that Principle 3 is satisfied (e.g., in the examples outlined Figure 1) we introduce a novel class of *Adaptive Exact Algorithms (AEAs)* for both diffusions and jump diffusions (which are again in effect a Rao–Blackwellisation of the Unbounded Exact Algorithm (UEA) requiring fewer points of the proposal sample path in order to evaluate whether to accept or reject).

By direct application of the methodology we develop for Adaptive Exact Algorithms (AEA), we significantly extend ε-*Strong Simulation* methodology [8] (which allows the simulation of upper and lower bounding processes which almost surely constrain stochastic process sample paths to any specified tolerance), from Brownian motion sample paths to a general class of jump diffusions, and introduce novel results to ensure the methodology in [8] can be implemented exactly. Finally, we highlight a number of possible applications of the methodology developed in this paper by returning to the examples introduced in Figure 1. We demonstrate that it is possible not only to simulate skeletons exactly from the correct target measure but also to evaluate exactly whether or not non-trivial barriers have been crossed and so construct Monte Carlo estimators for computing barrier crossing probabilities. It should be noted that there are a wide range of other possible direct applications of the methodology in this paper, for instance, the evaluation of path space integrals and barrier hitting times to arbitrary precision, among many others.

In summary, the main contributions of this paper are as follows:

- A mathematical framework for constructing exact algorithms for both diffusions and jump diffusions, enabling improvements to existing exact algorithms, extension of existing exact algorithms to satisfy Principle 3 and a new class of adaptive exact algorithms (see Sections 3 and 4).
- Methodology for the ε-strong simulation of diffusion and jump diffusion sample paths, along with a novel exact algorithm based on this construction (see Sections 5 and 8).
- New methodology for constructing Monte Carlo estimators to compute irregular barrier crossing probabilities, simulating first hitting times to any specified tolerance and simulating killed diffusion sample path skeletons (see Sections 5 and 9). This work is reliant on the methodological extensions of Sections 3–5 and is presented along with examples based on the illustrations in Figure 1.

We also make a number of other contributions which are necessary for the implementation of our methodology. In particular, we detail how to simulate unbiasedly events of probability corresponding to various Brownian path space probabilities (see Section 6); and, we make significant extensions to existing ε-strong methodology enabling the initialisation of the algorithm and ensuring exactness (see Sections 5 and 8).

This paper is organised as follows: in Section 2, we detail conditions sufficient to establish results necessary for applying the methodology in this paper. In Sections 3 and 4, we outline our exact algorithm framework for diffusions and jump diffusions, respectively, presenting the resulting UEA, Bounded Jump Exact Algorithm (BJEA), Unbounded Jump Exact Algorithm (UJEA) and AEAs. In Section 5, we apply our methodology enabling the ε-strong simulation of (jump) diffusions. We extend existing layered Brownian bridge constructions in Section 7, introducing novel constructions for the AEAs in Section 8 (both of which rely on novel Brownian path space simulation results which are summarised in Section 6). Finally in Section 9, we revisit the examples outlined in Figure 1 to which we apply our methodology.

2. Preliminaries

In order to apply the methodology in this paper, we require conditions in order to establish Results 1–4 which we introduce below. To present our work in some generality we assume Conditions 1–5 hold (see below) along with some indication of why each is required. However, these conditions can be difficult to check and so in Section 2.1 we discuss verifiable sufficient conditions under which Results 1–4 hold.

Condition 1 (Solutions). *The coefficients of* (1) *are sufficiently regular to ensure the existence of a unique*, *non-explosive*, *weak solution.*

Condition 2 (Continuity). *The drift coefficient* $\beta \in C^1$. *The volatility coefficient* $\sigma \in C^2$ *and is strictly positive.*

Condition 3 (Growth bound). *We have that* $\exists K > 0$ *such that* $|\beta(x)|^2 + |\sigma(x)|^2 \leq K(1 + |x|^2)$ $\forall x \in \mathbb{R}$.

Condition 4 (Jump rate). *λ is non-negative and locally bounded.*

Conditions 2 and 3 are sufficient to allow us to transform our SDE in (1) into one with unit volatility,

Result 1 (Lamperti transform [24], Chapter 4.4). *Let $\eta(V_t) =: X_t$ be a transformed process, where $\eta(V_t) := \int_{v^*}^{V_t} 1/\sigma(u)\,\mathrm{d}u$ (where v^* is an arbitrary element in the state space of V). We denote by $\psi_1, \dots, \psi_{N_T}$ as the jump times in the interval $[0, T]$, $\psi_0 := 0$, $\psi_{N_T+1}- := \psi_{N_T+1} := T$ and $N_t := \sum_{i\geq 1} \mathbb{1}\{\psi_i \leq t\}$ a Poisson jump counting process. Further denoting by V^{cts} the continuous component of V and applying Itô's formula for jump diffusions to find $\mathrm{d}X_t$ we have (where $\mu_t \sim f_\mu(\cdot; V_{t-}) = f_\mu(\cdot; \eta^{-1}(X_{t-}))$),*

$$\begin{aligned}\mathrm{d}X_t &= \bigl[\eta'\,\mathrm{d}V_t^{\mathrm{cts}} + \eta''\bigl(\mathrm{d}V_t^{\mathrm{cts}}\bigr)^2/2\bigr] + \bigl(\eta[V_{t-} + \mu_t] - \eta(V_{t-})\bigr)\,\mathrm{d}N_t \\ &= \underbrace{\left[\frac{\beta(\eta^{-1}(X_{t-}))}{\sigma(\eta^{-1}(X_{t-}))} - \frac{\sigma'(\eta^{-1}(X_{t-}))}{2}\right]}_{\alpha(X_{t-})}\mathrm{d}t + \mathrm{d}W_t + \underbrace{\bigl(\eta\bigl[\eta^{-1}(X_{t-}) + \mu_t\bigr] - X_{t-}\bigr)\,\mathrm{d}N_t}_{\mathrm{d}J_t^{\lambda,\nu}}.\end{aligned} \tag{5}$$

This transformation is typically possible for univariate diffusions and for many multivariate diffusions [1]. A significant class of multivariate diffusions can be simulated by direct application of our methodology, and ongoing work is aimed at extending these methodologies more broadly to multivariate diffusions (see [35]).

In the remainder of this paper, we assume (without loss of generality) that we are dealing with transformed SDEs with unit volatility coefficient as in (5). As such, we introduce the following simplifying notation. In particular, we denote by $\mathbb{Q}_{0,T}^x$ the measure induced by (5) and by $\mathbb{W}_{0,T}^x$ the measure induced by the driftless version of (5). We define $A(u) := \int_0^u \alpha(y)\,\mathrm{d}y$ and set $\phi(X_s) := \alpha^2(X_s)/2 + \alpha'(X_s)/2$. If $\lambda = 0$ in (5) then $\mathbb{W}_{0,T}^x$ is Wiener measure. Furthermore, we impose the following final condition.

Condition 5 (Φ). *There exists a constant $\Phi > -\infty$ such that $\Phi \leq \phi$.*

It is necessary for this paper to establish that the Radon–Nikodým derivative of $\mathbb{Q}_{0,T}^x$ with respect to $\mathbb{W}_{0,T}^x$ exists (Result 2) and can be bounded on compact sets (Results 3 and 4) under Conditions 1–5.

Result 2 (Radon–Nikodým derivative [28,30]). *Under Conditions 1–4, the Radon–Nikodým derivative of $\mathbb{Q}_{0,T}^x$ with respect to $\mathbb{W}_{0,T}^x$ exists and is given by Girsanov's formula*:

$$\frac{\mathrm{d}\mathbb{Q}_{0,T}^x}{\mathrm{d}\mathbb{W}_{0,T}^x}(X) = \exp\left\{\int_0^T \alpha(X_s)\,\mathrm{d}W_s - \frac{1}{2}\int_0^T \alpha^2(X_s)\,\mathrm{d}s\right\}. \tag{6}$$

As a consequence of Condition 2, we have $A \in C^2$ and so we can apply Itô's formula to remove the stochastic integral,

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_s)\,\mathrm{d}s - \sum_{i=1}^{N_T}\bigl[A(X_{\psi_i}) - A(X_{\psi_i -})\bigr]\right\}. \tag{7}$$

In the particular case where we have a diffusion ($\lambda = 0$),

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_s)\,\mathrm{d}s\right\}. \tag{8}$$

Result 3 (Quadratic growth). *As a consequence of Condition 3, we have that A has a quadratic growth bound and so there exists some $T_0 < \infty$ such that $\forall T \le T_0$:*

$$c(x,T) := \int_{\mathbb{R}} \exp\left\{A(y) - \frac{(y-x)^2}{2T}\right\}\mathrm{d}y < \infty. \tag{9}$$

Throughout this paper, we rely on the fact that upon simulating a path space layer (see Definition 2) then $\forall s \in [0,T]$ $\phi(X_s)$ is bounded, however this follows directly from the following result.

Result 4 (Local boundedness). *By Condition 2, α and α' are bounded on compact sets. In particular, suppose $\exists \ell, \upsilon \in \mathbb{R}$ such that $\forall t \in [0,T]$, $X_t(\omega) \in [\ell, \upsilon]$ $\exists L_X := L(X(\omega)) \in \mathbb{R}, U_X := U(X(\omega)) \in \mathbb{R}$ such that $\forall t \in [0,T]$, $\phi(X_t(\omega)) \in [L_X, U_X]$.*

2.1. Verifiable Sufficient Conditions

As discussed in [28], Theorem 1.19, to ensure Condition 1 it is sufficient to assume that the coefficients of (1) satisfy the following linear growth and Lipschitz continuity conditions for some constants $C_1, C_2 < \infty$ (recalling that f_μ is the density of the jump sizes),

$$\bigl|\beta(x)\bigr|^2 + \bigl|\sigma(x)\bigr|^2 + \int_{\mathbb{R}} \bigl|f_\mu(z;x)\bigr|^2 \lambda(\mathrm{d}z) \le C_1\bigl(1 + |x|^2\bigr), \qquad \forall x \in \mathbb{R}, \tag{10}$$

$$\begin{aligned} &\bigl|\beta(x) - \beta(y)\bigr|^2 + \bigl|\sigma(x) - \sigma(y)\bigr|^2 + \int_{\mathbb{R}} \bigl|f_\mu(z;x) - f_\mu(z;y)\bigr|^2 \lambda(\mathrm{d}z) \\ &\quad \le C_2|x-y|^2, \qquad \forall x, y \in \mathbb{R}. \end{aligned} \tag{11}$$

(10) and (11) together with Condition 2 are sufficient for the purposes of implementing the methodology in this paper (in particular, Conditions 1, 3, 4 and 5 will hold in this situation) but are not necessary. Although easy to verify, (10) and (11) are somewhat stronger than necessary for our purposes and so we impose Condition 1 instead.

It is of interest to note that if we have a diffusion (i.e., in (1) we have $\lambda = 0$), then, by application of the Mean Value Theorem, Condition 2 ensures β and σ are locally Lipschitz and so (1) admits a unique weak solution [27] and so Condition 1 holds. In particular, the methodology in this paper will hold under Conditions 2, 3 and 5.

3. Exact Simulation of Diffusions

In this section, we outline how to simulate skeletons for diffusion sample paths (we will return to jump diffusions in Section 4) which can be represented (under the conditions in Section 2 and following the transformation in (5)), as the solution to SDEs with unit volatility,

$$\mathrm{d}X_t = \alpha(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \qquad X_0 = x \in \mathbb{R}, t \in [0, T]. \tag{12}$$

As discussed in Section 1, exact algorithms are a class of rejection samplers operating on diffusion path space. In this section, we begin by reviewing rejection sampling and outline an (idealised) rejection sampler originally proposed in [9] for simulating entire diffusion sample paths. However, for computational reasons this idealised rejection sampler can't be implemented so instead, with the aid of new results and algorithmic step reordering, we address this issue and construct a rejection sampler for simulating sample path skeletons which only requires finite computation. A number of existing exact algorithms exist based on this approach [5,6,9], however, in this paper we present two novel algorithmic interpretations of this rejection sampler. In Section 3.1, we present the Unbounded Exact Algorithm (UEA), which is a methodological extension of existing exact algorithms [6], requiring less of the proposed sample path to be simulated in order to evaluate acceptance or rejection. Finally, in Section 3.2 we introduce the novel Adaptive Unbounded Exact Algorithm (AUEA), which as noted in the Introduction, is well suited to problems in which further simulation of a proposed sample path *after* it has been accepted is required.

Rejection sampling [34] is a standard Monte Carlo method in which we can sample from some (inaccessible) target distribution π by means of an accessible dominating distribution g with respect to which π is absolutely continuous with bounded Radon–Nikodým derivative. In particular, if we can find a bound M such that $\sup_x \frac{\mathrm{d}\pi}{\mathrm{d}g}(x) \le M < \infty$, then drawing $X \sim g$ and accepting the draw (setting $I = 1$) with probability $P_g(X) := \frac{1}{M}\frac{\mathrm{d}\pi}{\mathrm{d}g}(X) \in [0, 1]$ then $(X|I = 1) \sim \pi$.

Similarly, we could simulate sample paths from our target measure (the measure induced by (12) and denoted $\mathbb{Q}^x_{0,T}$) by means of a proposal measure which we can simulate proposal sample paths from, provided a bound for the Radon–Nikodým derivative can be found. A natural equivalent measure to choose as a proposal is Wiener measure ($\mathbb{W}^x_{0,T}$, as (12) has unit volatility). In particular, drawing $X \sim \mathbb{W}^x_{0,T}$ and accepting the sample path ($I = 1$) with probability $P_{\mathbb{W}^x_{0,T}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) \in [0, 1]$ (where $\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X)$ is as given in (8)) then $(X|I = 1) \sim \mathbb{Q}^x_{0,T}$. On average sample paths would be accepted with probability $P_{\mathbb{W}^x_{0,T}} := \mathbb{E}_{\mathbb{W}^x_{0,T}}[P_{\mathbb{W}^x_{0,T}}(X)]$. However, the function $A(X_T)$ in (8) only has a quadratic growth bound (see Result 3), so typically no appropriate bound ($M < \infty$) exists.

To remove the unbounded function $A(X_T)$ from the acceptance probability, one can use Biased Brownian motion (BBM) [9] as the proposal measure and consider the resulting modification to the acceptance probability.

Definition 3. *Biased Brownian motion is the process* $Z_t \overset{\mathcal{D}}{=} (W_t | W_0 = x, W_T := y \sim h)$ *with measure* $\mathbb{Z}^x_{0,T}$, *where* $x, y \in \mathbb{R}$, $t \in [0, T]$ *and* h *is defined as (by Result 3 we have* $\forall T \leq T_0$, $h(y; x, T)$ *is integrable),*

$$h(y; x, T) := \frac{1}{c(x,T)} \exp\left\{ A(y) - \frac{(y-x)^2}{2T} \right\}. \tag{13}$$

Theorem 1 (Biased Brownian motion [9], Proposition 3). $\mathbb{Q}^x_{0,T}$ *is equivalent to* $\mathbb{Z}^x_{0,T}$ *with Radon–Nikodým derivative*:

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{Z}^x_{0,T}}(X) \propto \exp\left\{ -\int_0^T \phi(X_s)\,\mathrm{d}s \right\}. \tag{14}$$

Sample paths can be drawn from $\mathbb{Z}^x_{0,T}$ in two steps by first simulating the end point $X_T =: y \sim h$ (although h doesn't have a tractable form, a rejection sampler with Gaussian proposal can typically be constructed) and then simulating the remainder of the sample path in $(0, T)$ from the law of a Brownian bridge, $(X | X_0 = x, X_T = y) \sim \mathbb{W}^{x,y}_{0,T}$. We can now construct an (idealised) rejection sampler to draw sample paths from $\mathbb{Q}^x_{0,T}$ as outlined in Algorithm 1, noting that as $\inf_{u \in [0,T]} \phi(X_u) \geq \Phi$ (see Condition 5) we can identify Φ and choose $M := \exp\{-\Phi T\}$ to ensure $P_{\mathbb{Z}^x_{0,T}}(X) \in [0, 1]$.

Unfortunately, Algorithm 1 can't be implemented directly as it isn't possible to draw entire sample paths from $\mathbb{W}^{x,y}_{0,T}$ in Step 1(b) (they're infinite dimensional random variables) and it isn't possible to evaluate the integral expression in the acceptance probability in Step 2.

The key to constructing an implementable algorithm (which requires only finite computation), is to note that by first simulating some finite dimensional auxiliary random variable $F \sim \mathbb{F}$ (the details of which are in Sections 3.1 and 3.2), an unbiased estimator of the acceptance probability can be constructed which can be evaluated using only a finite dimensional subset of the proposal sample path. As such, we can use the simulation of F to inform us as to what finite dimensional subset of the proposal sample path to simulate ($X^{\mathrm{fin}} \sim \mathbb{W}^{x,y}_{0,T} | F$) in Step 1(b) in order to evaluate the acceptance probability. The rest of the sample path can be simulated as required *after* the acceptance of the sample path from the proposal measure conditional on the simulations performed, $X^{\mathrm{rem}} \sim \mathbb{W}^{x,y}_{0,T} | (X^{\mathrm{fin}}, F)$ (noting that $X = X^{\mathrm{fin}} \cup X^{\mathrm{rem}}$). The synthesis of this argument is presented in Algorithm 2.

Note that Algorithm 2 Step 5 is separated from the rest of the algorithm and asterisked. This convention is used within this paper to indicate the final step within an exact algorithm, which

Algorithm 1 Idealised Rejection Sampler [9]

```
1. Simulate $X \sim \mathbb{Z}^x_{0,T}$,
    (a) Simulate $X_T =: y \sim h$.
    (b) Simulate $X_{(0,T)} \sim \mathbb{W}^{x,y}_{0,T}$.
2. With probability $P_{\mathbb{Z}^x_{0,T}}(X) = \exp\{-\int_0^T \phi(X_s)\,\mathrm{d}s\} \cdot \exp\{\Phi T\}$ accept, else reject and return
   to Step 1.
```

Algorithm 2 Implementable Exact Algorithm [5,9]

```
1. Simulate $X_T =: y \sim h$.
2. Simulate $F \sim \mathbb{F}$.
3. Simulate $X^{\text{fin}} \sim \mathbb{W}^{x,y}_{0,T}|F$.
4. With probability $P_{\mathbb{Z}^x_{0,T}|F}(X)$ accept, else reject and return to Step 1.
5. *Simulate $X^{\text{rem}} \sim \mathbb{W}^{x,y}_{0,T}|(X^{\text{fin}}, F)$.
```

cannot be conducted in its entirety as it involves simulating an infinite dimensional random variable. However, as noted in the introductory remarks to this section, our objective is to simulate a finite dimensional sample path skeleton, with which we can simulate the accepted sample path at any other finite collection of time points without error. This final step simply indicates how this subsequent simulation may be conducted.

In conclusion, although it isn't possible to simulate entire sample paths from $\mathbb{Q}^x_{0,T}$, it is possible to simulate exactly a finite dimensional subset of the sample path, characterised by its *skeleton* $\mathcal{S}(X) := \{X_0, X^{\text{fin}}, X_T, F\}$. Careful consideration has to be taken to construct $\mathbb{F}$ which existing exact algorithms [5,6,9] achieve by applying Principles 1 and 2. However, no existing exact algorithm addresses how to construct $\mathbb{F}$ under the conditions in Section 2 to additionally perform Algorithm 2 Step 5. We address this in Sections 3.1 and 3.2.

In the next two sections, we present two distinct, novel interpretations of Algorithm 2. In Section 3.1, we present the UEA which is a methodological extension of existing exact algorithms and direct interpretation of Algorithm 2. In Section 3.2, we introduce the AUEA which takes a recursive approach to Algorithm 2 Steps 2, 3 and 4.

3.1. Bounded and Unbounded Exact Algorithms

In this section, we present the Unbounded Exact Algorithm (UEA) along with the Bounded Exact Algorithm (BEA) (which can viewed as a special case of the UEA) by revisiting Algorithm 2 and considering how to construct a suitable finite dimensional random variable $F \sim \mathbb{F}$.

As first noted in [5], it is possible to construct and simulate the random variable F required in Algorithm 2, provided $\phi(X_{[0,T]})$ can be bounded above and below. It was further noted in [6] that if a Brownian bridge proposal sample path was simulated along with an interval in which is was contained, and that conditional on this interval $\phi(X_{[0,T]})$ was bounded above and below, then F could similarly be constructed and simulated. Finding a suitable set of information that establishes an interval in which $\phi(X_{[0,T]})$ is contained (by means of finding and mapping an interval in which the sample path $X_{[0,T]}$ is contained), is the primary motivation behind the notion of a sample path layer (see Definition 2). In this paper, we discuss more than one layer construction (see Sections 7 and 8), both of which complicate the key ideas behind the UEA and so layers are only discussed in abstract terms at this stage.

Further to [6], we note that $\phi(X_{[0,T]})$ is bounded on compact sets (see Result 4) and so if, after simulating the end point from Biased Brownian motion (BBM), we partition the path space of $\mathbb{Z}^x|X_T$ into disjoint layers and simulate the layer to which our proposal sample path belongs (see Principle 1, denoting $R := R(X) \sim \mathcal{R}$ as the simulated layer the precise details of which are given in Section 7), then an upper and lower bound for $\phi(X_{[0,T]})$ can always be found conditional on this layer ($U_X \in \mathbb{R}$ and $L_X \in \mathbb{R}$, resp.). As such, we have for all test functions $H \in \mathcal{C}_{\mathrm{b}}$,

$$\begin{aligned}\mathbb{E}_{\mathbb{Z}^x_{0,T}}\big[P_{\mathbb{Z}^x_{0,T}}(X)\cdot H(X)\big] &= \mathbb{E}_h\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}}\big[P_{\mathbb{Z}^x_{0,T}}(X)\cdot H(X)\big] \\ &= \mathbb{E}_h\mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|R}\big[P_{\mathbb{Z}^x_{0,T}}(X)\cdot H(X)\big],\end{aligned} \tag{15}$$

recalling that,

$$P_{\mathbb{Z}^x_{0,T}}(X) = \exp\left\{-\int_0^T \phi(X_s)\,\mathrm{d}s\right\}\cdot \mathrm{e}^{\Phi T}. \tag{16}$$

Proceeding in a similar manner to [7] to construct our finite dimensional estimator we consider a Taylor series expansion of the exponential function in (16),

$$\begin{aligned}P_{\mathbb{Z}^x_{0,T}}(X) &= \mathrm{e}^{-(L_X-\Phi)T}\cdot \mathrm{e}^{-(U_X-L_X)T}\exp\left\{\int_0^T U_X-\phi(X_s)\,\mathrm{d}s\right\} \\ &= \mathrm{e}^{-(L_X-\Phi)T}\cdot\left[\sum_{j=0}^{\infty}\frac{\mathrm{e}^{-(U_X-L_X)T}[(U_X-L_X)T]^j}{j!}\left\{\int_0^T\frac{U_X-\phi(X_s)}{(U_X-L_X)T}\,\mathrm{d}s\right\}^j\right],\end{aligned} \tag{17}$$

again employing methods found in [7], we note that if we let $\mathbb{K}_R$ be the law of $\kappa \sim \mathrm{Poi}((U_X - L_X)T)$, $\mathbb{U}\kappa$ the distribution of $(\xi_1,\dots,\xi_\kappa) \overset{\mathrm{iid}}{\sim} \mathrm{U}[0,T]$ we have,

$$\begin{aligned}P_{\mathbb{Z}^x_{0,T}}(X) &= \mathrm{e}^{-(L_X-\Phi)T}\cdot\mathbb{E}_{\mathbb{K}_R}\left[\left(\int_0^T\frac{U_X-\phi(X_s)}{(U_X-L_X)T}\,\mathrm{d}s\right)^\kappa \Big| X\right] \\ &= \mathrm{e}^{-(L_X-\Phi)T}\cdot\mathbb{E}_{\mathbb{K}_R}\left[\mathbb{E}_{\mathbb{U}\kappa}\left[\prod_{i=1}^{\kappa}\left(\frac{U_X-\phi(X_{\xi_i})}{U_X-L_X}\right)\Big| X\right]\Big| X\right].\end{aligned} \tag{18}$$

The key observation to make from (18) is that the acceptance probability of a sample path $X \sim \mathbb{Z}^x_{0,T}$ can be evaluated without using the entire sample path, and can instead be evaluated using a finite dimensional realisation, X^{fin}. Simulating a finite dimensional proposal as suggested by (18) and incorporating it within Algorithm 2 results directly in the UEA presented in Algorithm 3. A number of alternate methods for simulating unbiasedly layer information (Step 2), layered Brownian bridges (Step 4), and the sample path at further times after acceptance (Step 6), are given in Section 7.

The UEA can be viewed as a multi-step rejection sampler in which the acceptance probability is broken into a computational inexpensive step (Step 3), and a computationally expensive step (Step 5) which to evaluate requires partial simulation of the proposal sample path (Step 4). Unlike existing exact algorithms (EA3 in [6]), the UEA conducts early rejection to avoid any further

Algorithm 3 Unbounded Exact Algorithm (UEA)

```
1. Simulate skeleton end point $X_T =: y \sim h$.
2. Simulate layer information $R \sim \mathcal{R}$.
3. With probability $(1 - \exp\{-(L_X - \Phi)T\})$ reject path and return to Step 1.
4. Simulate skeleton points $(X_{\xi_1}, \dots, X_{\xi_\kappa})|R$,
    (a) Simulate $\kappa \sim \mathrm{Poi}((U_X - L_X)T)$ and skeleton times $\xi_1, \dots, \xi_\kappa \overset{\text{iid}}{\sim} \mathrm{U}[0, T]$.
    (b) Simulate sample path at skeleton times $X_{\xi_1}, \dots, X_{\xi_\kappa} \sim \mathbb{W}^{x,y}_{0,T}|R$.
5. With probability $\prod_{i=1}^{\kappa}[(U_X - \phi(X_{\xi_i}))/(U_X - L_X)]$, accept entire path, else reject and
   return to Step 1.
-----------------------------------------------------------------
6. *Simulate $X^{\text{rem}} \sim (\bigotimes_{i=1}^{\kappa+1} \mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i})|R$.
```

unnecessary simulation of the rejected sample path. In particular, the UEA requires simulation of fewer points of the proposal sample path in order to evaluate whether to accept or reject.

