THE UNIVERSITY OF
WARWICK

Library Declaration and Deposit Agreement

1. STUDENT DETAILS

Please complete the following:

Full name: Murray Pollock

University ID number: 0859881

2. THESIS DEPOSIT

2.1 I understand that under my registration at the University, I am required to deposit my thesis with the University in BOTH hard copy and in digital format. The digital version should normally be saved as a single pdf file.

2.2 The hard copy will be housed in the University Library. The digital version will be deposited in the University's Institutional Repository (WRAP). Unless otherwise indicated (see 2.3 below) this will be made openly accessible on the Internet and will be supplied to the British Library to be made available online via its Electronic Theses Online Service (EThOS) service.
[At present, theses submitted for a Master's degree by Research (MA, MSc, LLM, MS or MMedSci) are not being deposited in WRAP and not being made available via EthOS. This may change in future.]

2.3 In exceptional circumstances, the Chair of the Board of Graduate Studies may grant permission for an embargo to be placed on public access to the hard copy thesis for a limited period. It is also possible to apply separately for an embargo on the digital version. (Further information is available in the *Guide to Examinations for Higher Degrees by Research*.)

2.4 *If you are depositing a thesis for a Master's degree by Research, please complete section (a) below. For all other research degrees, please complete both sections (a) and (b) below:*

(a) Hard Copy

I hereby deposit a hard copy of my thesis in the University Library to be made publicly available to readers (please delete as appropriate) EITHER immediately OR after an embargo period of ……..... months/years as agreed by the Chair of the Board of Graduate Studies.

I agree that my thesis may be photocopied. (YES / NO) (*Please delete as appropriate*)

(b) Digital Copy

I hereby deposit a digital copy of my thesis to be held in WRAP and made available via EThOS.

Please choose one of the following options:

EITHER My thesis can be made publicly available online. (YES / NO) (*Please delete as appropriate*)

OR My thesis can be made publicly available only after…..[date] (Please give date)
YES / NO (*Please delete as appropriate*)

OR My full thesis cannot be made publicly available online but I am submitting a separately identified additional, abridged version that can be made available online.
YES / NO (*Please delete as appropriate*)

OR My thesis cannot be made publicly available online. YES / NO (*Please delete as appropriate*)

3. **GRANTING OF NON-EXCLUSIVE RIGHTS**

Whether I deposit my Work personally or through an assistant or other agent, I agree to the following:

Rights granted to the University of Warwick and the British Library and the user of the thesis through this agreement are non-exclusive. I retain all rights in the thesis in its present version or future versions. I agree that the institutional repository administrators and the British Library or their agents may, without changing content, digitise and migrate the thesis to any medium or format for the purpose of future preservation and accessibility.

4. **DECLARATIONS**

(a) I DECLARE THAT:

- I am the author and owner of the copyright in the thesis and/or I have the authority of the authors and owners of the copyright in the thesis to make this agreement. Reproduction of any part of this thesis for teaching or in academic or other forms of publication is subject to the normal limitations on the use of copyrighted materials and to the proper and full acknowledgement of its source.

- The digital version of the thesis I am supplying is the same version as the final, hard-bound copy submitted in completion of my degree, once any minor corrections have been completed.

- I have exercised reasonable care to ensure that the thesis is original, and does not to the best of my knowledge break any UK law or other Intellectual Property Right, or contain any confidential material.

- I understand that, through the medium of the Internet, files will be available to automated agents, and may be searched and copied by, for example, text mining and plagiarism detection software.

(b) IF I HAVE AGREED (in Section 2 above) TO MAKE MY THESIS PUBLICLY AVAILABLE DIGITALLY, I ALSO DECLARE THAT:

- I grant the University of Warwick and the British Library a licence to make available on the Internet the thesis in digitised format through the Institutional Repository and through the British Library via the EThOS service.

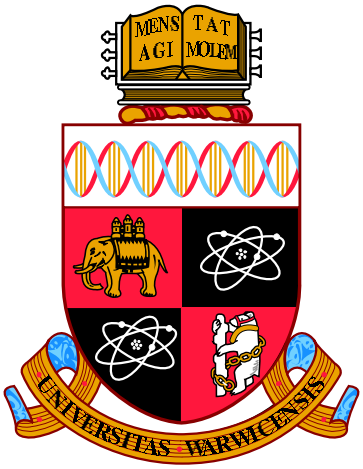
- If my thesis does include any substantial subsidiary material owned by third-party copyright holders, I have sought and obtained permission to include it in any version of my thesis available in digital format and that this permission encompasses the rights that I have granted to the University of Warwick and to the British Library.

5. **LEGAL INFRINGEMENTS**

I understand that neither the University of Warwick nor the British Library have any obligation to take legal action on behalf of myself, or other rights holders, in the event of infringement of intellectual property rights, breach of contract or of any other right, in the thesis.

Please sign this agreement and return it to the Graduate School Office when you submit your thesis.

Student's signature: ...  ..... Date: .....



Some Monte Carlo Methods for Jump Diffusions

by

Murray Pollock

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Statistics

September 2013

THE UNIVERSITY OF WARWICK

To my family

Contents

Declarations

I hereby declare that this thesis is the result of my own work and research, except where otherwise indicated. This thesis has not been submitted for examination to any institution other than the University of Warwick.

Signed: 

Murray Pollock

30th September 2013

Acknowledgments

"There are two types of sacrifices: correct ones, and mine."

— Mikhail Tal

First and foremost I would like to extend my deepest gratitude to Adam Johansen and Gareth Roberts. It has been a great privilege to have the opportunity to work with them over the past four years. Without their patience, encouragement and guidance it wouldn't have been possible to complete this thesis.

I would also like to thank Alexandros Beskos, Flávio Gonçalves, Omiros Papaspiliopoulos and Giorgos Sermaidis for a number of stimulating discussions during my PhD studies, which (together with their excellent PhD theses!) have resulted in me having a fuller understanding of the topic of my research and have ultimately contributed to the development of my thesis. More broadly I would like to thank the current and former members of the Statistics Department at Warwick for contributing to the inspiring academic environment – particularly the regulars of the Feynman-Kac and Algorithms reading groups.

Some special thanks are due to Gavin Gibson and Daniel Moreno for the career advice which led me to Warwick, and of course Jim Smith for his colourful thesis, academic, teaching (and somewhat questionable personal) advice throughout my time at Warwick. Last, but certainly not least, many thanks to Hugo for his support over the past few years, and my parents for their support over many years!

I would also like to acknowledge the Engineering and Physical Sciences Research Council (EPSRC grant number EP/P50516X/1), the Centre for Research in Statistical Methodology (EPSRC grant number EP/D002060/1) and the University of Warwick for financial support during my PhD studies.

Murray Pollock

30th September 2013

Many thanks to Petros Dellaportas and Andrew Stuart for not only taking the time to read and examine my thesis, but also for taking an interest in my research and asking so many pertinent questions.

Murray Pollock

6th December 2013

Abstract

In this thesis we develop computationally efficient methods to simulate finite dimensional representations of (jump) diffusion and (jump) diffusion bridge sample paths over finite intervals, without discretisation error (*exactly*), in such a way that the sample path can be restored at any desired finite collection of time points. Furthermore, we extend methodology for particle filters to the setting in which the transition density of the latent process is governed by a jump diffusion. Finally, we present methodology which allows the simulation of upper and lower bounding processes which almost surely constrain (jump) diffusion and (jump) diffusion bridge 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.

List of Algorithms

List of Figures

List of Tables

1 Introduction

"I'm going for fearsome here, but I just don't feel it! I think I'm just coming off as annoying."

— Rex, *Toy Story*

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 (see for instance, Black and Scholes [1973] and Merton [1973, 1976]) to the present (for instance, Eraker et al. [2003] and Barndorff-Nielsen and Shephard [2004]). Other applications can be easily found within both the physical sciences (Picchini et al. [2009]) and life sciences (Golightly and Wilkinson [2006, 2008]) to name but a few. A jump diffusion $V : \mathbb{R} \to \mathbb{R}$ is a Markov process, which in this thesis we define to be the solution to a stochastic differential equation (SDE) of the following form (denoting $V_{t-} := \lim_{s \uparrow t} V_s$),

$$\mathrm{d}V_t = \beta(V_{t\text{-}})\,\mathrm{d}t + \sigma(V_{t\text{-}})\,\mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\mu}, \qquad V_0 = v \in \mathbb{R}, \;\; t \in [0, T], \tag{1.1}$$

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 for instance [Øksendal and Sulem, 2004, Chap. 1] and [Platen and Bruti-Liberati, 2010, Chap. 1.9]). To apply the methodology developed within this thesis we primarily restrict our attention to univariate diffusions and require a number of additional conditions on the coefficients of (1.1), details and a discussion of which can be found in Section 1.3.

Motivated by the wide range of possible applications we are typically interested in (directly or indirectly) the measure of V on the path space induced by (1.1), denoted $\mathbb{T}^v$. As $\mathbb{T}^v$ is typically not explicitly known then in order to compute expected values $\mathbb{E}_{\mathbb{T}^v}[h(V)]$, for various test functions h, we can construct a Monte Carlo estimator. In particular, if it is possible to draw independently $V^{(1)}, V^{(2)}, \ldots, V^{(N)} \sim \mathbb{T}^v$ then by applying the strong law of large numbers we can construct a consistent estimator of the expectation (unbiasedness following directly by linearity),

$$\text{w.p. 1:} \quad \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{n} h(V^{(i)}) = \mathbb{E}_{\mathbb{T}^v}[h(V)]\,. \tag{1.2}$$

Unfortunately, as diffusion sample paths are infinite dimensional random variables it isn't possible to draw an entire sample path from $\mathbb{T}^v$ – 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 (1.2). 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 Figures 1.0.1(a), 1.0.1(b) and 1.0.1(c) we are interested in whether a simulated sample path $V \sim \mathbb{T}^v$, crosses some barrier (i.e. for some set A we have $h := \mathbb{1}(V \in A)$). 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 1.0.1(c), even in the case where $\mathbb{T}^v$ is known (for instance when $\mathbb{T}^v$ is Wiener measure) and the barrier is known in advance and has a simplistic form, there may still not exist any exact approach to evaluate barrier crossing.

Diffusion sample paths can be simulated approximately at a finite collection of time points by *discretisation* (see for instance Jacod and Protter [2012], Kloeden and Platen [1992] and Platen and Bruti-Liberati [2010] for an extensive account of such methods), noting that as Brownian motion has a Gaussian transition density then over short intervals the transition density of (1.1) can be approximated by one 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 (for instance, of size Δt), then iteratively (at each mesh point) fixing the coefficients and simulating the sample path to the next mesh point.

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 $\mathbb{T}^v$. More complex discretisation schemes exist, 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 1.0.2, 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 (1.2). If $\mathbb{T}^v$ has a highly non-linear drift, or includes a compound Poisson 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 1.0.1, mesh based discretisation schemes don't sufficiently characterise simulated sample paths for the evaluation of h.

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 [Beskos and Roberts, 2005; Beskos et al., 2006a, 2008; Chen and Huang] and jump diffusions [Casella and Roberts, 2010; Giesecke and Smelov, Forthcoming; Gonçalves and Roberts, 2013]. These algorithms are based on rejection sampling, noting that sample paths can be drawn from the (target) measure $\mathbb{T}^v$ by instead drawing sample paths from an equivalent proposal measure $\mathbb{P}^v$, and accepting or rejecting them with probability proportional to the Radon-Nikodým derivative of $\mathbb{T}^v$ with respect to $\mathbb{P}^v$. 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 1.0.1.

The key contribution of this thesis 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 ($V \sim \mathbb{T}^v$), that can be simulated without any approximation error by means of a proposal sample path drawn from an equivalent proposal measure ($\mathbb{P}^v$) and accepted with probability proportional to $\frac{\mathrm{d}\mathbb{T}^v}{\mathrm{d}\mathbb{P}^v}$, which is sufficient to restore the sample path at any finite collection of time points exactly with finite computation where $V|\mathcal{S} \sim \mathbb{P}^v|\mathcal{S}$. 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 $\mathbb{P}^v$ 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 $V \sim \mathbb{P}^v$ 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 $R(V) = \inf\{i \in \mathbb{N} : \forall u \in [0, T], V_u \in [v - i, v + i]\}$ 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, $R(V) \sim \mathcal{R} := \mathbb{P}^v \circ R^{-1}$.*

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

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.

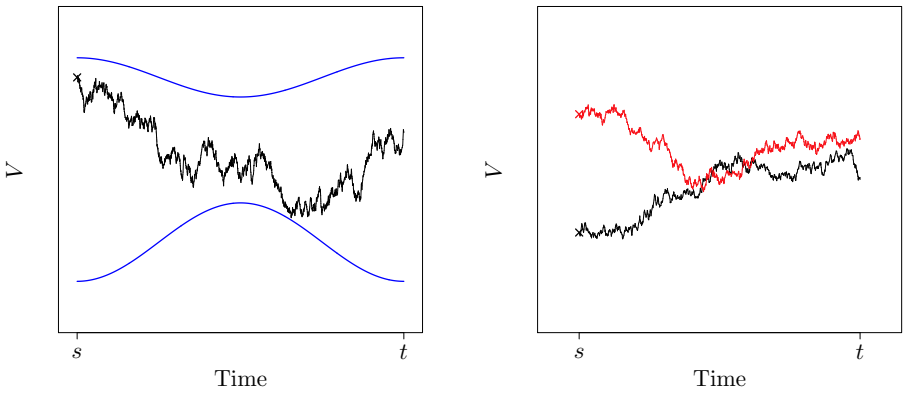
Principle 3 (Path Restoration)**.** *Any finite collection of intermediate (inference) points, conditional on the skeleton, can be simulated exactly,* $V_{t_1}, \ldots, V_{t_n} \sim \mathbb{P}^v|\mathcal{S}$.

In developing a methodological framework for simulating exact skeletons of diffusion sample paths we make several additional contributions. We make a number of methodological improvements to existing exact algorithms with potential for substantial computational benefit and extension of the applicability of existing algorithms. In addition, we introduce a novel class of *adaptive exact algorithms* for diffusions, diffusion bridges, jump diffusions and jump diffusion bridges, underpinned by new results for simulating Brownian path space probabilities (which are of separate interest) and *layered Brownian motion* (Brownian motion conditioned to remain in a layer).

By application of the results developed in this thesis we present methodology for particle filtering for partially observed (jump) diffusions. Furthermore, we present a significant extension to ϵ-*Strong Simulation* methodology (recently introduced by Beskos et al. [2012], and allowing 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 exactness of the methodology.

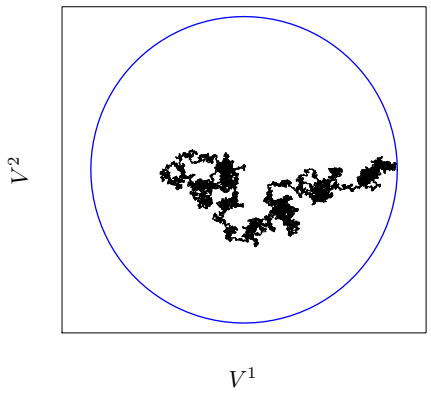
Finally, we highlight a number of possible applications of the methodology developed in this thesis by returning to the examples introduced in Figure 1.0.1. We demonstrate that it is possible not only to simulate skeletons exactly from the correct target measure but also to evaluate exactly whether or 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 thesis, for instance, the evaluation of path space integrals and barrier hitting times to arbitrary precision, among many others.

The remainder of this introductory chapter is structured as follows: In Section 1.1 we detail the structure of this thesis and how the content of the chapters relate to one another. In Section 1.2 we provide a summary of the key contributions of this thesis. Finally, in Section 1.3 we briefly discuss the recurrent conditions required to implement the methodology developed in this thesis.



(a) Non-linear two sided barrier

(b) Diffusion barrier



(c) 2-D circular barrier

Figure 1.0.1: Examples of test functions in which evaluation requires the characterisation of an entire sample path.

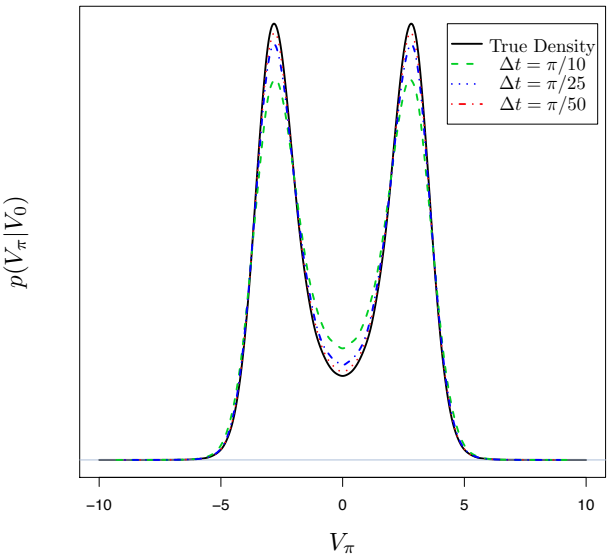


Figure 1.0.2: Density of V_π and approximations given by an Euler discretisation with various mesh sizes, given $V_0 = 0$ where $\mathrm{d}V_t = \sin(V_t)\,\mathrm{d}t + \mathrm{d}W_t$.

1.1 Thesis Structure

In Figure 1.1.1 we present a schematic diagram of how the content of each of the chapters in this thesis relate to one another. This thesis is essentially broken into two key parts: Part I is primarily a review of relevant existing literature pertinent to this thesis; whereas in Part II we present our methodology (as discussed in the introductory remarks of this chapter and summarised in Section 1.2) within the context of existing literature.

Part I is comprised of Chapters 2, 3 and 4. In Chapter 2 we selectively review a number of elementary Monte Carlo methods employed within this thesis, including Monte Carlo methods for simulating finite dimensional sample path trajectories of Brownian motion and related stochastic processes, and Monte Carlo methods for simulating sample paths of Poisson processes. In Chapter 3 we motivate and present a review of sequential Monte Carlo methods. We conclude Part I of this thesis in Chapter 4, where we provide an introductory level overview of elements of stochastic calculus required in this thesis and briefly review existing discretisation methods for simulating diffusion sample paths.

Part II is comprised of Chapters 5, 6, 7 and 8. In Chapter 5 we present a novel mathematical framework for simulating (jump) diffusion and (jump) diffusion bridge sample path skeletons without approximation error (exact algorithms). In Chapter 6 we present new results for simulating quantities related to various Brownian bridge path space constructions, which together allow the simulation of layered Brownian bridge sample path skeletons, and hence the implementation of the exact algorithms in Chapter 5. In Chapter 7 we present methodology for particle filtering for partially observed (jump) diffusions. Finally, we conclude Part II of this thesis in Chapter 8 where we outline methodology for simulating upper and lower bounding processes which almost surely constrain (jump) diffusion and (jump) diffusion bridge sample paths to any specified tolerance (ϵ-strong simulation). We additionally demonstrate that our methodology can be applied to determine exactly whether a diffusion or jump diffusion sample path crosses various types of non-trivial barrier.

Finally, in Chapter 9 we conclude this thesis, presenting a summary along with avenues for possible future research.

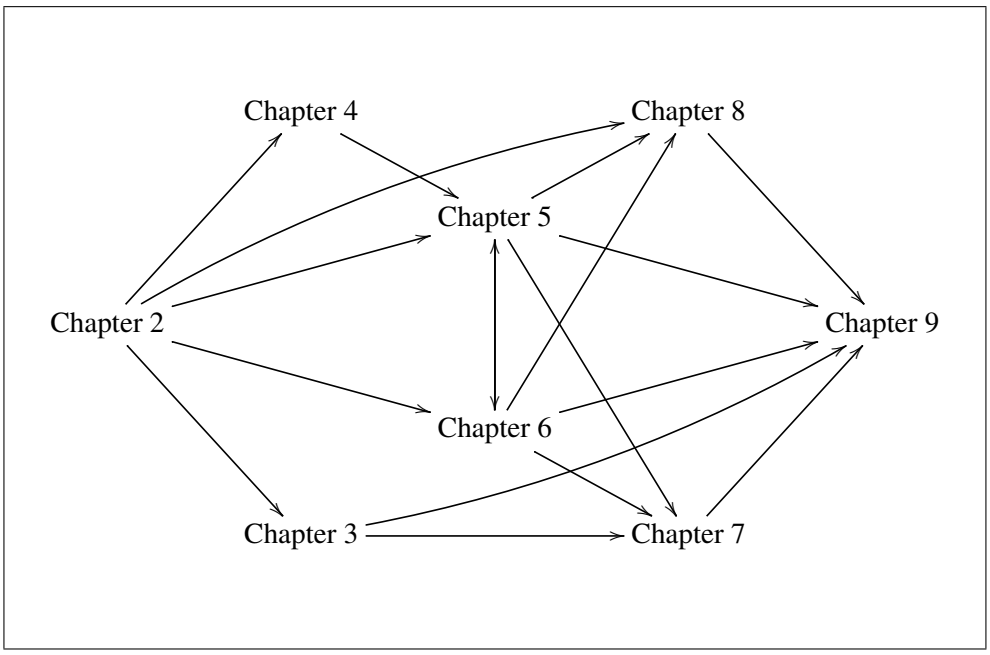


Figure 1.1.1: Schematic diagram of thesis structure.

1.2 Thesis Contributions

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

- A mathematical framework for constructing exact algorithms, along with a new class of adaptive exact algorithms, which allow both (jump) diffusion and (jump) diffusion bridge sample path skeletons to be simulated without discretisation error (see Chapter 5).
- An extension of existing exact algorithms to satisfy Principle 3 (see Chapter 5 and in particular Sections 5.1.1, 5.3.1, 5.4 and 6.2), including a number of general methodological improvements (see Chapters 5 and 6).
- Methodology for simulating unbiasedly events of probability corresponding to various Brownian path space probabilities and simulating layered Brownian bridge sample path trajectories (see Chapter 6).
- An extension of methodology for particle filters to the setting in which the transition density of the latent process is governed by a jump diffusion (see Chapter 7).

– Methodology for the ϵ-strong simulation of (jump) diffusion and (jump) diffusion bridge sample paths, along with a novel exact algorithm based on this construction (see Chapter 8).

– A new approach 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 Section 8.3). This work is presented along with examples based on the illustrations in Figure 1.0.1.

Throughout this thesis we make a number of additional minor contributions, which are pointed out in the relevant sections.

1.3 Thesis Conditions

Throughout this thesis a number of the methodological results and algorithms that we present share a common set of recurrent conditions. For convenience we briefly introduce these conditions and motivate their requirement in this section, referring back to them as necessary (a number of the methodological results and algorithms require additional conditions, however these are stated and motivated in the appropriate sections). The motivation for the conditions in this section is to establish a number of results (Results 1–4) that we also summarise in this section, however these results are revisited as required and extended upon in later chapters.

To present our work in full generality we assume Conditions 1–5 hold (see below), however, these conditions can be difficult to check and so in Section 1.3.1 we discuss verifiable sufficient conditions under which Results 1–4 hold.

Condition 1 (Solutions)**.** *The coefficients of (1.1) 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 (1.1) into one with unit volatility (letting $\psi_1, \ldots, \psi_{N_T}$ denote the jump times in the interval $[0, T]$, $\psi_0 := 0$ and $\psi_{N_T+1}- := \psi_{N_T+1} := T$, and denoting by V^{cts} as the continuous component of V),

Result 1 (Lamperti Transform [Kloeden and Platen, 1992, Chap. 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). Denoting by $N_t := \sum_{i\geq 1} \mathbb{1}\{\psi_i \leq t\}$ a Poisson jump counting process (with respect to $\mathcal{F}_t^N$) and applying Itô's formula for jump diffusions to find $\mathrm{d}X_t$ we have,*

$$\begin{aligned}\mathrm{d}X_t &= \left[\eta'\,\mathrm{d}V_t^{cts} + \eta''\left(\mathrm{d}V_t^{cts}\right)^2/2\right] + \left[\eta\left(V_{t\text{-}} + \mu\left(V_{t\text{-}}\right)\right) - \eta\left(V_{t\text{-}}\right)\right]\mathrm{d}N_t \\ &= \underbrace{\left[\frac{\beta\left(\eta^{-1}\left(X_{t\text{-}}\right)\right)}{\sigma\left(\eta^{-1}\left(X_{t\text{-}}\right)\right)} - \frac{\sigma'\left(\eta^{-1}\left(X_{t\text{-}}\right)\right)}{2}\right]}_{\alpha(X_{t\text{-}})}\mathrm{d}t + \mathrm{d}W_t + \underbrace{\left(\eta\left[\eta^{-1}\left(X_{t\text{-}}\right) + \mu\left(\eta^{-1}\left(X_{t\text{-}}\right)\right)\right] - X_{t\text{-}}\right)\mathrm{d}N_t}_{\mathrm{d}J_t^{\lambda,\nu}}.\end{aligned} \tag{1.3}$$

This transformation is typically possible for univariate diffusions and for a significant class of multivariate diffusions (see for instance, Aït-Sahalia [2008]). We revisit the Lamperti transform in Section 4.1.2, providing a more detailed account.

As a consequence of Result 1, in this thesis we frequently restrict our attention to SDEs with unit volatility coefficient as in (1.3) without loss of generality. As such we introduce the following simplifying notation. In particular, we denote by $\mathbb{Q}_{0,T}^x$ the measure induced by (1.3), by $\mathbb{W}_{0,T}^x$ the measure induced by the driftless version of (1.3), $A(u) := \int_0^u \alpha(y)\,\mathrm{d}y$ and set $\phi(X_s) := \alpha^2(X_s)/2 + \alpha'(X_s)/2$. If $\lambda = 0$ in (1.3) 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 $\Phi \leq \inf_{s\in[0,T]} \phi(X_s)$.*

It is necessary within this thesis 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. We provide a more detailed account of Result 2 and the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^x$ with respect to alternate measures in Section 4.1.3.

Result 2 (Radon-Nikodým derivative [Øksendal and Sulem, 2004; Platen and Bruti-Liberati, 2010])**.** *Under Conditions 1–4, the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^x$ with respect*

to $\mathbb{W}^x_{0,T}$ exists and is given by Girsanov's formula.

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{\int_0^T \alpha\left(X_s\right)\,\mathrm{d}W_s - \frac{1}{2}\int_0^T \alpha^2(X_s)\,\mathrm{d}s\right\}. \tag{1.4}$$

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,

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_s)\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}. \tag{1.5}$$

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

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_s)\,\mathrm{d}s\right\}. \tag{1.6}$$

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 $\forall T \leq T_0$:*

$$c(y; x, T) := \int_{\mathbb{R}} \exp\left\{A(y) - \frac{(y-x)^2}{2T}\right\}\mathrm{d}y < \infty. \tag{1.7}$$

Throughout this thesis 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 $\forall\ t \in [0, T]$, $X_t(\omega) \in [\ell, \upsilon]$ $\exists\, L_X := L\left(X(\omega)\right) \in \mathbb{R}, U_X := U\left(X(\omega)\right) \in \mathbb{R}$ such that $\forall\ t \in [0, T]$, $\phi\left(X_t(\omega)\right) \in [L_X, U_X]$.*

1.3.1 Verifiable Sufficient Conditions

As discussed in [Øksendal and Sulem, 2004, Thm. 1.19] and [Mao and Yuan, 2006, Sec. 3.3], to ensure Condition 1 it is sufficient to assume that the coefficients of (1.1) 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),

$$|\beta(x)|^2 + \|\sigma(x)\|^2 + \int_{\mathbb{R}} |f_\mu(x,z)|^2 \lambda(\,\mathrm{d}z) \leq C_1(1+|x|^2), \quad \forall x \in \mathbb{R}, \tag{1.8}$$

$$|\beta(x)-\beta(y)|^2 + \|\sigma(x)-\sigma(y)\|^2 + \int_{\mathbb{R}} |f_\mu(x,z)-f_\mu(y,z)|^2 \lambda(\,\mathrm{d}z) \leq C_2(|x-y|^2), \quad \forall x, y \in \mathbb{R}. \tag{1.9}$$

(1.8) and (1.9) together with Condition 2 are sufficient for the purposes of ensuring Conditions 1, 3, 4 and 5 hold, but are not necessary. Although easy to verify, (1.8) and (1.9) 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 (1.1) we have $\lambda = 0$) then, by application of the Mean Value Theorem, Condition 2 ensures β and σ are locally Lipschitz and so (1.1) admits a unique weak solution (see Øksendal [2007]) and so Condition 1 holds. In particular, in this setting Results 1–4 will hold under Conditions 2, 3 and 5.

Part I

Literature Review

2 Monte Carlo Methods

"If the only tool you have is a hammer, you tend to see every problem as a nail."

— Abraham Maslow

In this chapter we review a number of elementary *Monte Carlo methods* which are of particular relevance to the methodology developed in this thesis as outlined in Chapter 1 and explored later. A fuller account of the methods discussed in this chapter along with the broader literature and applications can be found in a number of texts (see for instance Robert and Casella [2004], Kloeden and Platen [1992] and Kingman [1992]).

Monte Carlo methods are a class of statistical algorithms which, by means of exploiting the comparatively recent introduction of cheap and powerful computing, use the simulation of random processes to draw inference on quantities of interest. To motivate this we consider expectations of the following form which may, or may not be, analytically tractable (where h is some test function, π some probability density and X is a random variable with law π),

$$\mathbb{E}_\pi [h(X)] := \int_{\mathbb{R}} h(x) \cdot \pi(x) \, \mathrm{d}x. \tag{2.1}$$

Employing the notion first proposed by Metropolis and Ulam [1949] (which we term *Naïve Monte Carlo* and present in Algorithm 2.0.1), note that if we were to simply simulate large numbers of *random samples*[1] from the density π (and this were possible),

[1] **Random Number Generators**: In this thesis we assume that it is possible to simulate any number of independent uniform random variables ($u_1, u_2, \ldots \sim \mathrm{U}[0,1]$). Some care ensuring generated random variables are truly random has to be taken (see for instance [Ripley, 1987, Chap. 1] and [Robert and Casella, 2004, Chap. 2]). Particular consideration of random number generators in the context of diffusions can be

then these could be used to compute an estimate of the expectation in (2.1) (which we term the *Monte Carlo estimator*). More precisely, if we were to draw independently $X_1, X_2, \ldots, X_N \sim \pi$, then by applying the *Strong Law of Large Numbers* (SLLN) we can construct a consistent estimator of the expectation (with unbiasedness following directly by linearity),

$$\text{w.p. 1: } \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} h\left(X_i\right) = \mathbb{E}_\pi\left[h(X)\right]. \tag{2.2}$$

Furthermore, provided $\mathbb{V}\text{ar}_\pi[h(X)] =: \sigma^2 < \infty$, it can be shown by the *Central Limit Theorem* (CLT) that we have,

$$\lim_{N\to\infty} \sqrt{N}\left[\mathbb{E}_\pi\left[h(X)\right] - \frac{1}{N}\sum_{i=1}^{n} h\left(X_i\right)\right] \overset{\mathcal{D}}{=} \xi, \qquad \text{where } \xi \sim \text{N}(0, \sigma^2). \tag{2.3}$$

The number of algorithms and applications which stem from this simple simulation approach is vast. The central difficulty in employing this approach in practice is being able to simulate independent and identically distributed random variables from the density π. Addressing this problem is the theme linking the various Monte Carlo methods we review in the remainder of this chapter.

Algorithm 2.0.1 Naïve Monte Carlo Algorithm (N random samples) [Metropolis and Ulam, 1949].

---

1. For i in 1 to N simulate $X_i \sim \pi$.
2. Compute the Monte Carlo estimator $\widehat{I(X)} := \frac{1}{N}\sum_{i=1}^{N} h\left(X_i\right)$.

2.1 Inversion Sampling

Inversion Sampling (see for example [Devroye, 1986, Part 3 Chap. 2]) is a method of sampling from a density π, by inverting a randomly drawn uniform random variable $u \sim \text{U}[0, 1]$. Denoting F_π as the cumulative distribution function (CDF) of π, we define the

found in [Kloeden and Platen, 1992, Chap. 1.9].

generalised inverse as follows,

$$F_\pi^{-1}(u) := \inf_x \{F_\pi(x) \geq u\}. \tag{2.4}$$

Noting that $F_\pi(x) \in [0,1]\ \forall x \in \mathbb{R}$, it is possible to draw from π by generating and transforming a uniform random variable $u \sim \mathrm{U}[0,1]$ as illustrated in Figure 2.1.1 and presented in Algorithm 2.1.1.

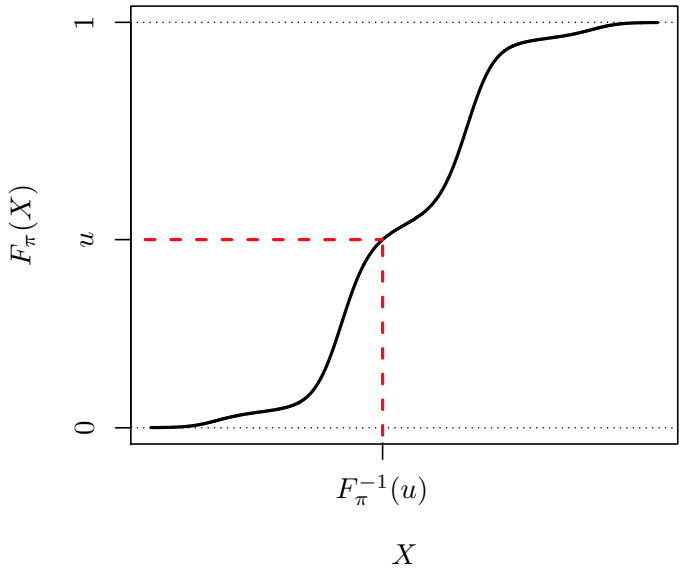


Figure 2.1.1: An illustration of the simulation of $X \sim \pi$ by means of Inversion Sampling.

Algorithm 2.1.1 Inversion Sampling Algorithm (N random samples) [Devroye, 1986, Part 3 Chap. 2].

1. For i in 1 to N simulate $u_i \sim \mathrm{U}[0,1]$ and set $X_i = F_\pi^{-1}(u_i)$.

This notion can be formalised by considering the following,

$$\mathbb{P}\left(F_\pi^{-1}(u) \leq x\right) = \mathbb{P}\left(u \leq F_\pi(x)\right) = F_\pi(x) = \mathbb{P}(X \leq x). \tag{2.5}$$

It should be noted that this algorithm can be extended to draw in a computationally efficient manner from $F_\pi(x)$ conditional on $x \in [\ell, \upsilon]$ (or similarly a percentile of $F_\pi(x)$) by modifying Algorithm 2.1.1 such that $u_i \sim \mathrm{U}\left[F_\pi^{-1}(\ell), F_\pi^{-1}(\upsilon)\right]$.

2.2 Composition Sampling

Suppose we wish to draw a random sample from a density formed through a weighted composition of a number of other densities. In particular (denoting $w_j\,(\geq 0)$ as the contribution of the density π_j to the composition such that $\sum_{i=1}^{n} w_i = 1$) suppose we have,

$$\pi = \sum_{j=1}^{r} w_j \pi_j, \tag{2.6}$$

an example of which is illustrated in Figure 2.2.1.

As shown in [Ripley, 1987, Chap. 3.2] and as outlined in Algorithm 2.2.1, we can draw a random sample from π by first sampling a contributing density π_j with probability proportional to its weight (w_j), then drawing from π_j.

Algorithm 2.2.1 Composition Sampling Algorithm (N random samples) [Ripley, 1987].

1. For i in 1 to N sample $J_i \sim$ categorical$(w_1, \ldots, w_r)$ and sample $X_i|(J_i = j) \sim \pi_j$.

2.3 Demarginalisation

Demarginalisation 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 from $\pi(x)$, but this is not directly possible. However, suppose that with the introduction of an auxiliary variable Y, sampling from

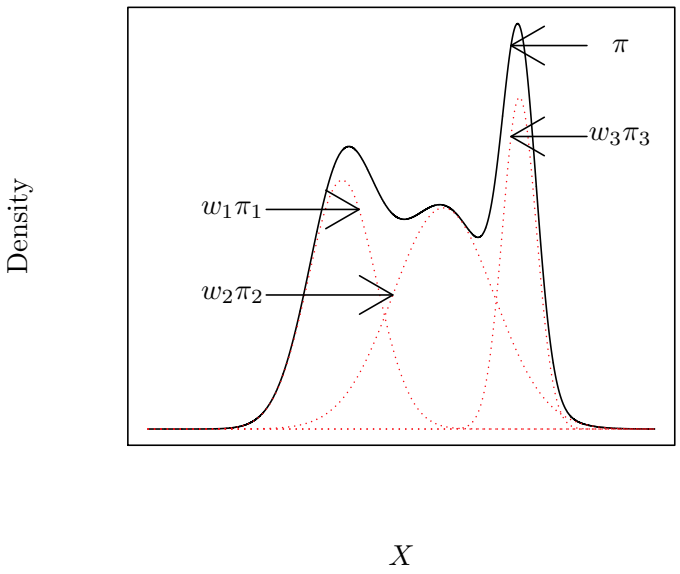


Figure 2.2.1: An illustration of a density composed of three weighted Normal densities.

$\pi(x|y)$ is possible and $\pi(x, y)$ admits $\pi(x)$ as a marginal. In particular, we have,

$$\pi(x) = \int_{\mathcal{Y}} \pi(x|y) \cdot \pi(y) \, \mathrm{d}y. \tag{2.7}$$

We can sample from $\pi(x)$ by first sampling Y from $\pi(y)$ and then sampling from $\pi(x|y)$. This algorithm can be viewed as a black box to generate samples from $\pi(x)$ – Y can be simply marginalised out (i.e. 'thrown' away). A fuller account of demarginalisation can be found in [Robert and Casella, 2004, Chap. 5.3], however it is worth noting that composition sampling (see Section 2.2) is an example of demarginalisation.

2.4 Rejection Sampling

Rejection sampling (von Neumann [1951]) is a Monte Carlo method in which we can sample from some (inaccessible) target density π by means of an accessible dominating

density q, where the distribution of q with respect to the distribution of π is absolutely continuous with bounded Radon-Nikodým derivative. In particular, if we can find a bound M such that,

$$\sup_{x \in \mathbb{R}} \frac{\mathrm{d}\pi}{\mathrm{d}q}(x) \leq M < \infty, \tag{2.8}$$

then drawing $X \sim q$ and accepting the draw ($I = 1$) with probability $P_q(X) := \frac{1}{M}\frac{\mathrm{d}\pi}{\mathrm{d}q}(X) \in [0, 1]$ then $(X|I = 1) \sim \pi$. This argument is presented in Algorithm 2.4.1.

Algorithm 2.4.1 Rejection Sampling Algorithm (N random samples) [von Neumann, 1951].

1. For i in 1 to N,
   (a) Simulate $X_i \sim q$ and $u \sim \mathrm{U}[0, 1]$.
   (b) If $u \leq \dfrac{\pi(X_i)}{M \cdot q(X_i)}$ then accept, else reject and return to Step 1a.

An intuitive way to consider rejection sampling is as follows (which is best read in conjunction with Figure 2.4.1 and provides useful insight to the more complicated methodology later in this thesis). Suppose we wanted to simulate N points uniformly on the graph $\mathcal{G}_A := \{(x, y) \in \mathbb{R} \times \mathbb{R}_+ : \pi(x) \leq y\}$, however due to the inaccessibility of π this was not possible. Now if there existed an accessible density q such that $\mathcal{G}_P := \{(x, y) \in \mathbb{R} \times \mathbb{R}_+ : M \cdot q(x) \leq y\} \supseteq \mathcal{G}_A$, we could simulate points uniformly on $\mathcal{G}_P$ by employing a two stage algorithm whereby we first simulate each of their locations on the x-axis, $X_1, \ldots \overset{\text{iid}}{\sim} q$, and then their location on the y-axis, $u_1, \ldots \overset{\text{iid}}{\sim} \mathrm{U}\left[0, M \cdot q\left(X_i\right)\right]$. Points can be uniformly simulated on $\mathcal{G}_P$ and retained if they fall in $\mathcal{G}_A$ until the N desired samples are obtained. Now as $\mathcal{G}_A \subseteq \mathcal{G}_P$, if we consider the restriction of the set of simulated points to those which lie in $\mathcal{G}_A$ then those points will be indistinguishable from an alternate scheme in which points were uniformly simulated on $\mathcal{G}_A$. More formally, if we simulate $X_1, \ldots \overset{\text{iid}}{\sim} q$ and for each sample X_i with probability $\pi(X_i)/\left(M \cdot q(X_i)\right)$ accept the sample, then considering

the conditional cumulative distribution function we have,

$$\begin{aligned}
\mathbb{P}\left(X \le x \,\middle|\, u \le \frac{\pi(X)}{M \cdot q(X)}\right) &= \frac{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{1} \mathbb{1}_{[-\infty,x]}(z) \cdot \mathbb{1}_{[0,\pi(z)/Mq(z)]}(u) \cdot q(z)\,\mathrm{d}u\,\mathrm{d}z}{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{1} \mathbb{1}_{[0,\pi(z)/Mq(z)]}(u) \cdot q(z)\,\mathrm{d}u\,\mathrm{d}z} \\
&= \frac{\displaystyle\int_{-\infty}^{x} \frac{\pi(z)}{M \cdot q(z)} \cdot q(z)\,\mathrm{d}z}{\displaystyle\int_{-\infty}^{\infty} \frac{\pi(z)}{M \cdot q(z)} \cdot q(z)\,\mathrm{d}z} \\
&= \frac{1/M \cdot \mathbb{P}_{\pi}(X \le x)}{1/M \cdot 1} = F_{\pi}(x). \qquad (2.9)
\end{aligned}$$

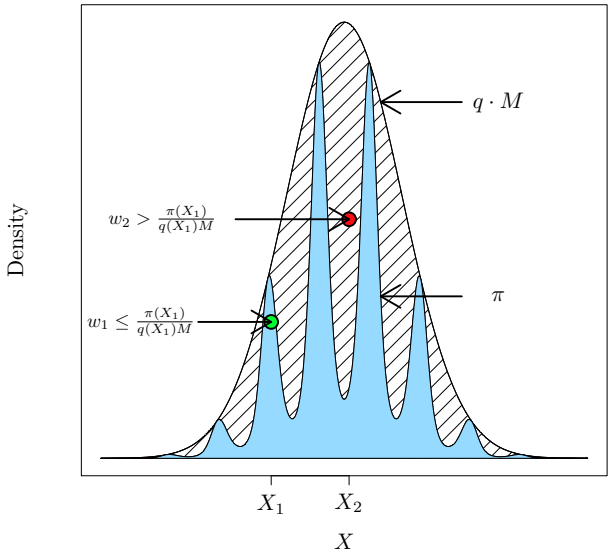


Figure 2.4.1: An illustration of the simulation of $X \sim \pi$ by means of Rejection Sampling.

It is worth noting that upon simulating $X \sim q$, the probability that (X, u) falls on $\mathcal{G}_A$ is precisely $\pi(X)/\left(M \cdot q(X)\right)$. Furthermore, any jointly simulated point (X, u), will fall on $\mathcal{G}_A$ with probability $P_q(X) := \mathbb{E}_q\left[\frac{\pi(X)}{M \cdot q(X)}\right] = 1/M$, which is simply the ratio of $\mathcal{G}_A$ to $\mathcal{G}_P$. Clearly the number of draws from q required to obtain a single draw from π is geometri-

cally distributed with mean M. As such rejection sampling can be made computationally more efficient by finding a density q which is as close as possible to π (i.e., such that the bound M is as close as possible to 1).

2.5 Importance Sampling

Clearly rejection sampling can be wasteful as a large number of random samples are drawn and then discarded. In particular, upon evaluating whether to accept or reject any particular sample, some useful information about the density at that point is gleaned.

A natural approach to incorporating this information (known as *Importance Sampling*), is to instead evaluate our Monte Carlo estimator (2.2) using a weighted sample of draws from some accessible proposal density q (with appropriate conditions which we address at the end of this section). In particular (denoting $w(X) = \pi(X)/q(X)$ which we call the sample *weight*) we have,

$$\mathbb{E}_\pi [h(X)] = \int_{-\infty}^{\infty} h(x) \cdot \pi(x) \, \mathrm{d}x = \int_{-\infty}^{\infty} h(x) \cdot \frac{\pi(x)}{q(x)} \cdot q(x) \, \mathrm{d}x = \mathbb{E}_q [h(X) \cdot w(X)]. \quad (2.10)$$

Now, in analogous form to the Monte Carlo estimator presented in (2.2), if we were to draw independently $X_1, X_2, \ldots, X_N \sim q$ then by applying the SLLN we can construct a consistent estimator of our desired expectation (with unbiasedness following by linearity),

$$\text{w.p. 1:} \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} h(X_i) \cdot w(X_i) = \mathbb{E}_\pi [h(X)] . \quad (2.11)$$

Importance sampling is a classical Monte Carlo method which dates back to at least Goertzel [1949] and Kahn [1949], yet (although a relatively simple extension of Naïve Monte Carlo) is the central idea behind the more modern Sequential Monte Carlo methods we discuss in Chapter 3.

One of the key advantages of importance sampling is that we can, rather remarkably, construct an asymptotically unbiased estimator of $\mathbb{E}_\pi [h(X)]$ even if we only know our target density π to some normalising constant Z (i.e. we know $\pi^* := \pi \cdot Z$). In particular,

denoting $w^*(X) := \pi^*(X)/q(X)$ we have,

$$\begin{aligned}\frac{\mathbb{E}_q\left[h(X)w^*(X)\right]}{\mathbb{E}_q\left[w^*(X)\right]} &:= \frac{\int_{-\infty}^{\infty} h(x)\cdot\frac{\pi(x)\cdot Z}{q(x)}\cdot q(x)\,\mathrm{d}x}{\int_{-\infty}^{\infty}\frac{\pi(x)\cdot Z}{q(x)}\cdot q(x)\,\mathrm{d}x}\\ &= \frac{\int_{-\infty}^{\infty} h(x)\cdot\pi(x)\,\mathrm{d}x}{\int_{-\infty}^{\infty}\pi(x)\,\mathrm{d}x}\\ &= \mathbb{E}_\pi\left[h(X)\right]. \end{aligned} \tag{2.12}$$

So, by application of Slutsky's lemma[2], and denoting $w(X_i) := w^*(X_i)/\sum_{j=1}^N w^*(X_j)$, then if we draw independently $X_1, X_2, \ldots, X_N \sim q$ we have our required asymptotically unbiased estimator,

$$\text{w.p. 1: } \lim_{N\to\infty}\sum_{i=1}^{N}\frac{h\left(X_i\right)\cdot w^*\left(X_i\right)}{\sum_{j=1}^N w^*(X_j)} = \lim_{N\to\infty}\sum_{i=1}^{N} h\left(X_i\right)\cdot w(X_i) = \mathbb{E}_\pi\left[h(X)\right]. \tag{2.13}$$

We term this particular version of importance sampling as *Self-Normalised Importance Sampling* and present the synthesis of the above argument in Algorithm 2.5.1. In Figure 2.5.1 we present an illustration of random samples being drawn by means of self-normalised importance sampling.

Algorithm 2.5.1 Self-Normalised Importance Sampling Algorithm (N random samples) [Goertzel, 1949; Kahn, 1949].

---

1. For i in 1 to N simulate $X_i \sim q$ and set $w^*(X_i) = \pi^*(X_i)/q(X_i)$.
2. For i in 1 to N set $w(X_i) = w^*(X_i)/\sum_{j=1}^N w^*(X_j)$.

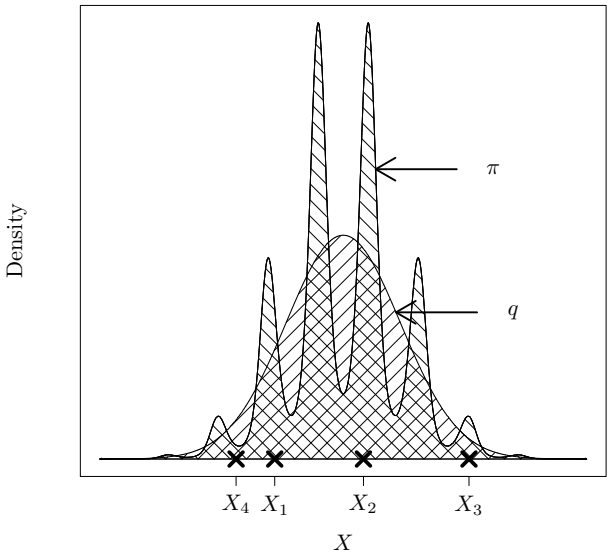
Clearly some care has to be taken when choosing an appropriate proposal density q. At a minimum we want to ensure that the resulting estimator (2.11) has finite variance, $\mathbb{V}\mathrm{ar}_\pi\left[h(X)\right] < \infty$. As $\mathbb{E}_\pi\left[h(X)\right] = \mathbb{E}_q\left[h(X)\cdot w(X)\right]$ we can determine whether or not our importance sampler has finite variance by simply considering the second moment,

$$\mathbb{E}_q\left[\left(h(X)\cdot w(X)\right)^2\right] = \int_{-\infty}^{\infty} h^2(x)\frac{\pi^2(x)}{q^2(x)}q(x)\,\mathrm{d}x = \mathbb{E}_\pi\left[h^2(X)\cdot w(X)\right]. \tag{2.14}$$

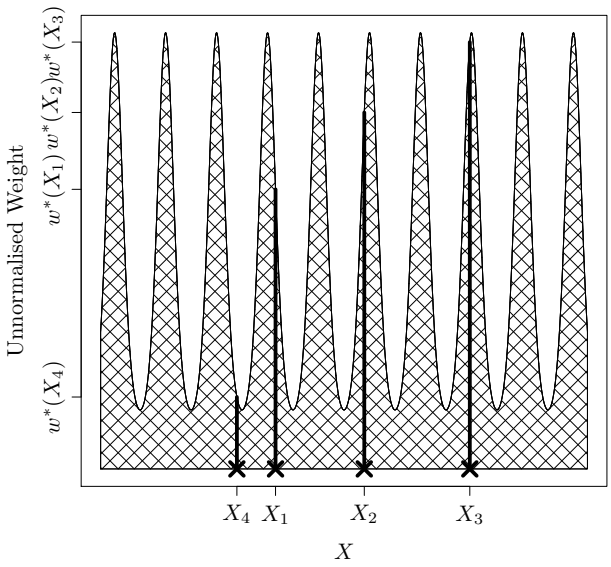
[2] *Slutsky's lemma (Slutsky [1925]):* Let X_n converge in distribution to some random element X and let Y_n converge in probability to a constant c then the following holds - $X_n + Y_n \xrightarrow{\mathcal{D}} X + c$; $Y_n X_n \xrightarrow{\mathcal{D}} cX$; $\frac{X_n}{Y_n} \xrightarrow{\mathcal{D}} \frac{X}{c}$.

As shown in Geweke [1989] and [Robert and Casella, 2004, Sec. 3.3.2], (2.14) will be finite provided (i) π/q is bounded by some known M (which in the case of q being chosen as in the rejection sampling approach of Section 2.4 will hold), or (ii) if the sample space (denoted by Ω, but in general for our purposes $\Omega := \mathbb{R}$) is compact and $\forall x \in \Omega$ we have $\pi(x) \leq M(x) < \infty$ and $q(x) > \epsilon$. The general principle upon having found proposal densities which result in Monte Carlo estimators with finite variance is to choose the one which results in the lowest variance.

A natural question to ask is when its appropriate to choose an importance sampler over a rejection sampler (as in Section 2.4). Clearly as importance sampling has weaker conditions than rejection sampling there are settings in which only an importance sampler can be employed. However, comparison of the Monte Carlo estimator variance under a rejection and importance sampler in more general settings is not straight forward. If we are to fix the number of proposed random samples then an importance sampler is simply a Rao-Blackwellised rejection sampler (see for instance Casella and Robert [1996]). In particular we could draw N random samples $\{X_i, w(X_i)\}_{i=1}^N$ from an importance sampler and then accept or reject each with probability $w(X_i)$. More typically the number of random samples generated by a rejection sampler is of a random size and this adds complication to making any comparison (for a more detailed discussion see for instance Casella and Robert [1996], [Robert and Casella, 2004, Sec. 3.3.3] and Chen [2005]), however typically we would expect an importance sampler to be an improvement over a rejection sampler.



(a) Graph of the proposal density q along with independent random samples $X_1, \ldots, X_4 \sim q$, overlaid with the target density π.



(b) Graph of $\frac{\pi}{q}$ for the assignment of weights for the drawn samples $X_1, \ldots, X_4$. The size of the vertical bar associated with each sample represents its relative weight.

Figure 2.5.1: An illustration of the simulation of $X \sim \pi$ by means of Self-Normalised Importance Sampling.

2.6 Series Sampling

Series Sampling (Devroye [1980], [Devroye, 1986, Part 4 Chap. 5]) is a method of drawing random samples from a target density π which cannot be evaluated exactly at any point. We will suppose there exists an accessible dominating density q (as in Section 2.4) and that we can find some upper and lower monotonically convergent bounding functions ($\lim_{n\to\infty} \pi_n^{\uparrow} \to \pi$ and $\lim_{n\to\infty} \pi_n^{\downarrow} \to \pi$ such that for any $x \in \mathbb{R}$ and $\epsilon > 0$ $\exists$ an $n^*(x)$ such that $\forall\, n \geq n^*(x)$ we have $\pi_n^{\uparrow}(x) - \pi_n^{\downarrow}(x) < \epsilon$). An illustration of such a density can be found in Figure 2.6.1.

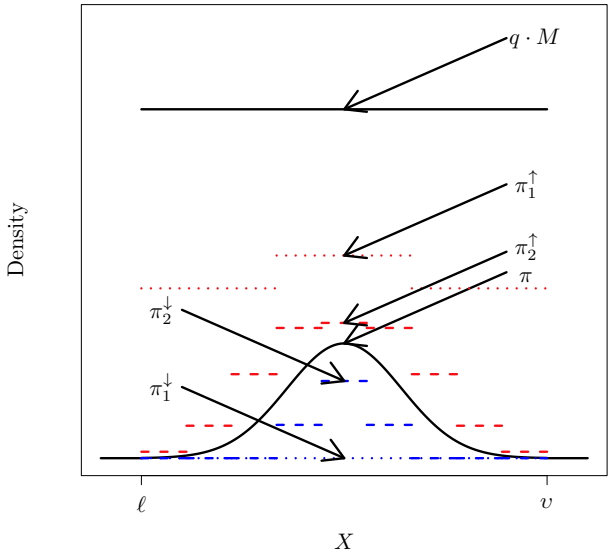


Figure 2.6.1: An illustration of a situation in which Series Sampling can be performed, whereby there exists for the target density π, upper and lower monotonically bounding functions $\pi\uparrow$ and $\pi\downarrow$. In this example π has compact support in the interval $[\ell, \upsilon]$ and the dominating density q, is a uniform density.

We can begin as with the rejection sampling approach taken in Section 2.4 and draw a proposal sample from the dominating density, $X \sim q$. However unlike rejection sampling, we can not evaluate the target density at $\pi(X)$ in order to determine directly whether to accept or reject the sample (with probability $P_q(X) := \frac{\pi(X)}{q(X)\cdot M}$). Instead we employ a retrospective approach to simulate unbiasedly an event of probability $P_q(X)$, guided by

the following theorem,

Theorem 2.6.1 (Series Sampling [Devroye, 1986, Part 4 Chap. 5])**.** *An event of (unknown) probability $p \in [0,1]$, where there exists monotonically increasing and decreasing sequences, $(S_k^+ : k \in \mathbb{Z}_{\geq 0})$ and $(S_k^- : k \in \mathbb{Z}_{\geq 0})$ 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 constructed and simulated (where $u \sim U[0,1]$), noting that as there almost surely exists a finite $K := \inf\{k : u \notin (S_k^-, S_k^+)\}$ we have $\mathbb{1}(u \leq p) = \mathbb{1}(u \leq S_K^-)$ and $\mathbb{E}[\mathbb{1}(u \leq S_K^-)] = p$.*

The key idea is that as for any $X \sim q$ we have upper ($\pi_n^{\uparrow}(X)$) and lower ($\pi_n^{\downarrow}(X)$) convergent bounding sequences we can iteratively evaluate these sequences ($n = 1, 2 \ldots$) until such point that a uniform random variable $u \sim \mathrm{U}[0, q(X) \cdot M]$ lies either below $\pi_n^{\downarrow}(X)$ (in which case it must also lie below $\pi(X)$) or above $\pi_n^{\uparrow}(X)$ (in which case it must also lie above $\pi(X)$). This approach is what is used to construct Algorithm 2.6.1.

Algorithm 2.6.1 Series Sampling Algorithm (N random samples) [Devroye, 1980], [Devroye, 1986, Part 4 Chap. 5].

1. For i in 1 to N,
   (a) Simulate $X_i \sim q$, $u \sim \mathrm{U}[0, q(X_i) \cdot M]$ and set $n = 1$.
   (b) While $u \in \left(\pi_n^{\downarrow}(X_i), \pi_n^{\uparrow}(X_i)\right)$ then $n = n + 1$.
   (c) If $u \leq \pi_n^{\downarrow}(X_i)$ then accept else return to Step 1a.

2.7 Retrospective Bernoulli Sampling

Retrospective Bernoulli Sampling is a method to simulate unbiasedly an event of some unknown probability p, where p can be represented as the limit of an alternating Cauchy sequence $(S_k : k \in \mathbb{Z}_{\geq 0})$. Without loss of generality, throughout this thesis (unless otherwise stated), we will assume we have an alternating Cauchy sequence in which the even terms of the sequence converge from below and the odd terms of the sequence converge from above as follows,

$$0 = S_0 < S_2 < S_4 < S_6 < \ldots < p < \ldots < S_5 < S_3 < S_1 \leq 1 \tag{2.15}$$

As noted in [Beskos et al., 2008, Prop. 1], we can split the alternating Cauchy sequence $(S_k : k \in \mathbb{Z}_{\geq 0})$ into upper and lower subsequences (composed of the odd and even terms respectively) converging to p (in particular, $(S_{2k+1} : k \in \mathbb{Z}_{\geq 0})$ and $(S_{2k} : k \in \mathbb{Z}_{\geq 0})$). As the upper subsequence will be monotonically decreasing and the lower subsequence monotonically increasing we can employ a similar approach as in series sampling (see Section 2.6) and inversion sampling (see Section 2.1) to simulate an event of probability p.

In particular, we can simply draw a uniform random variable $u \sim \mathrm{U}[0, 1]$ and evaluate the upper and lower sequences until $u \notin (S_{2k}, S_{2k+1})$, at which point we know whether u lies above or below p. This approach is detailed in Algorithm 2.7.1 and an illustrative example is presented in Figure 2.7.1.

Algorithm 2.7.1 Retrospective Bernoulli Sampling [Beskos et al., 2008].

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 $u \leq S_{2k}$ then $u < p$ so return 1 else $u > p$ so return 0.

We draw attention to the fact that the indexing in (2.15) does not matter. In particular, if the index was increased by 1 then the limit of the alternating Cauchy sequence would still converge to p, albeit the even terms would converge from above and the odd terms from below. As such a suitably modified version of Algorithm 2.7.1 can be employed instead.

It is of interest to note that this approach can be straightforwardly extended to the case where p can be represented as the limit of some more general sequence than presented in (2.15), provided it is possible to find an alternating Cauchy sequence with which to extract upper and lower subsequences which monotonically converge to p. As will prove particularly useful later in this thesis, if it is possible to determine that the sequence *eventually* becomes an alternating Cauchy sequence, then Algorithm 2.7.1 can be directly employed. An example of such a situation is illustrated in Figure 2.7.2.

Retrospective Bernoulli sampling can also be employed to simulate unbiasedly an event of probability p, where p can be represented as a linear transformation of a number

of constituent alternating Cauchy sequences. More formally, in the case where $p := f(p_1, \ldots, p_m)$ for some linear function f and $p_1, \ldots, p_m$, each of which can be represented as the limit of alternating Cauchy sequences $(S_k^1, \ldots, S_k^m$ where $k \in \mathbb{Z}_{>0}$ respectively). In particular, this is possible by first noting that p can itself be represented as the limit of an alternating Cauchy sequence. Intuitively this is clear as we can simply align the index of each of the constituent Cauchy sequences (S_k^i where $i \in \{1, \ldots, m\}$) to ensure that the under and over estimations of p occur on alternating indices. Such an alignment can be achieved by increasing the index of some of the constituent alternating Cauchy sequences by 1. As such Algorithm 2.7.1 can be directly applied. We state the argument above more formally in the following corollary.

Corollary 2.7.1 (Linear 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 2.6.1. 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* $(T_k^{i,-} : k \in \mathbb{Z}_{\geq 0})$ *and* $(T_k^{i,+} : k \in \mathbb{Z}_{\geq 0})$ *as follows,*

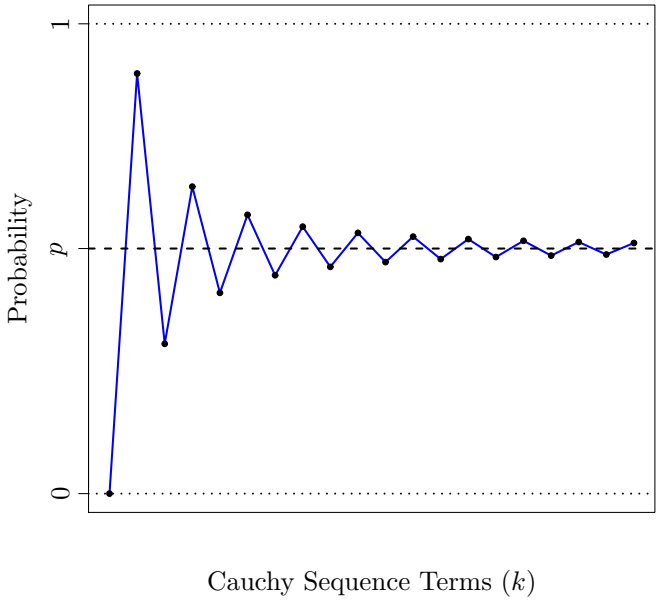
$$T_k^{i,-} = \begin{cases} S_k^{i,-} & \text{if } \mathrm{d}f/\mathrm{d}u_i > 0 \\ S_k^{i,+} & \text{if } \mathrm{d}f/\mathrm{d}u_i < 0 \end{cases}, \qquad T_k^{i,+} = \begin{cases} S_k^{i,+} & \text{if } \mathrm{d}f/\mathrm{d}u_i > 0 \\ S_{k+1}^{i,-} & \text{if } \mathrm{d}f/\mathrm{d}u_i < 0 \end{cases}. \tag{2.16}$$

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.*

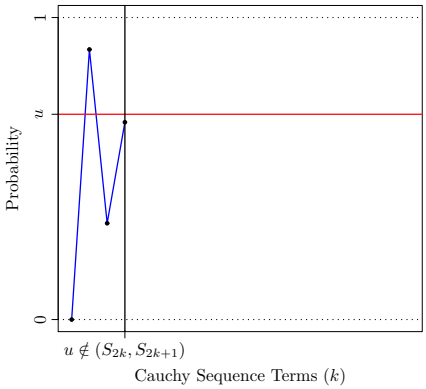
Clearly as we have that the number of computations required in implementing retrospective Bernoulli sampling is of stochastic length as a consequence of Algorithm 2.7.1 Step 2, then the efficiency of the algorithm will be dependent upon the expected number of iterations of that step which are required (where $u \sim \mathrm{U}[0, 1]$ as per Algorithm 2.7.1 Step 1),

$$\mathbb{E}[K] = \sum_{k=0}^{\infty} \mathbb{P}\left(K \geq k\right) = \sum_{k=0}^{\infty} \mathbb{P}\left(u \leq |S_{2k+1} - S_{2k}|\right) = \sum_{k=0}^{\infty} |S_{2k+1} - S_{2k}|\,. \tag{2.17}$$

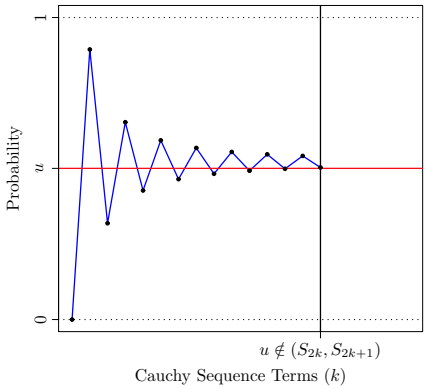
At a minimum for any practical implementation we require that the $\mathbb{E}[K] < \infty$, which can't be ensured without imposing further conditions. However, as we will encounter in Section 6, the alternating Cauchy sequences which we consider in this thesis converge exponentially fast and so finiteness is ensured.



(a) An illustration of the unknown probability p, overlaid with the graph composed of the estimate of p formed with the inclusion of the first k terms of the alternating Cauchy sequence.

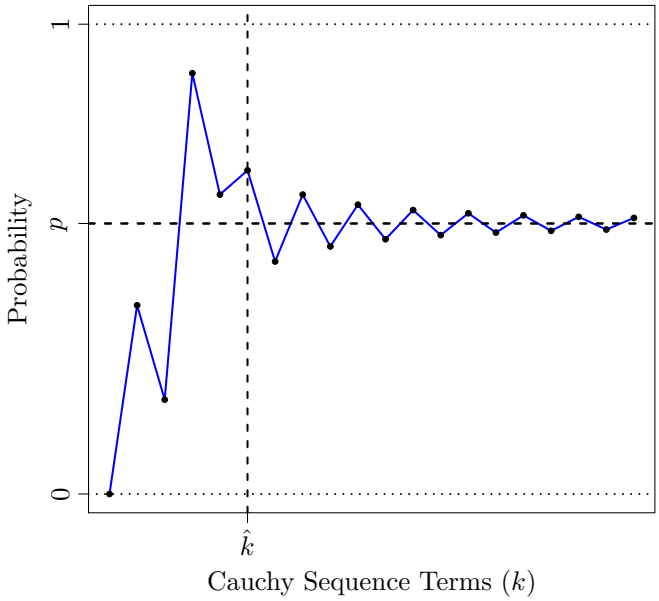


(b) Case i) $u > p$.

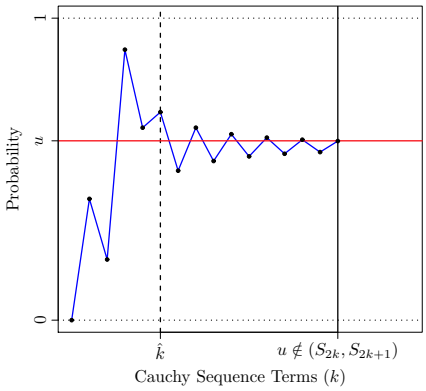


(c) Case ii) $u < p$.

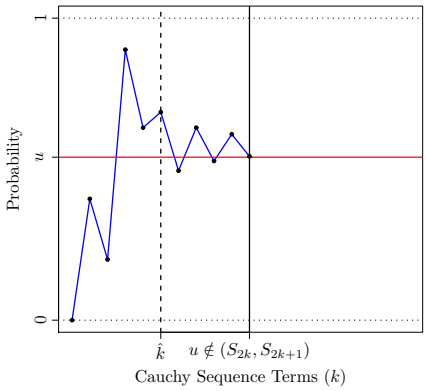
Figure 2.7.1: An illustration of the unbiased simulation of an event of unknown probability p, which can be represented as the limit of an alternating Cauchy sequence, by means of Retrospective Bernoulli Sampling.



(a) An illustration of the unknown probability p, overlaid with the graph composed of the estimate of p formed with the inclusion of the first k terms of an alternating sequence. Upon the inclusion of the first $\hat{k}$ terms the sequence becomes an alternating Cauchy sequence.



(b) Case i) $u > p$.



(c) Case ii) $u < p$.

Figure 2.7.2: An illustration of the unbiased simulation of an event of unknown probability p, which can be represented as the limit of a sequence, by means of Retrospective Bernoulli Sampling.

2.8 Simulating Brownian Motion and Related Processes

In this section we outline Monte Carlo methods for simulating (finite dimensional) sample path trajectories of Brownian motion, related stochastic processes and properties of these stochastic processes. To begin with we provide a brief introduction to this class of processes before moving on to discuss their simulation.

Brownian Motion (also known as a *Wiener process*), $W = \{W_t; t \geq 0\}$, is a continuous time stochastic process which forms the key building block for simulating sample path trajectories of the diffusion processes we study throughout this thesis (see Chapter 1). Many texts provide a detailed discussion of Brownian motion (including [Karatzas and Shreve, 1991, Chap. 2] and [Øksendal, 2007, Sec. 2.2], with [Kloeden and Platen, 1992, Sec. 1.8] providing an introduction with a focus on simulating (finite dimensional) sample path trajectories), however, we restrict our attention to the properties of Brownian motion which are explicitly used within this thesis.

We call a process (standard) Brownian motion if it satisfies the following properties,

Property 1 (Initial Value)**.** $W_0 = 0$.

Property 2 (Independent and Normally Distributed Increments)**.** *If* $r < r + s \leq t < t + s$ *then* $(W_{t+s} - W_t) \perp (W_{r+s} - W_r)$ *and* $W_{t+s} - W_t \overset{\mathcal{D}}{=} W_{r+s} - W_r \sim N(0, s)$.

Property 3 (Continuous Paths)**.** *With probability* 1*,* W_t *is a continuous function of* t.

Furthermore, Brownian motion satisfies a number of self-similarity properties. In particular, if W is Brownian motion, then so are the following,

Self-Similarity 1 (Scaling)**.** $B_t = \frac{1}{c} W_{c^2 t}$ *for some constant* $c > 0$.

Self-Similarity 2 (Symmetry)**.** $B_t = -W_t$.

Self-Similarity 3 (Increments)**.** $B_t = W_{t+s} - W_s$ *for fixed* s.

Self-Similarity 4 (Time Inversion)**.** $B_t = tW_{1/t}$ *(where* $B_0 := 0$*).*

Although Brownian motion sample paths are continuous, they are nowhere differentiable and furthermore are infinite dimensional random variables. As such, it isn't possible to simulate (and store) entire sample path trajectories (i.e. it is not possible to simulate $W \sim \mathbb{W}^x_{s,t}$, where we denote by $\mathbb{W}^x_{s,t}$ as Wiener measure – the law of Brownian motion

over the interval $[s, t]$ given $W_s = x$). However, as a direct consequence of Property 2, the transition density of Brownian motion is known (over any fixed finite interval) and so it is possible to simulate a Brownian motion sample path at any finite collection of time points $(q_1, \ldots, q_n)$ as per Algorithm 2.8.1.

Algorithm 2.8.1 Brownian Motion Simulation (at times $\{q_1, \ldots, q_n\}$).

1. For i in 1 to n, simulate $W_{q_i} \sim \mathrm{N}(W_{q_{i-1}}, q_i - q_{i-1})$.

It transpires that given a Brownian motion sample path at a finite collection of time points, simulated as per Algorithm 2.8.1, then the law of the process between any two consecutive points of the sample path is conditionally independent of the other simulated points (by the Markov property which follows from Property 2), and furthermore is known to be the law of a *Brownian Bridge*. A Brownian bridge is simply a Brownian Motion, W, which in addition to being conditioned to have a start point $(s, W_s = x)$ is also conditioned to have some end point $(t, W_t = y)$, the law of which we denote by $\mathbb{W}_{s,t}^{x,y}$. If we are interested in the density of the Brownian bridge at some intermediate time $q \in (s, t)$ then it can be shown to also be Gaussian by application of Property 2,

$$\begin{aligned} p(W_q = w | W_s = x, W_t = y) &\propto p(W_t = y | W_q = w, W_s = x) \cdot p(W_q = w | W_s = x) \\ &= p(W_t = y | W_q = w) \cdot p(W_q = w | W_s = x) \\ &\propto \exp\left\{ -\frac{1}{2} \frac{(y-w)^2}{(t-q)} \right\} \cdot \exp\left\{ -\frac{1}{2} \frac{(w-x)^2}{(q-s)} \right\} \\ &\propto \exp\left\{ -\frac{1}{2} \frac{w^2 - 2w\left[x + (q-s)(y-x)/(t-s)\right]}{(t-q)(q-s)/(t-s)} \right\} \\ &\propto \mathrm{N}\left(x + \frac{(q-s)(y-x)}{(t-s)}, \frac{(t-q)(q-s)}{(t-s)} \right) =: \mathrm{N}(\bar{w}, \sigma_w^2) \end{aligned} \quad (2.18)$$

This result is of fundamental importance throughout this thesis. In particular, although it is not possible to simulate and store an entire Brownian motion sample path trajectory, it can instead be characterised by its value at a finite collection of time points and known tractable transition densities between any two consecutive points. As detailed in Algorithm 2.8.2, the sample path can then be simulated at any other desired time points(s) (even if the sample path has already been partially simulated). As such, a Brownian motion sample path simulated as per Algorithms 2.8.1 and 2.8.2 forms the most elementary

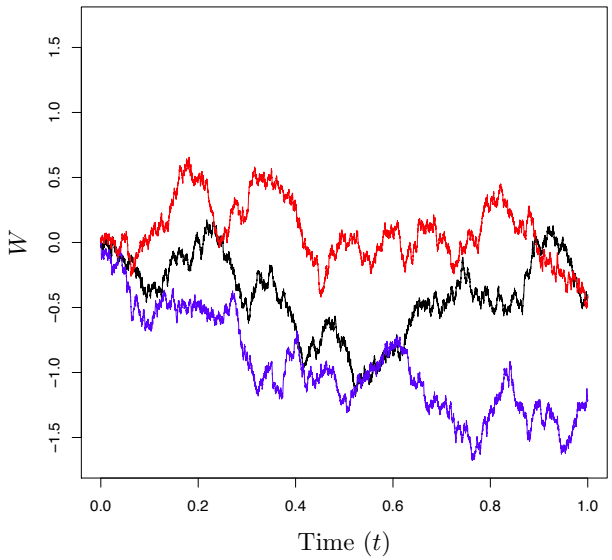
diffusion *skeleton* (see Definition 1 on page 4 of Chapter 1). In Figure 2.8.1 we present an illustration of Brownian motion and Brownian bridge sample path skeletons (simulated on a fine mesh).

Algorithm 2.8.2 Brownian Bridge Simulation (at times $\{q_1, \ldots, q_n\}$ given the process at times $\{s, p_1, \ldots, p_m, t\}$).

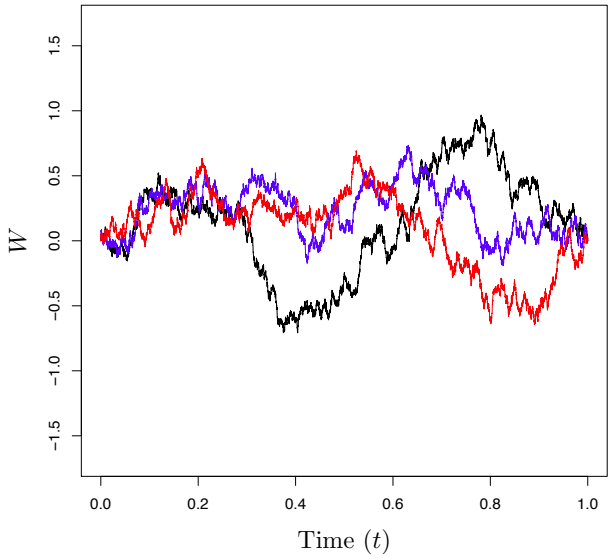
---

1. Set $\mathcal{S} := \left\{(s, X_s), (p_i, X_{p_i})_{i=1}^m, (t, X_t)\right\}$.
2. For i in 1 to n,
   (a) Set $l := \sup\{\mathcal{S} : \mathcal{S} \leq q_i\}$ and $r := \inf\{\mathcal{S} : \mathcal{S} \geq q_i\}$.
   (b) Simulate $W_{q_i} \sim \mathrm{N}\left(W_l + \frac{(q_i - l)(W_r - W_l)}{r - l}, \frac{(r - q_i)(q_i - l)}{(r - l)}\right)$.
   (c) Set $\mathcal{S} := \mathcal{S} \cup \{q_i\}$.

In the remainder of this section we introduce related stochastic processes which are used in this thesis. In Section 2.8.1 we detail how to simulate the minimum (or maximum) of a Brownian bridge sample path. Finally, in Section 2.8.2 we detail how to simulate an intermediate point of a Brownian bridge sample path conditional on the minimum (or maximum) that it attains (a so called *Bessel Bridge*).



(a) Brownian motion sample path trajectories, $W \sim \mathbb{W}_{0,1}^{0}$



(b) Brownian bridge sample path trajectories, $W \sim \mathbb{W}_{0,1}^{0,0}$

Figure 2.8.1: An illustration of Brownian motion and Brownian bridge sample path trajectories simulated on a fine mesh.