The skeleton of an accepted sample path includes any information simulated for the purpose of evaluating the acceptance probability (any subsequent simulation must be consistent with the skeleton). As such, the skeleton is composed of terminal points, skeletal points $(X_{\xi_1}, \dots, X_{\xi_\kappa})$ and layer R (denoting $\xi_0 := 0$ and $\xi_{\kappa+1} := T$),

$$\mathcal{S}_{\text{UEA}}(X) := \left\{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, R\right\}. \tag{19}$$

After simulating an accepted sample path skeleton, we may want to simulate the sample path at further intermediate points. In the particular case in which $\phi(X_{[0,T]})$ is almost surely bounded, there is no need to simulate layer information in Algorithm 3, the skeleton can be simulated from the law of a Brownian bridge and given the skeleton we can simulate further intermediate points of the sample path from the law of a Brownian bridge (so we satisfy Principle 3). This leads to the *Exact Algorithm* 1 (EA1) proposed in [5], which we term the BEA,

$$\mathcal{S}_{\text{BEA}}(X) := \left\{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}\right\}. \tag{20}$$

A second exact algorithm (EA2) was also proposed in [5] (the details of which we omit from this paper), in which simulating the sample path at further intermediate points after accepting the sample path skeleton was possible by simulating from the law of two independent Bessel bridges. However, EA1 (BEA) and EA2 both have very limited applicability and are the only existing exact algorithms which directly satisfy Principle 3.

Unlike existing exact algorithms [5,6,9], after accepting a sample path skeleton using the UEA, it is possible to simulate the sample path at further finite collections of time points without approximation under the full generality of the conditions outlined in Section 2 (so satisfying Principle 3). Algorithm 3 Step 6 can't be conducted in existing exact algorithms as the layer imparts information across the entire interval. However, in Section 7 we show that Step 6 is possible (with additional computation), by augmenting the skeleton with sub-interval layer information (denoting $R_X^{[a,b]}$ as the layer for the sub-interval $[a,b]\subseteq[0,T]$),

$$\begin{aligned}\mathcal{S}'_{\mathrm{UEA}}(X) &:= \underbrace{\left\{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, R, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1}\right\}}_{A} \\ &\equiv \underbrace{\left\{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1}\right\}}_{B}.\end{aligned} \tag{21}$$

The augmented skeleton allows the sample path to be decomposed into conditionally independent paths between each of the skeletal points and so the layer R no longer imparts information across the entire interval $[0,T]$. In particular, the sets in (21) are equivalent in the sense that $\mathbb{W}_{0,T}^{x,y}|A = \mathbb{W}_{0,T}^{x,y}|B$. As such, simulating the sample path at further times after acceptance as in Algorithm 3 Step 6 is direct,

$$X^{\mathrm{rem}} \sim \mathbb{W}_{0,T}^{x,y}|\mathcal{S}'_{\mathrm{UEA}} = \bigotimes_{i=1}^{\kappa+1}\left(\mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}}|R_X^{[\xi_{i-1},\xi_i]}\right). \tag{22}$$

3.1.1. *Implementational Considerations – Interval Length*

It transpires that the computational cost of simulating a sample path scales worse than linearly with interval length. However, this scaling problem can be addressed by exploiting the fact that sample paths can be simulated by successive simulation of sample paths of shorter length over the required interval by applying the strong Markov property, noting the Radon–Nikodým derivative in (8) decomposes as follows (for any $t\in[0,T]$),

$$\begin{aligned}\frac{\mathrm{d}\mathbb{Q}_{0,T}^{x}}{\mathrm{d}\mathbb{W}_{0,T}^{x}}(X) &= \exp\left\{A(X_t) - A(x) - \int_0^t \phi(X_s)\,\mathrm{d}s\right\} \\ &\quad\times \exp\left\{A(X_T) - A(X_t) - \int_t^T \phi(X_s)\,\mathrm{d}s\right\}.\end{aligned} \tag{23}$$

3.2. Adaptive Unbounded Exact Algorithm

Within this section, we outline a novel Adaptive Unbounded Exact Algorithm (AUEA). To motivate this, we revisit Algorithm 2 noting that the acceptance probability (16) of a sample path $X\sim\mathbb{Z}_{0,T}^{x}$ can be rewritten as follows,

$$P_{\mathbb{Z}_{0,T}^{x}}(X) = \exp\left\{-\int_0^T \big(\phi(X_s) - L_X\big)\,\mathrm{d}s\right\}\cdot \mathrm{e}^{-(L_X-\Phi)T} =: \tilde{P}_{\mathbb{Z}_{0,T}^{x}|R}(X)\cdot \mathrm{e}^{-(L_X-\Phi)T}. \tag{24}$$

Now following Algorithm 3, after simulating layer information (Step 2) and conditionally accepting the proposal sample path in the first (inexpensive) part of the multi-step rejection sampler (Step 3) the probability of accepting the sample path is,

$$\tilde{P}_{\mathbb{Z}_{0,T}^{x}|R}(X) \in \left[\mathrm{e}^{-(U_X-L_X)T}, 1\right] \subseteq (0,1]. \tag{25}$$

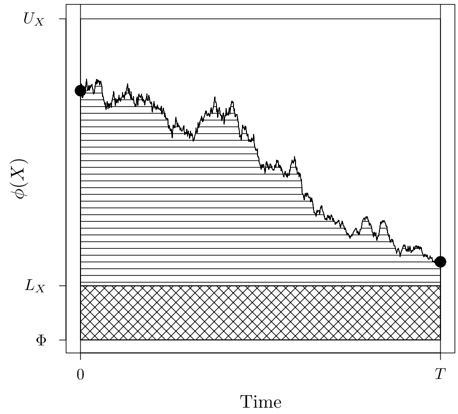


Figure 3. Example trajectory of $\phi(X)$ where $X \sim \mathbb{W}_{0,T}^{x,y}|R(X)$.

Reinterpreting the estimator in (18) in light of (25) and with the aid of Figure 3, we are exploiting the fact that $\tilde{P}_{\mathbb{Z}_{0,T}^{x}|R}(X)$ is precisely the probability a Poisson process of intensity 1 on the graph $\mathcal{G}_A := \{(x,y) \in [0,T] \times [L_X, \infty)\colon\ y \leq \phi(x)\}$ contains no points. As this is a difficult set in which to simulate a Poisson process (we don't even know the entire trajectory of X), we are instead simulating a Poisson process of intensity 1 on the larger graph $\mathcal{G}_P := [0,T] \times [L_X, U_X] \supseteq \mathcal{G}_A$ (which is easier as $U_X - L_X$ is a constant), and then conducting Poisson thinning by first computing $\phi(X)$ at the finite collection of time points of the simulated Poisson process and then determining whether or not any of the points lie in $\mathcal{G}_A$ (accepting the entire sample path if there are no Poisson points in $\mathcal{G}_A \subseteq \mathcal{G}_P$). This idea was first presented in [5] and formed the basis of the Bounded Exact Algorithm (BEA) discussed in Section 3.1.

As an aside, it should be noted that conditional acceptance of the proposal sample path with probability $\mathrm{e}^{-(L_X - \Phi)T}$ in Algorithm 3 Step 3 is simply the probability that a Poisson process of intensity 1 has no points on the graph $\mathcal{G}_R := [0,T] \times [\Phi, L_X]$ (the crosshatched region in Figure 3).

In some settings $\mathcal{G}_P$ can be much larger than $\mathcal{G}_A$ and the resulting exact algorithm can be inefficient and computationally expensive. In this section, we propose an *adaptive* scheme which exploits the simulation of intermediate skeletal points of the proposal sample path in Algorithm 3 Step 4. In particular, note that each simulated skeletal point implicitly provides information regarding the layer the sample path is contained within in both the sub-interval before and after it. As such, by simulating each point separately we can use this information to construct a modified bounding region $\mathcal{G}_P^M$ such that $\mathcal{G}_A \subseteq \mathcal{G}_P^M \subseteq \mathcal{G}_P$, composed of a Poisson process with piecewise constant intensity, for the simulation of the remaining points.

In Algorithm 3 Step 4(a), we simulate a Poisson process of intensity $\Delta_X T := (U_X - L_X)T$ on the interval $[0,T]$ to determine the skeletal points $(\xi_1, \dots, \xi_\kappa)$. Alternatively we can exploit the *exponential waiting time* property between successive events [23]. In particular, denoting $T_1, \dots, T_\kappa$ as the time between each event $\xi_1, \dots, \xi_\kappa$, then the timing of the events can be simulated by successive $\mathrm{Exp}(\Delta_X)$ waiting times while $\sum_i T_i \leq T$.

The independence of arrival times of the points of a homogeneous Poisson process allows us to simulate them in any convenient order. In our case, it is likely the sample path at points closer to the mid-point of the interval will contain more information about the layer structure of the entire sample path. As such, there is an advantage in simulating these points first. If we begin at the interval mid-point ($T/2$), we can find the skeletal point closest to it by simulating an $\text{Exp}(2\Delta_X)$ random variable, τ (we are simulating the first point at *either* side of the mid-point). As such, the simulated point (denoted ξ) will be with equal probability at either $T/2-\tau$ or $T/2+\tau$. Considering this in the context of (25), upon simulating ξ we have simply broken the acceptance probability into the product of three probabilities associated with three disjoint sub-intervals, the realisation of the sample path at X_ξ providing a binary unbiased estimate of the probability corresponding to the central sub-interval (where the expectation is with respect to $u \sim \text{U}[0,1]$),

$$\begin{aligned}\tilde{P}_{\mathbb{Z}^x_{0,T}|R,X_\xi}(X) = \exp\biggl\{-\int_0^{T/2-\tau}\bigl[\phi(X_s)-L_X\bigr]\,\mathrm{d}s - \int_{T/2+\tau}^{T}\bigl[\phi(X_s)-L_X\bigr]\,\mathrm{d}s\biggr\} \\ \times\, \mathbb{E}\biggl(\mathbb{1}\biggl[u \le \frac{U_X-\phi(X_\xi)}{U_X-L_X}\biggr]\bigg|X_\xi\biggr).\end{aligned} \tag{26}$$

If the central sub-interval is rejected, the entire sample path can be discarded. However, if it is accepted then the acceptance of the entire sample path is conditional on the acceptance of *both* the left- and right-hand sub-intervals in (26), each of which has the same structural form as we originally had in (25). As such, for each we can simply iterate the above process until we have exhausted the entire interval $[0,T]$.

As outlined above, our approach is an algorithmic reinterpretation, but otherwise identical, to Algorithm 3. However, we now have the flexibility to exploit the simulated skeletal point X_ξ, to simulate new layer information for the remaining sub-intervals conditional on the existing layer R_X (which we detail in Section 8). In particular, considering the left-hand sub-interval in (26), we can find new layer information (denoted $R^{[0,\xi]}_X$) which will contain tighter bound information regarding the sample path ($\ell_X \le \ell^{[0,\xi]}_X \le X_{[0,\xi]}(\omega) \le \upsilon^{[0,\xi]}_X \le \upsilon_X$) and so (as a consequence of Result 4) can be used to compute tighter bounds for $\phi(X_{[0,\xi]})$ (denoted $U^{[0,\xi]}_X(\le U_X)$ and $L^{[0,\xi]}_X(\ge L_X)$),

$$\begin{aligned}&\tilde{P}^{[0,\xi]}_{\mathbb{Z}^x_{0,T}|R^{[0,\xi]}_X,X_\xi}(X) \\ &\quad= \exp\biggl\{-\int_0^{T/2-\tau}\bigl[\phi(X_s)-L_X\bigr]\,\mathrm{d}s\biggr\} \\ &\quad= \exp\biggl\{-\bigl(L^{[0,\xi]}_X-L_X\bigr)\cdot\biggl(\frac{T}{2}-\tau\biggr)\biggr\}\cdot\exp\biggl\{-\int_0^{T/2-\tau}\bigl[\phi(X_s)-L^{[0,\xi]}_X\bigr]\,\mathrm{d}s\biggr\}.\end{aligned} \tag{27}$$

The left-hand exponential in (27) is a constant and it is trivial to immediately reject the entire path with the complement of this probability. The right-hand exponential of (27) has the same form as (25) and so the same approach as outlined above can be employed, but over the shorter interval $[0,T/2-\tau]$ and with the lower rate $\Delta^{[0,\xi]}_X(:=U^{[0,\xi]}_X-L^{[0,\xi]}_X \le \Delta_X)$. As a consequence, the expected number of intermediary points required in order to evaluate the acceptance probability in (25) is lower than the Unbounded Exact Algorithm (UEA) in Algorithm 3.

Algorithm 4 Adaptive Unbounded Exact Algorithm (AUEA)

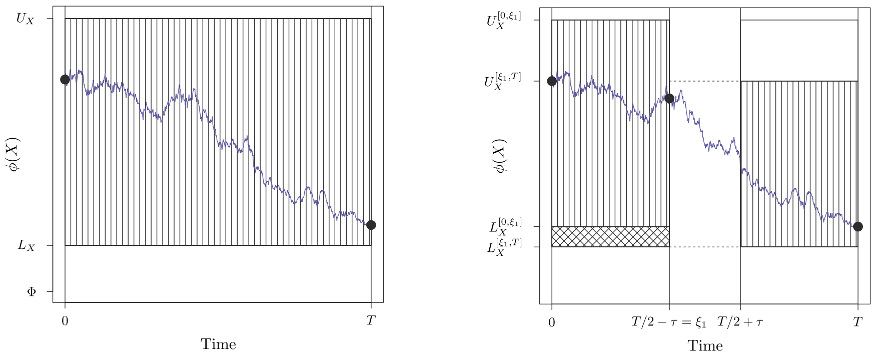
```
1. Simulate skeleton end point $X_T =: y \sim h$.
2. Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\Pi := \{\Xi\} := \{\{[0, T], X_0, X_T, R_X\}\}$ and $\kappa = 0$.
3. With probability $(1 - \exp\{-(L_X - \Phi)T\})$ reject path and return to Step 1.
4. While $|\Pi| \neq 0$,
   (a) Set $\Xi = \Pi(1)$.
   (b) Simulate $\tau \sim \mathrm{Exp}(2\Delta^{\Xi}_X)$. If $\tau > d(\Xi)$ then set $\Pi := \Pi \setminus \Xi$ else,
       i. Set $\kappa = \kappa + 1$ and with probability $1/2$ set $\xi'_\kappa = m(\Xi) - \tau$ else $\xi'_\kappa = m(\Xi) + \tau$.
       ii. Simulate $X_{\xi'_\kappa} \sim \mathbb{W}^{x(\Xi), y(\Xi)}_{\overleftarrow{s}(\Xi), \overrightarrow{t}(\Xi)} | R^{\Xi}_X$.
       iii. With probability $(1 - [U^{\Xi}_X - \phi(X_{\xi'_\kappa})]/\Delta^{\Xi}_X)$ reject path and return to Step 1.
       iv. Simulate new layer information $R^{[\overleftarrow{s}(\Xi), \xi'_\kappa]}_X$ and $R^{[\xi'_\kappa, \overrightarrow{t}(\Xi)]}_X$ conditional on $R^{\Xi}_X$.
       v. With probability $(1 - \exp\{-[L^{[\overleftarrow{s}(\Xi), \xi'_\kappa]}_X + L^{[\xi'_\kappa, \overrightarrow{t}(\Xi)]}_X - 2L^{\Xi}_X][d(\Xi) - \tau]\})$ reject path and return to Step 1.
       vi. Set $\Pi := \Pi \cup \{[s(\Xi), m(\Xi) - \tau], X^{\Xi}_{\overleftarrow{s}}, X_{\xi'_\kappa}, R^{[\overleftarrow{s}(\Xi), \xi'_\kappa]}_X\} \cup \{[m(\Xi) + \tau, t(\Xi)], X_{\xi'_\kappa}, X^{\Xi}_{\overrightarrow{t}}, R^{[\xi'_\kappa, \overrightarrow{t}(\Xi)]}_X\} \setminus \Xi$.
5. Define skeletal points $\xi_1, \dots, \xi_\kappa$ as the order statistics of the set $\{\xi'_1, \dots, \xi'_\kappa\}$.
---
6. *Simulate $X^{\mathrm{rem}} \sim (\bigotimes_{i=1}^{\kappa+1} \mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1}, \xi_i} | R^{[\xi_{i-1}, \xi_i]}_X)$.
```

This leads to the novel AUEA detailed in Algorithm 4, the recursive nature of the algorithm being illustrated in Figure 4 which is an extension to the example in Figure 3. We outline how to simulate (unbiasedly) layer information (Step 2), intermediate skeletal points (Step 4(b)ii) and new layer information (Step 4(b)iv) in a variety of ways in Section 8. Our iterative scheme outputs a skeleton comprising skeletal points and layer information for the intervals between consecutive skeletal points,

$$\mathcal{S}_{\mathrm{AUEA}}(X) := \{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, (R^{[\xi_{i-1}, \xi_i]}_X)_{i=1}^{\kappa+1}\}. \tag{28}$$

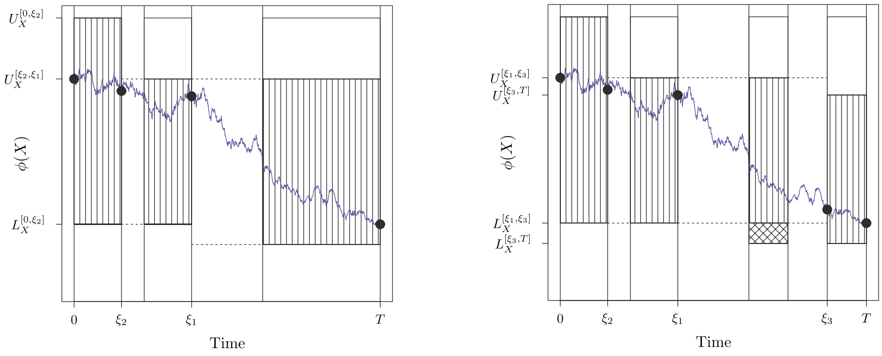
The AUEA with this skeleton has the distinct advantage that Principles 1, 2 and 3 are satisfied directly. In particular, any finite collection of intermediate points required after the skeleton has been simulated can be simulated directly (by application of Algorithm 4 Step 4(b)ii and Step 4(b)iv), without any augmentation of the skeleton (as in Algorithm 3). If necessary, further refinement of the layers given the additionally simulated points can be performed as outlined in Section 8.

It was noted in Section 3.1 that the skeleton from the UEA (Algorithm 3) could be augmented so that it satisfies Principle 3. However, the augmentation requires the application of the same methodological steps as that developed for the AUEA (see Section 8), but without the advantage of reducing the number of points required in the proposal sample path in order to evaluate whether to accept or reject. As such the AUEA can be viewed as a Rao–Blackwellised version of the UEA which should be adopted whenever further simulation of the proposal sample path is



(a) After preliminary acceptance (Algorithm 4 Step 3)

(b) After simulating ξ_1 (Algorithm 4 Step 4)



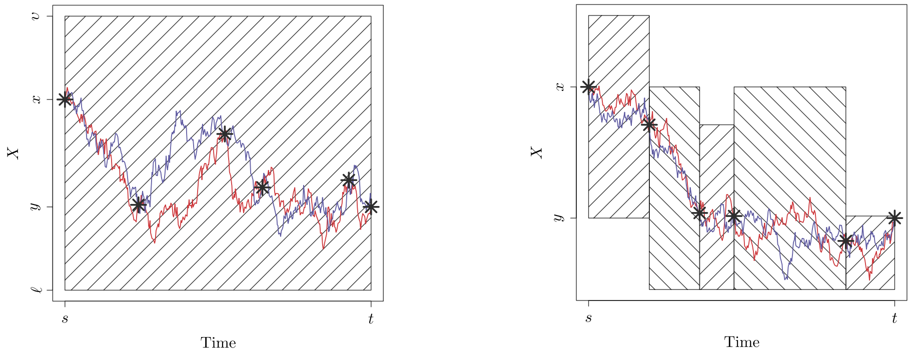
(c) After simulating ξ_2

(d) After simulating ξ_3

Figure 4. AUEA applied to the trajectory of $\phi(X)$ in Figure 3 (where $X \sim \mathbb{W}^{x,y}_{0,T}|R(X)$).

required after acceptance (e.g., in the settings outlined in Sections 5 and 9). To emphasise this, in Figure 5 we contrast the skeleton output from the UEA (Algorithm 3, prior to augmentation) and the AUEA (Algorithm 4).

In Algorithm 4, we introduce simplifying notation, motivated by the algorithm's recursive nature in which (as shown in (26)) the acceptance probability is iteratively decomposed over intervals which have been estimated and are yet to be estimated. Π denotes the set comprising information required to evaluate the acceptance probability for each of the intervals still to be estimated, $\Pi := \{\Pi(i)\}_{i=1}^{|\Pi|}$. Each $\Pi(i)$ contains information regarding the time interval it applies to, the sample path at known points at either side of this interval (which *do not* necessarily align with the end points of the sub-intervals corresponding to the remaining probabilities requiring simulation (as illustrated in Figure 4)) and the associated simulated layer information, which we denote $[s(\Pi(i)), t(\Pi(i))]$, $x(\Pi(i)) := X^{\Pi(i)}_{\overleftarrow{s}}$, $y(\Pi(i)) := X^{\Pi(i)}_{\overrightarrow{t}}$ and $R^{\Pi(i)}_X$, respectively (where



(a) Example sample path skeleton output from the Unbounded Exact Algorithm (UEA; Algorithm 3), $\mathcal{S}_{\mathrm{UEA}}(X)$, overlaid with two possible example sample path trajectories consistent with the skeleton

(b) Example sample path skeleton output from the Adaptive Unbounded Exact Algorithm (AUEA; Algorithm 4), $\mathcal{S}_{\mathrm{AUEA}}(X)$, overlaid with two possible example sample path trajectories consistent with the skeleton

Figure 5. Comparison of UEA and AUEA skeleton output. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

$\overleftarrow{s}(\Pi(i)) \leq s(\Pi(i)) < t(\Pi(i)) \leq \overrightarrow{t}(\Pi(i)))$. As before, $R_X^{\Pi(i)}$ can be used to directly compute bounds for ϕ for this specific sample path over the interval $[s(\Pi(i)), t(\Pi(i))]$ (namely $L_X^{\Pi(i)}$, $U_X^{\Pi(i)}$ and $\Delta_X^{\Pi(i)}$). We further denote $m(\Pi(i)) := (s(\Pi(i)) + t(\Pi(i)))/2$, $d(\Pi(i)) := (t(\Pi(i)) - s(\Pi(i)))/2$ and $\Xi := \Pi(1)$.

3.2.1. *Implementational Considerations – Known Intermediate Points*

It should be noted that if a number of intermediate points of a sample path are required and the time points at which they occur are known in advance, then rather than simulating them after the acceptance of the sample path skeleton in Algorithm 4 Step 6, their simulation can be incorporated into Algorithm 4. In particular, if these points are simulated immediately after Algorithm 4 Step 3 (this can be performed using Algorithm 22 as detailed in Section 8.5), then we have additional layer information regarding the sample path which can be used to compute tighter bounds for $\phi(X_{[0,T]})$ leading to a more efficient algorithm (as in Section 3.2). A drawback of this approach is that these additional points of the sample path constitute part of the resulting skeleton.

4. Exact Simulation of Jump Diffusions

In this section, we extend the methodology of Section 3, constructing exact algorithms for simulating skeletons of *jump diffusion* sample paths which can be represented as the solution to the following SDE,

$$\mathrm{d}X_t = \alpha(X_{t-})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \qquad X_0 = x \in \mathbb{R}, t \in [0, T]. \tag{29}$$

Denoting by $\mathbb{Q}^x_{0,T}$ the measure induced by (29), we can draw sample paths from $\mathbb{Q}^x_{0,T}$ by instead drawing sample paths from an equivalent proposal measure $\mathbb{W}^x_{0,T}$ (a natural choice being a driftless version of (29), which will no longer coincide with Wiener measure), and accepting them with probability proportional to the Radon–Nikodým derivative of $\mathbb{Q}^x_{0,T}$ with respect to $\mathbb{W}^x_{0,T}$. The resulting Radon–Nikodým derivative (7) differs from that for diffusions (8) with the inclusion of an additional term, so the methodology of Section 3 can't be applied. However, (7) can be re-expressed in a product form similar to (23) (with $\psi_1, \ldots, \psi_{N_T}$ denoting the jump times in the interval $[0, T]$, $\psi_0 := 0$, $\psi_{N_T+1} := T$ and $N_t := \sum_{i\geq 1} \mathbb{1}\{\psi_i \leq t\}$),

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \prod_{i=1}^{N_T+1}\left[\exp\left\{A(X_{\psi_i -}) - A(X_{\psi_{i-1}}) - \int_{\psi_{i-1}}^{\psi_i -} \phi(X_s)\,\mathrm{d}s\right\}\right]. \tag{30}$$

This form of the Radon–Nikodým derivative is the key to constructing *Jump Exact Algorithms* (JEAs). Recall that in Section 3.1.1, decomposing the Radon–Nikodým derivative for diffusions justified the simulation of sample paths by successive simulation of sample paths of shorter length over the required interval (see (23)). Similarly, jump diffusion sample paths can be simulated by simulating diffusion sample paths of shorter length between consecutive jumps.

In this section, we present three novel Jump Exact Algorithms (JEAs). In contrast with existing algorithms [13,17,20], we note that the Bounded, Unbounded and Adaptive Unbounded Exact Algorithms in Section 3 can all be incorporated (with an appropriate choice of layered Brownian bridge construction) within any of the JEAs we develop. In Section 4.1, we present the Bounded Jump Exact Algorithm (BJEA), which is a reinterpretation and methodological extension of [13], addressing the case where there exists an explicit bound for the intensity of the jump process. In Section 4.2, we present the Unbounded Jump Exact Algorithm (UJEA) which is an extension to existing exact algorithms [17,20] in which the jump intensity is only locally bounded. Finally, in Section 4.3 we introduce an entirely novel Adaptive Unbounded Jump Exact Algorithm (AUJEA) based on the adaptive approach of Section 3.2, which should be adopted in the case where further simulation of a proposed sample path is required after acceptance.

4.1. Bounded Jump Intensity Jump Exact Algorithm

The case where the jump diffusion we want to simulate (29) has an explicit jump intensity bound ($\sup_{u\in[0,T]} \lambda(X_u) \leq \Lambda < \infty$) is of specific interest as an especially efficient exact algorithm can be implemented in this context. In particular, proposal jump times, $\Psi_1, \ldots, \Psi_{N_T^\Lambda}$ can be simulated according to a Poisson process with the homogeneous intensity Λ over the interval $[0, T]$ (where N_T^Λ denotes the number of events in the interval $[0, T]$ of a Poisson process of intensity Λ). A simple Poisson thinning argument [23] can be used to accept proposal jump times with probability $\lambda(X_{\Psi_i})/\Lambda$. As noted in [13], this approach allows the construction of a highly efficient algorithmic interpretation of the decomposition in (30). The interval can be

Algorithm 5 Bounded Jump Exact Algorithm (BJEA) [13]

```
1. Set $j = 0$. While $\Psi_j < T$,
   (a) Simulate $\tau \sim \text{Exp}(\Lambda)$. Set $j = j + 1$ and $\Psi_j = \Psi_{j-1} + \tau$.
   (b) Apply an exact algorithm to the interval $[\Psi_{j-1}, (\Psi_j \wedge T))$, obtaining skeleton $\mathcal{S}^j_{\text{EA}}$.
   (c) If $\Psi_j > T$ then set $X_T = X_{T-}$ else,
       i. With probability $\lambda(X_{\Psi_i})/\Lambda$ set $X_{\Psi_j} := X_{\Psi_j -} + \mu_{\Psi_j}$ where $\mu_{\Psi_j} \sim f_\nu(\cdot; X_{\Psi_j -})$, else
          set $X_{\Psi_j} := X_{\Psi_j -}$.
2. *Simulate $X^{\text{rem}} \sim \bigotimes_{j=1}^{N_T^\Lambda+1}(\bigotimes_{i=1}^{\kappa_j+1} \mathbb{W}_{\xi_{j,i-1},\xi_{j,i}}^{X_{\xi_{j,i-1}},X_{\xi_{j,i}}} | R^{\text{EA}}_{X[\xi_{j,0},\xi_{j,\kappa_j+1}]})$.
```

broken into segments corresponding precisely to the intervals between proposal jump times, then iteratively between successive times, an exact algorithm (as outlined in Section 3) can be used to simulate a diffusion sample path skeleton. The terminal point of each skeleton can be used to determine whether the associated proposal jump time is accepted (and if so a jump is simulated).

The Bounded Jump Exact Algorithm (BJEA) we outline in Algorithm 5 is a modification of that originally proposed in [13] (where we define $\Psi_0 := 0$, $\Psi_{N_T^\Lambda+1} := T$ and $X[s, t]$ to be the trajectory of X in the interval $[s, t] \subseteq [0, T]$). In particular, we simulate the proposal jump times iteratively (exploiting the exponential waiting time property of Poisson processes [23] as in Section 3.2), noting that the best proposal distribution may be different for each sub-interval. Furthermore, we note that any of the exact algorithms we introduced in Section 3 can be incorporated within the BJEA (and so the BJEA will satisfy at least Principles 1 and 2). In particular, the BJEA skeleton is a concatenation of exact algorithm skeletons for the intervals between each proposal jump time, so to satisfy Principle 3 and simulate the sample path at further intermediate time points (Step 2), we either augment the skeleton if the exact algorithm chosen is the Unbounded Exact Algorithm (UEA) (as discussed in Sections 3.1 and 7), or, if the exact algorithm chosen is the Adaptive Unbounded Exact Algorithm (AUEA) then simulate them directly (as discussed in Sections 3.2 and 8),

$$\mathcal{S}_{\text{BJEA}}(X) := \bigcup_{j=1}^{N_T^\Lambda+1} \mathcal{S}^j_{\text{EA}}(X). \tag{31}$$

4.2. Unbounded Jump Intensity Jump Exact Algorithm

Considering the construction of a Jump Exact Algorithm (JEA) under the weaker Condition 4 (in which we assume only that the jump intensity in (29) is locally bounded), it is not possible to first simulate the jump times as in Section 4.1. However (as in Section 3 and as noted in [17,20]), it is possible to simulate a layer $R(X) \sim \mathcal{R}$, and then compute a jump intensity bound ($\lambda \leq \Lambda_X < \infty$) conditional on this layer. As such, we can construct a JEA in this case by simply incorporating the jump intensity bound simulation within the layer framework of the Unbounded Exact Algorithm (UEA) and Adaptive Unbounded Exact Algorithm (AUEA).

Algorithm 6 Unbounded Jump Exact Algorithm (UJEA) [20]

```
1. Set $j = 0$, $\psi_j = 0$ and $N^{\lambda}_T = 0$,
   (a) Simulate skeleton end point $X_T =: y \sim h(y; X_{\psi_j}, T - \psi_j)$.
   (b) Simulate layer information $R^j_{X[\psi_j,T]} \sim \mathcal{R}$ and compute $\Lambda^j_{X[\psi_j,T]}$.
   (c) Simulate proposal jump times $N^{\Lambda,j}_T \sim \mathrm{Poi}(\Lambda^j_{X[\psi_j,T]}(T - \psi_j))$ and $\Psi^j_1, \ldots, \Psi^j_{N^{\Lambda,j}_T} \overset{\text{iid}}{\sim} \mathrm{U}[\psi_j, T]$.
   (d) Simulate skeleton points and diffusion at proposal jump times $(X_{\xi^j_1}, \ldots, X_{\xi^j_\kappa}, X_{\Psi^j_1}, \ldots, X_{\Psi^j_{N(\Lambda,j,T)}})$,
       i. Simulate $\kappa_j \sim \mathrm{Poi}([U^j_{X[\psi_j,T]} - L^j_{X[\psi_j,T]}] \cdot (T - \psi_j))$ and skeleton times $\xi^j_1, \ldots, \xi^j_\kappa \overset{\text{iid}}{\sim} \mathrm{U}[\psi_j, T]$.
       ii. Simulate sample path at $X_{\xi^j_1}, \ldots, X_{\xi^j_\kappa}, X_{\Psi^j_1}, \ldots, X_{\Psi^j_{N(\Lambda,j,T)}} \sim \mathbb{W}^{x,y}_{\psi_j,T} | R^j_{X[\psi_j,T]}$.
   (e) With probability $(1 - \prod_{i=1}^{\kappa_j} [(U^j_{X[\psi_j,T]} - \phi(X_{\xi^j_i})) / (U^j_{X[\psi_j,T]} - L^j_{X[\psi_j,T]})])$, reject and return to Step 1(a).
   (f) For $i$ in 1 to $N^{\Lambda,j}_T$,
       i. With probability $\lambda(X_{\Psi^j_i}) / \Lambda^j_{X[\psi_j,T]}$ set $X_{\Psi^j_i -} = X_{\Psi^j_i}$, $X_{\Psi^j_i} := X_{\Psi^j_i -} + \mu_{\Psi^j_i}$ where $\mu_{\Psi^j_i} \sim f_\nu(\cdot; X_{\Psi^j_i})$, $\psi_{j+1} := \Psi^j_i$, $j = j + 1$, $N^{\lambda}_T = j$, and return to Step 1(a).
2. *Simulate $X^{\mathrm{rem}} \sim \bigotimes_{j=0}^{N^{\lambda}_T} [(\bigotimes_{i=1}^{\kappa_j+1} \mathbb{W}^{X_{\xi_{j,i-1}}, X_{\xi_{j,i}}}_{\xi_{j,i-1}, \xi_{j,i}}) | R^j_{X[\psi_j,T]}]$.
```

The Unbounded Jump Exact Algorithm (UJEA) which we present in Algorithm 6 is a JEA construction based on the UEA and extended from [20]. The UJEA is necessarily more complicated than the Bounded Jump Exact Algorithm (BJEA) as simulating a layer in the UEA requires first simulating an end point. Ideally we would like to segment the interval the jump diffusion is to be simulated over into sub-intervals according to the length of time until the next jump (as in the BJEA), however, as we have simulated the end point in order to find a jump intensity bound then this is not possible. Instead we need to simulate a diffusion sample path skeleton over the entire interval (along with all proposal jump times) and then determine the time of the first accepted jump (if any) and simulate it. If a jump is accepted another diffusion sample path has to be proposed from the time of that jump to the end of the interval. This process is then iterated until no further jumps are accepted. The resulting UJEA satisfies Principles 1 and 2, however, as a consequence of the layer construction, the jump diffusion skeleton is composed of the *entirety* of each proposed diffusion sample path skeleton. In particular, we can't apply the strong Markov property to discard the sample path skeleton after an accepted jump because of the interaction between the layer and the sample path before and after the time of that jump.

$$\mathcal{S}_{\mathrm{UJEA}}(X) := \bigcup_{j=0}^{N^{\lambda}_T} \left\{ \left(\xi^j_i, X_{\xi^j_i}\right)_{i=0}^{\kappa_j+1}, \left(\Psi^j_1, X_{\Psi^j_1}\right)_{i=1}^{N^{\Lambda,j}_T}, R^j_{X[\psi_j,T]} \right\}. \tag{32}$$

The UJEA doesn't satisfy Principle 3 unless the skeleton is augmented (as with the UEA outlined in Sections 3.1 and 7). As this is computationally expensive, it is not recommended in practice. Alternatively we could use the AUEA within the UJEA to directly satisfy Principle 3, however it is more efficient in this case to implement the Adaptive Unbounded Jump Exact Algorithm (AUJEA) which will be described in Section 4.3 (for the same reasons in which the AUEA is more efficient than the UEA, as detailed in Section 3.2).

4.3. Adaptive Unbounded Jump Intensity Jump Exact Algorithm

The novel Adaptive Unbounded Jump Exact Algorithm (AUJEA) which we present in Algorithm 7 is based on the Adaptive Unbounded Exact Algorithm (AUEA) and a reinterpretation of the Unbounded Jump Exact Algorithm (UJEA). Considering the UJEA, note that if we simulate diffusion sample path skeletons using the AUEA then, as the AUEA satisfies Principle 3 directly, we can simulate proposal jump times after proposing and accepting a diffusion sample path as opposed to simulating the proposal times in conjunction with the sample path (see Algorithm 6 Step 1(d)ii). As such, we only need to simulate the next proposal jump time (as opposed to all of the jump times), which (as argued in Section 3.2), provides further information about the sample path. In particular, the proposal jump time necessarily lies between two existing

Algorithm 7 Adaptive Unbounded Jump Exact Algorithm (AUJEA)

```
1. Set $j = 0$ and $\psi_j = 0$.
2. Apply AUEA to interval $[\psi_j, T]$, obtaining skeleton $\mathcal{S}^{[\psi_j,T]}_{\text{AUEA}}$.
3. Set $k = 0$ and $\Psi^j_k = \psi_j$. While $\Psi^j_k < T$,
   (a) Compute $\Lambda^j_{X[\Psi^j_k,T]}$.
   (b) Simulate $\tau \sim \text{Exp}(\Lambda^j_{X[\Psi^j_k,T]})$. Set $k = k+1$ and $\Psi^j_k = \Psi^j_{k-1} + \tau$.
   (c) If $\Psi^j_k \le T$,
      i. Simulate $X_{\Psi^j_k} \sim \mathbb{W}^{X_{\psi_j},X_T}_{\psi_j,T} | \mathcal{S}^{[\psi_j,T]}_{\text{AUEA}}$.
      ii. Simulate $R_X^{[\xi^j_-,\Psi^j_k]}$ and $R_X^{[\Psi^j_k,\xi_+]}$ and set $\mathcal{S}^{[\psi_j,T]}_{\text{AUEA}} := \mathcal{S}^{[\psi_j,T]}_{\text{AUEA}} \cup \{X_{\Psi^j_k}, R_X^{[\xi^j_-,\Psi^j_k]}, R_X^{[\Psi^j_k,\xi_+]}\} \setminus R_X^{[\xi^j_-,\xi_+]}$.
      iii. With probability $\lambda(X_{\Psi^j_k})/\Lambda^j_{X[\Psi^j_{k-1},T]}$ set $X_{\Psi^j_k-} = X_{\Psi^j_k}$, $X_{\Psi^j_k} := X_{\Psi^j_k-} + \mu_{\Psi^j_k}$ where $\mu_{\Psi^j_k} \sim f_\nu(\cdot; X_{\Psi^j_k})$, $\psi_{j+1} := \Psi^j_k$, retain $\mathcal{S}^{[\psi_j,\psi_{j+1})}_{\text{AUEA}}$, discard $\mathcal{S}^{[\psi_{j+1},T]}_{\text{AUEA}}$, set $j = j+1$ and return to Step 2.
4. Let skeletal points $\chi_1, \ldots, \chi_m$ denote the order statistics of the time points in $\mathcal{S}_{\text{AUJEA}} := \bigcup_{i=1}^{j+1} \mathcal{S}^{[\psi_{i-1},\psi_i)}_{\text{AUEA}}$.
------------------------------------------------------------
5. *Simulate $X^{\text{rem}} \sim \bigotimes_{i=1}^{m+1} \mathbb{W}^{[X_{\chi_{i-1}},X_{\chi_i})}_{[\chi_{i-1},\chi_i)} | R_X^{[\chi_{i-1},\chi_i)}$.
```

skeletal times, $\xi_- \leq \Psi \leq \xi_+$, so the layer information for that interval, $R_X^{[\xi_-,\xi_+]}$ can be updated with layer information for each sub-interval $R_X^{[\xi_-,\Psi]}$ and $R_X^{[\Psi,\xi_+]}$ (the mechanism is detailed in Section 8.5). Furthermore, upon accepting a proposal jump time Ψ, the sample path skeleton in the sub-interval after Ψ contains no information regarding the skeleton preceding Ψ (so it can be discarded). As such, the AUJEA satisfies Principles 1, 2 and 3 and the skeleton is composed of only the accepted segments of each AUEA skeleton,

$$\mathcal{S}_{\text{AUJEA}}(X) := \bigcup_{j=1}^{N_T^\lambda+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{j-1},\psi_j)}(X). \tag{33}$$

4.4. An Extension to the Unbounded and Adaptive Unbounded Jump Exact Algorithms

In both the UJEA and AUJEA, we are unable to segment the interval the jump diffusion is to be simulated over into sub-intervals according to the length of time until the next jump (in contrast to the Bounded Jump Exact Algorithm (BJEA)). As a consequence, we simulate diffusion sample paths which are longer than necessary (so computationally more expensive), then (wastefully) partially discard them. To avoid this problem we could break the interval into segments and iteratively simulate diffusion sample paths of shorter length over the interval (as in (23)), thereby minimising the length of discarded segments beyond an accepted jump. However, the computational cost of simulating a sample path does not scale linearly with the interval it has to be simulated over, so the optimal length to decompose the interval is unknown.

It is possible to extend the UJEA and AUJEA based on this decomposition and Poisson superposition [23]. In particular, if it is possible to find a *lower* bound for the jump intensity $\lambda{\downarrow} \in (0,\lambda)$, then we can consider the target jump process as being the superposition of two jump processes (one of homogeneous intensity $\lambda{\downarrow}$ and the other with inhomogeneous intensity $\lambda - \lambda{\downarrow}$). As such we can simulate the timing of an accepted jump in the jump diffusion sample path under the homogeneous jump intensity $\lambda{\downarrow}$ by means of a $\tau \sim \text{Exp}(\lambda{\downarrow})$ random variable. If $\tau \in [0,T]$ then there is no need to simulate proposal diffusion skeletons over the entire interval $[0,T]$, instead we can simulate them over $[0,\tau]$. Furthermore, we can modify the bounding jump intensity in the UJEA and AUJEA for generating the proposal jump times in the proposal diffusion sample path skeletons from Λ_X to $\Lambda_X - \lambda{\downarrow}$.

5. ε-Strong Simulation of (Jump) Diffusions

In this section, we outline a novel approach for simulating upper and lower bounding processes which almost surely constrain (jump) diffusion sample paths to any specified tolerance. We do this by means of a significant extension of the *ε-Strong Simulation* algorithm proposed in [8], based upon the adaptive exact algorithms we developed in Sections 3.2 and 4.3.

As originally proposed in [8] and presented in Algorithm 8, ε-strong simulation is an algorithm which simulates upper and lower convergent bounding processes ($X^{\uparrow}$ and $X^{\downarrow}$) which enfold,

Algorithm 8 ε-Strong Simulation of Brownian Motion sample paths (n bisections)

```
1. Simulate $X_T =: y \sim \mathrm{N}(0, T)$.
2. Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\mathcal{S} := \{\Xi\} := \{\{[0, T], X_0, X_T, R_X\}\}$.
3. While $|\mathcal{S}| \leq 2^{n-1}$,
   (a) Set $\Gamma = \varnothing$.
   (b) For $i$ in 1 to $|\mathcal{S}|$,
       i. Set $\Xi = \mathcal{S}_i$ and $q := (s(\Xi) + t(\Xi))/2$.
       ii. Simulate $X_q \sim \mathbb{W}^{x(\Xi),y(\Xi)}_{s(\Xi),t(\Xi)} | R^{\Xi}_X$.
       iii. Simulate new layer information $R^{[s(\Xi),q]}_X$ and $R^{[q,t(\Xi)]}_X$ conditional on $R^{\Xi}_X$.
       iv. Refine layer information $R^{[s(\Xi),q]}_X$ and $R^{[q,t(\Xi)]}_X$.
       v. Set $\Gamma := \Gamma \cup \{[s(\Xi), q], X^{\Xi}_s, X_q, R^{[s(\Xi),q]}_X\} \cup \{[q, t(\Xi)], X_q, X^{\Xi}_t, R^{[q,t(\Xi)]}_X\}$.
   (c) Set $\mathcal{S} = \Gamma$.
```

almost surely, Brownian motion sample paths over some finite interval $[0, T]$. In particular, we have $\forall u \in [0, T]$ and some counter n,

$$X^{\downarrow}_u(n) \leq X^{\downarrow}_u(n+1) \leq X_u \leq X^{\uparrow}_u(n+1) \leq X^{\uparrow}_u(n), \qquad \text{w.p. } 1. \tag{34}$$

The details on how to simulate initial layer information (Algorithm 8 Step 2) can be found in Section 8.1 (Algorithm 16), simulate intermediate points (Algorithm 8 Step 3(b)ii) can be found in Section 8.2, and simulate new layer information (Algorithm 8 Step 3(b)iii, also know as *Bisection*) can be found in Section 8.3 (Algorithm 20) – note that each of these steps can also be found within the Adaptive Unbounded Exact Algorithm (AUEA; Algorithm 4). The additional *Refinement* Step 3(b)iv in Algorithm 8 is a method by which a tighter interval in which the sample path is constrained can be found and, as shown in [8], ensures a rate of convergence of $\mathcal{O}(2^{-n/2})$ of the upper and lower bounding processes ($X^{\uparrow}$ and $X^{\downarrow}$) to X. Details on how to conduct Algorithm 8 Step 3(b)iv can be found in Section 8.4 (Algorithm 21).

Note that the ε-strong simulation algorithm presented in Algorithm 8 differs from that presented in [8]. In particular, in contrast to [8], we simulate initial layer information unbiasedly (Algorithm 8 Step 2) and simulate intermediate points exactly (Algorithm 8 Step 3(b)ii). It should be further noted that Algorithm 8 doesn't guarantee any particular tolerance in the difference between the integrals of the upper and lower bounding processes over the interval $[0, T]$ can be achieved as currently written (we address this later in this section), but does have controlled computational cost.

The upper and lower convergent bounding processes ($X^{\uparrow}$ and $X^{\downarrow}$) can be found as a function of the layer information for each interval in Algorithm 8, and as shown in [8] Proposition 3.1, convergence in the supremum norm holds (denoting by s_i and t_i the left and right hand time points of $\mathcal{S}_i$, respectively, and $\ell^i_{s,t}$ and $\upsilon^i_{s,t}$ as the infimum and supremum of $R^{\mathcal{S}(i)}_X$, resp.),

$$\text{w.p. 1:} \qquad \lim_{n\to\infty} \sup_u \left| X^{\uparrow}_u(n) - X^{\downarrow}_u(n) \right| \to 0, \tag{35}$$

Algorithm 9 ε-Strong Simulation of Jump Diffusion sample paths (n bisections)

```
1. Simulate jump diffusion skeleton as per Algorithm 4 to obtain initial intersection layer.
2. Simulate further intersection layers as required ($n$ bisections) as per Algorithm 8.
```

where

$$X_u^{\uparrow}(n) := \inf\bigl\{\upsilon_{s,t}^i\colon\ u \in [s_i, t_i]\bigr\}, \qquad X_u^{\downarrow}(n) := \sup\bigl\{\ell_{s,t}^i\colon\ u \in [s_i, t_i]\bigr\}. \tag{36}$$

Furthermore, it was shown in [8], Proposition 3.2, that dominating processes can be constructed which converge in the L_1 norm with rate of the order $\mathcal{O}(2^{-n/2})$,

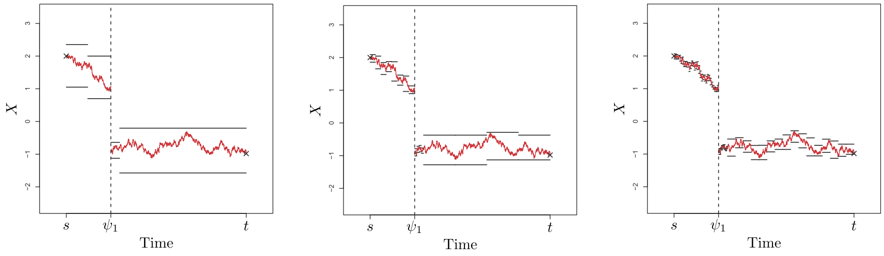
$$2^{n/2} \times \mathbb{E}\bigl[\bigl|X^{\uparrow} - X^{\downarrow}\bigr|_1\bigr] = \mathcal{O}(1), \tag{37}$$

where

$$X^{\uparrow}(n) := \sum_{i=1}^{2^{n-1}} \upsilon_{s,t}^i \cdot (t_i - s_i), \qquad X^{\downarrow}(n) := \sum_{i=1}^{2^{n-1}} \ell_{s,t}^i \cdot (t_i - s_i). \tag{38}$$

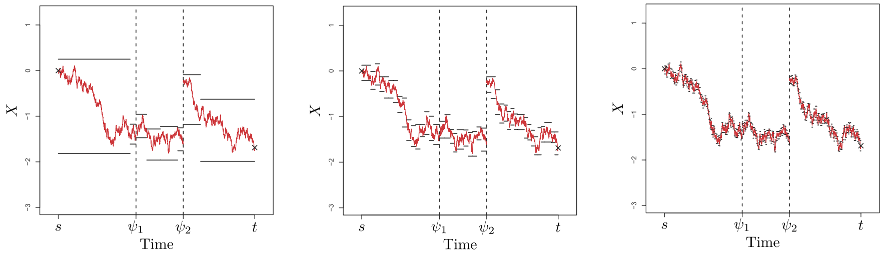
Now, considering the ε-*Strong Simulation of Jump Diffusions*, note that upon simulating a jump diffusion sample path skeleton (as per the AUJEA), it has a form (see (28) and (33)) that can be used in Algorithm 8. As such, Algorithm 8 can be extended to jump diffusions (Algorithm 9), and (35) and (37) still hold.

As far as we are aware, there are no existing methods for the ε-strong simulation of jump diffusions. The class of jump diffusions to which this methodology can be applied is broad (the conditions outlined in Section 2 are sufficient) and motivate a number of avenues for future research. In particular, non-trivial characteristics of the diffusion path can be simulated (e.g., extrema, hitting times, integrals) and can be applied to areas such as option pricing, rare event simulation and the simulation of stochastic volatility models (which are currently being explored in related work). The precise implementation of Algorithm 9 can be tailored to the specific application. For instance, in Figure 6 we present the ε-strong simulation of a jump diffusion sample path as detailed in Algorithm 9, whereas in Figure 7 we instead consider an alternate tolerance-based ε-strong simulation of a jump diffusion sample path in which we are instead interested in



(a) Sample path skeleton (b) After $n = 2$ bisections (c) After $n = 4$ bisections

Figure 6. Illustration of standard ε-strong simulation of a jump diffusion sample path, overlaid with sample path.



(a) Sample path skeleton (b) $|X^{\uparrow} - X^{\downarrow}| \leq 0.5$ (c) $|X^{\uparrow} - X^{\downarrow}| \leq 0.2$

Figure 7. Illustration of modified tolerance-based ε-strong simulation of a jump diffusion sample path, overlaid with sample path.

minimising (for any given computational budget) the L_1 distance. In particular, in this example an iterative procedure was established (in which the intersection layer with the greatest contribution to the L_1 distance was selected and split into two), until such point that the difference between the integrals of the upper and lower bounding processes over the entire interval reached a predetermined tolerance.