2.8.1 Brownian Bridge at its Minimum or Maximum Point

The joint distribution of the minimum value attained by a Brownian bridge sample path trajectory ($\hat{m} := \inf\{W_q; q \in [s,t]\}$ where $W \sim \mathbb{W}_{s,t}^{x,y}$), and the time at which it is attained ($\tau := \sup\{q \in [s,t] : W_q = \hat{m}\}$) is given in [Karatzas and Shreve, 1991, Chap 2.8 D & Chap. 4.3 C],

$$\mathbb{P}\left(\hat{m} \in \mathrm{d}w, \tau \in \mathrm{d}q \,|\, W_s = x, W_t = y\right) \propto \frac{(w-x)(w-y)}{\sqrt{(t-q)^3(q-s)^3}} \exp\left\{-\frac{(w-x)^2}{2(q-s)} - \frac{(w-y)^2}{2(t-q)}\right\} \mathrm{d}w\,\mathrm{d}q. \tag{2.19}$$

As shown in [Beskos et al., 2006a, Sec. 3.1] (and reformulated here), it is possible to draw jointly the minimum value and the time it occurs from (2.19). In particular simulating $u_1, u_2 \sim \mathrm{U}[0,1]$ and setting,

$$\hat{m} = x - \frac{1}{2}\left[\sqrt{(y-x)^2 - 2(t-s)\log(u_1)} - (y-x)\right], \tag{2.20}$$

denoting by $\mathrm{IGau}(u;\mu,\lambda) = \sqrt{\lambda/2\pi u^3}\exp\{-\lambda(u-\mu)^2/2\mu^2 u\}$ with $u > 0$ as the density of the inverse Gaussian distribution and setting,

$$V = \begin{cases} \xi_1, & \text{where } \xi_1 \sim \mathrm{IGau}\left(\dfrac{y-\hat{m}}{x-\hat{m}}, \dfrac{(y-\hat{m})^2}{t-s}\right), \quad \text{if } u_2 < \dfrac{x-\hat{m}}{x+y-2\hat{m}}, \\ \dfrac{1}{\xi_2}, & \text{where } \xi_2 \sim \mathrm{IGau}\left(\dfrac{x-\hat{m}}{y-\hat{m}}, \dfrac{(x-\hat{m})^2}{t-s}\right), \quad \text{if } u_2 \geq \dfrac{x-\hat{m}}{x+y-2\hat{m}}. \end{cases} \tag{2.21}$$

then $(\tau, \hat{m})$ is a sample from (2.19).

It transpires later in this thesis that it is necessary to simulate the sample path minimum conditional on being within a particular interval, $(\tau, \hat{m})|\,(\hat{m} \in [a_1, a_2])$ where $a_1 < a_2 \leq (x \wedge y)$. This can be achieved as suggested in Beskos et al. [2006a] by rejection sampling (see Section 2.4), in which sample path minima are simulated until one lies in the desired interval. Alternatively, we note that it is also possible (and computationally more efficient when the interval is small) to simulate $(\tau, \hat{m})$ by inversion sampling (see Section 2.1), by simply modifying how the uniform random variable u_1 is simulated as follows,

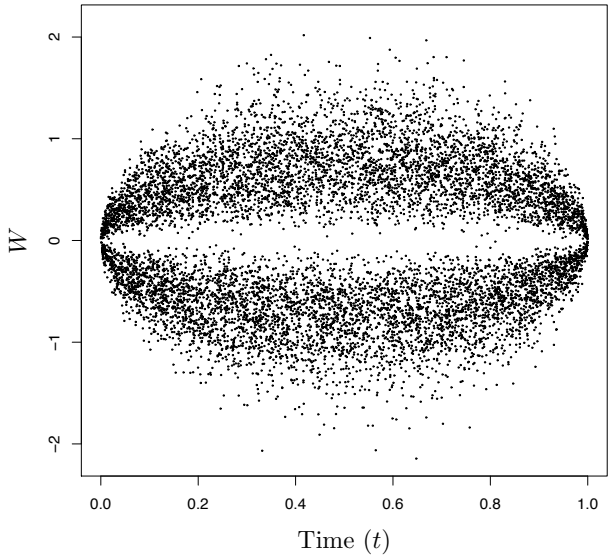
$$u_1 \sim \mathrm{U}\left[M(a_2), M(a_1)\right], \quad \text{where } M(a) := \exp\left\{-2(a-x)(a-y)/(t-s)\right\}. \tag{2.22}$$

The synthesis of the above argument is presented in Algorithm 2.8.3, along with an illustration of simulated minima (and maxima) points in Figure 2.8.2. Analogously the maximum point of a Brownian bridge sample path ($(\tau, \check{m})$ where $\check{m} := \sup\{W_q; q \in [s,t]\}$) can be simulated by a standard reflection argument (recalling Self-Similarity Property 2). In particular, if we consider a reflected Brownian bridge with reflected start and end points ($(s', W'_s = -x)$ and $(t', W'_t = -y)$ respectively) and simulate the minimum point of this reflected Brownian bridge as per Algorithm 2.8.3, then the reflection of the minimum point is the maximum point for the unreflected Brownian bridge ($\tau := \tau', \check{m} := -\hat{m}'$).

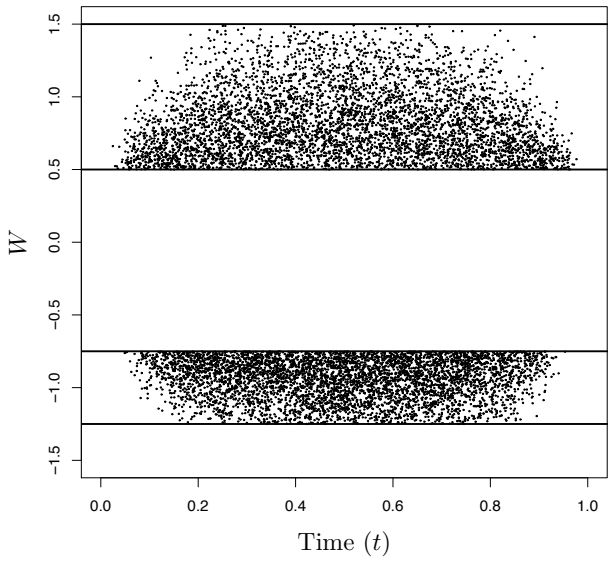
Algorithm 2.8.3 Brownian Bridge Simulation at its Minimum Point (constrained to the interval $[a_1, a_2]$ where $a_1 < a_2 \leq x \wedge y$ and conditional on $W_s = x$ and $W_t = y$ (denoting IGau(μ, λ) as the inverse Gaussian distribution with mean μ and shape parameter λ).

---

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



(a) Minimum or maximum without restriction.



(b) Minimum with restriction, $W \sim \mathbb{W}_{0,1}^{0,0} \big| (\hat{m} \in [-1.25, -0.75])$, or maximum with restriction, $W \sim \mathbb{W}_{0,1}^{0,0} \big| (\check{m} \in [0.5, 1.5])$.

Figure 2.8.2: An illustration of the minimum of 5,000 Brownian bridge sample path trajectories, and the maximum of 5,000 Brownian bridge sample path trajectories.

2.8.2 Bessel Bridge

Conditional on a Brownian bridge sample path minimum or maximum point ($(\tau, \hat{m})$ or $(\tau, \check{m})$, simulated as per Section 2.8.1 where $W \sim \mathbb{W}_{s,t}^{x,y}$), the law of the remainder of the trajectory is that of a *Bessel bridge*, which can be constructed by means of a 3-dimensional Brownian bridge of unit length conditioned to start and end at zero. A detailed proof is outlined in [Asmussen et al., 1995, Prop. 2] with the formulation that we present being that of [Beskos et al., 2006a, Thm. 2]. To guide intuition for the result presented in Beskos et al. [2006a], lets first consider the case where we are interested in simulating the sample path at some intermediate time $q \in (\tau, t)$.

Note that we are in effect simulating the value of a Brownian bridge path at time q, conditioned to start at the point $(\tau, \hat{m})$, end at the point (t, y) and remain above $\hat{m}$. By rescaling location and time (as per Self-Similarity Properties 1 and 3), we find that this is equivalent to simulating the value of a Brownian bridge sample path at time $q' := (q - \tau)/(t - \tau)$ with start point $(0, 0)$ and end point $(1, y' := y - \hat{m})$ conditioned to remain above 0, provided the simulated value is rescaled appropriately ($w := W_{q'} + \hat{m}$). Simulation is possible by first simulating 3 independent realisations of a Brownian bridge of unit length conditioned with start and end points of zero at time q', denoted $\{b_1, b_2, b_3\}$. Noting that we additionally require an offset so that the terminal value of the Bessel bridge is $y - \hat{m}$ we have,

$$w = \hat{m} + \sqrt{(t-\tau)\left[\frac{(y-\hat{m})(q-\tau)}{(t-\tau)(t-\tau)^{1/2}} + b_1\right]^2 + (t-\tau)b_2^2 + (t-\tau)b_3^2}. \tag{2.23}$$

Now in the case where we are interested in simulating the sample path at the intermediate time $q \in (s, \tau)$, then by reversing time and applying the same argument as above then,

$$w = \hat{m} + \sqrt{(\tau-s)\left[\frac{(x-\hat{m})(\tau-q)}{(\tau-s)(\tau-s)^{1/2}} + b_1\right]^2 + (\tau-s)b_2^2 + (\tau-s)b_3^2}. \tag{2.24}$$

The synthesis of the above argument is presented in Algorithm 2.8.4, along with an illustration of example Bessel bridge sample paths simulated on a fine mesh in Figure 2.8.3. If we are instead interested in simulating an intermediate point of a Brownian bridge sample path conditional on the maximum attained point $(\tau, W_\tau = \check{m})$, then this can be achieved by the standard reflection argument we detailed in Section 2.8.1.

Algorithm 2.8.4 (Minimum) Bessel Bridge Simulation (at time $q \in (s,t)$ given $W_s = x, W_t = y$ and $W_\tau = \hat{m}$) [Asmussen et al., 1995].

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

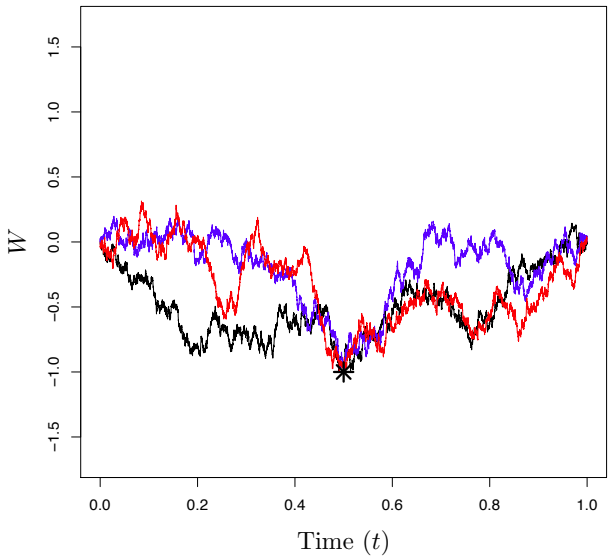


Figure 2.8.3: An illustration of Bessel bridge sample path trajectories, $W \sim \mathbb{W}_{0,1}^{0,0}\Big|(\tau, \hat{m})$ where $\tau = 0.5$ and $\hat{m} = -1$, simulated on a fine mesh.

2.9 Simulating Poisson Processes

In this section we outline Monte Carlo methods for simulating Poisson processes, beginning with a brief overview of this class of stochastic processes.

A *Poisson Process* is a continuous time stochastic process $\{N(t) : t \geq 0\}$, parameterised with rate function (or intensity) $\lambda(t)$, which counts the number of events which occur in the interval $[0, t]$ and satisfies the following properties:-

Property 1. $N(0) = 0$

Property 2 (Poisson Distributed Number of Events)**.** $N(t+s) - N(t) \sim Poi\left(\int_t^{t+s} \lambda(u)\,\mathrm{d}u\right)$. *In particular we have,* $\mathbb{P}(N(t+s) - N(t) = k) = \exp\left\{-\int_t^{t+s} \lambda(u)\,\mathrm{d}u\right\} \cdot \left(\int_t^{t+s} \lambda(u)\,\mathrm{d}u\right)^k / k!$

Property 3 (Independent Increments)**.** *If* $r < r + s \leq t < t + s$ *then* $[N(t + s) - N(t)] \perp [N(r + s) - N(r)]$.

A fuller account of Poisson processes can be found in a number of texts (see for instance [Cox and Isham, 1980; Kingman, 1992; Daley and Vere-Jones, 2003, 2008]). However, in this section we detail how to simulate Poisson process sample paths, an example of which is illustrated in Figure 2.9.1.

As suggested by Figure 2.9.1, in order to simulate a sample path it is sufficient to simulate the sample path event times and so we focus on this in this section. For ease of exposition we consider separately the simulation of sample paths of Poisson processes with constant intensity (a time homogeneous Poisson process) and non-constant intensity (a time inhomogeneous Poisson process) in Sections 2.9.1 and 2.9.2 respectively. Finally, in Section 2.9.3 we introduce and detail how to simulate sample paths from the related class of *Compound Poisson Processes*.

2.9.1 Time Homogeneous Poisson Processes

In this section we outline two alternate approaches for simulating sample paths of a time homogeneous Poisson process with constant intensity λ, each of which is used in different contexts throughout this thesis. As a consequence of Properties 2 and 3 of Poisson processes given in Section 2.9, a time homogeneous Poisson process has the following additional property, which is what we exploit in our two simulation approaches.

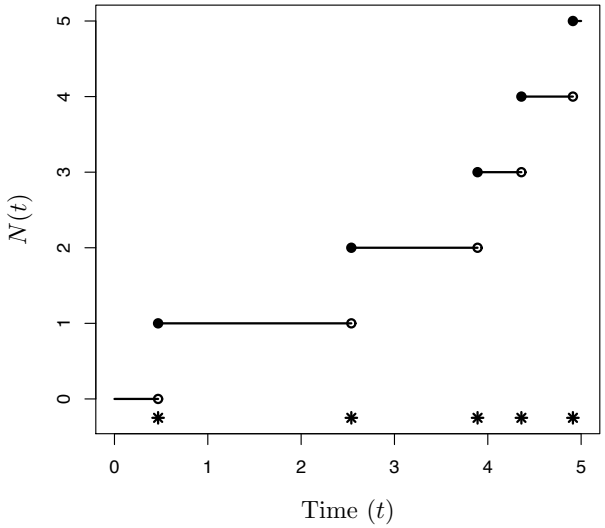


Figure 2.9.1: An illustration of a sample path of a time homogeneous Poisson process with intensity $\lambda = 1$ over the interval $[0, 5]$, where each asterisk indicates an event time.

Property 4 (Independent and Identically Distributed Increments)**.** *If* $r < r + s \leq t < t + s$ *then* $[N(t + s) - N(t)] \perp\!\!\!\perp [N(r + s) - N(r)] \perp\!\!\!\perp N(s) \sim Poi(\lambda s)$

To introduce our first approach note that, as a consequence of Property 4, for any given sample path we can directly simulate the number of events that occur in the interval $[0, t]$ (in particular, $N(t) \sim \text{Poi}(\lambda t)$), but not when they occur. However, it can be shown that conditional on the number of events that occur in $[0, t]$ that they are independently and uniformly distributed on the interval (see for instance [Kingman, 1992, Chap. 2.4]). To show this suppose we know that a total of n events occur ($N(t) = n$), and we are interested in how many of those n events occur in the subinterval $[0, r] \subseteq [0, t]$. Noting that at most $k \leq n$ events could occur in this subinterval and recalling that we have independent and

identically distributed increments (Property 4), then we have,

$$\begin{aligned}\mathbb{P}\left(N(r) = k \,|\, N(t) = n\right) &= \frac{\mathbb{P}\left(N(r) = k\right) \cdot \mathbb{P}\left(N(t) - N(r) = n - k\right)}{\mathbb{P}\left(N(t) = n\right)} \\ &= \frac{n!}{(n-k)!k!} \cdot \frac{\left[\exp\{-\lambda r\}\,(\lambda r)^k\right] \cdot \left[\exp\{-\lambda(t-r)\}\,(\lambda(t-r))^{n-k}\right]}{\exp\{-\lambda t\}\,(\lambda t)^n} \\ &= \frac{n!}{(n-k)!k!} \cdot \frac{r^k (t-r)^{n-k}}{t^n}, \end{aligned} \tag{2.25}$$

which is the probability that k out of n independent $\mathrm{U}[0,t]$ random variables fall in the interval $[0,r]$. As such, the event times of a time homogeneous Poisson process sample path (and hence the process itself) can be simulated by means of a two stage algorithm whereby we first simulate the number of events that occur in in $[0,t]$ and then uniformly scatter them, as detailed in Algorithm 2.9.1.

Algorithm 2.9.1 Time Homogeneous Poisson Process Simulation Algorithm (Conditional Uniform Dispersal Approach) [Kingman, 1992].

1. Simulate $n \sim \mathrm{Poi}(\lambda t)$.
2. If $n \neq 0$, simulate $u_1, \ldots, u_n \overset{\text{iid}}{\sim} \mathrm{U}[0,t]$.
3. If $n \neq 0$, set $q_1, \ldots, q_n$ to be the order statistics of the set $\{u_1, \ldots, u_n\}$.

Our second approach for simulating a time homogeneous Poisson process relies on the fact that for any given sample path the waiting time between successive events is exponentially distributed (see for instance [Kingman, 1992, Chap. 4.1]). To show this suppose the $(n-1)^{\text{th}}$ event occurred at time q_{n-1} and consider the probability that in an additional period of length s there has been no further events (denoting $T_1, \ldots, T_n$ as the interval time before each successive event),

$$\begin{aligned}&\mathbb{P}\left(T_n > s \,|\, T_1 = q_1, T_2 = q_2 - q_1, \ldots, T_{n-1} = q_{n-1} - q_{n-2}\right) \\ &\quad = \mathbb{P}\left(N(q_{n-1} + s) = n - 1 \,|\, N(t_{n-1}) = n - 1\right) \\ &\quad = \mathbb{P}\left(N(s) = 0\right) = \exp\{-\lambda s\}, \end{aligned} \tag{2.26}$$

which is precisely the probability that an $\mathrm{Exp}(\lambda)$ distributed random variable is greater than s. As a result, sample path event times $(q_1, \ldots, q_n)$ in the interval $[0,t]$ can be sim-

ulated by simply simulating successive Exp(λ) waiting times while $\sum_i T_i \leq t$, as detailed in Algorithm 2.9.2.

Algorithm 2.9.2 Time Homogeneous Poisson Process Simulation Algorithm (Exponential Waiting Time Approach) [Kingman, 1992].

1. Set $i = 0$ and $T_0 = 0$. While $\sum_i T_i \leq t$,
   (a) Set $i = i + 1$ and simulate $T_i \sim \text{Exp}(\lambda)$.
   (b) If $\sum_i T_i \leq t$ then set $q_i = \sum_i T_i$.
2. Set $n = i - 1$

2.9.2 Time Inhomogeneous Poisson Processes

It is possible to simulate sample paths of a time inhomogeneous Poisson process, with intensity $\lambda(t)$, by means of simulating a dominating time homogeneous Poisson process with constant intensity Λ (such that $\forall u \in [0, t]\ \lambda(u) \leq \Lambda$) and conducting *Poisson Thinning* (see for instance [Kingman, 1992, Chap. 5.1]). Poisson thinning can be thought of as a method similar to rejection sampling (as discussed in Section 2.4), whereby simulated events of a sample path from a dominating time homogeneous Poisson process are used as proposals, which are then accepted or rejected as events arising from the target time inhomogeneous Poisson process.

To justify this consider a time homogeneous Poisson process $\{N(t), t \geq 0\}$, with intensity Λ, such that each event that arises can be classified either as a 'Type 1' event (with probability p) or a 'Type 2' event (with probability $(1 - p)$). Now, if we consider the stochastic process counting the number of each type of event which occurs in the interval $[0, t]$ ($N_1(t)$ and $N_2(t)$ respectively), then by applying a partitioning argument we have,

$$\begin{aligned}
\mathbb{P}\left(N_1(t) = m, N_2(t) = n\right) &= \sum_{k=0}^{\infty} \mathbb{P}\left(N_1(t) = m, N_2(t) = n \,|\, N(t) = k\right) \cdot \mathbb{P}\left(N(t) = k\right) \\
&= \mathbb{P}\left(N_1(t) = m, N_2(t) = n \,|\, N(t) = m + n\right) \cdot \mathbb{P}\left(N(t) = m + n\right) \\
&= \left(\frac{(m+n)!}{m!n!} p^m (1-p)^n\right) \cdot \frac{\exp\{-\Lambda t\}(\Lambda t)^{m+n}}{(m+n)!} \\
&= \frac{\exp\{-\Lambda p t\}\,(\Lambda p t)^m}{m!} \cdot \frac{\exp\{-\Lambda(1-p)t\}\,(\Lambda(1-p)t)^n}{n!}, \quad (2.27)
\end{aligned}$$

and so we can conclude that $\{N_1(t), t \geq 0\}$ is a Poisson process with intensity Λp, $\{N_2(t), t \geq 0\}$ is a Poisson process with intensity $\Lambda(1-p)$ and they are independent.

We can alternatively view the time homogeneous Poisson process with intensity Λ as arising from the superposition of a target time inhomogeneous Poisson process with intensity $\lambda(t)$ and another with intensity $(\Lambda - \lambda(t))$. Any event arising (say at time q) can be assigned to the target time inhomogeneous Poisson process with probability $\lambda(q)/\Lambda$. As such the target time inhomogeneous Poisson process can be simulated by first simulating a sample path from the time homogeneous Poisson process (using either Algorithm 2.9.1 or Algorithm 2.9.2) and then conducting Poisson thinning, as illustrated in Figure 2.9.2 and detailed in Algorithms 2.9.3 and 2.9.4. The reason for the two alternate algorithms will become apparent later in this thesis. Clearly, Algorithm 2.9.3 is a more natural implementation, however it transpires that there is computational advantage in simulating each event separately which is what leads to Algorithm 2.9.4.

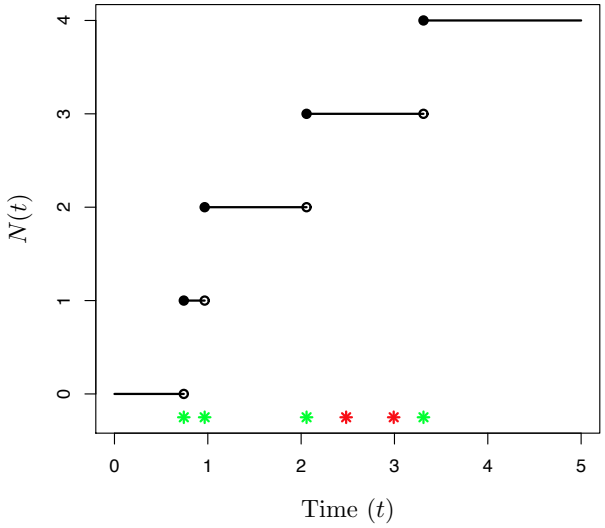


Figure 2.9.2: An illustration of a sample path of a time inhomogeneous Poisson process with intensity $\lambda(t) = |\sin(t)|$ over the interval $[0, 5]$. Each asterisk indicates an event simulated under the dominating time homogeneous Poisson process, with those in green denoting the accepted events arising under the target Poisson process.

Algorithm 2.9.3 Time Inhomogeneous Poisson Process Simulation Algorithm [Kingman, 1992].

1. Simulate proposal event times $(p_1, \ldots, p_n)$ of a time homogeneous Poisson process with intensity Λ, as per Algorithm 2.9.1 or Algorithm 2.9.2.
2. If $n \neq 0$, then set $j = 0$ and for i in 1 to n,
   (a) With probability $\lambda(p_i)/\Lambda$ set $j = j + 1$ and $q_j = p_i$.

Algorithm 2.9.4 Time Inhomogeneous Poisson Process Simulation Algorithm (Exponential Waiting Time Approach) [Kingman, 1992].

1. Set $i = 0$, $j = 0$ and $T_0 = 0$. While $\sum_i T_i \leq t$,
   (a) Find $\Lambda_i \geq \sup_{u \in [\sum_i T_i, t]} \lambda(u)$, set $i = i + 1$ and simulate $T_i \sim \text{Exp}(\Lambda_i)$.
   (b) If $\sum_i T_i \leq t$ then with probability $\lambda(\sum_i T_i)/\Lambda_i$ set $j = j + 1$ and $q_j = \sum_i T_i$.

2.9.3 Compound Poisson Processes

A *Compound Poisson Process*, $\{J(t) : t \geq 0\}$, parameterised with intensity $\lambda(t)$ and jump size function $\nu(t)$, is a continuous time stochastic process which sums in the interval $[0, t]$ a collection of $f_\nu(t)$ distributed random variables (or jumps), the number of which are Poisson distributed with intensity $\lambda(t)$. Typically each of these parameters are themselves parameterised by some other stochastic process, for instance the compound Poisson process itself (however to ease notation we omit this at this stage). In particular, we have,

$$J(t) := \sum_{i=1}^{N(t)} \nu(q_i), \qquad \text{where } N(t) \sim \text{Poi}\left(\int_0^t \lambda(u)\,\mathrm{d}u\right) \text{ and } \nu(q_i) \sim f_\nu(q_i). \tag{2.28}$$

In the case where the parameters of the compound Poisson process are not themselves parameterised by the process then simulating a sample path is relatively straight-forward. In particular, as detailed in Algorithm 2.9.5, we can first simulate over the interval $[0, t]$ the jump times (Poisson events) and then simulate and sum the f_ν distributed random variables of the sample path. If instead the compound Poisson process is self-parameterised, then simulating each jump time and then the jump size iteratively is advantageous as for each subinterval a dominating time homogeneous Poisson process with lower intensity can be found which, in analogous fashion to finding a tighter dominating proposal den-

sity in rejection sampling (see Section 2.4), will be more computationally efficient. This alternate algorithm is presented in Algorithm 2.9.6, with illustrative examples of compound Poisson process sample paths given in Figure 2.9.3.

Algorithm 2.9.5 Time Inhomogeneous Compound Poisson Process Simulation Algorithm [Kingman, 1992].

1. Set $J(0) = 0$.
2. Simulate jump times $(q_1, \ldots, q_n)$ as per Algorithm 2.9.3.
3. If $n \neq 0$, for i in 1 to n simulate $\nu(q_i) \sim f_\nu(q_i)$ and set $J(q_i) = J(q_{i-1}) + \nu(q_i)$.

Algorithm 2.9.6 Self-Parameterised Time Inhomogeneous Compound Poisson Process Simulation Algorithm [Kingman, 1992].

1. Set $J(0) = 0$, $i = 0$, $j = 0$ and $T_0 = 0$. While $\sum_i T_i \leq t$,
   (a) Find $\Lambda_i \geq \sup_{u \in [\sum_i T_i, t]} \lambda(u)$, set $i = i + 1$ and simulate $T_i \sim \text{Exp}(\Lambda_i)$.
   (b) If $\sum_i T_i \leq t$, then with probability $\lambda\left(\sum_i T_i\right)/\Lambda_i$ set $j = j + 1$, $q_j = \sum_i T_i$, simulate $\nu(q_j) \sim f_\nu(q_i)$ and set $J(q_j) = J(q_{j-1}) + \nu(q_j)$.

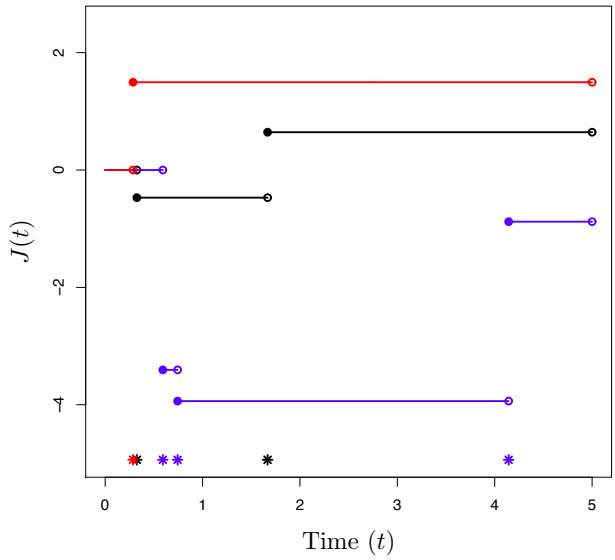


Figure 2.9.3: An illustration of sample paths of a compound Poisson process with intensity $\lambda(t) = |\cos(J(t))|$ and jump sizes $\mu(t) \sim \mathrm{N}(-J(t-)/2, \pi)$, over the interval $[0, 5]$. The colour of each asterisk corresponds to the jump times of the similarly coloured compound Poisson process sample path.

3 Sequential Monte Carlo Methods

"Study the past, if you would divine the future."

— Confucius

As discussed in Chapter 1, in this thesis we develop methodology for filtering jump diffusions observed discretely with error. Underpinning this methodology is existing literature on the application of *Sequential Monte Carlo methods* (SMC) to *Hidden Markov Models* (HMMs). In this chapter we provide a brief overview of this background material.

To motivate this class of Monte Carlo methods we begin by introducing HMMs in Section 3.1. HMMs are the natural and flexible framework under which the jump diffusions we consider in thesis are observed, whereby at discrete points in time we observe some underlying evolving process of interest with error. HMMs are particularly appealing due to their suitability in tackling *online* problems (whereby sequential information must be processed on arrival without loss of computational efficiency) which is a consequence of the recursive nature in which various inferential problems can be represented. In Section 3.1.1 we draw particular attention to the *filtering* problem that we tackle in this thesis (in which we use all observations to any point in time to make a probabilistic interpretation of the state of the process at that point in time), highlighting in Section 3.1.4 the situations in which solutions can be found. Unfortunately, for our purposes in this thesis we require methodology for problems in which analytic solutions can't be found, which motivates our use of Monte Carlo methodology.

SMC methods are a class of Monte Carlo methods in which in order to draw inference on some quantity of interest we would require the simulation of a distribution which is difficult to simulate directly, but can be simulated by means of constructing a sequence

of intermediate distributions (in which for each distribution simulation is comparatively simple). The key idea is that by sampling some initial distribution (either directly or using one of the Monte Carlo methods in Section 2), these samples (or *particles*) can be used as proposals in an importance sampler (see Section 2.5) for the next distribution in the sequence. This approach is then applied iteratively until we reach the target distribution, at which point the importance weighted particles (which jointly form an empirical representation of the target density) can be used to draw inference on the problem of interest.

The application of SMC methodology to tackle the filtering problem in HMMs we use in this thesis, which is also known as *Particle Filtering* methodology, has gained traction in recent years as the inherent time structure in HMMs is particularly well suited to SMC. In the most simplistic setting time itself is used to sequence the intermediate distributions, with particles being propagated between observation times using the dynamics of the underlying process and importance weighted according to the information obtained from the noisy observations.

In the second half of this chapter we review particle filtering methodology pertinent for our needs. In Section 3.2 we introduce *Sequential Importance Sampling* (SIS). As with importance sampling, care has to be taken in SIS in choosing an appropriate dominating proposal density (*importance function*), which is a topic we discuss in detail in Section 3.3. In Section 3.4 we draw attention to the problem of importance weight degeneracy that occurs in SIS over long sequences of intermediate distributions (in particular, the resulting empirical representation of the target density is reliant on a small number of particles of large weight if SIS is used). To remedy this degeneracy we outline the standard *resampling* approaches taken and the alternate *Sequential Importance Sampling / Resampling* (SISR) algorithm that can be taken instead of SIS in Sections 3.4 and 3.5. Finally, in Section 3.6 we motivate and introduce the *Auxiliary Particle Filter* (APF).

The development of particle filtering methodology is rather disjointed as it was driven primarily by applications in a broad range of areas. Within this chapter we consider existing literature within the context of importance sampling (see Section 2.5) as in this manner it is clear to see that they are variations on a single idea. It is worth noting that a number of more detailed tutorials and texts exist (such as Doucet et al. [2000], Doucet and Johansen [2011], Maskell and Gordon [2001] and Doucet et al. [2001]).

3.1 Hidden Markov Models

Hidden Markov models (HMMs) are classes of models which have broad applicability in a number of diverse areas (see Cappé et al. [2005] for a general overview), such as genome analysis (Krogh et al. [2001]), robotics (Hovland and McCarragher [1998]), neural networks (de Freitas et al. [2000]) and financial modelling (Mamon and Elliot [2007]). Advancement of this area has been rapid with, for example, consideration to problems of the control of linear systems in the 1980s (see for instance Davis and Vinter [1985] and Hannan and Deistler [1988]) and sequential Monte Carlo in the 1990s (see Doucet et al. [2001]).

HMMs assume that there is some *latent* (hidden or unobserved) underlying $\mathcal{X}^{\mathbb{N}}$-valued generating process $\{X_t\}_{t\geq 0}$ of interest known to satisfy the Markov property. However, the latent process is not observed directly and instead is observed with error through a $\mathcal{Y}^{\mathbb{N}}$-valued *observation process* $\{Y_t\}_{t\geq 1}$. In general we want to conduct inference on the latent process by using information from the observation process.

Throughout this section we will assume we are considering HMMs with $\mathbb{R}^p$-valued latent processes and $\mathbb{R}^q$-valued observation processes and will further assume the following *State Space Dynamics* (SSDs) hold for some static parameter $\theta \in \Theta$,

Property 1 (Initial Latent Process SSD)**.** *The latent process has a known initial density. In particular we have,* $X_0 \sim \mu_\theta(\cdot)$.

Property 2 (Latent Process SSD)**.** *The latent process* $\{X_t\}_{t\geq 0}$ *is a first order*[1] *Markov chain which, conditional on the prior latent state, has a known transition density. In particular we have,* $X_t|(X_{t-1} = x_{t-1}, \ldots, X_0 = x_0) = X_t|(X_{t-1} = x_{t-1}) \sim f_\theta(\cdot|x_{t-1})$.

Property 3 (Observation Process SSD)**.** *The observation process* $\{Y_t\}_{t\geq 1}$ *is conditionally independent given the latent process, with known density. In particular we have,* $Y_t|(X_t = x_{t-1}, Y_{t-1} = y_{t-1}, \ldots, Y_0 = y_0) = Y_t|(X_t = x_{t-1}) \sim g_\theta(\cdot|x_{t-1})$.

The relationship between the latent and observation process in a HMM is illustrated as a directed acyclic graph in Figure 3.1.1.

For ease of exposition, in the remainder of this chapter we introduce some simplifying notation. We denote by $x_{a:b} := (x_a, x_{a+1}, \ldots, x_{b-1}, x_b)$ (where $0 \leq a \leq b$) and

[1] *n-th order Markov property:* The process is dependent on the current state and the $n-1$ prior states.

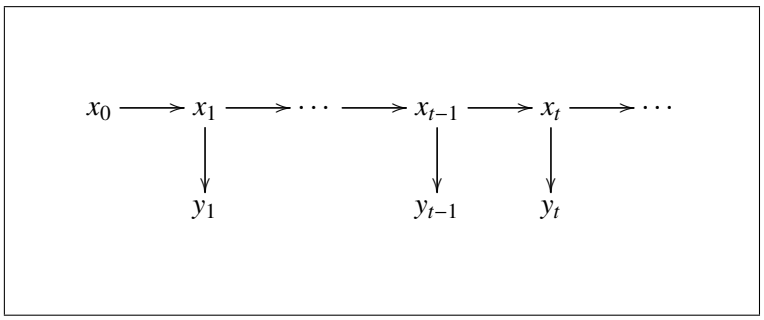


Figure 3.1.1: Directed acyclic graph representing the latent and observation processes of a Hidden Markov Model.

$p_\theta(\cdot)$ as a density. We additionally denote from the known SSDs that $\mu_\theta(x_0) := p_\theta(x_0)$, $f_\theta(x_t|x_{t-1}) := p_\theta(x_t|x_{t-1})\ \forall t$ and $g_\theta(y_t|x_t) := p_\theta(y_t|x_t)\ \forall t$. We reserve $\pi_\theta(\cdot)$ to denote a (context specific) target density or a primitive form of the target density. Furthermore we assume all densities are parameterised by θ[2].

The Markov property of the latent process (Property 2) is particularly useful as it allows the joint density of a sequence of latent states to be represented as the product of states conditional on a finite number of prior states. In particular, by application of Bayes rule we have,

$$\begin{aligned}\pi_\theta(x_{0:t}) &= p_\theta(x_t|x_{0:t-1}) \cdot p_\theta(x_{t-1}|x_{0:t-2}) \cdot \ldots \cdot p_\theta(x_0) \\ &= \mu_\theta(x_0) \cdot \prod_{i=1}^{t} f_\theta(x_i|x_{i-1}). \end{aligned} \tag{3.1}$$

The manner in which the latent process of a HMM forms this recursive decomposition is what makes it particularly appealing for tackling online problems. In particular, it is possible to extend inference to include further observations by reusing existing inference, so avoiding increased computational cost with increasing numbers of observations. For instance, considering the joint density of the latent and observation process to the time t ($X_{0:t}$ and $Y_{1:t}$ respectively) then by applying Bayes rule, conditional probability arguments

[2]For instance, we more formally have, $\int_{\theta\in\Theta} p_\theta(x_{0:t}|y_{1:t}) \cdot \lambda(\mathrm{d}x_{0:t}) \cdot \mu(\mathrm{d}\theta) = \mathbb{P}(x_{0:t} \in \mathrm{d}x_{0:t}|y_{1:t}, \theta \in \Theta)$.

and the Markov property this can be represented by the following recursion,

$$\begin{aligned}
\pi_\theta(x_{0:t}, y_{1:t}) &= p_\theta(y_{1:t}|x_{0:t}) \cdot p_\theta(x_{0:t}) \\
&= p_\theta(y_t|x_{0:t}) \cdot p_\theta(y_{1:t-1}|y_t, x_{0:t}) \cdot p_\theta(x_t|x_{0:t-1}) \cdot p_\theta(x_{0:t-1}) \\
&= g_\theta(y_t|x_t) \cdot p_\theta(y_{1:t-1}|x_{0:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot p_\theta(x_{0:t-1}) \\
&= \pi_\theta(x_{0:t-1}, y_{1:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t) = \ldots \qquad (3.2) \\
&= \mu_\theta(x_0) \cdot \prod_{k=1}^{t} \left[f_\theta(x_k|x_{k-1}) \cdot g_\theta(y_k|x_k) \right], \qquad (3.3)
\end{aligned}$$

and so at each time point the joint density can be simply updated by the application of the latent process transition density convoluted with the likelihood given by the observation as suggested by (3.2).

Typical inferential problems for HMMs include *filtering* (where we use the entire observation process to any point in time to make a probabilistic interpretation of the state of the latent process at that point in time), *prediction* (where we use the observation process to the current point in time to make probabilistic interpretations of the latent process at future time points) and *smoothing* (where we make improved probabilistic interpretations of the latent process at some time point given the observation process before and after that time point). Within a HMM framework the filtering, prediction and smoothing problems can all be represented in a recursive manner which we present in Sections 3.1.1, 3.1.2 and 3.1.3 respectively. Finally, in Section 3.1.4 we outline situations in which analytic solutions to these recursive representations can be found (in particular the filtering problem), highlighting the difficulty in finding solutions in more generalised situations.

3.1.1 The Filtering Problem

The *filtering* problem is that of making a probabilistic interpretation of the state of the latent process at a point in time, using the information obtained from the observation process up to that point in time. We will consider two variants of this problem: the *joint* filtering problem, in which we consider the latent process at all times up to that point in time; and the marginal filtering problem, in which we consider the latent process solely at that point in time. First considering the joint filtering problem then if we again apply

Bayes rule and recall (3.2) we have,

$$\begin{aligned}\pi_\theta(x_{0:t}|y_{1:t}) &= \frac{p_\theta(x_{0:t-1}, y_{1:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1}) \cdot p_\theta(y_{1:t-1})} \\ &= \pi_\theta(x_{0:t-1}|y_{1:t-1}) \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1})}, \end{aligned} \tag{3.4}$$

which can be easily interpreted as the normalised recursion of the solution to the joint filtering problem at the prior time point, with the known latent and observation process densities (see HMM Properties 2 and 3). Clearly, we can again apply this decomposition to the joint filtering problem at the prior time point iteratively backwards until such point that (3.4) is expressed in terms of the known initial latent process density (see Property 1).

Similarly for the marginal filtering problem, by marginalisation and the same arguments as above we have,

$$\pi_\theta(x_t|y_{1:t}) = \int p_\theta(x_{t-1:t}|y_{1:t}) \, \mathrm{d}x_{t-1} \tag{3.5}$$

$$\begin{aligned} &= \int p_\theta(x_{t-1}|y_{1:t}) \cdot p_\theta(x_t|x_{t-1}, y_{1:t}) \, \mathrm{d}x_{t-1} \\ &= \int \left[\frac{\pi_\theta(x_{t-1}|y_{1:t-1}) \cdot p_\theta(y_t|x_{t-1})}{p_\theta(y_t|y_{1:t-1})}\right] \cdot \left[\frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{p_\theta(y_t|x_{t-1})}\right] \mathrm{d}x_{t-1} \\ &= \int \pi_\theta(x_{t-1}|y_{1:t-1}) \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1})} \, \mathrm{d}x_{t-1} \end{aligned} \tag{3.6}$$

which has a similar interpretation to the joint filtering problem in (3.4).

3.1.2 The Prediction Problem

The *prediction* problem is that of making probabilistic interpretations of the latent process at future time points given information about the observation process up to the current time point. Supposing we are interested in the latent state in p time-steps (given at time t

the state of the latent process is $X_t = x_t$), then by the same arguments as before we have,

$$\begin{aligned}
\pi_\theta(x_{t+p}|y_{1:t}) &= \int p_\theta(x_{t:t+p}|y_{1:t})\,\mathrm{d}x_{t:t+p-1} \\
&= \int \pi_\theta(x_t|y_{1:t}) \cdot p_\theta(x_{t+1:t+p}|x_t, y_{1:t})\,\mathrm{d}x_{t:t+p-1} \\
&= \int \pi_\theta(x_t|y_{1:t}) \cdot p_\theta(x_{t+1:t+p}|x_t)\,\mathrm{d}x_{t:t+p-1} = \ldots \\
&= \int \pi_\theta(x_t|y_{1:t}) \cdot \prod_{k=1}^{p} f_\theta(x_{t+k}|x_{t+k-1})\,\mathrm{d}x_{t:t+p-1}. \qquad (3.7)
\end{aligned}$$

This recursive representation is intuitively appealing as it can be thought of simply as the current filtered distribution propagated p time-steps into the future using the known latent process transition density (see Property 2).

3.1.3 The Smoothing Problem

Finally, the *smoothing* problem is that of using all information from the observed process currently available to make revised estimates of the marginal filtering problem for states of the latent process at some prior time. The intuition is that we would expect to achieve better estimates of the latent state than in the filtering problem as more (future) information is available. In particular, we are interested in the marginal distribution of the latent process X_k (for some $k \in [0, t)$) conditional on $Y_{1:t}$, which by applying the same arguments as before we can find as follows (where we slightly abuse our notation and denote by π_θ^m the solution to the marginal filtering problem from Section 3.1.1),

$$\begin{aligned}
\pi_\theta(x_k|y_{1:t}) &= \int p_\theta(x_{k:k+1}|y_{1:t})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot p_\theta(x_k|x_{k+1}, y_{1:t})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot \frac{p_\theta(x_k|x_{k+1}, y_{1:k}) \cdot p_\theta(y_{k+1:t}|x_{k:k+1}, y_{1:k})}{p_\theta(y_{k+1:t}|x_{k+1}, y_{1:k})}\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot p_\theta(x_k|x_{k+1}, y_{1:k})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot \frac{p_\theta(x_k|y_{1:k}) \cdot p_\theta(x_{k+1}|x_k, y_{1:k})}{p_\theta(x_{k+1}|y_{1:k})}\,\mathrm{d}x_{k+1} \\
&= \pi_\theta^m(x_k|y_{1:k}) \cdot \int \frac{\pi_\theta(x_{k+1}|y_{1:t}) \cdot f_\theta(x_{k+1}|x_k)}{\int \pi_\theta^m(x_k|y_{1:k}) \cdot f_\theta(x_{k+1}|x_k)\,\mathrm{d}x_k}\,\mathrm{d}x_{k+1}. \qquad (3.8)
\end{aligned}$$

Again the solution is intuitively appealing as it is simply the initial estimate of the filtering problem with a backwards recursive modification representing currently observed information (the terms of which are known).

3.1.4 Implementational Problems and the Kalman Filter

Although the properties of HMMs outlined in Section 3.1 provide explicit recursive representations for various inferential problems (as shown in Section 3.1.1 – Section 3.1.3), explicit calculation relies on the tractability of the constituent densities. When dealing with high dimensional or highly non-linear set-ups these issues are compounded.

In the specific case of the joint and marginal filtering problems (see (3.4) and (3.5)) the lack of tractability of the normalising constant $p_\theta(y_{1:t})$, is what causes particular problems. A number of instances where direct and explicit calculation of the filtering problem is possible exist (namely those illustrated by Vidoni [1999]), however, by far the most studied is the formulation of the SSDs given within the context of the *Kalman filter* which was developed in Kálmán [1960] and Kálmán and Bucy [1961]. In this instance the SSDs are are all Gaussian, which allows the filtering problem to be solved without the need to find the normalising constant (although the normalising constant can be found as a by-product).

A brief illustration of the Kalman filter is considered as it neatly illustrates the use of HMMs and is useful later when considering both the *optimal marginal importance function* (see Section 3.3.1) and the *Auxiliary Particle Filter* (see Section 3.6). A fuller account of the Kalman filter can be found in a number of texts, for instance, Meinhold and Singpurwalla [1983].

To introduce the Kalman filter, suppose the following Gaussian SSDs hold $\forall t \geq 1$ (denoting matrices by upper case emboldened letters and MVN as the multivariate Gaussian density),

$$X_0 \sim \text{MVN}(m_0, \mathbf{C}_0), \tag{3.9}$$

$$X_t | (X_{t-1} = x_{t-1}) \sim \text{MVN}(\mathbf{A}x_{t-1}, \mathbf{V}_t), \tag{3.10}$$

$$Y_t | (X_t = x_t) \sim \text{MVN}(\mathbf{B}x_t, \mathbf{W}_t), \tag{3.11}$$

and further suppose we are interested in the marginal filtering problem of (3.5), noting that it can be re-expressed as follows,

$$\pi_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t) \cdot p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})}. \tag{3.12}$$

Following an induction argument it can be shown that,

$$X_t|\left(Y_{1:t} = y_{1:t}\right) \sim \text{MVN}\left(m_t, \mathbf{C}_t\right). \tag{3.13}$$

Trivially, for $t = 0$ we have $X_0 \sim \text{MVN}(m_0, \mathbf{C}_0)$ which agrees with (3.13). Assuming the solution to the marginal filtering problem at time $t-1$ is,

$$X_{t-1}|(Y_{1:t-1} = y_{1:t-1}) \sim \text{MVN}(m_{t-1}, \mathbf{C}_{t-1}), \tag{3.14}$$

we will show that (3.13) holds by computing analytically the density of (3.12) and showing it is Gaussian.

First considering the predictive density $p_\theta(x_t|y_{1:t-1})$ in (3.12), it can be shown that it is also Gaussian. In particular, applying a marginalisation argument, the latent process transition density (3.10) and the density of the marginal filtering problem at time $t-1$ (3.14), we have,

$$\begin{aligned}
p_\theta(x_t|y_{1:t-1}) &= \int p_\theta(x_{t-1:t}|y_{1:t-1})\,\mathrm{d}x_{t-1} = \int f_\theta(x_t|x_{t-1}) \cdot \pi_\theta(x_{t-1}|y_{1:t-1})\,\mathrm{d}x_{t-1} \\
&\propto \int \exp\left\{-\frac{1}{2}\left[(x_t - \mathbf{A}x_{t-1})^T \mathbf{V}_t^{-1} (x_t - \mathbf{A}x_{t-1}) + (x_{t-1} - m_{t-1})^T \mathbf{C}_{t-1}^{-1} (x_{t-1} - m_{t-1})\right]\right\} \mathrm{d}x_{t-1} \\
&= \int \exp\left\{-\frac{1}{2}\left[x_{t-1}^T \left(\mathbf{A}^T \mathbf{V}_t^{-1} \mathbf{A} + \mathbf{C}_{t-1}^{-1}\right) x_{t-1} - 2x_{t-1}^T \left(\mathbf{A}^T \mathbf{V}_t^{-1} x_t + \mathbf{C}_{t-1}^{-1} m_{t-1}\right) + x_t^T \mathbf{V}_t^{-1} x_t\right]\right\} \mathrm{d}x_{t-1},
\end{aligned}$$

denoting by $\mathbf{Q}_t^{-1} := \mathbf{A}^T \mathbf{V}_t^{-1} \mathbf{A} + \mathbf{C}_{t-1}^{-1}$ and $n_t := \mathbf{Q}_t(\mathbf{A}^T \mathbf{V}_t^{-1} x_t + \mathbf{C}_{t-1}^{-1} m_{t-1})$, then we have,

$$\begin{aligned}
p_\theta(x_t|y_{1:t-1}) &\propto \exp\left\{-\frac{1}{2}\left[x_t^T \mathbf{V}_t^{-1} x_t - n_t^T \mathbf{Q}_t^{-1} n_t\right]\right\} \underbrace{\int \exp\left\{-\frac{1}{2}(x_{t-1} - n_t)^T \mathbf{Q}_t^{-1} (x_{t-1} - n_t)\right\} \mathrm{d}x_{t-1}}_{\propto 1} \\
&\propto \exp\left\{-\frac{1}{2}\left[x_t^T \left(\mathbf{V}_t^{-1} - \mathbf{V}_t^{-1} \mathbf{A} \mathbf{Q}_t \mathbf{A}^T \mathbf{V}_t^{-1}\right) x_t - 2x_t^T \left(\mathbf{V}_t^{-1} \mathbf{A} \mathbf{Q}_t \mathbf{C}_{t-1}^{-1} m_{t-1}\right)\right]\right\},
\end{aligned}$$

denoting by $\mathbf{P}_t^{-1} := \mathbf{V}_t^{-1} - \mathbf{V}_t^{-1}\mathbf{A}\mathbf{Q}_t\mathbf{A}^T\mathbf{V}_t^{-1}$ and applying the Woodbury Matrix Identity (see Woodbury [1950]) we have $\mathbf{P}_t = \mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T$. Now, noting that $\mathbf{V}_t^{-1}\mathbf{A}\mathbf{Q}_t = (\mathbf{V}_t^{-1} - \mathbf{P}_t^{-1})\mathbf{V}_t(\mathbf{A}^T)^{-1}$ then we have directly that $\mathbf{P}_t(\mathbf{V}_t^{-1}\mathbf{A}\mathbf{Q}_t\mathbf{C}_{t-1}^{-1}m_{t-1}) = \mathbf{A}m_{t-1}$, and so we have,

$$\begin{aligned} p_\theta(x_t|y_{1:t-1}) &\propto \exp\left\{-\frac{1}{2}\left(x_t - \mathbf{A}m_{t-1}\right)^T \mathbf{P}_t^{-1}\left(x_t - \mathbf{A}m_{t-1}\right)\right\} \\ &\propto \text{MVN}\left(x_t; \mathbf{A}m_{t-1}, \mathbf{P}_t\right), \end{aligned} \tag{3.15}$$

and so we have shown the predictive density $p_\theta(x_t|y_{1:t-1})$ is Gaussian as desired.

Returning to the marginal filtering density in (3.12) and applying the known observation process error density (3.11) and the result in (3.15), we find,

$$\pi_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t)\cdot p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})} \propto g_\theta(y_t|x_t)\cdot p_\theta(x_t|y_{1:t-1}) \tag{3.16}$$

$$\propto \exp\left\{-\frac{1}{2}\left[(y_t - \mathbf{B}x_t)^T \mathbf{W}_t^{-1}(y_t - \mathbf{B}x_t) + (x_t - \mathbf{A}m_{t-1})^T \mathbf{P}_t^{-1}(x_t - \mathbf{A}m_{t-1})\right]\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\left[x_t^T\left(\mathbf{P}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}\right)x_t - 2x_t^T\left(\mathbf{P}_t^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t\right)\right]\right\}, \tag{3.17}$$

denoting by $\mathbf{C}_t^{-1} := \mathbf{P}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B} = (\mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T)^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}$ and by $m_t := \mathbf{C}_t(\mathbf{P}_t^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t) = \mathbf{C}_t[(\mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T)^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t]$, then we have,

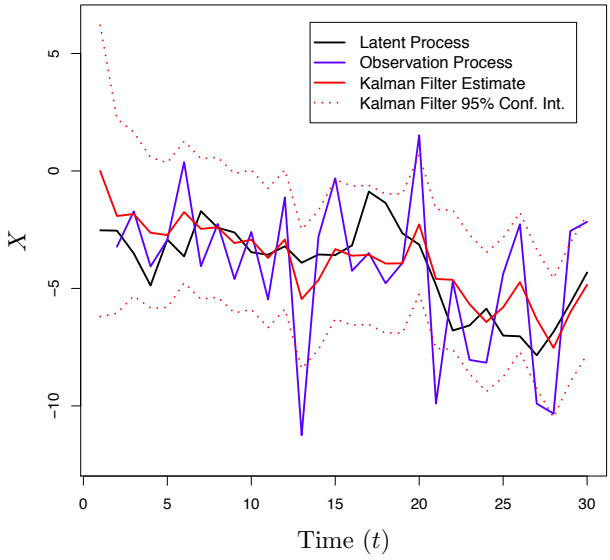
$$\begin{aligned} \pi_\theta(x_t|y_{1:t}) &\propto \exp\left\{-\frac{1}{2}\left(x_t - m_t\right)^T \mathbf{C}_t^{-1}\left(x_t - m_t\right)\right\} \\ &\propto \text{MVN}(x_t; m_t, \mathbf{C}_t). \end{aligned} \tag{3.18}$$

As a consequence we have shown that the marginal filtering density in (3.12) is Gaussian and so (3.13) holds by induction as desired.

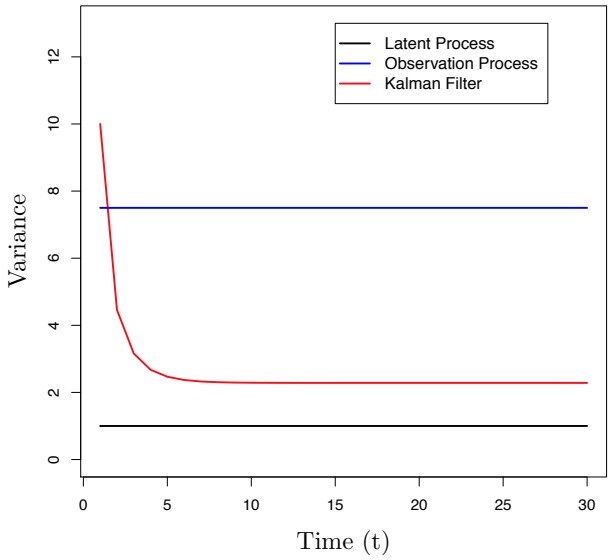
Intuitively the Kalman filter can be thought of (rather simply) as the recursion of a prior Gaussian state and Gaussian information to give a Gaussian posterior, which is then used at the next time point as the prior. In Figure 3.1.2 we present an example of a Kalman filter applied to a simulated observation process overlaid with the underlying latent process.

Although a HMM with Gaussian SSDs may seem limiting various extensions have been made which generalise the methodology to more complicated HMMs which have non-

linear and non-Gaussian SSDs. This is achieved by modifying the SSDs to fit them within the framework of the Kalman filter. Examples of such extensions include the *Extended Kalman Filter*, in which in cases where the SSDs are non-linear but sufficiently smooth considers a linearised version of them (see Anderson and Moore [1979] and Section 3.3.1.2 for a more detailed discussion), and the *Unscented Kalman Filter*, in which for highly non-linear SSDs involves deterministically sampling from the prior density (drawing so called *Sigma Points*) propagating the samples through the model SSDs and reconstructing the mean and covariance of the posterior (see Julier and Uhlmann [2004]). However, in all of these modifications to the Kalman filter we are making implicit approximations which, depending on the degree of true non-linearity and non-Gaussianity in the HMM, result in the particular choice of modified Kalman filter performing arbitrarily badly (without any grasp on how badly that may be).



(a) The observation process and Kalman filter with confidence interval, overlaid with the underlying latent process.



(b) Trace of the process variances over time (the latent and observation process being conditional variances, whereas the Kalman filter being the marginal variance).

Figure 3.1.2: An illustrative example of the Kalman filter applied to the HMM filtering problem with SSDs as follows, $X_0 \sim \mathrm{N}(0, 10)$, $X_t|(X_{t-1} = x_{t-1}) \sim \mathrm{N}(0, 1)$ and $Y_t|(X_t = x_t) \sim \mathrm{N}(0, 7.5)$.

3.2 Sequential Importance Sampling

Sequential Importance Sampling (SIS) (which in its simplest form dates back to at least Hammersley and Morton [1954] and Rosenbluth and Rosenbluth [1955], where the focus was on developing Monte Carlo methods for simulating long chain polymers), is simply an application of importance sampling (see Section 2.5) to the joint filtering problem (3.4), which as noted in Section 3.1.1 can be represented as the normalised recursion of the joint filtering density at the prior time point with the transition density of the latent process and the observation process error density. As briefly discussed in Section 3.1.4, the key problem to finding a representation of the filtering density is calculating the normalising constant, which in the case of the Kalman filter does not need to be calculated directly as the Gaussian SSDs and the resulting Gaussian densities which arise from the multiplication and convolution of these SSDs allows for it to be calculated indirectly. However, this is not possible for more general SSDs. The key idea presented in Gordon et al. [1993] is that evaluating the normalising constant can be circumvented by using importance sampling as the resulting importance weighted samples can jointly provide an unbiased estimate of the normalising constant (as we highlighted in (2.12) on page 23 of Section 2.5). As such we can restrict our attention to the unnormalised joint filtering problem,

$$\pi_\theta(x_{0:t}|y_{1:t}) \propto \pi_\theta(x_{0:t-1}|y_{1:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t). \tag{3.19}$$

Proceeding in the same manner as Section 2.5 we will suppose there exists some dominating density $q_\theta(x_{0:t}|y_{1:t})$ (which we will term the *joint importance function*), which has support including that of $\pi_\theta(x_{0:t}|y_{1:t})$. Furthermore, we will choose the joint importance function in such a way that it allows recursive updates in time as subsequent observations from the observation process become available,

$$q_\theta(x_{0:t}|y_{1:t}) = q_\theta(x_{0:t-1}|y_{1:t-1}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t}). \tag{3.20}$$

To avoid confusion we term the primitive $q_\theta(x_{0:t-1}|y_{1:t-1})$ as the *prior joint importance function* and $q_\theta(x_t|x_{0:t-1}, y_{1:t})$ as the *marginal importance function*. We avoid at this stage discussing the explicit form of the importance functions, which we discuss later in Section 3.3.

Now, suppose we want to evaluate expectations of the following form (where h is some

test function),

$$\mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t})\right] := \int h(x_{0:t}) \cdot \pi_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t}, \tag{3.21}$$

then re-expressing this expectation as an expectation with respect to our joint importance function $q_\theta(x_{0:t}|y_{1:t})$, and applying (3.19) and (3.20) we have,

$$\begin{aligned}
\mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t})\right] &= \int h(x_{0:t}) \cdot \frac{\pi_\theta(x_{0:t}|y_{1:t})}{q_\theta(x_{0:t}|y_{1:t})} \cdot q_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \\
&\propto \int h(x_{0:t}) \cdot \underbrace{\frac{\pi_\theta(x_{0:t-1}|y_{1:t-1})}{q_\theta(x_{0:t-1}|y_{1:t-1})} \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{q_\theta(x_t|x_{0:t-1}, y_{1:t})}}_{:=w_t^*(x_{0:t}) := w_{t-1}^*(x_{0:t-1}) \cdot w_{t|(t-1)}^*(x_{0:t})} \cdot q_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \qquad (3.22) \\
&= \mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t}) \cdot w_t^*(X_{0:t})\right], \qquad (3.23)
\end{aligned}$$

and so by applying the same argument as in (2.12) we have,

$$\frac{\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t}) \cdot w_t^*(X_{0:t})\right]}{\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[w_t^*(X_{0:t})\right]} = \mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t})\right]. \tag{3.24}$$

We could, provided $q_\theta(x_{0:t}|y_{1:t})$ is accessible, draw $X_{0:t}^{(1)}, X_{0:t}^{(2)}, \ldots, X_{0:t}^{(N)} \overset{\text{iid}}{\sim} q_\theta(x_{0:t}|y_{1:t})$ and assign the samples importance weights $w_t(X_{0:t}^{(i)}) := w_t^*(X_{0:t}^{(i)}) / \sum_{j=1}^N w_t^*(X_{0:t}^{(j)})$ in order to construct an asymptotically unbiased estimator of the target expectation in (3.21) (as we did in Section 2.5). By convention within the literature these samples are termed *particles* and the set of importance weighted particles (the *particle set*) are denoted by $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N := \{X_{0:t}^{(i)}, w_t(X_{0:t}^{(i)})\}_{i=1}^N$. In effect the particle set forms a discrete approximation of the joint filtering density (denoting δ as the Dirac delta measure),

$$\pi_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \approx \pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}). \tag{3.25}$$

The problem with simulating the particle set as suggested above is three-fold: $q_\theta(x_{0:t}|y_{1:t})$ may not be directly accessible (typically this density is accessible by construction, but in the setting we consider in this thesis, as detailed in Chapter 7, this density is not accessible); evaluating the weight function in (3.22) requires access to the prior joint filtering problem; and, with increasing observations the computational cost of constructing the unbiased estimator will increase (we would ideally have instead an online solution). Note however that the joint importance function has a recursive form by construction (see

(3.20)), and the unnormalised particle weights also have a recursive form as suggested by (3.22),

$$w_t^{*(i)} := w_{t-1}^{*(i)} \cdot \frac{f_\theta\left(x_t^{(i)} \,\middle|\, x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right)}{q_\theta\left(x_t^{(i)} \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)}. \tag{3.26}$$

As suggested by these recursive representations, given a particle set for the prior filtering density $\{x_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$ (which forms a discrete approximation of $\pi_\theta(x_{0:t-1}, y_{1:t-1})$), the trajectory of each particle can be extended to time t, re-weighted and then the weights normalised. Extension of each particle trajectory to time t can be preformed by sampling the marginal importance function, $x_t^{(i)} \sim q_\theta(x_t | x_{0:t-1}^{(i)}, y_{1:t})$ (which, as discussed later in Section 3.3, is typically accessible by construction), and then setting $x_{0:t}^{(i)} := \{x_{0:t-1}^{(i)}, x_t^{(i)}\}$. Updating the unnormalised importance weights can be performed as per (3.26) and normalised such that $w_t^{(i)} := w_t^{*(i)} / \sum_{j=1}^N w_t^{*(j)}$. The resulting particle set then forms a discrete approximation of the current joint filtering density (as in (3.25)).

In order to simulate an initial particle set we can exploit the known SSDs (see Section 3.1 Properties 1–3) and simply draw $x_0^{(1)}, x_0^{(2)}, \ldots, x_0^{(N)} \overset{\text{iid}}{\sim} \mu_\theta(x_0)$, which will either be possible directly or by employing one of the Monte Carlo methods presented in Chapter 2. Following an inductive argument it is possible to extend the particle set at any point in time to include further observations, and in doing so provide an online solution to filtering sequential observations. Calculating the normalising constant is entirely circumvented as suggested by (3.24), as the particle weights themselves jointly form an unbiased estimate

of $p_\theta(y_{1:t})$,

$$\begin{aligned}
\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}[w_t^*] &= \mathbb{E}_{q_\theta(x_{0:t-1}|y_{1:t-1})}\left[\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left(w_{t-1}^* \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{q_\theta(x_t|x_{0:t-1}, y_{1:t})} \,\middle|\, x_{0:t-1}\right)\right] \\
&= \mathbb{E}_{q_\theta(x_{0:t-1}|y_{1:t-1})}\left[w_{t-1}^* \cdot \mathbb{E}_{q_\theta(x_t|x_{0:t-1},y_{1:t})}\left(\frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{q_\theta(x_t|x_{0:t-1}, y_{1:t})}\right)\right] \\
&= \mathbb{E}_{q_\theta(x_{0:t-1}|y_{1:t-1})}\left[w_{t-1}^* \cdot p_\theta(y_t|x_{t-1})\right] \qquad (3.27) \\
&= \mathbb{E}_{q_\theta(x_{0:t-2}|y_{1:t-2})}\left[\mathbb{E}_{q_\theta(x_{0:t-1}|y_{1:t-1})}\left[w_{t-2}^* \cdot \frac{f_\theta(x_{t-1}|x_{t-2}) \cdot g_\theta(y_{t-1}|x_{t-1})}{q_\theta(x_{t-1}|x_{0:t-2}, y_{1:t-1})} \cdot p_\theta(y_t|x_{t-1}) \,\middle|\, x_{0:t-2}\right]\right] \\
&= \mathbb{E}_{q_\theta(x_{0:t-2}|y_{1:t-2})}\left[w_{t-2}^* \cdot \int f_\theta(x_{t-1}|x_{t-2}) \cdot g_\theta(y_{t-1}|x_{t-1}) \cdot p_\theta(y_t|x_{t-1}) \,\mathrm{d}x_{t-1}\right] \\
&= \mathbb{E}_{q_\theta(x_{0:t-2}|y_{1:t-2})}\left[w_{t-2}^* \cdot \int p_\theta(x_{t-1}, y_{t-1:t}|x_{t-2}) \,\mathrm{d}x_{t-1}\right] \\
&= \mathbb{E}_{q_\theta(x_{0:t-2}|y_{1:t-2})}\left[w_{t-2}^* \cdot p_\theta(y_{t-1:t}|x_{t-2})\right] = \cdots \\
&= \int p_\theta(y_{1:t}|x_0) \cdot \mu_\theta(x_0) \,\mathrm{d}x_0 = p_\theta(y_{1:t}). \qquad (3.28)
\end{aligned}$$

The above argument leads to the SIS algorithm which we present in Algorithm 3.2.1, which can be used for tackling filtering problems for a broader class of SSDs than the Kalman filter in Section 3.1.4 (which was limited to Gaussian SSDs).

It should be noted that there are a number of implementational considerations which we have not yet discussed but instead address later in this chapter. In particular, in analogous fashion to importance sampling, it is critical to choose an appropriate marginal importance function. In the context of SIS this can be interpreted as ensuring the prior particle set is propagated into areas of the sample space which are sufficiently likely, which is an issue we address in Sections 3.3 and 3.6. Another closely related issue is that of particle *degeneracy*, whereby the particle set becomes increasingly dominated by heavily weighted particles over increasing numbers of observations, thereby resulting in coarse approximations to the joint filtering density. This can be addressed by *resampling* and the alternate *Sequential Importance Sampling / Resampling* (SISR) algorithm which we discuss in Section 3.4.

Algorithm 3.2.1 Sequential Importance Sampling (SIS) Algorithm [Gordon et al., 1993].

Initialisation Step ($t = 0$)**:-**

1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$ and set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. For i in 1 to N,
   (a) Simulate $x_t^{(i)} \sim q_\theta\left(x_t \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)$ and set $x_{0:t}^{(i)} := \left\{x_{0:t-1}^{(i)}, x_t^{(i)}\right\}$.
   (b) Set $w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \dfrac{f_\theta\left(x_t^{(i)} \,\middle|\, x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right)}{q_\theta\left(x_t^{(i)} \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)}$.
2. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
3. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.

3.3 Marginal Importance Function Selection

As outlined briefly in Section 3.2, in order to implement Algorithm 3.2.1 we need to choose an appropriate marginal importance function, $q_\theta(x_t|x_{0:t-1}, y_{1:t})$. As with importance sampling (see Section 2.5), a natural choice of the marginal importance function is the one that minimises the variance of the importance weights, or noting the following representation, minimises the marginal increase in the variance of the importance weights,

$$\mathbb{V}\mathrm{ar}_{q_\theta(x_{0:t}\,|\,y_{1:t})}\left[w_t^* \,\middle|\, x_{0:t-1}\right] = \left(w_{t-1}^*\right)^2 \cdot \mathbb{V}\mathrm{ar}_{q_\theta(x_t\,|\,x_{0:t-1}, y_{1:t})}\left[\frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{q_\theta(x_t|x_{0:t-1}, y_{1:t})}\right]. \quad (3.29)$$

Denoting by $w_{t\,|\,(t-1)}^*(x_{0:t}) := f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)/q_\theta(x_t|x_{0:t-1}, y_{1:t})$ – which is the marginal change in the weight function from (3.22) – we want to choose $q_\theta(x_t|x_{0:t-1}, y_{1:t})$ to min-

imise the following expression,

$$\begin{aligned}
&\mathbb{V}\mathrm{ar}_{q_\theta(x_t|x_{0:t-1},y_{1:t})}\left[w^*_{t|(t-1)}(x_{0:t})\right] \\
&\quad= \mathbb{E}_{q_\theta(x_t|x_{0:t-1},y_{1:t})}\left[\left(w^*_{t|(t-1)}(x_{0:t})\right)^2\right] - \left(\mathbb{E}_{q_\theta(x_t|x_{0:t-1},y_{1:t})}\left[w^*_{t|(t-1)}(x_{0:t})\right]\right)^2 \\
&\quad= \int \frac{(f_\theta(x_t|x_{t-1})\cdot g_\theta(y_t|x_t))^2}{q_\theta(x_t|x_{0:t-1},y_{1:t})}\,\mathrm{d}x_t - \left(\int f_\theta(x_t|x_{t-1})\cdot g_\theta(y_t|x_t)\,\mathrm{d}x_t\right)^2 \\
&\quad= \left(\int \frac{(f_\theta(x_t|x_{t-1})\cdot g_\theta(y_t|x_t))^2}{q_\theta(x_t|x_{0:t-1},y_{1:t})}\,\mathrm{d}x_t\right) - (p_\theta(y_t|x_{t-1}))^2\,. \qquad (3.30)
\end{aligned}$$

In the rest of this section we present a number of alternate choices for the marginal importance function. In Section 3.3.1 we present the *Optimal Marginal Importance Function* along with illustrative examples of SSDs in which it can be found in Sections 3.3.1.1 and 3.3.1.2. In practice finding a tractable representation for the optimal importance function can be difficult, so we also introduce the *Prior Marginal Importance Function* in Section 3.3.2 (which was the selection originally proposed in Gordon et al. [1993] for inclusion within the SIS algorithm of Section 3.2) and, for completeness, the *Fixed Marginal Importance Function* in Section 3.3.3.

3.3.1 Optimal Marginal Importance Function

Selection of the marginal importance function $q_\theta(x_t|x_{0:t-1},y_{1:t}) := p_\theta(x_t|x_{t-1},y_t)$ was first introduced in Zaritskii et al. [1975] and Akashi and Kumamoto [1977]. It was further shown in Doucet et al. [2000] that it is the *Optimal Marginal Importance Function* as substituting it into (3.30) and applying Bayes' rule reduces the variance of the marginal importance weight function to zero,

$$\begin{aligned}
\mathbb{V}\mathrm{ar}_{q_\theta(x_t|x_{0:t-1},y_{1:t})}\left[w^*_{t|(t-1)}(x_{0:t})\right] &= \left(\int \frac{(f_\theta(x_t|x_{t-1})\cdot g_\theta(y_t|x_t))^2}{p_\theta(x_t|x_{t-1},y_t)}\,\mathrm{d}x_t\right) - (p_\theta(y_t|x_{t-1}))^2 \\
&= \left(\int f_\theta(x_t|x_{t-1})\cdot g_\theta(y_t|x_t)\cdot p_\theta(y_t|x_{t-1})\,\mathrm{d}x_t\right) - (p_\theta(y_t|x_{t-1}))^2 \\
&= \left(p_\theta(y_t|x_{t-1})\cdot\int p_\theta(x_t,y_t|x_{t-1})\,\mathrm{d}x_t\right) - (p_\theta(y_t|x_{t-1}))^2 \\
&= (p_\theta(y_t|x_{t-1}))^2 - (p_\theta(y_t|x_{t-1}))^2 = 0. \qquad (3.31)
\end{aligned}$$

Furthermore, with this selection of marginal importance function the importance weights in Algorithm 3.2.1 Step 1b are updated as follows,

$$w_t^{*(i)} := w_{t-1}^{*(i)} \cdot \frac{f_\theta\left(x_t \middle| x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right)}{p_\theta\left(x_t \middle| x_{t-1}^{(i)}, y_t\right)} = w_{t-1}^{*(i)} \cdot p_\theta\left(y_t \middle| x_{t-1}^{(i)}\right). \tag{3.32}$$

The optimal marginal importance function is typically not implementable directly as samples are required from the future state $p_\theta(x_t|x_{t-1}, y_t)$ and additionally the generally intractable $p_\theta(y_t|x_{t-1})$ has to be evaluated. Techniques to overcome these issues are typically analogous to techniques used for the Kalman filter such as local linearisation of the HMM (similar to the extended Kalman filter) and local linearisation of the optimal importance function (using a Taylor expansion) as illustrated in Doucet et al. [2000]. Note however that unlike analytic approximations of the filter, the approximation error introduced here is controlled. In particular, it results in an increase in Monte Carlo variance which can be reduced by simply increasing the size of the particle set.

3.3.1.1 Linear Gaussian State Space Dynamics Example

As discussed in Section 3.3.1, in order to implement Algorithm 3.2 with the optimal marginal importance function we require access to both $p_\theta(x_t|x_{t-1}, y_t)$ and $p_\theta(y_t|x_{t-1})$. In order to illustrate a simple example of this lets consider a HMM with the same linear Gaussian SSDs as the Kalman filter in Section 3.1.4 (see (3.9), (3.10) and (3.11)).

First considering the density $p_\theta(x_t|x_{t-1}, y_t)$ note that, in a similar manner to Section 3.1.4, by applying Bayes rule then under the SSDs of (3.9) – (3.11) it can be represented as the multiplication of two Gaussian densities and hence is Gaussian itself,

$$\begin{aligned}
p_\theta(x_t|x_{t-1}, y_t) &\propto f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t) \\
&\propto \exp\left\{-\frac{1}{2}\left[(x_t - \mathbf{A}x_{t-1})^T \mathbf{V}_t^{-1} (x_t - \mathbf{A}x_{t-1}) + (y_t - \mathbf{B}x_t)^T \mathbf{W}_t^{-1} (y_t - \mathbf{B}x_t)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[x_t^T \left(\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}\right) x_t - 2x_t^T \left(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t\right)\right]\right\} \\
&\propto \mathrm{MVN}\left(\left(\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}\right)^{-1} \left(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t\right), \left(\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}\right)^{-1}\right).
\end{aligned} \tag{3.33}$$

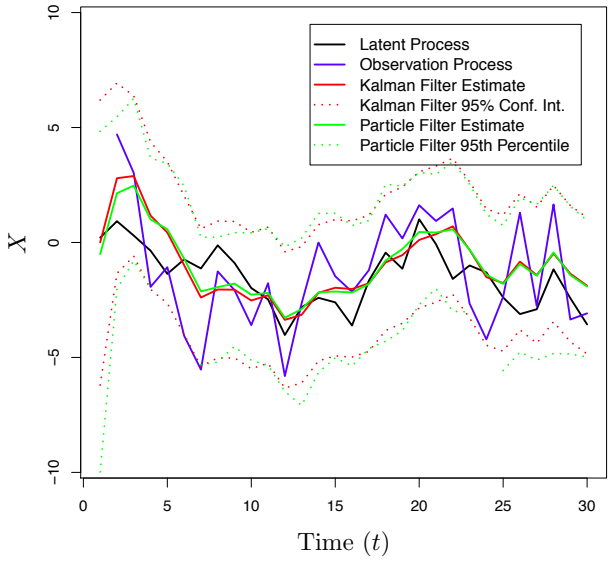
Now considering the density $p_\theta(y_t|x_{t-1})$, it can be shown to be the convolution of two Gaussian densities and hence also Gaussian. In particular, we have,

$$p_\theta(y_t|x_{t-1}) = \int p_\theta(x_t, y_t|x_{t-1})\,\mathrm{d}x_t = \int f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)\,\mathrm{d}x_t$$
$$\propto \int \exp\left\{-\frac{1}{2}\left[x_t^T\left(\mathbf{V}_t^{-1}+\mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}\right)x_t - 2x_t^T\left(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1}+\mathbf{B}^T\mathbf{W}_t^{-1}y_t\right)+y_t^T\mathbf{W}_t^{-1}y_t\right]\right\}\mathrm{d}x_t,$$

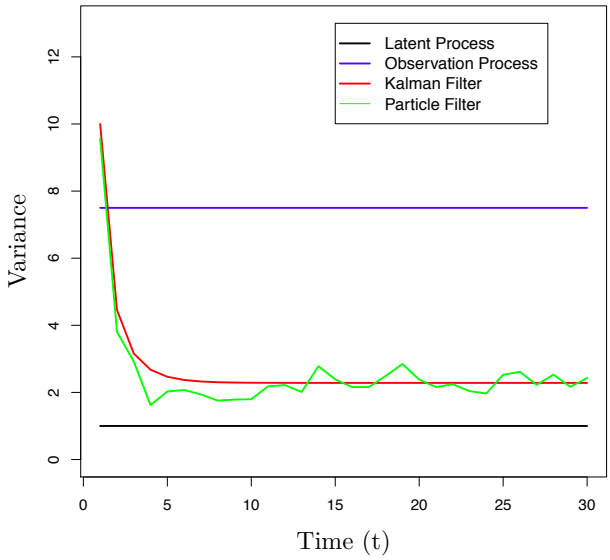
now, following the same calculation as in (3.15) and again applying the Woodbury Matrix Identity (see Woodbury [1950]), we have (denoting by $\mathbf{Z}_t^{-1} := \mathbf{V}_t^{-1}+\mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}$ and $s_t := \mathbf{Z}_t(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1}+\mathbf{B}^T\mathbf{W}_t^{-1}y_t)$),

$$\begin{aligned}
p_\theta(y_t|x_{t-1}) &\propto \exp\left\{-\frac{1}{2}\left[y_t^T\mathbf{W}_t^{-1}y_t - s_t^T\mathbf{Z}_t^{-1}s_t\right]\right\}\underbrace{\int \exp\left\{-\frac{1}{2}(x_t-s_t)^T\,\mathbf{Z}_t^{-1}\,(x_t-s_t)\right\}\mathrm{d}x_t}_{\propto 1}\\
&\propto \exp\left\{-\frac{1}{2}\left[y_t^T\left(\mathbf{W}_t^{-1}-\mathbf{W}_t^{-1}\mathbf{B}\mathbf{Z}_t\mathbf{B}^T\mathbf{W}_t^{-1}\right)y_t - 2y_t^T\left(\mathbf{W}_t^{-1}\mathbf{B}\mathbf{Z}_t\mathbf{V}_t^{-1}\mathbf{A}x_{t-1}\right)\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}(y_t-\mathbf{B}\mathbf{A}x_{t-1})^T\left(\mathbf{W}_t+\mathbf{B}\mathbf{V}_t\mathbf{B}^T\right)^{-1}(y_t-\mathbf{B}\mathbf{A}x_{t-1})\right\}\\
&\propto \mathrm{MVN}\left(\mathbf{B}\mathbf{A}x_{t-1}, \mathbf{W}_t+\mathbf{B}\mathbf{V}_t\mathbf{B}^T\right). \qquad (3.34)
\end{aligned}$$

We conclude this subsection by providing an illustrative example of a particle filter with optimal marginal importance function in Figure 3.3.1.