5.1. An ε-Strong Exact Algorithm for Diffusions

Reconsidering our initial Implementable Exact Algorithm (Algorithm 2), recall that after simulating the end point from biased Brownian motion (Algorithm 2 Step 1), the remainder of the proposal sample path can be simulated exactly from the law of a Brownian bridge (see Theorem 1). In order to determine whether to accept or reject a sample path simulated from our proposal measure ($X \sim \mathbb{Z}^x_{0,T}$) as a sample path from our target measure (denoted $\mathbb{Q}^x_{0,T}$), we accept the sample path with probability $P_{\mathbb{Z}^x_{0,T}}(X)$. In Section 3, we explored how to simulate an event of probability $P_{\mathbb{Z}^x_{0,T}}(X)$ using only a finite dimensional realisation of the proposal sample path, however, it is interesting to note that if we reconsider the simulation of the proposal sample path in light of Algorithm 8 we can find upper and lower convergent bounding sequences for $P_{\mathbb{Z}^x_{0,T}}(X)$ (in analogous fashion to (34)) by directly mapping the upper and lower bounds of the underlying proposal sample path X obtained from ε-strong simulation (recalling that $P_{\mathbb{Z}^x_{0,T}}(X) = \exp\{-\int_0^T (\phi(X_s) - \Phi)\,\mathrm{d}s\}$),

$$0 \leq \cdots \leq \phi^{\downarrow}_n \leq \phi^{\downarrow}_{n+1} \leq \cdots \leq P_{\mathbb{Z}^x_{0,T}}(X) \leq \cdots \leq \phi^{\uparrow}_{n+1} \leq \phi^{\uparrow}_n \leq \cdots \leq 1, \tag{39}$$

where we define the bounding sequences as follows (recalling $\phi(X)$ is bounded on compact sets (Result 4), and that s_i and t_i depend explicitly on n (see Algorithm 8)),

$$\phi^{\downarrow}_n := \exp\left\{-\sum_{i=1}^{n}\Big(\sup_{u \in [\ell^i_{s,t}, \upsilon^i_{s,t}]} \big(\phi(u) - \Phi\big)\Big) \cdot (t_i - s_i)\right\}, \tag{40}$$

$$\phi^{\uparrow}_n := \exp\left\{-\sum_{i=1}^{n}\Big(\inf_{u \in [\ell^i_{s,t}, \upsilon^i_{s,t}]} \big(\phi(u) - \Phi\big)\Big) \cdot (t_i - s_i)\right\}. \tag{41}$$

Algorithm 10 ε-Strong Exact Algorithm (εEA)

```
1. Simulate skeleton end point $X_T =: y \sim h$.
2. Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\mathcal{S} := \{\Xi\} := \{\{[0,T], X_0, X_T, R_X\}\}$.
3. Simulate $u \sim \mathrm{U}[0,1]$, set $n = 1$ and compute $\phi_n^{\downarrow}$ and $\phi_n^{\uparrow}$.
4. While $u \in (\phi_n^{\downarrow}, \phi_n^{\uparrow})$,
    (a) Set $\Xi = \mathcal{S}_{i^*}$ (where $i^*$ is as in (42)) and $q := (s(\Xi) + t(\Xi))/2$.
    (b) Simulate $X_{q_n} \sim \mathbb{W}^{x(\Xi),y(\Xi)}_{s(\Xi),t(\Xi)} | R^{\Xi}_X$.
    (c) Simulate new layer information $R_X^{[s(\Xi),q(n)]}$ and $R_X^{[q(n),t(\Xi)]}$ conditional on $R^{\Xi}_X$.
    (d) Refine layer information $R_X^{[s(\Xi),q(n)]}$ and $R_X^{[q(n),t(\Xi)]}$.
    (e) Set $\mathcal{S} := \mathcal{S} \cup \{[s(\Xi), q_n], X^{\Xi}_s, X_{q_n}, R_X^{[s(\Xi),q(n)]}\} \cup \{[q_n, t(\Xi)], X_{q_n}, X^{\Xi}_t, R_X^{[q(n),t(\Xi)]}\} \setminus \Xi$, set $n = n + 1$ and compute $\phi_n^{\downarrow}$ and $\phi_n^{\uparrow}$.
5. If $u \leq \phi_n^{\downarrow}$ accept skeleton, defining $\xi_0, \xi_1, \dots, \xi_{n+1}$ as the order statistics of the set $\{s, q_1, \dots, q_n, t\}$ else if $u \geq \phi_n^{\uparrow}$ reject and return to Step 1.
---
6. *Simulate $X^{\mathrm{rem}} \sim \bigotimes_{i=1}^{n+1} (\mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i} | R_X^{[\xi_{i-1},\xi_i]})$.
```

As such we can simulate events of probability $P_{\mathbb{Z}^x_{0,T}}(X)$ by direct application of series sampling (see Theorem 2 in Section 6) and hence construct an *ε-Strong Exact Algorithm* (εEA) as outlined in Algorithm 10. The precise implementation of Algorithm 10 differs from that of Algorithm 8 as at each iteration of the algorithm we want to select an interval to bisect and refine from the existing finite dimensional realisation of the proposal sample path in order to find bounds for $P_{\mathbb{Z}^x_{0,T}}(X)$ which are as tight as possible (this is similar to the tolerance-based ε-strong simulation illustrated in Figure 7). More precisely, at step $(n+1)$ we choose to bisect and refine the following interval,

$$i^* := \operatorname*{arg\,max}_{i\in\{1,\dots,n\}} \Big[\Big(\sup_{u\in[\ell^i_{s,t},\upsilon^i_{s,t}]} \phi(u) - \inf_{u\in[\ell^i_{s,t},\upsilon^i_{s,t}]} \phi(u)\Big) \cdot (t_i - s_i)\Big]. \tag{42}$$

Algorithm 10 can be employed to simulate the same class of diffusions as outlined in Section 2 and furthermore satisfies Principles 1, 2 and 3. The resulting skeleton comprises all simulated intersection layers as shown in (43) and admits the further simulation of intermediate points by direct application of Algorithm 22 (as detailed in Section 8.5). It should be noted that extension of this exact algorithm to jump diffusions can be straight forwardly performed in an analogous fashion to the extension of the exact algorithms in Section 3 to exact algorithms for jump diffusions in Section 4.

$$\mathcal{S}_{\varepsilon\mathrm{EA}}(X) := \Big\{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, \big(R_X^{[\xi_{i-1},\xi_i]}\big)_{i=1}^{\kappa+1}\Big\}. \tag{43}$$

The natural extension to Algorithm 10 is the AUEA presented in Algorithm 4 of Section 3.2, which on implementation is far more computationally efficient than Algorithm 10 due to the slow convergence of the bounding sequences enfolding $P_{\mathbb{Z}^x_{0,T}}(X)$ in (39). However, we have included this algorithm here as in addition to providing a direct application of ε-strong simulation as presented in Section 5, it is a novel approach to the exact algorithm which opens up interesting avenues to tackle related problems (which we are currently exploring in related work).

6. Brownian Path Space Simulation

In this section, we present key results which we use to construct layered Brownian bridges in Sections 7 and 8. In Section 6.1, we outline a number of established results pertaining to the simulation of a variety of aspects of Brownian bridge sample paths. In Section 6.2, we consider known results (along with some extensions) for simulating events corresponding to the probability that Brownian and Bessel bridge sample paths are contained within particular intervals. Finally, in Section 6.3 we present novel work in which we consider simulating probabilities corresponding to a more complex Brownian path space partitioning. Central to Sections 6.2 and 6.3 are Theorem 2 and Corollaries 1 and 2, which together form the basis for simulating events of unknown probabilities p, which can be represented as alternating Cauchy sequences of the following form,

$$0 = S_0 \leq S_2 < S_4 < S_6 < \cdots < p < \cdots < S_5 < S_3 < S_1 \leq 1. \tag{44}$$

Theorem 2 (Series sampling [15], Section 4.5). *An event of (unknown) probability $p \in [0, 1]$, where there exists monotonically decreasing and increasing sequences, $(S_k^+ \colon k \in \mathbb{Z}_{\geq 0})$ and $(S_k^- \colon k \in \mathbb{Z}_{\geq 0})$ respectively, such that $\lim_{k\to\infty} S_k^+ \downarrow p$ and $\lim_{k\to\infty} S_k^+ \uparrow p$, can be simulated unbiasedly. In particular, a binary random variable $P := \mathbb{1}(u \leq p)$ can be simulated (where $u \sim \mathrm{U}[0, 1]$), noting that as there almost surely exists a finite $K := \inf\{k \colon u \notin (S_k^-, S_k^+)\}$ we have $\mathbb{1}(u \leq p) = \mathbb{1}(u \leq S_K^-)$ and $\mathbb{E}[\mathbb{1}(u \leq S_K^-)] = p$.*

Corollary 1 (Transformation). *Probabilities which are linear transformations or ratios of a collection of probabilities, each of which have upper and lower convergent sequences can be simulated by extension of Theorem 2. In particular, suppose $f : \mathbb{R}_+^m \to \mathbb{R}_+ \in C^1$ such that $|\mathrm{d}f/\mathrm{d}u_i(u)| > 0$ $\forall 1 \leq i \leq m$ and $u \in \mathbb{R}_+^m$ and that the probability $p := f(p_1, \ldots, p_m)$ then defining the sequences $(T_k^{i,-} \colon k \in \mathbb{Z}_{\geq 0})$ and $(T_k^{i,+} \colon k \in \mathbb{Z}_{\geq 0})$ as follows,*

$$T_k^{i,-} = \begin{cases} S_k^{i,-}, & \text{if } \mathrm{d}f/\mathrm{d}u_i > 0, \\ S_k^{i,+}, & \text{if } \mathrm{d}f/\mathrm{d}u_i < 0, \end{cases} \qquad T_k^{i,+} = \begin{cases} S_k^{i,+}, & \text{if } \mathrm{d}f/\mathrm{d}u_i > 0, \\ S_{k+1}^{i,-}, & \text{if } \mathrm{d}f/\mathrm{d}u_i < 0. \end{cases} \tag{45}$$

We have that $S_k^- := f(T_k^{1,-}, \ldots, T_k^{m,-})$ is monotonically increasing and converges to p from below and $S_k^+ := f(T_k^{1,+}, \ldots, T_k^{m,+})$ is monotonically decreasing and converges to p from above.

Corollary 2 (Retrospective Bernoulli sampling [6], Proposition 1). *If p can be represented as the limit of an alternating Cauchy sequence of the form of (44) $(S_k \colon k \in \mathbb{Z}_{\geq 0})$, then splitting the sequence into subsequences composed of the odd and even terms, respectively, each subsequence will converge to p, one of which will be monotonically decreasing and the other monotonically increasing, so events of probability p can be simulated by extension of Theorem 2.*

Algorithm 11 Retrospective Bernoulli Sampling [6]

```
1. Simulate $u \sim \mathrm{U}[0,1]$ and set $k = 1$.
2. While $u \in (S_{2k}, S_{2k+1})$, $k = k + 1$.
3. If $u \le S_{2k}$ then $u < p$ so return 1 else $u > p$ so return 0.
```

We conclude the introductory remarks to this section by presenting Algorithm 11, in which we outline by application of Corollary 2 how an unknown probability which can be represented as an alternating Cauchy sequence in which (without loss of generality) the even terms converge from below and the odd terms from above, can be simulated unbiasedly.

Clearly as we have that the number of computations required to implement retrospective Bernoulli sampling is stochastic (as a consequence of Algorithm 11 Step 2), the efficiency of the algorithm is dependent upon the expected number of iterations of that step required (where $u \sim \mathrm{U}[0,1]$ as per Algorithm 11 Step 1),

$$\mathbb{E}[K] = \sum_{k=1}^{\infty} \mathbb{P}(K \ge k) = \sum_{k=0}^{\infty} \mathbb{P}\bigl(u \in [S_{2k}, S_{2k+1}]\bigr) = \sum_{k=0}^{\infty} |S_{2k+1} - S_{2k}|. \tag{46}$$

At a minimum for any practical implementation, we require that the $\mathbb{E}[K] < \infty$, which can't be ensured without imposing further conditions. However, as we will encounter in Sections 6.2 and 6.3, the alternating Cauchy sequences which we consider in this paper converge exponentially fast and so finiteness is ensured.

6.1. Simulating Brownian Bridges and Related Processes

The density of a Brownian bridge sample path $W^{x,y}_{s,t}$, at an intermediate time $q \in (s,t)$ is Gaussian with mean $\mu_w := x + (q-s)(y-x)/(t-s)$ and variance $\sigma^2_w := (t-q)(q-s)/(t-s)$ (so can be simulated directly). The joint distribution of the minimum value reached by $W^{x,y}_{s,t}$, and the time at which it is attained $(\tau, \hat{m})$, is given by [22],

$$\begin{aligned} &\mathbb{P}(\hat{m} \in \mathrm{d}w, \tau \in \mathrm{d}q \,|\, W_s = x, W_t = y) \\ &\quad \propto \frac{(w-x)(w-y)}{\sqrt{(t-q)^3(q-s)^3}} \exp\left\{-\frac{(w-x)^2}{2(q-s)} - \frac{(w-y)^2}{2(t-q)}\right\} \mathrm{d}w\, \mathrm{d}q. \end{aligned} \tag{47}$$

Analogously the maximum $(\tau, \check{m})$, can be considered by reflection. We can jointly draw $(\tau, \hat{m})$ (or $(\tau, \check{m})$) as outlined in Algorithm 12, which is similar to the approach taken in [5], noting that it is possible to condition the minimum to occur within a particular interval. In particular, we can simulate $(\tau, \hat{m}) | (\hat{m} \in [a_1, a_2])$ where $a_1 < a_2 \le (x \wedge y)$.

Conditional on a Brownian bridge sample path minimum (or maximum), the law of the remainder of the trajectory is that of a *Bessel bridge*, which can be simulated by means of a 3-dimensional Brownian bridge of unit length conditioned to start and end at zero as outlined in [3] and Algorithm 13 (maximum by reflection).

Algorithm 12 Brownian Bridge Simulation at its Minimum Point (constrained to the interval $[a_1, a_2]$ where $a_1 < a_2 \leq x \wedge y$ and conditional on $W_s = x$ and $W_t = y$ (denoting $\mathrm{IGau}(\mu, \lambda)$ as the inverse Gaussian distribution with mean μ and shape parameter λ)

```
1. Simulate $u_1 \sim \mathrm{U}[M(a_1), M(a_2)]$ where $M(a) := \exp\{-2(a-x)(a-y)/(t-s)\}$ and $u_2 \sim \mathrm{U}[0,1]$.
2. Set $\hat{m} := x - [\sqrt{(y-x)^2 - 2(t-s)\log(u_1)} - (y-x)]/2$.
3. If $u_2 \leq \frac{x-\hat{m}}{x+y-2\hat{m}}$ then $V \sim \mathrm{IGau}(\frac{y-\hat{m}}{x-\hat{m}}, \frac{(y-\hat{m})^2}{t-s})$ else $\frac{1}{V} \sim \mathrm{IGau}(\frac{x-\hat{m}}{y-\hat{m}}, \frac{(x-\hat{m})^2}{t-s})$.
4. Set $\tau := \frac{sV+t}{1+V}$.
```

6.2. Simulating Elementary Brownian Path Space Probabilities

In this section, we briefly outline results pertaining to the probability that a Brownian bridge sample path is contained within a particular interval [2,32] (Theorem 3) and how to simulate events of this probability [6] (Corollary 3). In Figure 8, we show example sample path trajectories of a Brownian bridge $W \sim \mathbb{W}^{x,y}_{s,t}$, which remain in the interval $[\ell, \upsilon]$. Similarly in Theorems 4 and 5 we outline a result (first shown in [6]) that shows that the probability a Bessel bridge sample path is contained within a particular interval can be represented as an infinite series. We reproduce these results as they are used and extended extensively throughout the remainder of this paper. In the rest of this paper, with a slight abuse of notation, we write $\{W \in [\ell, \upsilon]\}$ to mean $\{W_u\colon s \leq u \leq t\} \subset [\ell, \upsilon]$. Further details on the results developed in this section can be found in [31], Chapter 6.1.1.

Of particular importance for what follows is Corollary 5, in which we establish that it is possible to simulate events with a probability corresponding to the probability that a Bessel bridge sample path is contained within a particular interval (without assumptions on the size of the interval), by application of Corollary 2.

Theorem 3 ([32], Theorem 3). *The probability that a Brownian bridge sample path $W \sim \mathbb{W}^{x,y}_{s,t}$, remains in the interval $[\ell, \upsilon]$ (i.e., $\forall u \in [s,t]$ $W_u \in [\ell, \upsilon]$) can be represented as an infinite series,*

$$\gamma^{\ell,\upsilon}_{s,t}(x,y) := \mathbb{P}\big(W \in [\ell, \upsilon]\big) = 1 - \sum_{j=1}^{\infty} \big\{\varsigma^{\ell,\upsilon}_{s,t}(j; x, y) - \varphi^{\ell,\upsilon}_{s,t}(j; x, y)\big\}, \tag{48}$$

Algorithm 13 (Minimum) Bessel Bridge Simulation (at time $q \in (s,t)$ given $W_s = x$, $W_t = y$ and $W_\tau = \hat{m}$) [3]

```
1. If $q < \tau$ then $r = s$ else $r = t$. Simulate $b_1, b_2, b_3 \overset{\text{iid}}{\sim} \mathrm{N}(0, \frac{|\tau-q|\cdot|q-r|}{(\tau-r)^2})$.
2. Set $W_q := \hat{m} + \sqrt{|\tau-r|} \cdot \sqrt{(\frac{(W_r-\hat{m})\cdot|\tau-q|}{|\tau-r|^{3/2}} + b_1)^2 + b_2^2 + b_3^2}$.
```

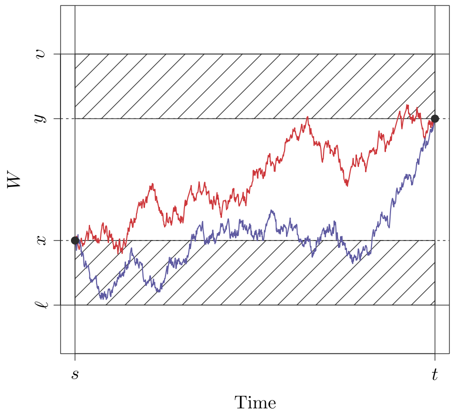


Figure 8. Example sample path trajectories $W \sim \mathbb{W}^{x,y}_{s,t}|(W_{(s,t)} \in [\ell,\upsilon])$.

where $\varsigma^{\ell,\upsilon}_{s,t}(j;x,y) := \bar{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y) + \bar{\varsigma}^{-\ell,-\upsilon}_{s,t}(j;-x,-y)$, $\varphi^{\ell,\upsilon}_{s,t}(j;x,y) := \bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y) + \bar{\varphi}^{-\ell,-\upsilon}_{s,t}(j;-x,-y)$ *and,*

$$\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y) := \exp\left\{-\frac{2}{t-s}\Big(|\upsilon-\ell|j + (\ell\wedge\upsilon) - x\Big)\cdot\Big(|\upsilon-\ell|j + (\ell\wedge\upsilon) - y\Big)\right\}, \tag{49}$$

$$\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y) := \exp\left\{-\frac{2j}{t-s}\Big(|\upsilon-\ell|^2 j + |\upsilon-\ell|(x-y)\Big)\right\}. \tag{50}$$

Corollary 3 ([6], Proposition 2). $\gamma^{\ell,\upsilon}_{s,t}(x,y)$ *is an alternating Cauchy sequence, so events of probability* $\gamma^{\ell,\upsilon}_{s,t}(x,y)$ *can be simulated by retrospective Bernoulli sampling* (*Corollary* 2 *and Algorithm* 11) *using the following sequence,*

$$S^{\gamma}_{2k} := 1 - \sum_{j=1}^{k}\left\{\varsigma^{\ell,\upsilon}_{s,t}(j;x,y) - \varphi^{\ell,\upsilon}_{s,t}(j;x,y)\right\}, \qquad S^{\gamma}_{2k+1} := S^{\gamma}_{2k} - \varsigma^{\ell,\upsilon}_{s,t}(k+1;x,y). \tag{51}$$

As shown in [6], Theorem 3 and Corollary 3 can be extended to consider simulating events with a probability corresponding to the probability a Bessel bridge sample path is contained within a particular interval. As indicated in Definition 4 we have to consider two possible cases where either of the end points attain the sample path minimum (or maximum) or not.

Definition 4. *We allow* $\delta^{\hat{m},\upsilon}_{s,t}(x,y)$ *to denote the probability that a Bessel bridge sample path* $W \sim \mathbb{W}^{x,y}_{s,t}|\hat{m}$ (*with minimum* $\hat{m}$) *remains in the interval* $[\hat{m},\upsilon]$. *We further denote* $\delta^{\hat{m},\upsilon}_{s,t}(1;x,y) := \mathbb{P}(W \in [\hat{m},\upsilon] | W \geq \hat{m}, (x\wedge y) > \hat{m})$ *and* $\delta^{\hat{m},\upsilon}_{s,t}(2;x,y) := \mathbb{P}(W \in [\hat{m},\upsilon] | W \geq \hat{m}, (x\wedge y) = \hat{m})$ *noting that* $\delta^{\hat{m},\upsilon}_{s,t}(x,y) = \mathbb{1}\{\hat{m} < (x\wedge y)\}\cdot\delta^{\hat{m},\upsilon}_{s,t}(1;x,y) + \mathbb{1}\{\hat{m} = (x\wedge y)\}\cdot\delta^{\hat{m},\upsilon}_{s,t}(2;x,y)$.

Note that we can similarly consider the probability that a Bessel bridge sample path $W \sim \mathbb{W}^{x,y}_{s,t}|\check{m}$ (with maximum $\check{m}$) remains in the interval $[\ell, \check{m}]$ ($\forall u \in [s,t]$ $W_u \in [\ell, \check{m}]$) by a simple reflection argument.

We first consider the case where neither end point attains the Bessel bridge minimum.

Theorem 4 ([6], Proposition 3). *The probability that a Bessel bridge sample path* $W \sim \mathbb{W}^{x,y}_{s,t}|\hat{m}$, (*with minimum* $\hat{m} < (x \wedge y)$) *remains in the interval* $[\hat{m}, \upsilon]$ ($\forall u \in [s,t]$ $W_u \in [\hat{m}, \upsilon]$) *can be represented as an infinite series*,

$$\begin{aligned} \delta^{\hat{m},\upsilon}_{s,t}(1; x, y) &:= \mathbb{P}\big(W \in [\hat{m}, \upsilon] | W \geq \hat{m}, (x \wedge y) > \hat{m}\big) \\ &= \frac{\gamma^{\hat{m},\upsilon}_{s,t}(x,y)}{1 - \exp\{-2(x-\hat{m})(y-\hat{m})/(t-s)\}}. \end{aligned} \tag{52}$$

Corollary 4 ([6], Proposition 3). *Events of probability* $\delta^{\hat{m},\upsilon}_{s,t}(1; x, y)$ *can be simulated by application of retrospective Bernoulli sampling* (*as per Corollaries* 1, 2 *and Algorithm* 11) *using the following sequence*,

$$S^{\delta,1}_k := \frac{S^{\gamma}_k}{1 - \exp\{-2(x-\hat{m})(y-\hat{m})/(t-s)\}}. \tag{53}$$

We now consider the case where either one of the end points attains the Bessel bridge minimum.

Theorem 5 ([6], Proposition 3). *The probability that a Bessel bridge sample path* $W \sim \mathbb{W}^{x,y}_{s,t}|\hat{m}$ (*with minimum* $\hat{m} = x < y$) *remains in the interval* $[\hat{m}, \upsilon]$ ($\forall u \in [s,t]$ $W_u \in [\hat{m}, \upsilon]$) *can be represented as an infinite series*,

$$\begin{aligned} \delta^{\hat{m},\upsilon}_{s,t}(2; x, y) &:= \mathbb{P}\big(W \in [\hat{m}, \upsilon] | W \geq \hat{m}\big) \\ &= 1 - \frac{1}{(y-\hat{m})} \sum_{j=1}^{\infty} \big\{\psi^{\hat{m},\upsilon}_{s,t}(j; y) - \chi^{\hat{m},\upsilon}_{s,t}(j; y)\big\}, \end{aligned} \tag{54}$$

where we denote,

$$\psi^{\hat{m},\upsilon}_{s,t}(j; y) := \big(2|\upsilon - \hat{m}|j - (y-\hat{m})\big) \exp\left\{-\frac{2|\upsilon-\hat{m}|j}{t-s}\big(|\upsilon-\hat{m}|j - (y-\hat{m})\big)\right\}, \tag{55}$$

$$\chi^{\hat{m},\upsilon}_{s,t}(j; y) := \big(2|\upsilon - \hat{m}|j + (y-\hat{m})\big) \exp\left\{-\frac{2|\upsilon-\hat{m}|j}{t-s}\big(|\upsilon-\hat{m}|j + (y-\hat{m})\big)\right\}. \tag{56}$$

Remark 1 ([6], Proposition 3). As before, we can consider the probability a Bessel bridge sample path $W \sim \mathbb{W}^{x,y}_{s,t}|\hat{m}$ (with minimum $\hat{m} = y < x$) remains in the interval $[\hat{m}, \upsilon]$ by a simple reflection argument of Theorem 5.

We conclude this section by showing that it is possible to simulate events with probability corresponding to the probability a Bessel bridge sample path is contained within a particular interval, without any further assumption regarding the interval size (unlike existing methods [6], Proposition 3, in which one requires that $3(\upsilon-\hat{m})^2 > (t-s)$). As we consider in this setting $s, t, x, y, \hat{m}, \upsilon$ fixed, for conciseness we denote $\psi_j := \psi^{\hat{m},\upsilon}_{s,t}(j;y)$ and $\chi_j := \chi^{\hat{m},\upsilon}_{s,t}(j;y)$.

Corollary 5. *After the inclusion of the first* $\hat{k} := \sqrt{(t-s)+|\upsilon-\hat{m}|^2}/(2|\upsilon-\hat{m}|)$ *terms,* $\delta^{\hat{m},\upsilon}_{s,t}(2;x,y)$ *is an alternating Cauchy sequence, so events of probability* $\delta^{\hat{m},\upsilon}_{s,t}(2;x,y)$ *can be simulated by retrospective Bernoulli sampling* (*as per Corollary 2 and Algorithm 11*) *using the following sequence* (*where* $k\in\mathbb{N}$ *such that* $k\geq\hat{k}$),

$$S^{\delta,2}_{2k} := 1-\frac{1}{(y-\hat{m})}\sum_{j=1}^{k}\bigl\{\psi^{\hat{m},\upsilon}_{s,t}(j;y)-\chi^{\hat{m},\upsilon}_{s,t}(j;y)\bigr\}, \tag{57}$$

$$S^{\delta,2}_{2k+1} := S^{\delta,2}_{2k}-\frac{1}{y-\hat{m}}\psi^{\hat{m},\upsilon}_{s,t}(k+1;y). \tag{58}$$

Proof. As $(y-\hat{m})\in(0,(\upsilon-\hat{m})]$ then $\forall j$ we have $\psi_j,\chi_j\geq 0$. As such it is sufficient to show that $\forall j\geq\hat{k}$ that $\psi_j\geq\chi_j\geq\psi_{j+1}\geq\chi_{j+1}\geq\cdots$ which can be proved inductively by first showing that $\forall j\ \psi_j\geq\chi_j$ and then $\forall j\ \chi_j\geq\psi_{j+1}$. Considering ψ_j/χ_j if $j\geq\hat{k}$ then this is minimised when $y=\hat{m}$ and $\psi_j/\chi_j>1$. Similarly considering χ_j/ψ_{j+1} if $j\geq\hat{k}$ then this is minimised when $y=\upsilon$ where $\chi_j/\psi_{j+1}>1$. □

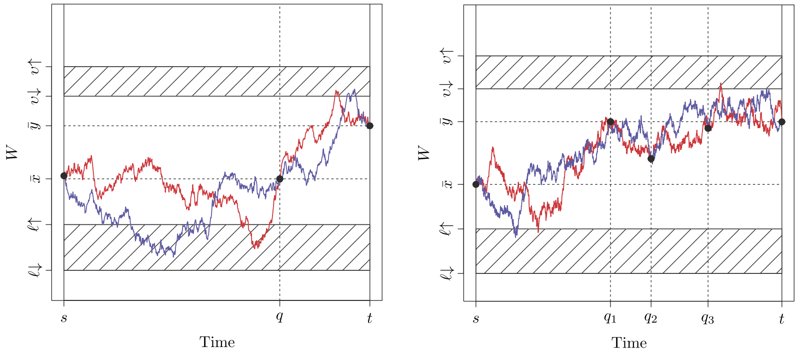
6.3. Simulating Brownian Path Space Probabilities

In this section, we establish that the probability a Brownian bridge sample path, conditioned on a number of intermediate points $(q_1,\ldots,q_n)$, has a minimum in a lower interval and a maximum in an upper interval (or in each sub-interval a minimum in a lower interval and a maximum in an upper interval) can be represented as an infinite series (Theorems 6 and 7, resp.), and events of this probability can be simulated (Corollaries 6 and 7, resp.). Further details on the results developed in this section can be found in [31], Chapter 6.1.2.