(a) The observation process, particle filter and Kalman filter with confidence interval, overlaid with the underlying latent process.



(b) Trace of the process variances over time (the latent and observation process being conditional variances, whereas the particle filter is the posterior variance estimate).

Figure 3.3.1: An illustrative example of a particle filter (with an optimal marginal importance function selection) of 100 particles applied to the HMM filtering problem with SSDs as follows, $X_0 \sim \mathrm{N}(0, 10)$, $X_t|(X_{t-1} = x_{t-1}) \sim \mathrm{N}(0, 1)$ and $Y_t|(X_t = x_t) \sim \mathrm{N}(0, 7.5)$.

3.3.1.2 Non-linear Gaussian State Space Dynamics Example

In a similar fashion to the extension of the Kalman filter to the extended Kalman filter it is worthwhile considering cases where the HMM has non-linear SSDs,

$$X_0 \sim \text{MVN}(m_0, \mathbf{C}_0), \tag{3.35}$$

$$X_t|(X_{t-1} = x_{t-1}) \sim \text{MVN}\left(a(x_{t-1}), \mathbf{V}_t\right), \tag{3.36}$$

$$Y_t|(X_t = x_t) \sim \text{MVN}\left(b(x_t), \mathbf{W}_t\right). \tag{3.37}$$

Assuming that both a and b are differentiable, then as per Doucet et al. [2000] and Anderson and Moore [1979], applying a first order approximation of the observation mean we have (letting D represent some constant),

$$\begin{aligned} b(x_t) &\approx b\left(a(x_{t-1})\right) + \left.\frac{\partial b(x_t)}{\partial x_t}\right|_{x_t = a(x_{t-1})} \left(x_t - a(x_{t-1})\right) \\ &= \left.\frac{\partial b(x_t)}{\partial x_t}\right|_{x_t = a(x_{t-1})} x_t + \left(b\left(a(x_{t-1})\right) - \left.\frac{\partial b(x_t)}{\partial x_t}\right|_{x_t = a(x_{t-1})} a(x_{t-1})\right) =: \mathbf{M}x_t + D. \end{aligned} \tag{3.38}$$

Now considering the approximated optimal marginal importance function $q_\theta(x_t|x_{0:t-1}, y_{1:t}) := \hat{p}_\theta(x_t|x_{t-1}, y_t)$ (or more intuitively the optimal marginal importance function for the approximated HMM), then in a similar manner to (3.33) we find,

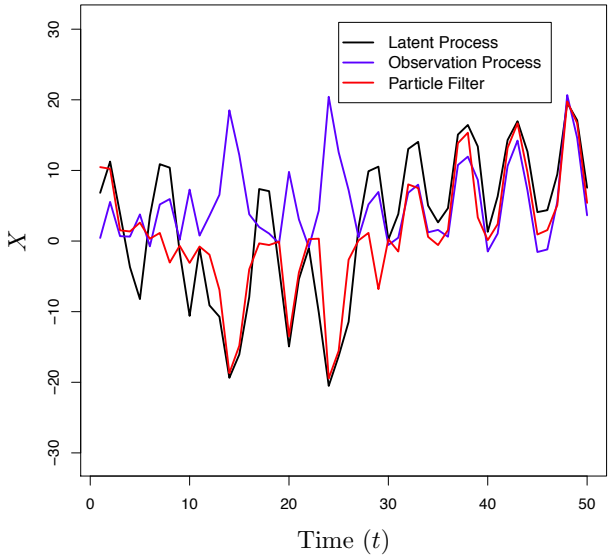
$$\begin{aligned} \hat{p}_\theta(x_t|x_{t-1}, y_t) &\propto f_\theta(x_t|x_{t-1}) \cdot \hat{g}_\theta(y_t|x_t) \\ &\propto \exp\left\{-\frac{1}{2}\left[\left(x_t - a(x_{t-1})\right)^T \mathbf{V}_t^{-1}\left(x_t - a(x_{t-1})\right) + \left(y_t - \mathbf{M}x_t - D\right)^T \mathbf{W}_t^{-1}\left(y_t - \mathbf{M}x_t - D\right)\right]\right\} \\ &\propto \text{MVN}\left(\left(\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M}\right)^{-1}\left(\mathbf{V}_t^{-1}a(x_{t-1}) + \mathbf{M}^T\mathbf{W}_t^{-1}(y_t - D)\right), \left(\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M}\right)^{-1}\right). \end{aligned}$$

Now considering the approximated density $\hat{p}_\theta(y_t|x_{t-1})$, in analogous fashion to (3.15) and (3.34), we have,

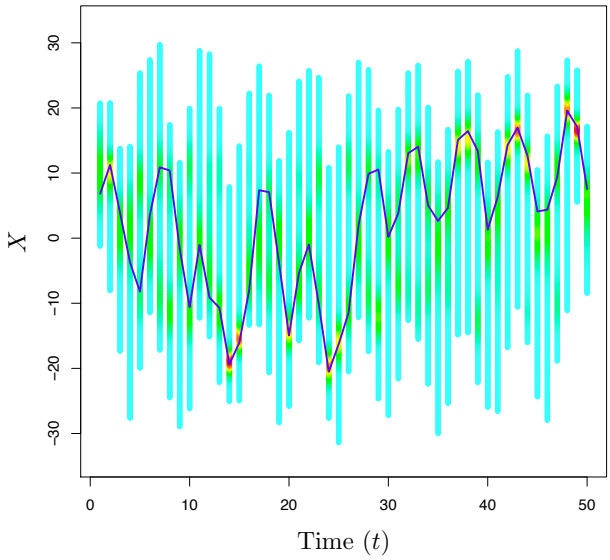
$$\begin{aligned} \hat{p}_\theta(y_t|x_{t-1}) &= \int f_\theta(x_t|x_{t-1}) \cdot \hat{g}_\theta(y_t|x_t)\,\mathrm{d}x_t \\ &\propto \exp\left\{-\frac{1}{2}\left[(y_t - D)^T\mathbf{W}_t^{-1}(y_t - D) - \left(\mathbf{V}_t^{-1}a(X_t) + \mathbf{M}^T\mathbf{W}_t^{-1}(y_t - D)\right)^T\right.\right. \\ &\qquad\qquad \left.\left.\left(\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M}\right)^{-1}\left(\mathbf{V}_t^{-1}a(X_t) + \mathbf{M}^T\mathbf{W}_t^{-1}(y_t - D)\right)\right]\right\} \\ &\propto \text{MVN}\left(\mathbf{M}a(x_{t-1}), \mathbf{W}_t + \mathbf{M}\mathbf{V}_t\mathbf{M}^T\right). \end{aligned} \tag{3.39}$$

It should be noted that although many of the techniques employed to the optimal marginal importance function in order to broaden its application are similar to those of the Kalman filter, particle filters have the distinct advantage that the sub-optimality of any given marginal importance function proposed is captured within the state estimates (as an increase in Monte Carlo variance) whereas the Kalman filter simply provides an approximation to the Gaussian case without any explicit evaluation of the discrepancy.

To finish this section, we provide an example of a particle filter (with linearised optimal marginal importance function) applied to a highly non-linear HMM in Figure 3.3.2.



(a) The observation process and particle filter, overlaid with the underlying latent process.



(b) A heat map of the empirical marginal filtering density over time (from blue indicating low density, to red indicating high density), overlaid with the underlying latent process (black line).

Figure 3.3.2: An illustrative example of a particle filter (with a linearised optimal marginal importance function selection) of 1000 particles applied to a highly non-linear HMM filtering problem with SSDs as follows, $X_0 \sim \mathrm{N}(0.05 + 2.5/(1 + 0.1^2) + 8, \sqrt{10})$, $X_t|(X_{t-1} = x_{t-1}) \sim \mathrm{N}(0.5x_{t-1} + 2x_{t-1}/(1 + x_{t-1}^2) + 8\cos(1.2(t-1)), \sqrt{10})$ and $Y_t|(X_t = x_t) \sim \mathrm{N}\left(x_t^2/20, \sqrt{1}\right)$.

3.3.2 Prior Marginal Importance Function

The *Prior Importance Function* with the importance function selection $q_\theta(x_t|x_{0:t-1}, y_{1:t}) := f_\theta(x_t|x_{t-1})$, was introduced in Handschin and Mayne [1969] and is so called as the choice of importance function does not take into account the current observation. It should be noted that the original particle filter proposed in Gordon et al. [1993] (*the bootstrap filter*), in effect used a prior marginal importance function. The importance weights for implementation in Algorithm 3.2 can be shown to be,

$$w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \frac{f_\theta\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right)}{f_\theta\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right)} = w_{t-1}^{*(i)} \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right). \tag{3.40}$$

The prior marginal importance function is considerably easier to evaluate as the update to the importance weights is a known element of the SSDs. However it suffers from more rapid degeneracy (see Section 3.4) as it is overly sensitive to the current observation. In effect information regarding the observation process is incorporated one step later than optimal resulting in poor propagation of the prior samples to the posterior state. More formally the results relating to importance weight variance in the optimal case (Section 3.3.1) show the prior importance function to be sub-optimal. On the other hand it is very intuitively appealing as the new importance weights are simply the prior importance weights adjusted for the new observation.

3.3.3 Fixed Marginal Importance Function

The *Fixed Marginal Importance Function* with the marginal importance function selection $q_\theta(x_t|x_{0:t-1}, y_{1:t}) := p_\theta(x_t)$ was first proposed in Tanizaki [1993] and Tanizaki and Mariano [1994] and is arguably an even simpler importance function selection. However, as it in no way incorporates the observation, or the sample path dynamics, it performs very poorly. For completeness, the importance weights can be found as follows,

$$w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \frac{f_\theta\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right)}{p_\theta\left(x_t^{(i)}\right)}. \tag{3.41}$$

3.4 Sequential Importance Sampling / Resampling

The Sequential Importance Sampling algorithm detailed in Algorithm 3.2 suffers from particle *degeneracy*, whereby the particle set on average becomes increasingly dominated by heavily weighted particles over increasing numbers of observations, thereby resulting in coarse approximations to the joint filtering density. As shown in Kong et al. [1994] and Doucet et al. [2000], the variance of the sample path importance weights stochastically increases as the random observation process evolves.

To show this we can consider the i^{th} particle process, which can be thought of as evolving as per the directed acyclic graph in Figure 3.4.1, and examine how its associated importance weight changes over one time step with the inclusion of one further observation. To begin note that the joint density of the i^{th} particle process and the observation process is as follows,

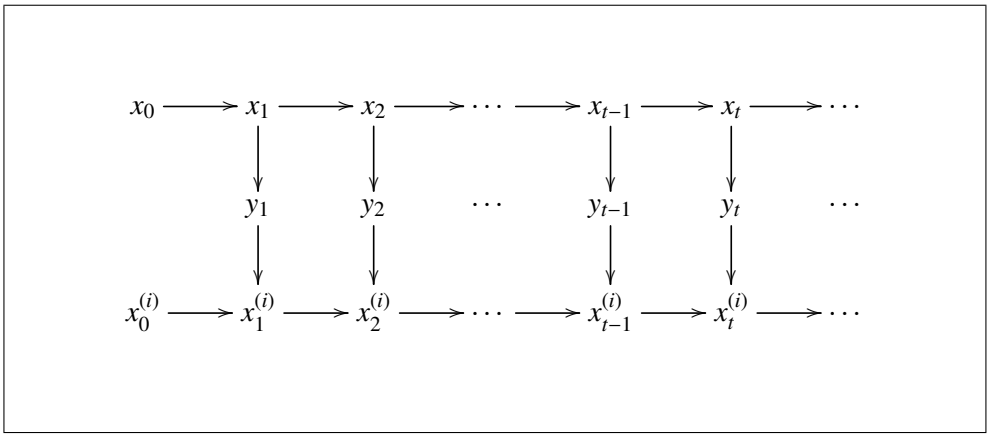


Figure 3.4.1: Directed acyclic graph representing the observation process of a Hidden Markov Model and the i^{th} particle process under the SIS algorithm.

$$\begin{aligned} q_\theta\left(x_{0:t}^{(i)}, y_{1:t}\right) &= \int \mu_\theta(x_0) \cdot \left[\prod_{n=1}^{t} f_\theta(x_n|x_{n-1}) \cdot g_\theta(y_n|x_n) \cdot q_\theta\left(x_n^{(i)} \,\middle|\, x_{0:n-1}^{(i)}, y_{1:n}\right)\right] \mathrm{d}x_{0:t} \\ &= \prod_{n=1}^{t} p_\theta(y_n|y_{1:n-1}) \cdot q_\theta\left(x_n^{(i)} \,\middle|\, x_{0:n-1}^{(i)}, y_{1:n}\right). \end{aligned} \tag{3.42}$$

Now, under the random evolution of this process the importance weight of the i^{th} particle is a martingale,

$$\begin{aligned}
&\mathbb{E}_{q_\theta\left(x_t^{(i)}, y_t \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t-1}\right)}\left[\frac{w_t^{*(i)}}{p_\theta(y_{1:t})}\right] \\
&= \int\int \frac{w_{t-1}^{*(i)}}{p_\theta(y_{1:t-1})} \cdot \frac{f_\theta\left(x_t^{(i)} \,\middle|\, x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right)}{p_\theta(y_t|y_{1:t-1}) \cdot q\left(x_t^{(i)} \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)} \cdot \left(p_\theta(y_t|y_{1:t-1}) \cdot q\left(x_t^{(i)} \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)\right) \mathrm{d}x_t^{(i)}\, \mathrm{d}y_t \\
&= \frac{w_{t-1}^{*(i)}}{p_\theta(y_{1:t-1})} \cdot \int\int p_\theta\left(x_t^{(i)}, y_t \,\middle|\, x_{0:t-1}^{(i)}\right) \mathrm{d}x_t^{(i)}\, \mathrm{d}y_t = \frac{w_{t-1}^{*(i)}}{p_\theta(y_{1:t-1})},
\end{aligned} \tag{3.43}$$

and so by the law of total variance we have that the variance of the importance weight is stochastically increasing over time,

$$\begin{aligned}
\mathbb{V}\mathrm{ar}_{q_\theta\left(x_{0:t}^{(i)}, y_{1:t}\right)}\left[\frac{w_t^{*(i)}}{p_\theta(y_{1:t})}\right] &\geq \mathbb{V}\mathrm{ar}_{q_\theta\left(x_{0:t-1}^{(i)}, y_{1:t-1}\right)}\left[\mathbb{E}_{q_\theta\left(x_t^{(i)}, y_t \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)}\left(\frac{w_t^{*(i)}}{p_\theta(y_{1:t})}\right)\right] \\
&= \mathbb{V}\mathrm{ar}_{q_\theta\left(x_{0:t-1}^{(i)}, y_{1:t-1}\right)}\left[\frac{w_{t-1}^{*(i)}}{p_\theta(y_{1:t-1})}\right].
\end{aligned} \tag{3.44}$$

Resampling is the natural solution to particle degeneracy. If the importance weights are deemed to have degenerated sufficiently then new sample paths are drawn from the weighted empirical distribution of the current state and the importance weights are reset to have equal weighting. In order to decide at which point to resample the heuristic is that we compare the number of particles of the particle set of size N which effectively contribute to the state estimates, N_{eff}, to some user specified threshold, N_{th}. As proposed by Kong [1992] and extended in Kong et al. [1994], a natural approach to choosing and calculating N_{eff} (or the *Effective Sample Size*) is to set it equal to the number of equally weighted independent samples drawn from the joint filtering density ($\pi_\theta(x_{0:t}|y_{1:t})$) that would be required in order to achieve the same Monte Carlo variance as the existing N particles from the joint importance function ($q_\theta(x_{0:t}|y_{1:t})$). This leads to the following heuristic,

$$N_{\text{eff}} := N \cdot \frac{\mathbb{V}\mathrm{ar}_{\pi_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t})\right]}{\mathbb{V}\mathrm{ar}_{q_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t}) \cdot w_t(X_{0:t})\right]}. \tag{3.45}$$

As (3.45) can't readily be evaluated, Kong et al. [1994] apply the delta method using the first two moments of w and h which results in the following approximation,

$$N_{\text{eff}} \approx \frac{N}{1 + \mathbb{V}\text{ar}_{q_\theta(x_{0:t}|y_{1:t})}\left[w_t(X_{0:t})\right]} \tag{3.46}$$

$$= \frac{N}{\left(\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[w_t(X_{0:t})\right]\right)^2 + \mathbb{V}\text{ar}_{q_\theta(x_{0:t}|y_{1:t})}\left[w_t(X_{0:t})\right]}$$

$$= \frac{N}{\mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[(w_t(X_{0:t}))^2\right]}$$

$$\approx \frac{1}{\sum_{i=1}^{N}\left(w_t^{(i)}\right)^2} =: \widehat{N_{\text{eff}}}. \tag{3.47}$$

In summary, it can be shown that the importance weights in the SIS algorithm (see Algorithm 3.2.1) are stochastically increasing over time (see (3.44)) and that to combat this a resampling mechanism can be introduced whereby resampling occurs whenever the effective sample size degenerates below some threshold level. The precise manner in which resampling is conducted is addressed in Section 3.5. This leads to the extension of the SIS algorithm to the *Sequential Importance Sampling / Resampling* (SISR) algorithm which we present in Algorithm 3.4.1. In its simplest form the SISR algorithm was proposed in Gordon et al. [1993], however, the algorithm we present accounts for the insight offered by the effective sample size argument presented above (as proposed by Kong et al. [1994]), along with the importance function selection discussed in Section 3.3 and the resampling algorithms which are presented in Section 3.5.

Additional complications arise from the SISR algorithm that are not present in the SIS algorithm. As illustrated in Figure 3.4.2, as an artefact of resampling a number of the particles at any time point will share common ancestors and so sample paths are no longer independent. Furthermore this problem is exacerbated over time as at resampling times there is some probability that entire ancestor paths are lost. As such obtaining convergence results is more involved in this setting (although there exists an extensive literature, for instance, Chopin [2004], Künsch [2005] and [Del Moral, 2004, Chap. 9]). These issues however are not addressed in this thesis.

Algorithm 3.4.1 Sequential Importance Sampling / Resampling (SISR) Algorithm [Gordon et al., 1993].

Initialisation Step ($t = 0$)**:-**

1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$, set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. If $\widehat{N_{\text{eff}}} \leq N_{\text{th}}$ then for i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t-1})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Simulate $x_t^{(i)} \sim q_\theta\left(x_t \middle| x_{0:t-1}^{(i)}, y_{1:t}\right)$ and set $x_{0:t}^{(i)} := \left\{x_{0:t-1}^{(i)}, x_t^{(i)}\right\}$.
   (b) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot \dfrac{f_\theta\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right)}{q_\theta\left(x_t^{(i)} \middle| x_{0:t-1}^{(i)}, y_{1:t}\right)}$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.

3.5 Resampling Methods

Within the Sequential Importance Sampling / Resampling algorithm (Algorithm 3.4) and as discussed in Section 3.4, in order to combat importance weight degeneracy the particle set is resampled if the effective sample size (N_{eff}) falls below some user specified threshold (N_{th}). This ensures that future state estimation isn't reliant on only a small proportion of the particles in the particle set (which are relatively heavily weighted), and that computation isn't spent on updating particles which have negligible weight. However, resampling necessarily increases current state estimate variance as the resampled state estimate is merely an approximation of the original state estimate. Indeed, for this reason in any particular iteration of SISR the particle set which is resampled is that which forms the state estimate of the prior state, $\{x_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$, and is conducted prior to the propagation and re-weighting of the particle set at the current iteration.

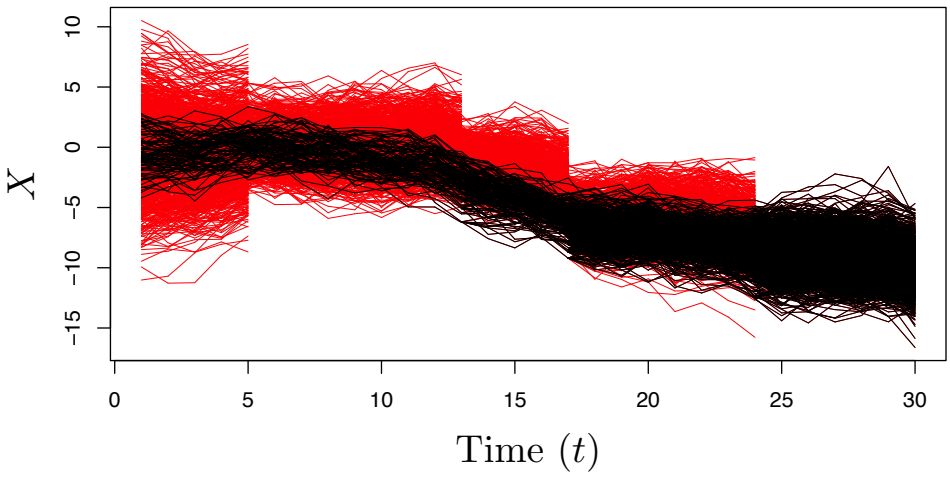


Figure 3.4.2: An illustration of 250 particle sample paths arising from the Sequential Importance Sampling / Resampling algorithm under the SSDs in Figure 3.3.1 over the interval $[1, 30]$ using the optimal marginal importance function and multinomial resampling (see Section 3.5.1). Sample paths in black denote the ancestry of the particles constituting the empirical joint filtering density at time 30, whereas sample paths in red indicate those which were no longer included after a resampling point.

Now, addressing how to best resample the particle set $(x^{(i)}_{0:t-1} \sim \pi^N_\theta(\mathrm{d}x_{0:t-1}|y_{1:t-1}))$, we follow some guiding principles. To begin with we would typically want the resampled particle set, $\{\tilde{x}^{(i)}_{0:t-1}, 1/N\}^N_{i=1}$, to be composed of N particles as having a stable number of particles at all times is advantageous for implementational reasons. Furthermore, as we are sampling from the existing particle set we naturally consider the number of *offspring* $O^{(i)}_{t-1}$ of each existing particle (the number of times each particle sample path is resampled). Considering the offspring gives us some traction in determining how to resample. In particular, it is clearly desirable to ensure unbiasedness, $\mathbb{E}(O^{(i)}_{t-1}|w^{(i)}_{t-1}) = N \cdot w^{(i)}_{t-1}\ \forall i \in N$, while minimising the introduction of additional variance, $\mathbb{V}\mathrm{ar}(O^{(i)}_{t-1}|w^{(i)}_{t-1})$.

In the remainder of this section we consider a number of standard resampling methods. In Section 3.5.1 we consider *Multinomial Resampling*, which is the intuitively appealing method originally proposed for inclusion within the SISR algorithm by Gordon et al. [1993]. However, multinomial resampling typically performs very poorly against other popular resampling methodologies such as *Systematic Resampling*, *Stratified Resampling* and *Residual Resampling* which we detail in Sections 3.5.2, 3.5.3 and 3.5.4 respectively

(see for instance Douc et al. [2005] for a comparison). An extensive literature on resampling methods is outlined in Doucet et al. [2001], including unequally resampled importance weights [Kitagawa, 1996] and partial resampling [Carpenter et al., 1999b]. It is worth noting that there are also methods which seek to minimise mean square error as opposed to variance (resulting in bias) [Kitagawa, 1996], however, the methods considered within this section are the most popular ones.

3.5.1 Multinomial Resampling

Intuitively, in order to draw a single particle from the empirical estimate of the prior filtering density $\pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t-1})$, constructed from the particle set $\{x_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$, one could simply draw randomly one of the particles (where the probability of any individual particle being selected corresponds to its weight). *Multinomial Resampling*, as detailed in Algorithm 3.5.1 and first employed in the context of the SISR algorithm by Gordon et al. [1993], is simply the extension of this notion to the simulation of N particles. Clearly employing this algorithm results in the desired unbiasedness, $\mathbb{E}(O_{t-1}^{(i)}|w_{t-1}^{(i)}) = N.w_{t-1}^{(i)}$, and furthermore results in the following resampled empirical density estimate,

$$\tilde{\pi_\theta}^N(\mathrm{d}x_{0:t-1}|y_{1:t-1}) = \sum_{i=1}^{N} \frac{O_{t-1}^{(i)}}{N} \cdot \delta_{\tilde{x}_{0:t-1}^{(i)}}(\mathrm{d}x_{0:t-1}). \tag{3.48}$$

Algorithm 3.5.1 Multinomial Resampling Algorithm [Gordon et al., 1993].

1. For i in 1 to N sample $J \sim \text{categorical}(w_{t-1}^{(1)}, \ldots, w_{t-1}^{(N)})$ and set $\tilde{x}_{0:t-1}^{(i)} := x_{0:t-1}^{(J)}$.

3.5.2 Systematic Resampling

Systematic Resampling, as introduced by Kitagawa [1996] and detailed in Algorithm 3.5.2, is a resampling method that partitions the cumulative distribution function (CDF) of the empirical joint filtering density into equal segments of size corresponding to the size of a fairly weighted particle (i.e. of size $1/N$), as illustrated in Figure 3.5.1(a) on page 82. As illustrated in Figure 3.5.1(b), one particle sample path is then drawn from each of the segments at equally spaced points on the CDF, the first draw in the sequence being uniformly chosen in the first segment to ensure inherent structure in the CDF isn't

missed.

Algorithm 3.5.2 Systematic Resampling Algorithm [Kitagawa, 1996].

1. Set $C := 0$. Set $j := 1$. Simulate $u \sim U[0, 1]$.
2. For i in 1 to N,
   (a) Set $C := C + w_{t-1}^{(i)}$.
   (b) While $\frac{u + j - 1}{N} \leq C$ set $\tilde{x}_{0:t-1}^{(j)} := x_{0:t-1}^{(i)}$ and set $j := j + 1$.

3.5.3 Stratified Resampling

Stratified Resampling, as introduced by Carpenter et al. [1999a] and detailed in Algorithm 3.5.3, is a similar resampling method to systematic resampling (see Section 3.5.2). In particular, the CDF of the empirical joint filtering density is similarly partitioned into equal segments of size $1/N$ (as illustrated in Figure 3.5.2(a) on page 83), however, one particle is drawn uniformly from each of the segments (as illustrated in Figure 3.5.2(b)).

Algorithm 3.5.3 Stratified Resampling Algorithm [Carpenter et al., 1999a].

1. Set $C := 0$ and $j := 1$.
2. For i in 1 to N simulate $u_i \sim U[0, 1]$ and set $B_i := \frac{u_i + i - 1}{N}$.
3. For i in 1 to N,
   (a) Set $C := C + w_{t-1}^{(i)}$.
   (b) While $B_i \leq C$ set $\tilde{x}_{0:t-1}^{(j)} := x_{0:t-1}^{(i)}$ and set $j := j + 1$.

3.5.4 Residual Resampling

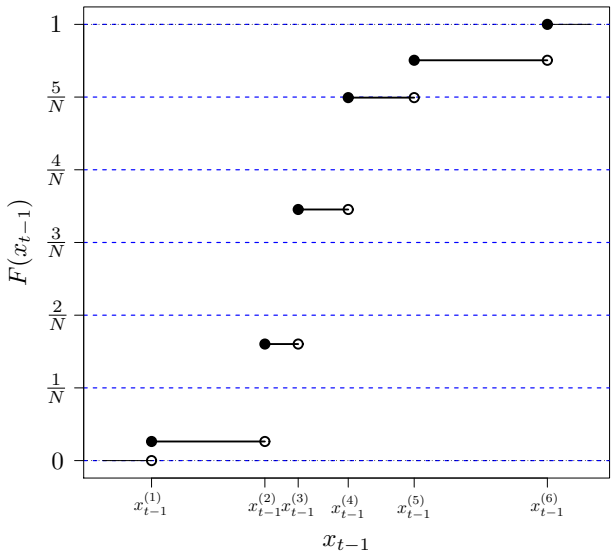
Residual Resampling, as proposed by [Higuchi, 1997; Liu and Chen, 1998] and detailed in Algorithm 3.5.4, is in effect a resampling scheme which is broken into a purely deterministic step and a step which introduces randomness. As illustrated in Figure 3.5.3 on page 84, to begin with any sample paths which have weighting greater than their fair share $(1/N)$ are resampled at least as many times as increments thereof deterministically.

In the second step, the remaining required draws $(N - R)$ are resampled using another resampling method from the sample paths with modified weights taking into account any selection in the first step.

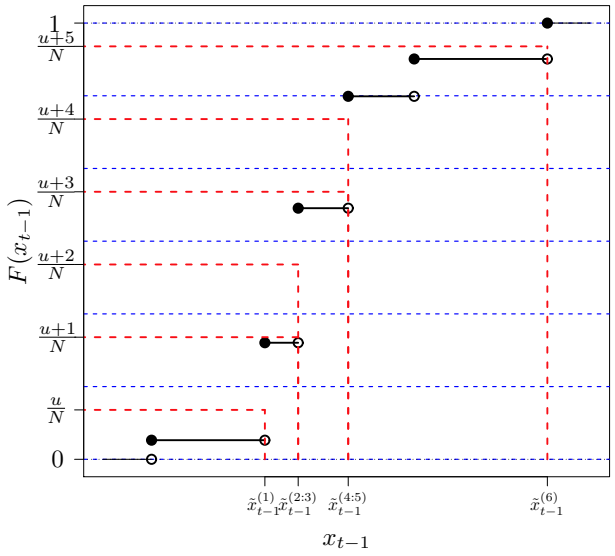
Algorithm 3.5.4 Residual Resampling Algorithm [Higuchi, 1997; Liu and Chen, 1998].

---

1. Set $j = 1$.
2. For i in 1 to N, if $w_{t-1}^{(i)} \geq 1/N$,
   (a) Set $\tilde{x}_{0:t-1}^{\left(j:\left\lfloor j+\left\lfloor N\cdot w_{t-1}^{(i)}\right\rfloor -1\right\rfloor\right)} := x_{0:t-1}^{(i)}$ and set $j := j + \left\lfloor N \cdot w_{t-1}^{(i)} \right\rfloor$.
   (b) $\tilde{w}_{t-1}^{(i)} := w_{t-1}^{(i)} - \left\lfloor N \cdot w_{t-1}^{(i)} \right\rfloor / N$.
3. For i in 1 to N set $\tilde{w}_{t-1}^{(i)} = \dfrac{\tilde{w}_{t-1}^{(i)}}{\sum_{k=1}^{N} \tilde{w}_{t-1}^{(k)}}$.
4. Set $r(\mathrm{d}x_{0:t-1}|y_{1:t-1}) := \sum_{i=1}^{N} \tilde{w}_{t-1}^{(i)} \cdot \delta_{x_{0:t-1}^{(i)}}(\mathrm{d}x_{0:t-1})$.
5. For i in j to N sample $\tilde{x}_{0:t-1}^{(i)} \sim r(\mathrm{d}x_{0:t-1}|y_{1:t-1})$ (as per Algorithm 3.5.1, 3.5.2 or 3.5.3).

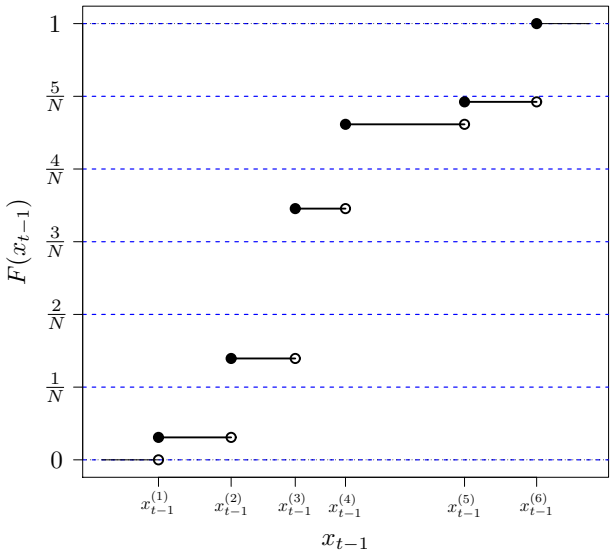


(a) Partitioning of the CDF.

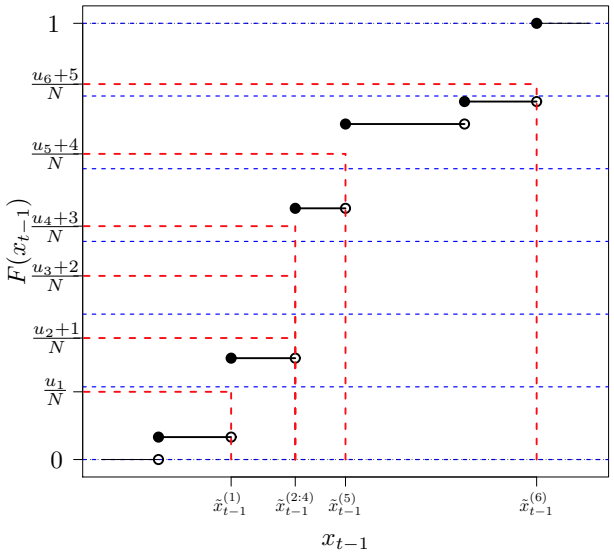


(b) Resampling of one sample from each partition of the CDF at equidistant intervals.

Figure 3.5.1: An illustration of Systematic Resampling.

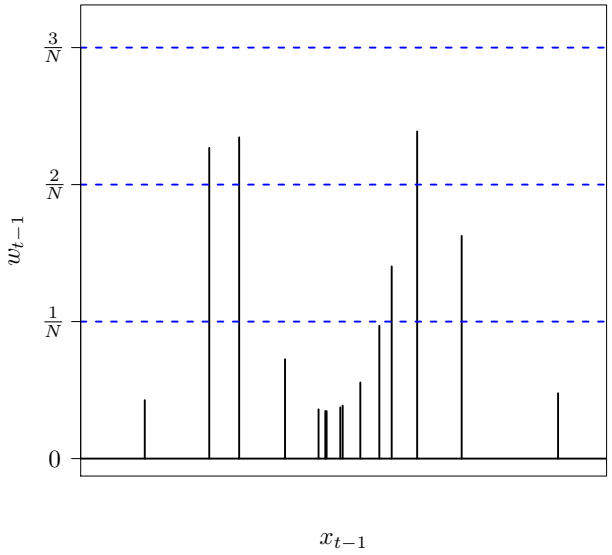


(a) Partitioning of the CDF.

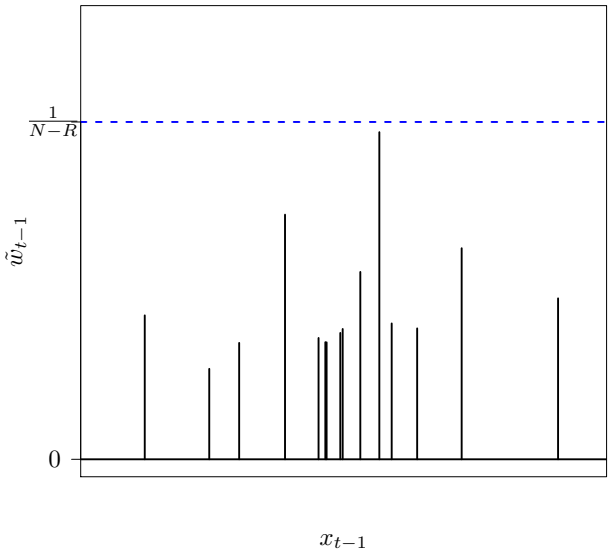


(b) Resampling of one sample uniformly from each partition of the CDF.

Figure 3.5.2: An illustration of Stratified Resampling.



(a) Before Residual Resampling.



(b) After the completion of the deterministic step in Residual Resampling (Algorithm 3.5.4 Step 3).

Figure 3.5.3: An illustration of Residual Resampling, where each vertical black line represents the location of a particle and its associated weight.

3.6 Auxiliary Particle Filter

Within the Sequential Importance Sampling / Resampling algorithm (Algorithm 3.4) we determine whether to resample the particle set (which forms the prior empirical joint filtering density) and if necessary resample at each iteration, before propagating and reweighting this particle set to the next time point to form an updated empirical joint filtering density (as motivated in Sections 3.4 and 3.5). As discussed in Sections 3.3 and 3.3.1, we would ideally propagate the particle set according to the optimal marginal importance function, however difficulties arise in evaluating explicitly the density $p_\theta(y_t|x_{t-1})$. As such this choice of marginal importance function is typically unavailable and instead the prior marginal importance function is typically chosen (as discussed in Section 3.3.2). The advantage in trying to select the optimal marginal importance function for propagating the particle set is that it incorporates the next observation – so the natural question that arises is if it is unavailable then is it instead possible to find an alternative to the SISR algorithm which incorporates the next observation.

The *Auxiliary Particle Filter* (APF) introduced by Pitt and Shephard [1999, 2001], so called as the original formulation was justified by the introduction of auxiliary variables, addresses this notion. In essence the APF attempts to selectively resample particle sample paths from the prior particle set which are consistent with the next observation. The APF has proven to be a popular alternative to the SISR algorithm, however, as shown in [Johansen and Doucet, 2008; Doucet and Johansen, 2011], it can be interpreted within the framework of the SISR algorithm. We introduce the APF within this framework in this section as doing so is consistent with the rest of this chapter and has the added benefit that existing analysis for the SIS and SISR algorithms can be directly extended to the APF (for example, convergence results which we do not address in this thesis but can be found in a number of texts such as Del Moral [2004]).

We begin by noting that although it is not typically possible to find $p_\theta(y_t|x_{t-1})$, we can find an approximation to this density,

$$\widehat{p_\theta}(y_t|x_{t-1}) \approx p_\theta(y_t|x_{t-1}) = \int f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)\,\mathrm{d}x_t. \tag{3.49}$$

Now, the APF can be thought of as an SISR algorithm targeting the density,

$$\widehat{p}_\theta(x_{0:t}|y_{1:t+1}) \propto \pi(x_{0:t}|y_{1:t}) \cdot \widehat{p}_\theta(y_{t+1}|x_t), \tag{3.50}$$

which by construction is an approximation of the density $p_\theta(x_{0:t}|y_{1:t+1})$. Choosing the decomposable joint importance function $q_\theta(x_{0:t}|y_{1:t})$ in order to construct our SISR algorithm (recalling from (3.20) on page 61 in Section 3.2 that $q_\theta(x_{0:t}|y_{1:t}) = q_\theta(x_{0:t-1}|y_{1:t-1}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t})$), note that if we want to evaluate expectations of the following form (where h is some test function),

$$\mathbb{E}_{\widehat{p}_\theta(x_{0:t}|y_{1:t+1})}\left[h(X_{0:t})\right] := \int h(x_{0:t}) \cdot \widehat{p}_\theta(x_{0:t}|y_{1:t+1}) \, \mathrm{d}x_{0:t}, \tag{3.51}$$

then, in the same manner as Section 3.2, we can re-express this expectation as an expectation with respect to our joint importance function $q_\theta(x_{0:t}|y_{1:t})$,

$$\begin{aligned}
\mathbb{E}_{\widehat{p}_\theta(x_{0:t}|y_{1:t+1})}\left[h(X_{0:t})\right] &= \int h(x_{0:t}) \cdot \frac{\widehat{p}_\theta(x_{0:t}|y_{1:t+1})}{q_\theta(x_{0:t}|y_{1:t})} \cdot q_\theta(x_{0:t}|y_{1:t}) \, \mathrm{d}x_{0:t} \\
&\propto \int h(x_{0:t}) \cdot \underbrace{\frac{\widehat{p}_\theta(x_{0:t-1}|y_{1:t})}{q_\theta(x_{0:t-1}|y_{1:t-1})} \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t) \cdot \widehat{p}_\theta(y_{t+1}|x_t)}{\widehat{p}_\theta(y_t|x_{t-1}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t})}}_{:= w_t^*(x_{0:t}) := w_{t-1}^*(x_{0:t-1}) \cdot w_{t|(t-1)}^*(x_{0:t})} \cdot q_\theta(x_{0:t}|y_{1:t}) \, \mathrm{d}x_{0:t} \qquad (3.52) \\
&= \mathbb{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[h(X_{0:t}) \cdot w_t^*(X_{0:t})\right]. \qquad (3.53)
\end{aligned}$$

Proceeding by the same inductive argument as presented in Section 3.2 we arrive directly at our desired SISR algorithm, which is detailed in Algorithm 3.6.1.

The particle set which arises from Algorithm 3.6.1 forms an empirical representation of the density $\widehat{p}_\theta(x_{0:t}|y_{1:t+1})$, however, we require an empirical representation of the joint filtering density $\pi_\theta(x_{0:t}|y_{1:t})$ instead. Noting that the joint filtering density decomposes as follows,

$$\pi_\theta(x_{0:t}|y_{1:t}) = p_\theta(x_{0:t-1}|y_{1:t}) \cdot p_\theta(x_t|x_{t-1}.y_t), \tag{3.54}$$

Algorithm 3.6.1 Sequential Importance Sampling / Resampling Algorithm for $\widehat{p}_\theta(x_{0:t}|y_{1:t+1})$ [Johansen and Doucet, 2008; Doucet and Johansen, 2011].

Initialisation Step ($t = 0$)**:-**

1. For i in 1 to N, simulate $x_0^{(i)} \sim \mu_\theta(x_0)$ and set $w_t^{*(i)} = \widehat{p}_\theta\left(y_1 \,\middle|\, x_0^{(i)}\right)$.
2. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
3. Set $\pi_\theta^N(\mathrm{d}x_0|y_1) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. If $\widehat{N_{\text{eff}}} \leq N_{\text{th}}$ then for i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Simulate $x_t^{(i)} \sim q_\theta\left(x_t \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)$ and set $x_{0:t}^{(i)} := \left\{x_{0:t-1}^{(i)}, x_t^{(i)}\right\}$.
   (b) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot \dfrac{f_\theta\left(x_t^{(i)} \,\middle|\, x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right) \cdot \widehat{p}_\theta\left(y_{t+1} \,\middle|\, x_t^{(i)}\right)}{\widehat{p}\left(y_t \,\middle|\, x_{t-1}^{(i)}\right) \cdot q_\theta\left(x_t^{(i)} \,\middle|\, x_{0:t-1}^{(i)}, y_{1:t}\right)}$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t+1}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.

then importance sampling can applied (see Section 2.5) where it is appropriate to use the following dominating density,

$$\widehat{q}_\theta(x_{0:t}|y_{0:t}) := \widehat{p}_\theta(x_{0:t-1}|y_{0:t}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t}). \tag{3.55}$$

In particular we have,

$$\begin{aligned}
\mathbb{E}_{\pi(x_{0:t}|y_{1:t})}\left[h(X_{0:t})\right] &= \mathbb{E}_{\widehat{q}_\theta(x_{0:t}|y_{0:t})}\left[h(X_{0:t}) \cdot \frac{\pi(x_{0:t}|y_{1:t})}{\widehat{q}_\theta(x_{0:t}|y_{0:t})}\right] \\
&\propto \mathbb{E}_{\widehat{q}_\theta(x_{0:t}|y_{0:t})}\left[h(X_{0:t}) \cdot \frac{\pi_\theta(x_{0:t-1}|y_{1:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{\widehat{p}_\theta(x_{0:t-1}|y_{0:t}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t})}\right] \\
&= \mathbb{E}_{\widehat{q}_\theta(x_{0:t}|y_{0:t})}\left[h(X_{0:t}) \cdot \frac{\pi_\theta(x_{0:t-1}|y_{1:t-1})}{\pi_\theta(x_{0:t-1}|y_{1:t-1})} \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{\widehat{p}_\theta(y_t|x_{t-1}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t})}\right] \\
&= \mathbb{E}_{\widehat{q}_\theta(x_{0:t}|y_{0:t})}\left[h(X_{0:t}) \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{\widehat{p}_\theta(y_t|x_{t-1}) \cdot q_\theta(x_t|x_{0:t-1}, y_{1:t})}\right]. \qquad (3.56)
\end{aligned}$$

As a consequence, at each iteration of Algorithm 3.6.1 we obtain an empirical representation of the density $\widehat{p}_\theta(x_{0:t-1}|y_{1:t})$, which can be used as part of the proposal in an importance sampler to construct an empirical representation of the joint filtering density $\pi_\theta(x_{0:t}|y_{1:t})$. Note that the proposal density used (3.55), is precisely what is obtained at each iteration after the propagation step in Algorithm 3.6.1 (Step 2a) but before the importance weighting step (Step 2b). The importance weight function in (3.56) has a form similar to the marginal importance weight function in Algorithm 3.6.1. As such, it is easy to construct our desired algorithm targeting the joint filtering density by iterating over each time step between constructing the empirical representation of the density $\widehat{p}_\theta(x_{0:t-1}|y_{1:t})$ and the target joint filtering density $\pi_\theta(x_{0:t}|y_{1:t})$. We present the synthesis of this argument in Algorithm 3.6.2. As the motivation for the APF was to avoid spending unnecessary computation propagating particles from the prior particle set which are not consistent with the next observation, it is typical within the APF literature to conduct a resampling step at every iteration, which is what we present in Algorithm 3.6.2.

We conclude this section by providing an illustrative example of an auxiliary particle filter in Figure 3.6.1.

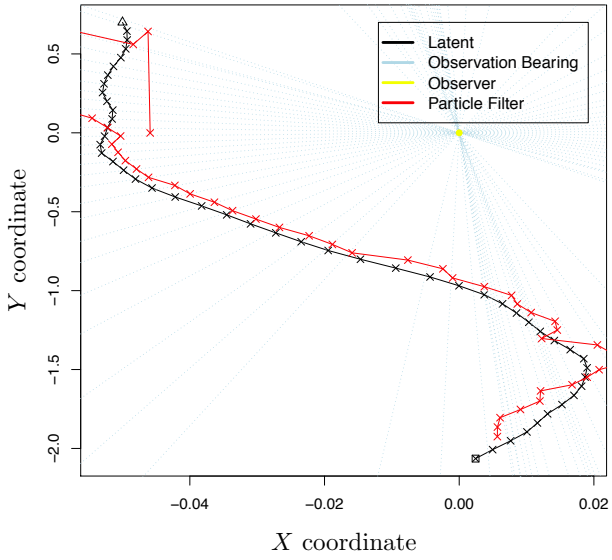
Algorithm 3.6.2 Auxiliary Particle Filter (APF) Algorithm [Pitt and Shephard, 1999, 2001; Johansen and Doucet, 2008; Doucet and Johansen, 2011].

Initialisation Step ($t = 0$)**:-**

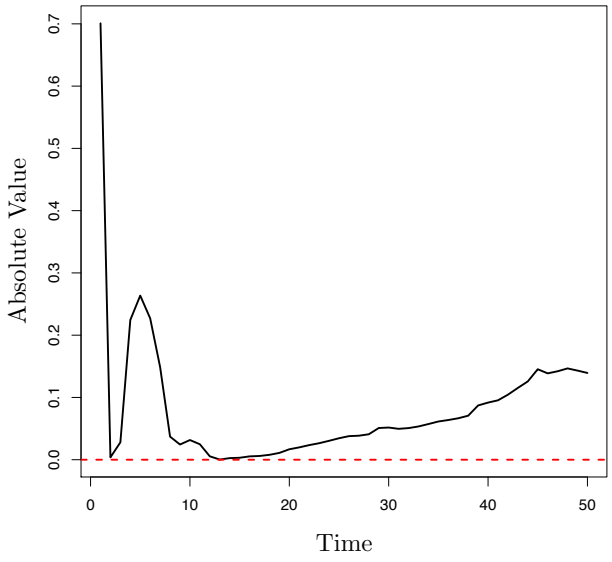
1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$, set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. For i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t}) \propto \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{0:t-1}) \cdot \widehat{p}_\theta(y_t|x_{t-1}) := \sum_{i=1}^N w_{t-1}^{(i)} \cdot \widehat{p}_\theta(y_t|x_{t-1}^{(i)}) \cdot \delta_{x_{0:t-1}^{(i)}}(\mathrm{d}x_{0:t-1})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Simulate $x_t^{(i)} \sim q_\theta\left(x_t^{(i)} \middle| x_{0:t-1}^{(i)}, y_{1:t}\right)$ and set $x_{0:t}^{(i)} := \left\{x_{0:t-1}^{(i)}, x_t^{(i)}\right\}$.
   (b) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot \dfrac{f_\theta\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right) \cdot g_\theta\left(y_t \middle| x_t^{(i)}\right)}{\widehat{p}_\theta\left(y_t \middle| x_{t-1}^{(i)}\right) \cdot q_\theta\left(x_t^{(i)} \middle| x_{0:t-1}^{(i)}, y_{1:t}\right)}$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.



(a) Observation process and particle filter, overlaid with the underlying latent process.



(b) Absolute distance between the particle filter estimate and the underlying latent process over time.

Figure 3.6.1: "Bearings-Only Tracking" example of Gordon et al. [1993] – An illustrative example of an auxiliary particle filter of 2000 particles in which we wish to track a four dimensional target $x_t = (z_x, \bar{z_x}, z_y, \bar{z_y})^T$ which moves on the x-y plane (where (z_x, z_y) denotes the targets position and $(\bar{z_x}, \bar{z_y})$ denotes the targets velocity) according to the following HMM SSDs, $X_0 \sim \text{MVN}(m_0, \mathbf{M}_0)$, $X_t|(X_{t-1} = x_{t-1}) \sim \text{MVN}(\mathbf{\Phi} x_{t-1}, \mathbf{\Gamma V}_t)$ and $Y_t|(X_t = x_t) \sim \text{N}(\tan^{-1}(z_y/z_x), \mathbf{W}_t)$ (where $m_0, \mathbf{M}_0, \mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{V}$ and $\mathbf{W}$ are parameterised as in Gordon et al. [1993]).

4 An Introduction to Simulating Diffusions and Jump Diffusions

"I turn with terror and horror from this lamentable scourge of continuous functions with no derivatives."

— Charles Hermite

To recap from Chapter 1, a jump diffusion $V : \mathbb{R} \to \mathbb{R}$ is a Markov process, which in this thesis we define to be the solution to a stochastic differential equation (SDE) of the following form (denoting $V_{t-} := \lim_{s \uparrow t} V_s$),

$$\mathrm{d}V_t = \beta(V_{t\text{-}})\,\mathrm{d}t + \sigma(V_{t\text{-}})\,\mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\mu}, \qquad V_0 = v \in \mathbb{R}, \;\; t \in [0, T], \tag{4.1}$$

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 (see Section 2.8) and $J_t^{\lambda,\mu}$ denotes a compound Poisson process (see Section 2.9). $J_t^{\lambda,\mu}$ 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. We briefly note that the notation $J_t^{\lambda,\mu}$ to denote a compound Poisson process is a minor abuse of standard notation in this field, however it is equivalent and is more suited to our focus on simulation. Further note that all coefficients of (4.1) are themselves typically dependent on V_t. In essence the SDE above describes the instantaneous behaviour of the process and can be interpreted in integrated form,

$$V_T = V_0 + \int_0^T \beta(V_{t\text{-}})\,\mathrm{d}t + \int_0^T \sigma(V_{t\text{-}})\,\mathrm{d}W_t + \int_0^T \mathrm{d}J_t^{\lambda,\mu}. \tag{4.2}$$

As discussed in Chapter 1, regularity conditions on the coefficients of (4.1) are assumed to hold to ensure the existence of a unique non-explosive weak solution (see for instance [Øksendal and Sulem, 2004, Chap. 1] and [Platen and Bruti-Liberati, 2010, Chap. 1.9]).

A discussion of conditions sufficient to allow the application of the methodology presented within this thesis were given in Section 1.3.

The objective of this chapter is two-fold. In the first half of this chapter (Section 4.1) we provide a brief introductory level overview of elements of stochastic calculus pertinent to this thesis. In the second half of this chapter (Section 4.2) we briefly review existing discretisation methods for simulating sample paths from the measure induced by V, denoted by $\mathbb{T}^v$.

4.1 Stochastic Calculus Preliminaries

In this section we outline some material which is central to the development of the methodology in this thesis. In particular: in Section 4.1.1 we introduce and motivate the Itô integral and Itô calculus, in Section 4.1.2 we introduce the Lamperti transform, in Section 4.1.3 we provide a informal proof of Girsanov's theorem and the Radon-Nikodým derivative for a class of diffusions and jump diffusions, and finally, in Section 4.1.4 we find a representation of the transition density for this class of diffusions and jump diffusions.

4.1.1 The Itô Integral & Itô's Formulae

Motivated by the integrated form of our target SDE (4.2), we would like to find a definition for integrals of the following form,

$$\int_0^T f(t)\,\mathrm{d}W_t, \tag{4.3}$$

where f is some suitable function which may be additionally parameterised by some other stochastic process (for instance, a Brownian motion sample path) and $W \sim \mathbb{W}_{0,T}^0$ (as we introduced in Section 2.8). Of course, in order to define such an object we must first prove its existence, which is what we provide an informal outline of in this section (in the style of [Øksendal, 2007, Chap. 3]).

In general to evaluate an integral with respect to a process H we can partition the interval $[0, T]$ into a fine mesh (i.e. divide it into N sub-intervals of length T/N and denote by $t_i = t_i^{(N)} := iT/N$) then, provided H is of bounded variation on compact time intervals

(which as shown in Banach and Steinhaus [1927] is a necessary condition),

$$V_{[0,T]}(H) := \lim_{N\to\infty} \sum_{i=1}^{N} \left| H(t_i) - H(t_{i-1}) \right| < \infty, \tag{4.4}$$

we can define our integral in the Riemann-Stieljes sense,

$$\int_0^T f(t)\,\mathrm{d}H(t) := \lim_{N\to\infty} \sum_{i=1}^{N} f(s_i)\cdot[H(t_i) - H(t_{i-1})], \quad \text{where } s_i \in [t_{i-1}, t_i]. \tag{4.5}$$

However, Brownian motion has infinite variation (w.p. 1: $V_{[0,T]}(W) = \infty$) and so the definition in (4.5) isn't suitable. In particular, selecting different evaluation points in the partition when evaluating the function f in the integral (4.5) produces different answers. For instance, recalling the Brownian motion properties from Section 2.8 and considering the expectation of (4.5) with respect to the measure $\mathbb{W}^0_{0,T}$ in the case where $f(t) := W_t$ and $s_i := t_{i-1}$, then we have,

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N} W_{t_{i-1}}\cdot[W_{t_i} - W_{t_{i-1}}]\right] &= \sum_{i=1}^{N} \mathbb{E}\left[W_{t_{i-1}}\cdot[W_{t_i} - W_{t_{i-1}}]\right] \\ &= \sum_{i=1}^{N} \mathbb{E}\left[W_{t_{i-1}}\right]\cdot\mathbb{E}\left[W_{t_i} - W_{t_{i-1}}\right] = 0\end{aligned} \tag{4.6}$$

whereas, if we instead consider the evaluation point $s_i := t_i$ we have,

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N} W_{t_i}\cdot[W_{t_i} - W_{t_{i-1}}]\right] &= \sum_{i=1}^{N} \mathbb{E}\left[W_{t_i}^2 - W_{t_{i-1}}\cdot W_{t_i}\right] \\ &= \sum_{i=1}^{N} (t_i - t_{i-1}) = T.\end{aligned} \tag{4.7}$$

The approximation in (4.6) to (4.5) leads to the Itô integral (which we consider in the remainder of this section), whereas selecting the evaluation point $s_i := (t_{i-1} + t_i)/2$ and approximating (4.5) as above leads to the Stratonovich integral (which we do not cover in this thesis). The approximation in (4.6) is intuitively more appealing for our purposes as it is not only non-anticipative, but the independence of the increments of Brownian motion can be exploited (see Brownian motion Property 2). The key to finding an appropriate definition of (4.3) is to note that although Brownian motion has infinite variation, the

quadratic variation of Brownian motion is finite[1],

$$\text{w.p. 1: } [W]_T := \lim_{N\to\infty} \sum_{i=1}^{N} (W_{t_i} - W_{t_{i-1}})^2 = T. \tag{4.9}$$

In particular, following the approach taken in [Øksendal, 2007, Chap. 3], we construct the Itô integral by first considering the integration of simple processes of the following form (where the e_i are $\mathcal{F}_{t_{i-1}}$-measurable random variables),

$$Y_N(t) = \sum_{i=1}^{N} e_i \cdot \mathbb{1}\left[t \in [t_{i-1}, t_i)\right], \tag{4.10}$$

and defining $\int_0^T Y_N(t)\,\mathrm{d}W_t := \sum_{i=1}^N e_i \cdot [W_{t_i} - W_{t_{i-1}}]$. With this definition we can establish the following identity,

$$\begin{aligned}
\mathbb{E}\left[\left(\int_0^T Y_N(t)\,\mathrm{d}W_t\right)^2\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{N} e_i \cdot [W_{t_i} - W_{t_{i-1}}]\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} e_i^2 \cdot [W_{t_i} - W_{t_{i-1}}]^2 + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} e_i \cdot [W_{t_i} - W_{t_{i-1}}] \cdot e_j \cdot \left[W_{t_j} - W_{t_{j-1}}\right]\right] \\
&= \sum_{i=1}^{N} \mathbb{E}\left[e_i^2\right] \cdot \mathbb{E}\left[[W_{t_i} - W_{t_{i-1}}]^2\right] = \sum_{i=1}^{N} \mathbb{E}\left[e_i^2\right] \cdot (t_i - t_{i-1}) \\
&= \mathbb{E}\left[\int_0^T Y_N^2(t)\,\mathrm{d}t\right].
\end{aligned} \tag{4.11}$$

As discussed in [Øksendal, 2007, Chap. 3], it transpires that any continuous adapted process $f(t)$ with $\mathbb{E}\left[\int_0^T f^2(t)\,\mathrm{d}t\right] < \infty$ can be approximated by a sequence of simple processes in the sense that there exists a sequence of simple processes $\{Y_N : N \in \mathbb{N}\}$, each defined as in (4.10) such that,

$$\text{w.p. 1: } \lim_{N\to\infty} \mathbb{E}\left[\int_0^T \left|f(t) - Y_N(t)\right| \mathrm{d}t\right] = 0, \tag{4.12}$$

[1] In general considering the variation of Brownian motion over the time interval $[0, T]$ we have,

$$\text{w.p. 1: } \lim_{N\to\infty} \sum_{i=1}^{N} \left|W_{t_i} - W_{t_{i-1}}\right|^p = \begin{cases} \infty & \text{if } p = 1 \\ T & \text{if } p = 2 \\ 0 & \text{if } p \geq 3 \end{cases} \tag{4.8}$$

and so together with (4.11) we can conclude that there exits a limit to $\int_0^T Y_N(t)\,\mathrm{d}W_t$ (which we denote by $I(T)$) such that,

$$\text{w.p. 1: } \lim_{N\to\infty} \mathbb{E}\left[\left|\int_0^T Y_N(t)\,\mathrm{d}W_t - I(T)\right|^2\right] = 0. \tag{4.13}$$

We consequently define our integral in (4.3) to be this limit,

$$\int_0^T f(t)\,\mathrm{d}W_t := I(T) = \lim_{N\to\infty} \int_0^T Y_N(t)\,\mathrm{d}W_t, \tag{4.14}$$

additionally noting that (4.11) still holds for this limit (which is a property known as *Itô's isometry*),

$$\mathbb{E}\left[\left(\int_0^T f(t)\,\mathrm{d}W_t\right)^2\right] = \mathbb{E}\left[\int_0^T f^2(t)\,\mathrm{d}t\right]. \tag{4.15}$$

Now, denoting by F as the anti-derivative of f and supposing $F \in C^2$, then by considering a Taylor series expansion of F and recalling footnote 1 we arrive at a result known as *Itô's formula*,

$$\begin{aligned} F(T) &= F(0) + \lim_{N\to\infty}\left[\sum_{i=1}^N f(t_i)\cdot[W_{t_i} - W_{t_{i-1}}] + \frac{1}{2!}\sum_{i=1}^N f'(t_i)\cdot[W_{t_i} - W_{t_{i-1}}]^2 \right. \\ &\qquad \left. + \frac{1}{3!}\sum_{i=1}^N f''(t_i)\cdot[W_{t_i} - W_{t_{i-1}}]^3 + \cdots\right] \\ &= F(0) + \int_0^T f(t)\,\mathrm{d}W_t + \frac{1}{2}\int_0^T f'(t)\,\mathrm{d}t + 0, \end{aligned} \tag{4.16}$$

or alternatively in (the more usual) differential form we have,

$$\mathrm{d}\left(F(t)\right) = \frac{1}{2}f'(t)\,\mathrm{d}t + f(t)\,\mathrm{d}W_t. \tag{4.17}$$

Considering more generally diffusions of the following form (where $\mathbb{E}\left[\int_0^T \sigma^2(V_t)\,\mathrm{d}t\right] < \infty$),

$$\mathrm{d}V_t = \beta(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t, \tag{4.18}$$

then we are often interested in finding the differential form of $X_t := g(t, V_t)$ where $g \in C^2$. Again, considering a Taylor series expansion of X we find (where the derivatives of g are evaluated at the points (t_i, V_{t_i})),

$$X_T = X_0 + \lim_{N\to\infty}\left[\sum_{i=1}^{N}\frac{\partial g}{\partial t}\cdot(t_i - t_{i-1}) + \sum_{i=1}^{N}\frac{\partial g}{\partial v}\cdot(V_{t_i} - V_{t_{i-1}}) + \frac{1}{2}\sum_{i=1}^{N}\frac{\partial^2 g}{\partial t^2}\cdot(t_i - t_{i-1})^2 + \sum_{i=1}^{N}\frac{\partial^2 g}{\partial t\partial v}\cdot(t_i - t_{i-1})\cdot(V_{t_i} - V_{t_{i-1}}) + \frac{1}{2}\sum_{i=1}^{N}\frac{\partial^2 g}{\partial v^2}\cdot(V_{t_i} - V_{t_{i-1}})^2 + \cdots\right]. \tag{4.19}$$

Noting that over small time intervals of length $h > 0$ that $V_{t+h} - V_t \approx \beta(V_t)\cdot((t+h) - t) + \sigma(V_t)\cdot(W_{t+h} - W_t)$, we have,

$$\begin{aligned}
&\lim_{N\to\infty}\left[\sum_{i=1}^{N}\frac{\partial g}{\partial v}\cdot(V_{t_i} - V_{t_{i-1}})\right] \\
&\qquad = \lim_{N\to\infty}\left[\sum_{i=1}^{N}\frac{\partial g}{\partial v}\cdot\beta(V_{t_i})\cdot(t_i - t_{i-1}) + \sum_{i=1}^{N}\frac{\partial g}{\partial v}\cdot\sigma(V_{t_i})\cdot(W_{t_i} - W_{t_{i-1}})\right] \\
&\qquad = \int_0^T \frac{\partial g(t, V_t)}{\partial v}\cdot\beta(V_t)\,\mathrm{d}t + \int_0^T \frac{\partial g(t, V_t)}{\partial v}\cdot\sigma(V_t)\,\mathrm{d}W_t, \qquad (4.20) \\
&\lim_{N\to\infty}\left[\sum_{i=1}^{N}\frac{\partial^2 g}{\partial v^2}\cdot(V_{t_i} - V_{t_{i-1}})^2\right] \\
&\qquad = \lim_{N\to\infty}\left[\sum_{i=1}^{N}\frac{\partial^2 g}{\partial v^2}\cdot\beta^2(V_{t_i})\cdot(t_i - t_{i-1})^2 + \sum_{i=1}^{N}\frac{\partial^2 g}{\partial v^2}\cdot\sigma^2(V_{t_i})\cdot(W_{t_i} - W_{t_{i-1}})^2 \right. \\
&\qquad\qquad \left. +2\sum_{i=1}^{N}\frac{\partial^2 g}{\partial v^2}\cdot\beta(V_{t_i})\cdot\sigma(V_{t_i})\cdot(t_i - t_{i-1})\cdot(W_{t_i} - W_{t_{i-1}})\right] \\
&\qquad = 0 + \int_0^T \frac{\partial^2 g(t, V_t)}{\partial v^2}\cdot\sigma^2(V_t)\,\mathrm{d}t + 0, \qquad (4.21)
\end{aligned}$$

and so by substituting (4.20) and (4.21) into (4.19), we arrive at a result known as *Itô's formula for diffusions*,

$$X_T = X_0 + \int_0^T\left(\frac{\partial g(t, V_t)}{\partial t} + \frac{\partial g(t, V_t)}{\partial v}\beta(V_t) + \frac{1}{2}\frac{\partial^2 g(t, V_t)}{\partial v^2}\sigma^2(V_t)\right)\mathrm{d}t + \int_0^T \frac{\partial g(t, V_t)}{\partial v}\cdot\sigma(V_t)\,\mathrm{d}W_t, \tag{4.22}$$

which in its more common differential form is as follows,

$$\begin{aligned} \mathrm{d}X_t = &\left(\frac{\partial g(t, V_t)}{\partial t} + \frac{\partial g(t, V_t)}{\partial v} \beta(V_t) + \frac{1}{2} \frac{\partial^2 g(t, V_t)}{\partial v^2} \sigma^2(V_t) \right) \mathrm{d}t \\ &+ \frac{\partial g(t, V_t)}{\partial v} \cdot \sigma(V_t) \, \mathrm{d}W_t. \end{aligned} \tag{4.23}$$

Finally, considering jump diffusions of the following form (where again $\mathbb{E}\left[\int_0^T \sigma^2(V_t) \, \mathrm{d}t\right] < \infty$),

$$\mathrm{d}V_t = \underbrace{\beta(V_{t\text{-}}) \, \mathrm{d}t + \sigma(V_{t\text{-}}) \, \mathrm{d}W_t}_{\mathrm{d}V_t^{\mathrm{cts}}} + \mathrm{d}J_t^{\lambda,\mu}, \tag{4.24}$$

then the differential form of $X_t := g(t, V_t)$ (where again $g \in C^2$) is given by *Itô's formula for jump diffusions* (see for instance [Kloeden and Platen, 1992, Chap. 1.5]). The key observation in this case is that jump diffusion sample paths can be decomposed into a continuous diffusion component ($\mathrm{d}V_t^{\mathrm{cts}}$) and a compound Poisson process. Informally, if we solely consider the compound Poisson process we have (denoting by $N_t := \sum_{i \geq 1} \mathbb{1}\{\psi_i \leq t\}$ as a Poisson jump counting process with respect to $\mathcal{F}_t$),

$$\mathrm{d}J_t^{\lambda,\mu} = \left[(V_{t\text{-}} + \mu(V_{t\text{-}})) - V_{t\text{-}} \right] \mathrm{d}N_t, \tag{4.25}$$

and so considering the continuous component (4.23) together with compound Poisson process component (4.25) we arrive at the following result,

$$\begin{aligned} \mathrm{d}X_t = &\left(\frac{\partial g(t, V_{t\text{-}})}{\partial t} + \frac{\partial g(t, V_{t\text{-}})}{\partial v} \beta(V_{t\text{-}}) + \frac{1}{2} \frac{\partial^2 g(t, V_{t\text{-}})}{\partial v^2} \sigma^2(V_{t\text{-}}) \right) \mathrm{d}t \\ &+ \frac{\partial g(t, V_{t\text{-}})}{\partial v} \cdot \sigma(V_{t\text{-}}) \, \mathrm{d}W_t + \left[g\left(V_{t\text{-}} + \mu(V_{t\text{-}})\right) - X_{t\text{-}} \right] \mathrm{d}N_t. \end{aligned} \tag{4.26}$$

4.1.2 Lamperti Transformation

As briefly discussed in Chapter 1, throughout this thesis we develop methodology for simulating diffusions and jump diffusions with unit volatility. However, we are interested in the broader class of diffusions and jump diffusions with non-unit volatility (see (4.1)). In order to consider this broader class we first *transform* the target diffusion (4.1) into one with unit volatility and then, after applying the methodology developed in this thesis, project any simulations by the inverse of this transformation.

To perform this transformation we exploit Itô's formula for jump diffusions. In particular note that from (4.26) we can deduce that if we define $X_t := g(t, V_t)$ then the differential form of X_t will only have unit volatility if $\partial g(t, V_t)/\partial v = 1/\sigma(V_t)$.

This particular transformation is known as the *Lamperti transform* (see for instance [Kloeden and Platen, 1992, Chap. 4.4], Casella and Roberts [2010] and Giesecke and Smelov [Forthcoming] for a more extensive discussion). In particular, denoting by $X_t := \eta(V_t)$ as the transformed process, where $\eta(V_t) := \int_{v^*}^{V_t} 1/\sigma(u)\,\mathrm{d}u$ (v^* is an arbitrary element in the state space of V), assuming η is invertible and denoting by η^{-1} as the inverse of the function η, and applying Itô's formula for jump diffusions (4.26) to find $\mathrm{d}X_t$ we have,

$$\begin{aligned}
\mathrm{d}X_t &= \left(\frac{\partial \eta(t,\eta^{-1}(X_{t\text{-}}))}{\partial t} + \frac{\partial \eta(t,\eta^{-1}(X_{t\text{-}}))}{\partial v}\beta(\eta^{-1}(X_{t\text{-}})) + \frac{1}{2}\frac{\partial^2 \eta(t,\eta^{-1}(X_{t\text{-}}))}{\partial v^2}\sigma^2(\eta^{-1}(X_{t\text{-}}))\right)\mathrm{d}t \\
&\quad + \frac{\partial \eta(t,\eta^{-1}(X_{t\text{-}}))}{\partial v}\cdot\sigma(\eta^{-1}(X_{t\text{-}}))\,\mathrm{d}W_t + \left[\eta\left(\eta^{-1}(X_{t\text{-}}) + \mu(\eta^{-1}(X_{t\text{-}}))\right) - X_{t\text{-}}\right]\mathrm{d}N_t \\
&= \underbrace{\left[\frac{\beta(\eta^{-1}(X_{t\text{-}}))}{\sigma(\eta^{-1}(X_{t\text{-}}))} - \frac{\sigma'(\eta^{-1}(X_{t\text{-}}))}{2}\right]}_{\alpha(X_{t\text{-}})}\mathrm{d}t + \mathrm{d}W_t + \underbrace{\left[\eta\left(\eta^{-1}(X_{t\text{-}}) + \mu(\eta^{-1}(X_{t\text{-}}))\right) - X_{t\text{-}}\right]\mathrm{d}N_t}_{\mathrm{d}J_t^{\lambda,\nu}} \qquad (4.27)\\
&=: \alpha(X_{t\text{-}})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu} \qquad (4.28)
\end{aligned}$$

This transformation is typically possible for univariate diffusions and for many multivariate diffusions (see Aït-Sahalia [2008]), however, as discussed in Section 1.3 and in particular Result 1, under the conditions assumed in this thesis it is possible to apply the Lamperti transformation to any target diffusion of interest.

4.1.3 Girsanov's Theorem

In this section we state and then provide a informal derivation of Girsanov's theorem for a specific class of jump diffusions. Recall that the methodology we develop in this thesis assumes Conditions 1 – 5 in Section 1.3 hold and so, as discussed briefly in Section 1.3, Conditions 2 and 3 are sufficient to allow us to transform our SDE in (4.1) into one with unit volatility (as per Section 4.1.2). As such, in this section we can restrict our attention

to the following class of jump diffusions with unit volatility,

$$\mathrm{d}X_t = \alpha(X_{t\text{-}})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \qquad X_0 = x \in \mathbb{R}, \;\; t \in [0,T]. \tag{4.29}$$

Denoting by $\mathbb{Q}_{0,T}^x$ the measure induced by (4.29), in this thesis we are interested in the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^x$ with respect to $\mathbb{W}_{0,T}^x$ (as discussed in Chapter 1), where $\mathbb{W}_{0,T}^x$ is the measure induced by a driftless jump diffusion with unit volatility,

$$\mathrm{d}X_t = \mathrm{d}W_t + \mathrm{d}J_t^{\Lambda,\delta}, \qquad X_0 = x \in \mathbb{R}, \;\; t \in [0,T], \tag{4.30}$$

where $J_t^{\Lambda,\delta}$ is a compound Poisson process parameterised with constant finite jump intensity $\Lambda > 0$, jump size coefficient $\delta : \mathbb{R} \to \mathbb{R}$ and with jumps distributed with density f_δ. We further assume that $\exists\, M < \infty$ such that $f_\nu(x)/f_\delta(x) \leq M \;\forall x \in \mathbb{R}$. Together with the fact that (4.29) and (4.30) both have unit volatility coefficient this ensures that $\mathbb{Q}_{0,T}^x$ is absolutely continuous with respect to $\mathbb{W}_{0,T}^x$.

Recalling that N_T denotes the number of jumps in the interval $[0,T]$, $\psi_1, \ldots, \psi_{N_T}$ denotes the jump times (where we set $\psi_0 := 0$ and $\psi_{N_T+1-} := \psi_{N_T+1} := T$), $\phi(X_s) := \alpha^2(X_s)/2 + \alpha'(X_s)/2$ and further introducing the notation $A(u) := \int_0^u \alpha(y)\,\mathrm{d}y$, we arrive at the following theorem.

Theorem 4.1.1 (Radon-Nikodým derivative (see Øksendal and Sulem [2004] and Platen and Bruti-Liberati [2010])**.** *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 as follows,*

$$\begin{aligned}\frac{\mathrm{d}\mathbb{Q}_{0,T}^x}{\mathrm{d}\mathbb{W}_{0,T}^x}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_{s-})\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\} \\ \cdot \exp\left\{-\int_0^T \left[\lambda(X_{s-}) - \Lambda\right]\,\mathrm{d}s\right\} \cdot \prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right)}{\Lambda \cdot f_\delta\left(X_{\psi_i}; X_{\psi_i-}\right)}\end{aligned} \tag{4.31}$$

Proof. (Sketch). We begin by recalling that by construction $\mathbb{Q}_{0,T}^x$ is absolutely continuous with respect to $\mathbb{W}_{0,T}^x$. Now, the informal derivation will proceed as follows: Firstly, we consider the behaviour under the two measures of a sample path X over small time intervals; Next, we employ a mesh based argument to consider the approximate behaviour of a sample path over the interval $[0,T]$ under the two measures; Finally, we consider the desired change of measure and the limit as the mesh interval size decreases to zero.

Now, considering the approximate behaviour under $\mathbb{Q}^x_{0,T}$ and $\mathbb{W}^x_{0,T}$ of a sample path X over small time intervals, we begin by noting that the underlying SDEs inducing both measures ((4.29) and (4.30) respectively) have finite jump intensity. As such, in any given finite interval there can only (almost surely) ever occur a finite number of jumps ($N_T < \infty$), and so there will exist some $h > 0$ such that in any given interval $[t, t+h]$ there will either be a single jump (with probability $\lambda(X_{t\text{-}}) \cdot h$) or no jump (with probability $1-\lambda(X_{t\text{-}})\cdot h$). If within the interval there is a jump then any change in X in that interval will be dominated by the jump and the contribution to the change from the continuous component will be negligible. As such, denoting by $\mathbb{1}_{[t,t+h]} := \mathbb{1}\left([t, t+h] \bigcap \{\psi_1, \ldots, \psi_{N_T}\} \neq \emptyset\right)$ as indicating the occurrence of a jump in the interval $[t, t+h]$, we have,

$$\mathbb{Q}^x_{0,T}(X_{t+h} \in \mathrm{d}y \,|\, X_t) \approx \left(\frac{1}{\sqrt{2\pi h}} \exp\left\{ -\frac{(y - X_t - \alpha(X_t)\cdot h)^2}{2h} \right\} \cdot (1 - \lambda(X_t)\cdot h) \cdot (1 - \mathbb{1}_{[t,t+h]}) \right.$$
$$\left. + (\lambda(X_t)\cdot h) \cdot f_\nu(y; X_t) \cdot \mathbb{1}_{[t,t+h]} \right) \mathrm{d}y, \tag{4.32}$$

$$\mathbb{W}^x_{0,T}(X_{t+h} \in \mathrm{d}y \,|\, X_t) \approx \left(\frac{1}{\sqrt{2\pi h}} \exp\left\{ -\frac{(y - X_t)^2}{2h} \right\} \cdot (1 - \Lambda \cdot h) \cdot (1 - \mathbb{1}_{[t,t+h]}) \right.$$
$$\left. + (\Lambda \cdot h) \cdot f_\delta(y; X_t) \cdot \mathbb{1}_{[t,t+h]} \right) \mathrm{d}y. \tag{4.33}$$

The approximate behaviour of the sample path X over the interval $[0, T]$ under $\mathbb{Q}^x_{0,T}$ and $\mathbb{W}^x_{0,T}$ can be considered in the same manner by first partitioning $[0, T]$ into T/h intervals

of length h and applying the Markov property,

$$\mathbb{Q}^x_{0,T}\left(X_{T/h} \in \mathrm{d}y_1, X_{2T/h} \in \mathrm{d}y_2 \ldots, X_T \in \mathrm{d}y_{T/h} \,|\, X_0\right) = \prod_{k=1}^{T/h} \mathbb{Q}^x_{0,T}\left(X_{kh} \in \mathrm{d}y_k \,|\, X_{(k-1)h}\right)$$

$$\approx \prod_{k=1}^{T/h} \Bigg(\frac{1}{\sqrt{2\pi h}} \exp\left\{ -\frac{\left(y_k - X_{(k-1)h} - \alpha\left(X_{(k-1)h}\right)\cdot h\right)^2}{2h} \right\} \cdot \left(1 - \lambda\left(X_{(k-1)h}\right)\cdot h\right) \cdot \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right)$$

$$+ \left(\lambda\left(X_{(k-1)h}\right)\cdot h\right) \cdot f_\nu\left(y_k; X_{(k-1)h}\right) \cdot \mathbb{1}_{[(k-1)h,kh]} \Bigg) \mathrm{d}y_k, \tag{4.34}$$

$$\mathbb{W}^x_{0,T}\left(X_{T/h} \in \mathrm{d}y_1, X_{2T/h} \in \mathrm{d}y_2 \ldots, X_T \in \mathrm{d}y_{T/h} \,|\, X_0\right) = \prod_{k=1}^{T/h} \mathbb{W}^x_{0,T}\left(X_{kh} \in \mathrm{d}y_k \,|\, X_{(k-1)h}\right)$$

$$\approx \prod_{k=1}^{T/h} \Bigg(\frac{1}{\sqrt{2\pi h}} \exp\left\{ -\frac{\left(y_k - X_{(k-1)h}\right)^2}{2h} \right\} \cdot \left(1 - \Lambda\cdot h\right) \cdot \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right)$$

$$+ \left(\Lambda\cdot h\right) \cdot f_\delta\left(y_k; X_{(k-1)h}\right) \cdot \mathbb{1}_{[(k-1)h,kh]} \Bigg) \mathrm{d}y_k. \tag{4.35}$$

Now, considering the change of measure we have,

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}\left(X_{T/h}, X_{2T/h}, \ldots, X_T \,|\, X_0\right) \approx \frac{\mathbb{Q}^x_{0,T}\left(X_{T/h} \in \mathrm{d}y_1, X_{2T/h} \in \mathrm{d}y_2 \ldots, X_T \in \mathrm{d}y_{T/h} \,|\, X_0\right)}{\mathbb{W}^x_{0,T}\left(X_{T/h} \in \mathrm{d}y_1, X_{2T/h} \in \mathrm{d}y_2 \ldots, X_T \in \mathrm{d}y_{T/h} \,|\, X_0\right)}$$

$$\approx \prod_{k=1}^{T/h} \Bigg[\exp\left\{ \frac{\left(y_k - X_{(k-1)h}\right)^2 - \left(y_k - X_{(k-1)h} - \alpha\left(X_{(k-1)h}\right)h\right)^2}{2h} \right\} \cdot \frac{1 - \lambda\left(X_{(k-1)h}\right)h}{1 - \Lambda h} \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right)$$

$$+ \frac{\lambda\left(X_{(k-1)h}\right)\cdot f_\nu\left(y_k; X_{(k-1)h}\right)}{\Lambda\cdot f_\delta\left(y_k; X_{(k-1)h}\right)} \mathbb{1}_{[(k-1)h,kh]} \Bigg]$$

$$= \exp\left\{ \sum_{k=1}^{T/h} \left[\alpha\left(X_{(k-1)h}\right)\cdot\left(y_k - X_{(k-1)h}\right) - \frac{1}{2}\alpha^2\left(X_{(k-1)h}\right)\cdot h \right] \cdot \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right) \right\}$$

$$\cdot \exp\left\{ \sum_{k=1}^{T/h} \left[\log\left(1 - \lambda(X_{(k-1)h})\cdot h\right) - \log\left(1 - \Lambda\cdot h\right)\right] \cdot \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right) \right\}$$

$$\cdot \prod_{k=1}^{T/h} \left[\frac{\lambda\left(X_{(k-1)h}\right)\cdot f_\nu\left(y_k; X_{(k-1)h}\right)}{\Lambda\cdot f_\delta\left(y_k; X_{(k-1)h}\right)} \cdot \mathbb{1}_{[(k-1)h,kh]} + \left(1 - \mathbb{1}_{[(k-1)h,kh]}\right) \right]. \tag{4.36}$$

Applying a MacLaurin series expansion ($f(x) = \sum_{0\cup\mathbb{N}} f^i(0)x^i/i!$) to the second exponential term in (4.36), recalling that $\log(1-x) = \sum_{i=1}^{\infty} -x^i/i!$ and considering this result as

$h \to 0$ noting that $\lim_{h\to 0}\left[\bigcup_{i=1}^{T/h}[(k-1)h, kh]\right] = [0,T]$, we have,

$$\begin{aligned}
&\lim_{h\to 0}\left[\exp\left\{\sum_{k=1}^{T/h}\left[\log\left(1-\lambda(X_{(k-1)h})\cdot h\right)-\log\left(1-\Lambda\cdot h\right)\right]\cdot\left(1-\mathbb{1}_{[(k-1)h,kh]}\right)\right\}\right] \\
&\quad=\lim_{h\to 0}\left[\exp\left\{\sum_{k=1}^{T/h}\left[-\left(\lambda\left(X_{(k-1)h}\right)-\Lambda\right)\cdot h\right]\cdot\left(1-\mathbb{1}_{[(k-1)h,kh]}\right)\right.\right. \\
&\qquad\qquad\left.\left.+\sum_{k=1}^{T/h}\left[\sum_{m=2}^{\infty}\frac{-\left(\lambda^m\left(X_{(k-1)h}\right)-\Lambda^m\right)\cdot h^m}{m!}\right]\cdot\left(1-\mathbb{1}_{[(k-1)h,kh]}\right)\right\}\right] \\
&\quad=\exp\left\{-\int_0^T\left[\lambda\left(X_{s-}\right)-\Lambda\right]\,\mathrm{d}s+0\right\}. \qquad (4.37)
\end{aligned}$$

Now, considering the limit as $h \to 0$ of (4.36) and substituting in (4.37), further noting that in the limit the third term in (4.36) will simply be a product at the jump times $\psi_1, \ldots, \psi_{N_T}$, we have (recalling that $\psi_0 := 0$ and $\psi_{N_T+1} := T$),

$$\begin{aligned}
\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) &= \exp\left\{\sum_{i=1}^{N_T+1}\int_{\psi_{i-1}}^{\psi_i-}\alpha\left(X_s\right)\,\mathrm{d}W_s-\frac{1}{2}\int_0^T\alpha^2\left(X_{s-}\right)\,\mathrm{d}s\right\} \\
&\quad\cdot\exp\left\{-\int_0^T\left[\lambda\left(X_{s-}\right)-\Lambda\right]\,\mathrm{d}s\right\}\cdot\prod_{i=1}^{N_T}\frac{\lambda(X_{\psi_i-})\cdot f_\nu\left(X_{\psi_i};X_{\psi_i-}\right)}{\Lambda\cdot f_\delta\left(X_{\psi_i};X_{\psi_i-}\right)} \qquad (4.38)
\end{aligned}$$

To evaluate each of the stochastic integrals in (4.38) we can apply Itô's formula to $A(u) := \int_0^u \alpha(s)\,\mathrm{d}s$ (as per (4.17) noting that as a consequence of Condition 2 on page 10 we have that $A \in C^2$),

$$A(X_{\psi_i-}) - A(X_{\psi_{i-1}}) = \int_{\psi_{i-1}}^{\psi_i-}\mathrm{d}A(X_s) = \frac{1}{2}\int_{\psi_{i-1}}^{\psi_i-}\alpha'(X_s)\,\mathrm{d}s + \int_{\psi_{i-1}}^{\psi_i-}\alpha(X_s)\,\mathrm{d}W_s. \qquad (4.39)$$

Rearranging (4.39) we have,

$$\begin{aligned}
\int_0^T\alpha\left(X_{s-}\right)\,\mathrm{d}W_s &= \sum_{i=1}^{N_T+1}\int_{\psi_{i-1}}^{\psi_i-}\alpha\left(X_s\right)\,\mathrm{d}W_s \\
&= \sum_{i=1}^{N_T+1}\left[A(X_{\psi_i-})-A(X_{\psi_{i-1}})-\frac{1}{2}\int_{\psi_{i-1}}^{\psi_i-}\alpha'(X_s)\,\mathrm{d}s\right] \\
&= A(X_T)-A(x)-\frac{1}{2}\int_0^T\alpha'\left(X_{s-}\right)\,\mathrm{d}s-\sum_{i=1}^{N_T}\left[A(X_{\psi_i})-A(X_{\psi_i-})\right], \qquad (4.40)
\end{aligned}$$

which upon substituting into (4.38) and recalling that $\phi(X_s) := \alpha^2(X_s)/2 + \alpha'(X_s)/2$ we arrive at our desired result,

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_{s-})\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}$$
$$\cdot \exp\left\{-\int_0^T \left[\lambda(X_{s-}) - \Lambda\right]\,\mathrm{d}s\right\} \cdot \prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right)}{\Lambda \cdot f_\delta\left(X_{\psi_i}; X_{\psi_i-}\right)}. \tag{4.41}$$

□

Similarly, we can find the following two special cases of the Radon-Nikodým derivative of $\mathbb{Q}^x_{0,T}$ with respect to $\mathbb{W}^x_{0,T}$ which are used within this thesis.

Corollary 4.1.1 (Radon-Nikodým derivative for jump diffusions with common jump intensity and size coefficients)**.** *Under Conditions 1–4 and if $\lambda = \Lambda$ and $f_\nu = f_\delta$, the Radon-Nikodým derivative of $\mathbb{Q}^x_{0,T}$ with respect to $\mathbb{W}^x_{0,T}$ exists and is given by Girsanov's formula as follows,*

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_{s-})\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}, \tag{4.42}$$

which can alternatively be represented as follows,

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \prod_{i=1}^{N_T+1}\left[\exp\left\{A(X_{\psi_i-}) - A(X_{\psi_{i-1}}) - \int_{\psi_{i-1}}^{\psi_i-} \phi(X_s)\,\mathrm{d}s\right\}\right]. \tag{4.43}$$

Corollary 4.1.2 (Radon-Nikodým derivative for diffusions)**.** *Under Conditions 1–3 and if $\lambda = 0$, the Radon-Nikodým derivative of $\mathbb{Q}^x_{0,T}$ with respect to $\mathbb{W}^x_{0,T}$ exists and is given by Girsanov's formula as follows,*

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \exp\left\{A(X_T) - A(x) - \int_0^T \phi(X_s)\,\mathrm{d}s\right\}. \tag{4.44}$$

4.1.4 Transition Density

As discussed in the introduction to this chapter, we are interested in finding a representation for the transition density of jump diffusions. Noting again that we can restrict our attention to jump diffusions with unit volatility (under the conditions assumed in this

thesis in Section 1.3 and as discussed in Sections 4.1.2 and 4.1.3), we are interested in finding a representation for the following transition density,

$$p_{t-s}(x,y) := \mathbb{P}\left(X_t \in \mathrm{d}y \,|\, X_s = x\right) / \mathrm{d}y, \quad \text{where } 0 \leq s < t. \tag{4.45}$$

We approach this problem in the manner introduced by Dachuna-Castelle and Florens-Zmirou [1986], by expressing (4.45) as an expectation with respect to Brownian bridge measure with jumps. Recalling that we denote by $\mathbb{Q}^x_{s,t}$ and $\mathbb{W}^x_{s,t}$ as the measures induced by the target and driftless jump diffusions ((4.29) and (4.30) respectively) initialised at $X_s = x$ over the interval $[s,t]$, we further denote by $\mathbb{Q}^{x,y}_{s,t}$ and $\mathbb{W}^{x,y}_{s,t}$ the measures $\mathbb{Q}^x_{s,t}$ and $\mathbb{W}^x_{s,t}$ with the additional restriction that $X_t = y$ and denote by $w_{t-s}(x,y)$ the transition density of the driftless jump diffusion (4.30)), we have,

$$\frac{\mathrm{d}\mathbb{Q}^x_{s,t}}{\mathrm{d}\mathbb{W}^x_{s,t}}(X) = \frac{p_{t-s}(x,y)}{w_{t-s}(x,y)} \frac{\mathrm{d}\mathbb{Q}^{x,y}_{s,t}}{\mathrm{d}\mathbb{W}^{x,y}_{s,t}}(X), \tag{4.46}$$

which upon taking expectations with respect to $\mathbb{W}^{x,y}_{s,t}$ we have,

$$\begin{aligned}\mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\frac{\mathrm{d}\mathbb{Q}^x_{s,t}}{\mathrm{d}\mathbb{W}^x_{s,t}}(X)\right] &= \mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\frac{p_{t-s}(x,y)}{w_{t-s}(x,y)} \frac{\mathrm{d}\mathbb{Q}^{x,y}_{s,t}}{\mathrm{d}\mathbb{W}^{x,y}_{s,t}}(X)\right] \\ &= \frac{p_{t-s}(x,y)}{w_{t-s}(x,y)} \cdot 1, \end{aligned} \tag{4.47}$$

and so after rearrangement we arrive at our desired representation of (4.45),

$$p_{t-s}(x,y) = w_{t-s}(x,y) \cdot \mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\frac{\mathrm{d}\mathbb{Q}^x_{s,t}}{\mathrm{d}\mathbb{W}^x_{s,t}}(X)\right]. \tag{4.48}$$

In the specific case where the target jump diffusion has zero jump intensity ($\lambda = 0$), $\mathbb{W}^{x,y}_{s,t}$ is Brownian bridge measure and so (with reference to Section 2.8 and Corollary 4.1.2) we more explicitly have,

$$\begin{aligned} p_{t-s}(x,y) &= \frac{1}{\sqrt{2\pi(t-s)}} \exp\left\{-\frac{(y-x)^2}{2(t-s)}\right\} \cdot \mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\exp\left\{A(y) - A(x) - \int_s^t \phi\left(X_u\right)\, \mathrm{d}u\right\}\right] \\ &= \frac{1}{\sqrt{2\pi(t-s)}} \exp\left\{A(y) - A(x) - \frac{(y-x)^2}{2(t-s)}\right\} \cdot \mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\exp\left\{-\int_s^t \phi\left(X_u\right)\, \mathrm{d}u\right\}\right] \end{aligned} \tag{4.49}$$

As we discuss in Section 4.2, typically the expectation with respect to Brownian bridge measure renders the transition density intractable. However, the methodology we present

in Chapter 5 and Chapter 7 enables simulation from the transition density without any form of error.

4.2 Approximate Methods for Simulating Diffusions and Jump Diffusions

In this section we briefly review existing discretisation methods for simulating sample paths from the measure induced by (4.1) (denoted by $\mathbb{T}^v$),

$$\mathrm{d}V_t = \beta(V_{t\text{-}})\,\mathrm{d}t + \sigma(V_{t\text{-}})\,\mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\mu}, \qquad V_0 = v \in \mathbb{R}, \quad t \in [0,T]. \tag{4.50}$$

This problem has garnered much attention as $\mathbb{T}^v$ is typically not explicitly known and so to compute expected values with respect to this measure for various test functions h (or more precisely compute $\mathbb{E}_{\mathbb{T}^v}[h(V)]$), a Monte Carlo approach is often taken. In particular, following the Monte Carlo method discussed in Chapter 2, if it is possible to draw independently $V^{(1)}, V^{(2)}, \ldots, V^{(N)} \sim \mathbb{T}^v$ then a consistent estimator of the expectation can be constructed as follows,

$$\text{w.p. 1:} \quad \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} h(V^{(i)}) = \mathbb{E}_{\mathbb{T}^v}[h(V)]\,, \tag{4.51}$$

with convergence in N as given in (2.3). However, as it isn't possible to draw entire sample paths from $\mathbb{T}^v$ (as discussed in Chapter 1, sample paths are infinite dimensional random variables), consideration has typically only been given to computing expectations for test functions which require the simulation of sample paths at a predefined finite collection of time points in the interval $[0,T]$. As such the emphasis on research has traditionally been focused on simulating from the transition density of V (see for instance [Kloeden and Platen, 1992; Platen and Bruti-Liberati, 2010]),

$$p_t(v,z) := \mathbb{P}\left(V_t \in \mathrm{d}z \,|\, V_0 = v\right) / \mathrm{d}z. \tag{4.52}$$

Unfortunately, in all but a small number of trivial cases (such as Brownian motion as discussed in Section 2.8) there exists no closed form representation of (4.52), and so it is not possible to draw sample paths even at a finite collection of time points in order to construct a Monte Carlo estimator as in (4.51). As such a number of numerical methods have been developed in order to approximately simulate from (4.52). These numerical

methods are generally categorised as *strong* schemes (in which a realisation is a pathwise approximation of a diffusion sample path) and *weak* schemes (in which a realisation approximates $\mathbb{T}^v$).

The general strategy employed in order to simulate a sample path at some time T using either a strong or weak scheme (i.e. simulate from the transition density p_T), is to hope that as Brownian motion has a Gaussian transition density (see Section 2.8) then over short intervals (4.52) it is well approximated by one with fixed drift, volatility and jump process coefficients (by a continuity argument). As such, by breaking the interval the sample path is to be simulated over into a fine mesh (for instance, of regular size Δ), then iteratively (at each mesh point) fixing the coefficients and simulating the sample path to the next mesh point using an approximation of the transition density $p^{\mathrm{A}}_{\Delta} \approx p_{\Delta}$, then the resulting realisation will be an approximate draw from (4.52). The adequacy of any given strong or weak scheme is typically measured using the strong or weak convergence criterion respectively (where we denote by V^{Δ} as the approximation of V and $C^l_p(\mathbb{R}^d, \mathbb{R})$ as the class of l times continuously differentiable functions $h : \mathbb{R}^d \to \mathbb{R}$ with their partial derivatives up to order l having polynomial growth).

Convergence Criteria 1 (Strong Convergence [Kloeden and Platen, 1992])**.** *The discrete time approximation V^{Δ} converges strongly to V at time T if,*

$$\lim_{\Delta \downarrow 0} \mathbb{E}\left[\left|V_T - V_T^{\Delta}\right|\right] = 0. \tag{4.53}$$

Furthermore, we say that V^{Δ} converges strongly to V with order $\gamma > 0$ at time T if there exists constants $C > 0$ and $\Delta_0 > 0$ such that,

$$\mathbb{E}\left[\left|V_T - V_T^{\Delta}\right|\right] \leq C\Delta^{\gamma}, \tag{4.54}$$

for each $\Delta \in (0, \Delta_0)$.

Convergence Criteria 2 (Weak Convergence [Kloeden and Platen, 1992])**.** *The discrete time approximation V^{Δ} converges weakly to V at time T with respect to a class of test functions $h : \mathbb{R}^d \to \mathbb{R}$ if,*

$$\lim_{\Delta \downarrow 0} \left|\mathbb{E}\left[h(V_T)\right] - \mathbb{E}\left[h(V_T^{\Delta})\right]\right| = 0. \tag{4.55}$$

Furthermore, we say that V^{Δ} converges weakly to V with order $\beta > 0$ at time T as

$\Delta \downarrow 0$ *if for every* $h \in C_P^{2(\beta+1)}(\mathbb{R}^d, \mathbb{R})$ *there exists constants* $C > 0$ *and* $\Delta_0 > 0$ *such that,*

$$\left|\mathbb{E}\left[h(V_T)\right] - \mathbb{E}\left[h(V_T^{\Delta})\right]\right| \leq C\Delta^{\beta}, \tag{4.56}$$

for each $\Delta \in (0, \Delta_0)$.

In the following subsections we briefly detail some of the more common discretisation schemes (a more detailed account can be found in a number of texts, for instance, in Kloeden and Platen [1992], Iacus [2008] and Platen and Bruti-Liberati [2010]). However, our exposition is restricted to discretisation schemes for unconditioned diffusions (as opposed to diffusion bridges, unconditioned jump diffusions and jump diffusion bridges) so that the key ideas are conveyed without the considerable and unnecessary complications introduced in broader settings (note that in the case of unconditioned jump diffusions with bounded jump intensity, then by a trivial extension of Section 2.9.3 approximate simulation is possible as detailed in Algorithm 4.2.1). In particular, we consider diffusions which are solutions to the following SDE,

$$\mathrm{d}V_t = \beta(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t, \qquad V_0 = v \in \mathbb{R}, \quad t \in [0, T]. \tag{4.57}$$

It is important to note at this point that we are omitting details on discretisation schemes as there does not exist any existing discretisation scheme which can be used to directly compare with the methodology developed in this thesis. In particular, unlike the *exact algorithms* (which we introduce and discuss in Chapter 5), discretisation schemes suffer from a number of inherent problems.

To emphasise this point we return to and extend the discussion of the weaknesses of discretisation schemes (for our purposes) that we made in Chapter 1. To begin with we note that all discretisation schemes introduce implicit approximation error and output some finite dimensional subset of an approximated sample path. Although this approximation error can be reduced by increasing the mesh size (i.e. decreasing Δ), it 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 discussed in Chapter 1 and illustrated in Figure 4.2.1 (which is a reproduction of Figure 1.0.2 included here again for convenience), even when our test function h only requires the simulation of sample paths at a single terminal time point, discretisation introduces

approximation error resulting in the loss of unbiasedness of our Monte Carlo estimator (4.51) (the Euler discretisation scheme with which we make a comparison is introduced and discussed in Section 4.2.1). If $\mathbb{T}^v$ has a highly non-linear drift, includes a compound Poisson process component or is the measure induced by a diffusion bridge; or the test function h itself requires simulation of sample paths at a collection of time points, then this problem is exacerbated. Note that unbiasedness itself is a desirable characteristic of a Monte Carlo estimator and is necessary in the context of a number of applications (for instance, within the pseudo-marginal MCMC framework of Andrieu and Roberts [2009]).

Furthermore, although discretisation schemes output a finite dimensional subset of an approximated sample path, they don't sufficiently characterise the entire sample path for the evaluation of some test functions. As noted in Chapter 1, this point is well illustrated by Figures 4.2.2(a), 4.2.2(b) and 4.2.2(c) (which again is a reproduction of Figure 1.0.1 included here for convenience) in which we are interested in whether a simulated sample path $V \sim \mathbb{T}^v$, crosses some barrier (i.e. for some set A we have $h := \mathbb{1}(V \in A)$). 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. The alternate methodology we develop in this thesis can be used to address these types of applications (as we discuss in Chapter 8).

Algorithm 4.2.1 Generic Jump Diffusion Discretisation Scheme.

1. Set $j = 0$ and $\Psi_0 = 0$. While $\Psi_j < T$,
   (a) Simulate $\tau \sim \text{Exp}(\Lambda)$. Set $j = j + 1$ and $\Psi_j = \Psi_{j-1} + \tau$.
   (b) For i in 1 to $\left\lfloor \left[(\Psi_j \wedge T) - \Psi_{j-1} \right] / \Delta \right\rfloor$,
      i. Simulate $V^{\Delta}_{\Psi_{j-1}+i\Delta} \sim p^{A}_{\Delta}(V^{\Delta}_{\Psi_{j-1}+(i-1)\Delta}, z)$.
   (c) Simulate $V^{\Delta}_{(\Psi_j \wedge T)-} \sim p^{A}_{\Psi_j - \Psi_{j-1} - i\Delta}(V^{\Delta}_{\Psi_{j-1}+i\Delta}, z)$.
   (d) If $\Psi_j > T$ then set $V^{\Delta}_{T} = V^{\Delta}_{T-}$ else,
      i. With probability $\lambda\left(V^{\Delta}_{\Psi_j-}\right)/\Lambda$ simulate $\nu \sim f_{\nu}\left(V^{\Delta}_{\Psi_j-}\right)$ and set $V^{\Delta}_{\Psi_j} = V^{\Delta}_{\Psi_j-} + \nu$, else set $V^{\Delta}_{\Psi_j} = V^{\Delta}_{\Psi_j-}$.

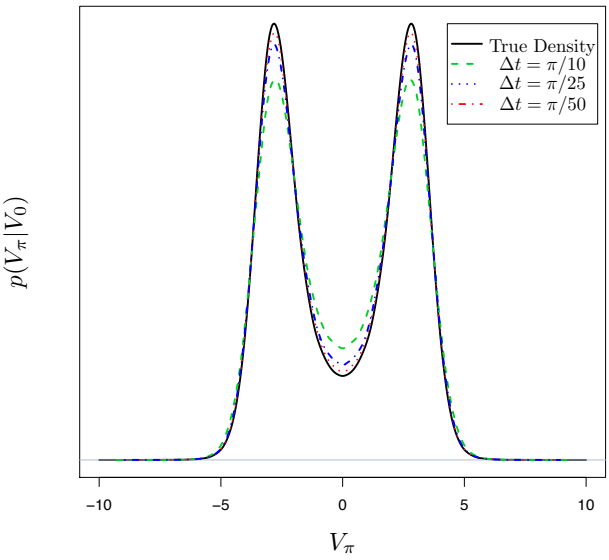
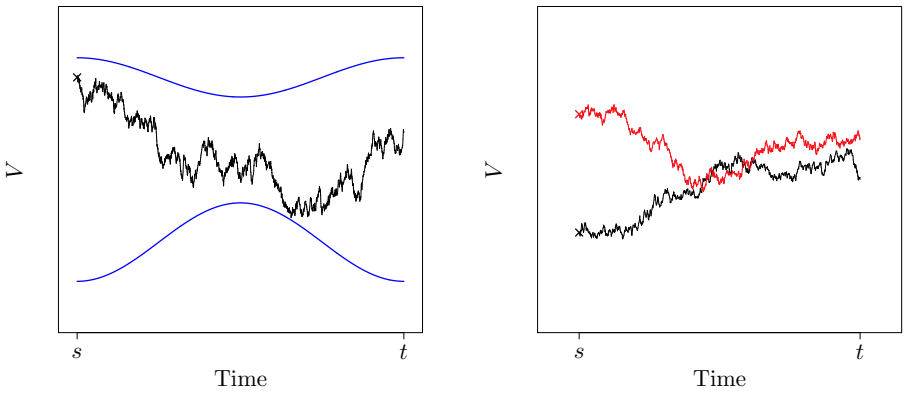
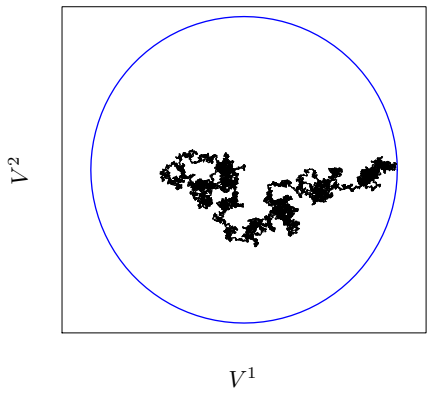


Figure 4.2.1: Density of V_π and approximations given by an Euler discretisation with various mesh sizes, given $V_0 = 0$ where $\mathrm{d}V_t = \sin(V_t)\,\mathrm{d}t + \mathrm{d}W_t$.



(a) Non-linear two sided barrier

(b) Diffusion barrier



(c) 2-D circular barrier

Figure 4.2.2: Examples of test functions in which evaluation requires the characterisation of an entire sample path.

4.2.1 Strong Taylor Schemes

The most common (and simplest) discretisation scheme for simulating diffusions of the form in (4.57) is the Euler-Maruyama scheme (proposed by Maruyama [1955]), in which a sample path is approximated at each point in time on a mesh by means of the following recursion,

$$V^{\Delta}_{t+\Delta} = V^{\Delta}_t + \beta(V^{\Delta}_t) \cdot \Delta + \sigma(V^{\Delta}_t) \cdot \xi, \qquad \text{where } \xi \sim \mathrm{N}(0, \Delta). \tag{4.58}$$

The intuition behind the Euler-Maruyama scheme is based on the integrated form of (4.57), noting that if Δ is small enough then the coefficients of (4.57) will be well approximated by a diffusion with constant coefficients. Indeed, under Conditions 2 and 3 the Euler-Maruyama scheme converges strongly (see Convergence Criteria 1) with order $\gamma = 0.5$ (see [Kloeden and Platen, 1992, Thm. 10.2.2]). Implementing the Euler-Maruyama scheme is straightforward as the transition density has a closed form representation,

$$p^{E}_{\Delta}(v, z) := \mathrm{N}\left(z; v + \beta(v)\Delta, \sigma^2(v)\Delta\right). \tag{4.59}$$

Clearly the assumption that the coefficients of (4.57) will be well approximated by a diffusion with constant coefficients is unsound for highly non-linear drift and volatility coefficients (unless the mesh is much finer and Δ is much smaller). The Milstein scheme (proposed by Milstein [1979]) is an alternate discretisation scheme which converges strongly with order $\gamma = 1$ (under suitable conditions and as detailed in [Kloeden and Platen, 1992, Chap. 10.3 Thm. 19.3.5]) and is obtained by approximating the sample path at each point on a mesh by means of the following recursion,

$$V^{\Delta}_{t+\Delta} = V^{\Delta}_t + \beta(V^{\Delta}_t) \cdot \Delta + \sigma(V^{\Delta}_t) \cdot \xi + \frac{1}{2} \cdot \sigma(V^{\Delta}_t) \cdot \sigma'(V^{\Delta}_t) \cdot \left[\xi^2 - \Delta\right], \quad \text{where } \xi \sim \mathrm{N}(0, \Delta). \tag{4.60}$$

It was shown in [Elerian, 1998, Thm. 2.1] that the transition density of the Milstein scheme has the following closed form representation,

$$p^{M}_{\Delta}(v, z) := \frac{\exp\{-B/2\}}{|A| \cdot \sqrt{2\pi}} \cdot \sqrt{\frac{A}{z - C}} \cdot \exp\left\{-\frac{z - C}{2A}\right\} \cdot \cosh\left(\sqrt{\frac{(z - C) \cdot B}{A}}\right), \tag{4.61}$$

where,

$$A := \frac{\sigma(v) \cdot \sigma'(v) \cdot \Delta}{2}, \qquad B := \frac{1}{(\sigma'(v))^2 \cdot \Delta}, \tag{4.62}$$

$$C := -\frac{\sigma(v)}{2\sigma'(v)} + v + \beta(v) \cdot \Delta - A. \tag{4.63}$$

Both the Euler-Maruyama and Milstein schemes are examples of the class of *Strong Taylor Schemes*, which are obtained by applying Itô's formulae (see Section 4.1.1) to conduct a stochastic Taylor expansion of the integrated form of (4.57) and truncating the expansion at various levels [Kloeden and Platen, 1992, Chap. 10]. In principle, any order of strong convergence can be obtained ($\gamma \in \{0.5, 1, 1.5, 2 \ldots\}$) by including more terms within the truncated stochastic Taylor expansion (under suitable conditions).

4.2.2 Other Discretisation Schemes

A number of alternative discretisation schemes to the strong Taylor schemes exist, which in various settings will outperform the strong Taylor schemes.

The *Runge-Kutta* discretisation schemes (see [Kloeden and Platen, 1992, Chap. 11]) are a class of explicit strong schemes which avoid the need to evaluate derivatives (of various orders) of the drift and volatility coefficients at each mesh point in the simulation of an approximated sample path. For each strong Taylor scheme there exists an analogous Runge-Kutta scheme (of the same strong order) in which derivatives are approximated. For instance, in the Runge-Kutta analog of the Milstein scheme the sample path is approximated at each point in time on a mesh by means of the following recursion,

$$V^{\Delta}_{t+\Delta} = V^{\Delta}_t + \beta(V^{\Delta}_t) \cdot \Delta + \sigma(V^{\Delta}_t) \cdot \xi + \frac{1}{2} \cdot \sigma(V^{\Delta}_t) \cdot \frac{\sigma\left(\hat{V}^{\Delta}_{t+\Delta}\right) - \sigma(V^{\Delta}_t)}{\sqrt{\Delta}} \cdot \left[\xi^2 - \Delta\right], \tag{4.64}$$

where,

$$\hat{V}^{\Delta}_{t+\Delta} := V^{\Delta}_t + \beta(V^{\Delta}_t) \cdot \Delta + \sigma(V^{\Delta}_t) \cdot \sqrt{\Delta}, \quad \text{and } \xi \sim \mathrm{N}(0, \Delta). \tag{4.65}$$

The *Shoji-Ozaki* scheme (as proposed in Shoji [1995, 1998] and Shoji and Ozaki [1998]) is a discretisation scheme in which the target diffusion is first transformed into one with unit volatility coefficient (under suitable conditions and as per Section 4.1.2) and then an approximated sample path is simulated on a mesh by means of a Taylor expansion of the

drift coefficient. The transformed approximate sample path is then projected by the inverse of this transformation resulting in an approximate sample path simulated under the target measure $\mathbb{T}^v$. It was shown in Shoji and Ozaki [1997] that in numerical comparison the Shoji-Ozaki scheme compares favourably to other discretisation schemes (including the Euler-Maruyama scheme) in settings where it can be applied.

A related transformation based scheme is the *Hermite polynomial expansions* proposed by Aït-Sahalia [2002, 2008], in which in addition to transforming the diffusion into one with unit volatility (under suitable conditions and as per Section 4.1.2) a second time transformation is also conducted so that the resulting transition density over small time intervals is well approximated by a standard Normal density. The transformed transition density is then approximated by a Hermite polynomial expansion which is truncated at some user specified level.

Implicit strong Taylor schemes (see [Kloeden and Platen, 1992, Chap. 12.2] are discretisation schemes in which the approximate sample path, which is generated recursively on a mesh, includes within the recursion the subsequent state of the approximate sample path. For instance, the implicit Euler scheme converges strongly with order $\gamma = 0.5$ and has the following form (where $\zeta \in (0, 1]$ is a tuning parameter which characterises the degree of implicitness),

$$V^{\Delta}_{t+\Delta} = V^{\Delta}_t + (1-\zeta)\cdot\beta(V^{\Delta}_t)\cdot\Delta + \zeta\cdot\beta(V^{\Delta}_{t+\Delta})\cdot\Delta + \sigma(V^{\Delta}_t)\cdot\xi, \qquad \text{where } \xi \sim \mathrm{N}(0,\Delta). \tag{4.66}$$

Finally, we conclude this section by noting (but not detailing) that there is also an extensive literature on *weak* discretisation schemes (see [Kloeden and Platen, 1992, Chap. 14 and Chap. 15]) which, as discussed in Section 4.2, output sample paths which approximate $\mathbb{T}^v$ (as opposed to strong schemes in which a simulated sample path is a path-wise approximation of a diffusion sample path). Many strong discretisation schemes have a weak analog, however, weak schemes can be simpler to implement as we only need to approximate the measure induced by the target diffusion. For instance, the Gaussian increments of Brownian motion (see Section 2.8) could be substituted by other more easily generated random variables (with similar moment properties).

Part II

Methodology

5 Exact Algorithms for Simulating Diffusions and Jump Diffusions

"The surface of things gives enjoyment, their interiority gives life."

— Piet Mondrian

In this chapter we introduce a novel mathematical framework for simulating (jump) diffusion and (jump) diffusion bridge sample path skeletons (see Definition 1) without approximation error (so called *exact algorithms*). As discussed in Chapter 1, existing exact algorithms for diffusions [Beskos and Roberts, 2005; Beskos et al., 2006a, 2008; Chen and Huang] and jump diffusions [Casella and Roberts, 2010; Giesecke and Smelov, Forthcoming; Gonçalves and Roberts, 2013] focus on the problem of simulating sample paths at finite collections of time points. However, our extended notion of simulating sample path skeletons which characterise the entire sample path not only provides potential computational improvement over existing exact algorithms, but provides a far more flexible structure for use within a variety of applications (as we explore in Chapters 8 and 9).

This chapter is organised as follows: In Section 5.1 and Section 5.3 we outline methodology for simulating diffusion and jump diffusion sample path skeletons respectively, whereas in Sections 5.2 and 5.4 we outline methodology for simulating diffusion and jump diffusion bridge sample path skeletons.

5.1 Exact Algorithms for Unconditioned Diffusions

In this section we outline how to simulate skeletons for diffusion sample paths which can be represented (under the conditions in Section 1.3 and following the Lamperti transfor-

mation in Section 4.1.2), as the solution to SDEs with unit volatility,

$$\mathrm{d}X_t = \alpha(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \quad X_0 = x \in \mathbb{R}, \quad t \in [0, T]. \tag{5.1}$$

As discussed in Chapter 1, exact algorithms are a class of rejection samplers (see Section 2.4) operating on diffusion path space. We begin by outlining an idealised rejection sampler originally proposed in Beskos and Roberts [2005] 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 (for instance those introduced in Beskos and Roberts [2005], Beskos et al. [2006a] and Beskos et al. [2008]), however, in this thesis we present two novel algorithmic interpretations of this rejection sampler. In Section 5.1.1 we present the Unbounded Exact Algorithm (UEA) which is a methodological extension of existing exact algorithms, and in Section 5.1.2 we introduce the novel Adaptive Unbounded Exact Algorithm (AUEA).

Recalling the rejection sampling approach outlined in Section 2.4, we could simulate sample paths from our target measure (the measure induced by (5.1) and denoted by $\mathbb{Q}^x_{0,T}$) by means of a proposal measure which we can simulate proposal sample paths from. A natural equivalent measure to choose as a proposal is Wiener measure, $\mathbb{W}^x_{0,T}$ (see Section 2.8), as (5.1) 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 Corollary 4.1.2) then $(X \,|\, I = 1) \sim \mathbb{Q}^x_{0,T}$. On average sample paths are accepted with probability $P_{\mathbb{W}^x_{0,T}} := \mathbb{E}_{\mathbb{W}^x_{0,T}}\left[P_{\mathbb{W}^x_{0,T}}(X)\right]$. However, the function $A(X_T)$ in (4.44) 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 (as introduced in Beskos and Roberts [2005]) as the proposal measure and consider the resulting modification to the acceptance probability.

Definition 5.1.1. *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 follows (by Result 3 we*

have $\forall T \leq T_0$*,* $h(y; x, T)$ *is integrable),*

$$h(y; x, T) := \frac{1}{c(x, T)} \exp\left\{A(y) - \frac{(y-x)^2}{2T}\right\}. \tag{5.2}$$

Theorem 5.1.1 (Biased Brownian Motion [Beskos and Roberts, 2005, Prop. 3])**.** $\mathbb{Q}^x_{0,T}$ *is equivalent to* $\mathbb{Z}^x_{0,T}$ *with Radon-Nikodým derivative:*

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{Z}^x_{0,T}}(X) \propto \exp\left\{-\int_0^T \phi(X_s)\,\mathrm{d}s\right\}. \tag{5.3}$$

Proof. Denoting by $\mathbb{Z}^{x,y}_{0,T}$ and $\mathbb{W}^{x,y}_{0,T}$ as the measures induced by Z and Wiener measure respectively on $[0, T]$ with initial condition $X_0 = x \in \mathbb{R}$ and terminal condition $X_T = y \in \mathbb{R}$, we have (noting that $\mathbb{Z}^{x,y}_{0,T} = \mathbb{W}^{x,y}_{0,T}$),

$$\frac{\mathrm{d}\mathbb{Z}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \frac{h\,(y; x, T)}{w_T(x, y)} \frac{\mathrm{d}\mathbb{Z}^{x,y}_{0,T}}{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}(X) \propto \exp\{A(y)\}\,. \tag{5.4}$$

Now, noting that $\mathrm{d}\mathbb{Q}^x_{0,T}/\,\mathrm{d}\mathbb{Z}^x_{0,T}(X) = \left(\mathrm{d}\mathbb{Q}^x_{0,T}/\,\mathrm{d}\mathbb{W}^x_{0,T}(X)\right)/\left(\mathrm{d}\mathbb{Z}^x_{0,T}/\,\mathrm{d}\mathbb{W}^x_{0,T}(X)\right)$ (see for instance [Halmos, 1974, Sec. 32, Thm. A]) and recalling (4.44) then we arrive at (5.3) as required. □

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, $(X_{(0,T)} \,|\, X_0 = x, X_T = y) \sim \mathbb{W}^{x,y}_{0,T}$. Then if the sample path is accepted ($I = 1$) with probability $P_{\mathbb{Z}^x_{0,T}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{Z}^x_{0,T}}(X) \in [0, 1]$ then $(X \,|\, I = 1) \sim \mathbb{Q}^x_{0,T}$. More formally we have,

$$P_{\mathbb{Z}^x_{0,T}} := \mathbb{E}_{\mathbb{Z}^x_{0,T}}\left[P_{\mathbb{Z}^x_{0,T}}(X)\right] = \mathbb{E}_h\left[\mathbb{E}_{\mathbb{Z}^x_{0,T}}\left[P_{\mathbb{Z}^x_{0,T}}(X)\,\middle|\,X_T\right]\right] =: \mathbb{E}_h\left[\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}}\left[P_{\mathbb{Z}^x_{0,T}}(X)\right]\right] \tag{5.5}$$

We can now construct an (idealised) rejection sampler to draw sample paths from $\mathbb{Q}^x_{0,T}$ as outlined in Algorithm 5.1.1, noting that as $\inf_{u\in[0,T]} \phi(X_u) \geq \Phi$ (see Condition 5) we can choose $M := \exp\{-\Phi T\}$ to ensure $P_{\mathbb{Z}^x_{0,T}}(X) \in [0, 1]$.

Unfortunately, Algorithm 5.1.1 can't be directly implemented as it isn't possible to draw entire sample paths from $\mathbb{W}^{x,y}_{0,T}$ in Step 1b (they're infinite dimensional random variables)

Algorithm 5.1.1 Idealised Diffusion Rejection Sampler [Beskos and Roberts, 2005].

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 $P_{\mathbb{Z}^x_{0,T}}(X) = \exp\left\{-\int_0^T \phi(X_s)\,\mathrm{d}s\right\} \cdot \exp\{\Phi T\}$ accept, else reject and return to Step 1.

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 $F := F(X) \sim \mathbb{F}$ (the details of which are in Sections 5.1.1 and 5.1.2), 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. More formally we have $\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}}[P_{\mathbb{Z}^x_{0,T}}(X)] = \mathbb{E}_{\mathbb{F}}[\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|F}[P_{\mathbb{Z}^x_{0,T}}(X)]]$. 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 ($X^{\text{fin}} \sim \mathbb{W}^{x,y}_{0,T}$) in Step 1b in order to evaluate the acceptance probability. The remainder 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, $X^{\text{rem}} \sim \mathbb{W}^{x,y}_{0,T}\,\big|\,(X^{\text{fin}}, F)$ (noting that $X = X^{\text{fin}} \cup X^{\text{rem}}$). The synthesis of this argument is presented in Algorithm 5.1.2[1]. It should be noted that this is related to the notion of demarginalisation (see Section 2.3), in which the addition of an auxiliary random variable simplifies sampling. In this case as we are interested in evaluating acceptance of a proposal sample path, but not simulating it in its entirety, this enables a practical implementation (Algorithm 5.1.2) as opposed to the impractical Algorithm 5.1.1.

In conclusion, although it isn't possible to simulate entire sample paths from $\mathbb{Q}^x_{0,T}$, it

[1]Note that Algorithm 5.1.2 Step 5 is separated from the rest of the algorithm and asterisked. This convention is used within this thesis to indicate the final step within an Exact Algorithm, which 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.

Algorithm 5.1.2 Implementable Exact Algorithm [Beskos and Roberts, 2005; Beskos et al., 2006a].

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

5. * *Simulate* $X^{rem} \sim \mathbb{W}_{0,T}^{x,y} \Big| (X^{fin}, F)$.

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 [Beskos and Roberts, 2005; Beskos et al., 2006a, 2008] achieve by applying Principles 1 and 2. However, no existing exact algorithm addresses how to construct $\mathbb{F}$ under the conditions in Section 1.3 to additionally perform Algorithm 5.1.2 Step 5. We address this in Sections 5.1.1 and 5.1.2.

In the next two sections we present two distinct, novel interpretations of Algorithm 5.1.2. In Section 5.1.1 we present the Unbounded Exact Algorithm (UEA) which is a methodological extension of existing exact algorithms and direct interpretation of Algorithm 5.1.2. In Section 5.1.2 we introduce the Adaptive Unbounded Exact Algorithm (AUEA) which takes a recursive approach to Algorithm 5.1.2 Steps 2, 3 and 4.

5.1.1 Bounded & 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 5.1.2 and considering how to construct a suitable finite dimensional random variable $F \sim \mathbb{F}$.

As first noted in Beskos et al. [2006a], it is possible to construct and simulate the random variable F required in Algorithm 5.1.2, provided $\phi(X_{[0,T]})$ can be bounded above and below (where we denote by $X_{[0,T]}$ as the trajectory of the sample path X in the interval $[0, T]$). It was further noted in Beskos et al. [2008] that F could be constructed

and simulated provided it were possible to simulate a Brownian bridge proposal sample path in conjunction with information as to an interval in which it was contained, and that conditional on this interval, $\phi(X_{[0,T]})$ was bounded above and below. 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 thesis we discuss two alternative layer constructions (see Sections 6.2 and 6.3), both of which complicate the key ideas behind the UEA and so are only discussed in abstract terms at this stage.

Further to Beskos et al. [2008], 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, we partition the path space of $\mathbb{Z}^x_{0,T} \,\big|\, X_T = \mathbb{W}^{x,y}_{0,T}$ 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 6.2), 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 $L_X \in \mathbb{R}$ respectively, which are functions of the simulated layer R). As such we have,

$$\begin{aligned} P_{\mathbb{Z}^x_{0,T}} &:= \mathbb{E}_{\mathbb{Z}^x_{0,T}}\left[P_{\mathbb{Z}^x_{0,T}}(X)\right] = \mathbb{E}_h\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}}\left[P_{\mathbb{Z}^x_{0,T}}(X)\right] = \mathbb{E}_h\mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|R}\left[P_{\mathbb{Z}^x_{0,T}}(X)\right] \\ &= \mathbb{E}_h\mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|R}\left[\exp\left\{-\int_0^T \phi(X_s)\,\mathrm{d}s\right\}\cdot e^{\Phi T}\right]. \end{aligned} \tag{5.6}$$

Proceeding in a similar manner to Beskos et al. [2006b] to construct our finite dimensional unbiased estimator of the acceptance probability of a sample path, we consider a Taylor series expansion of the exponential function in (5.6),

$$\begin{aligned} P_{\mathbb{Z}^x_{0,T}}(X) &= e^{-(L_X-\Phi)T}\cdot e^{-(U_X-L_X)T}\exp\left\{\int_0^T U_X-\phi(X_s)\,\mathrm{d}s\right\} \\ &= e^{-(L_X-\Phi)T}\cdot\left[\sum_{j=0}^{\infty}\frac{e^{-(U_X-L_X)T}[(U_X-L_X)T]^j}{j!}\left\{\int_0^T\frac{U_X-\phi(X_s)}{(U_X-L_X)T}\,\mathrm{d}s\right\}^j\right], \end{aligned} \tag{5.7}$$

again employing methods found in Beskos et al. [2006b], we note that if we let $\mathbb{K}_R$ be the law of $\kappa \sim \text{Poi}((U_X - L_X)T)$, $\mathbb{U}\kappa$ the distribution of $(\xi_1, \ldots, \xi_\kappa) \overset{\text{iid}}{\sim} \text{U}[0, T]$ we have,

$$\begin{aligned} P_{\mathbb{Z}^x_{0,T}}(X) &= e^{-(L_X - \Phi)T} \cdot \mathbb{E}_{\mathbb{K}_R}\left[\left(\int_0^T \frac{U_X - \phi(X_s)}{(U_X - L_X)T}\,\mathrm{d}s\right)^\kappa \middle| X\right] \\ &= e^{-(L_X - \Phi)T} \cdot \mathbb{E}_{\mathbb{K}_R}\mathbb{E}_{\mathbb{U}\kappa}\left[\prod_{i=1}^{\kappa}\left(\frac{U_X - \phi(X_{\xi_i})}{U_X - L_X}\right)\middle| X\right]. \end{aligned} \tag{5.8}$$

The key observation to make from (5.8) is that the acceptance probability of a sample path $X \sim \mathbb{Z}^x_{0,T}$ can be evaluated without the need of the sample path in its entirety, and can instead be evaluated using a finite dimensional realisation, X^{fin}. Simulating a finite dimensional proposal as suggested by (5.6) and (5.8) and incorporating it within Algorithm 5.1.2 results directly in the UEA presented in Algorithm 5.1.3. 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 6.2.

Algorithm 5.1.3 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 $\left(X_{\xi_1}, \ldots, X_{\xi_\kappa}\right)\Big| R$,
   (a) Simulate $\kappa \sim \text{Poi}((U_X - L_X)T)$ and skeleton times $\xi_1, \ldots, \xi_\kappa \overset{\text{iid}}{\sim} \text{U}[0, T]$.
   (b) Simulate sample path at skeleton times $X_{\xi_1}, \ldots, X_{\xi_\kappa} \sim \mathbb{W}^{x,y}_{0,T}\Big| R$.
5. With probability $\prod_{i=1}^{\kappa}\left[\left(U_X - \phi(X_{\xi_i})\right)/\left(U_X - L_X\right)\right]$, accept entire path, else reject and return to Step 1.

---

6. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{\kappa+1}\mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i}\right)\Big| R$.

The UEA can be viewed as a nested 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 Beskos et al. [2008]), the UEA conducts early rejection to avoid any further unnecessary simulation of the rejected sample path.

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 $(X_{\xi_1}, \ldots, X_{\xi_\kappa})$ and layer R (denoting $\xi_0 := 0$ and $\xi_{\kappa+1} := T$),

$$\mathcal{S}_{\mathrm{UEA}}(X) := \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{\kappa+1}, R \right\}. \tag{5.9}$$

An illustrative example of an accepted sample path skeleton simulated by means of the UEA is given in Figure 5.1.1.

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 5.1.3, the skeleton (5.10) 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 Beskos et al. [2006a], which we term the BEA.

$$\mathcal{S}_{\mathrm{BEA}}(X) := \left\{ \left(\xi_i, X_{\xi_i} \right)_{i=0}^{\kappa+1} \right\}. \tag{5.10}$$

A second exact algorithm (EA2) was also proposed in Beskos et al. [2006a] (the details of which we omit from this thesis), 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 [Beskos and Roberts, 2005; Beskos et al., 2006a, 2008], 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 1.3 (so satisfying Principle 3). Algorithm 5.1.3

Step 6 can't be conducted in existing exact algorithms as the layer imparts information across the entire interval. However, in Section 6.2 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 $[a,b] \subseteq [0,T]$),

$$\mathcal{S}'_{\mathrm{UEA}}(X) := \left\{ \left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, R, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1} \right\} = \left\{ \left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1} \right\}. \tag{5.11}$$

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 $[0,T]$. As such, simulating the sample path at further times after acceptance as in Algorithm 5.1.3 Step 6 is direct,

$$X^{\mathrm{rem}} \sim \mathbb{W}_{0,T}^{x,y} \left| \mathcal{S}'_{\mathrm{UEA}} = \otimes_{i=1}^{\kappa+1} \left(\mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}} \left| R_X^{[\xi_{i-1},\xi_i]} \right. \right). \right. \tag{5.12}$$

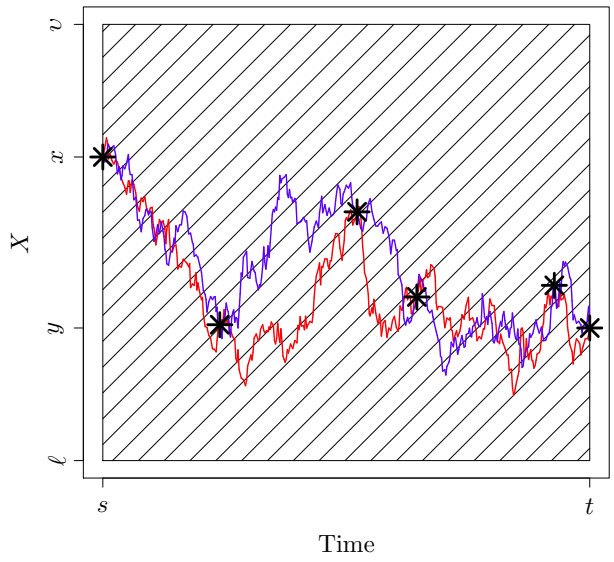
It should be noted that it is possible to simulate Algorithm 5.1.3 Steps 2 and 3 jointly by suitably modifying the mechanism by which layers are proposed and accepted from that outlined in Section 6.2. However, it is unclear if this is computationally advantageous and so we omit details.

5.1.1.1 Implementational Considerations – Interval Length

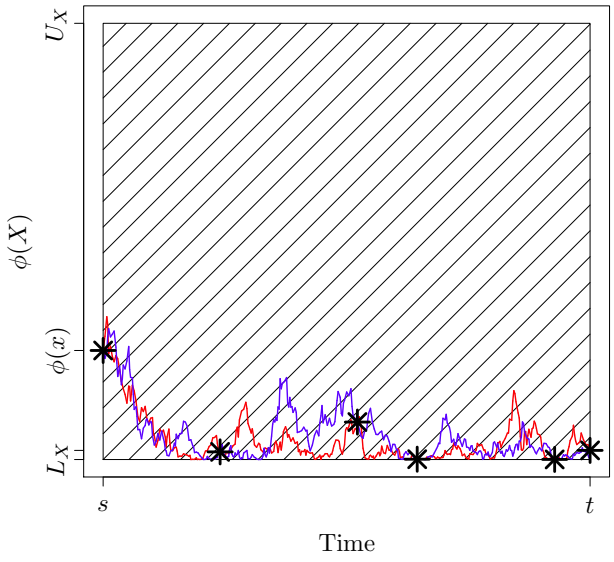
It transpires that the computational cost of simulating a sample path doesn't scale linearly with interval length. However, this problem can be addressed by exploiting the fact that sample paths can be simulated by successive simulation of sample paths of shorter length over the required interval by applying the strong Markov property, noting the Radon-Nikodým derivative in (4.44) decomposes as follows (for any $t \in [0,T]$),

$$\frac{\mathrm{d}\mathbb{Q}_{0,T}^x}{\mathrm{d}\mathbb{W}_{0,T}^x}(X) = \left[\exp\left\{ A(X_t) - A(x) - \int_0^t \phi(X_s)\,\mathrm{d}s \right\} \right] \cdot \left[\exp\left\{ A(X_T) - A(X_t) - \int_t^T \phi(X_s)\,\mathrm{d}s \right\} \right]. \tag{5.13}$$

A direct application of this decomposition is used in Section 5.3.



(a) Example sample path skeleton $\mathcal{S}_{\text{UEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{UEA}}(X)$, and example sample path trajectories.

Figure 5.1.1: Illustrative sample path skeleton output from the Unbounded Exact Algorithm (UEA; Algorithm 5.1.3), $\mathcal{S}_{\text{UEA}}(X) = \{(\xi_i, X_{\xi_i})_{i=0}^{\kappa+1}, R\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}}\right) \Big| R$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

5.1.2 Adaptive Unbounded Exact Algorithm

Within this section we outline a novel Adaptive Unbounded Exact Algorithm (AUEA). To motivate this we revisit Algorithm 5.1.2 noting that the acceptance probability (5.6) can be re-written as follows,

$$\begin{aligned} P_{\mathbb{Z}^x_{0,T}} &= \mathbb{E}_h \mathbb{E}_{\mathcal{R}} \mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|R} \left[\exp\left\{ -\int_0^T (\phi(X_s) - L_X) \, \mathrm{d}s \right\} \right] \\ &=: \mathbb{E}_h \mathbb{E}_{\mathcal{R}} \mathbb{E}_{\mathbb{W}^{x,y}_{0,T}|R} \left[e^{-(L_X - \Phi)T} \cdot \tilde{P}_{\mathbb{Z}^x_{0,T}|R}(X) \right]. \end{aligned} \tag{5.14}$$

Now following Algorithm 5.1.3, after simulating layer information (Step 2) and conditionally accepting the proposal sample path in the first (inexpensive) part of the nested rejection sampler (Step 3), the probability of accepting the sample path is,

$$\tilde{P}_{\mathbb{Z}^x_{0,T}|R}(X) \in \left[e^{-(U_X - L_X)T}, 1 \right] \subseteq [0, 1]. \tag{5.15}$$

Reinterpreting the estimator in (5.8) in light of (5.15) and with the aid of Figure 5.1.2, we are exploiting the fact that $\tilde{P}_{\mathbb{Z}^x_{0,T}|R}(X)$ is equal to the probability a Poisson process of intensity 1 on the graph $\mathcal{G}_A := \{(x, y) \in [0, T] \times [L_X, \infty) : y \leq \phi(x)\}$ contains no points. As this is a difficult space 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 (see Section 2.9.2), by first computing $\phi(X)$ at a finite collection of points (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 Beskos et al. [2006a] and formed the basis of the Bounded Exact Algorithm (BEA) discussed in Section 5.1.1.

As an aside, it should be noted that conditional acceptance of the proposal sample path with probability $e^{-(L_X - \Phi)T}$ in Algorithm 5.1.3 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 5.1.2).

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 5.1.3 Step 4. In particular, note that each simulated skeletal point

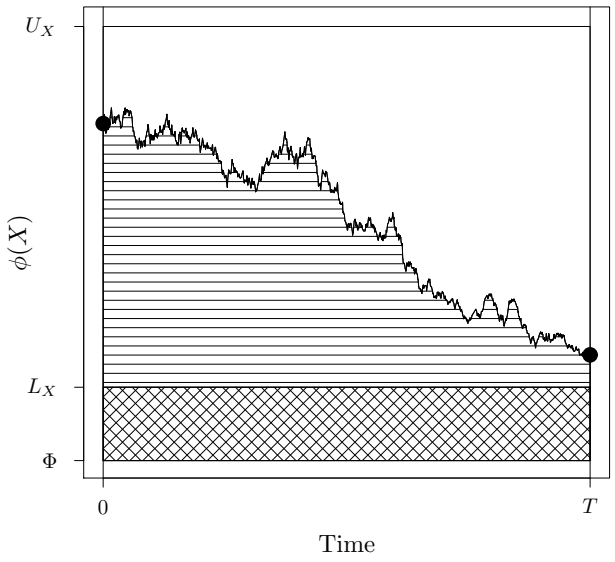


Figure 5.1.2: Example trajectory of $\phi(X)$ where $X \sim \mathbb{W}_{0,T}^{x,y} \,\big|\, R(X)$.

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 proposal $\mathcal{G}_P^M$ 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 5.1.3 Step 4a 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 $(\xi_1, \ldots, \xi_\kappa)$. Alternatively we can exploit the exponential waiting time property between successive events (see Section 2.9.1). In particular, denoting $T_1, \ldots, T_\kappa$ as the time between each event $\xi_1, \ldots, \xi_\kappa$, then the timing of the events can be simulated by successive $\text{Exp}(\Delta_X)$ waiting times while $\sum_i T_i \leq T$.

The independence of events of a 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 (5.15), 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 $u \sim \mathrm{U}[0,1]$),

$$\tilde{P}_{\mathbb{Z}^x_{0,T}|R,X_\xi}(X) = \exp\left\{-\int_0^{T/2-\tau}[\phi(X_s) - L_X]\,\mathrm{d}s - \int_{T/2+\tau}^{T}[\phi(X_s) - L_X]\,\mathrm{d}s\right\} \cdot \mathbb{E}\left(\mathbb{1}\left[u \leq \frac{U_X - \phi(X_\xi)}{U_X - L_X}\right]\middle| X_\xi\right). \tag{5.16}$$

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 (5.16), each of which have the same structural form as we originally had in (5.15). 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 5.1.3. 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 6.3). In particular, considering the left hand sub-interval in (5.16), we can find new layer information (denoted $R_X^{[0,\xi]}$) which will contain tighter bound information regarding the sample path ($\ell_X \leq \ell_X^{[0,\xi]} \leq X_{[0,\xi]}(\omega) \leq \upsilon_X^{[0,\xi]} \leq \upsilon_X$) and so (as a consequence of Result 4) can be used to compute tighter bounds for $\phi(X_{[0,\xi]})$ (denoted $U_X^{[0,\xi]}$ ($\leq U_X$) and $L_X^{[0,\xi]}$ ($\geq L_X$)),

$$\begin{aligned}\tilde{P}^{[0,\xi]}_{\mathbb{Z}^x_{0,\xi}|R_X^{[0,\xi]},X_\xi}(X) &= \exp\left\{-\int_0^{\frac{T}{2}-\tau}[\phi(X_s) - L_X]\,\mathrm{d}s\right\} \\ &= \exp\left\{-\left(L_X^{[0,\xi]} - L_X\right)\left(\frac{T}{2} - \tau\right)\right\}\exp\left\{-\int_0^{\frac{T}{2}-\tau}\left[\phi(X_s) - L_X^{[0,\xi]}\right]\mathrm{d}s\right\}.\end{aligned} \tag{5.17}$$

The left hand hand exponential in (5.17) is a constant and it is trivial to immediately reject the entire path with the complement of this probability. The right hand exponential of (5.17) has the same form as (5.15) 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 $\Delta_X^{[0,\xi]}$ $\left(:= U_X^{[0,\xi]} - L_X^{[0,\xi]} \leq \Delta_X\right)$. As a consequence, the expected number of intermediate points required in order to evaluate the acceptance probability in (5.15) is lower than the Unbounded Exact Algorithm (UEA) in Algorithm 5.1.3.

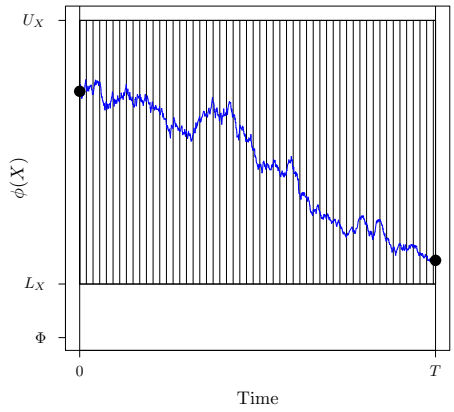
This leads to the novel AUEA detailed in Algorithm 5.1.4, the recursive nature of the algorithm being illustrated in Figure 5.1.3 which is an extension to the example in Figure 5.1.2. We outline how to simulate (unbiasedly) layer information (Step 2), intermediate skeletal points (Step 4(b)ii) and new layer information (Step 4(b)iv) in a variety of ways in Section 6.3. Our iterative scheme outputs a skeleton comprising skeletal points and layer information for the intervals between consecutive skeletal points. 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 5.1.4 Step 4(b)ii and Step 4(b)iv), without any augmentation of the skeleton (as in Algorithm 5.1.3). If necessary, further refinement of the layers given the additionally simulated points can be performed as outlined in Section 6.3. An illustrative example of an accepted sample path skeleton simulated by means of the AUEA is given in Figure 5.1.4.

$$\mathcal{S}_{\text{AUEA}}(X) := \left\{ \left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1} \right\} \tag{5.18}$$

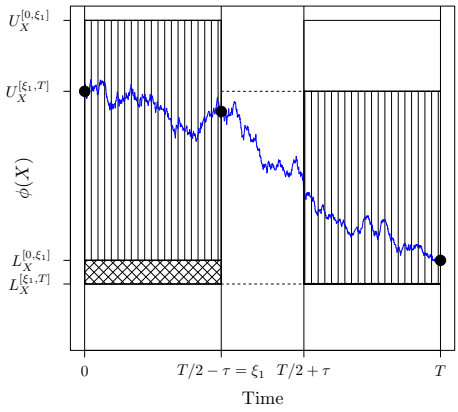
In Algorithm 5.1.4 we introduce simplifying notation, motivated by its recursive nature in which (as shown in (5.16)) the acceptance probability is iteratively decomposed into intervals which have been estimated and are yet to be estimated. Π denotes the set comprising information required to evaluate the acceptance probability for each of the intervals still to be estimated, $\Pi := \{\Pi_i\}_{i=1}^{|\Pi|}$. Each Π_i contains information regarding the time interval it applies to, the sample path at known points at either side of this interval and the associated simulated layer information, which we denote $[s_{\Pi(i)}, t_{\Pi(i)}]$, $x_{\Pi(i)} := X_{s-}^{\Pi(i)}$, $y_{\Pi(i)} := X_{t+}^{\Pi(i)}$ and $R_X^{\Pi(i)}$ respectively (where $s_{\Pi(i)}^- \leq s_{\Pi(i)} < t_{\Pi(i)} \leq t_{\Pi(i)}^+$). This distinction is necessary as known points of the sample path at either end of a given sub-interval do not necessarily align with the end points of the sub-intervals corresponding to the remaining probabilities requiring simulation. 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 $\Xi := \Pi_1$.

5.1.2.1 Implementational Considerations – Known Intermediate Points

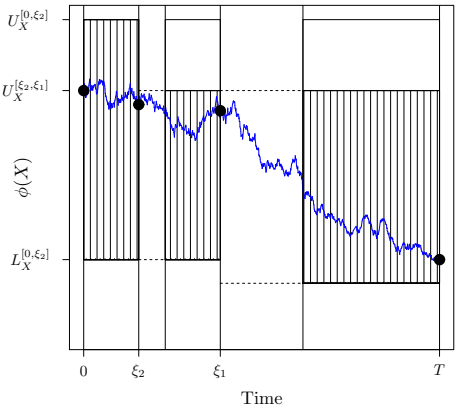
It should be noted that if a number of intermediate points of a sample path are required and the time points at which they occur are known in advance, then rather than simulating them after the acceptance of the sample path skeleton in Algorithm 5.1.4 Step 6, their simulation can be incorporated into Algorithm 5.1.4. In particular, if these points are simulated immediately after Algorithm 5.1.4 Step 3 (this can be performed using Algorithm 6.3.7), 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 5.1.2). The drawback of this approach is that these additional points of the sample path constitute part of the resulting skeleton.



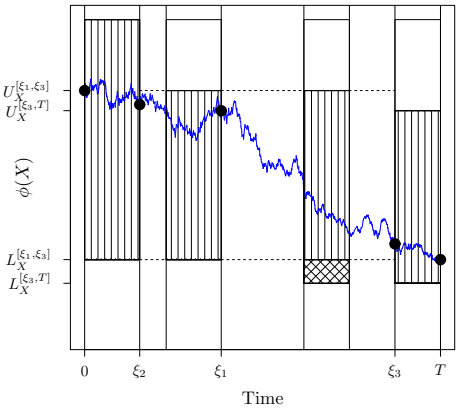
(a) After preliminary acceptance (Algorithm 5.1.4 Step 3)



(b) After simulating ξ_1 (Algorithm 5.1.4 Step 4)



(c) After simulating ξ_2



(d) After simulating ξ_3

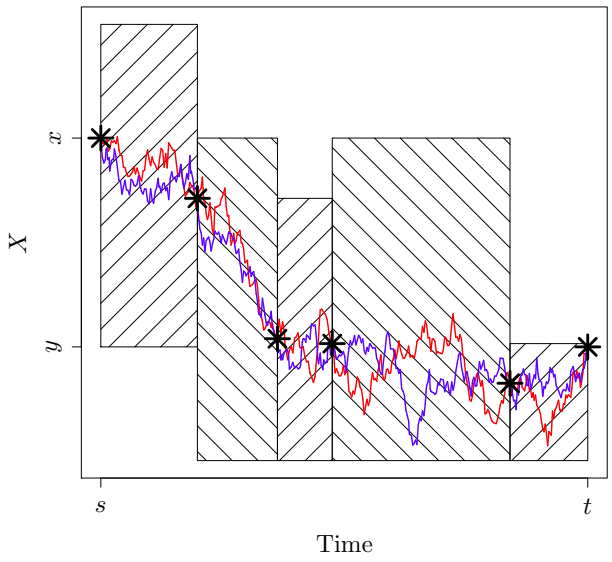
Figure 5.1.3: AUEA applied to the trajectory of $\phi(X)$ in Figure 5.1.2 (where $X \sim \mathbb{W}_{0,T}^{x,y}\Big|R(X)$).

Algorithm 5.1.4 Adaptive Unbounded Exact Algorithm (AUEA).

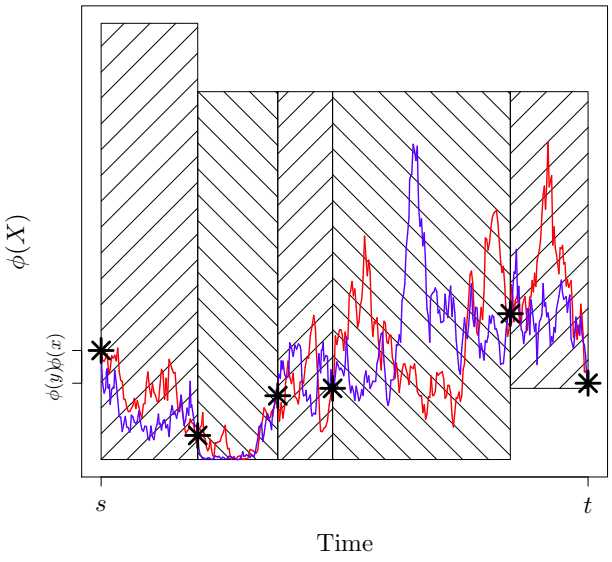
1. Simulate skeleton end point $X_T := y \sim h$.
2. Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\Pi := \{\Xi\} := \{\{[0,T], X_0, X_T, R_X\}\}$ and $\kappa = 0$.
3. With probability $(1 - \exp\{-(L_X - \Phi)T\})$ reject path and return to Step 1.
4. While $\left|\Pi\right| \neq 0$,
   (a) Set $\Xi = \Pi_1$.
   (b) Simulate $\tau \sim \mathrm{Exp}\left(2\Delta_X^{\Xi}\right)$. 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 $X_{\xi'_\kappa} \sim \mathbb{W}^{x(\Xi),y(\Xi)}_{s-(\Xi),t+(\Xi)} \left| R_X^{\Xi}\right.$.
      iii. With probability $\left(1 - \left[U_X^{\Xi} - \phi\left(X_{\xi'_\kappa}\right)\right] / \Delta_X^{\Xi}\right)$ reject path and return to Step 1.
      iv. Simulate new layer information $R_X^{\left[s-(\Xi),\xi'_\kappa\right]}$ and $R_X^{\left[\xi'_\kappa,t+(\Xi)\right]}$ conditional on R_X^{Ξ}.
      v. With probability $\left(1 - \exp\left\{-\left[L_X^{\left[s-(\Xi),\xi'_\kappa\right]} + L_X^{\left[\xi'_\kappa,t+(\Xi)\right]} - 2L_X^{\Xi}\right][d_{\Xi} - \tau]\right\}\right)$ reject path and return to Step 1.
      vi. Set $\Pi := \Pi \bigcup \left\{[s_{\Xi}, m_{\Xi} - \tau], X^{\Xi}_{s-}, X_{\xi'_\kappa}, R_X^{\left[s-(\Xi),\xi'_\kappa\right]}\right\} \bigcup \left\{[m_{\Xi} + \tau, t_{\Xi}], X_{\xi'_\kappa}, X^{\Xi}_{t+}, R_X^{\left[\xi'_\kappa,t+(\Xi)\right]}\right\} \setminus \Xi$.
5. Define skeletal points $\xi_1, \ldots, \xi_\kappa$ as the order statistics of the set $\left\{\xi'_1, \ldots, \xi'_\kappa\right\}$.

---

6. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i} \left| R_X^{[\xi_{i-1},\xi_i]}\right.\right)$.



(a) Example sample path skeleton $\mathcal{S}_{\text{AUEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{AUEA}}(X)$, and example sample path trajectories.

Figure 5.1.4: Illustrative sample path skeleton output from the Adaptive Unbounded Exact Algorithm (AUEA; Algorithm 5.1.4), $\mathcal{S}_{\text{AUEA}}(X) := \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1}\right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}} \,\middle|\, R_X^{[\xi_{i-1},\xi_i]}\right)$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

5.2 Exact Algorithms for Conditioned Diffusions

In this section we outline how to simulate skeletons of diffusion bridge sample paths which can be represented as the solution to the following SDE,

$$\mathrm{d}X_t = \alpha(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \quad X_0 = x \in \mathbb{R},\, X_T = y \in \mathbb{R},\, t \in [0, T]. \tag{5.19}$$

A diffusion bridge is simply a diffusion which in addition to being conditioned to have a given start point is also conditioned to have some specified end point (note in (5.19) we have that $X_T = y$).

Denoting by $\mathbb{Q}^{x,y}_{0,T}$ as the measure induced by (5.19) then, as noted in Beskos et al. [2006a], a similar rejection sampling approach as presented in Section 5.1 can be employed to draw sample path skeletons from $\mathbb{Q}^{x,y}_{0,T}$. In particular, proposal sample path skeletons can be drawn from the equivalent measure $\mathbb{W}^{x,y}_{0,T}$ (Brownian bridge measure as in Section 2.8) and accepted with probability proportional to the Radon-Nikodým derivative of $\mathbb{Q}^{x}_{0,T}$ with respect to $\mathbb{W}^{x}_{0,T}$.

We begin by noting that as a consequence of Corollary 4.1.2 and Theorem 5.1.1 we have that under Conditions 1–3 the Radon-Nikodým derivative of $\mathbb{Q}^{x,y}_{0,T}$ with respect to $\mathbb{W}^{x,y}_{0,T}$ is as follows,

$$\frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}(X) \propto \exp\left\{-\int_0^T \phi(X_s)\,\mathrm{d}s\right\}. \tag{5.20}$$

Noting that $\mathrm{d}\mathbb{Q}^{x,y}_{0,T}/\,\mathrm{d}\mathbb{W}^{x,y}_{0,T}(X) = \mathrm{d}\mathbb{Q}^{x}_{0,T}/\,\mathrm{d}\mathbb{Z}^{x}_{0,T}(X)$ (see (5.3)) then simulating a diffusion bridge skeleton can be performed in precisely the same manner as a diffusion sample path skeleton after the simulation of the end point from Biased brownian motion. As such we can proceed directly to presenting the Conditioned Unbounded Exact Algorithm (CUEA) in Algorithm 5.2.1 and the Adaptive Unbounded Exact Algorithm (CAUEA) in Algorithm 5.2.2, which are direct extensions of Algorithm 5.1.3 and Algorithm 5.1.4 respectively and output skeletons with the same structure as their unconditioned diffusion counterparts,

$$\mathcal{S}_{\mathrm{CUEA}}(X) := \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, R\right\}, \quad \mathcal{S}_{\mathrm{CAUEA}}(X) := \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1}\right\}. \tag{5.21}$$

In Figures 5.2.1 and 5.2.2 we present illustrative examples of accepted sample path skeletons simulated by means of the CUEA and CAUEA respectively.

Algorithm 5.2.1 Conditioned Unbounded Exact Algorithm (CUEA).

---

1. Simulate layer information $R \sim \mathcal{R}$.
2. With probability $(1 - \exp\{-(L_X - \Phi)T\})$ reject path and return to Step 1.
3. Simulate skeleton points $\left(X_{\xi_1}, \ldots, X_{\xi_\kappa}\right) \Big| R$,
   (a) Simulate $\kappa \sim \text{Poi}((U_X - L_X)T)$ and skeleton times $\xi_1, \ldots, \xi_\kappa \overset{\text{iid}}{\sim} \text{U}[0, T]$.
   (b) Simulate sample path at skeleton times $X_{\xi_1}, \ldots, X_{\xi_\kappa} \sim \mathbb{W}_{0,T}^{x,y} \Big| R$.
4. With probability $\prod_{i=1}^{\kappa} \left[\left(U_X - \phi(X_{\xi_i})\right) / (U_X - L_X)\right]$, accept entire path, else reject and return to Step 1.