In this section, we introduce the following simplifying notation, $q_{1:n} := \{q_1,\ldots,q_n\}$, $q_0 := s$ and $q_{n+1} := t$. We further denote $\hat{m}_{s,t} := \inf\{W_q; q\in[s,t]\}$, $\check{m}_{s,t} := \sup\{W_q; q\in[s,t]\}$, $\mathcal{W} := \{W_{q_1}=w_1,\ldots,W_{q_n}=w_n\}$, $\mathcal{L} := \{\hat{m}_{s,q_1}\in[\ell^{\downarrow}_{s,q_1},\ell^{\uparrow}_{s,q_1}],\ldots,\hat{m}_{q_n,t}\in[\ell^{\downarrow}_{q_n,t},\ell^{\uparrow}_{q_n,t}]\}$, $\mathcal{U} := \{\check{m}_{s,q_1}\in[\upsilon^{\downarrow}_{s,q_1},\upsilon^{\uparrow}_{s,q_1}],\ldots,\check{m}_{q_n,t}\in[\upsilon^{\downarrow}_{q_n,t},\upsilon^{\uparrow}_{q_n,t}]\}$. We also use the following abuse of notation $\{W_{[s,t]}\in[\ell,\upsilon]\} := \{W_u\in[\ell,\upsilon]\ \forall u\in[s,t]\}$, noting that $\{W_{[s,t]}\in[\ell,\upsilon]\} = \{\hat{m}_{s,t}\in[\ell,(x\wedge y)],\check{m}_{s,t}\in[(x\vee y),\upsilon]\}$.

Theorem 6. *The probability a Brownian bridge sample path* $W\sim\mathbb{W}^{x,y}_{s,t}|\mathcal{W}$ *has a minimum* $\hat{m}_{s,t}\in[\ell\downarrow,\ell\uparrow]$ *and a maximum* $\check{m}_{s,t}\in[\upsilon\downarrow,\upsilon\uparrow]$ *can be represented as an infinite series,*

$$\begin{aligned}
&{}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W})\\
&\quad := \mathbb{P}\bigl(\hat{m}_{s,t}\in[\ell\downarrow,\ell\uparrow],\check{m}_{s,t}\in[\upsilon\downarrow,\upsilon\uparrow]|\mathcal{W}\bigr)\\
&\quad = \left[\prod_{i=0}^{n}\gamma^{\ell\downarrow,\upsilon\uparrow}_{q_i,q_{i+1}}\right]-\left[\prod_{i=0}^{n}\gamma^{\ell\uparrow,\upsilon\uparrow}_{q_i,q_{i+1}}\right]-\left[\prod_{i=0}^{n}\gamma^{\ell\downarrow,\upsilon\downarrow}_{q_i,q_{i+1}}\right]+\left[\prod_{i=0}^{n}\gamma^{\ell\uparrow,\upsilon\downarrow}_{q_i,q_{i+1}}\right].
\end{aligned} \tag{59}$$



(a) $n = 1$ (b) $n = 3$

Figure 9. Example sample path trajectories $W \sim \mathbb{W}^{x,y}_{s,t} | (\hat{m} \in [\ell\downarrow, \ell\uparrow], \check{m} \in [\upsilon\downarrow, \upsilon\uparrow], q_{1:n}, \mathcal{W})$.

Proof. Follows by sample path inclusion–exclusion and the Markov property for diffusions,

$$\begin{aligned}
&{}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n}, \mathcal{W}) \\
&\quad = \mathbb{P}\big(W \in [\ell\downarrow, \upsilon\uparrow] | \mathcal{W}\big) - \mathbb{P}\big(W \in [\ell\uparrow, \upsilon\uparrow] | \mathcal{W}\big) \\
&\qquad - \mathbb{P}\big(W \in [\ell\downarrow, \upsilon\downarrow] | \mathcal{W}\big) + \mathbb{P}\big(W \in [\ell\uparrow, \upsilon\downarrow] | \mathcal{W}\big) \qquad (60) \\
&\quad = \prod_{i=0}^{n} \mathbb{P}\big(W_{[q_i,q_{i+1}]} \in [\ell\downarrow, \upsilon\uparrow] | W_{q_i}, W_{q_{i+1}}\big) - \prod_{i=0}^{n} \mathbb{P}\big(W_{[q_i,q_{i+1}]} \in [\ell\uparrow, \upsilon\uparrow] | W_{q_i}, W_{q_{i+1}}\big) \qquad (61) \\
&\qquad - \prod_{i=0}^{n} \mathbb{P}\big(W_{[q_i,q_{i+1}]} \in [\ell\downarrow, \upsilon\downarrow] | W_{q_i}, W_{q_{i+1}}\big) + \prod_{i=0}^{n} \mathbb{P}\big(W_{[q_i,q_{i+1}]} \in [\ell\uparrow, \upsilon\downarrow] | W_{q_i}, W_{q_{i+1}}\big). \qquad \square
\end{aligned}$$

Intuitively ${}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n}, \mathcal{W})$ corresponds to the proportion of Brownian bridge sample paths with the restriction $\mathcal{W}$, which also have the restriction $\{\hat{m}_{s,t} \in [\ell\downarrow, \ell\uparrow], \check{m}_{s,t} \in [\upsilon\downarrow, \upsilon\uparrow]\}$ (paths of which are illustrated in Figure 9).

Corollary 6. *Events of probability* ${}^{(n)}\rho$ *can be simulated by retrospective Bernoulli sampling* (*as per Corollaries* 1, 2 *and Algorithm* 11), *noting that* ${}^{(n)}\rho$ *is a function of* γ *probabilities*, *using the following sequence*,

$$\begin{aligned}
S^{\rho(n)}_k := {}& \left[\prod_{i=0}^{n} S^{\gamma(q_i,q_{i+1},\downarrow,\uparrow)}_k\right] - \left[\prod_{i=0}^{n} S^{\gamma(q_i,q_{i+1},\uparrow,\uparrow)}_{k+1}\right] \\
& - \left[\prod_{i=0}^{n} S^{\gamma(q_i,q_{i+1},\downarrow,\downarrow)}_{k+1}\right] + \left[\prod_{i=0}^{n} S^{\gamma(q_i,q_{i+1},\uparrow,\downarrow)}_k\right].
\end{aligned} \qquad (62)$$

***Definition* 5.** *We define* $\rho(s,q,t,x,w,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) := {}^{(1)}\rho_{s,t,x,y}^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}(\{q\},\{w\})$, *which coincides with* ρ *in* [8].

Theorem 7. *The probability that a Brownian bridge sample path* $W \sim \mathbb{W}_{s,t}^{x,y}|\mathcal{W}$, *has in the sub-intervals between successive points in* $\mathcal{W}$, *a minimum and maximum in particular intervals,* $\mathcal{L}$ *and* $\mathcal{U}$ *respectively, can be represented as an infinite series,*

$$ {}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W}) := \mathbb{P}(\mathcal{L},\mathcal{U}|\mathcal{W}) = \prod_{i=0}^{n}\left[\gamma_{q_i,q_{i+1}}^{\ell\downarrow,\upsilon\uparrow} - \gamma_{q_i,q_{i+1}}^{\ell\uparrow,\upsilon\uparrow} - \gamma_{q_i,q_{i+1}}^{\ell\downarrow,\upsilon\downarrow} + \gamma_{q_i,q_{i+1}}^{\ell\uparrow,\upsilon\downarrow}\right]. \tag{63}$$

Proof. Follows by the strong Markov property for diffusions and sample path inclusion–exclusion,

$$\begin{aligned} {}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W}) = \prod_{i=0}^{n}\Big[\mathbb{P}\big(&\hat{m}_{q_i,q_{i+1}} \in \big[\ell_{q_i,q_{i+1}}^{\downarrow},\ell_{q_i,q_{i+1}}^{\uparrow}\big],\\ &\check{m}_{q_i,q_{i+1}} \in \big[\upsilon_{q_i,q_{i+1}}^{\downarrow},\upsilon_{q_i,q_{i+1}}^{\uparrow}\big]|W_{q_i}=w_i, W_{q_{i+1}}=w_{i+1}\big)\Big] \end{aligned} \tag{64}$$

$$\begin{aligned} = \prod_{i=0}^{n}\Big[&\mathbb{P}\big(W_{[q_i,q_{i+1}]} \in \big[\ell_{q_i,q_{i+1}}^{\downarrow},\upsilon_{q_i,q_{i+1}}^{\uparrow}\big]|W_{q_i},W_{q_{i+1}}\big)\\ &-\mathbb{P}\big(W_{[q_i,q_{i+1}]} \in \big[\ell_{q_i,q_{i+1}}^{\uparrow},\upsilon_{q_i,q_{i+1}}^{\uparrow}\big]|W_{q_i},W_{q_{i+1}}\big)\\ &-\mathbb{P}\big(W_{[q_i,q_{i+1}]} \in \big[\ell_{q_i,q_{i+1}}^{\downarrow},\upsilon_{q_i,q_{i+1}}^{\downarrow}\big]|W_{q_i},W_{q_{i+1}}\big)\\ &+\mathbb{P}\big(W_{[q_i,q_{i+1}]} \in \big[\ell_{q_i,q_{i+1}}^{\uparrow},\upsilon_{q_i,q_{i+1}}^{\downarrow}\big]|W_{q_i},W_{q_{i+1}}\big)\Big]. \end{aligned} \tag{65}$$

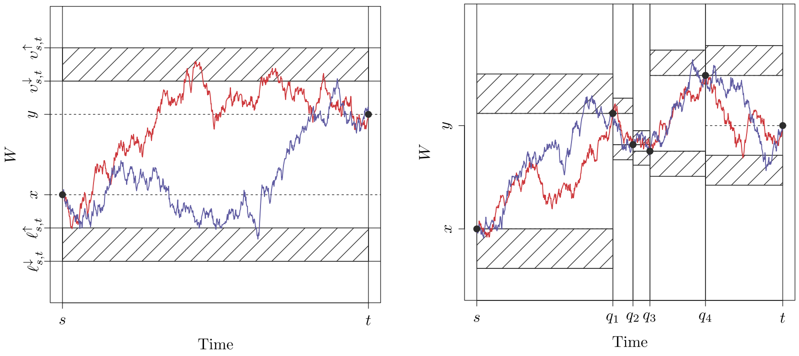
□

As before, intuitively ${}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W})$ corresponds to the proportion of Brownian bridge sample paths with the restriction $\mathcal{W}$, which also have the restriction $\{\mathcal{L},\mathcal{U}\}$ (paths of which are illustrated in Figure 10).

Corollary 7. *Events of probability* ${}^{(n)}\beta$ *can be simulated by retrospective Bernoulli sampling* (*as per Corollaries* 1, 2 *and Algorithm* 11), *noting that* ${}^{(n)}\beta$ *is a function of* γ *probabilities, using the following sequence,*

$$S_k^{\beta(n)} := \prod_{i=0}^{n}\left[S_k^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\downarrow)} + S_k^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\downarrow)}\right]. \tag{66}$$

***Definition* 6.** *We define* $\beta(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) := {}^{(0)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(\varnothing,\varnothing)$, *which coincides with* β *in* [8] (*where here we have* $\mathcal{L} = \{\hat{m}_{s,t} \in [\ell\downarrow,\ell\uparrow]\}$ *and* $\mathcal{U} := \{\check{m}_{s,t} \in [\upsilon\downarrow,\upsilon\uparrow]\}$).



(a) $n = 0$ (b) $n = 4$

Figure 10. Example sample path trajectories $W \sim \mathbb{W}^{x,y}_{s,t} | (\mathcal{L}, \mathcal{U}, q_{1:n}, \mathcal{W})$.

7. Layered Brownian Bridge Constructions

In this section, we outline how to construct and simulate finite dimensional skeletons of layered Brownian bridges for use within the Unbounded Exact Algorithm (UEA) (Algorithm 3), which is in turn used within the Bounded Jump Exact Algorithm (BJEA) (Algorithm 5) and Unbounded Jump Exact Algorithm (UJEA) (Algorithm 6). In particular, we address the simulation of layer information (Algorithm 3 Step 2), intermediate skeletal points (Algorithm 3 Step 4) and the process at further times after acceptance of the proposed sample path (Algorithm 3 Step 6).

We present two alternate layered Brownian bridge constructions based on extensions to existing exact algorithms. In Section 7.1, we present the *Bessel Approach*, which is a reinterpretation of part of the *Exact Algorithm* 3 (EA3) proposed in [6], in which we incorporate the methodological improvements outlined in Sections 3 and 6 and introduce a novel approach for conducting Algorithm 3 Step 6 (which could not previously be achieved). As a consequence, the resulting (complete) UEA, with the inclusion of the Bessel approach, satisfies Principles 1, 2 and 3 (as opposed to only Principles 1 and 2 in EA3 [6]). Finally, in Section 7.2 we briefly outline a *Localised Approach* for constructing a layered Brownian bridge (based on the *Localised Exact Algorithm* (LEA) [14,17]), showing that the resulting UEA only satisfies Principles 1 and 2 and discussing the difficulties in conducting Algorithm 3 Step 6 and satisfying Principle 3.

In neither the Bessel nor the Localised approaches is it possible to directly simulate intermediate points conditional on a simulated layer (as required in Algorithm 3 Step 2). Instead, in order to simulate proposal sample path skeletons we employ other Monte Carlo techniques, including rejection sampling (see Section 3) and demarginalisation.

Demarginalisation [34], Section 5.3, is a technique whereby artificial extension of a density (with the incorporation of auxiliary variables) simplifies sampling from it. To illustrate this, consider the case where we want to draw a sample $g(X)$, but this is not directly possible. However, suppose that with the introduction of an auxiliary variable Y, sampling from $g(Y)$ and $g(X|Y)$

is possible and $g(X,Y)$ admits $g(X)$ as a marginal,

$$g(X)=\int_{\mathcal{Y}} g(X|Y)g(Y)\,\mathrm{d}Y. \tag{67}$$

We can sample from $g(X)$ by first sampling Y from $g(Y)$ and then sampling from $g(X|Y)$. This algorithm can be viewed as a black box to generate samples from $g(X)$ – Y can be simply marginalised out (i.e., "thrown" away). Considering demarginalisation in the context of the exact algorithms, we can simulate any (auxiliary) aspect of the proposal diffusion sample path in addition to the skeleton to aid sampling. Provided the auxiliary information does not influence the acceptance probability then it is not part of the skeleton and doesn't need to be retained.

7.1. Bessel Approach

The central idea of the *Bessel Approach* is that finite dimensional subsets of Brownian bridge sample paths can be simulated jointly with information regarding the interval in which they are constrained (Algorithm 3 Step 2), by means of a partitioning of Brownian bridge path space with an (arbitrary) increasing sequence, $\{a_\iota\}_{\iota\geq 0}$, $a_0=0$, which radiates outwards from the interval $[(x\wedge y),(x\vee y)]$, demarcating layers (recalling Definition 2). We term this particular layer construction the *Bessel layer*. For instance, the ιth Bessel layer is defined as follows,

$$\mathcal{I}_\iota=\bigl[(x\wedge y)-a_\iota,(x\vee y)+a_\iota\bigr]. \tag{68}$$

The (smallest) Bessel layer, $\mathcal{I}=\iota$, in which a particular Brownian bridge sample path is constrained can be simulated by retrospective Bernoulli sampling and inversion sampling [15], Section 2.1, as detailed in Algorithm 14 (where we denote by $S^{\gamma}_{k}(s,t,x,y,\ell,\upsilon)$ as the alternating Cauchy sequence whose limit as $k\to\infty$ is $\gamma^{\ell,\upsilon}_{s,t}(x,y)$). The CDF of ι can be written as follows (with reference to Theorem 3 and as shown in [6]),

$$\mathbb{P}(\mathcal{I}\leq\iota)=\mathbb{P}\bigl(W^{x,y}_{s,t}\in\bigl[(x\wedge y)-a_\iota,(x\vee y)+a_\iota\bigr]\bigr)=\gamma^{(x\wedge y)-a_\iota,(x\vee y)+a_\iota}_{s,t}(x,y). \tag{69}$$

Now, we require a method of simulating intermediate points (Algorithm 3 Step 4) from a Brownian bridge sample path restricted to remain in the Bessel layer simulated in Algorithm 14. In particular, denoting with D_ι the set of sample paths which are contained in the ιth Bessel layer we have,

$$D_\iota=L_\iota\cup U_\iota, \tag{70}$$

Algorithm 14 Simulation of a Brownian Bridge Bessel Layer [6]

```
1. Simulate $u\sim \mathrm{U}[0,1]$ and set $\iota=1$, $k=0$.
2. While $u\in(S^{\gamma}_{2k+1}(s,t,x,y,(x\wedge y)-a_\iota,(x\vee y)+a_\iota), S^{\gamma}_{2k}(s,t,x,y,(x\wedge y)-a_\iota,(x\vee y)+a_\iota))$, $k=k+1$.
3. If $u\geq S^{\gamma}_{2k}$ set $\iota=\iota+1$ and return to Step 2 else set $\mathcal{I}=\iota$ and end.
```

where

$$\begin{aligned} L_\iota &:= \bigl\{W_{[s,t]}\colon \hat{m}_{s,t} \in \bigl[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\bigr)\bigr\} \\ &\quad \cap \bigl\{W_{[s,t]}\colon \check{m}_{s,t} \in \bigl[(x \vee y), (x \vee y) + a_\iota\bigr]\bigr\}, \end{aligned} \tag{71}$$

$$\begin{aligned} U_\iota &:= \bigl\{W_{[s,t]}\colon \hat{m}_{s,t} \in \bigl[(x \wedge y) - a_\iota, (x \wedge y)\bigr]\bigr\} \\ &\quad \cap \bigl\{W_{[s,t]}\colon \check{m}_{s,t} \in \bigl((x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\bigr]\bigr\}. \end{aligned} \tag{72}$$

Directly simulating intermediate points from a sample path restricted to D_ι (denoted $\mathbb{D}_\iota$) is not possible. Instead (as proposed in [6]) we can propose sample paths from the mixture measure $\mathbb{B}_\iota := \hat{\mathbb{M}}_\iota/2 + \check{\mathbb{M}}_\iota/2$ ($\hat{\mathbb{M}}_\iota$ and $\check{\mathbb{M}}_\iota$ being the law induced by the restriction of $\mathbb{W}^{x,y}_{s,t}$ to the sets $\hat{M}_\iota$ and $\check{M}_\iota$, resp.) and accept them with probability given by the Radon–Nikodým derivative of $\mathbb{D}_\iota$ with respect to $\mathbb{B}_\iota$, where

$$\hat{M}_\iota = \bigl\{W_{[s,t]}\colon \hat{m}_{s,t} \in \bigl[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\bigr]\bigr\}, \tag{73}$$

$$\check{M}_\iota = \bigl\{W_{[s,t]}\colon \check{m}_{s,t} \in \bigl[(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\bigr]\bigr\}. \tag{74}$$

It was shown in [6] that $\mathbb{D}_\iota$ is absolutely continuous with respect to $\mathbb{B}_\iota$ with Radon–Nikodým derivative,

$$\frac{\mathrm{d}\mathbb{D}_\iota}{\mathrm{d}\mathbb{B}_\iota}(x) \propto \frac{\mathbb{1}(W \in D_\iota)}{1 + \mathbb{1}(W \in \hat{M}_\iota \cap \check{M}_\iota)}. \tag{75}$$

Sample paths can be drawn from $\mathbb{D}_\iota$ by proposing them from $\mathbb{B}_\iota := \hat{\mathbb{M}}_\iota/2 + \check{\mathbb{M}}_\iota/2$ and then accepting them with probability given by (75). For instance, with probability $1/2$ we sample from $\hat{\mathbb{M}}_\iota$ and accept with probability 1 if the sample path maximum is contained within the $(\iota - 1)$th Bessel layer or with probability $1/2$ if it is contained between the $(\iota - 1)$th and ιth Bessel layer (and with probability 0 otherwise). In practice we first simulate the sample path minimum $X_\tau = \hat{m}_{s,t}$ (or maximum $X_\tau = \check{m}_{s,t}$) as per Algorithm 11, and subsequently simulate any required intermediate points $\xi_1, \dots, \xi_\kappa$ from a Bessel bridge as per Algorithm 13. As we can only simulate our sample path at a finite collection of points we can't directly evaluate (75). However, we can obtain unbiased estimate and so simulate an event of this probability by application of Corollaries 4 and 5 and Lemmata 4 and 5 (letting $\chi_1, \dots, \chi_{\kappa+3}$ be the order statistics of $\{\xi_1, \dots, \xi_\kappa, s, \tau, t\}$),

$$\begin{aligned} \mathbb{P}_{\hat{\mathbb{M}}_\iota}(X \in D_\iota | X_{\chi_1}, \dots, X_{\chi_{\kappa+3}}) &= \mathbb{P}\bigl(X \in \bigl[(x \wedge y) - a_\iota, (x \vee y) + a_\iota\bigr] | X_{\chi_1}, \dots, X_{\chi_{\kappa+3}}\bigr) \\ &= \prod_{i=1}^{\kappa+2} \delta^{\hat{m},(x\vee y)+a_\iota}_{\chi_i,\chi_{i+1}}(X_{\chi_i}, X_{\chi_{i+1}}), \end{aligned} \tag{76}$$

$$\begin{aligned} \mathbb{P}_{\hat{\mathbb{M}}_\iota}(X \in \hat{M}_\iota \cap \check{M}_\iota | X_{\chi_1}, \dots, X_{\chi_{\kappa+3}}) &= \mathbb{P}_{\hat{\mathbb{M}}_\iota}(X \in D_\iota | X_{\chi_1}, \dots, X_{\chi_{\kappa+3}}) \\ &\quad - \prod_{i=1}^{\kappa+2} \delta^{\hat{m},(x\vee y)+a_{\iota-1}}_{\chi_i,\chi_{i+1}}(X_{\chi_i}, X_{\chi_{i+1}}). \end{aligned} \tag{77}$$

Algorithm 15 Layered Brownian Bridge Simulation (Bessel Approach) – Sampling X at times $\xi_1, \dots, \xi_\kappa$

```
1. Simulate $u_1, u_2 \sim \mathrm{U}[0,1]$, set $j = k = 0$.
2. Simulate Auxiliary Information (conditional on $I = \iota$),
   (a) If $u_1 \leq 1/2$ simulate minimum point $(\tau, \hat{m}_{s,t})$ and set $\ell_1 = \ell_2 = \hat{m}_{s,t}$, $\upsilon_1 = (x \vee y) + a_{\iota-1}$ and $\upsilon_2 = (x \vee y) + a_\iota$.
   (b) If $u_1 > 1/2$ simulate maximum $(\tau, \check{m}_{s,t})$ and set $\ell_1 = (x \wedge y) - a_{\iota-1}$, $\ell_2 = (x \wedge y) - a_{\iota-1}$ and $\upsilon_1 = \upsilon_2 = \check{m}_{s,t}$.
3. Simulate intermediate times $X_{\xi_1}, \dots, X_{\xi_\kappa}$ from a Bessel Bridge conditional on $X_\tau$.
4. While $u_2 \in (\prod_{i=1}^{\kappa+2} S^\delta_{2j+1}(\ell_1, \upsilon_1), \prod_{i=1}^{\kappa+2} S^\delta_{2j}(\ell_1, \upsilon_1))$, $j = j + 1$,
   (a) If $u_2 \leq \prod_{i=1}^{\kappa+2} S^\delta_{2j+1}(\ell_1, \upsilon_1)$, then accept sample path.
   (b) If $u_2 \geq \prod_{i=1}^{\kappa+2} S^\delta_{2j}(\ell_1, \upsilon_1)$ while $u_2 \in (\prod_{i=1}^{\kappa+2} S^\delta_{2k+1}(\ell_2, \upsilon_2), \prod_{i=1}^{\kappa+2} S^\delta_{2k}(\ell_2, \upsilon_2))$, $k = k + 1$,
      i. If $u_2 \leq \prod_{i=1}^{\kappa+2} S^\delta_{2k+1}(\ell_2, \upsilon_2)$, then with probability $1/2$ accept sample path, else return to Step 1.
      ii. If $u_2 \geq \prod_{i=1}^{\kappa+2} S^\delta_{2k}(\ell_2, \upsilon_2)$, then reject sample path and return to Step 1.
5. Discard or Retain Auxiliary Information.
```

As both (76) and (77) are probabilities which can be represented as a function of δ probabilities, events of this probability can be simulated by retrospective Bernoulli sampling (as per Corollaries 1, 2 and Algorithm 11). The synthesis of the above approach for simulating a Brownian bridge conditional on the Bessel layer simulated in Algorithm 14 (i.e., conducting Algorithm 3 Step 4) leads to Algorithm 15.

Upon accepting a proposed sample path skeleton within the UEA (as simulated by Algorithm 14 and Algorithm 15 and so satisfying Principles 1 and 2), we need to be able to simulate the sample path at further times (Algorithm 3 Step 2) in order to satisfy Principle 3. Any further simulation is conditional on information obtained constructing the sample path skeleton. In particular, our sample path belongs to D_ι (by Algorithm 14), the sample path minimum (or maximum) belongs to a particular interval (w.p. 1, as a consequence of the mixture proposal in (73), (74)), we have simulated the sample path minimum (or maximum) (either $X_\tau = \hat{m}_{s,t}$ or $X_\tau = \check{m}_{s,t}$ by Algorithm 11) and skeletal points $(X_{\xi_1}, \dots, X_{\xi_\kappa})$ and finally we have simulated whether the sample path maximum (or minimum) is contained in the first $(\iota - 1)$ Bessel layers or in the ιth Bessel layer (by evaluating the Radon–Nikodým derivative in (75) by means of (76) and (77)). In summary, we have four possible sets of conditional information for our sample path,

$$\begin{aligned} S_1 := \bigl\{ & X_s, X_t, X \in D_\iota, \hat{m}_{s,t} \in \bigl[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\bigr], X_\tau = \hat{m}_{s,t}, \\ & X_{\xi_1}, \dots, X_{\xi_\kappa}, \check{m}_{s,t} \in \bigl[(x \vee y), (x \vee y) + a_{\iota-1}\bigr]\bigr\}, \end{aligned} \tag{78}$$

$$\begin{aligned} S_2 := \bigl\{ & X_s, X_t, X \in D_\iota, \hat{m}_{s,t} \in \bigl[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\bigr], X_\tau = \hat{m}_{s,t}, \\ & X_{\xi_1}, \dots, X_{\xi_\kappa}, \check{m}_{s,t} \in \bigl[(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\bigr]\bigr\}, \end{aligned} \tag{79}$$

$$\begin{aligned} S_3 := \big\{&X_s, X_t, X \in D_\iota, \check{m}_{s,t} \in \big[(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\big], X_\tau = \check{m}_{s,t}, \\ &X_{\xi_1}, \dots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \big[(x \wedge y) - a_{\iota-1}, (x \wedge y)\big]\big\}, \end{aligned} \tag{80}$$

$$\begin{aligned} S_4 := \big\{&X_s, X_t, X \in D_\iota, \check{m}_{s,t} \in \big[(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\big], X_\tau = \check{m}_{s,t}, \\ &X_{\xi_1}, \dots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \big[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\big]\big\}. \end{aligned} \tag{81}$$

The difficulty in simulating the process at further intermediate times conditional on the above is that information pertaining to the sample path minimum and maximum induces a dependency between the sub-interval in which we want to simulate an intermediate point, and all other sub-intervals. An additional complication arises as we know precisely the minimum (or maximum) of the sample path, so the law we need to simulate further points from is that of a Bessel bridge conditioned to remain in a given interval.