---

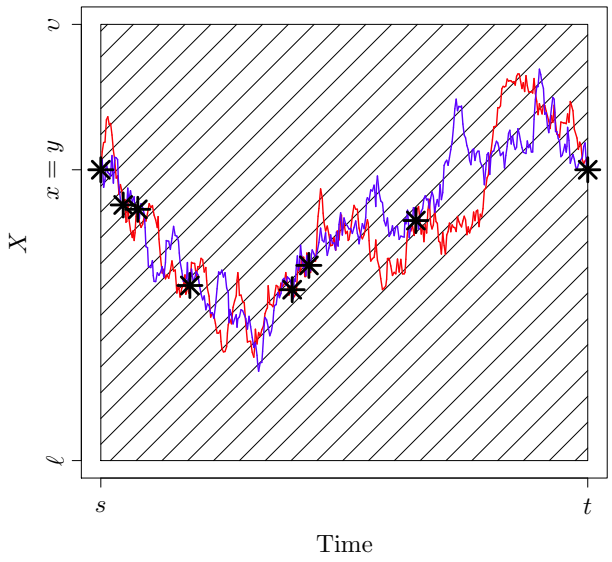
5. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}}\right) \Big| R$.

Algorithm 5.2.2 Conditioned Adaptive Unbounded Exact Algorithm (CAUEA).

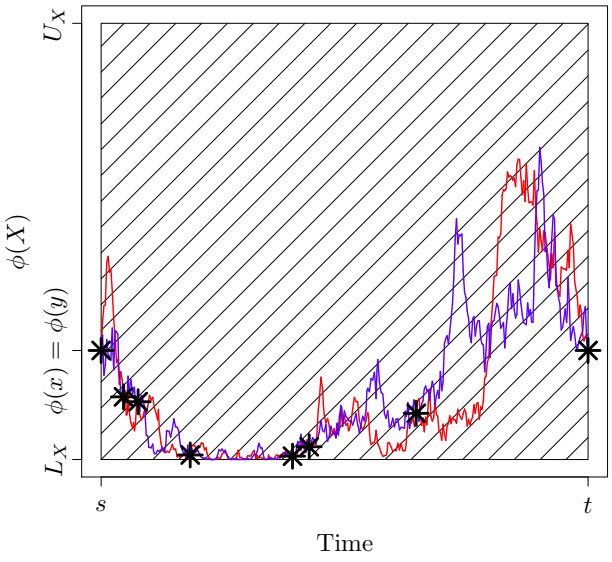
1. Simulate initial layer information $R_X \sim \mathcal{R}$, setting $\Pi := \{\Xi\} := \{\{[0,T], X_0, X_T, R_X\}\}$ and $\kappa = 0$.
2. With probability $(1 - \exp\{-(L_X - \Phi)T\})$ reject path and return to Step 1.
3. While $\left|\Pi\right| \neq 0$,
   (a) Set $\Xi = \Pi_1$.
   (b) Simulate $\tau \sim \text{Exp}\left(2\Delta_X^{\Xi}\right)$. 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 $X_{\xi'_\kappa} \sim \mathbb{W}^{x(\Xi),y(\Xi)}_{s-(\Xi),t+(\Xi)} \left| R_X^{\Xi}\right.$.
      iii. With probability $\left(1 - \left[U_X^{\Xi} - \phi\left(X_{\xi'_\kappa}\right)\right]/\Delta_X^{\Xi}\right)$ reject path and return to Step 1.
      iv. Simulate new layer information $R_X^{\left[s-(\Xi),\xi'_\kappa\right]}$ and $R_X^{\left[\xi'_\kappa,t+(\Xi)\right]}$ conditional on R_X^{Ξ}.
      v. With probability $\left(1 - \exp\left\{-\left[L_X^{\left[s-(\Xi),\xi'_\kappa\right]} + L_X^{\left[\xi'_\kappa,t+(\Xi)\right]} - 2L_X^{\Xi}\right][d_{\Xi} - \tau]\right\}\right)$ reject path and return to Step 1.
      vi. Set $\Pi := \Pi \bigcup \left\{\left[s_{\Xi}, m_{\Xi} - \tau\right], X_{s-}^{\Xi}, X_{\xi'_\kappa}, R_X^{\left[s-(\Xi),\xi'_\kappa\right]}\right\} \bigcup \{[m_{\Xi} + \tau, t_{\Xi}], X_{\xi'_\kappa}, X_{t+}^{\Xi}, R_X^{\left[\xi'_\kappa,t+(\Xi)\right]}\} \setminus \Xi$.
4. Define skeletal points $\xi_1, \ldots, \xi_\kappa$ as the order statistics of the set $\left\{\xi'_1, \ldots, \xi'_\kappa\right\}$.

---

5. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i} \left| R_X^{[\xi_{i-1},\xi_i]}\right.\right)$.

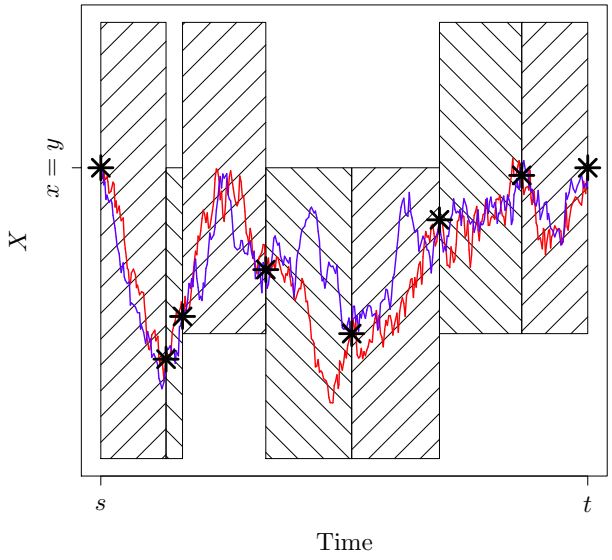


(a) Example sample path skeleton $\mathcal{S}_{\text{CUEA}}(X)$, overlaid with example sample path trajectories.

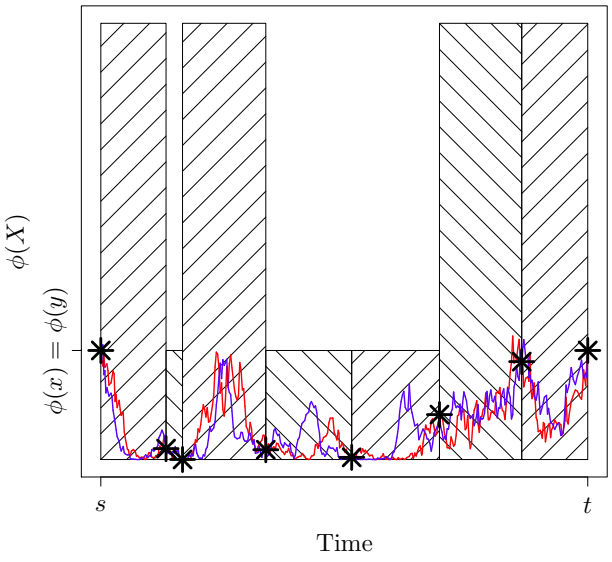


(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{CUEA}}(X)$, and example sample path trajectories.

Figure 5.2.1: Illustrative sample path skeleton output from the Conditioned Unbounded Exact Algorithm (CUEA; Algorithm 5.2.1), $\mathcal{S}_{\text{CUEA}}(X) = \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, R\right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}}\right) \Big| R$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.



(a) Example sample path skeleton $\mathcal{S}_{\text{CAUEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{CAUEA}}(X)$, and example sample path trajectories.

Figure 5.2.2: Illustrative sample path skeleton output from the Conditioned Adaptive Unbounded Exact Algorithm (CAUEA; Algorithm 5.2.2), $\mathcal{S}_{\text{CAUEA}}(X) = \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1}\right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{\kappa+1} \mathbb{W}_{\xi_{i-1},\xi_i}^{X_{\xi_{i-1}},X_{\xi_i}} \,\middle|\, R_X^{[\xi_{i-1},\xi_i]}\right)$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

5.3 Exact Algorithms for Unconditioned Jump Diffusions

In this section we extend the methodology of Section 5.1, constructing exact algorithms for simulating skeletons of *jump diffusion* sample paths which can be represented as the solution to the following SDE (denoting $X_{t-} := \lim_{s \uparrow t} X_s$),

$$\mathrm{d}X_t = \alpha(X_{t-})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \quad X_0 = x \in \mathbb{R}, \;\; t \in [0,T]. \tag{5.22}$$

Denoting by $\mathbb{Q}^x_{0,T}$ the measure induced by (5.22), 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 (5.22)), 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 ((4.42) in Corollary 4.1.1) differs from that for diffusions (4.44) with the inclusion of an additional term, so the methodology of Section 5 can't be applied. However, as discussed in Corollary 4.1.1, (4.43) can be re-expressed in a product form similar to (5.13) (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 $N_t := \sum_{i \geq 1} \mathbb{1}\{\psi_i \leq t\}$),

$$\frac{\mathrm{d}\mathbb{Q}^x_{0,T}}{\mathrm{d}\mathbb{W}^x_{0,T}}(X) = \prod_{i=1}^{N_T+1} \left[\exp\left\{ A(X_{\psi_i-}) - A(X_{\psi_{i-1}}) - \int_{\psi_{i-1}}^{\psi_i-} \phi(X_s)\,\mathrm{d}s \right\} \right]. \tag{5.23}$$

This form of the Radon-Nikodým derivative is the key to constructing *Jump Exact Algorithms* (JEA). Recall that in Section 5.1.1.1, 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 (5.13)). 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 JEAs. In contrast with existing algorithms (Casella and Roberts [2010], Giesecke and Smelov [Forthcoming] and Gonçalves and Roberts [2013]), we note that the Bounded, Unbounded and Adaptive Unbounded Exact Algorithms in Section 5.1 can all be incorporated within any of the JEAs we develop (with an appropriate choice of layered Brownian bridge construction from Chapter 6). In Section 5.3.1 we present the Bounded Jump Exact Algorithm (BJEA), which is a reinterpretation and methodological extension of Casella and Roberts [2010], addressing the case where there exists an explicit bound for the intensity of the jump process. In Section 5.3.2 we

present the Unbounded Jump Exact Algorithm (UJEA) which is an extension to existing exact algorithms (as introduced in Giesecke and Smelov [Forthcoming] and Gonçalves and Roberts [2013]) in which the jump intensity is only locally bounded. Finally, in Section 5.3.3 we introduce an entirely novel Adaptive Unbounded Jump Exact Algorithm (AUJEA) based on the adaptive approach of Section 5.1.2.

5.3.1 Bounded Jump Intensity Jump Exact Algorithm

The case where the jump diffusion we want to simulate (5.22) has an explicit jump intensity bound ($\sup_{u\in[0,T]} \lambda(X_u) \leq \Lambda < \infty$) is of specific interest as the proposal jump times can be simulated in advance. 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 $[0, T]$ and then a simple Poisson thinning argument can be used to accept proposal jump times with probability $\lambda(X_{\Psi_i})/\Lambda$ (as detailed in Section 2.9.2 by application of Algorithm 2.9.3 or Algorithm 2.9.4). As noted in Casella and Roberts [2010], this approach allows the construction of a highly efficient algorithmic interpretation of the decomposition in (5.23). The interval can be broken into segments corresponding precisely to the intervals between proposal jump times, then iteratively between successive times, an exact algorithm (as outlined in Section 5.1) can be used to simulate a diffusion sample path skeleton. The terminal point of each skeleton can be used to determine whether the proposal jump time is accepted (and if so a jump simulated).

The Bounded Jump Exact Algorithm (BJEA) we outline in Algorithm 5.3.1 is a modification of that originally proposed in Casella and Roberts [2010] (where we define $\Psi_0 := 0$ and $\Psi_{N_T^\Lambda+1} := T$). In particular, we simulate the proposal jump times iteratively (exploiting the exponential waiting time property of Poisson processes as in Section 2.9.1), noting that the best proposal distribution may be different for each sub-interval. Furthermore, we note that any of the exact algorithms introduced in Section 5.1 can be incorporated in the BJEA (and so the BJEA will satisfy at least Principle 1 and Principle 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 3), we either augment the skeleton if the exact algorithm chosen is the Unbounded Exact Algorithm (UEA) (as discussed in Sections 5.1.1 and 6.2), or, if the exact algorithm chosen is the Adaptive Unbounded Exact Algorithm

(AUEA) then simulate them directly (as discussed in Sections 5.1.2 and 6.3),

$$\mathcal{S}_{\text{BJEA}}(X) := \bigcup_{j=1}^{N_T^{\Lambda}+1} \mathcal{S}_{\text{EA}}^{j}(X). \tag{5.24}$$

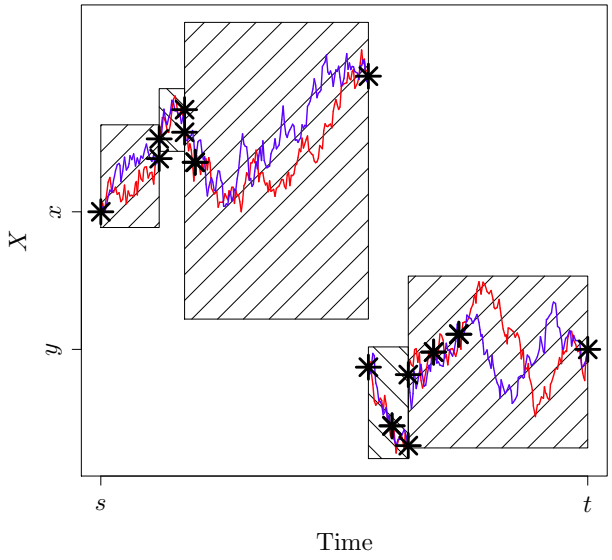
An illustrative example of an accepted sample path skeleton simulated by means of the UEA incorporated within the BJEA is given in Figure 5.3.1.

Algorithm 5.3.1 Bounded Jump Exact Algorithm (BJEA).

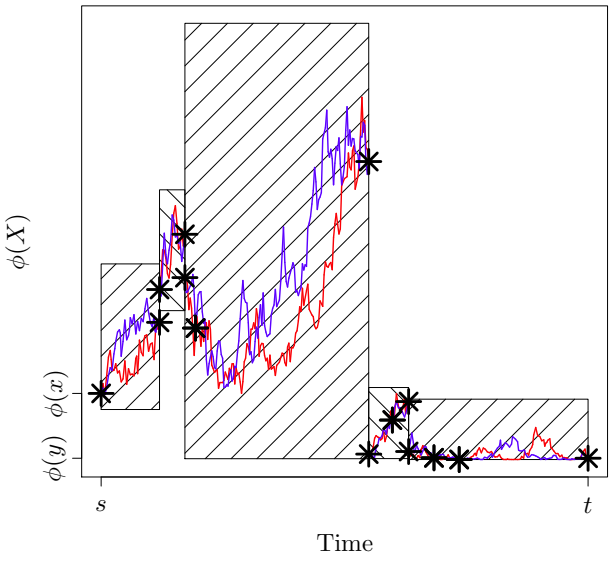
1. Set $j = 0$. While $\Psi_j < T$,
   (a) Simulate $\tau \sim \text{Exp}(\Lambda)$. Set $j = j + 1$ and $\Psi_j = \Psi_{j-1} + \tau$.
   (b) Apply exact algorithm to the interval $\left[\Psi_{j-1}, \left(\Psi_j \wedge T\right)\right)$, obtaining skeleton $\mathcal{S}_{\text{EA}}^{j}$.
   (c) If $\Psi_j > T$ then set $X_T = X_{T-}$ else,
      i. With probability $\lambda(X_{\Psi_i})/\Lambda$ set $X_{\Psi_j} := X_{\Psi_j-} + f_\nu\left(X_{\Psi_j-}\right)$ else set $X_{\Psi_j} := X_{\Psi_j-}$.
2. Accept sample path skeleton.

---

3. $*$ *Simulate* $X^{rem} \sim \otimes_{j=1}^{N_T^{\Lambda}+1} \left(\otimes_{i=1}^{\kappa_j+1} \mathbb{W}_{\xi_{j,i-1},\xi_{j,i}}^{X_{\xi_{j,i-1}}, X_{\xi_{j,i}}} \,\middle|\, R^{EA}_{X[\xi_{j,0},\xi_{j,\kappa_j+1}]} \right)$.



(a) Example sample path skeleton $\mathcal{S}_{\text{BJEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{BJEA}}(X)$, and example sample path trajectories.

Figure 5.3.1: Illustrative sample path skeleton output from the Unbounded Exact Algorithm (UEA; Algorithm 5.1.3) incorporated within the Bounded Jump Exact Algorithm (BJEA; Algorithm 5.3.1), $\mathcal{S}_{\text{BJEA}}(X) = \bigcup_{j=1}^{N_T^{\Lambda}+1} \left\{ \left(\xi_i^j, X_{\xi_i^j}\right)_{i=0}^{\kappa_j+1}, R_X^{[\Psi_{j-1},\Psi_j]} \right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \mathbb{W}_{0,T}^{x,y} \left| \mathcal{S}_{\text{BJEA}} \right.$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

5.3.2 Unbounded Jump Intensity Jump Exact Algorithm

Considering the construction of a JEA under the weaker Condition 4 (in which we assume only that the jump intensity in (5.22) is locally bounded), it is not possible to first simulate the jump times as in Section 5.3.1. However, (as in Section 5.1 and as noted in Giesecke and Smelov [Forthcoming] and Gonçalves and Roberts [2013]), 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 UEA and AUEA.

The Unbounded Jump Exact Algorithm (UJEA) which we present in Algorithm 5.3.2 is a JEA construction based on the UEA and extended from Gonçalves and Roberts [2013]. The UJEA is necessarily more complicated than the 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 Principle 1 and Principle 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.

$$\mathcal{S}_{\text{UJEA}}(X) := \bigcup_{j=0}^{N_T^{\lambda}} \left\{ \left(\xi_i^j, X_{\xi_i^j} \right)_{i=0}^{\kappa_j+1}, \left(\Psi_1^j, X_{\Psi_1^j} \right)_{i=1}^{N_T^{\Lambda,j}}, R_{X[\psi_j,T]}^j \right\}. \tag{5.25}$$

The UJEA doesn't satisfy Principle 3 unless the skeleton is augmented (as with the UEA outlined in Sections 5.1.1 and 6.2). 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 5.3.3.

Algorithm 5.3.2 Unbounded Jump Exact Algorithm (UJEA).

1. Set $j = 0$ and $\psi_j = 0$,
   (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) With probability $\left(1 - \exp\left\{-\left(L^j_{X[\psi_j,T]} - \Phi\right) \cdot (T - \psi_j)\right\}\right)$ reject path and return to Step 1a.
   (d) Simulate proposal jump times $N_T^{\Lambda,j} \sim \text{Poi}\left(\Lambda^j_{X[\psi_j,T]}(T - \psi_j)\right)$ and $\Psi^j_1, \ldots, \Psi^j_{N_T^{\Lambda,j}} \overset{\text{iid}}{\sim} \text{U}[\psi_j, T]$.
   (e) Simulate skeleton points and diffusion at proposal jump times $\left(X_{\xi^j_1}, \ldots, X_{\xi^j_\kappa}, X_{\Psi^j_1}, \ldots, X_{\Psi^j_{N(\Lambda,j,T)}}\right) \Big| R^j_{X[\psi_j,T]}$,
      i. Simulate $\kappa_j \sim \text{Poi}\left(\left[U^j_{X[\psi_j,T]} - L^j_{X[\psi_j,T]}\right] \cdot (T - \psi_j)\right)$ and skeleton times $\xi^j_1, \ldots, \xi^j_\kappa \overset{\text{iid}}{\sim} \text{U}[\psi_j, T]$.
      ii. Simulate sample path at times $X_{\xi^j_1}, \ldots, X_{\xi^j_\kappa}, X_{\Psi^j_1}, \ldots, X_{\Psi^j_{N(\Lambda,j,T)}} \sim \mathbb{W}^{x,y}_{\psi_j,T} \Big| R^j_{X[\psi_j,T]}$.
   (f) With probability $\left(1 - \prod_{i=1}^{\kappa_j}\left[\left(U^j_{X[\psi_j,T]} - \phi\left(X_{\xi^j_i}\right)\right) / \left(U^j_{X[\psi_j,T]} - L^j_{X[\psi_j,T]}\right)\right]\right)$, reject and return to Step 1a.
   (g) For i in 1 to $N_T^{\Lambda,j}$,
      i. With probability $\lambda(X_{\Psi^j_i})/\Lambda^j_{X[\psi_j,T]}$ set $X_{\Psi^j_i-} = X_{\Psi^j_i}$, $X_{\Psi^j_i} := X_{\Psi^j_i-} + f_\nu\left(X_{\Psi^j_i}\right)$, $\psi_{j+1} := \Psi^j_i$, $j = j + 1$, and return to Step 1a.
2. Accept sample path skeleton.

---

3. * *Simulate* $X^{rem} \sim \otimes_{j=0}^{N_T^\lambda}\left[\left(\otimes_{i=1}^{\kappa_j+1} \mathbb{W}^{X_{\xi_{j,i-1}}, X_{\xi_{j,i}}}_{\xi_{j,i-1},\xi_{j,i}}\right) \Big| R^j_{X[\psi_j,T]}\right]$.

5.3.3 Adaptive Unbounded Jump Intensity Jump Exact Algorithm

The novel Adaptive Unbounded Jump Exact Algorithm (AUJEA) which we present in Algorithm 5.3.3 is based on the AUEA and a reinterpretation of the 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 5.3.2 Step 1(e)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 5.1.2), provides further information about the sample path. In particular, the proposal jump time necessarily lies between two existing 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 6.3.5). 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,

$$\mathcal{S}_{\text{AUJEA}}(X) := \bigcup_{j=1}^{N_T^\lambda+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{j-1},\psi_j)}(X). \tag{5.26}$$

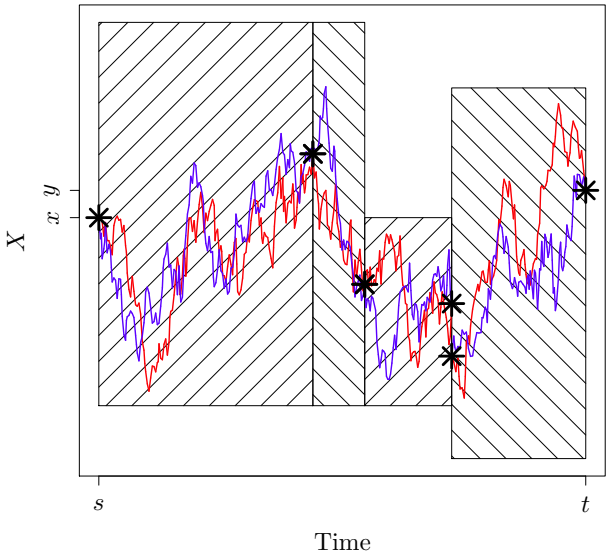
An illustrative example of an accepted sample path skeleton simulated by means of the AUJEA is given in Figure 5.3.2.

Algorithm 5.3.3 Adaptive Unbounded Jump Exact Algorithm (AUJEA).

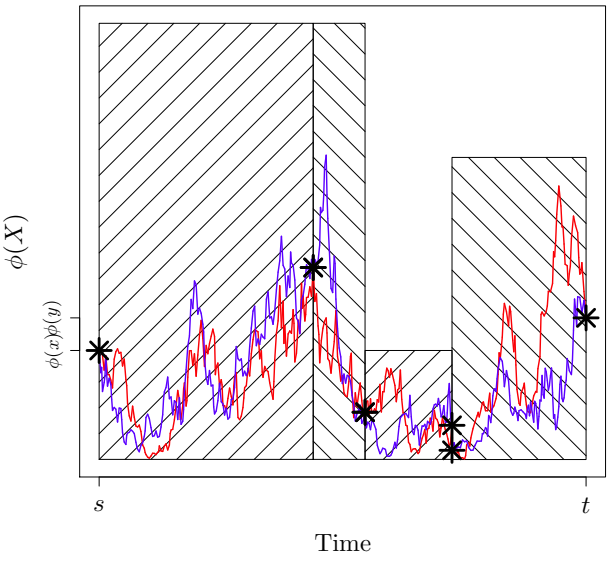
1. Set $j = 0$ and $\psi_j = 0$.
2. Apply AUEA to interval $\left[\psi_j, T\right]$, obtaining skeleton $\mathcal{S}_{\text{AUEA}}^{[\psi_j,T]}$.
3. Set $k = 0$ and $\Psi_k^j = \psi_j$. While $\Psi_k^j < T$,
   (a) Compute $\Lambda^j_{X[\Psi_k^j,T]}$.
   (b) Simulate $\tau \sim \text{Exp}\left(\Lambda^j_{X[\Psi_k^j,T]}\right)$. Set $k = k + 1$ and $\Psi_k^j = \Psi_{k-1}^j + \tau$.
   (c) If $\Psi_k^j \leq T$,
      i. Simulate $X_{\Psi_k^j} \sim \mathbb{W}^{X_{\psi_j},X_T}_{\psi_j,T} \left| \mathcal{S}_{\text{AUEA}}^{[\psi_j,T]} \right.$.
      ii. Simulate $R_X^{[\xi_-^j,\Psi_k^j]}$ and $R_X^{[\Psi_k^j,\xi_+]}$ and set $\mathcal{S}_{\text{AUEA}}^{[\psi_j,T]} := \mathcal{S}_{\text{AUEA}}^{[\psi_j,T]} \cup \left\{X_{\Psi_k^j}, R_X^{[\xi_-^j,\Psi_k^j]}, R_X^{[\Psi_k^j,\xi_+]}\right\} \setminus R_X^{[\xi_-^j,\xi_+]}$.
      iii. With probability $\lambda(X_{\Psi_k^j})/\Lambda^j_{X[\Psi_{k-1}^j,T]}$ set $X_{\Psi_k^{j-}} = X_{\Psi_k^j}$, $X_{\Psi_k^j} := X_{\Psi_k^{j-}} + f_\nu\left(X_{\Psi_k^j}\right)$, $\psi_{j+1} := \Psi_k^j$, retain $\mathcal{S}_{\text{AUEA}}^{[\psi_j,\psi_{j+1})}$, discard $\mathcal{S}_{\text{AUEA}}^{[\psi_{j+1},T]}$, set $j = j + 1$ and return to Step 2.
4. Accept sample path skeleton.
5. Let skeletal points $\chi_1, \ldots, \chi_m$ denote the order statistics of the time points in $\mathcal{S}_{\text{AUJEA}} := \bigcup_{i=1}^{j+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{i-1},\psi_i)}$.

---

6. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{m+1} \mathbb{W}^{X_{\chi_{i-1}},X_{\chi_i}}_{\chi_{i-1},\chi_i} \left| R_X^{[\chi_{i-1},\chi_i]} \right.\right)$.



(a) Example sample path skeleton $\mathcal{S}_{\text{AUJEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{AUJEA}}(X)$, and example sample path trajectories.

Figure 5.3.2: Illustrative sample path skeleton output from the Adaptive Unbounded Jump Exact Algorithm (AUJEA; Algorithm 5.3.3), $\mathcal{S}_{\text{AUJEA}}(X) := \bigcup_{j=1}^{N_T^{\lambda}+1} \mathcal{S}_{\text{AUEA}}^{[\psi_{j-1},\psi_j)}(X)$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{m+1} \mathbb{W}_{\chi_{i-1},\chi_i}^{X_{\chi_{i-1}},X_{\chi_i}} \,\middle|\, R_X^{[\chi_{i-1},\chi_i]} \right)$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

5.3.4 An Extension to the Unbounded & 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 with the 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 (5.13)), 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 (see Kingman [1992]). 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 $\Lambda_X - \lambda\downarrow$.

5.4 Exact Algorithms for Conditioned Jump Diffusions

In this section we extend the methodology of Section 5.1 and Section 5.3 to construct an exact algorithm for simulating skeletons of jump diffusion bridge sample paths which can be represented as the solution to the following SDE (denoting $X_{t-} := \lim_{s \uparrow t} X_s$),

$$\mathrm{d}X_t = \alpha(X_{t-})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \qquad X_0 = x \in \mathbb{R},\, X_T = y \in \mathbb{R},\, t \in [0,T]. \tag{5.27}$$

The approach we take in this section in constructing our exact algorithm is based on the recent methodology developed in Gonçalves and Roberts [2013]. However, we reformulate the exact algorithm presented in Gonçalves and Roberts [2013] to ensure that upon accepting a sample path skeleton then it is possible to simulate the sample path at further finite collections of time points without approximation (i.e. so that it satisfies Principle 3 in addition to Principles 1 and 2). Furthermore, the exact algorithm which we present offers some potential for computational improvement over existing exact algorithms.

To guide intuition, recall that if we were to employ the approach in Section 5.1 for constructing an exact algorithm to simulate sample skeletons from $\mathbb{Q}_{0,T}^{x,y}$ (the measure induced by (5.27)), we require an equivalent proposal measure from which it is possible to propose sample path skeletons (which are then accepted or rejected with probability proportional to the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^{x,y}$ with respect to the chosen proposal measure). As before, a natural equivalent proposal measure to choose is that induced by a driftless diffusion with unit volatility (with measure denoted by $\mathbb{W}_{0,T}^{x,y}$ and where Λ and δ are chosen to satisfy the conditions in Section 4.1.3),

$$\mathrm{d}X_t = \mathrm{d}W_t + \mathrm{d}J_t^{\Lambda,\delta}, \qquad X_0 = x \in \mathbb{R},\, X_T = y \in \mathbb{R},\, t \in [0,T]. \tag{5.28}$$

Unfortunately, $\mathbb{W}_{0,T}^{x,y}$ can't be chosen as a proposal measure in general as it is typically not possible to simulate a compound Poisson process conditioned to hit a specified end point. In particular, in contrast to the simulation of jump diffusions in Section 5.3 (i.e. the unconditioned end point case), the Markov property can't be exploited in simulating proposal sample path skeletons. To illustrate this, note that if we consider a sample path simulated from the measure $\mathbb{W}_{0,T}^{x,y}$ to some intermediate time point, then as the end point is known the number of jumps and size of jumps changes to ensure the end point is hit.

The key contribution of Gonçalves and Roberts [2013] was to note that an alternate equiv-

alent measure (denoted by $\mathbb{G}_{0,T}^{x,y}$) could be constructed to ensure the end point is hit. In particular, if a compound Poisson process is simulated first ($J_{[0,T]}$) then, to ensure the end point is hit ($X_T = y$), a Brownian bridge conditioned to start at $X'_0 = x$ and end at $X'_T = y - J_T$ could be used as the continuous component in the proposal sample path. Considering the superposition of the compound Poisson process sample path and the Brownian bridge sample path ($X_t = J_t + X'_t$), then the resulting sample path starts and ends at the desired points ($X_0 = x$ and $X_T = J_T + (y - J_T) = y$). More formally, the proposal measure is the measure induced by the following SDE,

$$\mathrm{d}X_t = \mathrm{d}Z_t + \mathrm{d}J_t^{\Lambda,\delta}, \qquad X_0 = x \in \mathbb{R},\ X_T = y \in \mathbb{R},\ t \in [0,T], \tag{5.29}$$

where,

$$\mathrm{d}Z_t = \mathrm{d}W_t, \qquad Z_0 = x \in \mathbb{R},\ Z_T = y - J_T \in \mathbb{R},\ t \in [0,T]. \tag{5.30}$$

Note that this approach necessitates that the chosen proposal jump intensity and jump size function (Λ and δ) *can't* be parameterised by any variable which depends upon the continuous component of the sample path.

Now, proceeding as in Section 5.1, as we have an equivalent proposal measure with which we can simulate sample path skeletons from ($\mathbb{G}_{0,T}^{x,y}$), we now require the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^{x,y}$ with respect to $\mathbb{G}_{0,T}^{x,y}$.

Theorem 5.4.1 (Radon-Nikodým derivative for conditioned jump diffusions [Gonçalves and Roberts, 2013, Lemma 2])**.** *$\mathbb{Q}_{0,T}^{x,y}$ is equivalent to $\mathbb{G}_{0,T}^{x,y}$ with Radon-Nikodým derivative:*

$$\begin{aligned}\frac{\mathrm{d}\mathbb{Q}_{0,T}^{x,y}}{\mathrm{d}\mathbb{G}_{0,T}^{x,y}}(X) \propto \exp\left\{-\frac{1}{2}\frac{(y - J_T - x)^2}{T} - \int_0^T \phi(X_{s-})\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\} \\ \cdot \exp\left\{-\int_0^T \left[\lambda(X_{s-}) - \Lambda\right]\,\mathrm{d}s\right\} \cdot \prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right)}{\Lambda \cdot f_\delta\left(X_{\psi_i} - X_{\psi_i-}\right)}.\end{aligned} \tag{5.31}$$

Proof. We begin by noting that we can express the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^{x,y}$

with respect to $\mathbb{G}^{x,y}_{0,T}$ as follows,

$$\begin{aligned}\frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) &= \frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}(X) \cdot \frac{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) \\ &= \frac{w_T(x,y)}{p_T(x,y)} \cdot \frac{\mathrm{d}\mathbb{Q}^{x}_{0,T}}{\mathrm{d}\mathbb{W}^{x}_{0,T}}(X) \cdot \frac{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X). \end{aligned} \tag{5.32}$$

Now, as the only term in (5.32) which we don't have a representation for is the Radon-Nikodým derivative of $\mathbb{W}^{x,y}_{0,T}$ with respect to $\mathbb{G}^{x,y}_{0,T}$ we focus on this. In particular we informally have (denoting by X^{cts} as the continuous component of X),

$$\begin{aligned}\frac{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x,y}_{0,T}(\mathrm{d}X)}{\mathbb{G}^{x,y}_{0,T}(\mathrm{d}X)} = \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x,y}_{0,T}(\mathrm{d}X^{\text{cts}}, \mathrm{d}J_{[0,T]})}{\mathbb{G}^{x,y}_{0,T}(\mathrm{d}X^{\text{cts}}, \mathrm{d}J_{[0,T]})} \\ &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x,y}_{0,T}(\mathrm{d}X^{\text{cts}} \,\big|\, J_{[0,T]}) \cdot \mathbb{W}^{x,y}_{0,T}(\mathrm{d}J_{[0,T]})}{\mathbb{G}^{x,y}_{0,T}(\mathrm{d}X^{\text{cts}} \,\big|\, J_{[0,T]}) \cdot \mathbb{G}^{x,y}_{0,T}(J_{[0,T]})} \\ &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x,y}_{0,T}(\mathrm{d}X^{\text{cts}} \,\big|\, J_{[0,T]}, X_T) \cdot \mathbb{W}^{x}_{0,T}(\mathrm{d}J_{[0,T]} \,\big|\, X_T)}{\mathbb{G}^{x}_{0,T}(\mathrm{d}X^{\text{cts}} \,\big|\, J_{[0,T]}, X_T) \cdot \mathbb{G}^{x}_{0,T}(\mathrm{d}J_{[0,T]} \,\big|\, X_T)}. \end{aligned} \tag{5.33}$$

Noting under the measure $\mathbb{G}^{x}_{0,T}$ we have that for any given sample path the compound Poisson process is independent of the continuous diffusion component we have,

$$\begin{aligned}\frac{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x}_{0,T}(\mathrm{d}J_{[0,T]} \,\big|\, X_T)}{\mathbb{G}^{x}_{0,T}(\mathrm{d}J_{[0,T]})} \\ &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x}_{0,T}(\mathrm{d}X_T \,\big|\, J_{[0,T]}) \cdot \mathbb{W}^{x}_{0,T}(\mathrm{d}J_{[0,T]}) / \mathbb{W}^{x}_{0,T}(\mathrm{d}X_T)}{\mathbb{G}^{x}_{0,T}(\mathrm{d}J_{[0,T]})} \\ &= \lim_{\mathbb{P}(\mathrm{d}X)\downarrow 0} \frac{\mathbb{W}^{x}_{0,T}\left(\mathrm{d}X^{\text{cts}}_T = \mathrm{d}X_T - \mathrm{d}J_T \,\middle|\, J_T\right)}{\mathbb{W}^{x}_{0,T}(\mathrm{d}X_T)} \\ &= \frac{1}{w_T(x,y)} \cdot \frac{1}{\sqrt{2\pi T}} \exp\left\{-\frac{1}{2}\frac{(y - J_T - x)^2}{T}\right\}, \end{aligned} \tag{5.34}$$

additionally noting that $\mathbb{W}^{x,y}_{0,T}$ is equivalent to $\mathbb{G}^{x,y}_{0,T}$. Furthermore, with reference to (5.32) and noting from Theorem 4.1.1 we have that $\mathbb{Q}^{x}_{0,T}$ is equivalent to $\mathbb{W}^{x}_{0,T}$, we can conclude that $\mathbb{Q}^{x,y}_{0,T}$ is equivalent to $\mathbb{G}^{x,y}_{0,T}$ as stated.

Now considering the Radon-Nikodým derivative of $\mathbb{Q}^{x,y}_{0,T}$ with respect to $\mathbb{G}^{x,y}_{0,T}$ we re-

turn to (5.32) and apply the results from Theorems 4.1.1 and 5.1.1, additionally noting that $\mathbb{Q}^x_{0,T}$ is absolutely continuous with respect to $\mathbb{W}^x_{0,T}$, to arrive at our desired result,

$$\begin{aligned}
\frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) &= \frac{w_T(x,y)}{p_T(x,y)} \cdot \frac{\mathrm{d}\mathbb{Q}^{x}_{0,T}}{\mathrm{d}\mathbb{W}^{x}_{0,T}}(X) \cdot \frac{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) \\
&\propto \exp\left\{-\frac{1}{2}\frac{(y-J_T-x)^2}{T}\right\} \cdot \frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{W}^{x}_{0,T}}(X) \\
&\propto \exp\left\{-\frac{1}{2}\frac{(y-J_T-x)^2}{T}\right\} \cdot \frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{W}^{x,y}_{0,T}}(X) \\
&\propto \exp\left\{-\frac{1}{2}\frac{(y-J_T-x)^2}{T} - \int_0^T \phi(X_{s-})\,\mathrm{d}s - \sum_{i=1}^{N_T}\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\} \\
&\quad \cdot \exp\left\{-\int_0^T \left[\lambda(X_{s-}) - \Lambda\right]\,\mathrm{d}s\right\} \cdot \prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right)}{\Lambda \cdot f_\delta\left(X_{\psi_i} - X_{\psi_i-}\right)}. \qquad (5.35)
\end{aligned}$$

□

Now that we have derived the Radon-Nikodým derivative of $\mathbb{Q}^{x,y}_{0,T}$ with respect to $\mathbb{G}^{x,y}_{0,T}$ we can construct an idealised jump diffusion bridge rejection sampler (as in Algorithm 5.1.1 of Section 5.1). In particular, drawing $X \sim \mathbb{G}^{x}_{0,T}$ and accepting the sample path ($I = 1$) with probability $P_{\mathbb{G}^{x,y}_{0,T}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) \in [0,1]$ (where M is a bound such that $\sup_X \frac{\mathrm{d}\mathbb{Q}^{x,y}_{0,T}}{\mathrm{d}\mathbb{G}^{x,y}_{0,T}}(X) \leq M < \infty$) then $(X \,|\, I = 1) \sim \mathbb{Q}^{x,y}_{0,T}$.

In order to find an appropriate bound $M < \infty$ for our idealised rejection sampler, we impose the following condition (in addition to the conditions in Section 1.3).

Condition 6 ($\varkappa$)**.** *We have that* $\exists\, \varkappa < \infty$ *such that,*

$$\frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right) \cdot \exp\left\{-\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}}{\Lambda \cdot f_\delta\left(X_{\psi_i} - X_{\psi_i-}\right)} \leq \varkappa \qquad (5.36)$$

As all exact algorithms are fundamentally rejection samplers on diffusion path space and in this particular scenario the user only has control over the choice of the functions Λ and δ then, in a similar manner to choosing the dominating density in rejection sampling (see Section 2.4), Λ and δ should be chosen to ensure that Condition 6 holds and $\varkappa$ is as close to 1 as possible (i.e. the dominating density should be chosen to closely bound the target

density). It is possible to choose an appropriate Λ and δ such that Condition 6 will hold in a variety of settings, for instance, if the target jump intensity is bounded, the target jump size density is time homogeneous and the function A is Lipschitz.

Now, returning to our idealised rejection sampler, an appropriate bound M can be found for the Radon-Nikodým derivative of $\mathbb{Q}_{0,T}^{x,y}$ with respect to $\mathbb{G}_{0,T}^{x,y}$ by noting that a separate bound for each component of the Radon-Nikodým derivative in (5.31) can be found,

$$\frac{\mathrm{d}\mathbb{Q}_{0,T}^{x,y}}{\mathrm{d}\mathbb{G}_{0,T}^{x,y}}(X) \propto \underbrace{\exp\left\{-\frac{1}{2}\frac{(y-J_T-x)^2}{T}\right\}}_{\leq 1} \cdot \underbrace{\exp\left\{-\int_0^T \phi(X_{s-})\,\mathrm{d}s\right\}}_{\leq \exp\{-\Phi T\}} \cdot \underbrace{\exp\left\{-\int_0^T [\lambda(X_{s-})-\Lambda]\,\mathrm{d}s\right\}}_{\leq \exp\{\Lambda T\}} \cdot \underbrace{\prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right) \cdot \exp\left\{-\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}}{\Lambda \cdot f_\delta\left(X_{\psi_i} - X_{\psi_i-}\right)}}_{\leq \varkappa^{N_T}}, \tag{5.37}$$

and so we can find an explicit representation for the acceptance probability of our idealised rejection sampler,

$$P_{\mathbb{G}_{0,T}^{x,y}}(X) := \frac{1}{1 \cdot \exp\{\Phi T\} \cdot \exp\{\Lambda T\} \cdot \varkappa^{N_T}} \frac{\mathrm{d}\mathbb{Q}_{0,T}^{x,y}}{\mathrm{d}\mathbb{G}_{0,T}^{x,y}}(X) \in [0,1]. \tag{5.38}$$

As in Section 5.1, the idealised rejection sampler described above can't be implemented as to do so would require the simulation of an entire infinite dimensional proposal sample path from $\mathbb{G}_{0,T}^{x,y}$. However, we can once again construct and simulate some finite dimensional auxiliary random variable $F \sim \mathbb{F}$ such that the acceptance probability can be unbiasedly estimated using only a finite dimensional subset of the proposal sample path. In particular we have,

$$P_{\mathbb{G}_{0,T}^{x,y}} := \mathbb{E}_{\mathbb{G}_{0,T}^{x,y}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\right] = \mathbb{E}_{\mathbb{F}}\mathbb{E}_{\mathbb{G}_{0,T}^{x,y}|F}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\right]. \tag{5.39}$$

Now, considering how to construct a suitable finite dimensional random variable $F \sim \mathbb{F}$, note that if we let $\mathcal{J}$ be the law of the compound Poisson process component of $\mathbb{G}_{0,T}^{x,y}$ then upon simulating $J_{[0,T]} \sim \mathcal{J}$ (using for instance Algorithm 2.9.5 or Algorithm 2.9.6)

we have,

$$P_{\mathbb{G}_{0,T}^{x,y}} = \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\mathbb{G}_{0,T}^{x,y}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\,\middle|\,J_{[0,T]}\right]$$
$$= \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\mathbb{G}_{0,T}^{x,y}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\,\middle|\,N_T,\psi_1,\ldots,\psi_{N_T},\delta_{\psi_1},\ldots,\delta_{\psi_{N_T}}\right]. \tag{5.40}$$

Further denoting by $\mathcal{W}\,|\,\mathcal{J}$ as the law induced by simulating $(X'_{\psi_1},\ldots,X'_{\psi_{N_T}}) \sim \tilde{\mathbb{W}}_{0,T}^{x,(y-J_T)}$ (where we denote by $\tilde{\mathbb{W}}_{0,T}^{x,(y-J_T)}$ as Brownian bridge measure as in Section 2.8 and simulated as per Algorithm 2.8.2) then we have,

$$\begin{aligned}
P_{\mathbb{G}_{0,T}^{x,y}} &= \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\mathbb{G}_{0,T}^{x,y}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\,\middle|\,N_T,\psi_1,\ldots,\psi_{N_T},\delta_{\psi_1},\ldots,\delta_{\psi_{N_T}}\right]\\
&= \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\tilde{\mathbb{W}}_{0,T}^{x,(y-J_T)}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\,\middle|\,N_T,\psi_1,\ldots,\psi_{N_T},\delta_{\psi_1},\ldots,\delta_{\psi_{N_T}}\right]\\
&= \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\mathcal{W}\,|\,\mathcal{J}}\mathbb{E}_{\tilde{\mathbb{W}}_{0,T}^{x,(y-J_T)}}\left[P_{\mathbb{G}_{0,T}^{x,y}}(X)\,\middle|\,X_{\psi_1},\ldots,X_{\psi_{N_T}},N_T,\psi_1,\ldots,\psi_{N_T},\delta_{\psi_1},\ldots,\delta_{\psi_{N_T}}\right]\\
&= \mathbb{E}_{\mathcal{J}}\mathbb{E}_{\mathcal{W}\,|\,\mathcal{J}}\mathbb{E}_{\tilde{\mathbb{W}}_{0,T}^{x,(y-J_T)}}\Bigg[\exp\left\{-\frac{1}{2}\frac{(y-J_T-x)^2}{T}\right\}\\
&\qquad\cdot\prod_{i=1}^{N_T}\frac{\lambda(X_{\psi_i-})\cdot f_{\nu}\left(X_{\psi_i};X_{\psi_i-}\right)\cdot\exp\left\{-\left[A(X_{\psi_i})-A(X_{\psi_i-})\right]\right\}}{\Lambda\cdot f_{\delta}\left(X_{\psi_i}-X_{\psi_i-}\right)\cdot\varkappa}\\
&\qquad\cdot\tilde{P}_{\mathbb{G}_{0,T}^{x,y}}(X)\,\Big|\,X_{\psi_1},\ldots,X_{\psi_{N_T}},N_T,\psi_1,\ldots,\psi_{N_T},\delta_{\psi_1},\ldots,\delta_{\psi_{N_T}}\Bigg],
\end{aligned} \tag{5.41}$$

where (denoting by $\psi_0 := 0$ and $\psi_{N_T+1} := T$),

$$\begin{aligned}
\tilde{P}_{\mathbb{G}_{0,T}^{x,y}}(X) &= \exp\left\{-\int_0^T \phi(X_{s-})+\lambda(X_{s-})\,\mathrm{d}s\right\}\cdot e^{\Phi T}\\
&= \prod_{i=1}^{N_T+1}\left(\exp\left\{-\int_{\psi_{i-1}}^{\psi_i}\phi(X_{s-})+\lambda(X_{s-})\,\mathrm{d}s\right\}\cdot e^{\Phi(\psi_i-\psi_{i-1})}\right).
\end{aligned} \tag{5.42}$$

Noting that between any two jump times with known end points that no further jumps occur and the sample path is a Brownian bridge, then each component of (5.42) can be considered directly using the methodology developed Sections 5.1.1 and 5.1.2. In particular, recalling that $\phi(X_{[\psi_{i-1},\psi_i]})$ is bounded on compact sets (see Result 4) and furthermore $\lambda(X_{[\psi_{i-1},\psi_i]}) \in [0,\Lambda]$, then in a similar manner to Section 5.1 we can partition the path space of $\mathbb{W}_{\psi_{i-1},\psi_i}\,|\,(X_{\psi_{i-1}},X_{\psi_i})$ into disjoint layers and simulate the layer to which our proposal sample path belongs (see Principle 1, denoting $R_i := R_{X[\psi_{i-1},\psi_i]} \sim \mathcal{R}$ as the simulated layer, the precise details of which are given in Section 6.2 and Section 6.3). Once

again, an upper and lower bound for $(\phi(X_{[\psi_{i-1},\psi_i]}) + \lambda(X_{[\psi_{i-1},\psi_i]}))$ can always be found conditional on R_i ($U_{X[\psi_{i-1},\psi_i]} \in \mathbb{R}$ and $L_{X[\psi_{i-1},\psi_i]} \in \mathbb{R}$ respectively) and so we have,

$$\begin{aligned}\tilde{P}_{\mathbb{W}_{\psi_{i-1},\psi_i}} &:= \mathbb{E}_{\mathbb{W}_{\psi_{i-1},\psi_i}}\left[\tilde{P}_{\mathbb{W}_{\psi_{i-1},\psi_i}}(X)\right] = \mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}_{\psi_{i-1},\psi_i}|R_i}\left[\tilde{P}_{\mathbb{W}_{\psi_{i-1},\psi_i}}(X)\right] \\ &:= \mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}_{\psi_{i-1},\psi_i}|R_i}\left[\exp\left\{-\int_{\psi_{i-1}}^{\psi_i}\phi(X_s)+\lambda(X_s)\,\mathrm{d}s\right\}\cdot e^{\Phi(\psi_i-\psi_{i-1})}\right]. \end{aligned} \tag{5.43}$$

Letting $\mathbb{K}_{R(i)}$ be the law of $\kappa(i) \sim \mathrm{Poi}\left((U_{X[\psi_{i-1},\psi_i]} - L_{X[\psi_{i-1},\psi_i]})\cdot(\psi_i - \psi_{i-1})\right)$ and $\mathbb{U}_{\kappa(i)}$ the distribution of $(\xi_{i,1},\ldots,\xi_{i,\kappa(i)}) \overset{\text{iid}}{\sim} \mathrm{U}[\psi_{i-1},\psi_i]$ we can find an unbiased estimate of our acceptance probability which only requires a finite dimensional realisation of the proposal sample path (in a similar fashion to the approach taken in Section 5.1.1),

$$\begin{aligned}\tilde{P}_{\mathbb{W}_{\psi_{i-1},\psi_i}}(X) &:= \exp\left\{-\int_{\psi_{i-1}}^{\psi_i}\phi(X_s)+\lambda(X_s)\,\mathrm{d}s\right\}\cdot e^{\Phi(\psi_i-\psi_{i-1})} \\ &= e^{-(L_{X[\psi_{i-1},\psi_i]}-\Phi)\cdot(\psi_i-\psi_{i-1})}\mathbb{E}_{\mathbb{K}_{R_i}}\mathbb{E}_{\mathbb{U}_{\kappa(i)}}\left[\prod_{j=1}^{\kappa(i)}\left(\frac{U_{X[\psi_{i-1},\psi_i]}-\phi(X_{\xi_j})-\lambda(X_{\xi_j})}{U_{X[\psi_{i-1},\psi_i]}-L_{X[\psi_{i-1},\psi_i]}}\right)\middle| X\right]. \end{aligned} \tag{5.44}$$

Simulating a finite dimensional proposal sample path as suggested by (5.41 – 5.44) and incorporating the ideas of *either* the Unbounded Exact Algorithm (UEA, Algorithm 5.1.3) in Section 5.1.1 *or* Adaptive Unbounded Exact Algorithm (AUEA, Algorithm 5.1.4) in Section 5.1.2, results directly in the Conditional Unbounded Jump Exact Algorithm (CUJEA) presented in Algorithm 5.4.1 or Conditional Adaptive Unbounded Jump Exact Algorithm (CAUJEA) presented in Algorithm 5.4.1.

$$\begin{aligned}\mathcal{S}_{\text{CUJEA}}(X) &:= \cup_{i=1}^{N_T+1}\left\{\left(\xi_{i,j}, X_{\xi_{i,j}}\right)_{j=0}^{\kappa(i)+1}, R_{X[\psi_{i-1},\psi_i]}\right\}. \\ \mathcal{S}_{\text{CAUJEA}}(X) &:= \cup_{i=1}^{N_T+1}\left\{\left(\xi_{i,j}, X_{\xi_{i,j}}\right)_{j=0}^{\kappa(i)+1}, \left(R^{[\xi_{i,j-1},\xi_{i,j}]}_{X[\psi_{i-1},\psi_i]}\right)_{j=1}^{\kappa(i)+1}\right\}. \end{aligned} \tag{5.45}$$

In Figures 5.4.1 and 5.4.2 we present illustrative examples of accepted sample path skeletons simulated by means of the CUJEA and CAUJEA respectively.

First considering the CUJEA, then note that we present a number of alternate methods for simulating unbiasedly layer information (Algorithm 5.4.1 Step 5a), layered Brownian bridges (Step 3 and Step 5(c)ii), and the sample path at further times after acceptance (Step 7), in Section 6.2. Unlike existing exact algorithms for simulating jump diffusion bridges [Gonçalves and Roberts, 2013], the CUJEA conducts early rejection to avoid any further unnecessary simulation of the rejected sample path (in a similar manner to the

UEA (Algorithm 5.1.3), the CUJEA is a nested rejection sampler). The early rejection steps in Algorithm 5.4.1 are Steps 2, 4, 5b and 5d.

Now considering the CAUJEA which is based on the AUEA (Algorithm 5.1.4) presented in Section 5.1.2, then recall that we outline how to simulate (unbiasedly) layer information (Algorithm 5.4.2 Step 2a), intermediate skeletal points (Step 2(c)iiB) and new layer information (Step 2(c)iiD) in a variety of ways in Section 6.3. After the skeleton has been accepted the sample path can be simulated at any other desired finite collection of time points and the layers refined as necessary (by application of Algorithm 5.1.4 Step 4(b)ii and Step 4(b)iv and as detailed in Section 6.3).

In analogous fashion to the difference between the UEA and AUEA, the CAUJEA is an iterative scheme which outputs a skeleton which satisfies Principles 1, 2 and 3 directly (i.e. it outputs a skeleton comprising skeletal points and layer information for the intervals between consecutive skeletal points), whereas the CUJEA only satisfies Principles 1 and 2 directly and requires the skeleton to be augmented in order to simulate the sample path at any additional point. Whether in practise the CUJEA or CAUJEA should be implemented is dependent on the particular problem and whether or not layer information is of use.

Algorithm 5.4.1 Conditioned Unbounded Jump Exact Algorithm (CUJEA).

1. Simulate compound Poisson process $J_{[0,T]} \sim \mathcal{J}$,
   (a) Simulate number of jumps $N_T \sim \text{Poi}(\Lambda \cdot T)$.
   (b) Simulate jump times $\psi_1, \ldots, \psi_{N_T} \overset{\text{iid}}{\sim} \text{U}[0, T]$.
   (c) Simulate jump sizes – For i in 1 to N_T, $\delta_{\psi_i} \sim f_\delta$.
2. With probability $\left(1 - \exp\left\{-\left(y - J_T - x\right)^2 / 2T\right\}\right)$ reject path and return to Step 1.
3. Simulate $X'_{\psi_1}, \ldots, X'_{\psi_{N_T}} \sim \tilde{\mathbb{W}}_{0,T}^{x,y-J_T}$, setting $X_{\psi_1} := X'_{\psi_1} + J_{\psi_1}, \ldots, X_{\psi_{N_T}} := X'_{\psi_{N_T}} + J_{\psi_{N_T}}$.
4. With probability $\left(1 - \prod_{i=1}^{N_T} \frac{\lambda(X_{\psi_i-}) \cdot f_\nu\left(X_{\psi_i}; X_{\psi_i-}\right) \cdot \exp\left\{-\left[A(X_{\psi_i}) - A(X_{\psi_i-})\right]\right\}}{\Lambda \cdot f_\delta\left(X_{\psi_i} - X_{\psi_i-}\right) \cdot \varkappa}\right)$ reject path and return to Step 1.
5. For i in 1 to $(N_T + 1)$,
   (a) Simulate layer information $R_i := R_{X[\psi_{i-1},\psi_i]} \sim \mathcal{R}$.
   (b) With probability $\left(1 - \exp\left\{-(L_{X[\psi_{i-1},\psi_i]} - \Phi) \cdot (\psi_i - \psi_{i-1})\right\}\right)$ reject path and return to Step 1.
   (c) Simulate skeleton points $\left(X_{\xi_{i,1}}, \ldots, X_{\xi_{i,\kappa_i}}\right) \Big| R_i$,
      i. Simulate $\kappa_i \sim \text{Poi}\left(\left(U_{X[\psi_{i-1},\psi_i]} - L_{X[\psi_{i-1},\psi_i]}\right) \cdot (\psi_i - \psi_{i-1})\right)$ and skeleton times $\xi_{i,1}, \ldots, \xi_{i,\kappa_i} \overset{\text{iid}}{\sim} \text{U}\left[\psi_{i-1}, \psi_i\right]$.
      ii. Simulate sample path at skeleton times $X_{\xi_{i,1}}, \ldots, X_{\xi_{i,\kappa_i}} \sim \mathbb{W}_{\psi_{i-1},\psi_i} \Big| \left(R_i, X_{\psi_{i-1}}, X_{\psi_i}\right)$.
   (d) With probability $\left(1 - \prod_{j=1}^{\kappa_i} \frac{\left(U_{X[\psi_{i-1},\psi_i]} - \phi(X_{\xi_{i,j}}) - \lambda(X_{\xi_{i,j}})\right)}{\left(U_{X[\psi_{i-1},\psi_i]} - L_{X[\psi_{i-1},\psi_i]}\right)}\right)$, reject path and return to Step 1.
6. Accept sample path skeleton.

---

7. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{N_T+1} \left(\otimes_{j=1}^{\kappa_i+1} \mathbb{W}_{\xi_{i,j-1},\xi_{i,j}}^{X_{\xi_{i,j-1}}, X_{\xi_{i,j}}}\right) \Big| R_i\right)$.

Algorithm 5.4.2 Conditioned Adaptive Unbounded Jump Exact Algorithm (CAUJEA).

1. Simulate compound Poisson process ($J_{[0,T]} \sim \mathcal{J}$) and the skeleton at jump times $(X_{\psi_1}, \ldots, X_{\psi_{N_T}})$ as per Algorithm 5.4.1 Steps 1 – 4.

2. For i in 1 to $(N_T + 1)$,

   (a) Simulate initial layer information $R_i := R_{X[\psi_{i-1},\psi_i]} \sim \mathcal{R}$, setting $\Pi := \{\Xi\} := \left\{\left\{[\psi_{i-1}, \psi_i], X_{\psi_{i-1}}, X_{\psi_i}, R_i\right\}\right\}$ and $\kappa_i = 0$.

   (b) With probability $\left(1 - \exp\left\{-(L_{X[\psi_{i-1},\psi_i]} - \Phi) \cdot (\psi_i - \psi_{i-1})\right\}\right)$ reject path and return to Step 1.

   (c) While $\left|\Pi\right| \neq 0$,

      i. Set $\Xi = \Pi_1$.

      ii. Simulate $\tau \sim \text{Exp}\left(2\Delta_X^{\Xi}\right)$. If $\tau > d_\Xi$ then set $\Pi := \Pi \setminus \Xi$ else,

         A. Set $\kappa_i = \kappa_i + 1$ and with probability $1/2$ set $\xi'_{\kappa_i} = m_\Xi - \tau$ else $\xi'_{\kappa_i} = m_\Xi + \tau$.

         B. Simulate $X_{\xi'_{\kappa_i}} \sim \mathbb{W}^{x(\Xi),y(\Xi)}_{s-(\Xi),t+(\Xi)} \left| R_i^{\Xi}\right.$.

         C. With probability $\left(1 - \left[U^{\Xi}_{X[\psi_{i-1},\psi_i]} - \phi\left(X_{\xi'_{\kappa_i}}\right) - \lambda\left(X_{\xi'_{\kappa_i}}\right)\right] / \Delta^{\Xi}_{X[\psi_{i-1},\psi_i]}\right)$ reject path and return to Step 1.

         D. Simulate new layer information $R_i^{\left[s-(\Xi),\xi'_{\kappa_i}\right]}$ and $R_i^{\left[\xi'_{\kappa_i},t+(\Xi)\right]}$ conditional on R_i^{Ξ}.

         E. With probability $\left(1 - \exp\left\{-\left[L^{\left[s-(\Xi),\xi'_{\kappa_i}\right]}_{X[\psi_{i-1},\psi_i]} + L^{\left[\xi'_{\kappa_i},t+(\Xi)\right]}_{X[\psi_{i-1},\psi_i]} - 2L^{\Xi}_{X[\psi_{i-1},\psi_i]}\right][d_\Xi - \tau]\right\}\right)$ reject path and return to Step 1.

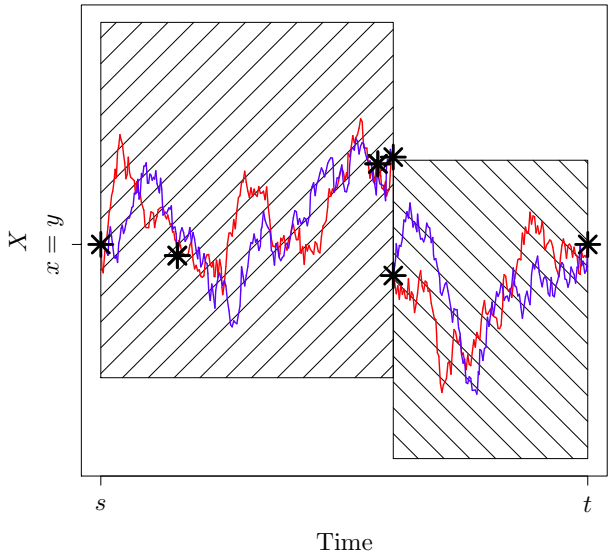
         F. Set $\Pi := \Pi \bigcup \left\{[s_\Xi, m_\Xi - \tau], X^{\Xi}_{s-}, X_{\xi'_{\kappa_i}}, R_i^{\left[s-(\Xi),\xi'_{\kappa_i}\right]}\right\} \bigcup \{[m_\Xi + \tau, t_\Xi], X_{\xi'_{\kappa_i}}, X^{\Xi}_{t+}, R_i^{\left[\xi'_{\kappa_i},t+(\Xi)\right]}\} \setminus \Xi$.

   (d) Define skeletal points $\xi_{i,1}, \ldots, \xi_{i,\kappa_i}$ as the order statistics of the set $\left\{\xi'_{i,1}, \ldots, \xi'_{i,\kappa_i}\right\}$.

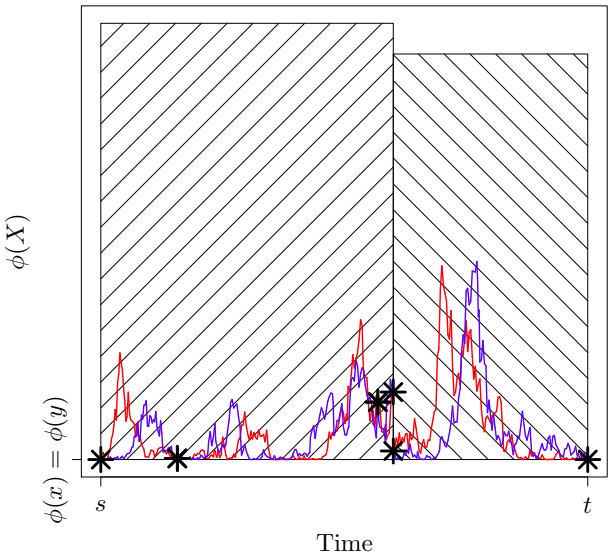
3. Accept sample path skeleton.

---

4. $*$ *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{N_T+1}\left(\otimes_{j=1}^{\kappa_i+1} \mathbb{W}^{X_{\xi_{i,j-1}}, X_{\xi_{i,j}}}_{\xi_{i,j-1},\xi_{i,j}} \left| R_i^{[\xi_{i,j-1},\xi_{i,j}]}\right.\right)\right)$.

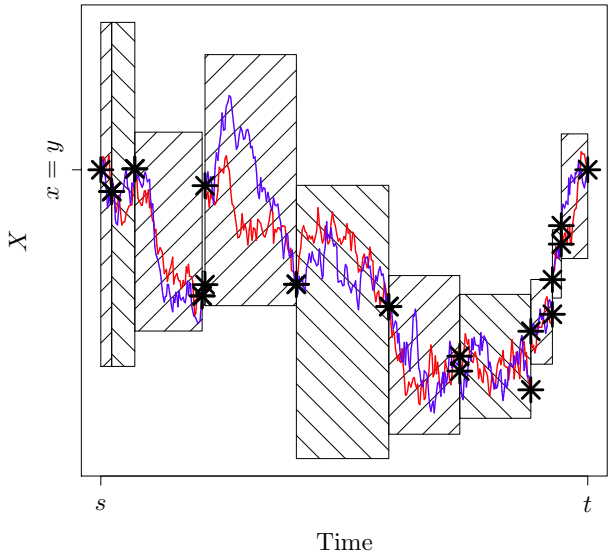


(a) Example sample path skeleton $\mathcal{S}_{\text{CUJEA}}(X)$, overlaid with example sample path trajectories.

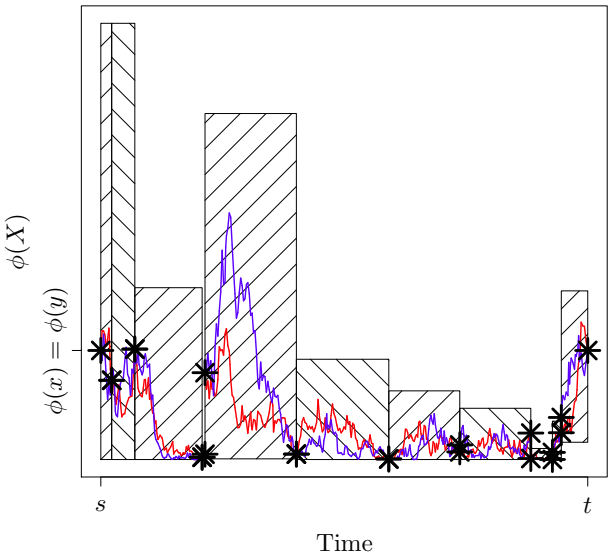


(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{CUJEA}}(X)$, and example sample path trajectories.

Figure 5.4.1: Illustrative sample path skeleton output from the Conditioned Unbounded Jump Exact Algorithm (CUJEA; Algorithm 5.4.1), $\mathcal{S}_{\text{CUJEA}}(X) := \bigcup_{i=1}^{N_T+1} \left\{ \left(\xi_{i,j}, X_{\xi_{i,j}} \right)_{j=0}^{\kappa(i)+1}, R_{X[\psi_{i-1},\psi_i]} \right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{N_T+1} \left(\otimes_{j=1}^{\kappa_i+1} \mathbb{W}_{\xi_{i,j-1},\xi_{i,j}}^{X_{\xi_{i,j-1}},X_{\xi_{i,j}}} \right) \middle| R_i \right)$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.



(a) Example sample path skeleton $\mathcal{S}_{\text{CAUJEA}}(X)$, overlaid with example sample path trajectories.



(b) ϕ mapping of example sample path skeleton $\mathcal{S}_{\text{CAUJEA}}(X)$, and example sample path trajectories.

Figure 5.4.2: Illustrative sample path skeleton output from the Conditioned Adaptive Unbounded Jump Exact Algorithm (CAUJEA; Algorithm 5.4.2), $\mathcal{S}_{\text{CAUJEA}}(X) := \bigcup_{i=1}^{N_T+1} \left\{ \left(\xi_{i,j}, X_{\xi_{i,j}}\right)_{j=0}^{\kappa(i)+1}, \left(R_{X[\psi_{i-1},\psi_i]}^{[\xi_{i,j-1},\xi_{i,j}]}\right)_{j=1}^{\kappa(i)+1} \right\}$, overlaid with example sample path trajectories $X^{\text{rem}} \sim \left(\otimes_{i=1}^{N_T+1} \left(\otimes_{j=1}^{\kappa_i+1} \mathbb{W}_{\xi_{i,j-1},\xi_{i,j}}^{X_{\xi_{i,j-1}},X_{\xi_{i,j}}} \middle| R_i^{[\xi_{i,j-1},\xi_{i,j}]} \right) \right)$. Hatched regions indicate layer information, whereas the asterisks indicate skeletal points.

6 Brownian Bridge Path Space Constructions and Simulation

"Very often the laws derived by physicists from a large number of observations are not rigorous, but approximate."

— Augustin Louis Cauchy

In Chapter 5 we introduced novel methodology for simulating skeletons of diffusion and jump diffusion sample paths (the *exact algorithms*). Each of the algorithms presented in Chapter 5 is dependent on being able to simulate *Layered Brownian Bridge* sample path skeletons. In particular, we require methodology for simulating (without discretisation error) skeletons of Brownian bridge sample paths with various restrictions, which will include some interval(s) within which they are almost surely constrained. Details on how to simulate layered Brownian bridge sample path skeletons were omitted from the exact algorithms presented in Chapter 5 in order to avoid entangling the key ideas.

In this chapter we present new results for simulating quantities related to various *Brownian Bridge Path Space* constructions, which together allow the simulation of layered Brownian bridge sample path skeletons, and hence the implementation of the exact algorithms in Chapter 5.

In Section 6.1 we detail how to simulate events of probability corresponding to the probability that Brownian and Bessel bridge sample paths, with various restrictions, are constrained within particular interval(s). In Section 6.2 we present methodology for simulating the layered Brownian bridge sample path skeletons required within the Unbounded and Conditioned Unbounded Exact Algorithms (Algorithms 5.1.3 and 5.2.1). Finally in Section 6.3 we present methodology for simulating the *Adaptive Layered Brownian*

Bridge sample path skeletons required within the Adaptive and Conditioned Adaptive Unbounded Exact Algorithms (Algorithms 5.1.4 and 5.2.2), along with the Adaptive and Conditioned Adaptive Unbounded Jump Exact Algorithms (Algorithms 5.3.3 and 5.4.2). Both the layered and adaptive layered Brownian bridge constructions (Sections 6.2 and 6.3 respectively) rely upon the results detailed in Section 6.1, in which we demonstrate how to simulate unbiasedly unknown probabilities which can be represented as the limit of alternating Cauchy sequences (as in Section 2.7 and by employing retrospective Bernoulli sampling).

6.1 Simulating Brownian Bridge Path Space Probabilities

Simulating events corresponding to the probability that Brownian and Bessel bridge sample paths are constrained within particular intervals (or alternatively the complement of the probability they exceed particular boundaries) is typically non-trivial. In particular, as discussed in detail in Pötzelberger and Wang [1997], it is typically only possible to represent such probabilities as the limit of alternating Cauchy sequences (of the form found in (2.15) of Section 2.7 and excluding a small number of special cases such as one-sided piecewise linear boundaries). Consequently, although these unknown probabilities can be approximated to arbitrary precision, simulating an event of these probabilities by truncating the alternating Cauchy sequences results in the introduction of bias. However, as first proposed by Beskos et al. [2008], the alternating Cauchy sequence representations can be exploited to simulate unbiasedly events of these unknown probabilities by application of retrospective Bernoulli sampling (see Section 2.7).

In this section we introduce both existing and novel methodology for simulating events corresponding to various Brownian path space probabilities (based on the retrospective Bernoulli sampling approach taken by Beskos et al. [2008]), which we require in order to construct the layered Brownian bridges we introduce in Sections 6.2 and 6.3.

We conclude the introductory comments in this section by providing an illustrative example of why the probabilities we are interested in simulating can be represented as the limit of alternating Cauchy sequences, and how the retrospective Bernoulli sampling approach can be employed. This will provide some insight to the key steps required in the more advanced Brownian path space constructions which we consider later.

Now, suppose we are interested in simulating an event of probability corresponding to the probability that a Brownian motion sample path W (see Section 2.8), starting at $W_0 = 0$, is constrained to the interval $[-1, 1]$ at all times $t \in [0, 1]$. Noting that $\mathbb{P}(W_t < -1) = \mathbb{P}(W_t > 1)$ by symmetry (see Self-Similarity Property 2) our desired probability can be represented as follows (where we denote by $\sup|W_{[s,t]}| := \sup_{u\in[s,t]}|W_u|$),

$$\begin{aligned}\mathbb{P}\left(\sup|W_{[0,1]}| \leq 1\right) &= 1 - \mathbb{P}\left(\sup|W_{(0,1]}| > 1\right) \\ &= 1 - \left[2 \cdot \mathbb{P}\left(\sup W_{[0,1]} > 1\right) - \mathbb{P}\left(\{\inf W_{[0,1]} < -1\} \bigcap \{\sup W_{[0,1]} > 1\}\right)\right]. \quad (6.1)\end{aligned}$$

Now, by reflection of Brownian motion (see for instance the argument in Section 2.8.1 recalling from Section 2.8 that Brownian motion has the Markov property, or alternatively [Rogers and Williams, 2000, Chap. I.13]) and as illustrated in Figure 6.1.1, we can equate the probability that a sample path exits the interval $[-1, 1]$ from both above (1) and below (-1) to the probability that a sample path exceeds a one-sided boundary of distance at least that to one boundary and then the difference between the boundaries (in effect three times the distance). In particular we have,

$$\mathbb{P}\left(\{\inf W_{[0,1]} < -1\} \bigcap \{\sup W_{[0,1]} > 1\}\right) = \mathbb{P}\left(\sup|W_{[0,1]}| > 3\right), \quad (6.2)$$

and so by applying this argument recursively and noting that $\mathbb{P}(\sup W_{[s,t]} > \upsilon \,|\, W_s = x) = \Phi([\upsilon - x]/\sqrt{t-s}) - \Phi([x - \upsilon]/\sqrt{t-s})$ (see for instance Pötzelberger and Wang [1997] where Φ denotes the CDF of the standard Normal distribution and $\upsilon \geq x$) we have,

$$\begin{aligned}\mathbb{P}\left(\sup|W_{[0,1]}| \leq 1\right) &= 1 - 2\left[\mathbb{P}\left(\sup W_{[0,1]} > 1\right) - \mathbb{P}\left(\sup W_{[0,1]} > 3\right) + \mathbb{P}\left(\sup W_{[0,1]} > 9\right) - \ldots\right] \\ &= 1 - 2\sum_{i=0}^{\infty}(-1)^i \mathbb{P}\left(\sup W_{[0,1]} > 3^i\right) = 1 - 2\sum_{i=0}^{\infty}(-1)^i\left[\Phi(3^i) - \Phi(-3^i)\right]. \quad (6.3)\end{aligned}$$

Now, noting that our desired probability can be represented as the limit of an infinite sequence,

$$\mathbb{P}\left(\sup|W_{[0,1]}| \leq 1\right) = \lim_{k\to\infty} S_k^W, \quad \text{where } S_k^W := 1 - 2\sum_{i=0}^{k-1}(-1)^i\left[\Phi(3^i) - \Phi(-3^i)\right], \quad (6.4)$$

and further noting that each subsequent term in the sequence S_k^W has alternating sign and is decreasing in size then, it can be easily shown that, our desired probability can be represented as the limit of an alternating Cauchy sequence of the form found in (2.15) of

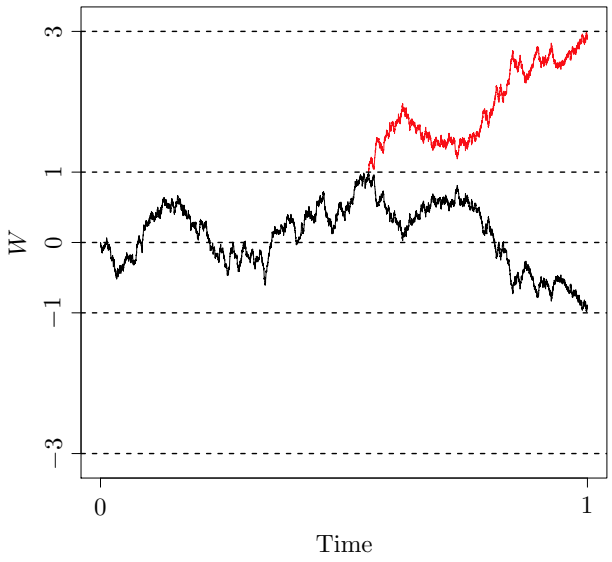


Figure 6.1.1: An illustration of the reflection of a Brownian motion sample path around a boundary.

Section 2.7 (albeit one in which the even terms of the sequence converge from above and the odd terms of the sequence converge from below). As such, events of probability corresponding to our desired probability can be simulated unbiasedly by direct application of retrospective Bernoulli sampling (see Section 2.7) and as detailed in Algorithm 6.1.1.

In the remainder of this section we consider an analogous partitioning for Brownian and Bessel bridges for more generalised settings with non-symmetric upper and lower boundaries. The complication arising from non-symmetric boundaries being that when considering the probability of reaching both boundaries the probability the upper boundary is reached first will not be equal to the probability the lower boundary is reached first. As such the symmetry argument of the above example no longer holds and more sophisticated arguments are required (which are not within the scope of this thesis but can be found in a number of papers, for instance Anderson [1960])).

Algorithm 6.1.1 Simulating an event corresponding to the probability that $W_{[0,1]} \in [-1, 1]$ where $W \sim \mathbb{W}^0_{0,1}$.

1. Simulate $u \sim \mathrm{U}[0, 1]$ and set $k = 1$.
2. While $u \in \left(S^W_{2k+1}, S^W_{2k}\right)$ (where $S^W_k := 1 - 2\sum_{i=0}^{k-1}(-1)^i[\Phi(3^i) - \Phi(-3^i)]$), $k = k + 1$.
3. If $u \leq S^W_{2k+1}$ then $u < p$ so return 1 else $u > p$ so return 0.