However, the minimum (or maximum) simulated in Algorithm 15 Step 2 is auxiliary sample path information (as in (67)) and doesn't constitute an essential part of the exact algorithm skeleton, so can be discarded. Furthermore, information regarding the sample path minimum and maximum is sufficient in determining an interval for the entire sample path. As such, reconsidering S_1 (S_2, S_3, S_4 can be similarly considered) we have,

$$\begin{aligned} \tilde{S}_1 := \big\{&X_s, X_t, X_{\xi_1}, \dots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \big[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\big], \\ &\check{m}_{s,t} \in \big[(x \vee y), (x \vee y) + a_{\iota-1}\big]\big\}. \end{aligned} \tag{82}$$

Now, to remove the induced dependency between sub-intervals of time we can simulate, for each sub-interval of time, an interval of path space in which the sample path minimum and maximum is constrained as outlined in Section 8.3 and Algorithm 20. Further intermediate points can then be simulated as outlined in Section 8.5.

7.2. Localised Approach

The *Localised Approach* is based on the layered Brownian bridge construction found in the *Localised Exact Algorithm* (LEA) originally proposed in [14,17]. The LEA is a modified construction of the exact algorithm based on the mathematical framework of EA3 (see [6]). We outline the LEA only to highlight to the reader the aspects of its construction which would lead to computational challenges if implemented within the context which we consider in this paper (in particular, significant computation is required in order to satisfy Principle 3).

The key notion in the Localised approach is that rather than proposing sample path skeletons from $\mathbb{Z}^x_{0,T}$ (where the end point $X_T =: y \sim h$ is first simulated), the interval to be simulated ($[0, T]$) can be instead broken into a number of *bounded* segments (as in (23)). Each segment is successively simulated by means of simulating the first hitting time, τ, of a Brownian motion proposal sample path (as outlined in [12]) of some user specified boundary symmetric around its start point (e.g., if $X_0 = x$ with boundary θ then $\tau := \inf\{s : X_s \notin [x - \theta, x + \theta]\}$), and simulating and accepting a sample path skeleton conditional on the simulated boundary (with a suitable modification of the acceptance probability to account for the modified proposal measure).

The benefit of the Localised approach is that simulating the first hitting time of a boundary acts as a layer for the bounded segment (i.e., $\forall u \in [0, \tau]$, $X_u(\omega) \in [x-\theta, x+\theta]$) and so $\phi(X_{0,\tau})$ is conditionally bounded (as per Result 4) and a bound can be found for $A(X_\tau)$ in (8). As such it is possible to bound the Radon–Nikodým derivative without the need for Condition 5, however the acceptance rate of proposal sample paths can be low as each component of the Radon–Nikodým derivative needs to be bounded (the incongruity being that this can be particularly problematic in the case where the diffusion doesn't satisfy Condition 5). Moreover, as with the UJEA and Adaptive Unbounded Jump Exact Algorithm (AUJEA) this approach to simulating sample path skeletons can result in simulating skeletons for intervals exceeding that required (which is computationally wasteful), further complicated by the need to specify the boundary θ. Furthermore (as discussed in [20]), this methodology can't be used to simulate *conditioned* diffusion and jump diffusion sample path skeletons (sample paths conditioned to hit some specified end point), whereas the methodology developed elsewhere in this paper can be directly extended to this setting (see [31], Chapter 5). Finally, unlike the Bessel approach, the minimum or maximum that is simulated forms part of the skeleton and so cannot be discarded. As such, the demarginalisation strategy taken in Section 7.1 in order to extend the UEA with the Bessel approach for simulating layered Brownian bridges to satisfy Principle 3 can't be conducted.

8. Adaptive Layered Brownian Bridge Constructions

In Section 3.2, we proposed the Adaptive Unbounded Exact Algorithm (AUEA) (Algorithm 4) as an alternative to the UEA (Algorithm 3). In this section, we outline how to simulate finite dimensional skeletons of layered Brownian bridges for use within the AUEA (and by extension the BJEA (Algorithm 5) and AUJEA (Algorithm 7)). In particular, we present new results for simulating an initial *intersection layer* (Algorithm 4 Step 2 – Section 8.1), intermediate points conditional on the layer (Algorithm 4 Step 3.1.2 – Section 8.2) and finally, new intersection layers for each sub-interval created by the intermediate point (Algorithm 4 Step 3.1.4 – Section 8.3).

We use the results we present in Sections 8.1–8.3 to outline novel layered Brownian bridge constructions in Section 8.5 which can used within the AUEA, all of which satisfy Principles 1, 2 and 3.

8.1. Simulating an Initial Intersection Layer

Upon simulating a proposal Brownian bridge layer as per Algorithm 14 in Section 7.1, we know that our entire Brownian bridge sample path is contained within the ιth Bessel layer, but is not contained within the $(\iota-1)$th Bessel layer. Simulating sample path intermediate points is complicated by this conditional information (and as discussed in Section 7, it is not possible to simulate intermediate points directly). The novel approach we take in this paper is to simulate further layer information regarding the minimum and maximum of the proposed sample path (which together provide a sample path layer). To achieve this recall (with reference to Section 3 and (70), (71), (72)) that, having simulated a layer for our proposal Brownian bridge sample path as per Algorithm 14, we know the sample path is restricted to the layer D_ι. We can then simply decompose the set D_ι into a disjoint union and simulate to which element our sample path belongs,

$$D_\iota = L_\iota \cup U_\iota = \underbrace{(L_\iota \cap U_\iota)}_{D_{\iota,1}} \uplus \underbrace{\big(U_\iota^{\mathrm{C}} \cap L_\iota\big)}_{D_{\iota,2}} \uplus \underbrace{\big(L_\iota^{\mathrm{C}} \cap U_\iota\big)}_{D_{\iota,3}}. \tag{83}$$

This decomposition corresponds to the sample path attaining the ιth Bessel layer at both its minimum and maximum ($D_{\iota,1}$) or its minimum ($D_{\iota,2}$) or its maximum ($D_{\iota,3}$). We can simulate to which set our sample path belongs by application of the following results and Algorithm 16. Recalling the definition of a layer from Definition 2, we term this particular layer construction the *Intersection Layer*.

Theorem 8 (Initial intersection layer). *The probability a Brownian bridge sample path is in $D_{\iota,1}$, given it is in D_ι, can be represented as follows (denoting $\ell\!\downarrow := (x \wedge y) - a_\iota$, $\ell\!\uparrow := (x \wedge y) - a_{\iota-1}$, $\upsilon\!\downarrow := (x \vee y) + a_{\iota-1}$, $\upsilon\!\uparrow := (x \vee y) + a_\iota$ and $\tilde{\beta}(s,t,x,y) := \beta(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,\upsilon\!\downarrow,\upsilon\!\uparrow) + \beta(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,(x \vee y),\upsilon\!\downarrow) + \beta(s,t,x,y,\ell\!\uparrow,(x \wedge y),\upsilon\!\downarrow,\upsilon\!\uparrow)$),*

$$p_{D_{\iota,1}} := \mathbb{P}(D_{\iota,1} | D_\iota, W_s = x, W_t = y) = \frac{\beta(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,\upsilon\!\downarrow,\upsilon\!\uparrow)}{\tilde{\beta}(s,t,x,y)}. \tag{84}$$

Proof. Follows by Bayes rule, Theorem 7 and the decomposition of D_ι in (83). □

Corollary 8. *Events of probability $p_{D_{\iota,1}}$ can be simulated by retrospective Bernoulli sampling (as per Corollaries 1, 2 and Algorithm 11), noting that $p_{D_{\iota,1}}$ is a function of β probabilities, defining*

$$\tilde{S}_k^\beta(s,t,x,y) := S_k^\beta(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,\upsilon\!\downarrow,\upsilon\!\uparrow) + S_k^\beta\big(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,(x \vee y),\upsilon\!\downarrow\big) \\ + S_k^\beta\big(s,t,x,y,\ell\!\uparrow,(x \wedge y),\upsilon\!\downarrow,\upsilon\!\uparrow\big),$$

and using the following sequence,

$$S_k^{D(\iota,1)} := \frac{S_k^\beta(s,t,x,y,\ell\!\downarrow,\ell\!\uparrow,\upsilon\!\downarrow,\upsilon\!\uparrow)}{\tilde{S}_k^\beta(s,t,x,y)}. \tag{85}$$

Noting that by symmetry we have $p_{D_{\iota,2}} := \mathbb{P}(D_{\iota,2} | D_\iota, W_s = x, W_t = y) = \mathbb{P}(D_{\iota,3} | D_\iota, W_s = x, W_t = y) =: p_{D_{\iota,3}}$ and furthermore $p_{D_{\iota,2}} + p_{D_{\iota,3}} = 1 - p_{D_{\iota,1}}$ it is possible to determine to which disjoint set ($D_{\iota,1}$. $D_{\iota,2}$ or $D_{\iota,3}$) our sample path belongs by direct application of Theorem 8, Corollary 8 and the following Algorithm 16.

8.2. Simulating Intersection Layer Intermediate Points

Having simulated an intersection layer we require a sampling scheme for simulating the conditional Brownian bridge at some intermediate time $q \in (s,t)$. As shown in [8], the density of the

Algorithm 16 Simulation of an Initial Brownian Bridge Intersection Layer

```
1. Simulate layer $\mathcal{I} = \iota$ as per Algorithm 14, simulate $u \sim \mathrm{U}[0,1]$ and set $k = 0$.
2. While $u \in (S^{D(\iota,1)}_{2k+1}, S^{D(\iota,1)}_{2k})$, $k = k + 1$.
3. If $u \leq S^{D(\iota,1)}_{2k+1}$, then set $D_\iota = D_{\iota,1}$.
4. If $u \geq S^{D(\iota,1)}_{2k}$, then with probability 0.5 set $D_\iota = D_{\iota,2}$ else set $D_\iota = D_{\iota,3}$.
```

sample path at the intermediate time point q can be written as follows (where μ_w and σ^2_w denote the mean and variance of a Brownian bridge as in Section 6.1),

$$\pi(w) := \mathbb{P}\big(W_q = w | W_s, W_t, \hat{m}_{s,t} \in \big[\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}\big], \check{m}_{s,t} \in \big[\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\big]\big) \tag{86}$$

$$\propto \rho\big(s, q, t, x, w, y, \ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\big) \cdot \mathrm{N}\big(w; \mu_w, \sigma^2_w\big). \tag{87}$$

A method of simulating from $\pi(w)$ was outlined in [8] based on inversion sampling and numerical methods, however, this scheme is formally inexact and given particular parameter values can be computationally extremely inefficient. We provide a number of alternative schemes which are exact.

In Section 8.2.1, we present a method of simulating from (87) by finding a bound constructed from a mixture of Normal densities which can be easily simulated from and conducting rejection sampling. It transpires that this scheme is typically highly computationally efficient, however, for a small number of parameter values the acceptance rate of the rejection sampler is very low. As such, in Section 8.2.2, we present an alternate rejection sampling scheme which exploits the known Lipschitz constants of the bounding sequence in (87) to construct an arbitrarily tight bound of the target density. This, however, comes at some computational expense, so we advocate using some mixture of these two approaches (which we discuss later in Sections 8.2.4 and 8.5). Finally, for completeness, in Section 8.2.3 we construct a third scheme inspired by the Bessel layer constructions found in Section 7.1 and [6]. This third scheme provides some insight into how the different layered Brownian bridge constructions of Section 7 and Section 8 relate to one another.

8.2.1. *Bounding Cauchy Sequence Approach*

Here we show that it is possible to extend [8], and simulate from $\pi(w)$ exactly by means of composition sampling (see [33]) and rejection sampling. We will begin by considering the upper convergent bounding sequence of $\rho(w)$ ($\forall k \in \mathbb{Z}_{\geq 0}$ we have $\rho(w) \leq S^{\rho}_{2k}(w)$ and $\lim_{k\to\infty} S^{\rho}_{2k}(w) = \rho(w)$), where for conciseness we additionally denote $\rho(w) := \rho(s, q, t, x, w, y, \ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t})$ (as in this setting we have $s, q, t, x, y, \ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}$ and $\upsilon^{\uparrow}_{s,t}$ are fixed). Decomposing (63) into its elementary form in terms of $\bar{\varsigma}$ and $\bar{\varphi}$ (see (49) and (50), resp.) yields $K = 64(k+1)^2$ of these elementary terms.

Recalling that $\varsigma^{\ell,\upsilon}_{s,t}(j; x, y) := \bar{\varsigma}^{\ell,\upsilon}_{s,t}(j; x, y) + \bar{\varsigma}^{-\ell,-\upsilon}_{s,t}(j; -x, -y)$ and $\varphi^{\ell,\upsilon}_{s,t}(j; x, y) := \bar{\varphi}^{\ell,\upsilon}_{s,t}(j; x, y) + \bar{\varphi}^{-\ell,-\upsilon}_{s,t}(j; -x, -y)$ it can be shown that each of the functions $\bar{\varsigma}$ and $\bar{\varphi}$ has the structural form $\exp(a_i + b_i w)$, with known coefficients a_i and b_i (see the Appendix for further details). As such, we can find a bound for our unnormalised target density (87) as follows (the $c_i \in \{0,1\}$ determine the sign of each density contribution),

$$\pi(w) \propto \rho(w) \cdot \mathrm{N}\big(w; \mu_w, \sigma_w^2\big) \le S_{2k}^{\rho}(w) \cdot \mathrm{N}\big(w; \mu_w, \sigma_w^2\big) \tag{88}$$

$$= \sum_{i=1}^{K} \big[(-1)^{c_i} \cdot \exp\{a_i + b_i w\} \cdot \mathbb{1}\big(w \in [\ell_i, \upsilon_i]\big) \cdot \mathrm{N}\big(w; \mu_w, \sigma_w^2\big)\big] \tag{89}$$

$$\begin{aligned} &= \sum_{i=1}^{K} \big[\underbrace{(-1)^{c_i} \cdot \exp\big\{a_i + \mu_w b_i + b_i \sigma_w^2/2\big\}}_{\omega_i :=} \\ &\qquad \times \mathbb{1}\big(w \in [\ell_i, \upsilon_i]\big) \cdot \underbrace{\mathrm{N}\big(w; \mu_w + b_i \sigma_w^2, \sigma_w^2\big)}_{\mathrm{N}(w;\mu_i,\sigma_w^2):=}\big]. \end{aligned} \tag{90}$$

Here we have a mixture of positively and negatively weighted truncated Normal densities (with common variance). Although each truncated Normal in the mixture is unique (due to the truncation points), a large proportion of them will have common location parameter. We exploit this by partitioning the interval that provides support for the target density (87) into sections corresponding to the truncation points (in particular, we consider the partitioning $\{[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}], [\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}], [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]\}$ which we denote by $j \in \{1,2,3\}$, resp.). As a consequence, the resulting mixture density has a number of positive and negative elements which cancel each other out (i.e., they can be *netted* from one another). Defining $\omega_{i,j}$ as the weight associated with the jth partition of the ith truncated Normal density of (90), and $\omega_{i,j}^{+} := (\omega_{i,j} \vee 0)$, we can find an upper bound by solely considering the mixture formed from the components with positive weights,

$$\begin{aligned} \pi(w) &\le \sum_{i=1}^{K} \omega_i \cdot \mathrm{N}\big(w; \mu_i, \sigma_w^2\big) \cdot \mathbb{1}\big(w \in [\ell_i, \upsilon_i]\big) \\ &\qquad \times \big[\mathbb{1}\big(w \in \big[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}\big]\big) + \mathbb{1}\big(w \in \big[\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}\big]\big) + \mathbb{1}\big(w \in \big[\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}\big]\big)\big] \end{aligned} \tag{91}$$

$$\begin{aligned} &\le \sum_{i=1}^{K} \mathrm{N}\big(w; \mu_i, \sigma_w^2\big) \mathbb{1}\big(w \in [\ell_i, \upsilon_i]\big) \\ &\qquad \times \big[\omega_{i,1}^{+} \mathbb{1}\big(w \in \big[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}\big]\big) + \omega_{i,2}^{+} \mathbb{1}\big(w \in \big[\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}\big]\big) + \omega_{i,3}^{+} \mathbb{1}\big(w \in \big[\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}\big]\big)\big] \end{aligned} \tag{92}$$

$$=: S_{2k}^{\rho,+}(w) \cdot \mathrm{N}\big(w; \mu_i, \sigma_w^2\big). \tag{93}$$

By application of composition sampling (see [33]) we can simulate from the probability density proportional to $S_{2k}^{\rho,+}(w) \cdot \mathrm{N}(w; \mu_w, \sigma_w^2)$ by first choosing one of the truncated Normal densities partitioned on the interval $[\mathcal{L}, \mathcal{U}]$ with probability proportional to,

$$\omega_{i,j}^{+} \cdot \big[\Phi\big(\mathcal{U} | \mu_i, \sigma_w^2\big) - \Phi\big(\mathcal{L} | \mu_i, \sigma_w^2\big)\big]. \tag{94}$$

Algorithm 17 Simulation of Intersection Layer Intermediate Points (Bounded Cauchy Sequence Approach)

```
1. Simulate $u \sim \mathrm{U}[0,1]$ and set $j = 1$.
2. Simulate $w \sim S^{\rho,+}_{2k}(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)/Z_D$ for some $k \in \mathbb{Z}_{\geq 0}$.
3. While $u \in (\frac{S^{\rho}_{2j+1}(w)}{S^{\rho,+}_{2k}(w)}, \frac{S^{\rho}_{2j}(w)}{S^{\rho,+}_{2k}(w)})$, $j = j + 1$.
4. If $u \leq \frac{S^{\rho}_{2j+1}(w)}{S^{\rho,+}_{2k}(w)}$ then accept else reject.
```

As $w \sim S^{\rho,+}_{2k}(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)/Z_D$ and we require $w \sim \rho(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)/Z_T$ (where Z_T and Z_D denote the normalising constants of the target and dominating densities, resp., noting that the rejection sampling bound $M = Z_D/Z_T$) we accept this draw with probability,

$$P = \frac{\rho(w) \cdot \mathrm{N}(w|\mu_w, \sigma^2_w)/Z_T}{M \cdot S^{\rho,+}_{2k}(w) \cdot \mathrm{N}(w|\mu_w, \sigma^2_w)/Z_D} = \frac{\rho(w)}{S^{\rho,+}_{2k}(w)} \leq 1. \tag{95}$$

Events of probability P can be simulated by retrospective Bernoulli sampling (as per Corollaries 1, 2 and Algorithm 11), noting that P is a function of $\rho(w)$. The complete rejection sampler is presented in Algorithm 17.

8.2.2. *Lipschitz Approach*

Simulating intermediate points as per Algorithm 17 is (typically) highly efficient as $S^{\rho,+}_{2}(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)$ typically tightly bounds $\pi(w)$ (as noted in [8]). If this is not the case (which occurs for a small number of parameter configurations), then sampling from the bounding density with $k > 1$ isn't usually effective as $S^{\rho,+}_{2k}(w)$ is only formed by the positive netted components of $S^{\rho}_{2k}(w)$. In this section we propose an alternative scheme in which we exploit the known Lipschitz constants of the bounding sequence in (87) to construct a tight bound of the target density.

If the rejection sampling scheme proposed in Section 8.2 is not efficient then this implies that $S^{\rho}_{2}(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)$ does not tightly bound $\pi(w)$. In this case, the natural question to ask is at what level the Cauchy sequence approximation ($S^{\rho}_{2k}(w)$) of $\rho(w)$ needs to be evaluated such that $S^{\rho}_{2k}(w) \cdot \mathrm{N}(w; \mu_w, \sigma^2_w)$ does form a tight bound of $\pi(w)$. To address this we note that in analogous form to Section 8.2.1 it is possible to also find a *lower* bound of the target density,

$$S^{\rho}_{2k+1}(w) \cdot \mathrm{N}\big(w; \mu_w, \sigma^2_w\big) \leq \pi(w) \leq S^{\rho}_{2k}(w) \cdot \mathrm{N}\big(w; \mu_w, \sigma^2_w\big). \tag{96}$$

The lower bound of the target density also has the form of a mixture of positively and negatively weighted Normal densities with known parameter values (recall the upper bound comprises $K\uparrow = 64(k+1)^2$ terms, similarly the lower bound comprises $K\downarrow = 64(k+1)^2 - 48$ terms). As such, the normalising constants of the upper and lower bounds of the target density can be calculated and this information used to determine whether the upper bound tightly bounds the target density. In particular, we advocate evaluating the alternating Cauchy sequence $S^{\rho}_{k}(w)$ until such time that it exceeds some user specified threshold,

$$T_Z \le \frac{Z^{\rho}_{2k+1}(w)}{Z^{\rho}_{2k}(w)} := \left[\int_{\ell^{\downarrow}_{s,t}}^{\upsilon^{\uparrow}_{s,t}} S^{\rho}_{2k+1}(w) \cdot \mathrm{N}\big(w;\mu_w,\sigma^2_w\big)\,\mathrm{d}w\right] \Big/ \left[\int_{\ell^{\downarrow}_{s,t}}^{\upsilon^{\uparrow}_{s,t}} S^{\rho}_{2k}(w) \cdot \mathrm{N}\big(w;\mu_w,\sigma^2_w\big)\,\mathrm{d}w\right]. \tag{97}$$

Upon finding an appropriately tight upper bound, a subset of the positive and negative Normal densities can be netted from one another leaving the following bounding density form (as argued in Section 8.2 and shown in (91)),

$$\begin{aligned} \pi(w) \le {} & \sum_{i=1}^{K\uparrow} \mathrm{N}\big(w;\mu_i,\sigma^2_w\big) \cdot \mathbb{1}\big(w \in [\ell_i, \upsilon_i]\big) \\ & \times \big[\omega_{i,1}\mathbb{1}\big(w \in \big[\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}\big]\big) + \omega_{i,2}\mathbb{1}\big(w \in \big[\ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}\big]\big) \\ & + \omega_{i,3}\mathbb{1}\big(w \in \big[\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\big]\big)\big] \\ =: {} & g(w). \end{aligned} \tag{98}$$

For any given interval $[q,r]$ (where $q<r$), it is possible to explicitly calculate for each of the contributing Normal densities (e.g., $\mathrm{N}(w;\mu_i,\sigma^2_w)$) the local Lipschitz constant (we denote $I := [\mu_i - \sigma_w, \mu_i + \sigma_w] \cap [q,r]$),

$$\begin{aligned} \alpha_i(q,r) := {} & \sup_{w\in[q,r]} \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}\big(w;\mu_i,\sigma^2_w\big) \\ = {} & \mathbb{1}(I \ne \varnothing) \cdot \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}\big(\mu_i - \sigma_w;\mu_i,\sigma^2_w\big) \\ & + \mathbb{1}(I = \varnothing) \cdot \max\left\{\frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}\big(q;\mu_i,\sigma^2_w\big), \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}\big(r;\mu_i,\sigma^2_w\big)\right\}. \end{aligned} \tag{99}$$

As such, it is possible to find for the bounding density ($g(w)$ in (98)) the local Lipschitz constant for the interval $[q,r]$ (where α is set to zero when considering an interval of zero length),

$$\begin{aligned} \sup_{u,v\in[q,r]} \frac{g(u)-g(v)}{u-v} \le {} & \sum_{j=1}^{K\uparrow} \big[|\omega_{j,1}|\alpha_j\big(q \vee \ell^{\downarrow}_{s,t}, r \wedge \ell^{\uparrow}_{s,t}\big) + |\omega_{j,2}|\alpha_j\big(q \vee \ell^{\uparrow}_{s,t}, r \wedge \upsilon^{\downarrow}_{s,t}\big) \\ & + |\omega_{j,3}|\alpha_j\big(q \vee \upsilon^{\downarrow}_{s,t}, r \wedge \upsilon^{\uparrow}_{s,t}\big)\big] \\ =: {} & \beta(q,r), \end{aligned} \tag{100}$$

and consequently, having evaluated the density at $g(q)$ and $g(r)$, we can find a bound for the upper bound of the target density for the interval $[q,r]$ (noting that the line $y = g(q) + \beta t$

intersects the line $y = g(r) + \beta(r-q) - \beta t$ at $t = [g(r) - g(q) + \beta(r-q)]/2\beta \in [q,r]$),

$$\sup_{w \in [q,r]} g(w) \leq g(r) + \beta(q,r) \cdot t = \frac{g(r)+g(q)}{2} + \beta(q,r) \cdot \frac{r-q}{2} =: M(q,r). \tag{101}$$

As the support of the target density $\pi(w)$ is contained within the interval $[\ell^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]$, if we construct a suitably fine mesh on this interval (for simplicity, we assume a mesh of size N with regular interval size $\Delta := (\upsilon^{\uparrow}_{s,t} - \ell^{\downarrow}_{s,t})/N$), we can find a piecewise uniform bound of this density with which to conduct rejection sampling,

$$g(w) \leq \sum_{i=1}^{N} \mathbb{1}\big(w \in \big[\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta\big]\big) \cdot M\big(\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta\big). \tag{102}$$

As in (97), we can calculate the normalising constant of this bounding density, so we advocate choosing the size of the mesh to be at least as fine as the following user specified threshold,

$$T_M \leq \frac{Z^{\rho}_{2k}(w)}{Z^{N}_{M}(w)} := \frac{Z^{\rho}_{2k}(w)}{\sum_{i=1}^{N} \Delta \cdot M(\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta)}. \tag{103}$$

We present the synthesis of the above argument in Algorithm 18. Clearly the acceptance rate of Algorithm 18 is at least $T_Z \cdot T_M$ and furthermore is more robust to different parameter values than the Cauchy sequence approach outlined in Algorithm 17, as given sufficient computation an arbitrarily tight bound of the target density can be found with which to conduct rejection sampling. In Figure 11, we present an example of a set of parameter values in which the acceptance rate under the Cauchy sequence approach was less than 10^{-8}, whereas with the approach outlined in Algorithm 18 a small mesh of size 20 was sufficient to find a tight upper bound of the target density.

8.2.3. *Bessel Approach*

An alternative scheme to simulate a single intermediate point from (86) is to apply an analogous decomposition of the law of the sample path as was constructed in the Bessel approach for

Algorithm 18 Simulation of Intersection Layer Intermediate Points (Lipschitz Approach)

```
1. Set $k = 0$, $N = 0$.
2. While $T_Z \geq \frac{Z^{\rho}_{2k+1}(w)}{Z^{\rho}_{2k}(w)}$, $k = k + 1$.
3. While $T_M \geq \frac{Z^{\rho}_{2k}(w)}{Z^{N}_{M}(w)}$, increase $N$.
4. Simulate mesh interval $i$ with probability $\Delta \cdot M(\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta)/Z^{N}_{M}$.
5. Simulate $w \sim \mathrm{U}[\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta]$, $u \sim \mathrm{U}[0, M(\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta)]$ and
   set $j = k$.
6. While $u \in (S^{\rho}_{2j+1}(w), S^{\rho}_{2j}(w))$, $j = j + 1$.
7. If $u \leq S^{\rho}_{2j+1}(w)$ then accept, else reject and return to Step 5.
```

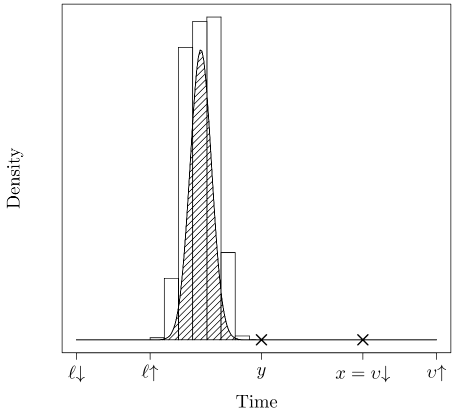


Figure 11. Density of intersection layer intermediate point overlaid with piecewise constant bound calculated using a mesh of size 20 over the interval $[\ell\downarrow, \upsilon\uparrow]$ and the corresponding local Lipschitz constants.

layered Brownian bridge outlined in Section 7.1. Recall in Section 7.1 that in order to simulate intermediate points from the sample path that we first simulated the minimum or maximum of the sample path conditional on the Bessel layer (with probability $1/2$) and then simulated proposal intermediate points from the law of a Bessel bridge. The proposal intermediate points were then accepted if the sample path remained in the appropriate Bessel layer.

We apply the same notion described in Section 7.1, however, a modification has to be made to the acceptance probability as the intersection layer provides more precise information regarding the interval in which both the minimum and maximum is contained than the Bessel layer. In particular, if we have simulated intersection layer $D_{\iota,1}$ then with probability $1/2$ we propose the auxiliary minimum (else maximum) in the ιth layer and then only accept the proposal sample path if the sample path maximum (else minimum) is contained between the $(\iota - 1)$th and ιth Bessel layer. In the case where we have either simulated intersection layer $D_{\iota,2}$ or $D_{\iota,3}$ then with probability $1/2$ we propose the auxiliary minimum (else maximum) in the ιth (else $(\iota - 1)$th) layer and then only accept the proposal sample path if the sample path maximum (else minimum) is contained within the $(\iota - 1)$th (else $(\iota - 1)$th) Bessel layer. The synthesis of the above argument which is based on Section 7.1 can be found in Algorithm 19.

Although given particular parameter values in (86) the Bessel approach can computationally outperform the Cauchy sequence approach or Lipschitz approached described in Section 8.2.1 and Section 8.2.2, respectively, as we will discuss in Section 8.2.4 we advocate a mixture of those two approaches instead as the Bessel approach can be particularly inefficient whenever a large intersection layer is proposed.

8.2.4. *Implementational Considerations – Recommended Approach*

In Sections 8.2.1, 8.2.2 and 8.2.3, we have outlined three separate approaches and algorithms for simulating from the density of a conditional Brownian bridge at some intermediate time $q \in (s, t)$ (87). As each of these algorithms is a rejection sampler in which independent proposals are drawn and then accepted or rejected, if a proposal is rejected one can change to another

Algorithm 19 Simulation of Intersection Layer Intermediate Points (Bessel Approach)

```
1. Simulate $u_1, u_2 \sim \mathrm{U}[0, 1]$, set $j = k = 0$.
2. Simulate Auxiliary Information as per Algorithm 13,
    (a) If $u_1 \leq 1/2$ simulate minimum $(\tau, X_\tau = \hat{m} \in [\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}])$ setting $c\downarrow := \upsilon^{\downarrow}_{s,t}$ and $c\uparrow := \upsilon^{\uparrow}_{s,t}$.
    (b) If $u_1 > 1/2$ simulate maximum $(\tau, X_\tau = \check{m} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}])$ setting $c\downarrow := \ell^{\downarrow}_{s,t}$ and $c\uparrow := \ell^{\uparrow}_{s,t}$.
3. Simulate $X_q$ from a Bessel bridge conditional on $X_\tau$ as per Algorithm 12.
4. While $u_2 \in (\prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}(X_\tau, c\downarrow), \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}(X_\tau, c\downarrow))$, $j = j + 1$,
    (a) If $u_2 \leq \prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}(X_\tau, c\downarrow)$ then reject sample path and return to Step 1.
5. While $u_2 \in (\prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}(X_\tau, c\uparrow), \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}(X_\tau, c\uparrow))$, $k = k + 1$,
    (a) If $u_2 \geq \prod_{i=1}^{\kappa+2} S^{\delta}_{2k}(X_\tau, c\uparrow)$ then reject sample path and return to Step 1.
6. Discard Auxiliary Information.
```

of these algorithms without introducing any bias. As empirically Algorithm 17 is highly computationally efficient compared to the other algorithms, but for a small number of parameters values has a very low acceptance rate, we suggest that on implementation a user specified threshold number of potential proposals from this algorithm is chosen (say N). If after the first N proposals there has been no acceptance, then this suggests that the acceptance rate for the particular parameter configuration is low. As such, at this point we suggest switching to Algorithm 18 which requires a significant initial computational effort to find a tight bound to the target density, but, the acceptance rate will be higher and the algorithm more robust to different parameter values than Algorithm 17. This particular combination of algorithms is advocated as Algorithm 19 can be inefficient whenever a large intersection layer is proposed.

8.3. Dissecting an Intersection Layer

Upon simulating intermediate points of a Brownian bridge sample path conditional on an intersection layer (e.g., in Section 8.2), simulating further intermediate points in a sub-interval between any two existing consecutive points is more complicated as there is a dependency between all sub-intervals (induced by the intersection layer). To simplify this problem, we can *dissect* an intersection layer into separate intersection layers for each pair of consecutive points by considering all possible dissections and unbiasedly simulating which one of these occurs.

To guide intuition, we first consider the case where we have a single intermediate point ($W_q = w$) within an existing intersection layer ($W_s = x$, $W_t = y$, $\hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}]$, $\check{m}_{s,t} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]$) and we want to simulate separate intersection layers for the intervals $[s, q]$ and $[q, t]$ conditional on the known intersection layer and the simulated point. We begin by noting that the simulated point provides further detail on the interval in which the minimum and maximum lies. In particular, if $w \in [\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}]$ we have that $\hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, w]$ and similarly if $w \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]$ then we have that $\check{m}_{s,t} \in [w, \upsilon^{\uparrow}_{s,t}]$. As such we denote $\ell^{\uparrow*}_{s,t} := (\ell^{\uparrow}_{s,t} \wedge w)$, $\upsilon^{\downarrow*}_{s,t} := (\upsilon^{\downarrow}_{s,t} \vee w)$ and we now

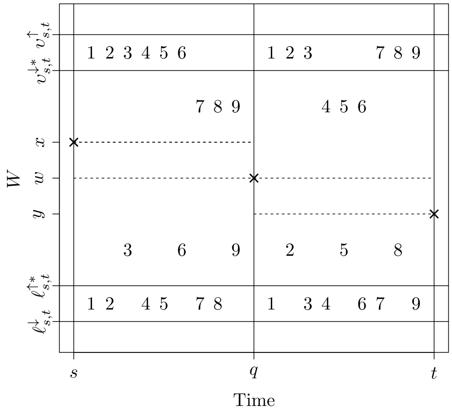

$\upsilon^{\uparrow}_{s,t}$
$\upsilon^{\downarrow *}_{s,t}$
1 2 3 4 5 6
1 2 3 7 8 9
7 8 9
4 5 6
x
W w
y
3 6 9
2 5 8
$\ell^{\uparrow *}_{s,t}$
$\ell^{\downarrow}_{s,t}$
1 2 4 5 7 8
1 3 4 6 7 9
s
q
t
Time

Figure 12. Illustration of 9 possible (disjoint) bisections.

replace D with,

$$D_* = \big\{W_{[s,t]}\colon \hat{m}_{s,t} \in \big[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\big]\big\} \cap \big\{W_{[s,t]}\colon \check{m}_{s,t} \in \big[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\big]\big\}. \tag{104}$$

The attainment of a particular layer in the interval $[s,t]$ by either the minimum or the maximum implies that the same layer is attained by the sample path in at least one of the sub-intervals $[s,q]$ or $[q,t]$. As such, in our case there are 9 possible (disjoint) bisections (which we denote as B_1–B_9 where $B := D_* = \biguplus_{i=1}^{9} B_i$) as illustrated in Figure 12. For instance, our sample path may lie in B_6, which more formally has the form,

$$\begin{aligned} B_6 := {}& \big(\big\{W_{[s,t]}\colon \hat{m}_{s,q} \in \big[\ell^{\uparrow *}_{s,t}, (x \wedge w)\big)\big\} \cap \big\{W_{[s,t]}\colon \check{m}_{s,q} \in \big[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\big]\big\}\big) \\ & \cap \big(\big\{W_{[s,t]}\colon \hat{m}_{q,t} \in \big[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\big]\big\} \cap \big\{W_{[s,t]}\colon \check{m}_{q,t} \in \big[(w \vee y), \upsilon^{\downarrow *}_{s,t}\big]\big\}\big). \end{aligned} \tag{105}$$

This notion can be extended to the case where we have multiple intermediate points ($\mathcal{W} := \{W_{q_1} = w_1, \ldots, W_{q_n} = w_n\}$), and want to dissect the interval into separate intersection layers. In particular, we are dissecting a single intersection layer into $(n+1)$ intersection layers, each with a layer for the minimum and maximum in their own sub-interval. As the sample path minimum and maximum must exist in one of the intersection layers there are $b := (2^{(n+1)} - 1)^2$ possible dissections $B^n_1, \ldots, B^n_b$. We can simulate which of these dissections our sample path lies in by application of the following results and Algorithm 20.

Theorem 9 (Intersection layer dissection). *The probability a Brownian bridge sample path is in B^n_i conditional on B and $\mathcal{W}$ is as follows (denoting by $\mathcal{L}(i)$ and $\mathcal{U}(i)$ the lower and upper layer sets for B^n_i),*

$$\begin{aligned} p_{B^n_i} &:= \mathbb{P}\big(B^n_i \,|\, \hat{m}_{s,t} \in \big[\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}\big], \check{m}_{s,t} \in \big[\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\big], W_s = x, W_t = y, \mathcal{W}\big) \\ &= \frac{{}^{(n)}\beta^{\mathcal{L}(i),\mathcal{U}(i)}_{s,t,x,y}(q_{1:n}, \mathcal{W})}{{}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n}, \mathcal{W})}. \end{aligned} \tag{106}$$

Algorithm 20 Dissecting an Intersection Layer

```
1. Simulate $u \sim \mathrm{U}[0,1]$ and set $j = 1$ and $k = 0$.
2. While $u \in (C^{B(n,j)}_{2k+1}, C^{B(n,j)}_{2k})$, $k = k + 1$.
3. If $u \leq C^{B(n,j)}_{2k+1}$ set dissection layer $B = B_j$ else set $j = j + 1$ and return to Step 2.
```

Proof. Follows directly by Bayes rule, Theorems 6 and 7. □

Remark 2 (Intersection layer bisection). In the particular case where we have a single intermediate point then the probability a Brownian bridge sample path is in B_i (conditional on B and $W_q = w$) reduces to that in [8] (denoting $\ell^{\downarrow,i}_{s,q}$, $\ell^{\uparrow,i}_{s,q}$ $\upsilon^{\downarrow,i}_{s,q}$, $\upsilon^{\uparrow,i}_{s,q}$ and $\ell^{\downarrow,i}_{q,t}$, $\ell^{\uparrow,i}_{q,t}$ $\upsilon^{\downarrow,i}_{q,t}$, $\upsilon^{\uparrow,i}_{q,t}$ as the bounds for B_i in the interval $[s,q]$ and $[q,t]$, resp.),

$$p_{B_i} = \frac{\beta(s,q,x,w,\ell^{\downarrow,i}_{s,q},\ell^{\uparrow,i}_{s,q},\upsilon^{\downarrow,i}_{s,q},\upsilon^{\uparrow,i}_{s,q}) \cdot \beta(q,t,w,y,\ell^{\downarrow,i}_{q,t},\ell^{\uparrow,i}_{q,t},\upsilon^{\downarrow,i}_{q,t},\upsilon^{\uparrow,i}_{q,t})}{\rho(s,q,t,x,w,y,\ell^{\downarrow}_{s,t},\ell^{\uparrow *}_{s,t},\upsilon^{\downarrow *}_{s,t},\upsilon^{\uparrow}_{s,t})}. \tag{107}$$

Corollary 9. *Events of probability p_{B_i} can be simulated by retrospective Bernoulli sampling* (*as per Corollaries* 1, 2 *and Algorithm* 11), *noting that it is a function of* ${}^{(n)}\beta^{\mathcal{L}(i),\mathcal{U}(i)}_{s,t,x,y}(q_{1:n},\mathcal{W})$ *and* ${}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W})$ *probabilities*, *using the following sequence*,

$$S^{B(n,i)}_k := \frac{S^{\beta(n)}_k(s,t,x,y,q_{1:n},\mathcal{W},\mathcal{L}(i),\mathcal{U}(i))}{S^{\rho(n)}_{k+1}(s,t,x,y,q_{1:n},\mathcal{W},\ell^{\downarrow}_{s,t},\ell^{\uparrow}_{s,t},\upsilon^{\downarrow}_{s,t},\upsilon^{\uparrow}_{s,t})}. \tag{108}$$

Unbiased simulation of the dissection the sample path lies in can be conducted by inversion sampling and an alternating Cauchy sequence representation of the CDF of B (109). In particular, by application of Corollary 2, our sample path lies in B^n_j if for some $k > 0$ and $u \sim \mathrm{U}[0,1]$ we have $u \in (C^{B(n,j-1)}_{2k+1}, C^{B(n,j)}_{2k})$.

$$C^{B(n,j)}_k := \sum_{i=1}^{j} S^{B(n,i)}_k. \tag{109}$$

8.4. Refining an Intersection Layer

Suppose we have already simulated layers for the maximum and minimum of our proposal Brownian bridge sample path ($\hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}]$ and $\check{m}_{s,t} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]$), but we require more *refined* layer information (i.e., we want a set of narrower layers $|\ell^{\uparrow *}_{s,t} - \ell^{\downarrow *}_{s,t}| \leq |\ell^{\uparrow}_{s,t} - \ell^{\downarrow}_{s,t}|$ or $|\upsilon^{\uparrow *}_{s,t} - \upsilon^{\downarrow *}_{s,t}| \leq |\upsilon^{\uparrow}_{s,t} - \upsilon^{\downarrow}_{s,t}|$). This can be achieved by noting that given some $\ell^{\updownarrow}_{s,t} \in (\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t})$ and $\upsilon^{\updownarrow}_{s,t} \in (\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t})$, the sample path falls in one of the following 4 possible (disjoint) intersection layer refinements (where $R := \biguplus_{i=1}^{4} R_i$),

$$R_1 = \{W_{[s,t]}\colon \hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, \ell^{\updownarrow}_{s,t}]\} \cap \{W_{[s,t]}\colon \check{m}_{s,t} \in [\upsilon^{\updownarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]\}, \tag{110}$$

$$R_2 = \{W_{[s,t]}\colon \hat{m}_{s,t} \in [\ell^{\updownarrow}_{s,t}, \ell^{\uparrow}_{s,t}]\} \cap \{W_{[s,t]}\colon \check{m}_{s,t} \in [\upsilon^{\updownarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}]\}, \tag{111}$$

Algorithm 21 Refining an Intersection Layer [8]

```
1. Simulate $u \sim \mathrm{U}[0,1]$ and set $j = 1$ and $k = 0$.
2. While $u \in (C^{R(j)}_{2k+1}, C^{R(j)}_{2k})$, $k = k + 1$.
3. If $u \le S^{R(j)}_{2k+1}$ set layer $R = R_j$ else set $j = j + 1$ and return to Step 2.
```

$$R_3 = \{W_{[s,t]}\colon \hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, \ell^{\updownarrow}_{s,t}]\} \cap \{W_{[s,t]}\colon \check{m}_{s,t} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\updownarrow}_{s,t}]\}, \tag{112}$$

$$R_4 = \{W_{[s,t]}\colon \hat{m}_{s,t} \in [\ell^{\updownarrow}_{s,t}, \ell^{\uparrow}_{s,t}]\} \cap \{W_{[s,t]}\colon \check{m}_{s,t} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\updownarrow}_{s,t}]\}, \tag{113}$$

and so we can simply unbiasedly simulate which one our sample path lies in.

In a similar fashion to Section 8.3, we can simulate unbiasedly which of the intersection layer refinements our sample path lies in by application of the following established results and Algorithm 21 (where we denote by $\ell^{\downarrow,i}_{s,t}$, $\ell^{\uparrow,i}_{s,t}$, $\upsilon^{\downarrow,i}_{s,t}$, $\upsilon^{\uparrow,i}_{s,t}$ with a superscript $i \in \{1,2,3,4\}$ the corresponding parameter selections from (110)–(113)).

Theorem 10 (Intersection layer refinement [8], Section 5.3). *The probability a Brownian bridge sample path contained within R is in R_i is as follows*,

$$\begin{aligned} p_{R_i} &:= \mathbb{P}\big(R_i \,|\, \hat{m}_{s,t} \in [\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}], \check{m}_{s,t} \in [\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}], W_s = x, W_t = y\big) \\ &= \frac{\beta(s,t,x,y,\ell^{\downarrow,i}_{s,t}, \ell^{\uparrow,i}_{s,t}, \upsilon^{\downarrow,i}_{s,t}, \upsilon^{\uparrow,i}_{s,t})}{\beta(s,t,x,y,\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t})}. \end{aligned} \tag{114}$$

Corollary 10 ([8], Section 5.3). *Events of probability p_{R_i} can be simulated by retrospective Bernoulli sampling* (*as per Corollaries* 1, 2 *and Algorithm* 11), *noting that it is a function of β probabilities*, *using the sequence*,

$$S^{R(i)}_k := \frac{S^{\beta}_k(s,t,x,y,\ell^{\downarrow,i}_{s,t}, \ell^{\uparrow,i}_{s,t}, \upsilon^{\downarrow,i}_{s,t}, \upsilon^{\uparrow,i}_{s,t})}{S^{\beta}_{k+1}(s,t,x,y,\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}, \upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t})}. \tag{115}$$

By application of Corollary 2, unbiased simulation of the refinement the sample path lies in can be conducted by inversion sampling and an alternating Cauchy sequence representation of the CDF of R (116). In particular, our sample path lies in R_j if for some $k > 0$ and $u \sim \mathrm{U}[0,1]$ we have $u \in (C^{R(j-1)}_{2k+1}, C^{R(j)}_{2k})$, where:

$$C^{R(j)}_k := \sum_{i=1}^{j} S^{R(i)}_k. \tag{116}$$

8.5. Simulating Layered Brownian Bridges

The *Intersection Layer Approach* for constructing a layered Brownian bridge is a direct application of the algorithms of Sections 8.1, 8.2 and 8.3. In particular, we simulate initial intersection

Algorithm 22 Layered Brownian Bridge Simulation (Intersection Layer Approach)

```
1. For each intermediate point required ($q$),
   (a) Select the appropriate existing intersection layer $\mathcal{S}_{a,b}$ from $\mathcal{S}$ such that $q \in (a, b)$.
   (b) Simulate $X_q$ as per Algorithm 17 or 18 or 19.
   (c) Dissect interval as per Algorithm 20 to find new intersection layers $\mathcal{S}_{a,q}$ and $\mathcal{S}_{q,b}$.
   (d) Set $\mathcal{S} = \mathcal{S} \cup \{\mathcal{S}_{a,q}, \mathcal{S}_{q,b}\} \setminus \mathcal{S}_{a,b}$.
```

layer information for the sample path (Algorithm 4 Step 2) by application of Algorithm 16. In Algorithm 4 Step 4, we iteratively simulate skeletal (intermediate) points, then new intersection layer information conditional on these points. This can be achieved directly by either Algorithm 17, 18, 19 or some mixture of these algorithms to simulate the intermediate point (as discussed in Section 8.2 and in particular Section 8.2.4) and Algorithm 20 to bisect the interval.

We present the iterative Algorithm 4 Step 4 in Algorithm 22 which can be additionally used to conduct Algorithm 4 Step 6. $\mathcal{S}$ denotes the set containing all intersection layer information. The set is composed of $(n - 1)$ elements corresponding to the intervals between n existing time points. In particular, each element $(\mathcal{S}_{a,b})$ between two successive time points $(a < b)$ contains information regarding the sample path at the end points and an upper and lower bound for both the minimum and maximum of the sample path in that interval $(\mathcal{S}_{a,b} := \{a, b, X_a, X_b, \ell^{\downarrow}_{a,b}, \ell^{\uparrow}_{a,b}, \upsilon^{\downarrow}_{a,b}, \upsilon^{\uparrow}_{a,b}\})$.

It should be noted that further refinements to Algorithm 22 could be made when considering any particular application, however, we have omitted the explicit algorithms here. For instance, if the simulation of intermediate points is required for the AUEA (Algorithm 4), then refining the intersection layers as outlined in Section 8.4 and detailed in Algorithm 21 would result in tighter upper and lower bounds for the sample path. As a consequence tighter upper and lower bounds for $\phi(X)$ could be computed, resulting in a more efficient algorithm. Similar notions to this are explored in Sections 5, 5.1 and 9.

9. Examples – Unbiased Estimation of Irregular Barrier Crossing Probabilities

In this section, we present a number of applications of the methodology developed in this paper. In particular, we show that it is possible to determine exactly whether a jump diffusion sample path simulated as per Algorithm 7 crosses various types of barriers. In Section 9.1 we consider a nonlinear two sided barrier, in Section 9.2 we consider the crossing of two jump diffusion sample paths from different laws and finally in Section 9.3 we consider the crossing of a circular barrier by a 2-dimensional jump diffusion sample path. The flexibility of the methodology developed in this paper allows us to by extension simulate various non-trivial quantities, for instance, we can construct an unbiased estimate of the probability that any barrier is crossed, an unbiased estimate of the probability a barrier is crossed by any particular time and the killed (or un-killed) diffusion transition density among many other interesting possibilities.

In each of our examples we employ variants of Algorithm 23. For simplicity in Algorithm 23 we only consider the crossing of an upper barrier by a one dimensional jump diffusion, however,

Algorithm 23 Unbiased Estimation of Upper Barrier Crossing

```
1. Simulate jump diffusion skeleton $\mathcal{S}_{\text{AUJEA}}(X) := \{(\xi_i, X_{\xi_i})_{i=0}^{\sum_j(\kappa_j+1)}, (R_X^{[\xi_{i-1},\xi_i]})_{i=1}^{\sum_j(\kappa_j+1)}\}$
   as per Algorithm 7.
2. Set $\mathcal{S} := \mathcal{S}_{\text{AUJEA}}(X)$, $\mathcal{C} := \emptyset$, $\mathcal{B} := \emptyset$ and $\mathcal{U} := \{[s^1, t^1], \dots, [s^{|\mathcal{S}|}, t^{|\mathcal{S}|}]\}$.
3. While $|\mathcal{C}| = 0$ and $|\mathcal{B}| < |\mathcal{S}|$,
   (a) For $i$ in 1 to $|\mathcal{U}|$,
       i. If $X_s^i \geq B_s^i$ or $X_t^i \geq B_t^i$ or $\upsilon_{s,t}^{i,\downarrow} \geq \sup_{u\in[s(i),t(i)]} B_u$ then $\mathcal{C} := \mathcal{C} \cup \{[s^i, t^i]\}$ and
          $\mathcal{U} := \mathcal{U} \setminus \{[s^i, t^i]\}$.
       ii. If $\upsilon_{s,t}^{i,\uparrow} \leq \inf_{u\in[s(i),t(i)]} B_u$ then $\mathcal{B} := \mathcal{B} \cup \{[s^i, t^i]\}$ and $\mathcal{U} := \mathcal{U} \setminus \{[s^i, t^i]\}$.
       iii. If $[s^i, t^i] \in \mathcal{U}$ then,
            A. Simulate $X_q$ where $q := (s^i + t^i)/2$ conditional on $\mathcal{S}_{s,t}^i$ as per Algorithm 17.
            B. Bisect and refine $\mathcal{S}_{s,t}^i$ into $\mathcal{S}_{s,q}^{i,1}$ and $\mathcal{S}_{q,t}^{i,2}$ as per Algorithm 20 and Algorithm 21.
            C. Set $\mathcal{S} := \mathcal{S} \cup \mathcal{S}_{s,q}^{i,1} \cup \mathcal{S}_{q,t}^{i,2} \setminus \mathcal{S}_{s,t}^i$ and $\mathcal{U} := \mathcal{U} \cup \{[s^i, q], [q, t^i]\} \setminus \{[s^i, t^i]\}$.
4. If $|\mathcal{C}| > 0$ then barrier crossed, else if $|\mathcal{B}| = |\mathcal{S}|$ barrier not crossed.
```

as we will discuss later in this section and in Sections 9.1–9.3 this can be straight-forwardly extended. To simplify notation we define B_u as the evaluation of the upper barrier at time point u. As before we denote $\mathcal{S}$ as the skeleton comprising intersection layer information, and further denote $\mathcal{C}$ as the set of intervals in which the sample path crosses the barrier, $\mathcal{B}$ as the set of intervals in which there is no crossing, and $\mathcal{U}$ as the set in which for each interval crossing is undetermined.