6.1.1 Simulating Elementary Brownian Path Space Probabilities

In this section we outline existing and novel methodology for simulating events of probability corresponding to the probability that Brownian and Bessel bridge sample paths are constrained within particular intervals.

We begin by presenting known results pertaining to the probability that a Brownian bridge sample path is constrained within a particular interval (see Anderson [1960] and Pötzelberger and Wang [2001]) and how to simulate events of this probability (as first detailed in Beskos et al. [2008]). We reproduce this result (with a more detailed proof) as it is used and extended extensively throughout the remainder of this chapter. In particular, we are interested in the probability that $\{W_u : s \leq u \leq t\} \in [\ell, \upsilon]$ for some sample path $W \sim \mathbb{W}^{x,y}_{s,t}$ (where in the remainder of this thesis, with a slight abuse of notation, we instead write $\{W \in [\ell, \upsilon]\}$ to mean $\{W_u : s \leq u \leq t\} \subseteq [\ell, \upsilon]$).

In Figure 6.1.2 we show example sample path trajectories of a Brownian bridge ($W \sim \mathbb{W}^{x,y}_{s,t}$), which remain in the interval $[\ell, \upsilon]$.

Theorem 6.1.1 ([Pötzelberger and Wang, 2001, 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. $W \in [\ell, \upsilon]$) can be represented by the following infinite series,*

$$\gamma^{\ell,\upsilon}_{s,t}(x, y) := \mathbb{P}\left(W \in [\ell, \upsilon]\right) = 1 - \sum_{j=1}^{\infty}\left\{\varsigma^{\ell,\upsilon}_{s,t}(j; x, y) - \varphi^{\ell,\upsilon}_{s,t}(j; x, y)\right\} =: 1 - \zeta^{\ell,\upsilon}_{s,t}(x, y), \quad (6.5)$$

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

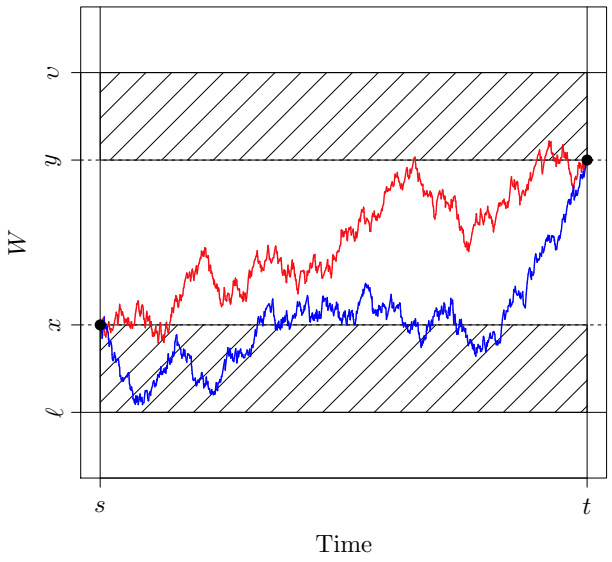


Figure 6.1.2: Example sample path trajectories $W \sim \mathbb{W}_{s,t}^{x,y} \left| (W \in [\ell, \upsilon]) \right.$.

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

$$\bar{\varsigma}_{s,t}^{\ell,\upsilon}(j; x, y) := \exp\left\{ -\frac{2}{t-s} \left(\left| \upsilon - \ell \right| j + (\ell \wedge \upsilon) - x \right) \cdot \left(\left| \upsilon - \ell \right| j + (\ell \wedge \upsilon) - y \right) \right\}, \tag{6.6}$$

$$\bar{\varphi}_{s,t}^{\ell,\upsilon}(j; x, y) := \exp\left\{ -\frac{2j}{t-s} \left(\left| \upsilon - \ell \right|^2 j + \left| \upsilon - \ell \right| (x - y) \right) \right\}. \tag{6.7}$$

Corollary 6.1.1 ([Beskos et al., 2008, Prop. 2])**.** $\gamma_{s,t}^{\ell,\upsilon}(x, y)$ *can be represented as the limit as* $k \to \infty$ *of the following alternating Cauchy sequence,*

$$S_{2k}^{\gamma} := 1 - \sum_{j=1}^{k} \left\{ \varsigma_{s,t}^{\ell,\upsilon}(j; x, y) - \varphi_{s,t}^{\ell,\upsilon}(j; x, y) \right\}, \qquad S_{2k+1}^{\gamma} := S_{2k}^{\gamma} - \varsigma_{s,t}^{\ell,\upsilon}(k+1; x, y). \tag{6.8}$$

Proof. Considering (6.6) and (6.7), clearly we have $\forall j \; \varsigma_{s,t}^{\ell,\upsilon}(j; x, y) > 0$ and $\varphi_{s,t}^{\ell,\upsilon}(j; x, y) > 0$ as a consequence of the exponential form of each of these terms. As such, to show that (6.5) can be represented as the limit of an alternating Cauchy sequence it is sufficient to show that $\forall j$ we have $\varsigma_{s,t}^{\ell,\upsilon}(j; x, y) \geq \varphi_{s,t}^{\ell,\upsilon}(j; x, y) \geq \varsigma_{s,t}^{\ell,\upsilon}(j+1; x, y) \geq \varphi_{s,t}^{\ell,\upsilon}(j+1; x, y) \geq \ldots$, which can be proved inductively by first showing that $\forall j \; \varsigma_{s,t}^{\ell,\upsilon}(j; x, y) / \varphi_{s,t}^{\ell,\upsilon}(j; x, y) \geq 1$ and

then $\forall j\, \varphi^{\ell,\upsilon}_{s,t}(j;x,y)/\varsigma^{\ell,\upsilon}_{s,t}(j+1;x,y) \geq 1$.

First considering $\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y)/\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)$, we have (denoting by $Z := 2|\upsilon-\ell|\, j+(\ell\wedge\upsilon)-y$),

$$\begin{aligned}\frac{\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y)}{\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)} &= \frac{\exp\left\{-\frac{2}{t-s}\left(\left|\upsilon-\ell\right| j+(\ell\wedge\upsilon)-x\right)\cdot\left(\left|\upsilon-\ell\right| j+(\ell\wedge\upsilon)-y\right)\right\}}{\exp\left\{-\frac{2j}{t-s}\left(\left|\upsilon-\ell\right|^2 j+\left|\upsilon-\ell\right|(x-y)\right)\right\}}\\ &= \exp\left\{-\frac{2}{t-s}\left(2\left|\upsilon-\ell\right| j(\ell\wedge\upsilon)+[(\ell\wedge\upsilon)]^2-(\ell\wedge\upsilon)y-2\left|\upsilon-\ell\right| jx-(\ell\wedge\upsilon)x+xy\right)\right\}\\ &= \exp\left\{-\frac{2}{t-s}\left(Z(\ell\wedge\upsilon)-Zx\right)\right\} = \exp\left\{-\frac{2Z}{t-s}\left[(\ell\wedge\upsilon)-x\right]\right\}. \end{aligned} \tag{6.9}$$

Noting that $y-(\ell\wedge\upsilon)\leq(\ell\vee\upsilon)-(\ell\wedge\upsilon) =: |\upsilon-\ell|$ we have that $Z\geq 0$ and furthermore $(\ell\wedge\upsilon)-x\leq 0$. As such $\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y)/\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)\geq 1$ and so by direct consequence $\forall j$ $\varsigma^{\ell,\upsilon}_{s,t}(j;x,y)/\varphi^{\ell,\upsilon}_{s,t}(j;x,y)\geq 1$.

Now, considering $\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)/\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j+1;x,y)$ we similarly have,

$$\begin{aligned}\frac{\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)}{\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j+1;x,y)} &= \frac{\exp\left\{-\frac{2j}{t-s}\left(\left|\upsilon-\ell\right|^2 j+\left|\upsilon-\ell\right|(x-y)\right)\right\}}{\exp\left\{-\frac{2}{t-s}\left(\left|\upsilon-\ell\right|(j+1)+(\ell\wedge\upsilon)-x\right)\cdot\left(\left|\upsilon-\ell\right|(j+1)+(\ell\wedge\upsilon)-y\right)\right\}}\\ &= \exp\left\{\frac{2}{t-s}\Big(\left|\upsilon-\ell\right|^2(2j+1)+2\left|\upsilon-\ell\right|(j+1)(\ell\wedge\upsilon)-\left|\upsilon-\ell\right| y\right.\\ &\qquad\left.+(\ell\wedge\upsilon)\left|\upsilon-\ell\right|(j+1)+[(\ell\wedge\upsilon)]^2-(\ell\wedge\upsilon)y-\left|\upsilon-\ell\right|(2j+1)x-(\ell\wedge\upsilon)x+xy\Big)\right\}\\ &= \exp\left\{\frac{2}{t-s}\Big(\left|\upsilon-\ell\right|\left[\left|\upsilon-\ell\right|(2j+1)+2(j+1)(\ell\wedge\upsilon)-y\right]\right.\\ &\qquad\left.+(\ell\wedge\upsilon)\left[\left|\upsilon-\ell\right|(j+1)+(\ell\wedge\upsilon)-y\right]-x\left[\left|\upsilon-\ell\right|(2j+1)+(\ell\wedge\upsilon)-y\right]\Big)\right\}. \end{aligned} \tag{6.10}$$

Again, noting that $\max\{x-(\ell\wedge\upsilon), y-(\ell\wedge\upsilon)\}\leq(\ell\vee\upsilon)-(\ell\wedge\upsilon) =: |\upsilon-\ell|$ we have that $\bar{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)/\bar{\varsigma}^{\ell,\upsilon}_{s,t}(j+1;x,y)\geq 1$ and so by direct consequence we have $\varphi^{\ell,\upsilon}_{s,t}(j;x,y)/\varsigma^{\ell,\upsilon}_{s,t}(j+1;x,y)\geq 1$. As a result we have that $\forall j\ \varsigma^{\ell,\upsilon}_{s,t}(j;x,y)\geq\varphi^{\ell,\upsilon}_{s,t}(j;x,y)\geq\varsigma^{\ell,\upsilon}_{s,t}(j+1;x,y)\geq\varphi^{\ell,\upsilon}_{s,t}(j+1;x,y)\geq\ldots$ and so $\gamma^{\ell,\upsilon}_{s,t}(x,y)$ can be represented as the limit of the alternating Cauchy sequence in (6.8) as desired. □

As a consequence of Corollary 6.1.1, events of probability $\gamma_{s,t}^{\ell,\upsilon}(x,y)$ can be simulated unbiasedly by retrospective Bernoulli sampling (see Section 2.7 and Algorithm 2.7.1) as detailed in Algorithm 6.1.2.

Algorithm 6.1.2 Simulating an event of probability $\gamma_{s,t}^{\ell,\upsilon}(x,y)$ [Beskos et al., 2008].

---

1. Simulate $u \sim \mathrm{U}[0,1]$ and set $k=1$.
2. While $u \in \left(S_{2k+1}^{\gamma}, S_{2k}^{\gamma}\right)$ (where $S_{2k}^{\gamma} := 1 - \sum_{j=1}^{k}\left\{\varsigma_{s,t}^{\ell,\upsilon}(j;x,y) - \varphi_{s,t}^{\ell,\upsilon}(j;x,y)\right\}$ and $S_{2k+1}^{\gamma} := S_{2k}^{\gamma} - \varsigma_{s,t}^{\ell,\upsilon}(k+1;x,y)$) then $k = k+1$.
3. If $u \leq S_{2k+1}^{\gamma}$ then $u < p$ so return 1 else $u > p$ so return 0.

Now, it is of interest to note that it is similarly possible to find a representation for the probability that a Brownian bridge sample path remains within an interval between a piecewise linear upper and lower boundary (as shown in Pötzelberger and Wang [2001]). As we will encounter later in this chapter, there is some merit in considering what upper and lower boundary pair would for any given probability constrain the smallest volume of path space. As we are restricted to linear boundaries such a pair would clearly have gradient $(y-x)/(t-s)$. In Figure 6.1.3 we show example sample path trajectories of a Brownian bridge constrained within these non-zero gradient boundaries. This notion results in the following modification of (6.5),

$$\begin{aligned}\tilde{\gamma}_{s,t}^{\ell,\upsilon}(x,y) &:= \mathbb{P}\left(W_u \in \left[x+\frac{(y-x)(u-s)}{t-s}+\ell, x+\frac{(y-x)(u-s)}{t-s}+\upsilon\right] \forall u \in [s,t] \,\middle|\, W_s=x, W_t=y\right)\\ &= 1 - \sum_{j=1}^{\infty}\left\{\tilde{\varsigma}_{s,t}^{\ell,\upsilon}(j;x,y) - \tilde{\varphi}_{s,t}^{\ell,\upsilon}(j;x,y)\right\} =: 1 - \tilde{\zeta}_{s,t}^{\ell,\upsilon}(x,y), \end{aligned} \tag{6.11}$$

where,

$$\tilde{\varsigma}_{s,t}^{\ell,\upsilon}(j;x,y) = \exp\left\{-\frac{2}{t-s}\left[j(\upsilon-\ell)+\ell\right]^2\right\} + \exp\left\{-\frac{2}{t-s}\left[j(\upsilon-\ell)-\upsilon\right]^2\right\}, \tag{6.12}$$

$$\tilde{\varphi}_{s,t}^{\ell,\upsilon}(j;x,y) = 2\exp\left\{-\frac{2}{t-s}\left[j(\upsilon-\ell)\right]^2\right\}. \tag{6.13}$$

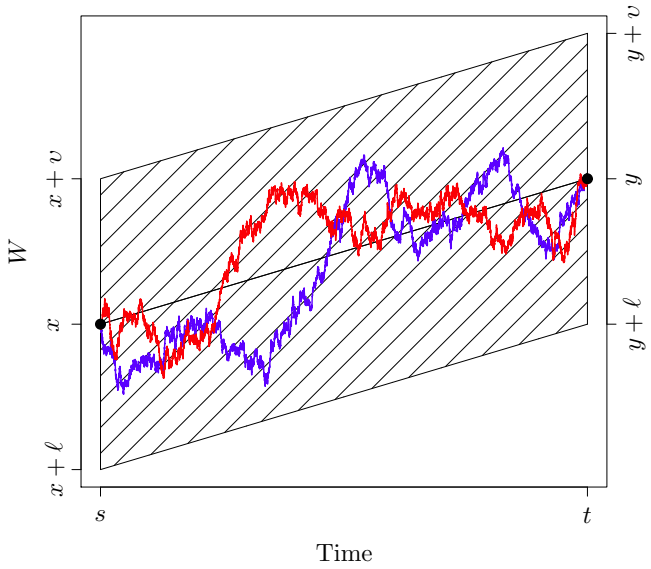


Figure 6.1.3: Example sample path trajectories $W \sim \mathbb{W}_{s,t}^{x,y}\big|(W_u \in [x+(y-x)\cdot(u-s)/(t-s)+\ell, x+(y-x)\cdot(u-s)/(t-s)+\upsilon], \forall u \in [s,t])$.

$\tilde{\gamma}_{s,t}^{\ell,\upsilon}(x,y)$ can be represented as the limit of an alternating Cauchy sequence (simply consider a transformation of (6.5) where $x = y$). As such, in analogous fashion to Algorithm 6.1.2, retrospective Bernoulli sampling (see Section 2.7) can be employed as detailed in Algorithm 6.1.3 to simulate events of probability $\tilde{\gamma}_{s,t}^{\ell,\upsilon}(x,y)$ unbiasedly by means of the following alternating Cauchy sequence (for $k \geq 0$),

$$S_{2k}^{\gamma*} := 1 - \sum_{j=1}^{k} \left\{ \tilde{\varsigma}_{s,t}^{\ell,\upsilon}(j;x,y) - \tilde{\varphi}_{s,t}^{\ell,\upsilon}(j;x,y) \right\}, \qquad S_{2k+1}^{\gamma*} := S_{2k}^{\gamma*} - \tilde{\varsigma}_{s,t}^{\ell,\upsilon}(k+1;x,y). \quad (6.14)$$

We now consider how to simulate events of probability corresponding to the probability that a Bessel bridge sample path, with known minimum (or maximum) point, stays within a particular interval. In Figure 6.1.4 we show example sample path trajectories of a Brownian bridge $W \sim \mathbb{W}_{s,t}^{x,y} \,\big|\, (X_\tau = \hat{m})$ (i.e. a Bessel bridge as in Section 2.8.2), which remains in the interval $[\hat{m}, \upsilon]$. In Theorems 6.1.2 and 6.1.3 we outline a result (first shown in Beskos et al. [2008]) that shows that such a probability can be represented as the limit

Algorithm 6.1.3 Simulating an event of probability $\tilde{\gamma}^{\ell,\upsilon}_{s,t}(x,y)$.

1. Simulate $u \sim \mathrm{U}[0,1]$ and set $k = 1$.
2. While $u \in \left(S^{\gamma*}_{2k+1}, S^{\gamma*}_{2k}\right)$ (where $S^{\gamma*}_{2k} := 1 - \sum_{j=1}^{k}\left\{\tilde{\varsigma}^{\ell,\upsilon}_{s,t}(j;x,y) - \tilde{\varphi}^{\ell,\upsilon}_{s,t}(j;x,y)\right\}$ and $S^{\gamma*}_{2k+1} := S^{\gamma*}_{2k} - \tilde{\varsigma}^{\ell,\upsilon}_{s,t}(k+1;x,y)$) then $k = k+1$.
3. If $u \leq S^{\gamma*}_{2k+1}$ then $u < p$ so return 1 else $u > p$ so return 0.

of an infinite series. However, we critically show (in Corollary 6.1.3 and Algorithms 6.1.4 and 6.1.5) that it is possible to simulate events of such a probability without assumption on the size of the interval (unlike existing methods [Beskos et al., 2008, Prop. 3] in which one requires that the interval is of size greater than $3(\upsilon - \hat{m})^2/(t-s)$).

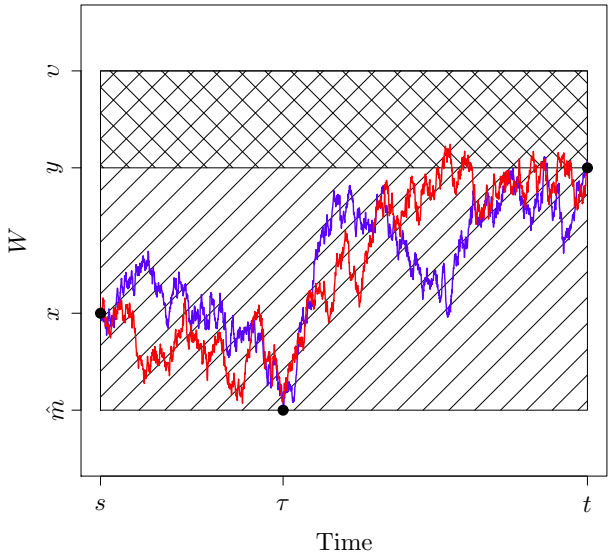


Figure 6.1.4: Example sample path trajectories $W \sim \mathbb{W}^{x,y}_{s,t}\left|(X_\tau = \hat{m}, \check{m} \in [(x \vee y), \upsilon]\right.)$.

We begin with Definition 6.1.1, as it transpires that it is convenient to consider two possible cases of our desired probability in which either of the end points attain the sample path minimum (or maximum) or neither of them attains the minimum (or maximum).

Definition 6.1.1. *We allow* $\delta^{\hat{m},\upsilon}_{s,t}(x,y)$ *to denote the probability that a Bessel bridge sample path* $W \sim \mathbb{W}^{x,y}_{s,t} \,\big|\, \hat{m}$*, (with minimum* $\hat{m}$*) remains in the interval* $[\hat{m},\upsilon]$*. We further denote* $\delta^{\hat{m},\upsilon}_{s,t}(1;x,y) := \mathbb{P}\left(W \in [\hat{m},\upsilon] \,|\, W \geq \hat{m}, (x\wedge y) > \hat{m}\right)$ *and* $\delta^{\hat{m},\upsilon}_{s,t}(2;x,y) := \mathbb{P}\left(W \in [\hat{m},\upsilon] \,|\, W \geq \hat{m}, (x\wedge y) = \hat{m}\right)$ *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 $W \sim \mathbb{W}^{x,y}_{s,t} \,\big|\, \check{m}$ (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 (see the argument presented in Section 2.8.1).

We first consider the case where neither end point attains the Bessel bridge minimum (or maximum).

Theorem 6.1.2 ([Beskos et al., 2008, Prop. 3])**.** *The probability that a Bessel bridge sample path* $W \sim \mathbb{W}^{x,y}_{s,t} \,\big|\, \hat{m}$*, (with minimum* $\hat{m} < (x\wedge y)$*) remains in the interval* $[\hat{m},\upsilon]$ *(i.e.* $\forall\, u \in [s,t]$ *we have* $W_u \in [\hat{m},\upsilon]$*) can be represented by the following infinite series,*

$$\begin{aligned}\delta^{\hat{m},\upsilon}_{s,t}(1;x,y) &:= \mathbb{P}\left(W \in [\hat{m},\upsilon] \,|\, W \geq \hat{m}, (x\wedge y) > \hat{m}\right)\\ &= \frac{\gamma^{\hat{m},\upsilon}_{s,t}(x,y)}{1-\exp\left\{-2\dfrac{(x-\hat{m})\cdot(y-\hat{m})}{t-s}\right\}}.\end{aligned} \tag{6.15}$$

Proof. We begin by considering the behaviour of a Brownian bridge in the interval $[\hat{m},\upsilon]$ conditioned on remaining in the interval $[\hat{m},\mathcal{U}]$ where $\mathcal{U} \geq \upsilon$. Denoting by $\delta^{\hat{m},\upsilon}_{s,t}(x,y\,|\,\mathcal{U})$ as the probability that a sample path from a Brownian bridge starting at $W_s = x$ and ending at $W_t = y$ remains in the interval $[\hat{m},\upsilon]$ conditioned on it remaining in the interval $[\hat{m},\mathcal{U}]$ we have (noting that $\hat{m} \leq (x\wedge y) \leq (x\vee y) \leq \upsilon \leq \mathcal{U}$),

$$\begin{aligned}\delta^{\hat{m},\upsilon}_{s,t}(x,y\,|\,\mathcal{U}) &:= \mathbb{P}\left(W^{x,y}_{s,t} \in [\hat{m},\upsilon] \,\middle|\, W^{x,y}_{s,t} \in [\hat{m},\mathcal{U}]\right)\\ &= \frac{\mathbb{P}\left(W^{x,y}_{s,t} \in [\hat{m},\upsilon]\right)}{\mathbb{P}\left(W^{x,y}_{s,t} \in [\hat{m},\mathcal{U}]\right)} =: \frac{\gamma^{\hat{m},\upsilon}_{s,t}(x,y)}{\gamma^{\hat{m},\mathcal{U}}_{s,t}(x,y)}.\end{aligned} \tag{6.16}$$

Now, considering the limit as $\mathcal{U} \to \infty$ (recalling that in this case $(x\wedge y) > \hat{m}$ and from Pötzelberger and Wang [2001] we have that $\gamma^{\hat{m},\infty}_{s,t}(x,y) = 1-\exp\{-2(x-\hat{m})(y-\hat{m})/(t-s)\}$)

we arrive at our desired result,

$$\begin{aligned}\delta_{s,t}^{\hat{m},\upsilon}(1;x,y) &:= \mathbb{P}\left(W \in [\hat{m},\upsilon] \,|\, W \geq \hat{m}, (x \wedge y) > \hat{m}\right) \\ &= \lim_{\mathcal{U}\to\infty}\left[\delta_{s,t}^{\hat{m},\upsilon}(1;x,y\,|\,\mathcal{U})\right] = \frac{\gamma_{s,t}^{\hat{m},\upsilon}(x,y)}{1-\exp\left\{-2\dfrac{(x-\hat{m})\cdot(y-\hat{m})}{t-s}\right\}}. \end{aligned} \tag{6.17}$$

□

Corollary 6.1.2. *Events of probability $\delta_{s,t}^{\hat{m},\upsilon}(1;x,y)$ can be represented as the limit as $k \to \infty$ of the following alternating Cauchy sequence,*

$$S_k^{\delta,1} := \frac{S_k^{\gamma}}{1-\exp\left\{-2\dfrac{(x-\hat{m})\cdot(y-\hat{m})}{t-s}\right\}}. \tag{6.18}$$

Proof. We begin by noting that $\lim_{k\to\infty} S_k^{\delta,1} = \delta_{s,t}^{\hat{m},\upsilon}(1;x,y)$. To show that $S_k^{\delta,1}$ is an alternating Cauchy sequence note that as S_k^{γ} is an alternating Cauchy sequence (see Corollary 6.1.1) and $1-\exp\{-2(x-\hat{m})(y-\hat{m})/(t-s)\}$ is a constant then $S_k^{\delta,1}$ is a linear transformation of an alternating Cauchy sequence. As such, by direct application of Corollary 2.7.1 we have that $S_k^{\delta,1}$ is an alternating Cauchy sequence. □

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

Theorem 6.1.3 ([Beskos et al., 2008, Prop. 3])**.** *The probability that a Bessel bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y}\,\big|\,\hat{m}$, (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 by the following infinite series,*

$$\begin{aligned}\delta_{s,t}^{\hat{m},\upsilon}(2;x,y) &:= \mathbb{P}\left(W \in [\hat{m},\upsilon] \,|\, W \geq \hat{m}\right) \\ &= 1 - \frac{1}{(y-\hat{m})}\sum_{j=1}^{\infty}\left\{\psi_{s,t}^{\hat{m},\upsilon}(j;y) - \chi_{s,t}^{\hat{m},\upsilon}(j;y)\right\}, \end{aligned} \tag{6.19}$$

where we denote,

$$\psi_{s,t}^{\hat{m},\upsilon}(j;y) := \Big(2\big|\upsilon-\hat{m}\big|\, j-(y-\hat{m})\Big)\exp\left\{-\frac{2\big|\upsilon-\hat{m}\big|\, j}{t-s}\Big(\big|\upsilon-\hat{m}\big|\, j-(y-\hat{m})\Big)\right\}, \tag{6.20}$$

$$\chi_{s,t}^{\hat{m},\upsilon}(j;y) := \Big(2\big|\upsilon-\hat{m}\big|\, j+(y-\hat{m})\Big)\exp\left\{-\frac{2\big|\upsilon-\hat{m}\big|\, j}{t-s}\Big(\big|\upsilon-\hat{m}\big|\, j+(y-\hat{m})\Big)\right\}. \tag{6.21}$$

Proof. Recall from the proof of Theorem 6.1.2 that we considered our desired probability by means of consideration of a Brownian bridge in the interval $[\hat{m},\upsilon]$ conditioned on remaining in the interval $[\hat{m},\mathcal{U}]$. In particular, supposing that $x=\hat{m}<y$ (without loss of generality), we have,

$$\delta_{s,t}^{\hat{m},\upsilon}(\hat{m},y\,|\,\mathcal{U}) = \frac{\gamma_{s,t}^{\hat{m},\upsilon}(\hat{m},y)}{\gamma_{s,t}^{\hat{m},\mathcal{U}}(\hat{m},y)}. \tag{6.22}$$

Now, considering the infinite series representation of the probability $\gamma_{s,t}^{\hat{m},\upsilon}(\hat{m},y)$ (see Theorem 6.1.1), note that a number of the constituent elementary terms will cancel with one another. In particular,

$$\begin{aligned}\bar{\varsigma}_{s,t}^{\hat{m},\upsilon}(j;\hat{m},y) &= \exp\left\{-\frac{2\big|\upsilon-\hat{m}\big|\, j}{t-s}\Big(\big|\upsilon-\hat{m}\big|\, j-(y-\hat{m})\Big)\right\}\\ &= \bar{\varphi}_{s,t}^{\hat{m},\upsilon}(j;\hat{m},y), \qquad (6.23)\\ \bar{\varsigma}_{s,t}^{-\hat{m},-\upsilon}(j;-\hat{m},-y) &= \exp\left\{-\frac{2\big|\upsilon-\hat{m}\big|\,(j-1)}{t-s}\Big(\big|\upsilon-\hat{m}\big|\,(j-1)+(y-\hat{m})\Big)\right\}\\ &= \bar{\varphi}_{s,t}^{-\hat{m},-\upsilon}(j-1;-\hat{m},-y), \qquad (6.24)\end{aligned}$$

and so we have,

$$\begin{aligned}\gamma_{s,t}^{\hat{m},\upsilon}(\hat{m},y) &= 1-\sum_{j=1}^{\infty}\Big\{\bar{\varsigma}_{s,t}^{\hat{m},\upsilon}(j;\hat{m},y)+\bar{\varsigma}_{s,t}^{-\hat{m},-\upsilon}(j;-\hat{m},-y)-\bar{\varphi}_{s,t}^{\hat{m},\upsilon}(j;\hat{m},y)-\bar{\varphi}_{s,t}^{-\hat{m},-\upsilon}(j;-\hat{m},-y)\Big\}\\ &= 1-\bar{\varsigma}_{s,t}^{-\hat{m},-\upsilon}(1;-\hat{m},-y)=0. \qquad (6.25)\end{aligned}$$

Similarly we have $\gamma_{s,t}^{\hat{m},\mathcal{U}}(\hat{m},y)=0$ and so we can not directly consider the limit as $\mathcal{U}\to\infty$ in (6.22). Instead we consider the limit as $x\to\hat{m}$ and apply l'Hôpital's rule[1] and introduce

[1] *l'Hôpital's rule*: If $\lim_{x\to c} f(x)=\lim_{x\to c} g(x)=0$ and $\exists\lim_{x\to c}\frac{f'(x)}{g'(x)}$ and $g'(x)\neq 0$ then $\lim_{x\to c}\frac{f(x)}{g(x)}=$

the following notation to denote the differentials of the elementary functions,

$$\begin{aligned}
\varsigma'_j(+,\upsilon) &:= \frac{\mathrm{d}}{\mathrm{d}x}\bar{\varsigma}^{\hat{m},\upsilon}_{s,t}(j;x,y)\Big|_{x=\hat{m}} \\
&= \frac{2}{t-s}\left(\left|\upsilon-\hat{m}\right| j-(y-\hat{m})\right)\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\left[\left|\upsilon-\hat{m}\right| j-(y-\hat{m})\right]\right\}, \\
\varsigma'_j(-,\upsilon) &:= \frac{\mathrm{d}}{\mathrm{d}x}\bar{\varsigma}^{-\hat{m},-\upsilon}_{s,t}(j;-x,-y)\Big|_{x=\hat{m}} \\
&= -\frac{2}{t-s}\left(\left|\upsilon-\hat{m}\right|(j-1)+(y-\hat{m})\right)\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right|(j-1)}{t-s}\left(\left|\upsilon-\hat{m}\right|(j-1)+(y-\hat{m})\right)\right\}, \\
\varphi'_j(+,\upsilon) &:= \frac{\mathrm{d}}{\mathrm{d}x}\bar{\varphi}^{\hat{m},\upsilon}_{s,t}(j;x,y)\Big|_{x=\hat{m}} = -\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\left(\left|\upsilon-\hat{m}\right| j-(y-\hat{m})\right)\right\}, \\
\varphi'_j(-,\upsilon) &:= \frac{\mathrm{d}}{\mathrm{d}x}\bar{\varphi}^{-\hat{m},-\upsilon}_{s,t}(j;-x,-y)\Big|_{x=\hat{m}} = \frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\left(\left|\upsilon-\hat{m}\right| j+(y-\hat{m})\right)\right\},
\end{aligned}$$

and so we find the following representation of our desired probability by reconsidering (6.22),

$$\begin{aligned}
\delta^{\hat{m},\upsilon}_{s,t}(2;x,y) &:= \mathbb{P}\left(W\in[\hat{m},\upsilon]\,|\,W\geq\hat{m}\right) = \lim_{\mathcal{U}\to\infty}\left[\delta^{\hat{m},\upsilon}_{s,t}(\hat{m},y\,|\,\mathcal{U})\right] \\
&= \lim_{\mathcal{U}\to\infty}\left[\lim_{x\to\hat{m}}\frac{\gamma^{\hat{m},\upsilon}_{s,t}(x,y)}{\gamma^{\hat{m},\mathcal{U}}_{s,t}(x,y)}\right] = \lim_{\mathcal{U}\to\infty}\left[\lim_{x\to\hat{m}}\frac{\frac{\mathrm{d}}{\mathrm{d}x}\gamma^{\hat{m},\upsilon}_{s,t}(x,y)}{\frac{\mathrm{d}}{\mathrm{d}x}\gamma^{\hat{m},\mathcal{U}}_{s,t}(x,y)}\right] \\
&= \lim_{\mathcal{U}\to\infty}\left[\frac{-\sum_{j=1}^{\infty}\left\{\varsigma'_j(+,\upsilon)+\varsigma'_j(-,\upsilon)-\varphi'_j(+,\upsilon)-\varphi'_j(-,\upsilon)\right\}}{-\sum_{j=1}^{\infty}\left\{\varsigma'_j(+,\mathcal{U})+\varsigma'_j(-,\mathcal{U})-\varphi'_j(+,\mathcal{U})-\varphi'_j(-,\mathcal{U})\right\}}\right] \\
&= \lim_{\mathcal{U}\to\infty}\left[\frac{\frac{t-s}{2}\left[-\varsigma'_1(-,\upsilon)-\sum_{j=1}^{\infty}\left\{\varsigma'_j(+,\upsilon)+\varsigma'_{j+1}(-,\upsilon)-\varphi'_j(+,\upsilon)-\varphi'_j(-,\upsilon)\right\}\right]}{\frac{t-s}{2}\left[-\varsigma'_1(-,\mathcal{U})-\sum_{j=1}^{\infty}\left\{\varsigma'_j(+,\mathcal{U})+\varsigma'_{j+1}(-,\mathcal{U})-\varphi'_j(+,\mathcal{U})-\varphi'_j(-,\mathcal{U})\right\}\right]}\right] \\
&= \lim_{\mathcal{U}\to\infty}\left[\frac{(y-\hat{m})-\sum_{j=1}^{\infty}\left\{\psi^{\hat{m},\upsilon}_{s,t}(j;y)-\chi^{\hat{m},\upsilon}_{s,t}(j;y)\right\}}{(y-\hat{m})-\sum_{j=1}^{\infty}\left\{\psi^{\hat{m},\mathcal{U}}_{s,t}(j;y)-\chi^{\hat{m},\mathcal{U}}_{s,t}(j;y)\right\}}\right] \\
&= 1-\frac{\sum_{j=1}^{\infty}\left\{\psi^{\hat{m},\upsilon}_{s,t}(j;y)-\chi^{\hat{m},\upsilon}_{s,t}(j;y)\right\}}{(y-\hat{m})},
\end{aligned} \tag{6.26}$$

$\lim_{x\to c}\frac{f'(x)}{g'(x)}$.

where we denote,

$$\psi_{s,t}^{\hat{m},\upsilon}(j;y) := \left(2\left|\upsilon-\hat{m}\right| j-(y-\hat{m})\right)\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\left(\left|\upsilon-\hat{m}\right| j-(y-\hat{m})\right)\right\}, \quad (6.27)$$

$$\chi_{s,t}^{\hat{m},\upsilon}(j;y) := \left(2\left|\upsilon-\hat{m}\right| j+(y-\hat{m})\right)\exp\left\{-\frac{2\left|\upsilon-\hat{m}\right| j}{t-s}\left(\left|\upsilon-\hat{m}\right| j+(y-\hat{m})\right)\right\}. \quad (6.28)$$

□

Remark 6.1.1 ([Beskos et al., 2008, Prop. 3])**.** *As before, we can consider the probability a Bessel bridge sample path* $W \sim \mathbb{W}_{s,t}^{x,y}\left|\,\hat{m}\right.$*, (with minimum* $\hat{m} = y < x$*) remains in the interval* $[\hat{m},\upsilon]$ *by a simple reflection argument of Theorem 6.1.3.*

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 [Beskos et al., 2008, Prop. 3] in which one requires that $3(\upsilon-\hat{m})^2 > (t-s)$).

Corollary 6.1.3. *After the inclusion of the first* $\hat{k} := \sqrt{(t-s)+|\upsilon-\hat{m}|^2}/\left(2|\upsilon-\hat{m}|\right)$ *terms,* $\delta_{s,t}^{\hat{m},\upsilon}(2;x,y)$ *can be represented as the limit as* $k\to\infty$ *of the following alternating Cauchy sequence (where* $k\in\mathbb{N}$ *such that* $k\geq\hat{k}$*),*

$$S_{2k}^{\delta,2} := 1-\frac{1}{(y-\hat{m})}\sum_{j=1}^{k}\left\{\psi_{s,t}^{\hat{m},\upsilon}(j;y)-\chi_{s,t}^{\hat{m},\upsilon}(j;y)\right\}, \quad S_{2k+1}^{\delta,2} := S_{2k}^{\delta,2}-\frac{1}{y-\hat{m}}\psi_{s,t}^{\hat{m},\upsilon}(k+1;y). \quad (6.29)$$

Proof. As $(y-\hat{m})\in(0,(\upsilon-\hat{m})]$ then $\forall j$ we have $\psi_{s,t}^{\hat{m},\upsilon}(j;y),\chi_{s,t}^{\hat{m},\upsilon}(j;y)\geq 0$. As such to show that after the inclusion of the first $\hat{k}$ that $\delta_{s,t}^{\hat{m},\upsilon}(2;x,y)$ can be represented as the limit of an alternating Cauchy sequence it is sufficient to show that $\forall j\geq\hat{k}$ that $\psi_{s,t}^{\hat{m},\upsilon}(j;y)\geq\chi_{s,t}^{\hat{m},\upsilon}(j;y)\geq\psi_{s,t}^{\hat{m},\upsilon}(j+1;y)\geq\chi_{s,t}^{\hat{m},\upsilon}(j+1;y)\geq\ldots$ which can be proved inductively by first showing that $\forall j\geq\hat{k}$ that $\psi_{s,t}^{\hat{m},\upsilon}(j;y)/\chi_{s,t}^{\hat{m},\upsilon}(j;y)\geq 1$ and then $\forall j\geq\hat{k}$ that $\chi_{s,t}^{\hat{m},\upsilon}(j;y)/\psi_{s,t}^{\hat{m},\upsilon}(j+1;y)\geq 1$.

We begin by considering $\psi^{\hat{m},\upsilon}_{s,t}(j;y)/\chi^{\hat{m},\upsilon}_{s,t}(j;y)$,

$$\begin{aligned}\frac{\psi^{\hat{m},\upsilon}_{s,t}(j;y)}{\chi^{\hat{m},\upsilon}_{s,t}(j;y)} &= \frac{\left(2\left|\upsilon-\hat{m}\right|j-(y-\hat{m})\right)\exp\left\{-\dfrac{2\left|\upsilon-\hat{m}\right|j}{t-s}\left(\left|\upsilon-\hat{m}\right|j-(y-\hat{m})\right)\right\}}{\left(2\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)\exp\left\{-\dfrac{2\left|\upsilon-\hat{m}\right|j}{t-s}\left(\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)\right\}} \\ &= \frac{\left(2\left|\upsilon-\hat{m}\right|j-(y-\hat{m})\right)}{\left(2\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)}\exp\left\{\frac{4\left|\upsilon-\hat{m}\right|(y-\hat{m})\,j}{t-s}\right\} =: f(y). \qquad (6.30)\end{aligned}$$

Now, as we have that $|\upsilon-\hat{m}|>0$, $(y-\hat{m})>0$ and $(t-s)>0$ and we are interested in showing conditions for which $\psi^{\hat{m},\upsilon}_{s,t}(j;y)/\chi^{\hat{m},\upsilon}_{s,t}(j;y)\geq 1$ then we naturally consider the behaviour of $\psi^{\hat{m},\upsilon}_{s,t}(j;y)/\chi^{\hat{m},\upsilon}_{s,t}(j;y)$ as a function of y. In particular we consider the derivative of $f(y)$,

$$\begin{aligned}f'(y) &= \frac{4\left|\upsilon-\hat{m}\right|j}{\left[2\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right]^2\cdot(t-s)}\exp\left\{\frac{4\left|\upsilon-\hat{m}\right|(y-\hat{m})j}{(t-s)}\right\} \\ &\qquad\cdot\left[4\left|\upsilon-\hat{m}\right|^2j^2-(y-\hat{m})^2-(t-s)\right] \\ &> \frac{4j}{\left|\upsilon-\hat{m}\right|\cdot(2j+1)^2\cdot(t-s)}\left[\left|\upsilon-\hat{m}\right|^2\left(4j^2-1\right)-(t-s)\right], \qquad (6.31)\end{aligned}$$

and so we have that $f'(y)>0$ if $[|\upsilon-\hat{m}|^2(4j^2-1)-(t-s)]\geq 0$ which is ensured if $j\geq\hat{k}=\sqrt{(t-s)+|\upsilon-\hat{m}|^2}/\left(2|\upsilon-\hat{m}|\right)$. As $f'(y)>0$ with this selection of j we have an increasing function and so $f(y)$ is minimised by selecting $y=\hat{m}$ where, with reference to (6.30), we find that $f(\hat{m})=1$. As such we have that $\forall\, j\geq\hat{k}$ that $\psi^{\hat{m},\upsilon}_{s,t}(j;y)/\chi^{\hat{m},\upsilon}_{s,t}(j;y)\geq 1$ as desired.

Now, considering $\chi^{\hat{m},\upsilon}_{s,t}(j;y)/\psi^{\hat{m},\upsilon}_{s,t}(j+1;y)$ we similarly have,

$$\begin{aligned}\frac{\chi^{\hat{m},\upsilon}_{s,t}(j;y)}{\psi^{\hat{m},\upsilon}_{s,t}(j+1;y)} &= \frac{\left(2\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)\exp\left\{-\dfrac{2\left|\upsilon-\hat{m}\right|j}{t-s}\left(\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)\right\}}{\left(2\left|\upsilon-\hat{m}\right|(j+1)-(y-\hat{m})\right)\exp\left\{-\dfrac{2\left|\upsilon-\hat{m}\right|(j+1)}{t-s}\left(\left|\upsilon-\hat{m}\right|(j+1)-(y-\hat{m})\right)\right\}} \\ &= \frac{\left(2\left|\upsilon-\hat{m}\right|j+(y-\hat{m})\right)}{\left(2\left|\upsilon-\hat{m}\right|(j+1)-(y-\hat{m})\right)}\exp\left\{\frac{2\left|\upsilon-\hat{m}\right|(2j+1)}{t-s}\left[\left|\upsilon-\hat{m}\right|-(y-\hat{m})\right]\right\} =: g(y). \qquad (6.32)\end{aligned}$$

As before we have that $|\upsilon - \hat{m}| > 0$, $(y - \hat{m}) > 0$ and $(t - s) > 0$ and so we consider the behaviour of (6.32) as a function of y and find the derivative of $g(y)$,

$$\begin{aligned} g'(y) &= \frac{2\left|\upsilon - \hat{m}\right|(2j+1)}{\left[2\left|\upsilon - \hat{m}\right|(j+1) - (y - \hat{m})\right]^2 \cdot (t-s)} \exp\left\{\frac{2\left|\upsilon - \hat{m}\right|(2j+1)}{t-s}\left[\left|\upsilon - \hat{m}\right| - (y - \hat{m})\right]\right\} \\ &\quad \cdot \left[(t-s) - \left(4\left|\upsilon - \hat{m}\right|^2 j(j+1) + 2|\upsilon - \hat{m}|(y - \hat{m}) - (y - \hat{m})^2\right)\right] \\ &< \frac{2}{\left|\upsilon - \hat{m}\right|(2j+1)(t-s)}\left[(t-s) - \left(4|\upsilon - \hat{m}|^2 j(j+1) - \left|\upsilon - \hat{m}\right|^2\right)\right] \\ &\quad \cdot \exp\left\{\frac{2\left|\upsilon - \hat{m}\right|^2(2j+1)}{t-s}\right\} \\ &< \frac{2}{\left|\upsilon - \hat{m}\right|(2j+1)(t-s)}\left[(t-s) - 4\left|\upsilon - \hat{m}\right|^2 j^2\right] \cdot \exp\left\{\frac{2\left|\upsilon - \hat{m}\right|^2(2j+1)}{t-s}\right\}, \end{aligned} \quad (6.33)$$

and so we have that $g'(y) < 0$ if $[(t-s) - 4|\upsilon - \hat{m}|^2 j^2] \leq 0$ which is ensured if $j \geq \sqrt{(t-s)}/(2|\upsilon - \hat{m}|)$. As $g'(y) < 0$ with this selection of j we have a decreasing function and so $g(y)$ is minimised by selecting $y = \upsilon$ where, with reference to (6.32) we have $g(\upsilon) = 1$. As such $\forall\, j \geq \hat{k} > \sqrt{(t-s)}/(2|\upsilon - \hat{m}|)$ we have that $\psi^{\hat{m},\upsilon}_{s,t}(j;y)/\chi^{\hat{m},\upsilon}_{s,t}(j;y) \geq 1$ as desired.

As a result we have $\forall\, j \geq \hat{k}$ that $\psi^{\hat{m},\upsilon}_{s,t}(j;y) \geq \chi^{\hat{m},\upsilon}_{s,t}(j;y) \geq \psi^{\hat{m},\upsilon}_{s,t}(j{+}1;y) \geq \chi^{\hat{m},\upsilon}_{s,t}(j{+}1;y) \geq \ldots$ and so $\delta^{\hat{m},\upsilon}_{s,t}(2;x,y)$ can be represented as the limit of the alternating Cauchy in (6.29). □

As a consequence of Corollaries 6.1.2 and 6.1.3, events of probability $\delta^{\hat{m},\upsilon}_{s,t}(x,y)$ can be simulated unbiasedly by retrospective Bernoulli sampling (see Section 2.7) as detailed in Algorithm 6.1.4. Similarly, events of probability $\delta^{\ell,\check{m}}_{s,t}(x,y)$ can be simulated unbiasedly as detailed in Algorithm 6.1.5 by applying the reflection argument detailed in Section 2.8.1.

6.1.2 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 6.1.4 and 6.1.5 respectively), and events of this probability can be simulated (Corollaries

Algorithm 6.1.4 Simulating an event of probability $\delta_{s,t}^{\hat{m},\upsilon}(x,y)$.

1. Simulate $u \sim \mathrm{U}[0,1]$.
2. If $(x \wedge y) > \hat{m}$ then,
   (a) Set $k = 1$
   (b) While $u \in \left(S_{2k+1}^{\delta,1}, S_{2k}^{\delta,1}\right)$ (where $S_k^{\delta,1} := \dfrac{S_k^{\gamma}}{1-\exp\{-2(x-\hat{m})\cdot(y-\hat{m})/(t-s)\}}$) then $k = k+1$.
   (c) If $u \leq S_{2k+1}^{\delta,1}$ then $u < p$ so return 1 else $u > p$ so return 0.
3. If $(x \wedge y) = \hat{m}$ then,
   (a) Set $k = \left\lceil \sqrt{(t-s)+|\upsilon-\hat{m}|^2}/\left(2|\upsilon-\hat{m}|\right)\right\rceil$
   (b) While $u \in \left(S_{2k+1}^{\delta,2}, S_{2k}^{\delta,2}\right)$, where $S_{2k}^{\delta,2} := 1 - \dfrac{1}{|x-y|}\sum_{j=1}^{k}\left\{\psi_{s,t}^{\hat{m},\upsilon}(j;(x\vee y)) - \chi_{s,t}^{\hat{m},\upsilon}(j;(x\vee y))\right\}$ and $S_{2k+1}^{\delta,2} := S_{2k}^{\delta,2} - \dfrac{1}{|x-y|}\psi_{s,t}^{\hat{m},\upsilon}(k+1;(x\vee y))$ then $k = k+1$.
   (c) If $u \leq S_{2k+1}^{\delta,2}$ then $u < p$ so return 1 else $u > p$ so return 0.

Algorithm 6.1.5 Simulating an event of probability $\delta_{s,t}^{\ell,\check{m}}(x,y)$.

1. Set $x' := -x$, $y' := -y$, $\hat{m}' := -\check{m}$ and $\upsilon' := -\ell$.
2. Simulate event of probability $\delta_{s,t}^{\hat{m}',\upsilon'}(x',y')$ as per Algorithm 6.1.4.

6.1.4 and 6.1.5 respectively).

We begin by introducing the following simplifying notation, $q_{1:n} := \{q_1,\ldots,q_n\}$, $q_0 := s$ and $q_{n+1} := t$. We further denote $\hat{m}_{s,t} := \inf\{W_q; q \in [s,t]\}$, $\check{m}_{s,t} := \sup\{W_q; q \in [s,t]\}$, $\mathcal{W} := \left\{W_{q_1} = w_1,\ldots,W_{q_n} = w_n\right\}$, $\mathcal{L} := \left\{\hat{m}_{s,q_1} \in [\ell_{s,q_1}^{\downarrow}, \ell_{s,q_1}^{\uparrow}],\ldots,\hat{m}_{q_n,t} \in [\ell_{q_n,t}^{\downarrow}, \ell_{q_n,t}^{\uparrow}]\right\}$, $\mathcal{U} := \left\{\check{m}_{s,q_1} \in [\upsilon_{s,q_1}^{\downarrow}, \upsilon_{s,q_1}^{\uparrow}],\ldots,\check{m}_{q_n,t} \in [\upsilon_{q_n,t}^{\downarrow}, \upsilon_{q_n,t}^{\uparrow}]\right\}$. 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 6.1.4. *The probability a Brownian bridge sample path* $W \sim \mathbb{W}_{s,t}^{x,y}\big|\mathcal{W}$ *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 by the*

following infinite series,

$$
\begin{aligned}
{}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W}) &:= \mathbb{P}\left(\hat{m}_{s,t}\in[\ell\downarrow,\ell\uparrow],\check{m}_{s,t}\in[\upsilon\downarrow,\upsilon\uparrow]\,\middle|\,\mathcal{W}\right)\\
&= \left[\prod_{i=0}^{n}\gamma^{\ell\downarrow,\upsilon\uparrow}_{q_i,q_{i+1}}(X_{q_i},X_{q_{i+1}})\right]-\left[\prod_{i=0}^{n}\gamma^{\ell\uparrow,\upsilon\uparrow}_{q_i,q_{i+1}}(X_{q_i},X_{q_{i+1}})\right]\\
&\quad-\left[\prod_{i=0}^{n}\gamma^{\ell\downarrow,\upsilon\downarrow}_{q_i,q_{i+1}}(X_{q_i},X_{q_{i+1}})\right]+\left[\prod_{i=0}^{n}\gamma^{\ell\uparrow,\upsilon\downarrow}_{q_i,q_{i+1}}(X_{q_i},X_{q_{i+1}})\right]. \qquad (6.34)
\end{aligned}
$$

Proof. We begin by noting that by an inclusion-exclusion argument our desired probability can be represented as follows,

$$
\begin{aligned}
{}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W}) &= \mathbb{P}\left(W\in[\ell\downarrow,\upsilon\uparrow]\,|\,\mathcal{W}\right)-\mathbb{P}\left(W\in[\ell\uparrow,\upsilon\uparrow]\,|\,\mathcal{W}\right)\\
&\quad-\mathbb{P}\left(W\in[\ell\downarrow,\upsilon\downarrow]\,|\,\mathcal{W}\right)+\mathbb{P}\left(W\in[\ell\uparrow,\upsilon\downarrow]\,|\,\mathcal{W}\right). \qquad (6.35)
\end{aligned}
$$

Now, considering $\mathbb{P}\left(W\in[\ell\downarrow,\upsilon\uparrow]\,|\,\mathcal{W}\right)$, by applying the strong Markov property we have,

$$
\begin{aligned}
&\mathbb{P}\left(W\in[\ell\downarrow,\upsilon\uparrow]\,|\,\mathcal{W}\right)=\mathbb{P}\left(W_{[s,q_1]}\in[\ell\downarrow,\upsilon\uparrow],\ldots,W_{[q_n,t]}\in[\ell\downarrow,\upsilon\uparrow]\,\middle|\,\mathcal{W}\right)\\
&=\prod_{i=0}^{n}\mathbb{P}\left(W_{[q_i,q_{i+1}]}\in[\ell\downarrow,\upsilon\uparrow]\,\middle|\,W_{q_i}=w_i,W_{q_{i+1}}=w_{i+1}\right)=:\prod_{i=0}^{n}\gamma^{\ell\downarrow,\upsilon\uparrow}_{q_i,q_{i+1}}(X_{q_i},X_{q_{i+1}}). \qquad (6.36)
\end{aligned}
$$

As each of the terms in (6.35) can be similarly considered we arrive at (6.34) as desired. □

As we have ${}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W})=\mathbb{P}\left(\hat{m}_{s,t}\in[\ell\downarrow,\ell\uparrow],\check{m}_{s,t}\in[\upsilon\downarrow,\upsilon\uparrow],\mathcal{W}\right)/\mathbb{P}\left(\mathcal{W}\right)$ it is of interest to consider of those Brownian bridge sample paths with the restriction $\mathcal{W}$, which of those also have the restriction $\{\hat{m}_{s,t}\in[\ell\downarrow,\ell\uparrow],\check{m}_{s,t}\in[\upsilon\downarrow,\upsilon\uparrow]\}$ (sample paths of which are illustrated in Figure 6.1.5).

Corollary 6.1.4. *Events of probability* ${}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W})$ *can be represented as the limit as* $k\to\infty$ *of the following alternating Cauchy sequence,*

$$
S_k^{\rho(n)}:=\left[\prod_{i=0}^{n}S_k^{\gamma(q_i,q_{i+1},\downarrow,\uparrow)}\right]-\left[\prod_{i=0}^{n}S_{k+1}^{\gamma(q_i,q_{i+1},\uparrow,\uparrow)}\right]-\left[\prod_{i=0}^{n}S_{k+1}^{\gamma(q_i,q_{i+1},\downarrow,\downarrow)}\right]+\left[\prod_{i=0}^{n}S_k^{\gamma(q_i,q_{i+1},\uparrow,\downarrow)}\right]. \qquad (6.37)
$$

Proof. As shown in (6.34), the probability ${}^{(n)}\rho^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}_{s,t,x,y}(q_{1:n},\mathcal{W})$ can be represented as a function of γ probabilities, each of which can be represented as the limit of an alternating

Cauchy sequence. As such, as a consequence of Corollary 2.7.1, we have that (6.37) follows directly by alignment of the indices of the Cauchy sequence representations of the γ probabilities in (6.34), and is itself an alternating Cauchy sequence. □

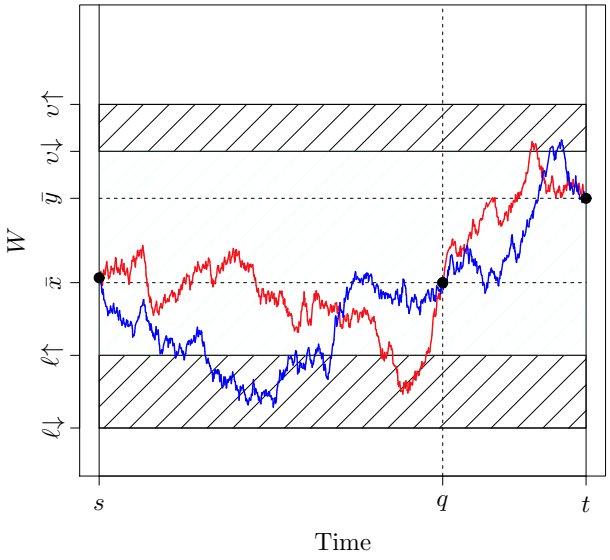
Definition 6.1.2. *We define* $\rho(s,q,t,x,w,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) := {}^{(1)}\rho_{s,t,x,y}^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}(\{q\},\{w\})$*, which coincides with* ρ *in Beskos et al. [2012].*

As a consequence of Corollary 6.1.4, events of probability ${}^{(n)}\rho_{s,t,x,y}^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}(q_{1:n},\mathcal{W})$ can be simulated unbiasedly by retrospective Bernoulli sampling (see Section 2.7) as detailed in Algorithm 6.1.6.

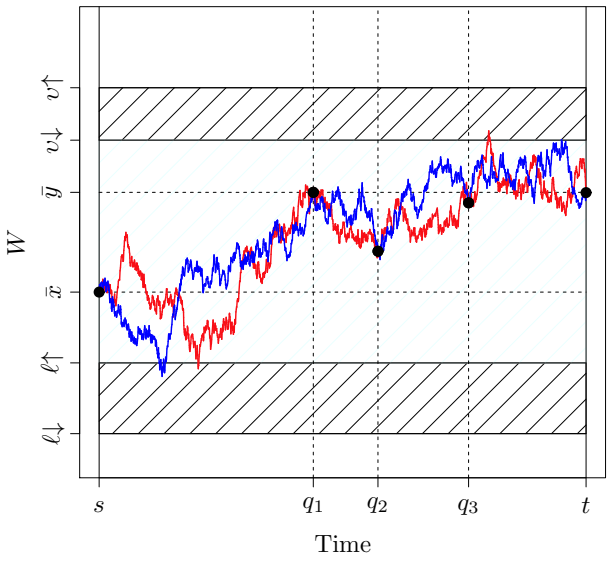
Algorithm 6.1.6 Simulating an event of probability ${}^{(n)}\rho_{s,t,x,y}^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}(q_{1:n},\mathcal{W})$.

---

1. Simulate $u \sim \mathrm{U}[0,1]$ and set $k = 1$.
2. While $u \in \left(S_{2k+1}^{\rho(n)}, S_{2k}^{\rho(n)}\right)$, where $S_k^{\rho(n)} = \left[\prod_{i=0}^n S_k^{\gamma(q_i,q_{i+1},\downarrow,\uparrow)}\right] - \left[\prod_{i=0}^n S_{k+1}^{\gamma(q_i,q_{i+1},\uparrow,\uparrow)}\right] - \left[\prod_{i=0}^n S_{k+1}^{\gamma(q_i,q_{i+1},\downarrow,\downarrow)}\right] + \left[\prod_{i=0}^n S_k^{\gamma(q_i,q_{i+1},\uparrow,\downarrow)}\right]$, then $k = k+1$.
3. If $u \leq S_{2k+1}^{\rho(n)}$ then $u < p$ so return 1 else $u > p$ so return 0.



(a) $n = 1$



(b) $n = 3$

Figure 6.1.5: Example sample path trajectories $W \sim \mathbb{W}_{s,t}^{x,y} \,\big|\, (\hat{m} \in [\ell\downarrow, \ell\uparrow], \check{m} \in [\upsilon\downarrow, \upsilon\uparrow], q_{1:n}, \mathcal{W})$.

Theorem 6.1.5. *The probability that a Brownian bridge sample path $W \sim \mathbb{W}_{s,t}^{x,y} \big| \mathcal{W}$, has in the sub-intervals between successive points a minimum and maximum in particular intervals ($\mathcal{L}$ and $\mathcal{U}$ respectively), can be represented by the following infinite series,*

$$
\begin{aligned}
&{}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W}) := \mathbb{P}\left(\mathcal{L},\mathcal{U}\,|\,\mathcal{W}\right) \\
&\Big[:= \mathbb{P}\Big(\hat{m}_{s,q_1} \in [\ell_{s,q_1}^{\downarrow},\ell_{s,q_1}^{\uparrow}], \check{m}_{s,q_1} \in [\upsilon_{s,q_1}^{\downarrow},\upsilon_{s,q_1}^{\uparrow}], \ldots, \hat{m}_{q_n,t} \in [\ell_{q_n,t}^{\downarrow},\ell_{q_n,t}^{\uparrow}], \check{m}_{q_n,t} \in [\upsilon_{q_n,t}^{\downarrow},\upsilon_{q_n,t}^{\uparrow}] \\
&\qquad\qquad \Big| W_s = x, W_{q_1} = w_1, \ldots, W_{q_n} = w_n, W_t = y\Big)\Big] \\
&= \prod_{i=0}^{n}\Big[\gamma_{q_i,q_{i+1}}^{\ell\downarrow,\upsilon\uparrow}(X_{q_i},X_{q_{i+1}}) - \gamma_{q_i,q_{i+1}}^{\ell\uparrow,\upsilon\uparrow}(X_{q_i},X_{q_{i+1}}) - \gamma_{q_i,q_{i+1}}^{\ell\downarrow,\upsilon\downarrow}(X_{q_i},X_{q_{i+1}}) + \gamma_{q_i,q_{i+1}}^{\ell\uparrow,\upsilon\downarrow}(X_{q_i},X_{q_{i+1}})\Big].
\end{aligned}
\tag{6.38}
$$

Proof. By the strong Markov property we have,

$$
\mathbb{P}\left(\mathcal{L},\mathcal{U}\,|\,\mathcal{W}\right) = \prod_{i=0}^{n}\left[\mathbb{P}\left(\hat{m}_{q_i,q_{i+1}} \in \left[\ell_{q_i,q_{i+1}}^{\downarrow},\ell_{q_i,q_{i+1}}^{\uparrow}\right], \check{m}_{q_i,q_{i+1}} \in \left[\upsilon_{q_i,q_{i+1}}^{\downarrow},\upsilon_{q_i,q_{i+1}}^{\uparrow}\right] \Big| W_{q_i}, W_{q_{i+1}}\right)\right].
$$

Now, applying a sample path inclusion-exclusion argument in a similar manner to the proof of Theorem 6.1.4 (i.e. we have $\mathbb{P}\left(\hat{m} \in [\ell\downarrow,\ell\uparrow], \check{m} \in [\upsilon\downarrow,\upsilon\uparrow]\right) = \mathbb{P}\left(W \in [\ell\downarrow,\upsilon\uparrow]\right) - \mathbb{P}\left(W \in [\ell\uparrow,\upsilon\uparrow]\right) - \mathbb{P}\left(W \in [\ell\downarrow,\upsilon\downarrow]\right) + \mathbb{P}\left(W \in [\ell\uparrow,\upsilon\downarrow]\right)$ then (6.38) follows directly. □

As before, note that ${}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W}) = \mathbb{P}\left(\mathcal{L},\mathcal{U},\mathcal{W}\right)/\mathbb{P}\left(\mathcal{W}\right)$, so it is of interest to consider of those Brownian bridge sample paths with the restriction $\mathcal{W}$ which of those also have the restriction $\{\mathcal{L},\mathcal{U}\}$ (sample paths of which are illustrated in Figure 6.1.6).

Corollary 6.1.5. *Events of probability ${}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W})$ can be represented as the limit as $k \to \infty$ of the following alternating Cauchy sequence,*

$$
S_k^{\beta(n)} := \prod_{i=0}^{n}\left[S_k^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\uparrow)} - S_{k+1}^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\downarrow)} + S_k^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\downarrow)}\right]. \tag{6.39}
$$

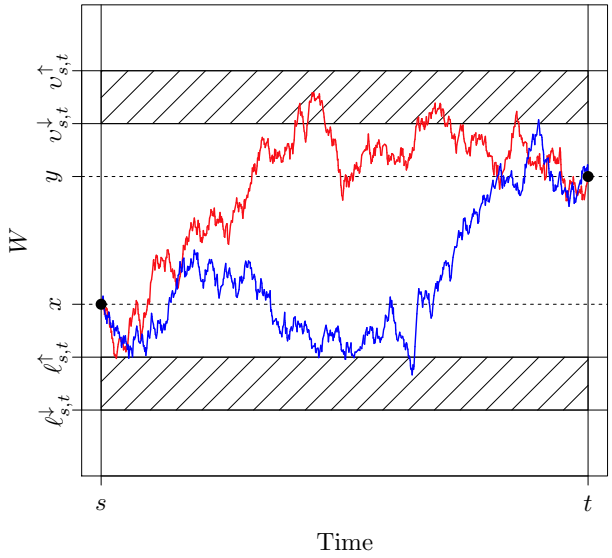
Proof. Follows by Corollary 2.7.1 in the same manner as in the proof of Corollary 6.1.4. □

Definition 6.1.3. *We define $\beta(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) := {}^{(0)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(\emptyset,\emptyset)$, which coincides with β in Beskos et al. [2012].*

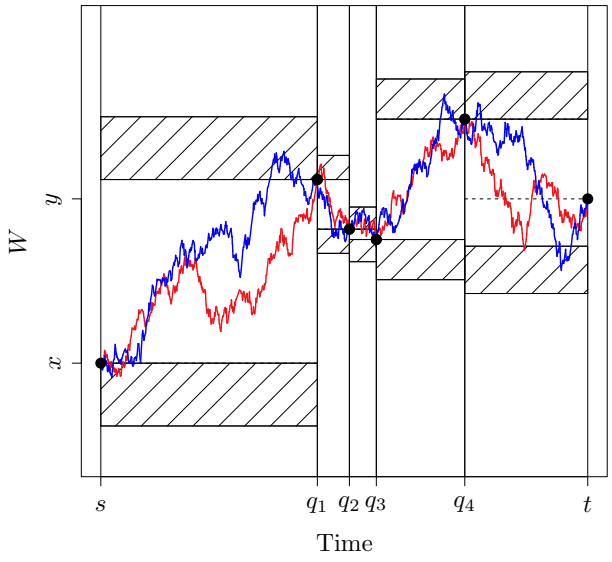
As a consequence of Corollary 6.1.4, events of probability ${}^{(n)}\beta_{s,t,x,y}^{\mathcal{L},\mathcal{U}}(q_{1:n},\mathcal{W})$ can be simulated unbiasedly by retrospective Bernoulli sampling as detailed in Algorithm 6.1.7.

Algorithm 6.1.7 Simulating an event of probability ${}^{(n)}\beta^{\mathcal{L},\mathcal{U}}_{s,t,x,y}(q_{1:n}, \mathcal{W})$.

1. Simulate $u \sim \mathrm{U}[0,1]$ and set $k = 1$.
2. While $u \in \left(S^{\beta(n)}_{2k+1}, S^{\beta(n)}_{2k}\right)$ (where $S^{\beta(n)}_{k} = \prod_{i=0}^{n}\left[S^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\uparrow)}_{k} - S^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\uparrow)}_{k+1} - S^{\gamma(q_i,q_{i+1},\ell\downarrow,\upsilon\downarrow)}_{k+1} + S^{\gamma(q_i,q_{i+1},\ell\uparrow,\upsilon\downarrow)}_{k}\right]$), then $k = k + 1$.
3. If $u \leq S^{\beta(n)}_{2k+1}$ then $u < p$ so return 1 else $u > p$ so return 0.



(a) $n = 0$



(b) $n = 4$

Figure 6.1.6: Example sample path trajectories $W \sim \mathbb{W}^{x,y}_{s,t} \,\big|\, (\mathcal{L}, \mathcal{U}, q_{1:n}, \mathcal{W})$.

6.2 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 5.1.3) and Conditioned Unbounded Exact Algorithm (CUEA; Algorithm 5.2.1), which in turn are used within the Bounded Jump Exact Algorithm (BJEA; Algorithm 5.3.1), Unbounded Jump Exact Algorithm (UJEA; Algorithm 5.3.2) and Conditioned Unbounded Jump Exact Algorithm (CUJEA; Algorithm 5.4.1). As the layered Brownian bridge construction for each of the exact algorithms above is identical, without loss of generality we relate the construction to the UEA (Algorithm 5.1.3). In particular, we address the simulation of layer information (Algorithm 5.1.3 Step 2), intermediate skeletal points (Algorithm 5.1.3 Step 4) and the process at further times after acceptance of the proposed sample path (Algorithm 5.1.3 Step 6).

We present two alternate layered Brownian bridge constructions based on extensions to existing exact algorithms. In Section 6.2.1 we present the *Bessel Approach*, which is a reinterpretation of part of the *Exact Algorithm 3* (EA3) proposed by Beskos et al. [2008], in which we incorporate the methodological improvements outlined in Chapter 5 and Section 6.1 and introduce a novel approach for conducting Algorithm 5.1.3 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 [Beskos et al., 2008]). Finally, in Section 6.2.2 we briefly outline a *Localised Approach* for constructing a layered Brownian bridge (based on the *Localised Exact Algorithm* (LEA) of Chen and Huang), showing that the resulting UEA only satisfies Principles 1 and 2 and discussing the difficulties in conducting Algorithm 5.1.3 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 5.1.3 Step 2). Instead, proposal sample path skeletons are simulated by other Monte Carlo techniques, including rejection sampling (see Section 2.4) and demarginalisation (see Section 2.3).

In the context of the exact algorithms of Chapter 5, demarginalisation allows us to 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.

6.2.1 Bessel Approach

The central idea in 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 5.1.3 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 the definition of a layer from Definition 2). We term this particular layer construction the *Bessel layer*. For instance, the ι^{th} Bessel layer is defined as follows,

$$\mathcal{I}_\iota = \left[(x \wedge y) - a_\iota, (x \vee y) + a_\iota\right]. \tag{6.40}$$

The (smallest) Bessel layer, $\mathcal{I} = \iota$, in which a particular Brownian bridge sample path is constrained can be simulated unbiasedly by retrospective Bernoulli sampling and inversion sampling (see Section 2.1) as detailed in Algorithm 6.2.1 (where we denote by $S_k^\gamma(s,t,x,y,\ell,\upsilon)$ as the alternating Cauchy sequence whose limit as $k \to \infty$ is $\gamma_{s,t}^{\ell,\upsilon}(x,y)$). The CDF of ι can be written as follows (with reference to Theorem 6.1.1 and as shown in Beskos et al. [2008]),

$$\mathbb{P}(\mathcal{I} \leq \iota) = \mathbb{P}\left(W_{s,t}^{x,y} \in \left[(x \wedge y) - a_\iota, (x \vee y) + a_\iota\right]\right) = \gamma_{s,t}^{(x\wedge y)-a_\iota,(x\vee y)+a_\iota}(x,y). \tag{6.41}$$

Algorithm 6.2.1 Simulation of a Brownian Bridge Bessel Layer [Beskos et al., 2008].

---

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

Now, we require a method of simulating intermediate points (Algorithm 5.1.3 Step 4) from a Brownian bridge sample path restricted to remain in the Bessel layer simulated in Algorithm 6.2.1. In particular, denoting with D_ι the set of sample paths which are

contained in the ι^{th} Bessel layer we have,

$$\begin{aligned}
D_\iota &:= \left(\left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_\iota, (x \wedge y)\Big]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big[(x \vee y), (x \vee y) + a_\iota\Big]\right\}\right) \\
&\quad \bigcap \left(\left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_{\iota-1}, (x \wedge y)\Big]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big[(x \vee y), (x \vee y) + a_{\iota-1}\Big]\right\}\right)^{\mathrm{C}} \\
&= \left(\left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_\iota, (x \wedge y)\Big]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big[(x \vee y), (x \vee y) + a_\iota\Big]\right\}\right) \\
&\quad \bigcap \left(\left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big(-\infty, (x \wedge y) - a_{\iota-1}\Big)\right\} \bigcup \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big((x \vee y) + a_{\iota-1}, \infty\Big)\right\}\right) \\
&= L_\iota \cup U_\iota, \qquad (6.42)
\end{aligned}$$

where,

$$L_\iota := \left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\Big)\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big[(x \vee y), (x \vee y) + a_\iota\Big]\right\}, \qquad (6.43)$$

$$U_\iota := \left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_\iota, (x \wedge y)\Big]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big((x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\Big]\right\}. \qquad (6.44)$$

Directly simulating intermediate points from from a sample path with law D_ι, (denoted $\mathbb{D}_\iota$) is not possible. Instead denoting by,

$$\hat{M}_\iota = \left\{W_{[s,t]} : \hat{m}_{s,t} \in \Big[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1}\Big]\right\}, \qquad (6.45)$$

$$\check{M}_\iota = \left\{W_{[s,t]} : \check{m}_{s,t} \in \Big[(x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota\Big]\right\}, \qquad (6.46)$$

and following the approach taken in Beskos et al. [2008], 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 $\mathbb{W}_{s,t}^{x,y}$ to the sets $\hat{M}_\iota$ and $\check{M}_\iota$, respectively) and accept them with probability given by the Radon-Nikodým derivative of $\mathbb{D}_\iota$ with respect to $\mathbb{B}_\iota$.

It was shown in Beskos et al. [2008] that $\mathbb{D}_\iota$ is absolutely continuous with respect to

$\mathbb{B}_\iota$ with Radon-Nikodým derivative,

$$\frac{\mathrm{d}\mathbb{D}_\iota}{\mathrm{d}\mathbb{B}_\iota}(x) \propto \frac{\mathbb{1}(W \in D_\iota)}{1 + \mathbb{1}(W \in \hat{M}_\iota \cap \check{M}_\iota)}. \tag{6.47}$$

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 (6.47). 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 $(\iota - 1)^{\text{th}}$ Bessel layer or with probability $1/2$ if it is contained between the $(\iota-1)^{\text{th}}$ and ι^{th} Bessel layer (and 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 2.7.1, and subsequently simulate any required intermediate points $\xi_1, \ldots, \xi_\kappa$ from a Bessel bridge as per Algorithm 2.8.4. As we can only simulate our sample path at a finite collection of points we can't directly evaluate (6.47). However, we can obtain an unbiased estimate and so simulate an event of this probability by application of Corollaries 6.1.2 and 6.1.3 and Lemmata 6.1.2 and 6.1.3 (letting $\chi_1, \ldots, \chi_{\kappa+3}$ as the order statistics of $\{\xi_1, \ldots, \xi_\kappa, s, \tau, t\}$),

$$\begin{aligned}\mathbb{P}_{\hat{\mathbb{M}}_\iota}(X \in D_\iota) &= \mathbb{P}\left(X \in [(x \wedge y) - a_\iota, (x \vee y) + a_\iota] \,\middle|\, X_{\chi_1}, \ldots, X_{\chi_{\kappa+3}}\right) \\ &= \prod_{i=1}^{\kappa+2} \delta^{\hat{m},(x\vee y)+a_\iota}_{\chi_i,\chi_{i+1}}\left(X_{\chi_i}, X_{\chi_{i+1}}\right), \end{aligned} \tag{6.48}$$

$$\mathbb{P}_{\hat{\mathbb{M}}_\iota}\left(X \in \hat{M}_\iota \cap \check{M}_\iota\right) = \mathbb{P}_{\hat{\mathbb{M}}_\iota}(X \in D_\iota) - \prod_{i=1}^{\kappa+2} \delta^{\hat{m},(x\vee y)+a_{\iota-1}}_{\chi_i,\chi_{i+1}}\left(X_{\chi_i}, X_{\chi_{i+1}}\right). \tag{6.49}$$

As both (6.48) and (6.49) are probabilities which can be represented as a linear function of δ probabilities, events of this probability can be simulated unbiasedly by retrospective Bernoulli sampling (as per Section 2.7, Corollary 2.7.1 and Algorithm 2.7.1). The synthesis of the above approach for simulating a Brownian bridge conditional on the Bessel layer simulated in Algorithm 6.2.1 (i.e. conducting Algorithm 5.1.3 Step 4) leads to Algorithm 6.2.2.

Upon accepting a proposed sample path skeleton within the UEA (as simulated by Algorithm 6.2.1 and Algorithm 6.2.2 and so satisfying Principles 1 and 2), we need to be able to simulate the sample path at further times (Algorithm 5.1.3 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 6.2.1), the sample path minimum (or maximum) belongs to a particular interval (as a con-

Algorithm 6.2.2 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_τ.
4. While $u_2 \in \Big(\prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}\big(\ell_1, \upsilon_1\big), \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}\big(\ell_1, \upsilon_1\big)\Big)$, $j = j+1$,
   (a) If $u_2 \leq \prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}\big(\ell_1, \upsilon_1\big)$ then accept sample path.
   (b) If $u_2 \geq \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}\big(\ell_1, \upsilon_1\big)$ while $u_2 \in \Big(\prod_{i=1}^{\kappa+2} S^{\delta}_{2k+1}\big(\ell_2, \upsilon_2\big), \prod_{i=1}^{\kappa+2} S^{\delta}_{2k}\big(\ell_2, \upsilon_2\big)\Big)$, $k = k+1$,
      i. If $u_2 \leq \prod_{i=1}^{\kappa+2} S^{\delta}_{2k+1}\big(\ell_2, \upsilon_2\big)$ then with probability $1/2$ accept sample path, else return to Step 1.
      ii. If $u_2 \geq \prod_{i=1}^{\kappa+2} S^{\delta}_{2k}\big(\ell_2, \upsilon_2\big)$ then reject sample path and return to Step 1.
5. Discard or Retain Auxiliary Information.

sequence of the mixture proposal in (6.46) and (6.45)), 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 2.7.1) 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 ι^{th} Bessel layer (by evaluating the Radon-Nikodým derivative in (6.47) by means of (6.48) and (6.49)). In

summary, we have four possible sets of conditional information for our sample path,

$$S_1 := \Big\{X_s, X_t, X \in D_\iota, \hat{m}_{s,t} \in \Big[(x\wedge y) - a_\iota, (x\wedge y) - a_{\iota-1}\Big], X_\tau = \hat{m}_{s,t}, \\ X_{\xi_1}, \ldots, X_{\xi_\kappa}, \check{m}_{s,t} \in \Big[(x\vee y), (x\vee y) + a_{\iota-1}\Big]\Big\}, \tag{6.50}$$

$$S_2 := \Big\{X_s, X_t, X \in D_\iota, \hat{m}_{s,t} \in \Big[(x\wedge y) - a_\iota, (x\wedge y) - a_{\iota-1}\Big], X_\tau = \hat{m}_{s,t}, \\ X_{\xi_1}, \ldots, X_{\xi_\kappa}, \check{m}_{s,t} \in \Big[(x\vee y) + a_{\iota-1}, (x\vee y) + a_\iota\Big]\Big\}, \tag{6.51}$$

$$S_3 := \Big\{X_s, X_t, X \in D_\iota, \check{m}_{s,t} \in \Big[(x\vee y) + a_{\iota-1}, (x\vee y) + a_\iota\Big], X_\tau = \check{m}_{s,t}, \\ X_{\xi_1}, \ldots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \Big[(x\wedge y) - a_{\iota-1}, (x\wedge y)\Big]\Big\}, \tag{6.52}$$

$$S_4 := \Big\{X_s, X_t, X \in D_\iota, \check{m}_{s,t} \in \Big[(x\vee y) + a_{\iota-1}, (x\vee y) + a_\iota\Big], X_\tau = \check{m}_{s,t}, \\ X_{\xi_1}, \ldots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \Big[(x\wedge y) - a_\iota, (x\wedge y) - a_{\iota-1}\Big]\Big\}. \tag{6.53}$$

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 dependancy 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 6.2.2 Step 2 is auxiliary sample path information (as in (2.7)) 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,

$$\tilde{S}_1 := \Big\{X_s, X_t, X_{\xi_1}, \ldots, X_{\xi_\kappa}, \hat{m}_{s,t} \in \Big[(x\wedge y) - a_\iota, (x\wedge y) - a_{\iota-1}\Big], \\ \check{m}_{s,t} \in \Big[(x\vee y), (x\vee y) + a_{\iota-1}\Big]\Big\}. \tag{6.54}$$

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 6.3.3 and Algorithm 6.3.5. Further intermediate points can then be simulated as outlined in Section 6.3.5.