9.1. Example 1 – Nonlinear two sided barrier

In this section, we consider the simulation of jump diffusion sample paths which can be represented as solutions to the following SDE,

$$\mathrm{d}X_t = \sin(X_{t-})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \tag{117}$$

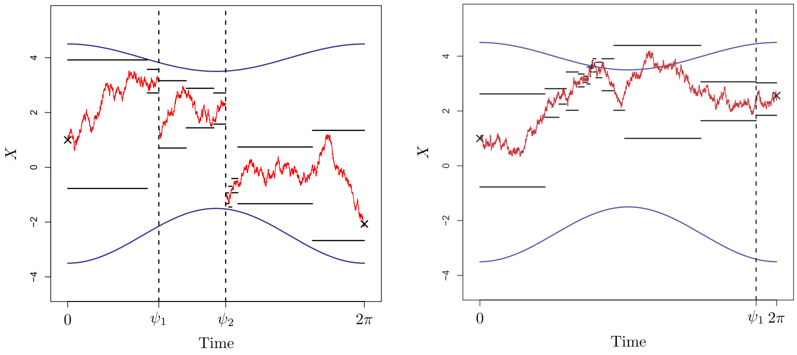
where

$$X_0 = 1, \qquad t \in [0, 2\pi], \qquad \lambda(X_{t-}) = |X_{t-}/4|, \qquad f_\nu(\cdot\,; X_{t-}) = \mathrm{N}(\cdot\,; -X_{t-}/2, 2), \tag{118}$$

determining whether or not they cross the following nonlinear two sided barrier (where $B_u^\downarrow$ and $B_u^\uparrow$ denote the lower and upper barriers at time point u, resp.),

$$B_u^\downarrow = -2.5 - \cos(u), \qquad B_u^\uparrow = 4 + 0.5\cos(u), \qquad u \in [0, 2\pi]. \tag{119}$$

In this case, as the jump intensity of (118) can't be bounded we simulate sample paths using the AUJEA (see Algorithm 7). In particular, we have $\phi(X_t) := (\sin^2(X_t) + \cos(X_t))/2 \in [-1/2, 5/8]$, $\lambda(X_t)|(L_X, U_X) \leq \max\{|L_X|, |U_X|\}/4$ and the end point is simulated according as follows, $X_T \sim h(y; x, T) \propto \exp\{-\cos(y) - y^2/6\}$.



(a) No barrier crossing (b) Barrier crossed

Figure 13. Illustration of the determination of whether a 2-sided non-linear barrier has been crossed by a sample path using a finite dimensional sample path skeleton, overlaid with an illustration of the underlying sample path.

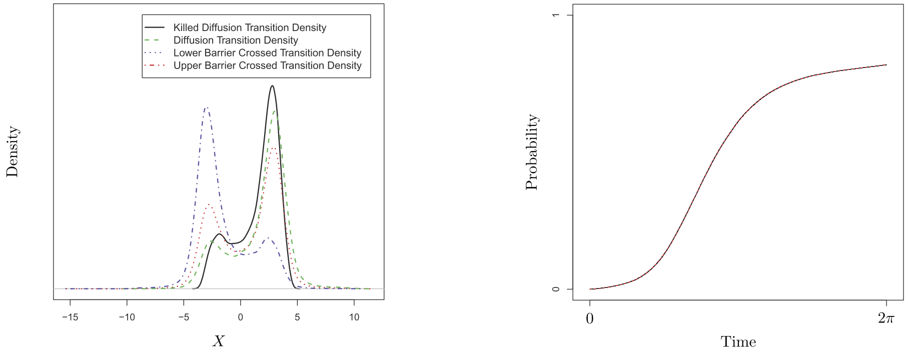
In Figure 13, we present illustrations of whether the two sided barrier (119) has been crossed using finite dimensional realisations of sample paths simulated according to the measure induced by (117) and by applying a modified version of Algorithm 23. This example is motivated by a number of possible applications in finance, such as the pay-off of barrier options.

In this example, we simulated 100 000 sample paths from the measure induced by (118) and determined whether the barrier (119) was crossed for each sample path. For each sample path we additionally determined whether one or both barriers were crossed and if both, which barrier was crossed first. From these simulations, we calculated unbiased estimates of various barrier crossing probabilities, the results of which are summarised in Table 1.

In Figure 14(a), we present kernel density estimates of the transition densities of various subsets of the sample paths simulated, including that for killed diffusions (i.e., sample paths from the measure induced by (118) with the restriction that they remain within the interval between the barriers in (119)). In Figure 14(b), we additionally determine for each sample path an interval of length $\varepsilon \leq 10^{-4}$, in which the first crossing time occurs (by modifying the ε-strong algorithms

Table 1. Nonlinear two sided barrier example: Barrier crossing probabilities (computed using 100 000 sample paths)

Crossing type	Empirical probability	95% Clopper–Pearson confidence interval
Neither barrier	18.09%	[17.85%, 18.36%]
Either barrier	81.91%	[81.67%, 82.15%]
Upper barrier only	43.98%	[43.67%, 44.29%]
Lower barrier only	29.02%	[28.74%, 29.30%]
Both barriers	8.92%	[8.74%, 9.09%]
Upper first given both barriers	80.45%	[79.61%, 81.27%]
Lower first given both barriers	19.55%	[18.73%, 20.39%]



(a) Kernel density estimates of the transition densities of subsets of sample paths simulated from the measure induced by (118)

(b) Empirical CDF of barrier crossing probability by time (crossing time evaluated within interval of length $\varepsilon \leq 10^{-4}$)

Figure 14. Nonlinear two-sided barrier example: summary figures computed using 100 000 sample paths.

presented in Section 5) to construct upper and lower bounds for the empirical CDF of the first barrier crossing time.

9.2. Example 2 – Jump diffusion barrier

In this section, we consider the simulation of jump diffusion sample paths which can be represented as solutions to the following SDE,

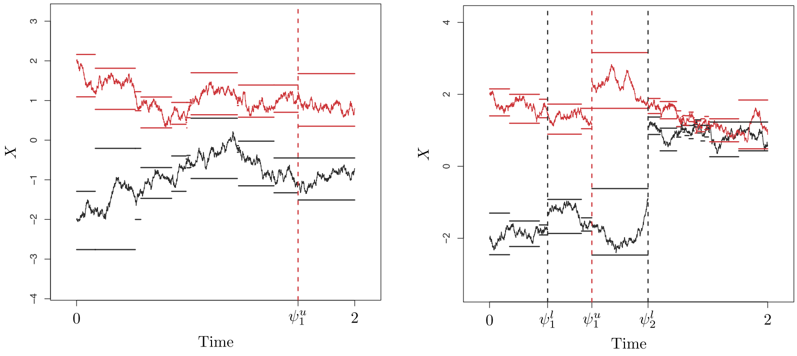
$$\mathrm{d}X_t = -X_{t-}\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \tag{120}$$

where

$$t \in [0,2], \qquad \lambda(X_{t-}) = \sin^2(X_{t-}), \qquad f_\nu(\cdot; X_{t-}) = \mathrm{N}(\cdot; -X_{t-}/2, 1). \tag{121}$$

We consider sample paths simulated from the measure induced by (120) initialised at two possible starting values $X_0^\ell = -2$ and $X_0^\upsilon = 2$ (where X^ℓ and X^υ denote the lower and upper diffusions, resp.). In this case, the jump intensity of (120) is bounded so we simulate sample paths using the AUEA (see Algorithm 4) within the BJEA (see Algorithm 5). Recall that in the BJEA the interval the sample path is to be simulated over ($t \in [0,2]$), is broken into segments corresponding to the proposed jump times ($\Psi_1, \ldots$). As such, if we consider the simulation of a diffusion sample path in the interval $[\Psi_1, \Psi_2]$ conditional on X_{Ψ_1} then the proposed end point is simulated as follows, $X_{\Psi_2} \sim h(X_{\Psi_2}; X_{\Psi_1}, \Psi_2 - \Psi_1) \propto \exp\{-X_{\Psi_2}^2/2 - (X_{\Psi_2} - X_{\Psi_1})^2/[2(\Psi_2 - \Psi_1)]\}$. Furthermore, we have $\phi(X_t) := (X_t^2 - 1)/2$, $\phi(X_t)|(L_X, U_X) \in [-1/2, (\max\{L_X^2, U_X^2\} - 1)/2]$ and $\lambda(X_t) \leq 1 =: \Lambda$.

In Figure 15, we present illustrations of two sample paths simulated from the measure induced by (120), initialised at $X_0^\ell = -2$ and $X_0^\upsilon = 2$, which do not cross and cross, respectively, determined using only a finite dimensional realisation of the sample paths. This example is motivated



(a) No crossing (b) Crossing

Figure 15. Illustration of the determination of whether two diffusion sample paths cross one another using finite dimensional sample path skeletons, overlaid with an illustration of the underlying sample paths.

by [11], in which (in part) the authors are interested in the probability that two Brownian motion sample paths cross one another.

In this example, we simulated 100 000 pairs of sample paths from the measure induced by (120) initialised at $X_0^\ell = -2$ and $X_0^\upsilon = 2$ and determined whether or not they crossed. We present a summary of the unbiased estimates calculated from these sample paths in Table 2. In Figure 16(a) we present kernel density estimates of the transition densities of various subsets of the sample paths simulated. In Figure 16(b), we determine for each sample path an interval of length $\varepsilon \leq 10^{-4}$ in which the first crossing time occurs in order to construct upper and lower bounds for the empirical CDF of the first crossing time.

9.3. Example 3 – 2-D jump diffusion with circular barrier

In this section, we consider the simulation of jump diffusion sample paths which can be represented as solutions to the following SDE,

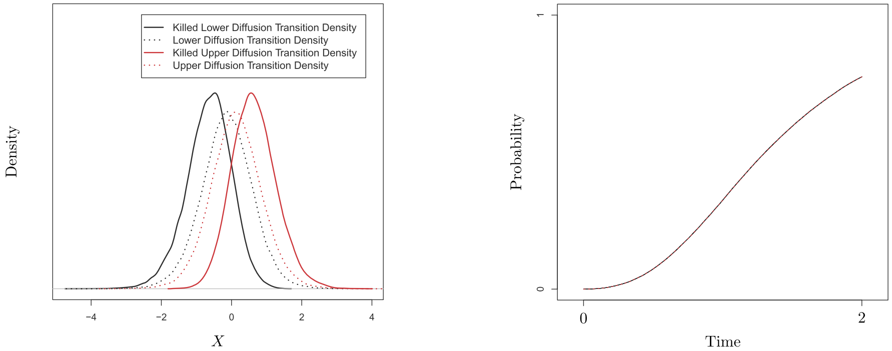
$$X := \left(X^{(1)}, X^{(2)}\right), \tag{122}$$

where

$$\mathrm{d}X_t^{(1)} = -X_{t-}^{(1)}\,\mathrm{d}t + \mathrm{d}W_t^{(1)} + \mathrm{d}J_t^{\lambda(X),\nu_1(X(1))}, \tag{123}$$

Table 2. Jump diffusion crossing example: crossing probabilities (computed using 100 000 sample paths)

Crossing type	Empirical probability	95% Clopper–Pearson confidence interval
No crossing	22.52%	[22.26%, 22.78%]
Crossing	77.48%	[77.22%, 77.74%]



(a) Kernel density estimates of the transition densities of subsets of sample paths simulated from the measure induced by (120)

(b) Empirical CDF of jump diffusion crossing probability (crossing time evaluated within interval of length $\varepsilon \leq 10^{-4}$)

Figure 16. Jump diffusion crossing example: summary figures computed using 100 000 sample paths.

$$\mathrm{d}X_t^{(2)} = -X_{t-}^{(2)}\,\mathrm{d}t + \mathrm{d}W_t^{(2)} + \mathrm{d}J_t^{\lambda(X),\nu_2(X(2))}, \tag{124}$$

and further denoting by $\theta := \arctan(X_t^{(2)}/X_t^{(1)})$ and $Z \sim \mathrm{U}[0, [(X_t^{(1)})^2 + (X_t^{(2)})^2]^{1/2}]$,

$$X_0 = (0, 0.5), \qquad t \in [0, 3], \qquad \lambda(X_t) = \bigl[\bigl(X_t^{(1)}\bigr)^2 + \bigl(X_t^{(2)}\bigr)^2\bigr]^{1/2}, \tag{125}$$

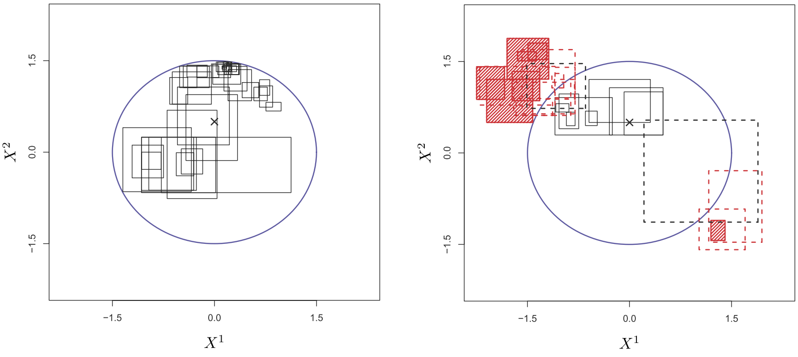
$$f_{\nu_1}\bigl(\cdot; X_{t-}^{(1)}\bigr) \sim -\cos(\theta)Z, \qquad f_{\nu_2}\bigl(\cdot; X_{t-}^{(2)}\bigr) \sim -\sin(\theta)Z, \tag{126}$$

determining whether or not they cross the following circular barriers,

$$x^2 + y^2 = r, \qquad \text{where } r = \{0.8, 1, \ldots, 2.8, 3\}. \tag{127}$$

The jump intensity of this SDE (122, 125, 126) can't be bounded so we simulate sample paths using the AUJEA (see Algorithm 7). In Figure 17, we present illustrations of whether one particular circular barrier ($r = 1.6$) has been crossed using finite dimensional realisations of sample paths simulated according to the measure induced by (122, 125, 126) and by applying a modified version of Algorithm 23.

In this example we simulated 50 000 sample paths from the measure induced by (122), (125), (126), determining for each circular barrier (127) whether or not it was crossed. In addition, for each circular barrier we simulated the time within an interval of length $\varepsilon \leq 10^{-3}$ in which the barrier was first crossed and an interval of length $\theta \leq 10^{-3}$ in which the exit angle lies. Calculating all circular barriers for every sample path ensures that the calculated probabilities retain any natural ordering (e.g., the first crossing time of a circular barrier of a given radius must occur before one of larger radius). In Figure 18, we present various results obtained from our simulations which may be of interest in any given application.



(a) No barrier crossing (b) Barrier crossed

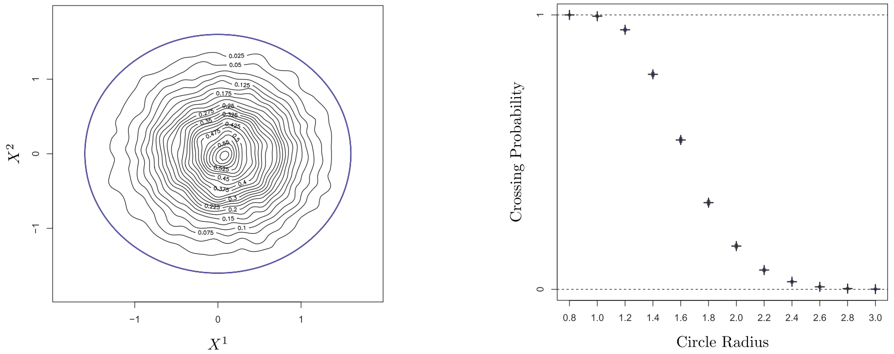
Figure 17. Illustration of the determination of whether a 2-D sample path crosses a circular barrier using a finite dimensional sample path skeleton. Inscribed rectangles denote regions where for some time interval sample paths are constrained. Black and infilled red rectangles denote intervals constrained entirely within or out-with the circle, respectively. Dotted black and red rectangles denote intervals with undetermined or partial crossing, respectively.

Appendix: Elementary Cauchy Sequence Functions

In Section 6.2, we define the functions $\bar{\varsigma}$ and $\bar{\varphi}$ which form the building blocks for the construction of all other alternating Cauchy sequences in this paper. In Section 8.2, we exploit the full representation of ρ found in Theorem 6 and Definition 5 in terms of $\bar{\varsigma}$ and $\bar{\varphi}$. In particular, we make the remark that each can be represented in the form $\exp(a_i + b_i w)$ where each function a_i and b_i can be explicitly calculated. Furthermore, note that the multiple of any two such functions can also be represented in the form $\exp(a_j + b_j w)$.

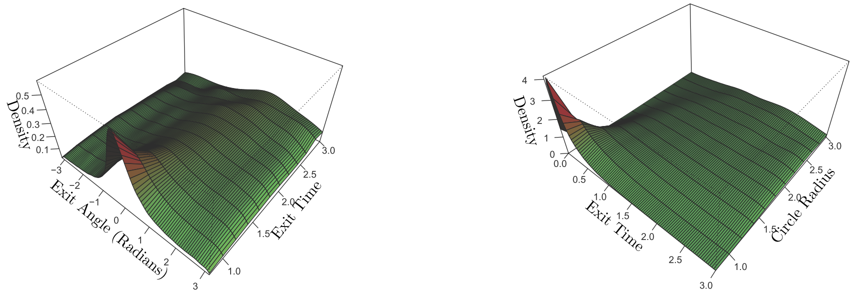
In this Appendix, we briefly detail the possible functions that can arise and show an explicit representation for each in terms of a_i and b_i. With reference to details in Section 6.2 and Section 6.3, we can limit our consideration to the following four functional forms (noting that there are also corresponding negative versions (in which the sign of x, w and y reverses), a set for each of the possible layer combinations (in which we substitute $[\ell_i, \upsilon_i]$ for $[\ell\downarrow, \upsilon\uparrow]$, $[\ell\uparrow, \upsilon\uparrow]$, $[\ell\downarrow, \upsilon\downarrow]$ or $[\ell\uparrow, \upsilon\downarrow]$) as well as various multiples of these functions (which as a consequence of their exponential form will simply result in the addition of the a and b terms)). Denoting $D := |\upsilon_i - \ell_i|$ and $M := (\ell_i \wedge \upsilon_i)$ we have,

$$\bar{\varsigma}_{s,q}^{\ell_i,\upsilon_i}(j; x, w) = \exp\Bigg\{\underbrace{-\frac{2}{q-s}\big(D^2 j^2 + 2DMj + M^2 - Djx - Mx\big)}_{a_{i,1}} \underbrace{-\frac{2}{q-s}(-Dj - M + x)\,w}_{b_{i,1}}\Bigg\}, \tag{128}$$



(a) Contour plot of kernel density estimate of killed diffusion transition density with circular barrier of radius 1.6

(b) Empirical probabilities of crossing centred circles of increasing radius overlaid with 95% Clopper–Pearson confidence intervals



(c) Circle of radius 1.6 exit angle by exit time (crossing time and exit angle evaluated within intervals of length $\varepsilon \leq 10^{-3}$ and $\theta \leq 10^{-3}$, resp.)

(d) Circle exit time by circle radius (crossing time evaluated within interval of length $\varepsilon \leq 10^{-3}$)

Figure 18. 2-D jump diffusion with circular barrier example: figures computed using 50 000 sample paths.

$$\bar{\varsigma}^{\ell_i,\upsilon_i}_{q,t}(j;w,y) = \exp\Big\{\underbrace{-\frac{2}{t-q}\big(D^2j^2+2DMj+M^2-Djy-My\big)}_{a_{i,2}} \underbrace{-\frac{2}{t-q}(-Dj-M+y)\,w}_{b_{i,2}}\Big\}, \tag{129}$$

$$\bar{\varphi}^{\ell_i,\upsilon_i}_{s,q}(j;x,w) = \exp\Big\{\underbrace{-\frac{2j}{q-s}\big(D^2j+Dx\big)}_{a_{i,3}} + \underbrace{\frac{2j}{q-s}D\,w}_{b_{i,3}}\Big\}, \tag{130}$$

$$\bar{\varphi}^{\ell_i,\upsilon_i}_{q,t}(j;w,y) = \exp\Big\{\underbrace{-\frac{2j}{t-q}\big(D^2j-Dy\big)}_{a_{i,4}} \underbrace{-\frac{2j}{t-q}D\,w}_{b_{i,4}}\Big\}. \tag{131}$$

Acknowledgements

M. Pollock would like to thank Dr. Alexandros Beskos and Prof. Omiros Papaspiliopoulos for stimulating discussion on aspects of this paper. The authors would like to thank the anonymous referees for their helpful comments. This work was supported by the Engineering and Physical Sciences Research Council. M. Pollock was supported by EPSRC Grant EP/P50516X/1; A.M. Johansen was partially supported by EPSRC Grant EP/1017984/1; G.O. Roberts was partially supported by EPSRC Grants EP/G026521/1, EP/D002060/1.

References

[1] Aït-Sahalia, Y. (2008). Closed-form likelihood expansions for multivariate diffusions. *Ann. Statist.* **36** 906–937. MR2396819

[2] Anderson, T.W. (1960). A modification of the sequential probability ratio test to reduce the sample size. *Ann. Math. Statist.* **31** 165–197. MR0116441

[3] Asmussen, S., Glynn, P. and Pitman, J. (1995). Discretization error in simulation of one-dimensional reflecting Brownian motion. *Ann. Appl. Probab.* **5** 875–896. MR1384357

[4] Barndorff-Nielsen, O.E. and Shephard, N. (2004). Power and bi-power variation with stochastic volatility and jumps. *J. Financ. Econom.* **2** 1–37.

[5] Beskos, A., Papaspiliopoulos, O. and Roberts, G.O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli* **12** 1077–1098. MR2274855

[6] Beskos, A., Papaspiliopoulos, O. and Roberts, G.O. (2008). A factorisation of diffusion measure and finite sample path constructions. *Methodol. Comput. Appl. Probab.* **10** 85–104. MR2394037

[7] Beskos, A., Papaspiliopoulos, O., Roberts, G.O. and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **68** 333–382. MR2278331

[8] Beskos, A., Peluchetti, S. and Roberts, G. (2012). ϵ-strong simulation of the Brownian path. *Bernoulli* **18** 1223–1248. MR2995793

[9] Beskos, A. and Roberts, G.O. (2005). Exact simulation of diffusions. *Ann. Appl. Probab.* **15** 2422–2444. MR2187299

[10] Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *J. Polit. Econ.* **81** 637–654.

[11] Bladt, M. and Sørensen, M. (2014). Simple simulation of diffusion bridges with application to likelihood inference for diffusions. *Bernoulli* **20** 645–675. MR3178513

[12] Burq, Z.A. and Jones, O.D. (2008). Simulation of Brownian motion at first-passage times. *Math. Comput. Simulation* **77** 64–71. MR2388251

[13] Casella, B. and Roberts, G.O. (2011). Exact simulation of jump-diffusion processes with Monte Carlo applications. *Methodol. Comput. Appl. Probab.* **13** 449–473. MR2822390

[14] Chen, N. and Huang, Z. (2013). Localization and exact simulation of Brownian motion-driven stochastic differential equations. *Math. Oper. Res.* **38** 591–616. MR3092549

[15] Devroye, L. (1986). *Non-Uniform Random Variate Generation*, 1st ed. New York: Springer.

[16] Eraker, B., Johannes, M. and Polson, N. (2003). The impact of jumps in volatility and returns. *J. Finance* **58** 1269–1300.

[17] Giesecke, K. and Smelov, D. (2013). Exact sampling of jump diffusions. *Oper. Res.* **61** 894–907. MR3105734

[18] Golightly, A. and Wilkinson, D.J. (2006). Bayesian sequential inference for nonlinear multivariate diffusions. *Stat. Comput.* **16** 323–338. MR2297534

[19] Golightly, A. and Wilkinson, D.J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Comput*. *Statist*. *Data Anal*. **52** 1674–1693. MR2422763

[20] Gonçalves, F.B. and Roberts, G.O. (2013). Exact simulation problems for jump-diffusions. *Methodol*. *Comput*. *Appl*. *Probab*. **15** 1–24.

[21] Jacod, J. and Protter, P. (2012). *Discretization of Processes*. *Stochastic Modelling and Applied Probability* **67**. Heidelberg: Springer. MR2859096

[22] Karatzas, I. and Shreve, S.E. (1991). *Brownian Motion and Stochastic Calculus*, 2nd ed. *Graduate Texts in Mathematics* **113**. New York: Springer. MR1121940

[23] Kingman, J.F.C. (1992). *Poisson Processes*, 1st ed. Oxford: Clarendon Press.

[24] Kloeden, P.E. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. *Applications of Mathematics* (*New York*) **23**. Berlin: Springer. MR1214374

[25] Merton, R.C. (1973). Theory of rational option pricing. *Bell J*. *Econ*. *Manag*. *Sci*. **4** 141–183. MR0496534

[26] Merton, R.C. (1976). Option pricing when underlying stock returns are discontinuous. *J*. *Financ*. *Econ*. **3** 125–144.

[27] Øksendal, B. (2007). *Stochastic Differential Equations*, 6th ed. Berlin: Springer.

[28] Øksendal, B. and Sulem, A. (2004). *Applied Stochastic Control of Jump Diffusions*, 2nd ed. Berlin: Springer.

[29] Picchini, U., De Gaetano, A. and Ditlevsen, S. (2010). Stochastic differential mixed-effects models. *Scand*. *J*. *Stat*. **37** 67–90. MR2675940

[30] Platen, E. and Bruti-Liberati, N. (2010). *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. *Stochastic Modelling and Applied Probability* **64**. Berlin: Springer. MR2723480

[31] Pollock, M. (2013). Some Monte Carlo methods for jump diffusions. Ph.D. thesis, Dept. Statistics, Univ. Warwick.

[32] Pötzelberger, K. and Wang, L. (2001). Boundary crossing probability for Brownian motion. *J*. *Appl*. *Probab*. **38** 152–164. MR1816120

[33] Ripley, B.D. (1987). *Stochastic Simulation*. New York: Wiley. MR0875224

[34] Robert, C.P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer. MR2080278

[35] Sermaidis, G., Papaspiliopoulos, O., Roberts, G.O., Beskos, A. and Fearnhead, P. (2013). Markov chain Monte Carlo for exact inference for diffusions. *Scand*. *J*. *Stat*. **40** 294–321. MR3066416

Received July 2013 and revised April 2014