6.2.2 Localised Approach

The *Localised Approach* is based on the layered Brownian bridge construction found in the *Localised Exact Algorithm* (LEA) originally proposed in [Chen and Huang; Giesecke and Smelov, Forthcoming]. The LEA is an alternative exact algorithm based on the mathematical framework of EA3 (see Beskos et al. [2008]). However, we don't go into detail as to its construction as in the context of this thesis it suffers from a number of computational weaknesses (in particular significant computation is required in order to satisfy Principle 3) so is not well suited for the purposes of this thesis. Instead, we briefly outline its construction and highlight which aspects of its construction present difficulties.

The key notion in the Localised approach is that rather than proposing sample path skeletons from $\mathbb{Z}^x$ (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 (5.13)). Each segment is successively simulated by means of simulating the first hitting time τ, of a Brownian motion proposal sample path (as outlined in Burq and Jones [2008]) of some user specified boundary symmetric around its start point (for instance, 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 (4.44). 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 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, this methodology can't be used to simulate conditioned diffusion and jump diffusion sample path skeletons. Finally, unlike the Bessel approach, the minimum or maximum that is simulated forms part of the skeleton and so can not be discarded. As such, the demarginalisation strategy taken in Section 6.2.1 in order to

extend the UEA (with the Bessel approach for simulating layered Brownian bridges) to satisfy Principle 3 can't be conducted.

6.3 Adaptive Layered Brownian Bridge Constructions

In Section 5.1.2 we proposed the Adaptive Unbounded Exact Algorithm (AUEA; Algorithm 5.1.4) as an alternative to the UEA (Algorithm 5.1.3). In this section we outline how to simulate finite dimensional skeletons of layered Brownian bridges for use within the AUEA and Conditioned Adaptive Unbounded Exact Algorithm (CAUEA; Algorithm 5.2.2), and by extension the Bounded Jump Exact Algorithm (BJEA; Algorithm 5.3.1), Adaptive Unbounded Jump Exact Algorithm (AUJEA; Algorithm 5.3.3) and Conditioned Adaptive Unbounded Jump Exact Algorithm (CAUJEA; Algorithm 5.4.2). As the layered Brownian bridge construction for each of these exact algorithms is identical, without loss of generality we relate the construction to the AUEA (Algorithm 5.1.4). In particular, we present new results for simulating an initial *intersection layer* (Algorithm 5.1.4 Step 2 – Section 6.3.1), intermediate points conditional on the layer (Algorithm 5.1.4 Step 3.1.2 – Section 6.3.2) and finally, new intersection layers for each sub-interval created by the intermediate point (Algorithm 5.1.4 Step 3.1.4 – Section 6.3.3).

We use the results we present in Sections 6.3.1–6.3.3 to outline novel layered Brownian bridge constructions in Section 6.3.5 which can be used within the AUEA, all of which satisfy Principles 1, 2 and 3.

6.3.1 Simulating an Initial Intersection Layer

Upon simulating a proposal Brownian bridge layer as per Algorithm 6.2.1 in Section 6.2.1, we know that our entire Brownian bridge sample path is contained within the ι^{th} Bessel layer, but is not contained within the $(\iota - 1)^{\text{th}}$ Bessel layer. Simulating sample path intermediate points is complicated by this conditional information (and as discussed in Section 6.2, it is not possible to simulate intermediate points directly). The novel approach we take in this thesis 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 5 and (6.42, 6.43, 6.44)) that, having simulated a layer for our proposal Brownian bridge sample path as per Algorithm 6.2.1, we know the sample path is restricted to the layer D_ι. We can then simply decom-

pose the set D_ι into a disjoint union and simulate to which our sample path belongs,

$$D_\iota = L_\iota \cup U_\iota = \underbrace{(L_\iota \cap U_\iota)}_{D_{\iota,1}} \uplus \underbrace{\left(U_\iota^{\mathrm{C}} \cap L_\iota\right)}_{D_{\iota,2}} \uplus \underbrace{\left(L_\iota^{\mathrm{C}} \cap U_\iota\right)}_{D_{\iota,3}}, \tag{6.55}$$

where,

$$\begin{aligned} D_{\iota,1} := L_\iota \cap U_\iota = &\left\{ W_{[s,t]} : \hat{m}_{s,t} \in \left[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1} \right) \right\} \\ &\bigcap \left\{ W_{[s,t]} : \check{m}_{s,t} \in \left((x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota \right] \right\}, \end{aligned} \tag{6.56}$$

$$\begin{aligned} D_{\iota,2} := U_\iota^{\mathrm{C}} \cap L_\iota = &\left\{ W_{[s,t]} : \hat{m}_{s,t} \in \left[(x \wedge y) - a_\iota, (x \wedge y) - a_{\iota-1} \right) \right\} \\ &\bigcap \left\{ W_{[s,t]} : \check{m}_{s,t} \in \left[(x \vee y), (x \vee y) + a_{\iota-1} \right] \right\}, \end{aligned} \tag{6.57}$$

$$\begin{aligned} D_{\iota,3} := L_\iota^{\mathrm{C}} \cap U_\iota = &\left\{ W_{[s,t]} : \hat{m}_{s,t} \in \left[(x \wedge y) - a_{\iota-1}, (x \wedge y) \right] \right\} \\ &\bigcap \left\{ W_{[s,t]} : \check{m}_{s,t} \in \left((x \vee y) + a_{\iota-1}, (x \vee y) + a_\iota \right] \right\}. \end{aligned} \tag{6.58}$$

This decomposition can be interpreted as the sample path attains the ι^{th} 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 6.3.1. Recalling the definition of a layer from Definition 2, we term this particular layer construction the *Intersection Layer*.

Theorem 6.3.1 (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 $\ell\downarrow := (x \wedge y) - a_\iota$, $\ell\uparrow := (x \wedge y) - a_{\iota-1}$, $\upsilon\downarrow := (x \vee y) + a_{\iota-1}$, $\upsilon\uparrow := (x \vee y) + a_\iota$),*

$$\begin{aligned} p_{D_{\iota,1}} &:= \mathbb{P}\left(D_{\iota,1}, \middle| D_\iota, W_s = x, W_t = y\right) \\ &= \frac{\beta(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow)}{\beta(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) + \beta(s,t,x,y,\ell\downarrow,\ell\uparrow,(x\vee y),\upsilon\downarrow) \\ &\qquad + \beta(s,t,x,y,\ell\uparrow,(x\wedge y),\upsilon\downarrow,\upsilon\uparrow)} \end{aligned} \tag{6.59}$$

Proof. By Bayes rule we have (recalling the decomposition of D_ι in (6.55)),

$$\mathbb{P}\left(D_{\iota,1}, \middle| D_\iota, W_s = x, W_t = y\right) = \frac{\mathbb{P}\left(D_{\iota,1} \middle| W_s = x, W_t = y\right)}{\sum_{j=1}^{3} \mathbb{P}\left(D_{\iota,j} \middle| W_s = x, W_t = y\right)} \tag{6.60}$$

and so (6.59) follows directly from Theorem 6.1.5 and Definition 6.1.3. □

Corollary 6.3.1. *Events of probability $p_{D_{\iota,1}}$ can be represented as the limit as $k \to \infty$ of the following alternating Cauchy sequence,*

$$S_k^{D(\iota,1)} := \frac{S_k^{\beta}(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow)}{S_{k+1}^{\beta}(s,t,x,y,\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow) + S_{k+1}^{\beta}(s,t,x,y,\ell\downarrow,\ell\uparrow,(x\vee y),\upsilon\downarrow) + S_{k+1}^{\beta}(s,t,x,y,\ell\uparrow,(x\wedge y),\upsilon\downarrow,\upsilon\uparrow)}. \quad (6.61)$$

Proof. Follows by Corollary 2.7.1 in the same manner as in the proof of Corollary 6.1.4. □

As a consequence of Corollary 6.3.1, events of probability $p_{D_{\iota,1}}$ can be simulated unbiasedly by retrospective Bernoulli sampling (as per Algorithm 2.7.1 in Section 2.7). Now, 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 $p_{D_{\iota,2}} + p_{D_{\iota,3}} = 1 - p_{D_{\iota,1}}$ then it is possible to determine which disjoint set ($D_{\iota,1}$. $D_{\iota,2}$ or $D_{\iota,3}$) our sample path belongs by direct application of Theorem 6.3.1, Corollary 6.3.1 and the following Algorithm 6.3.1.

Algorithm 6.3.1 Simulation of an Initial Brownian Bridge Intersection Layer.

1. Simulate layer $\mathcal{I} = \iota$ as per Algorithm 6.2.1, simulate $u \sim \mathrm{U}[0,1]$ and set $k = 0$.
2. While $u \in \left(S_{2k+1}^{D(\iota,1)}, S_{2k}^{D(\iota,1)}\right)$, $k = k + 1$.
3. If $u \leq S_{2k+1}^{D(\iota,1)}$ then set $D_\iota = D_{\iota,1}$.
4. If $u \geq S_{2k}^{D(\iota,1)}$ then with probability 0.5 set $D_\iota = D_{\iota,2}$ else set $D_\iota = D_{\iota,3}$.

6.3.2 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 Beskos et al. [2012], the density of the sample path at the intermediate time point q can be written as follows (where μ_w and σ_w^2 denote the mean and variance of a Brownian bridge as in

Section 2.8),

$$\pi(w) := \mathbb{P}\left(W_q = w \,\middle|\, W_s = x, W_t = y, \hat{m}_{s,t} \in \left[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}\right], \check{m}_{s,t} \in \left[\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}\right]\right) \tag{6.62}$$

$$\propto \rho\left(s, q, t, x, w, y, \ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}\right) \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right). \tag{6.63}$$

A method of simulating from $\pi(w)$ was outlined in Beskos et al. [2012] based on inversion sampling and numerical methods, however, this scheme is formally inexact and given particular parameter values can be computationally extremely inefficient. Clearly for our purposes ensuring exactness is necessary and so we provide a number of alternative schemes which are exact.

In Section 6.3.2.1 we present a method of simulating from (6.63) 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 6.3.2.2 we present an alternate rejection sampling scheme which exploits the known Lipschitz constants of the bounding sequence in (6.63) 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 6.3.2.4 and 6.3.5). Finally, for completeness, in Section 6.3.2.3 we construct a third scheme inspired by the Bessel layer constructions found in Section 6.2.1 and Beskos et al. [2008]. This third scheme provides some insight as to how the different layered Brownian bridge constructions of Section 6.2 and Section 6.3 relate to one another.

6.3.2.1 Bounding Cauchy Sequence Approach

Here we show that it is possible to extend Beskos et al. [2012], and simulate from this density exactly by means of composition sampling (see Section 2.2) and rejection sampling (see Section 2.4). We will begin by considering the upper convergent bounding sequence of ρ ($\forall k \in \mathbb{Z}_{\geq 0}$ we have $\rho \leq S_{2k}^{\rho}$ and $\lim_{k\to\infty} S_{2k}^{\rho} = \rho$). Considering the decomposition of (6.37) into its elementary form in terms of $\bar{\varsigma}$ and $\bar{\varphi}$ (see (6.6) and (6.7) respectively) note that it is composed of $K = 64(k+1)^2$ of these elementary terms.

Recalling that $\varsigma_{s,t}^{\ell,\upsilon}(j; x, y) := \bar{\varsigma}_{s,t}^{\ell,\upsilon}(j; x, y) + \bar{\varsigma}_{s,t}^{-\ell,-\upsilon}(j; -x, -y)$ and $\varphi_{s,t}^{\ell,\upsilon}(j; x, y) := \bar{\varphi}_{s,t}^{\ell,\upsilon}(j; x, y) + \bar{\varphi}_{s,t}^{-\ell,-\upsilon}(j; -x, -y)$ 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 Appendix A for further details). As such, we can find a bound for our unnormalised target density (6.63) as follows ($c_i \in \{0, 1\}$ determines the sign of the density contribution),

$$\begin{aligned}\pi(w) \leq S_{2k}^{\rho} \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right) &= \sum_{i=1}^{K} \left[(-1)^{c_i} \cdot \exp\{a_i + b_i w\} \cdot \mathbb{1}\left(w \in [\ell_i, \upsilon_i]\right) \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right)\right] \\ &= \sum_{i=1}^{K} \Big[\underbrace{(-1)^{c_i} \cdot \exp\left\{a_i + \mu_w b_i + b_i \sigma_w^2/2\right\}}_{:=\omega_i} \cdot \mathbb{1}\left(w \in [\ell_i, \upsilon_i]\right) \cdot \underbrace{\mathrm{N}\left(w; \mu_w + b_i\sigma_w^2, \sigma_w^2\right)}_{:=\mathrm{N}\left(w; \mu_i, \sigma_w^2\right)} \Big].\end{aligned} \tag{6.64}$$

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), for any given parameterisation a large proportion of them will have common mean. We exploit this by partitioning the interval that provides support for the target density (6.63) into sections corresponding to the truncation points (in particular, we consider the following partitioning $\left\{[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}], [\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}], [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]\right\}$ which we denote by $j \in \{1, 2, 3\}$ respectively). 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 partition j and $\omega_{i,j}^{+} := (\omega_{i,j} \vee 0)$ we can find an upper bound by solely considering the mixture formed from the positive weights,

$$\begin{aligned}\pi(w) &\leq \sum_{i=1}^{K} \omega_i \cdot \mathrm{N}\left(w; \mu_i, \sigma_w^2\right) \cdot \mathbb{1}\left(w \in [\ell_i, \upsilon_i]\right) \\ &\qquad \cdot \left[\mathbb{1}\left(w \in [\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}]\right) + \mathbb{1}\left(w \in [\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}]\right) + \mathbb{1}\left(w \in [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]\right)\right] \qquad (6.65) \\ &\leq \sum_{i=1}^{K} \mathrm{N}\left(w; \mu_i, \sigma_w^2\right) \cdot \mathbb{1}\left(w \in [\ell_i, \upsilon_i]\right) \cdot \left[\omega_{i,1}^{+} \cdot \mathbb{1}\left(w \in [\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}]\right)\right. \\ &\qquad \left. + \omega_{i,2}^{+} \cdot \mathbb{1}\left(w \in [\ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}]\right) + \omega_{i,3}^{+} \cdot \mathbb{1}\left(w \in [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]\right)\right] \qquad (6.66) \\ &=: S_{2k}^{\rho,+} \cdot N\left(w; \mu_i, \sigma_w^2\right). \qquad (6.67)\end{aligned}$$

By application of composition sampling (see Section 2.2) we can simulate from $S_{2k}^{\rho,+} \cdot \mathrm{N}(\mu_w, \sigma_w^2)$ by first choosing one of the truncated Normal densities partitioned on the in-

terval $[\mathscr{L}, \mathscr{U}]$ with probability proportional to,

$$\omega_{i,j}^{+} \cdot \left[\Phi\left(\mathscr{U} \,\middle|\, \mu_i, \sigma_w^2\right) - \Phi\left(\mathscr{L} \,\middle|\, \mu_i, \sigma_w^2\right)\right]. \tag{6.68}$$

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

$$P = \frac{\rho(w) \cdot \mathrm{N}\left(w \,\middle|\, \mu_w, \sigma_w^2\right)/Z_T}{M \cdot S_{2k}^{\rho,+}(w) \cdot \mathrm{N}\left(w \,\middle|\, \mu_w, \sigma_w^2\right)/Z_D} = \frac{\rho(w)}{S_{2k}^{\rho,+}(w)} \leq 1. \tag{6.69}$$

Events of probability P can be simulated unbiasedly by retrospective Bernoulli sampling (as per Corollary 2.7.1 and Algorithm 2.7.1), noting that P is a linear transformation of ρ. The complete rejection sampler is presented in Algorithm 6.3.2.

Algorithm 6.3.2 Simulation of Intersection Layer Intermediate Points (Bounded Cauchy Sequence Approach).

---

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

6.3.2.2 Lipschitz Approach

Simulating intermediate points as per Algorithm 6.3.2 is (typically) highly efficient as $S_2^{\rho,+} \cdot \mathrm{N}(\mu_w, \sigma_w^2)$ typically tightly bounds $\pi(w)$ (as noted in Beskos et al. [2012]). If this is not the case (which occurs in a small number of parameter configurations), then sampling from the bounding density with $k > 1$ isn't usually effective as $S_{2k}^{\rho,+}$ is only formed by the positive netted components of S_{2k}^{ρ}. In this section we propose an alternative scheme in which we exploit the known Lipschitz constants of the bounding sequence in (6.63) to

construct a tight bound of the target density.

If the rejection sampling scheme proposed in Section 6.3.2 is not efficient then typically we have that $S_2^\rho \cdot \mathrm{N}(\mu_w, \sigma_w^2)$ does not tightly bound $\pi(w)$. In this case the natural question to ask is at what level the alternating Cauchy sequence approximation (S_{2k}^ρ) of ρ needs to be evaluated such that $S_{2k}^\rho \cdot \mathrm{N}(\mu_w, \sigma_w^2)$ does form a tight bound of $\pi(w)$. To address this we note that in analogous form to Section 6.3.2 it is possible to also find a *lower* bound of the target density,

$$S_{2k+1}^\rho \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right) \leq \pi(w) \leq S_{2k}^\rho \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right). \tag{6.70}$$

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 Cauchy sequence S_k^ρ until such time that it exceeds some user specified threshold,

$$T_Z \leq \frac{Z_{2k+1}^\rho}{Z_{2k}^\rho} := \frac{\int_{\ell_{s,t}^\downarrow}^{\upsilon_{s,t}^\uparrow} S_{2k+1}^\rho \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right) \mathrm{d}w}{\int_{\ell_{s,t}^\downarrow}^{\upsilon_{s,t}^\uparrow} S_{2k}^\rho \cdot \mathrm{N}\left(w; \mu_w, \sigma_w^2\right) \mathrm{d}w}. \tag{6.71}$$

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 6.3.2 and shown in (6.65)),

$$\begin{aligned}\pi(w) \leq \sum_{i=1}^{K\uparrow} \mathrm{N}\left(w; \mu_i, \sigma_w^2\right) \cdot \mathbb{1}\left(w \in [\ell_i, \upsilon_i]\right) \cdot \Big[\omega_{i,1} \cdot \mathbb{1}\left(w \in [\ell_{s,t}^\downarrow, \ell_{s,t}^\uparrow]\right) \\ + \omega_{i,2} \cdot \mathbb{1}\left(w \in [\ell_{s,t}^\uparrow, \upsilon_{s,t}^\downarrow]\right) + \omega_{i,3} \cdot \mathbb{1}\left(w \in [\upsilon_{s,t}^\downarrow, \upsilon_{s,t}^\uparrow]\right)\Big] \\ =: g(x).\end{aligned} \tag{6.72}$$

For any given interval $[q, r]$ (where $q < r$), it is possible to explicitly calculate for each of the contributing Normal densities (for instance, $\mathrm{N}(\mu_i, \sigma_w^2)$) the local Lipschitz constant

(we denote $I := [\mu_i - \sigma_w, \mu_i + \sigma_w] \cap [q, r]$),

$$\begin{aligned}\alpha_i(q,r) &:= \sup_{w\in[q,r]} \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}(w;\mu_i,\sigma_w^2) \\ &= \mathbb{1}\left(I \neq \{\emptyset\}\right)\cdot \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}(\mu_i-\sigma_w;\mu_i,\sigma_w^2) \\ &\quad + \mathbb{1}\left(I = \{\emptyset\}\right)\cdot \max\left\{\frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}(q;\mu_i,\sigma_w^2), \frac{\mathrm{d}}{\mathrm{d}w}\mathrm{N}(r;\mu_i,\sigma_w^2)\right\}. \end{aligned} \tag{6.73}$$

As such, it is possible to find for the bounding density ($g(w)$ in (6.72)) the local Lipschitz constant for the interval $[q, r]$ (where α is set to zero when considering an interval of zero length),

$$\begin{aligned}\sup_{u,v\in[q,r]} \frac{g(u)-g(v)}{u-v} &\leq \sum_{j=1}^{K\uparrow}\Big[\left|\omega_{j,1}\right|\cdot\alpha_j\left(q\vee\ell_{s,t}^{\downarrow}, r\wedge\ell_{s,t}^{\uparrow}\right) + \left|\omega_{j,2}\right|\cdot\alpha_j\left(q\vee\ell_{s,t}^{\uparrow}, r\wedge\upsilon_{s,t}^{\downarrow}\right) \\ &\qquad + \left|\omega_{j,3}\right|\cdot\alpha_j\left(q\vee\upsilon_{s,t}^{\downarrow}, r\wedge\upsilon_{s,t}^{\uparrow}\right)\Big] \\ &=: \beta(q,r), \end{aligned} \tag{6.74}$$

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 $y = g(q)+\beta t$ and the line $y = g(r)+\beta(r-q)-\beta t$ both bound the target density on the interval $[q, r]$ and intersect at $t = [g(r) - g(q) + \beta(r - q)]/2\beta \in [q, r]$),

$$\begin{aligned}\sup_{w\in[q,r]} g(w) &\leq g(r) + \beta(q,r)\cdot t \\ &= \frac{g(r)+g(q)}{2} + \beta(q,r)\cdot\frac{r-q}{2} =: M(q,r). \end{aligned} \tag{6.75}$$

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

$$g(w) \leq m(w) := \sum_{i=1}^{N} \mathbb{1}(w \in [\ell_{s,t}^{\downarrow} + (i-1)\Delta, \ell_{s,t}^{\downarrow} + i\Delta])\cdot M\left(\ell_{s,t}^{\downarrow} + (i-1)\Delta, \ell_{s,t}^{\downarrow} + i\Delta\right). \tag{6.76}$$

As in (6.71) 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,

$$T_M \leq \frac{Z^{\rho}_{2k+1}}{Z^N_M} := \frac{Z^{\rho}_{2k+1}}{\sum_{i=1}^{N} \Delta \cdot M\left(\ell^{\downarrow}_{s,t} + (i-1)\Delta, \ell^{\downarrow}_{s,t} + i\Delta\right)}. \tag{6.77}$$

We present the synthesis of the above argument in Algorithm 6.3.3. Clearly the acceptance rate of Algorithm 6.3.3 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 6.3.2, as given sufficient computation an arbitrarily tight bound of the target density can be found with which to conduct rejection sampling. In Figure 6.3.1 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 6.3.3 a small mesh of size 20 was sufficient to find a tight upper bound of the target density.

Algorithm 6.3.3 Simulation of Intersection Layer Intermediate Points (Lipschitz Approach).

---

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

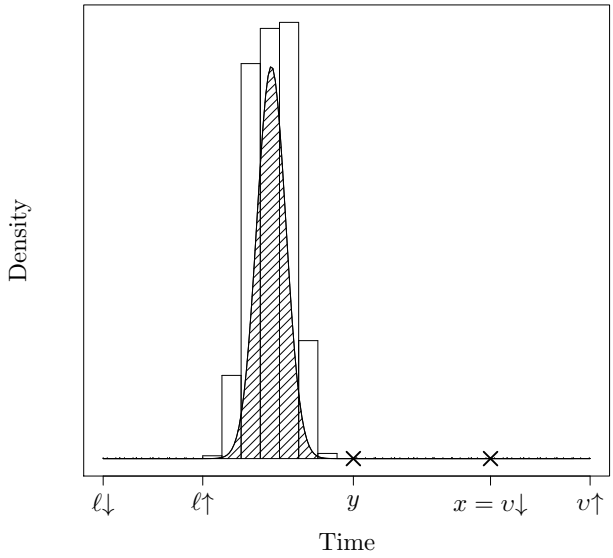


Figure 6.3.1: Density of intersection layer intermediate point overlaid with piecewise constant bound calculated using a mesh of size 20 over the interval $[\ell\downarrow, \upsilon\uparrow]$ and the corresponding local Lipschitz constants.

6.3.2.3 Bessel Approach

An alternative scheme to simulate a single intermediate point from (6.62) is to apply an analogous decomposition of the law of the sample path as was constructed in the Bessel approach for layered Brownian bridge outlined in Section 6.2.1. Recall in Section 6.2.1 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 6.2.1, 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 ι^{th} layer and

then only accept the proposal sample path if the sample path maximum (else minimum) is contained between the $(\iota-1)^{\text{th}}$ and ι^{th} 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 ι^{th} (else $(\iota-1)^{\text{th}}$) layer and then only accept the proposal sample path if the sample path maximum (else minimum) is contained within the $(\iota-1)^{\text{th}}$ (else $(\iota-1)^{\text{th}}$) Bessel layer. The synthesis of the above argument which is based on Section 6.2.1 can be found in Algorithm 6.3.4.

Although given particular parameter values in (6.62) the Bessel approach can computationally outperform the Cauchy sequence approach or Lipschitz approached described in Section 6.3.2.1 and Section 6.3.2.2 respectively, as we will discuss in Section 6.3.2.4 we advocate a mixture of these two approaches instead. The Bessel approach can be particularly inefficient whenever a large intersection layer is proposed, however, we have included this here for completeness and for the insight it offers into the similarities and constructions of the different layered Brownian bridge approaches discussed in Sections 6.2 and 6.3.

6.3.2.4 Implementational Considerations – Recommended Approach

In Sections 6.3.2.1, 6.3.2.2 and 6.3.2.3 we outlined three separate approaches and algorithms for simulating from the density of a conditional Brownian bridge at some intermediate time $q \in (s,t)$ (6.63). 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 of these algorithms without introducing any bias. As typically Algorithm 6.3.2 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 T_N). If after the first T_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 6.3.3 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 6.3.2. This particular combination of algorithms is advocated as Algorithm 6.3.4 can be inefficient whenever a large intersection layer is proposed.

Algorithm 6.3.4 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 2.8.4,
   (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_τ as per Algorithm 2.8.3.
4. While $u_2 \in \left(\prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}\big(X_\tau, c\downarrow\big), \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}\big(X_\tau, c\downarrow\big)\right)$, $j = j+1$,
   (a) If $u_2 \leq \prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}\big(X_\tau, c\downarrow\big)$ then reject sample path and return to Step 1.
5. While $u_2 \in \left(\prod_{i=1}^{\kappa+2} S^{\delta}_{2j+1}\big(X_\tau, c\uparrow\big), \prod_{i=1}^{\kappa+2} S^{\delta}_{2j}\big(X_\tau, c\uparrow\big)\right)$, $k = k+1$,
   (a) If $u_2 \geq \prod_{i=1}^{\kappa+2} S^{\delta}_{2k}\big(X_\tau, c\uparrow\big)$ then reject sample path and return to Step 1.
6. Discard Auxiliary Information.

6.3.3 Dissecting an Intersection Layer

Upon simulating intermediate points of a Brownian bridge sample path conditional on an intersection layer (for instance in Section 6.3.2), simulating further intermediate points in a sub-interval between any two existing consecutive points is more complicated as there is a dependancy 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 min-

imum 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 have,

$$D_*|(W_q = w) = \left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}. \tag{6.78}$$

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 $B := D_*|(W_q = w) = \uplus_{i=1}^{9} B_i$) as illustrated in Figure 6.3.2. For instance, our sample path may lie in B_6, which more formally has the form[2],

$$\begin{aligned} B_6 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\uparrow *}_{s,t}, (x \wedge w)\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \\ \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[(w \vee y), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right). \end{aligned} \tag{6.79}$$

This notion can be extended to the case where we have multiple intermediate points

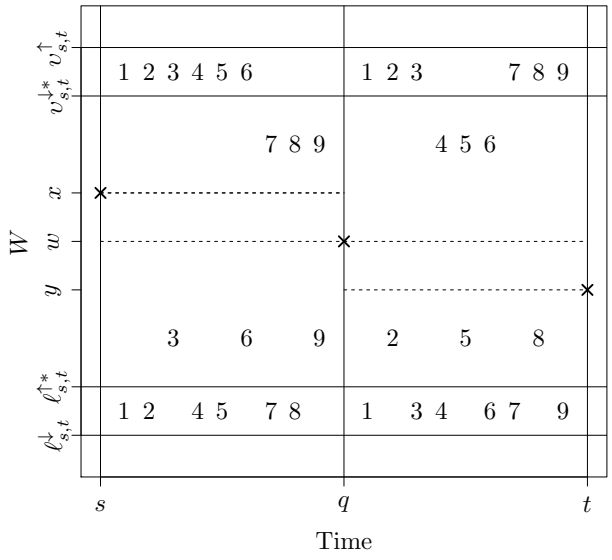


Figure 6.3.2: Illustration of 9 possible (disjoint) bisections.

[2]For a full list of the sets $B1 - B9$ we refer the interested reader to Appendix B.

($\mathcal{W} := \{W_{q_1} = w_1, \ldots, W_{q_n} = w_n\}$), 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 subinterval. 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 $B_1^n, \ldots, B_b^n$. We can simulate which of these dissections our sample path lies in by application of the following results and Algorithm 6.3.5.

Theorem 6.3.2 (Intersection Layer Dissection)**.** *The probability a Brownian bridge sample path is in B_i^n 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_i^n),*

$$\begin{aligned} p_{B_i^n} &:= \mathbb{P}\left(B_i^n \,\middle|\, \hat{m}_{s,t} \in [\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}], \check{m}_{s,t} \in [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}], W_s = x, W_t = y, \mathcal{W}\right) \\ &= \frac{{}^{(n)}\beta_{s,t,x,y}^{\mathcal{L}(i),\mathcal{U}(i)}(q_{1:n}, \mathcal{W})}{{}^{(n)}\rho_{s,t,x,y}^{\ell\downarrow,\ell\uparrow,\upsilon\downarrow,\upsilon\uparrow}(q_{1:n}, \mathcal{W})}. \end{aligned} \tag{6.80}$$

Proof. Follows directly by Bayes rule, Theorems 6.1.4 and 6.1.5. □

Remark 6.3.1 (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 the following (denoting $\ell_{s,q}^{\downarrow,i}$, $\ell_{s,q}^{\uparrow,i}$ $\upsilon_{s,q}^{\downarrow,i}, \upsilon_{s,q}^{\uparrow,i}$ and $\ell_{q,t}^{\downarrow,i}$, $\ell_{q,t}^{\uparrow,i}$ $\upsilon_{q,t}^{\downarrow,i}, \upsilon_{q,t}^{\uparrow,i}$ as the bounds for B_i in the interval $[s,q]$ and $[q,t]$ respectively),*

$$p_{B_i} = \frac{\beta\left(s, q, x, w, \ell_{s,q}^{\downarrow,i}, \ell_{s,q}^{\uparrow,i}, \upsilon_{s,q}^{\downarrow,i}, \upsilon_{s,q}^{\uparrow,i}\right) \cdot \beta\left(q, t, w, y, \ell_{q,t}^{\downarrow,i}, \ell_{q,t}^{\uparrow,i}, \upsilon_{q,t}^{\downarrow,i}, \upsilon_{q,t}^{\uparrow,i}\right)}{\rho\left(s, q, t, x, w, y, \ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow *}, \upsilon_{s,t}^{\downarrow *}, \upsilon_{s,t}^{\uparrow}\right)}. \tag{6.81}$$

Corollary 6.3.2. *Events of probability $p_{B_i^n}$ can be represented as the limit as $k \to \infty$ of the following alternating Cauchy sequence,*

$$S_k^{B(n,i)} := \frac{S_k^{\beta(n)}(s, t, x, y, q_{1:n}, \mathcal{W}, \mathcal{L}(i), \mathcal{U}(i))}{S_{k+1}^{\rho(n)}\left(s, t, x, y, q_{1:n}, \mathcal{W}, \ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}, \upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}\right)}. \tag{6.82}$$

Proof. Follows by Corollary 2.7.1 in the same manner as in the proof of Corollary 6.1.4. □

As a consequence of Corollary 6.3.2, events of probability $p_{B_i^n}$ can be simulated unbiasedly by retrospective Bernoulli sampling (as per Algorithm 2.7.1 in Section 2.7). Unbiased simulation of the dissection the sample path lies in can be conducted by inversion

sampling (see Section 2.1) and an alternating Cauchy sequence representation of the CDF of B (see (6.83)) as detailed in Algorithm 6.3.5.

$$C_k^{B(n,j)} := \sum_{i=1}^{j} S_k^{B(n,i)}. \tag{6.83}$$

Algorithm 6.3.5 Dissecting an Intersection Layer.

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

6.3.4 Refining an Intersection Layer

Suppose we have already simulated layers for the maximum and minimum of our proposal Brownian bridge sample path ($\hat{m}_{s,t} \in [\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}]$ and $\check{m}_{s,t} \in [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]$), but we require more *refined* layer information (i.e. we want a set of narrower layers $\left|\ell_{s,t}^{\uparrow^*} - \ell_{s,t}^{\downarrow^*}\right| \leq \left|\ell_{s,t}^{\uparrow} - \ell_{s,t}^{\downarrow}\right|$ or $\left|\upsilon_{s,t}^{\uparrow^*} - \upsilon_{s,t}^{\downarrow^*}\right| \leq \left|\upsilon_{s,t}^{\uparrow} - \upsilon_{s,t}^{\downarrow}\right|$). This can be achieved by first noting that the sample path falls in one of the following 4 possible (disjoint) intersection layer refinements (where $R := \uplus_{i=1}^{4} R_i$ as illustrated in Figure 6.3.3, $\ell_{s,t}^{\updownarrow} \in [\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\uparrow}]$ and $\upsilon_{s,t}^{\updownarrow} \in [\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\uparrow}]$),

$$\begin{aligned}
R_1 &= \left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\updownarrow}\right]\right\} \cap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon_{s,t}^{\updownarrow}, \upsilon_{s,t}^{\uparrow}\right]\right\}, \\
R_2 &= \left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell_{s,t}^{\updownarrow}, \ell_{s,t}^{\uparrow}\right]\right\} \cap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon_{s,t}^{\updownarrow}, \upsilon_{s,t}^{\uparrow}\right]\right\}, \\
R_3 &= \left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell_{s,t}^{\downarrow}, \ell_{s,t}^{\updownarrow}\right]\right\} \cap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\updownarrow}\right]\right\}, \\
R_4 &= \left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell_{s,t}^{\updownarrow}, \ell_{s,t}^{\uparrow}\right]\right\} \cap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon_{s,t}^{\downarrow}, \upsilon_{s,t}^{\updownarrow}\right]\right\}.
\end{aligned}$$

In a similar fashion to Section 6.3.3 we can simulate unbiasedly which of the intersection layer refinements our sample path lies in by application of the following established results and Algorithm 6.3.6 (where we denote by $\ell_{s,t}^{\downarrow,i}, \ell_{s,t}^{\uparrow,i}, \upsilon_{s,t}^{\downarrow,i}, \upsilon_{s,t}^{\uparrow,i}$ with a superscript $i \in \{1,2,3,4\}$ as the corresponding parameter selections from $R_1 - R_4$).

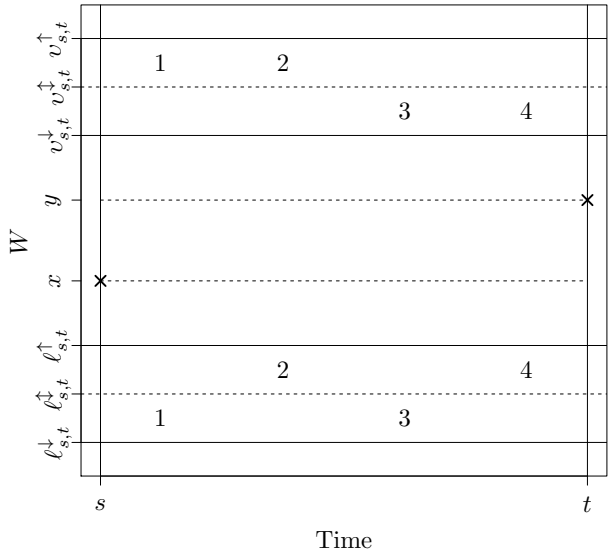


Figure 6.3.3: Illustration of 4 possible refinements.

Theorem 6.3.3 (Intersection Layer Refinement [Beskos et al., 2012, Section 5.3])**.** *The probability a Brownian bridge sample path is in R_i conditional on R is as follows,*

$$\begin{aligned} p_{R_i} &:= \mathbb{P}\left(R_i \,\middle|\, \hat{m}_{s,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}\right], \check{m}_{s,t} \in \left[\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\right], W_s = x, W_t = y\right) \\ &= \frac{\beta\left(s,t,x,y,\ell^{\downarrow,i}_{s,t},\ell^{\uparrow,i}_{s,t},\upsilon^{\downarrow,i}_{s,t},\upsilon^{\uparrow,i}_{s,t}\right)}{\beta\left(s,t,x,y,\ell^{\downarrow}_{s,t},\ell^{\uparrow}_{s,t},\upsilon^{\downarrow}_{s,t},\upsilon^{\uparrow}_{s,t}\right)}. \end{aligned} \tag{6.84}$$

Proof. Follows directly by Bayes rule and Theorem 6.1.5. □

Corollary 6.3.3 ([Beskos et al., 2012, Section 5.3])**.** *Events of probability p_{R_i} can be represented as the limit as $k \to \infty$ of the following alternating Cauchy sequence,*

$$S_k^{R(i)} := \frac{S_k^{\beta}\left(s,t,x,y,\ell^{\downarrow,i}_{s,t},\ell^{\uparrow,i}_{s,t},\upsilon^{\downarrow,i}_{s,t},\upsilon^{\uparrow,i}_{s,t}\right)}{S_{k+1}^{\beta}\left(s,t,x,y,\ell^{\downarrow}_{s,t},\ell^{\uparrow}_{s,t},\upsilon^{\downarrow}_{s,t},\upsilon^{\uparrow}_{s,t}\right)}. \tag{6.85}$$

Proof. Follows by Corollary 2.7.1 in the same manner as in the proof of Corollary 6.1.4. □

As a consequence of Corollary 6.3.3, events of probability p_{R_i} can be simulated by retrospective Bernoulli sampling (as per Algorithm 2.7.1 in Section 2.7). As such, unbiased simulation of the refinement the sample path lies in can be conducted by inversion sampling (see Section 2.1) and an alternating Cauchy sequence representation of the CDF of R (see (6.86)) in the same manner as Section 6.3.3 and as detailed in Algorithm 6.3.6

$$C_k^{R(j)} := \sum_{i=1}^{j} S_k^{R(i)}. \tag{6.86}$$

Algorithm 6.3.6 Refining an Intersection Layer [Beskos et al., 2012].

1. Simulate $u \sim \mathrm{U}[0,1]$ and set $j = 1$ and $k = 0$.
2. While $u \in \left(C_{2k+1}^{R(j)}, C_{2k}^{R(j)}\right)$, $k = k + 1$.
3. If $u \leq S_{2k+1}^{R(j)}$ set layer $R = R_j$ else set $j = j + 1$ and return to Step 2.

6.3.5 Simulating Layered Brownian Bridges

The *Intersection Layer Approach* for constructing a layered Brownian bridge is a direct application of the algorithms of Sections 6.3.1, 6.3.2 and 6.3.3. In particular, we simulate initial intersection layer information for the sample path (Algorithm 5.1.4 Step 2) by application of Algorithm 6.3.1. In Algorithm 5.1.4 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 6.3.2, 6.3.3, 6.3.4 or some mixture of these algorithms to simulate the intermediate point (as discussed in Section 6.3.2 and in particular Section 6.3.2.4) and Algorithm 6.3.5 to bisect the interval.

We present the iterative Algorithm 5.1.4 Step 4 in Algorithm 6.3.7 which can be additionally used to conduct Algorithm 5.1.4 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 $\left(\mathcal{S}_{a,b} := \left\{a, b, X_a, X_b, \ell_{a,b}^{\downarrow}, \ell_{a,b}^{\uparrow}, \upsilon_{a,b}^{\downarrow}, \upsilon_{a,b}^{\uparrow}\right\}\right)$.

Algorithm 6.3.7 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 6.3.2 or 6.3.3 or 6.3.4.
   (c) Bisect interval as per Algorithm 6.3.5 to find new intersection layers $\mathcal{S}_{a,q}$ and $\mathcal{S}_{q,b}$.
   (d) Set $\mathcal{S} = \mathcal{S} \cup \left\{\mathcal{S}_{a,q}, \mathcal{S}_{q,b}\right\} \setminus \mathcal{S}_{a,b}$.

It should be noted that further refinements to Algorithm 6.3.7 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 5.1.4), then refining the intersection layers as outlined in Section 6.3.4 and detailed in Algorithm 6.3.6 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 Chapter 8.

7 Particle Filtering for Diffusions and Jump Diffusions

"Alive without breath,
As cold as death;
Never thirsty, ever drinking,
All in mail never clinking."

— J.R.R. Tolkien, *The Hobbit*

In this chapter we present methodology for particle filters with jump diffusion latent transition density. In particular, referring to the illustration of a HMM in Figure 3.1.1 (which we reproduce here for convenience as Figure 7.0.1) and recalling from Section 3.1 that $\{X_t\}_{t\geq 0}$ is some latent (hidden or unobserved) process of interest which is observed indirectly through an observation process $\{Y_t\}_{t\geq 1}$, we are interested in the filtering problem, in which we want to make a probabilistic interpretation of the state of the latent process at a point in time, using the information obtained from the observation process up to that point in time.

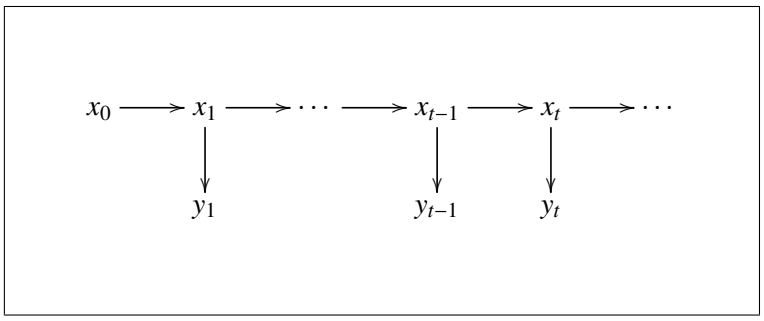


Figure 7.0.1: Directed acyclic graph representing the latent and observation processes of a Hidden Markov Model.

In the particular setting we consider in this chapter the latent process initial SSD (μ_θ, see HMM Property 1) and observation process SSD (g_θ, see HMM Property 3) are as found in Section 3.1 and Chapter 3, however the latent process SSD (f_θ, see HMM Property 2) is governed by a jump diffusion. Recalling from Section 4.1.4 that we found a (typically intractable) representation for the transition density of a jump diffusion, then in this setting the transition density of the latent process of interest is as follows,

$$\begin{aligned} p_{t-s}(x,y) &:= \mathbb{P}\left(X_t \in \mathrm{d}y \,|\, X_s = x\right) / \mathrm{d}y \\ &= w_{t-s}(x,y) \cdot \mathbb{E}_{\mathbb{W}^{x,y}_{s,t}}\left[\frac{\mathrm{d}\mathbb{Q}^x_{s,t}}{\mathrm{d}\mathbb{W}^x_{s,t}}(X)\right], \end{aligned} \tag{7.1}$$

where $\mathbb{Q}^x_{s,t}$ and $\mathbb{W}^x_{s,t}$ are the measures induced by the target and driftless jump diffusions ((4.29) and (4.30) respectively) initialised at $X_s = x$ over the interval $[s,t]$, $\mathbb{W}^{x,y}_{s,t}$ is the measure $\mathbb{W}^x_{s,t}$ with the additional restriction that $X_t = y$, and finally $w_{t-s}(x,y)$ is the transition density of the driftless jump diffusion (4.30)).

Building upon the exact algorithm methodology developed in Chapters 5 and 6, an obvious approach to the filtering problem is to employ the Sequential Importance Sampling / Resampling algorithm (Algorithm 3.4.1) of Section 3.4, with a prior marginal importance function (see Section 3.3.2), and simply propagate each particle between observation points using an appropriate exact algorithm (for instance, the Bounded Jump Exact Algorithm (BJEA; Algorithm 5.3.1), Unbounded Jump Exact Algorithm (UJEA; Algorithm 5.3.2) or Adaptive Unbounded Jump Exact Algorithm (AUJEA; Algorithm 5.3.3)). Indeed this is precisely what we present as the Exact Propagation Particle Filter (EPPF) in Algorithm 7.2.1 of Section 7.2 (following the naming convention of Fearnhead et al. [2008]).

Simulating exactly from the transition density of a jump diffusion in (7.1) using an exact algorithm and for application within a particle filter is a novel approach extending upon Fearnhead et al. [2008] (which considers particle filtering for partially observed diffusions). The methodology we present in this chapter has the distinct advantage over competing methodology employing an approximate method to simulate from the transition density (as detailed in Section 4.2) as it avoids introducing any implicit discretisation error to latent state estimates.

Extending upon the EPPF we introduce in Section 7.2, and employing the same approach as in Fearnhead et al. [2008], we can construct a more sophisticated particle filter by considering an importance sampling analog of the exact algorithms in Chapter 5. In particular, rather than for each particle proposing a sample path simulated from the dominating measure and accepting or rejecting it, we can instead assign it an importance weight. In Section 7.1 (which is a summary of Fearnhead et al. [2008]) we formalise this approach before returning in Section 7.2 to present alternative algorithms to the EPPF. As an aside we note at this stage that, unlike the methodology presented in Fearnhead et al. [2008] for particle filters with diffusion transition density, jump diffusions lose the necessary tractability which enables the direct application of an auxiliary particle filter based approach (see Section 3.6) without the inclusion of onerous additional conditions.

7.1 Poisson Estimators

In this section we restrict our attention to the (typically intractable) expectations encountered in the introductory remarks of Chapter 7,

$$\mathbb{E}_{\mathbb{W}_{s,t}^{x,y}}\left[\frac{\mathrm{d}\mathbb{Q}_{s,t}^{x}}{\mathrm{d}\mathbb{W}_{s,t}^{x}}(X)\right], \tag{7.2}$$

considering the specific case outlined in (4.49) of Section 4.1.4 (where $\mathbb{Q}_{s,t}^{x}$ is the measure induced by a diffusion, $\mathbb{W}_{s,t}^{x}$ is Wiener measure with start point $(s, W_s = x)$ and $\mathbb{W}_{s,t}^{x,y}$ is the measure induced by a Brownian bridge with start point $(s, W_s = x)$ and terminal point $(t, W_t = y)$), to outline general methodology for simulating an unbiased estimator for functions of the following form,

$$\psi(X) := \frac{\mathrm{d}\mathbb{Q}_{s,t}^{x}}{\mathrm{d}\mathbb{W}_{s,t}^{x}}(X) = \exp\left\{-\int_{s}^{t}\phi(X_u)\,\mathrm{d}u\right\}, \quad \text{where } X \sim \mathbb{W}_{s,t}^{x,y}. \tag{7.3}$$

Recall that throughout Chapter 5 that similar expectations and functions frequently arise when evaluating whether to accept or reject a proposal sample path as a sample path from some target measure (i.e. assign a simulated sample path with weight 1 or weight 0). As such, much of the methodology we detail in this section has a common link with the methodology in Chapter 5.

We require an estimator for $\psi(X)$ (which we denote by $\hat{\psi}(X)$) as it is not possible to simulate an entire (infinite dimensional) sample path $X \sim \mathbb{W}_{s,t}^{x,y}$ in order to evaluate $\psi(X)$

in (7.3). In particular, the estimator $\hat{\psi}(X)$ should be constructed such that it has finite variance and can be evaluated using a finite dimensional realisation of a sample path (X^{fin}). Clearly having an unbiased estimator is desirable (particularly in the particle filtering setting which we consider in this chapter) and as a consequence of the exponential form of (7.3) it is also desirable that the estimator is positive (again, in the particle filtering setting this is particularly important as the natural interpretation of this estimate is as an importance weight). Given more than one estimator which satisfies these properties, then a natural comparison to make in order to select between estimators is to consider the comparative variances of the estimators.

One naïve approach to construct an estimator for $\psi(X)$ (as proposed by Nicolau [2002]), is to simply use a Riemannian sum to approximate the integral in (7.3). In particular, partitioning the interval $[s, t]$ into a fine mesh (for instance, into M sub-intervals of length $\Delta = (t - s)/M$) and simulating a Brownian bridge sample path at the mesh points (as per Algorithm 2.8.2), then we arrive at the following estimator,

$$\hat{\psi}_M(X) := \exp\left\{-\Delta \sum_{i=0}^{M-1} \phi\left(X_{s+i\Delta}\right) \mathrm{d}u\right\}. \tag{7.4}$$

Unfortunately, this estimator is biased as a result of both the interval discretisation and the exponential function. An improvement to this estimator can be made by noting that instead of simulating the sample path on a deterministic mesh, an unbiased estimate of the path integral can be obtained by simulating a uniform random variable (denoting by $\mathbb{U}$ as the distribution of $\xi \sim \mathrm{U}[s, t]$),

$$\mathbb{E}_{\mathbb{U}}\left[\phi(X_\xi) \,\middle|\, X\right] = \frac{1}{t-s} \int_s^t \phi(X_u) \,\mathrm{d}u. \tag{7.5}$$

As a consequence of the exponential function, such a construction for an estimator is still biased (i.e. $\mathbb{E}_{\mathbb{U}}[\exp\{-\phi(X_\xi)\} \,|\, X] \neq \exp\{\frac{1}{t-s} \int_s^t \phi(X_u) \,\mathrm{d}u\}$), however it offers an important insight in how to construct an unbiased estimator.

Poisson Estimators are a class of unbiased estimators proposed by Beskos et al. [2006b], Fearnhead et al. [2008] and Fearnhead et al. [2010]. In Section 7.1.1 we introduce the (Vanilla) Poisson Estimator proposed by Beskos et al. [2006b], which although unbiased is not necessarily positive and does not necessarily have finite variance. In Section 7.1.2 we explore the Generalised Poisson Estimator introduced by Fearnhead et al. [2008],

which is an estimator which in addition to being unbiased can be constructed to ensure positivity and finite variance. As in this thesis we restrict our attention to expectations of the form (7.2) and constructing unbiased estimators for (7.3), we omit the Wald Poisson Estimator proposed by Fearnhead et al. [2010] which addresses the construction of positive unbiased estimators with finite variance in broader settings.

7.1.1 Vanilla Poisson Estimator

The (Vanilla) Poisson Estimator (VPE) introduced by Beskos et al. [2006b] is constructed by conducting a Taylor series expansion of the exponential function in (7.3), noting that the resulting expression takes the form of a Poisson distribution. In particular, letting $c \in \mathbb{R}$ and $\lambda \in \mathbb{R}_+$ be user specified constants and denoting by $\mathbb{K}$ as the law of $\kappa \sim \text{Poi}(\lambda(t-s))$, we have,

$$\begin{aligned}
\psi(X) &= \exp\left\{-\int_s^t \phi(X_u)\,\mathrm{d}u\right\} = e^{-c(t-s)} \cdot \exp\left\{\lambda(t-s)\int_s^t \frac{c-\phi(X_u)}{\lambda(t-s)}\,\mathrm{d}u\right\} \\
&= e^{-c(t-s)} \sum_{j=0}^{\infty}\left[\frac{1}{j!}\left(\lambda(t-s)\int_s^t \frac{c-\phi(X_u)}{\lambda(t-s)}\,\mathrm{d}u\right)^j\right] \\
&= e^{(\lambda-c)\cdot(t-s)} \sum_{j=0}^{\infty}\left[\frac{\exp\{-\lambda(t-s)\}(\lambda(t-s))^j}{j!}\left(\int_s^t \frac{c-\phi(X_u)}{\lambda(t-s)}\,\mathrm{d}u\right)^j\right] \\
&= e^{(\lambda-c)\cdot(t-s)} \cdot \mathbb{E}_{\mathbb{K}}\left[\left(\int_s^t \frac{c-\phi(X_u)}{\lambda(t-s)}\,\mathrm{d}u\right)^{\kappa}\middle| X\right]. \qquad (7.6)
\end{aligned}$$

Now, recalling from (7.5) that an unbiased estimate of a path integral can be obtained by simulating uniform random variables, then we have (denoting by $\mathbb{U}\kappa$ as the distribution of $(\xi_1,\ldots,\xi_\kappa) \overset{\text{iid}}{\sim} \text{U}[s,t]$, and noting that the expectation of the product of κ independent estimates of a random variable is the expectation of the estimate of the random variable to the κth power),

$$\psi(X) = e^{(\lambda-c)\cdot(t-s)} \cdot \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\prod_{i=1}^{\kappa}\frac{c-\phi(X_{\xi_i})}{\lambda}\middle| X\right]. \qquad (7.7)$$

As such, as suggested by (7.7) and detailed in Algorithm 7.1.1, we can construct an unbiased estimate of $\psi(X)$ by simply uniformly scattering $\kappa \sim \text{Poi}(\lambda(t-s))$ points on the

interval $[s,t]$, evaluating a sample path $X \sim \mathbb{W}_{s,t}^{x,y}$ at these points and setting,

$$\hat{\psi}_{\mathrm{VPE}}(X) := e^{(\lambda-c)\cdot(t-s)} \prod_{i=1}^{\kappa} \frac{c-\phi(X_{\xi_i})}{\lambda}. \tag{7.8}$$

Algorithm 7.1.1 Vanilla Poisson Estimator (VPE) [Beskos et al., 2006b].

1. Choose c and $\lambda > 0$.
2. Simulate $\kappa \sim \mathrm{Poi}(\lambda(t-s))$ and skeleton times $\xi_1, \ldots, \xi_\kappa \overset{\mathrm{iid}}{\sim} \mathrm{U}[s,t]$.
3. Simulate Brownian bridge sample path at skeleton times $X_{\xi_1}, \ldots, X_{\xi_\kappa} \sim \mathbb{W}_{s,t}^{x,y}$.
4. Set $\hat{\psi}_{\mathrm{VPE}}(X) = e^{(\lambda-c)\cdot(t-s)} \prod_{i=1}^{\kappa} [c-\phi(X_{\xi_i})]/\lambda$.

The second moment of the VPE can be derived as follows,

$$\begin{aligned}
\mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\left(e^{(\lambda-c)\cdot(t-s)}\prod_{i=1}^{\kappa}\left(\frac{c-\phi(X_{\xi_i})}{\lambda}\right)\right)^2 \middle| X\right] &= \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[e^{2(\lambda-c)\cdot(t-s)}\prod_{i=1}^{\kappa}\frac{[c-\phi(X_{\xi_i})]^2}{\lambda^2} \middle| X\right] \\
&= e^{2(\lambda-c)(t-s)} \cdot \mathbb{E}_{\mathbb{K}}\left[\left(\int_s^t \frac{[c-\phi(X_u)]^2}{\lambda^2(t-s)}\,\mathrm{d}u\right)^{\kappa} \middle| X\right] \\
&= e^{2(\lambda-c)\cdot(t-s)} \sum_{j=0}^{\infty}\left[\frac{\exp\{-\lambda(t-s)\}(\lambda(t-s))^j}{j!}\left(\int_s^t \frac{[c-\phi(X_u)]^2}{\lambda^2(t-s)}\,\mathrm{d}u\right)^j\right] \\
&= e^{(\lambda-2c)\cdot(t-s)} \sum_{j=0}^{\infty}\left[\frac{1}{j!}\left(\int_s^t \frac{[c-\phi(X_u)]^2}{\lambda}\,\mathrm{d}u\right)^j\right] \\
&= e^{(\lambda-2c)\cdot(t-s)} \exp\left\{\frac{1}{\lambda}\int_s^t [c-\phi(X_u)]^2\,\mathrm{d}u\right\}.
\end{aligned} \tag{7.9}$$

Although the VPE is unbiased (which is desirable), it is clear that the positivity of the estimator (see (7.7)) depends on the particular selection of c and λ. Furthermore, as a consequence of (7.9), the VPE need not necessarily have finite variance.

It was noted in Beskos et al. [2006b] that in the particular setting where ϕ is almost surely bounded (i.e $\forall u \in [s,t]$, $\phi(X_u) \in [L,U]$), then c and λ can be chosen such that the VPE is positive and has finite variance (which we call the *Bounded Vanilla Poisson Estimator*

(BVPE)). In particular, selecting $c := U$ we ensure positivity as (with reference to (7.7)) we have that $\inf_{u\in[s,t]} U - \phi(X_u) \geq 0$. Now, selecting $\lambda := U - L \geq \sup_{u\in[s,t]}(U - \phi(X_u))$ and reconsidering the second moment (7.9) we have,

$$\begin{aligned}\mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\left(e^{(\lambda-c)\cdot(t-s)}\prod_{i=1}^{\kappa}\left(\frac{c-\phi(X_{\xi_i})}{\lambda}\right)\right)^2\middle|X\right] &:= \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\left(e^{-L(t-s)}\prod_{i=1}^{\kappa}\left(\frac{U-\phi(X_{\xi_i})}{U-L}\right)\right)^2\middle|X\right] \\ &= e^{-(L+U)\cdot(t-s)}\exp\left\{\frac{1}{U-L}\int_s^t [U-\phi(X_u)]^2\,\mathrm{d}u\right\} \\ &\leq \exp\{-2L(t-s)\} < \infty, \end{aligned} \tag{7.10}$$

and so we can conclude that the BVPE has finite variance. Returning to (7.8) we find with this particular selection ($c := U$ and $\lambda := U - L$) the unbiased estimate of $\psi(X)$ obtained by the Poisson estimator is,

$$\hat{\psi}_{\mathrm{BVPE}}(X) := e^{-L(t-s)}\prod_{i=1}^{\kappa}\frac{U-\phi(X_{\xi_i})}{U-L}. \tag{7.11}$$

The BVPE has a strong connection with the construction of the Bounded Exact Algorithm (BEA) detailed in Section 5.1.1 in the setting where the target measure is that induced by a diffusion which, in addition to having a start point ($X_s = x$), is further conditioned to have some specified end point ($X_t = y$) (i.e. we want to simulate a diffusion bridge as per Algorithm 5.2.1 in Section 5.2 in the setting where ϕ is almost surely bounded). Indeed, the BVPE can be thought of as an importance sampling analog in which rather than accepting or rejecting a proposal sample path, it is instead assigned an importance weight.

Note that as a consequence of Condition 5 we have by assumption that $\exists\Phi > -\infty$ such that $\forall_u \in [s,t]$, $\phi(X_u) \geq \Phi$. However, as will become apparent later in this chapter, unlike the exact algorithms presented in Section 5 finding a tight lower bound (which can be non-trivial) is not critical, and instead it suffices to identify that there exists a lower bound.

7.1.2 Generalised Poisson Estimator

The Generalised Poisson Estimator (GPE) introduced in [Fearnhead et al., 2008, Chap. 4] is an extension to the Vanilla Poisson Estimator discussed in Section 7.1.1, and is an estimator which in addition to being unbiased is also positive and has finite variance.

As we noted in Result 4 of Section 1.3 and exploited in Section 5.1.1, $\phi(X)$ is bounded on compact sets. As such, if we can identify an interval in which the sample path X is almost surely constrained, we can find (conditionally) an upper and lower bound for $\phi(X)$ (which we denote by $U_X \in \mathbb{R}$ and $L_X \in \mathbb{R}$ respectively). Following the approach taken in Section 5.1.1 we can partition the path space of $\mathbb{W}_{s,t}^{x,y}$ into disjoint layers and simulate the layer to which our proposal sample path belongs (see Principle 1, denoting as before $R := R(X) \sim \mathcal{R}$ as the simulated layer). In particular, returning to the (typically intractable) expectations encountered in the introductory remarks of Chapter 7 we have,

$$\mathbb{E}_{\mathbb{W}_{s,t}^{x,y}}\left[\frac{\mathrm{d}\mathbb{Q}_{s,t}^{x}}{\mathrm{d}\mathbb{W}_{s,t}^{x}}(X)\right] = \mathbb{E}_{\mathcal{R}}\mathbb{E}_{\mathbb{W}_{s,t}^{x,y}}\left[\frac{\mathrm{d}\mathbb{Q}_{s,t}^{x}}{\mathrm{d}\mathbb{W}_{s,t}^{x}}(X)\,\middle|\,R\right], \tag{7.12}$$

and so we can instead focus on finding an unbiased estimator for functions of the following form,

$$\psi(X|R) := \frac{\mathrm{d}\mathbb{Q}_{s,t}^{x}}{\mathrm{d}\mathbb{W}_{s,t}^{x}}(X) = \exp\left\{-\int_s^t \phi(X_u)\,\mathrm{d}u\right\}, \quad \text{where } X \sim \mathbb{W}_{s,t}^{x,y}\,\big|\,R\,. \tag{7.13}$$

Recall that methodology on how to simulate a path space layer, and conditional on the path space layer simulate a finite dimensional realisation of a Brownian bridge sample path, was detailed in Sections 6.2 and 6.3.

Now, following a similar approach to Section 7.1.1 and denoting by $q(\kappa|R)$, $\kappa = 0, 1, \ldots$ as the pdf of a discrete random variable κ with law $\mathbb{K}$, we have,

$$\begin{aligned}
\psi(X|R) &= \exp\left\{-\int_s^t \phi(X_u)\,\mathrm{d}u\right\} \\
&= e^{-U_X(t-s)}\sum_{j=0}^{\infty}\left[\frac{(t-s)^j}{j!}\cdot\left(\int_s^t \frac{U_X-\phi(X_u)}{t-s}\,\mathrm{d}u\right)^j\right] \\
&= \mathbb{E}_{\mathbb{K}}\left[e^{-U_X(t-s)}\cdot\frac{(t-s)^\kappa}{\kappa!\cdot q(\kappa|R)}\cdot\left(\int_s^t \frac{U_X-\phi(X_u)}{t-s}\,\mathrm{d}u\right)^\kappa\,\middle|\,X\right] \\
&= \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[e^{-U_X(t-s)}\cdot\frac{(t-s)^\kappa}{\kappa!\cdot q(\kappa|R)}\cdot\prod_{i=1}^{\kappa}\left[U_X-\phi(X_{\xi_i})\right]\,\middle|\,X\right].
\end{aligned} \tag{7.14}$$

As such, as outlined in Algorithm 7.1.2, we can construct an unbiased estimate of $\psi(X|R)$ by uniformly scattering $\kappa \sim q(\kappa|R)$ points on the interval $[s,t]$, evaluating a sample path

$X \sim \mathbb{W}_{s,t}^{x,y} \,\big|\, R$ at these points and setting,

$$\hat{\psi}_{\mathrm{GPE}}(X) = e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \Big[U_X - \phi(X_{\xi_i}) \Big]. \tag{7.15}$$

Algorithm 7.1.2 Generalised Poisson Estimator (GPE) [Fearnhead et al., 2008].

1. Simulate $R \sim \mathcal{R}$.
2. Simulate $\kappa \sim q(\kappa|R)$ and skeleton times $\xi_1, \ldots, \xi_\kappa \overset{\text{iid}}{\sim} \mathrm{U}[s,t]$.
3. Simulate sample path at skeleton times $X_{\xi_1}, \ldots, X_{\xi_\kappa} \sim \mathbb{W}_{s,t}^{x,y} \,\big|\, R$.
4. Set $\hat{\psi}_{\mathrm{GPE}}(X) = e^{-U_X(t-s)} \cdot \dfrac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|U_X, L_X)} \cdot \prod_{i=1}^{\kappa} \Big[U_X - \phi(X_{\xi_i}) \Big]$.

The second moment of the GPE can be derived as follows,

$$\begin{aligned}
&\mathbb{E}_{\mathbb{K}} \mathbb{E}_{\mathbb{U}_\kappa} \left[\left(e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \Big[U_X - \phi(X_{\xi_i}) \Big] \right)^2 \,\middle|\, X \right] \\
&\quad = e^{-2U_X(t-s)} \cdot \mathbb{E}_{\mathbb{K}} \mathbb{E}_{\mathbb{U}_\kappa} \left[\frac{(t-s)^{2\kappa}}{\kappa!^2 \cdot q^2(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \Big[U_X - \phi(X_{\xi_i}) \Big]^2 \,\middle|\, X \right] \\
&\quad = e^{-2U_X(t-s)} \cdot \mathbb{E}_{\mathbb{K}} \left[\frac{(t-s)^{2\kappa}}{\kappa!^2 \cdot q^2(\kappa|R)} \cdot \left(\int_s^t \frac{[U_X - \phi(X_u)]^2}{t-s} \, \mathrm{d}u \right)^\kappa \,\middle|\, X \right] \\
&\quad = e^{-2U_X(t-s)} \sum_{j=0}^{\infty} \left[\frac{(t-s)^j}{j!^2 \cdot q(j|R)} \cdot \left(\int_s^t [U_X - \phi(X_u)]^2 \, \mathrm{d}u \right)^j \right] \\
&\quad =: e^{-2U_X(t-s)} \sum_{j=0}^{\infty} \frac{f_j}{q_j}
\end{aligned} \tag{7.16}$$

Now, as we have in this setting flexibility over the precise choice of the discrete probability density $q(j|R)$, then we choose $q(j|R)$ to minimise (7.16). In particular, using Lagrange multipliers, we want to optimise,

$$\sum_{i=0}^{\infty} \frac{f_i}{q_i} + \lambda \left(\sum_{i=0}^{\infty} q_i - 1 \right), \tag{7.17}$$

so we have,

$$\frac{\partial}{\partial q_j}\left[\sum_{i=0}^{\infty}\frac{f_i}{q_i}+\lambda\left(\sum_{i=0}^{\infty}q_i-1\right)\right]=-\frac{f_j}{q_j^2}+\lambda=0. \tag{7.18}$$

As $q_j \in [0,1]$ we have $q_j = \sqrt{f_j/\lambda}$. Now, finding λ, note that we have $\sum_{j=0}^{\infty} q_j = 1$, and so $\sum_{j=0}^{\infty}\sqrt{f_j/\lambda} = 1$ and $\lambda = \left(\sum_{j=0}^{\infty}\sqrt{f_j}\right)^2$ so we have $q_j = \sqrt{f_j}/\left(\sum_{i=0}^{\infty}\sqrt{f_i}\right)$. Substituting this result into (7.16) we find the second moment is minimised by selecting the discrete probability density as follows (recall that L_X and U_X are selected with reference to R),

$$\begin{aligned}
q(j|R) &\propto \left[\frac{(t-s)^j}{j!^2}\cdot\left(\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^j\right]^{1/2}\\
&= \frac{1}{j!}\left[\left((t-s)\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^{1/2}\right]^j\\
&\propto \frac{1}{j!}\left[\left((t-s)\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^{1/2}\right]^j\cdot\exp\left\{-\left((t-s)\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^{1/2}\right\}.
\end{aligned} \tag{7.19}$$

Noting that (7.19) is the pdf of a Poisson distribution, then (7.16) is minimised by selecting $q(j|R)$ to be a Poisson distribution with intensity,

$$\lambda_R := \left((t-s)\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^{1/2}. \tag{7.20}$$

Unfortunately, the integral expression in (7.20) precludes this choice. The most natural selection of $q(j|R)$ in this setting (which following the convention in Fearnhead et al. [2008] we call GPE-1) is therefore to choose a Poisson distribution with intensity λ_S, which bounds the intensity λ_R. In particular, noting that $\phi(X_u) \geq L_X$, we have,

$$\begin{aligned}
\left((t-s)\int_s^t\left[U_X-\phi(X_u)\right]^2\,\mathrm{d}u\right)^{1/2} &\leq \left((t-s)\int_s^t\left[U_X-L_X\right]^2\,\mathrm{d}u\right)^{1/2}\\
&= (U_X-L_X)\cdot(t-s) =: \lambda_S.
\end{aligned} \tag{7.21}$$

With this selection we ensure our estimator is unbiased and positive,

$$\begin{aligned}\hat{\psi}_{\text{GPE-1}}(X) &= e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \left[U_X - \phi(X_{\xi_i})\right] \\ &:= e^{-L_X(t-s)} \cdot \prod_{i=1}^{\kappa} \frac{U_X - \phi(X_{\xi_i})}{U_X - L_X} \in \left[0, e^{-L_X(t-s)}\right], \end{aligned} \tag{7.22}$$

with finite second moment,

$$\begin{aligned}&\mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\left(e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \left[U_X - \phi(X_{\xi_i})\right]\right)^2 \middle| X\right] \\ &:= \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[\left(e^{-L_X(t-s)} \cdot \prod_{i=1}^{\kappa} \frac{U_X - \phi(X_{\xi_i})}{U_X - L_X}\right)^2 \middle| X\right] \\ &\leq \mathbb{E}_{\mathbb{K}}\mathbb{E}_{\mathbb{U}_\kappa}\left[e^{-2\Phi(t-s)} \middle| X\right] = e^{-2\Phi(t-s)} < \infty. \end{aligned} \tag{7.23}$$

An alternate selection for $q(j|R)$, GPE-2, was proposed in Fearnhead et al. [2008]. Since the optimal selection for $q(j|R)$ is a Poisson distribution with (stochastic) optimal intensity λ_R, Fearnhead et al. [2008] instead propose the selection of the heavier tailed negative binomial distribution ($\text{NBin}(j; \gamma_R, \beta_r)$, where γ_R is the mean parameter and β_R is the dispersion parameter), setting the mean parameter of the negative binomial distribution equal to an approximation of the optimal intensity. This selection results in the following unbiased and positive estimator,

$$\begin{aligned}\hat{\psi}_{\text{GPE-2}}(X) &= e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa}{\kappa! \cdot q(\kappa|R)} \cdot \prod_{i=1}^{\kappa} \left[U_X - \phi(X_{\xi_i})\right] \\ &:= e^{-U_X(t-s)} \cdot \frac{(t-s)^\kappa \cdot \Gamma(\beta_R) \cdot (\gamma_R + \beta_R)^{\gamma_R+\kappa}}{\Gamma(\beta_R + \kappa) \cdot \gamma_R^\kappa \cdot \beta_R^{\beta_R}} \cdot \prod_{i=1}^{\kappa} \frac{U_X - \phi(X_{\xi_i})}{U_X - L_X}. \end{aligned} \tag{7.24}$$

Fearnhead et al. [2008] propose using the following approximation of the optimal intensity (motivated by Jensen's inequality),

$$\lambda_R \approx \gamma_R := U_X(t-s) - \int_s^t \phi\left(x \cdot \frac{t-s-u}{t-s} + y \cdot \frac{u}{t-s}\right) \mathrm{d}u, \tag{7.25}$$

noting that in simulations it compares favourably to both the VPE and GPE-1. Furthermore, in settings in which the VPE has finite variance then the GPE-2 also has finite variance, suggesting that in such settings the GPE-2 should be the favoured choice. However,

as noted in Section 7.1.1, settings in which it can be shown the VPE has finite variance are limiting and so throughout this chapter we employ the GPE-1.

7.2 Particle Filtering Algorithms for Jump Diffusions

Returning to the introductory remarks of this chapter, in this section we present a number of particle filtering algorithms for jump diffusions. We begin by presenting the Exact Propagation Particle Filter (EPPF; Algorithm 7.2.1) as motivated in the introductory remarks.

Noting that the GPE-1 presented in Section 7.1.2 can be used to assign an unbiased positive importance weight to a diffusion sample path conditioned to hit an end point simulated according to Biased Brownian motion (see Theorem 5.1.1), then the methodology developed for jump exact algorithms in Section 5.3 can be directly applied in this setting by appropriately substituting steps for accepting or rejecting proposal sample paths with an importance weight given by the GPE-1.

In Algorithm 7.2.2 we present the Bounded Random Weight Particle Filter (BRWPF) for jump diffusions with bounded jump intensity, which is based upon a direct application of the Sequential Importance Sampling / Resampling algorithm (SIS/R; Algorithm 3.4.1) and employing the ideas used in constructing the Bounded Jump Exact Algorithm (Algorithm 5.3.1). In Algorithm 7.2.3 we present the Unbounded Random Weight Particle Filter (URWPF) for jump diffusions with locally bounded jump intensity (see Condition 4), which is based upon the SIS/R algorithm and instead employing the ideas used in constructing the Unbounded Jump Exact Algorithm (Algorithm 5.3.2). Finally, in Figure 7.2.1 we present an illustrative example of the EPPF and BRWPF applied to a HMM.

Algorithm 7.2.1 Exact Propagation Particle Filter for Jump Diffusions (EPPF).

Initialisation Step ($t = 0$)**:-**

1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$, set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. If $\widehat{N_{\text{eff}}} \leq N_{\text{th}}$ then for i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t-1})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Simulate $x_t^{(i)} \sim p_1\left(x_{t-1}^{(i)}, z\right)$ as per the BJEA (Algorithm 5.3.1), UJEA (Algorithm 5.3.2) or AUJEA (Algorithm 5.3.3), and set $x_{0:t}^{(i)} := \left\{x_{0:t-1}^{(i)}, x_t^{(i)}\right\}$.
   (b) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right)$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.

Algorithm 7.2.2 Random Weight Particle Filter for Jump Diffusions with Bounded Jump Intensity (BRWPF).

Initialisation Step ($t = 0$)**:-**

1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$, set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

1. If $\widehat{N_{\text{eff}}} \leq N_{\text{th}}$ then for i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t-1})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Set $j = 0$ and $\Psi_j = t - 1$.
   (b) While $\Psi_j < t$,
      i. Simulate $\tau \sim \text{Exp}(\Lambda)$. Set $j = j + 1$ and $\Psi_j = \Psi_{j-1} + \tau$.
      ii. Simulate $x_{(\Psi_j \wedge t)-}^{(i)} \sim h\Big(z; x_{\Psi_{j-1}}^{(i)}, (\Psi_j \wedge t) - \Psi_{j-1}\Big)$.
      iii. Simulate $\pitchfork_j := \hat{\psi}_{\text{GPE-1}}\Big(x_{[\Psi_{j-1},(\Psi_j \wedge t))}^{(i)}\Big)$ as per Algorithm 7.1.2.
      iv. If $\Psi_j > t$ then set $x_t^{(i)} = x_{t-}^{(i)}$ and $x_{0:t}^{(i)} := \Big\{x_{0:t-1}^{(i)}, x_t^{(i)}\Big\}$ else,
         A. With probability $\lambda\Big(x_{\Psi_j}^{(i)}\Big)/\Lambda$ set $x_{\Psi_j}^{(i)} := x_{\Psi_{j-}}^{(i)} + f_\nu\Big(x_{\Psi_{j-}}^{(i)}\Big)$, else set $x_{\Psi_j}^{(i)} := x_{\Psi_{j-}}^{(i)}$.
   (c) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot g_\theta\Big(y_t \Big| x_t^{(i)}\Big) \cdot \prod_{k=1}^j \pitchfork_k$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.

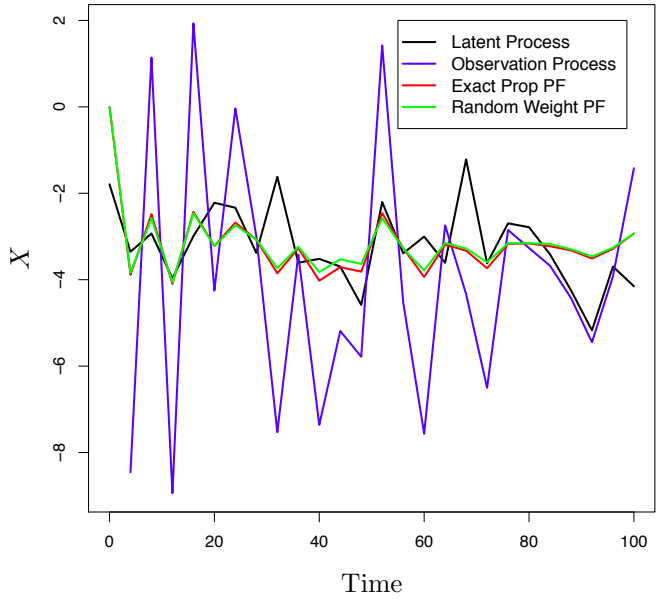
Algorithm 7.2.3 Random Weight Particle Filter for Jump Diffusions with Unbounded Jump Intensity (URWPF).

Initialisation Step ($t = 0$)**:-**

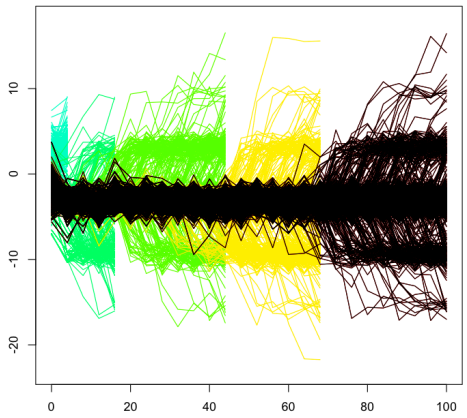
1. For i in 1 to N simulate $x_0^{(i)} \sim \mu_\theta(x_0)$, set $w_0^{(i)} = 1/N$.
2. Set $\pi_\theta^N(\mathrm{d}x_0) := \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$.

Update Steps ($t > 0$)**:-**

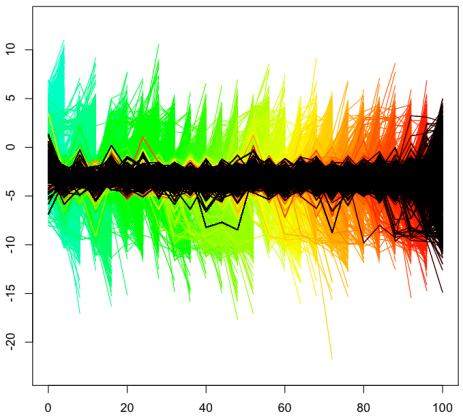
1. If $\widehat{N_{\text{eff}}} \leq N_{\text{th}}$ then for i in 1 to N sample $x_{0:t-1}^{(i)} \sim \pi_\theta^N(\mathrm{d}x_{0:t-1}|y_{1:t-1})$ (as per Algorithm 3.5.1, 3.5.2, 3.5.3 or 3.5.4) and set $w_{t-1}^{(i)} = 1/N$.
2. For i in 1 to N,
   (a) Set $j = 0$ and $\psi_j = t - 1$,
      i. Simulate skeleton end point $x_t^{(i)} \sim h(z; x_{\psi_j}^{(i)}, t - \psi_j)$.
      ii. Simulate layer information $R_{X[\psi_j,t]}^{i,j} \sim \mathcal{R}$ and compute $\Lambda_{X[\psi_j,t]}^{i,j}$.
      iii. Simulate proposal jump times $N_{t-\psi_j}^{\Lambda,i,j} \sim \text{Poi}\left(\Lambda_{X[\psi_j,t]}^{i,j}(t - \psi_j)\right)$ and $\Psi_1^{i,j}, \ldots, \Psi_{N_{t-\psi_j}^{\Lambda,i,j}}^{i,j} \overset{\text{iid}}{\sim} \text{U}[\psi_j, t]$.
      iv. Simulate $\kappa_{i,j} \sim \text{Poi}\left(\left[U_{X[\psi_j,t]}^{i,j} - L_{X[\psi_j,t]}^{i,j}\right] \cdot (t - \psi_j)\right)$ and skeleton times $\xi_1^{i,j}, \ldots, \xi_\kappa^{i,j} \overset{\text{iid}}{\sim} \text{U}[\psi_j, t]$.
      v. Simulate $x_{\xi_1^{i,j}}^{(i)}, \ldots, x_{\xi_\kappa^{i,j}}^{(i)}, x_{\Psi_1^{i,j}}^{(i)}, \ldots, x_{\Psi_{N(\Lambda,i,j,t-\psi_j)}^{i,j}}^{(i)} \sim \mathbb{W}_{\psi_j,t}^{x_{\psi_j}(i),x_t(i)} \Big| R_{X[\psi_j,t]}^{i,j}$.
      vi. Set $\Cap_j := \hat{\psi}_{\text{GPE-1}}\left(x_{[\Psi_{j-1},t]}^{(i)}\right)$.
      vii. For k in 1 to $N_{t-\psi_j}^{\Lambda,i,j}$,
         A. With probability $\lambda\left(x_{\Psi_k^{i,j}}^{(i)}\right)/\Lambda_{X[\psi_j,t]}^{i,j}$ set $x_{\Psi_k^{i,j}-}^{(i)} = x_{\Psi_k^{i,j}}^{(i)}$, $x_{\Psi_k^{i,j}}^{(i)} := x_{\Psi_k^{i,j}-}^{(i)} + f_\nu\left(x_{\Psi_k^{i,j}-}^{(i)}\right)$, $\psi_{j+1} := \Psi_k^{i,j}$, $j = j + 1$, and return to Step 2(a)i.
   (b) Set $w_t^{*(i)} = w_{t-1}^{(i)} \cdot g_\theta\left(y_t \,\middle|\, x_t^{(i)}\right) \cdot \prod_{k=1}^j \Cap_k$.
3. For i in 1 to N set $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$.
4. Set $\pi_\theta^N(\mathrm{d}x_{0:t}|y_{1:t}) := \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$.



(a) The observation process, EPPF and BRWPF, overlaid with the underlying latent process.



(b) EPPF particle ancestral paths.



(c) BRWPF particle ancestral paths.

Figure 7.2.1: An illustrative example of an Exact Propagation Particle Filter (EPPF; Algorithm 7.2.1) and a Bounded Random Weight Particle Filter (BRWPF; Algorithm 7.2.2) of 2,000 particles applied to the HMM filtering problem with SSDs as follows, $X_0 \sim \mathrm{N}(0, 5)$, $Y_t|(X_t = x_t) \sim \mathrm{N}(0, 10)$ and the latent process is governed by a jump diffusion with the following SDE, $\mathrm{d}X_t = \sin(X_{t^-})\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_T^{\lambda,\nu}$ where $\lambda(X_t) = \cos^2(X_t)$ and $f_\nu(X_t) = \mathrm{N}(\sin(X_t), 1)$. In subfigures (b) and (c) we show for this example the ancestral paths of the particles. Paths in black denote ancestral paths of the particles comprising the empirical filtering density at time 100, whereas those in other colours indicate that they are no longer included after some resampling point.

8 ϵ-Strong Simulation of Diffusions and Jump Diffusions

"Judges must beware of hard constructions and strained inferences, for there is no worse torture than that of laws."

— Francis Bacon

In this chapter we outline a novel approach for simulating upper and lower bounding processes which almost surely constrain (jump) diffusion and (jump) diffusion bridge sample paths to any specified tolerance. We do this by means of a significant extension to the ϵ-*Strong Simulation* algorithm proposed in Beskos et al. [2012], which we present in Section 8.1. In Section 8.2 we present an entirely novel approach to constructing an exact algorithm based on the ϵ-strong simulation methodology in Section 8.1.

Finally, in Section 8.3 we present a number of applications of the methodology developed throughout this chapter and the thesis in general. In particular we show that it is possible to determine exactly whether a diffusion or jump diffusion sample path, simulated as per one of the adaptive exact algorithms in Chapter 5, crosses various types of barriers. The flexibility of the adaptive exact algorithms developed in this thesis 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 simulate (without discretisation error) from the transition density of killed (or un-killed) diffusions and jump diffusions, among many other interesting possibilities.

8.1 ϵ-Strong Simulation Methodology

As originally proposed in Beskos et al. [2012] and presented in Algorithm 8.1.1, ϵ-strong simulation is an algorithm which simulates upper and lower convergent bounding processes ($X^{\uparrow}$ and $X^{\downarrow}$) which enfold almost surely Brownian motion sample paths over some finite interval $[0, T]$. In particular, we obtain a sequence of ever closer bounds such that $\forall\, u \in [0, T]$ and some counter n,

$$X_u^{\downarrow}(n) \leq X_u^{\downarrow}(n+1) \leq X_u \leq X_u^{\uparrow}(n+1) \leq X_u^{\uparrow}(n), \tag{8.1}$$

Algorithm 8.1.1 is not exactly that presented in Beskos et al. [2012], but incorporates a number of extensions based on the results in Section 6.3. In particular, in contrast to Beskos et al. [2012], we can now simulate an initial intersection layer (Algorithm 8.1.1 Step 2) and simulate the intermediate points exactly (Algorithm 8.1.1 Step 3(a)i).

Algorithm 8.1.1 ϵ-Strong Simulation of Brownian Motion sample paths (n bisections).

---

1. Simulate $X_T := y \sim \mathrm{N}(0, T)$ and set $i = 1$.
2. Simulate initial intersection layer $\mathcal{S} := \mathcal{S}_{0,T} = \left\{0, T, X_0, X_T, \ell_{0,T}^{\downarrow}, \ell_{0,T}^{\uparrow}, \upsilon_{0,T}^{\downarrow}, \upsilon_{0,T}^{\uparrow}\right\}$ as per Algorithm 6.3.1.
3. While $i \leq n$,
   (a) For each of the 2^{i-1} intersection layers in $\mathcal{S}$ (denoted $\mathcal{S}_{s,t}^{j} := \left\{s^j, t^j, X_s^j, X_t^j, \ell_{s,t}^{j,\downarrow}, \ell_{s,t}^{j,\uparrow}, \upsilon_{s,t}^{j,\downarrow}, \upsilon_{s,t}^{j,\uparrow}\right\}$),
      i. Simulate X_q where $q := (s^j + t^j)/2$ conditional on $\mathcal{S}_{s,t}^{j}$ as per Algorithm 6.3.2.
      ii. Bisect $\mathcal{S}_{s,t}^{j}$ into $\mathcal{S}_{s,q}^{j,1}$ and $\mathcal{S}_{q,t}^{j,2}$ as per Algorithm 6.3.5.
      iii. For $\mathcal{S}_{s,q}^{j,1}$ and $\mathcal{S}_{q,t}^{j,2}$, while $\left|\ell_{s,t}^{j,*,\uparrow} - \ell_{s,t}^{j,*,\downarrow}\right| > \sqrt{(t^j - s^j)/2}$ or $\left|\upsilon_{s,t}^{j,*,\uparrow} - \upsilon_{s,t}^{j,*,\downarrow}\right| > \sqrt{(t^j - s^j)/2}$ then refine intersection layer as per Algorithm 6.3.6.
   (b) Set $\mathcal{S} := \bigcup_{j=1}^{2^{i-1}} \left\{\mathcal{S}_{s,q}^{j,1} \bigcup \mathcal{S}_{q,t}^{j,2}\right\}$ and $i = i + 1$.

The intersection layer information can be used to find the bounding processes in (8.1) and, as shown in [Beskos et al., 2012, Prop. 3.1], convergence in the supremum norm

holds,

$$\text{w.p. 1: } \lim_{n\to\infty} \sup_{u} \left| X_u^{\uparrow}(n) - X_u^{\downarrow}(n) \right| \to 0, \tag{8.2}$$

where,

$$X_u^{\uparrow}(n) := \sum_{i=1}^{2^{n-1}} \upsilon_{s,t}^{i,\uparrow} \cdot \mathbb{1}(u \in [s^i, t^i]), \qquad X_u^{\downarrow}(n) := \sum_{i=1}^{2^{n-1}} \ell_{s,t}^{i,\downarrow} \cdot \mathbb{1}(u \in [s^i, t^i]). \tag{8.3}$$

The (seemingly arbitrary) square root term in Algorithm 8.1.1 Step 3(a)iii used to determine whether or not to continue refining an intersection layer (as per Algorithm 6.3.6), is chosen as it is sufficient to ensure convergence of the upper and lower bounding processes ($X^{\uparrow}$ and $X^{\downarrow}$) to X, while striking a balance computationally between bisecting an intersection layer (as per Algorithm 6.3.5) and refining an intersection layer (as per Algorithm 6.3.6). As noted by Beskos et al. [2012], as the range of Brownian motion on an interval of length 2^{-n} scales as $O(2^{-n/2})$, if we were able to identify for each interval the minimum and maximum of the sample path within that interval (using the minimum and maximum as the lower and upper convergent bounding processes) and conduct ϵ-strong simulation, then this (idealised) algorithm would have a rate of convergence of $O(2^{-n/2})$. As Algorithm 8.1.1 Step 3(a)iii ensures that for each intersection layer the lower and upper bounding processes are within $O(2^{-n/2})$ of the minimum and maximum of the sample path within that interval (respectively), then the rate of convergence of the idealised algorithm is preserved in the implementable algorithm (Algorithm 8.1.1).

It was further shown in [Beskos et al., 2012, Prop. 3.2] that the dominating processes converge in the L_1 norm with rate of the order $O(2^{-n/2})$,

$$2^{n/2} \times \mathbb{E}\left[\left| X^{\uparrow}(n) - X^{\downarrow}(n) \right|_1 \right] = O(1), \tag{8.4}$$

where,

$$\left| X^{\uparrow}(n) - X^{\downarrow}(n) \right|_1 = \int_0^T \left| X_u^{\uparrow}(n) - X_u^{\downarrow}(n) \right| \mathrm{d}u. \tag{8.5}$$

As the upper and lower bounding processes are piecewise constant functions then it is useful to note that we simply have,

$$X^{\uparrow}(n) = \sum_{i=1}^{2^{n-1}} \upsilon_{s,t}^{i,\uparrow} \cdot (t^i - s^i), \qquad X^{\downarrow}(n) = \sum_{i=1}^{2^{n-1}} \ell_{s,t}^{i,\uparrow} \cdot (t^i - s^i). \tag{8.6}$$

Now, considering the ϵ-*Strong Simulation of Jump Diffusions*, note that upon simulating a (jump) diffusion or (jump) diffusion bridge sample path skeleton (as per the AUEA, CAUEA, AUJEA or CAUJEA), it has a form (see (5.18), (5.21), (5.26) and (5.45)) that can be used in Algorithm 8.1.1. As such, Algorithm 8.1.1 can be extended to jump diffusions (Algorithm 8.1.2) by using the strategies presented earlier in Section 6.3, noting that (8.2) and (8.4) still hold.

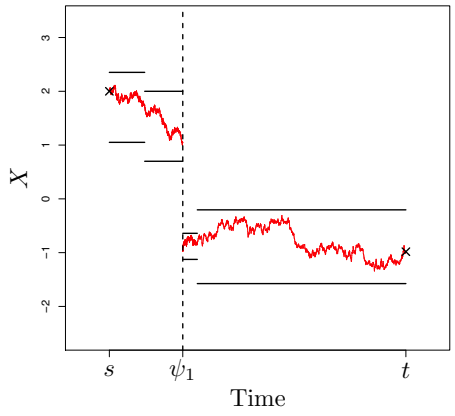
Algorithm 8.1.2 ϵ-Strong Simulation of Jump Diffusion sample paths (n bisections).

1. Simulate jump diffusion skeleton as per Algorithm 5.1.4 to obtain initial intersection layer.
2. Simulate further intersection layers as required (n bisections) as per Algorithm 8.1.1.

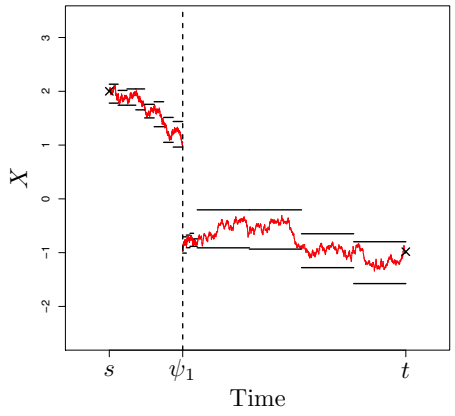
As far as we are aware there are no existing methods for the ϵ-strong simulation of jump diffusions. The class of jump diffusions this methodology can be applied to is broad (the conditions outlined in Section 1.3 are sufficient) and motivate a number of avenues for future research. In particular, non-trivial characteristics of the diffusion path can be simulated (for instance extrema, hitting times, integrals) and can be applied to areas such as option pricing and the simulation of stochastic volatility models (which are currently being explored in related work). The precise implementation of Algorithm 8.1.2 can be tailored to the specific application. For instance, in Figure 8.1.1 we present the ϵ-strong simulation of a jump diffusion sample path as detailed in Algorithm 8.1.2, whereas in Figure 8.1.2 we instead consider an alternate tolerance based ϵ-strong simulation of a jump diffusion sample path in which we are instead interested in minimising (for any given computational budget) the L_1 distance. In particular, at each point in the ϵ-strong simulation algorithm the next intersection layer of the existing set of intersection layers

(for convenience, numbered as $1, \ldots, M$) to be chosen to bisect, is the interval,

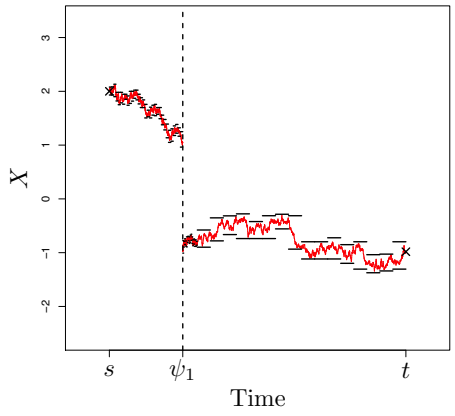
$$i^* := \underset{i \in \{1,\ldots,M\}}{\arg\max} \left[\left(\upsilon_{s,t}^{i,\uparrow} - \ell_{s,t}^{i,\uparrow} \right) \cdot (t^i - s^i) \right]. \tag{8.7}$$



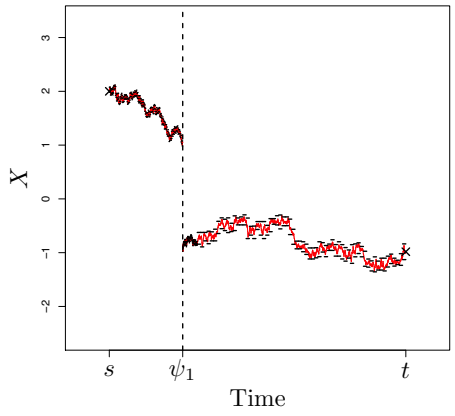
(a) Sample path skeleton



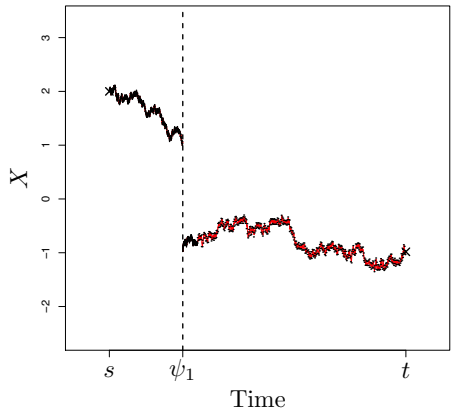
(b) After $n = 2$ bisections



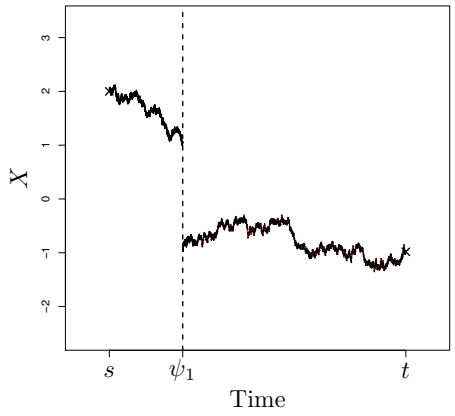
(c) After $n = 4$ bisections



(d) After $n = 6$ bisections

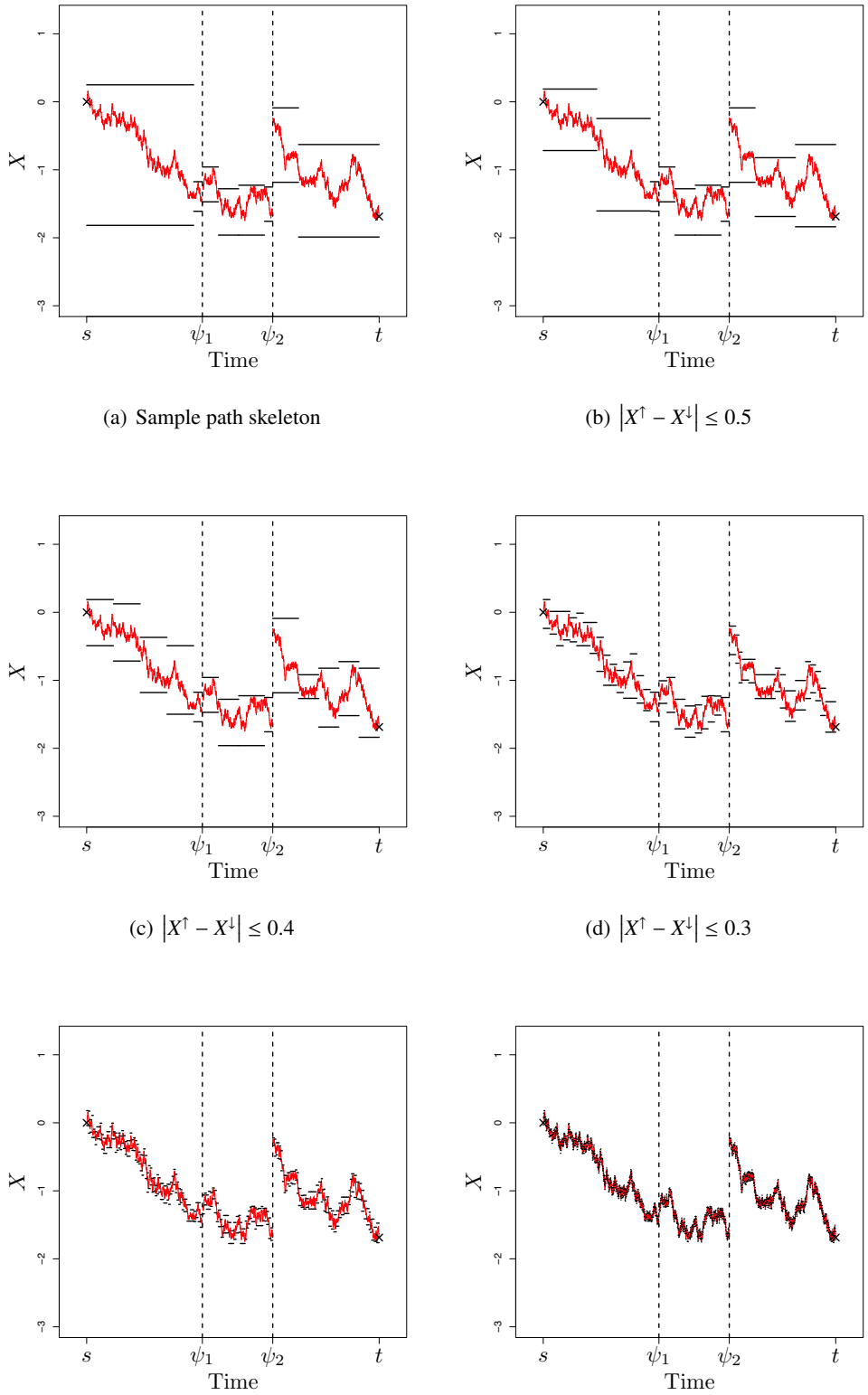


(e) After $n = 8$ bisections



(f) After $n = 10$ bisections

Figure 8.1.1: Illustration of standard ϵ-strong simulation of a jump diffusion sample path, overlaid with sample path.



(a) Sample path skeleton

(b) $\left|X^{\uparrow} - X^{\downarrow}\right| \leq 0.5$

(c) $\left|X^{\uparrow} - X^{\downarrow}\right| \leq 0.4$

(d) $\left|X^{\uparrow} - X^{\downarrow}\right| \leq 0.3$

(e) $\left|X^{\uparrow} - X^{\downarrow}\right| \leq 0.2$

(f) $\left|X^{\uparrow} - X^{\downarrow}\right| \leq 0.1$

Figure 8.1.2: Illustration of modified tolerance based ϵ-strong simulation of a jump diffusion sample path, overlaid with sample path.

8.2 An ϵ-Strong Exact Algorithm for Diffusions

In this section we detail a novel approach for constructing an exact algorithm based on the ϵ-strong simulation methodology in Section 8.1. We begin by recalling from the introductory remarks of Section 5.1 the key steps in constructing an exact algorithm. In particular, returning to our initial Implementable Exact Algorithm (Algorithm 5.1.2), then recall that after simulating the end point from biased Brownian motion (Algorithm 5.1.2 Step 1), the remainder of the proposal sample path can be simulated exactly from the law of a Brownian bridge (see Theorem 5.1.1). 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 by $\mathbb{Q}^x_{0,T}$) we accept the sample path with probability $P_{\mathbb{Z}^x_{0,T}}(X)$. In Section 5.1 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 8.1.1 we can find upper and lower convergent bounding sequences for $P_{\mathbb{Z}^x_{0,T}}(X)$ (in analogous fashion to (8.1)) by directly mapping the upper and lower bounds of the underlying proposal sample path X obtained from ϵ-strong simulation,

$$0 \leq \ldots \leq \phi_n^{\downarrow} \leq \phi_{n+1}^{\downarrow} \leq \ldots \leq P_{\mathbb{Z}^x_{0,T}}(X) \leq \ldots \leq \phi_{n+1}^{\uparrow} \leq \phi_n^{\uparrow} \leq \ldots \leq 1, \tag{8.8}$$

where we define $P_{\mathbb{Z}^x_{0,T}}(X)$ and the bounding sequences as follows (recalling $\phi(X)$ is bounded on compact sets (Result 4)),

$$P_{\mathbb{Z}^x_{0,T}}(X) := \exp\left\{-\int_0^T (\phi(X_s) - \Phi)\, \mathrm{d}s\right\}, \tag{8.9}$$

$$\phi_n^{\downarrow} := \exp\left\{-\sum_{i=1}^{n}\left(\sup_{u\in\left[\ell_{s,t}^{i,\downarrow}, v_{s,t}^{i,\uparrow}\right]} (\phi(u) - \Phi)\right)\cdot(t^i - s^i)\right\}, \tag{8.10}$$

$$\phi_n^{\uparrow} := \exp\left\{-\sum_{i=1}^{n}\left(\inf_{u\in\left[\ell_{s,t}^{i,\downarrow}, v_{s,t}^{i,\uparrow}\right]} (\phi(u) - \Phi)\right)\cdot(t^i - s^i)\right\}. \tag{8.11}$$

As such we can simulate events of probability $P_{\mathbb{Z}^x_{0,T}}(X)$ by direct application of series sampling (see Section 2.6) and hence construct an ϵ-*Strong Exact Algorithm* (ϵEA) as outlined in Algorithm 8.2.1. The precise implementation of Algorithm 8.2.1 differs from that of Algorithm 8.1.1 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 8.1.2). More precisely, at step $(n+1)$ we choose to bisect and refine interval,

$$i^* := \underset{i\in\{1,\ldots,n\}}{\arg\max}\left[\left(\sup_{u\in\left[\ell^{i,\downarrow}_{s,t},\upsilon^{i,\uparrow}_{s,t}\right]}\phi(u) - \inf_{u\in\left[\ell^{i,\downarrow}_{s,t},\upsilon^{i,\uparrow}_{s,t}\right]}\phi(u)\right)\cdot(t^i - s^i)\right]. \tag{8.12}$$

Algorithm 8.2.1 ϵ-Strong Exact Algorithm (ϵEA).

1. Simulate skeleton end point $X_T := y \sim h$.
2. Simulate initial intersection layer $\mathcal{S} := \mathcal{S}_{0,T} = \left\{0, T, X_0, X_T, \ell^{\downarrow}_{0,T}, \ell^{\uparrow}_{0,T}, \upsilon^{\downarrow}_{0,T}, \upsilon^{\uparrow}_{0,T}\right\}$ as per Algorithm 6.3.1.
3. Simulate $u \sim U[0,1]$, set $n = 1$ and compute $\phi^{\downarrow}_n$ and $\phi^{\uparrow}_n$.
4. While $u \in (\phi^{\downarrow}_n, \phi^{\uparrow}_n)$,
   (a) Select interval i^* as per (8.12).
   (b) Simulate X_{q_n} where $q_n := (s^{i*} + t^{i*})/2$ conditional on $\mathcal{S}^{i*}_{s,t}$ as per Algorithm 6.3.2.
   (c) Bisect and refine $\mathcal{S}^{i*}_{s,t}$ into $\mathcal{S}^{i*,1}_{s,q_n}$ and $\mathcal{S}^{i*,2}_{q_n,t}$ as per Algorithm 6.3.5 and Algorithm 8.1.1 Step 3(a)iii.
   (d) Set $\mathcal{S} := \mathcal{S} \cup \mathcal{S}^{i*,1}_{s,q_n} \cup \mathcal{S}^{i*,2}_{q_n,t} \setminus \mathcal{S}^{i*}_{s,t}$, set $n = n+1$ and compute $\phi^{\downarrow}_n$ and $\phi^{\uparrow}_n$.
5. If $u \le \phi^{\downarrow}_n$ accept skeleton, defining $\xi_0, \xi_1, \ldots, \xi_{n+1}$ as the order statistics of the set $\{s, q_1, \ldots, q_n, t\}$ else if $u \ge \phi^{\uparrow}_n$ reject and return to Step 1.

---

6. * *Simulate* $X^{rem} \sim \left(\otimes_{i=1}^{n+1} \mathbb{W}^{X_{\xi_{i-1}}, X_{\xi_i}}_{\xi_{i-1},\xi_i} \middle| R_X^{[\xi_{i-1},\xi_i]}\right)$.

Algorithm 8.2.1 can be employed to simulate the same class of diffusions as outlined in Section 1.3 and furthermore satisfies Principles 1, 2 and 3. The resulting skeleton comprises all simulated intersection layers as shown in (8.13) and admits the further simulation of intermediate points by direct application of Algorithm 6.3.7. It should be noted that extension of this exact algorithm to diffusion bridges, jump diffusions and jump diffusion bridges can be straight forwardly performed in an analogous fashion to the extension

of the exact algorithms in Section 5 to those exact algorithms presented in Sections 5.2, 5.3 and 5.4.

$$\mathcal{S}_{\epsilon\mathrm{EA}}(X) := \left\{ \left(\xi_i, X_{\xi_i}\right)_{i=0}^{\kappa+1}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\kappa+1} \right\}. \tag{8.13}$$

The natural extension to Algorithm 8.2.1 is the AUEA presented in Algorithm 5.1.4 of Section 5.1.2, which on implementation is far more computationally efficient than Algorithm 8.2.1 due to the slow convergence of the bounding sequences enfolding $\mathcal{P}_{\mathbb{Z}_{0,T}^x}(X)$ in (8.8). However, we have included this algorithm here as in addition to providing a direct application of ϵ-strong simulation as presented in Section 8, it is a novel approach to the exact algorithm which opens up interesting avenues to tackle related problems.

8.3 Unbiased Estimation of Irregular Barrier Crossing Probabilities

In this section we present a number of applications demonstrating that it is possible to determine exactly whether a jump diffusion sample path simulated as per the Adaptive Unbounded Jump Exact Algorithm (AUJEA; Algorithm 5.3.3) crosses various types of barriers. This is in contrast to existing works which are restricted to considering either piecewise linear barriers and Brownian motion sample paths (see Pötzelberger and Wang [1997] and Pötzelberger and Wang [2001]) or introducing approximation error (see for instance Milstein and Tretyakov [1999]). In Section 8.3.1 we consider a nonlinear two sided barrier, in Section 8.3.2 we consider the crossing of two jump diffusion sample paths from different laws, and finally in Section 8.3.3 we consider the crossing of a circular barrier by a 2 dimensional jump diffusion sample path.

In addition, for each example we construct and simulate other non-trivial quantities, such as an unbiased estimate of the probability that the barrier is crossed, an unbiased estimate of the probability that the barrier is crossed by any particular time, and the killed (or unkilled) diffusion transition density.

In each of our examples we employ variants of Algorithm 8.3.1. For simplicity in Algorithm 8.3.1 we only consider the crossing of an upper barrier by a one dimensional jump diffusion, however, as we will discuss later in this section and in Sections 8.3.1–8.3.3 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 introduce the notation 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.

Algorithm 8.3.1 Unbiased Estimation of Upper Barrier Crossing.

1. Simulate skeleton $\mathcal{S}_{\text{AUJEA}}(X) := \left\{\left(\xi_i, X_{\xi_i}\right)_{i=0}^{\sum_j(\kappa_j+1)}, \left(R_X^{[\xi_{i-1},\xi_i]}\right)_{i=1}^{\sum_j(\kappa_j+1)}\right\}$ as per Algorithm 5.3.3.
2. Set $\mathcal{S} := \mathcal{S}_{\text{AUJEA}}(X)$, $C := \emptyset$, $\mathcal{B} := \emptyset$ and $\mathcal{U} := \left\{\left[s^1, t^1\right], \ldots, \left[s^{|\mathcal{S}|}, t^{|\mathcal{S}|}\right]\right\}$.
3. While $|C| = 0$ 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 $v_{s,t}^{i,\downarrow} \geq \sup_{u\in[s(i),t(i)]} B_u$ then $C := C \cup \left\{\left[s^i, t^i\right]\right\}$ and $\mathcal{U} := \mathcal{U} \setminus \left\{\left[s^i, t^i\right]\right\}$.
      ii. If $v_{s,t}^{i,\uparrow} \leq \inf_{u\in[s(i),t(i)]} B_u$ then $\mathcal{B} := \mathcal{B} \cup \left\{\left[s^i, t^i\right]\right\}$ and $\mathcal{U} := \mathcal{U} \setminus \left\{\left[s^i, t^i\right]\right\}$.
      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 6.3.2.
         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 6.3.5 and Algorithm 8.1.1 Step 3(a)iii.
         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 $\mathcal{U} := \mathcal{U} \cup \left\{\left[s^i, q\right], \left[q, t^i\right]\right\} \setminus \left\{\left[s^i, t^i\right]\right\}$.
4. If $|C| > 0$ then barrier crossed, else (if $|\mathcal{B}| = |\mathcal{S}|$) barrier not crossed.

8.3.1 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,

$$\begin{aligned} &\mathrm{d}X_t = \sin X_{t\text{-}}\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu} \qquad X_0 = 1 \qquad t \in [0, 2\pi], \\ &\lambda(X_t) = |X_t/4| \qquad f_\nu(X_t) = \mathrm{N}(-X_t/2, 2), \end{aligned} \tag{8.14}$$

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 respectively),

$$B_u^{\downarrow} = -4.5 - \cos(u), \qquad B_u^{\uparrow} = 4 + 0.5\cos(u), \qquad u \in [0, 2\pi]. \tag{8.15}$$

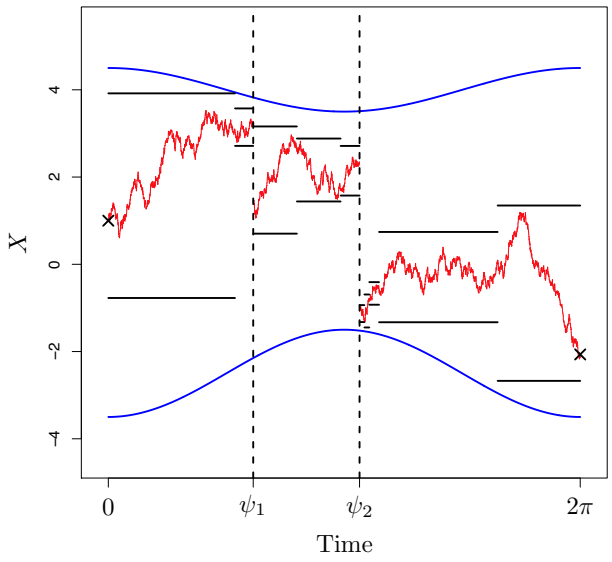
In this case, as the jump intensity of (8.14) can't be bounded we simulate sample paths using the AUJEA (see Algorithm 5.3.3). In particular, we have $\phi(X_t) := (\sin^2(X_t) + \cos(X_t))/2 \in [-1/2, 5/8]$, $\lambda(X_t)|\,(L_X, U_X) \leq \max\{L_X^2, U_X^2\}/4$ and the end point is simulated as follows, $X_T := y \sim h \propto \exp\{-\cos(y) - y^2/6\}$. In Figure 8.3.1 we present illustrations of whether the two sided barrier (8.15) has been crossed using finite dimensional realisations of sample paths simulated according to the measure induced by (8.14) and by applying a modified version of Algorithm 8.3.1. 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 100000 sample paths from the measure induced by (8.14) and determined whether the barrier (8.15) 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 8.1.

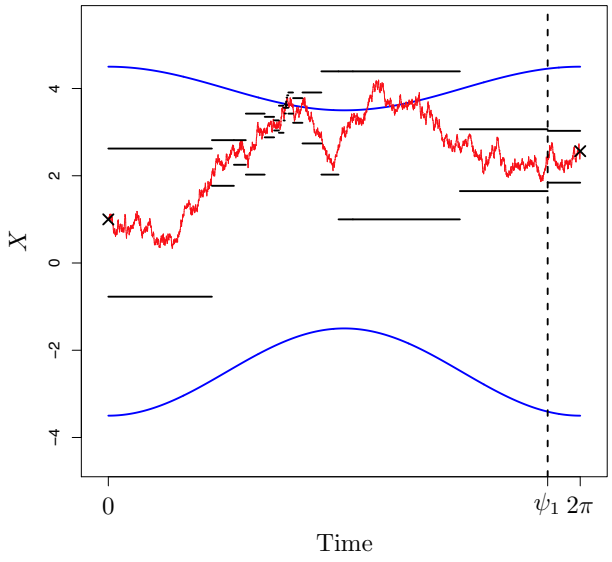
In Figure 8.3.2(a) we present kernel density estimates of the transition densities of various subsets of the sample paths simulated, including that for killed diffusions (i.e. sample paths from the measure induced by (8.14) with the restriction that they remain within the interval between the barriers in (8.15)). In Figure 8.3.2(b) we additionally determine for each sample path an interval of length $\epsilon \leq 10^{-4}$, in which the first crossing time occurs (by modifying the ϵ-strong algorithms presented in Section 8.1) to construct upper and lower bounds for the empirical CDF of the first barrier crossing time.

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%]

Table 8.1: Nonlinear two sided barrier example: Barrier crossing probabilities (computed using 100000 sample paths).

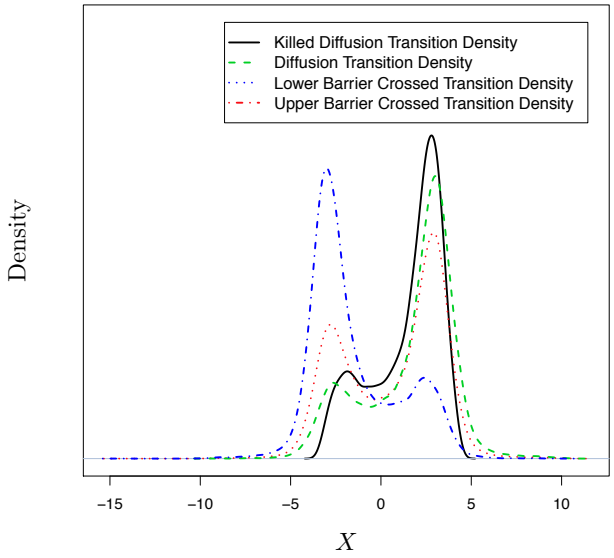


(a) No barrier crossing

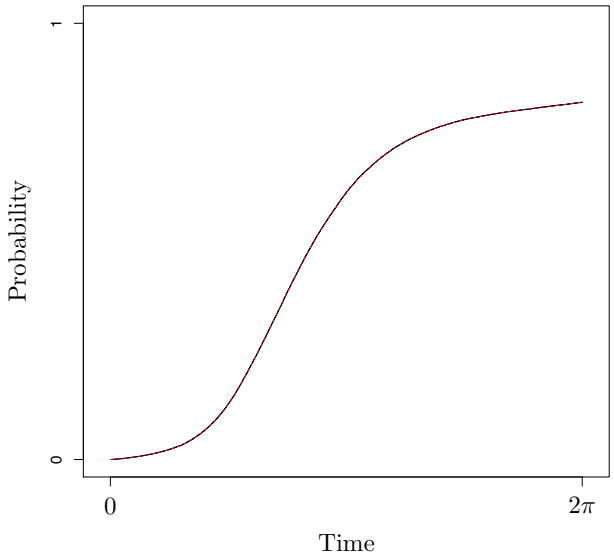


(b) Barrier crossed

Figure 8.3.1: 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.



(a) Kernel density estimates of the transition densities of subsets of sample paths simulated from the measure induced by (8.14).



(b) Empirical CDF of barrier crossing probability by time (crossing time evaluated within interval of length $\epsilon \leq 10^{-4}$). Upper and lower black lines indicate upper and lower bounds for the empirical CDF, whereas the red dotted line indicates the average of the two bounds.

Figure 8.3.2: Nonlinear two sided barrier example: Summary figures computed using 100000 sample paths.

8.3.2 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,

$$\begin{aligned} \mathrm{d}X_t &= -X_{t\text{-}}\,\mathrm{d}t + \mathrm{d}W_t + \mathrm{d}J_t^{\lambda,\nu}, \qquad t \in [0,2], \\ \lambda(X_t) &= \sin^2(X_t), \qquad f_\nu(X_t) = \mathrm{N}(-X_t/2, 1). \end{aligned} \tag{8.16}$$

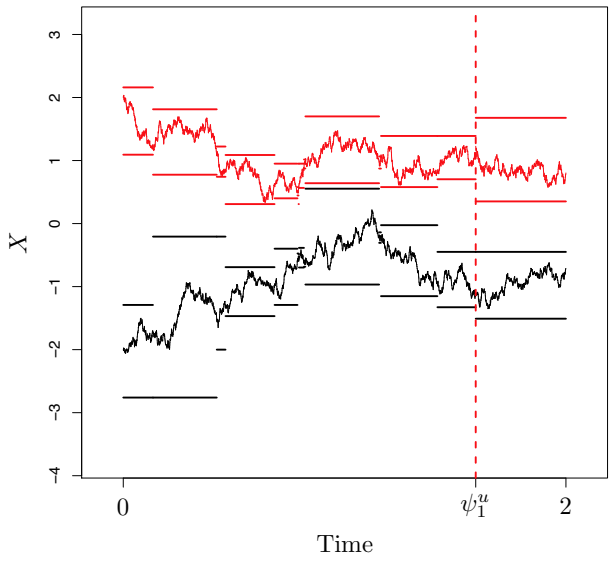
We consider sample paths simulated from the measure induced by (8.16) initialised at two possible starting values $X_0^\ell = -2$ and $X_0^\upsilon = 2$ (where X^ℓ and X^υ denote the lower and upper diffusions respectively). In this case the jump intensity of (8.16) is bounded so we simulate sample paths using the AUEA (see Algorithm 5.1.4) within the BJEA (see Algorithm 5.3.1). 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 8.3.3 we present illustrations of two sample paths simulated from the measure induced by (8.16), 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 by Bladt and Sørensen [Forthcoming], in which (in part) the authors are interested in the probability that two Brownian motion sample paths cross one another.

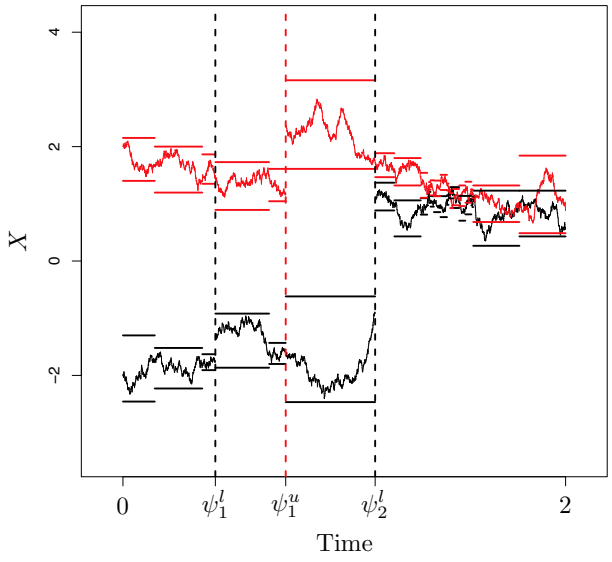
In this example we simulated 100000 pairs of sample paths from the measure induced by (8.16) initialised at $X_0^\ell = -2$ and $X_0^\upsilon = 2$ and determined whether or not they crossed. We present a summary of the unbiased estimates calculated from these sample paths in Table 8.2. In Figure 8.3.4(a) we present kernel density estimates of the transition densities of various subsets of the sample paths simulated. In Figure 8.3.4(b) we determine for each sample path an interval of length $\epsilon \leq 10^{-4}$ in which the first crossing time occurs in order to construct upper and lower bounds for the empirical CDF of the first crossing time.

Crossing Type	Empirical Probability	95% Clopper-Pearson Confidence Interval
No crossing	22.52%	[22.26%, 22.78%]
Crossing	77.48%	[77.22%, 77.74%]

Table 8.2: Jump diffusion crossing example: Crossing probabilities (computed using 100000 sample paths).

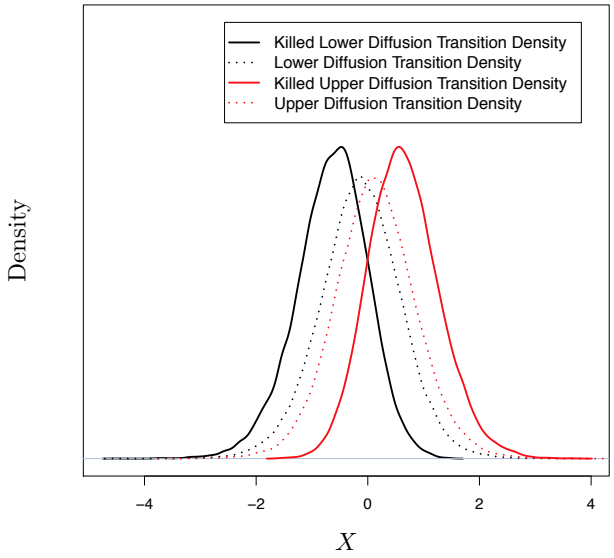


(a) No crossing

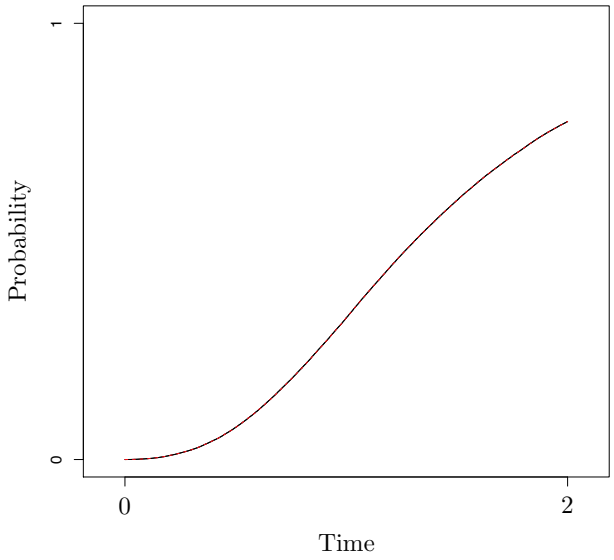


(b) Crossing

Figure 8.3.3: 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.



(a) Kernel density estimates of the transition densities of subsets of sample paths simulated from the measure induced by (8.16).



(b) Empirical CDF of jump diffusion crossing probability (crossing time evaluated within interval of length $\epsilon \leq 10^{-4}$). Upper and lower black lines indicate upper and lower bounds for the empirical CDF, whereas the red dotted line indicates the average of the two bounds.

Figure 8.3.4: Jump diffusion crossing example: Summary figures computed using 100000 sample paths.

8.3.3 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,

$$\begin{aligned}
&X := \left(X^{(1)}, X^{(2)}\right), \qquad X_0 = (0, 0.5), \qquad t \in [0, 3] \\
&\mathrm{d}X_t^{(1)} = -X_{t-}^{(1)}\,\mathrm{d}t + \mathrm{d}W_t^{(1)} + \mathrm{d}J_t^{\lambda(X),\nu(X(1))}, \quad \mathrm{d}X_t^{(2)} = -X_{t-}^{(2)}\,\mathrm{d}t + \mathrm{d}W_t^{(2)} + \mathrm{d}J_t^{\lambda(X),\nu(X(2))}, \\
&\lambda(X_t) = \sqrt{\left(X_t^{(1)}\right)^2 + \left(X_t^{(2)}\right)^2}, \quad Z \sim \mathrm{U}\left[0, \sqrt{\left(X_t^{(1)}\right)^2 + \left(X_t^{(2)}\right)^2}\right], \\
&f_\nu\left(X_t^{(1)}\right) = -\cos\left(\arctan\left(X_t^{(2)}/X_t^{(1)}\right)\right) \cdot Z, \quad f_\nu\left(X_t^{(2)}\right) = -\sin\left(\arctan\left(X_t^{(2)}/X_t^{(1)}\right)\right) \cdot Z
\end{aligned} \tag{8.17}$$

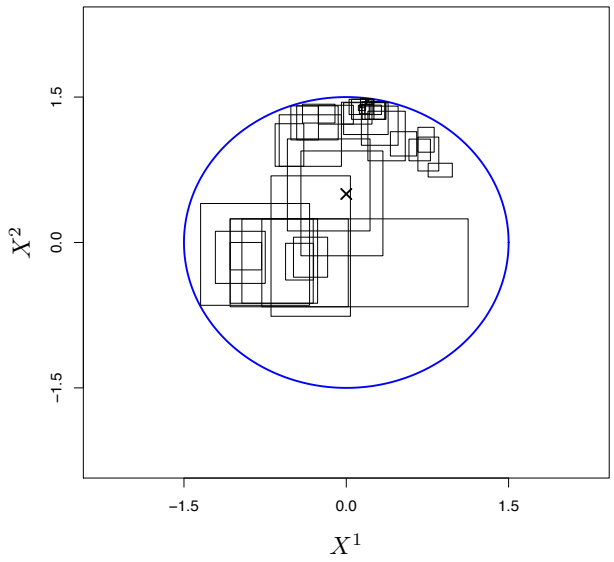
determining whether or not they cross the following circular barriers,

$$x^2 + y^2 = r, \qquad \text{where } r = \{0.8, 1, \ldots, 2.8, 3\}. \tag{8.18}$$

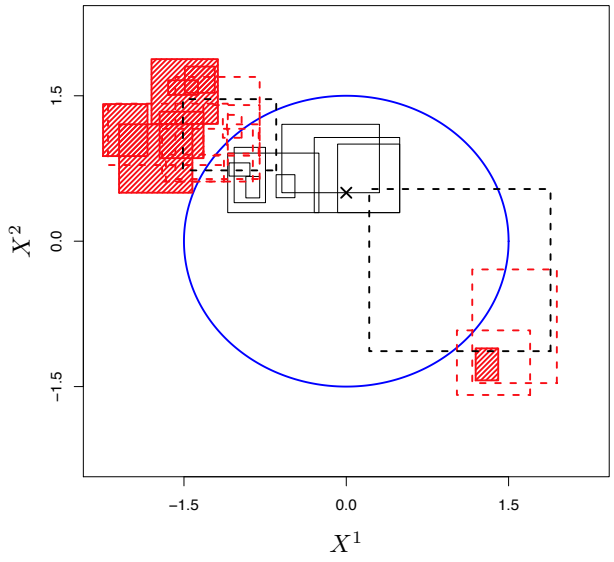
Intuitively, the compound Poisson process component of the jump diffusion in (8.17) can be interpreted as having increasing jump intensity as the jump diffusion moves away from the origin, however, if a jump occurs then the jump diffusion moves to a point uniformly distributed between the origin and its current location.

The jump intensity of this SDE (8.17) can't be bounded so we simulate sample paths using the AUJEA (see Algorithm 5.3.3). In Figure 8.3.5 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 (8.17) and by applying a modified version of Algorithm 8.3.1.

In this example we simulated 50000 sample paths from the measure induced by (8.17), determining for each circular barrier (8.18) whether or not it was crossed. In addition, for each circular barrier we simulated the time within an interval of length $\epsilon \leq 10^{-3}$ 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 a common collection of sample paths ensures that the calculated probabilities retain any natural ordering (for instance, the first crossing time of a circular barrier of a given radius must occur before one of larger radius). In Figures 8.3.6 and 8.3.7 we present various results obtained from our simulations which may be of interest in any given application.

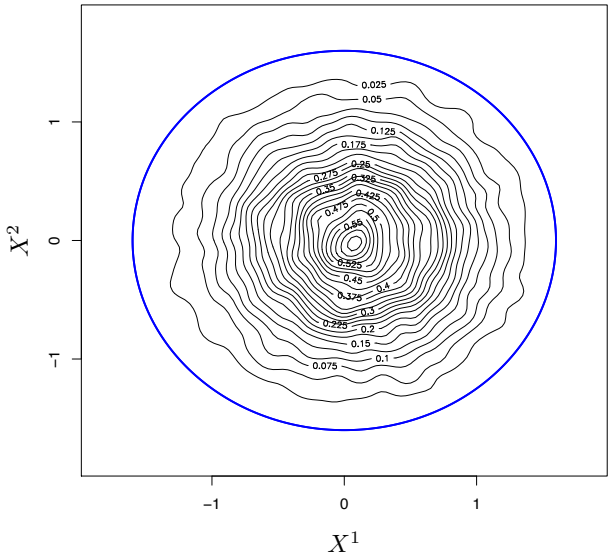


(a) No barrier crossing

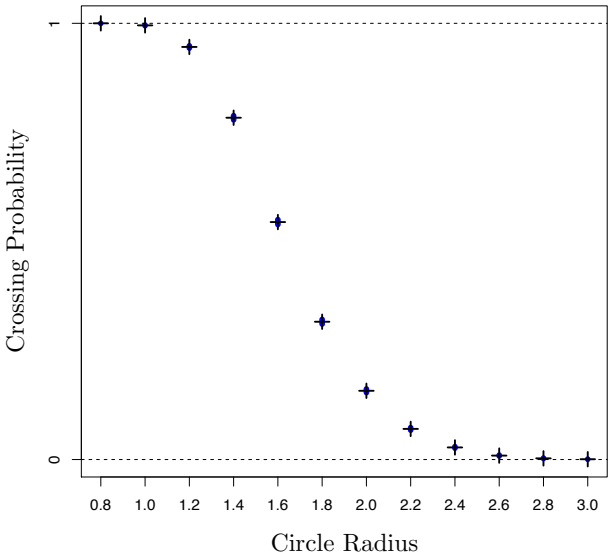


(b) Barrier crossed

Figure 8.3.5: 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.

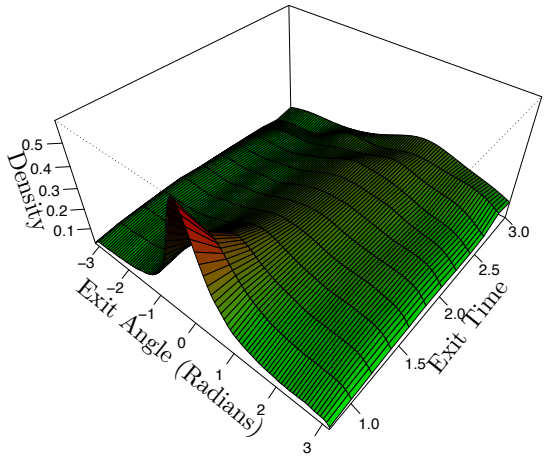


(a) Contour plot of kernel density estimate of killed diffusion transition density with circular barrier of radius 1.6.

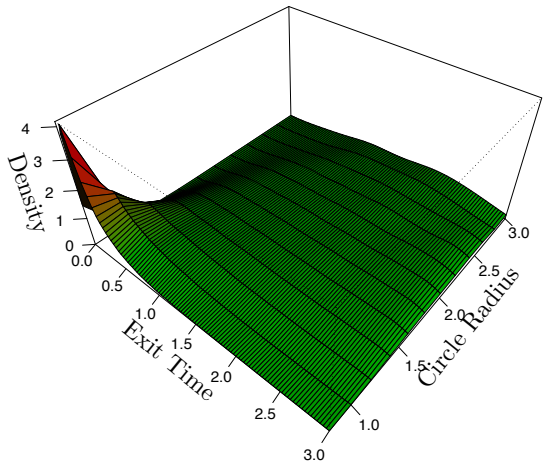


(b) Empirical probabilities of crossing centred circles of increasing radius (indicated by crosses) using a common collection of sample paths, overlaid with 95% Clopper-Pearson confidence intervals (indicated by blue bar).

Figure 8.3.6: 2-D jump diffusion with circular barrier example: Figures computed using 50000 sample paths.



(a) Circle of radius 1.6 exit angle by exit time (crossing time and exit angle evaluated within intervals of length $\epsilon \leq 10^{-3}$ and $\theta \leq 10^{-3}$ respectively).



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

Figure 8.3.7: 2-D jump diffusion with circular barrier example: Exit point figures (computed using a common collection of 50000 sample paths).

9 Concluding Remarks

"I hope we'll be able to solve these problems before we leave."

— Paul Erdős

In this thesis we have developed exact algorithms for simulating finite dimensional representations of (jump) diffusion and (jump) diffusion bridge sample paths over finite intervals, in such a way that the sample path can be restored at any desired finite collection of time points (Chapters 5 and 6).

In Chapter 7 we presented methodology for particle filters with jump diffusion latent transition density, and in Chapter 8 we presented an exact and methodologically extended version of ϵ-strong simulation for (jump) diffusions and (jump) diffusion bridges.

Finally, in Section 8.3 we presented a novel approach for determining (with finite computation) whether or not (jump) diffusion and (jump) diffusion bridge sample paths cross various irregular and non-trivial barriers, simulate to any specified tolerance the first hitting time of the barrier by a sample path, and simulate killed diffusion sample paths.

9.1 Future Directions

We believe there is considerable scope in investigating practical applications of the methodology developed in Section 8.3 (whereby we determine whether or not (jump) diffusions cross various irregular barriers). In particular, the methodology should be directly applicable in a flexible manner to tackling classical Dirichlet problems (see [Mörters and Peres, 2010, Chap. 4] for a discussion on Brownian motion and Dirichlet problems)

and problems in rare event simulation (see for instance Cérou and Guyader [2007] and Lagnoux-Renaudie [2009]). Other avenues of possible direct research for this methodology lie in finance (for instance, option pricing) and survival modelling.

One possible methodological application of Section 8.3 (and more generally of Chapter 8) is to address the time scaling issues of exact algorithms for simulating diffusion and jump diffusion bridges. In particular, as the exact algorithms are rejection based schemes the computational cost of simulating a diffusion sample path does not scale linearly with the time interval with which it has to be simulated over (which can be noted as a consequence of the form of the acceptance probability, recalling from Section 5.2 that $X \sim \mathbb{W}_{0,T}^{x,y}$ with acceptance probability $P_{\mathbb{W}_{0,T}^{x,y}}(X) = \exp\{-\int_0^T \phi(X_s)\,\mathrm{d}s\} \cdot \exp\{\Phi T\}$), rendering the approach impractical for very large time intervals. In the case of diffusions with unconditioned end point the Markov property can be exploited to break the interval into a number of smaller intervals. The sample path can then be simulated by concatenating a simulated diffusion sample path over each interval. Currently the choice of interval size is not well understood, and so one possible avenue of research is how to determine the 'optimal' size. In the case of conditioned diffusions this approach can not be taken, however, a recent novel time discretisation approach introduced by Bladt and Sørensen [Forthcoming], proposes the simulation and merging of two unconditioned diffusion sample paths, one simulated forwards in time and one backwards in time from the end point. If it were possible to adapt such an approach to fit within the exact algorithm framework then this would address the time scaling issues of (jump) diffusion bridges.

Another possible methodological application of Section 8.3 is to utilise the almost sure bounds the methodology provides on the first hitting time of an irregular barrier by a (jump) diffusion, in order to construct a retrospective algorithm to unbiasedly simulate the first hitting time with finite expected computation.

Recent exact algorithm methodology for simulating jump diffusion bridges has been used to extend MCMC methodology (see for instance Gonçalves [2011] and Gonçalves and Roberts [2013]), so one possible area of research would be to similarly extend the methodology developed within this thesis (in particular Chapters 5 and 6). Possible applications of such an extension include electricity pricing (see for instance Culot et al. [2006]).

A natural avenue of research for the particle filtering methodology in Chapters 3 and

7 is to investigate the smoothing and prediction problems (see Sections 3.1.3 and 3.1.2 respectively) in the context of HMMs with diffusion latent transition density. To motivate this note that the methodology developed in this thesis provides a finite dimensional representation of entire diffusion sample paths, and so in this particular setting each particle in the particle set contains rich information on the trajectory of the entire latent process.

Finally we note that the class of processes for which the exact algorithm methodology developed within this thesis can be applied is limited by the conditions imposed in Section 1.3. Clearly it would desirable to develop methodology enabling the simulation of sample path skeletons from a broader class of processes, including diffusions and jump diffusions with discontinuous drift and stochastic volatility models among others.

Part III

Appendices & Bibliography

A Elementary Cauchy Sequence Functions

"I can believe anything provided it's incredible."

— Oscar Wilde

In Section 6.1 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 thesis. In Section 6.3.2 we exploit the full representation of ρ found in Theorem 6.1.4 and Definition 6.1.2 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 Section 6.1 and Section 6.1.2, note that in addition to functions listed below there are also the corresponding negative versions, a set for each of the possible layer combinations ($[\ell\downarrow, \upsilon\uparrow]$, $[\ell\uparrow, \upsilon\uparrow]$, $[\ell\downarrow, \upsilon\downarrow]$ and $[\ell\uparrow, \upsilon\downarrow]$) as well as various multiples of these functions. Denoting $D := |\upsilon_i - \ell_i|$ and

$M := (\ell_i \wedge \upsilon_i)$ we have,

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

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

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

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

B Bisections & Dissections

"To live effectively is to live with adequate information."

— Norbert Wiener

In Section 6.3.3 we noted that if we consider sample paths which belonged to the set $D_*|(W_q = w)$, where $q \in (s,t)$ and,

$$D_* := \left(\left\{W_{[s,t]} : \hat{m}_{s,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow}_{s,t}\right]\right\} \bigcap \left\{W_{[s,t]} : \check{m}_{s,t} \in \left[\upsilon^{\downarrow}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right), \tag{B.1}$$

then any given sample path must belong to one of 9 possible disjoint sets $D_*|(W_q = w) = \uplus_{i=1}^{9} B_i$ (so called *bisections*). In Section 6.3.3 we provided a graphical illustration of the bisections (which we reproduce here for convenience in Figure B.0.1). In this appendix we formally list the sets $B_1 - B_9$, which are as follows,

$$\begin{aligned}
B_1 := &\left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \\
&\bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \qquad \text{(B.2)} \\
B_2 := &\left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \\
&\bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\uparrow *}_{s,t}, (w \wedge y)\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \qquad \text{(B.3)} \\
B_3 := &\left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\uparrow *}_{s,t}, (x \wedge w)\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \\
&\bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \qquad \text{(B.4)} \\
B_4 := &\left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \\
&\bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[(w \vee y), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right). \qquad \text{(B.5)}
\end{aligned}$$

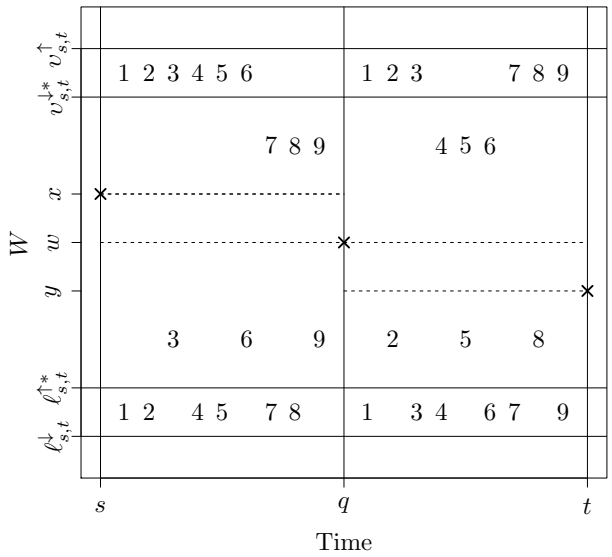


Figure B.0.1: Illustration of 9 possible (disjoint) bisections.

$$B_5 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\uparrow *}_{s,t}, (w \wedge y)\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[(w \vee y), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right). \quad \text{(B.6)}$$

$$B_6 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\uparrow *}_{s,t}, (x \wedge w)\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right) \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[(w \vee y), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right). \quad \text{(B.7)}$$

$$B_7 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[(x \vee w), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right) \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \quad \text{(B.8)}$$

$$B_8 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[(x \vee w), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right) \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\uparrow *}_{s,t}, (w \wedge y)\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \quad \text{(B.9)}$$

$$B_9 := \left(\left\{W_{[s,q]} : \hat{m}_{s,q} \in \left[\ell^{\uparrow *}_{s,t}, (x \wedge w)\right]\right\} \bigcap \left\{W_{[s,q]} : \check{m}_{s,q} \in \left[(x \vee w), \upsilon^{\downarrow *}_{s,t}\right]\right\}\right) \bigcup \left(\left\{W_{[q,t]} : \hat{m}_{q,t} \in \left[\ell^{\downarrow}_{s,t}, \ell^{\uparrow *}_{s,t}\right]\right\} \bigcap \left\{W_{[q,t]} : \check{m}_{q,t} \in \left[\upsilon^{\downarrow *}_{s,t}, \upsilon^{\uparrow}_{s,t}\right]\right\}\right). \quad \text{(B.10)}$$

Bibliography

Y. Aït-Sahalia. Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach. *Econometrica*, 70(1):223–262, 2002.

Y. Aït-Sahalia. Closed-form likelihood expansions for multivariate diffusions. *The Annals of Statistics*, 36:906–937, 2008.

H. Akashi and H. Kumamoto. Random sampling approach to state estimation in switching environments. *Automatica*, 13:429–434, 1977.

B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, 1st edition, 1979.

T.W. Anderson. A modification of the sequential probability ratio test to reduce the sample size. *Annals of Mathematical Statistics*, 31(1):165–197, 1960.

C. Andrieu and G.O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

S. Asmussen, P. Glynn, and J. Pitman. Discretization error in simulation of one-dimensional reflecting Brownian motion. *Annals of Applied Probability*, 5(4):875–896, 1995.

S. Banach and H. Steinhaus. Sur le principe de la condensation de singularités. *Fundamenta Mathematicae*, 9(1):50–61, 1927.

O.E. Barndorff-Nielsen and N. Shephard. Power and bi-power variation with stochastic volatility and jumps. *Journal of Financial Econometrics*, 2(1):1–37, 2004.

A. Beskos and G.O. Roberts. An exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444, 2005.

A. Beskos, O. Papaspiliopoulos, and G.O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12:1077–1098, 2006a.

A. Beskos, O. Papaspiliopoulos, G.O. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 68(3):333–382, 2006b.

A. Beskos, O. Papaspiliopoulos, and G.O. Roberts. A factorisation of diffusion measure and finite sample path constructions. *Methodology and Computing in Applied Probability*, 10:85–104, 2008.

A. Beskos, S. Peluchetti, and G.O. Roberts. ϵ-strong simulation of the Brownian path. *Bernoulli*, 18(4):1223–1248, 2012.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

M. Bladt and M. Sørensen. Simple simulation of diffusion bridges with application to likelihood inference for diffusions. *Bernoulli*, Forthcoming.

Z.A. Burq and O. Jones. Simulation of Brownian motion at first passage times. *Mathematics and Computers in Simulation*, 77:64–71, 2008.

O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models.* Springer, 1st edition, 2005.

J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. *IEEE Proceedings - Radar, Sonar and Navigation*, 146:2–7, 1999a.

J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets. Technical report, Department of Statistics, University of Oxford, 1999b.

B. Casella and G.O. Roberts. Exact simulation of jump-diffusion processes with Monte Carlo applications. *Methodology and Computing in Applied Probability*, 13(3):449–473, 2010.

G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007.

N. Chen and Z. Huang. Localisation and exact simulation of Brownian motion driven stochastic differential equations. *Mathematics of Operational Research.*

Y. Chen. Another look at rejection sampling through importance sampling. *Statistics and Probability Letters*, 72:277–283, 2005.

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

D.R. Cox and V. Isham. *Point Processes.* Chapman and Hall, 1st edition, 1980.

M. Culot, V. Goffin, S. Lawford, S. de Menten, and Y. Smeers. An affine jump diffusion model for electricity. In *Seminars, Groupement de Recherche en Économie Quantitative d'Aix-Marseille*, 2006.

D. Dachuna-Castelle and D. Florens-Zmirou. Estimation of the coefficients of a diffusion from discrete observations. *Stochastics*, 19:263–284, 1986.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods.* Springer, 2nd edition, 2003.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure.* Springer, 2nd edition, 2008.

M.H.A. Davis and R.B. Vinter. *Stochastic Modelling and Control.* Chapman and Hall, 1st edition, 1985.

J.F.G. de Freitas, M.A. Niranjan, A.H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.

P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications.* Springer, 1st edition, 2004.

L. Devroye. Generating the maximum of independent identically distributed random variables. *Computers and Mathematics with Applications*, 6:305–315, 1980.

L. Devroye. *Non-Uniform Random Variate Generation.* Springer, 1st edition, 1986.

R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September 2005.

A. Doucet and A.M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *Oxford Handbook of Nonlinear Filtering.* Oxford University Press, 1st edition, 2011.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice.* Springer, 1st edition, 2001.

O. Elerian. A note on the existence of a closed form conditional transition density for the Milstein scheme. Economics Series Working Papers 1998-W18, University of Oxford, Department of Economics, 1998.

B. Eraker, M. Johannes, and N. Polson. The impact of jumps in volatility and returns. *The Journal of Finance*, 58(3):1269–1300, 2003.

P. Fearnhead, O. Papaspiliopoulos, and G.O. Roberts. Particle filters for partially-observed diffusions. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 70(4): 755–777, 2008.

P. Fearnhead, O. Papaspiliopoulos, G.O. Roberts, and A.M. Stuart. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 72(4):497–512, 2010.

J. Geweke. Bayesian inference in econometrics models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, 1989.

K. Giesecke and D. Smelov. Exact sampling of jump-diffusions. *Operations Research*, Forthcoming.

G. Goertzel. Quota sampling and importance functions in stochastic solution of particle problems. Technical Report ORNL-434, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1949.

A. Golightly and D.J. Wilkinson. Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, 16(4):323–338, 2006.

A. Golightly and D.J. Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693, 2008.

F.B. Gonçalves. *Exact simulation and Monte Carlo inference for jump-diffusion processes.* PhD thesis, Department of Statistics, University of Warwick, 2011.

F.B. Gonçalves and G.O. Roberts. Exact simulation problems for jump-diffusions. *Methodology and Computing in Applied Probability*, 2013.

N. Gordon, J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

P.R. Halmos. *Measure Theory.* Springer-Verlag, 1st edition, 1974.

J.M. Hammersley and K.W. Morton. Poor Man's Monte Carlo. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 16(1):23–38, 1954.

J.E. Handschin and D.Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *Journal of Control*, 9:547–559, 1969.

E.J. Hannan and M. Deistler. *The Statistical Theory of Linear Systems.* John Wiley, 1st edition, 1988.

T. Higuchi. Monte Carlo filter using the genetic algorithm operators. *Journal of Computational and Graphical Statistics*, 59(1):1–23, 1997.

G. Hovland and B. McCarragher. Hidden Markov models as a process monitor in robotic assembly. *International Journal of Robotics Research*, 17(2):153–168, 1998.

S.M. Iacus. *Simulation and Inference for Stochastic Differential Equations: With R Examples.* Springer, 1st edition, 2008.

J. Jacod and P. Protter. *Discretization of Processes.* Springer, 1st edition, 2012.

A.M. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics and Probability Letters*, 78(12):1498–1504, 2008.

S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

H. Kahn. Stochastic (Monte Carlo) attenuation analysis. Technical Report P-88 A, The RAND Corporation, Santa Monica, California, 1949.

R.E. Kálmán. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering, Series D*, 82:35–45, 1960.

R.E. Kálmán and R. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering, Series D*, 83:95–108, 1961.

I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus.* Springer-Verlag, New York, 2nd edition, 1991.

J.F.C. Kingman. *Poisson Processes.* Clarendon Press, 1st edition, 1992.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer, 4th edition, 1992.

A. Kong. A note on importance sampling using renormalized weights. Technical report, Department of Statistics, University of Chicago, 1992.

A. Kong, S. Liu, and W. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

A. Krogh, B. Larsson, G. von Heijne, and E. Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580, 2001.

H.R. Künsch. Recursive Monte Carlo filters: Algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021, 2005.

A. Lagnoux-Renaudie. A two-step branching splitting model under cost constraint for rare event analysis. *Journal of Applied Probability*, 46(2):429–452, 2009.

J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

S. Mamon and R. Elliot. *Hidden Markov Models in Finance.* Springer, 1st edition, 2007.

X. Mao and C. Yuan. *Stochastic differential equations with Markovian switching.* Imperial College Press, 2006.

G. Maruyama. Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4(1):48–90, 1955.

S. Maskell and N. Gordon. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.

R. Meinhold and N. Singpurwalla. Understanding the Kalman filter. *Journal of the American Statistical Association*, 37(2):123–127, 1983.

R.C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1):125–144, 1976.

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

G.N. Milstein. A method of second-order accuracy integration of stochastic differential equations. *Theory of Probability & Its Applications*, 23(2):396–401, 1979.

G.N. Milstein and M.V. Tretyakov. Simulation of a space-time bounded diffusion. *The Annals of Applied Probability*, 9(3):732–779, 1999.

P. Mörters and Y. Peres. *Brownian motion.*, volume 30. Cambridge University Press, 2010.

J. Nicolau. A new technique for simulating the likelihood of stochastic differential equations. *Econometrics Journal*, (1):91–103, 2002.

B. Øksendal. *Stochastic Differential Equations.* Springer, 6th edition, 2007.

B. Øksendal and A. Sulem. *Applied Stochastic Control of Jump Diffusions.* Springer, 2nd edition, 2004.

U. Picchini, A. Gaetano, and S. Ditlevsen. Stochastic differential mixed-effects models. *Scandinavian Journal of Statistics*, 37(1):67–90, 2009.

M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

M. Pitt and N. Shephard. Auxiliary variable particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer, 1st edition, 2001.

E. Platen and N. Bruti-Liberati. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. Springer, 1st edition, 2010.

K. Pötzelberger and L. Wang. Boundary crossing probability for Brownian motion and general boundaries. *Journal of Applied Probability*, 34:54–65, 1997.

K. Pötzelberger and L. Wang. Boundary crossing probability for Brownian motion. *Journal of Applied Probability*, 38:152–164, 2001.

B.D. Ripley. *Stochastic Simulation*. John Wiley, 1st edition, 1987.

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

L.C.G. Rogers and D. Williams. *Diffusions, Markov Processes and Martingales, Vol. 1*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2nd edition, 2000.

M.N. Rosenbluth and A.W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics*, 23(2):356–359, 1955.

I. Shoji. *Estimation and inference for continuous time stochastic models*. PhD thesis, Institute of Statistical Mathematics, Tokyo, 1995.

I. Shoji. Approximation of continuous time stochastic processes by a local linearization method. *Mathematics of Computation of the American Mathematical Society*, 67(221):287–298, 1998.

I. Shoji and T. Ozaki. Comparative study of estimation methods for continuous time stochastic processes. *Journal of Time Series Analysis*, 18(5):485–506, 1997.

I. Shoji and T. Ozaki. A statistical method of estimation and simulation for systems of stochastic differential equations. *Biometrika*, 85(1):240–243, 1998.

E. Slutsky. *Über stochastische asymptoten und grenzwerte*. Amministrazione del "Metron", 1925.

H. Tanizaki. Nonlinear filters: Estimation and applications. In *Lecture Notes in Economics and Mathematical Systems, Vol. 400*. Springer, 1993.

H. Tanizaki and R.S. Mariano. Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration. *Journal of Applied Econometrics*, 9:163–179, 1994.

P. Vidoni. Exponential family state space models based on a conjugate latent process. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 61(1):213–221, 1999.

J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards – Applied Mathematics Series*, 12:36–38, 1951.

M.A. Woodbury. Inverting modified matrices. Technical Report 42, Statistical Research Group, Princeton University, Princeton, N.J., 1950.

V.S. Zaritskii, V.B. Svetnik, and L.I. Shimelevich. Monte Carlo technique in problems of optimal data processing. *Automation and Remote Control*, 12:95–103, 1975.

"We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance." — John Archibald Wheeler