WARWICK
THE UNIVERSITY OF WARWICK

A Thesis Submitted for the Degree of PHD at the University of Warwick

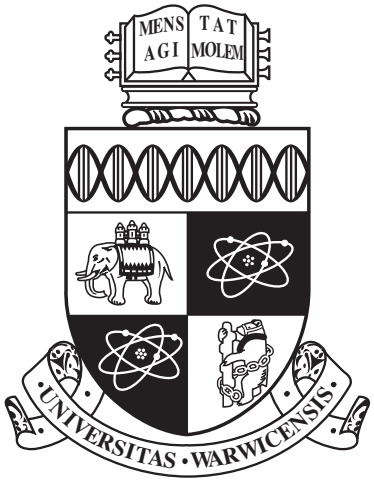
Permanent WRAP URL:

http://wrap.warwick.ac.uk/172475

Copyright and reuse:
This thesis is made available online and is protected by original copyright.
Please scroll down to view the document itself.
Please refer to the repository record for this item for information to help you to cite it.
Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

warwick.ac.uk/lib-publications



Exact and unbiased simulation of rare events

by

James Hodgson

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Statistics

January 2022

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgments

My deepest thanks to my supervisors Adam Johansen and Murray Pollock, for their unfailing generosity of time, energy and advice in guiding me through my PhD. Likewise, to my examiners Paul Jenkins and Nikolas Kantas for an engaging discussion and for suggesting many thoughtful improvements.

Thanks also to my friends on the OxWasp program, in the Warwick statistics department, and scattered over many other locations for the interesting discussions and good memories; and to HF, for arriving at the right time.

Finally, deep thanks to my family for their unwavering support and encouragement.

James Hodgson, June 2022.

Declarations

I declare that this thesis is the product of my own work and research. This thesis has not been submitted for examination to any institution other than the University of Warwick.

Chapter 4 of this thesis is based on the article *Unbiased Simulation of Rare Events in Continuous Time*, published in Methodology and Computing in Applied Probability, November 2021, and authored jointly with my supervisors Adam M. Johansen and Murray Pollock.

Abstract

Rare event estimation is the problem of quantifying how unlikely is the occurrence of an event which is already known to be unlikely. Rare event problems are found in many areas of study, including finance, chemistry, biology, and physics (a foundational problem for the field was the *neutron shielding problem* in physics: how likely is a particle to cross a material barrier without being absorbed or deflected?).

The type of rare event problem this thesis is concerned with is that of determining the probability that a continuous-time Markov process hits a certain known set before a certain stopping time. Numerous algorithms exist for this class of problems, requiring sample paths of the Markov process. Many of these algorithms are known to produce unbiased estimates for the rare event probability, and sometimes considerable effort has been spent to establish that they do so (for example, Bréhier et al. [2016]).

Many continuous-time Markov processes, such as diffusion processes, must be discretised in time to be simulated on a computer. But usually, access to a complete, continuous sample-path of the process is *necessary* to determine the course of a rare event algorithm. For the discretised samples which are usually used in practice then, the resulting rare event estimator cannot be guaranteed to be unbiased, although this is rarely acknowledged explicitly.

Recent work in the exact and ε-strong simulation of diffusions, and in unbiased inference for diffusions, seems to suggest solutions to this problem is some contexts. The contribution of this thesis will be to show how this synthesis can be carried out, and investigate the effectiveness of the resulting algorithms.

Chapter 1

Introduction

This thesis is concerned with developing and investigating methods for the exact and unbiased simulation and estimation of rare events for continuous-time processes. A *rare event* is one which has a very low probability of occurrence. These events are challenging to conduct inference on using conventional Monte Carlo methods, since it typically requires prohibitively many samples to observe a single instance of such an event.

Inference is further complicated in the continuous-time setting, since even when computationally feasible algorithms are available, theoretical guarantees of unbiasedness are typically justified by continuous-time theory, whereas the corresponding numerical simulations are usually carried out using time-discretised numerical approximations. Such methods introduce an approximation error whose size is difficult to quantify, which can be an especially sensitive issue when a very small probability is involved.

Several techniques are available for efficient inference in the rare event setting, many of which fall into the framework of Sequential Monte Carlo methods, or are closely related to this framework. Roughly speaking, these methods improve on conventional Monte Carlo simulation by systematically selecting and promoting simulated trajectories which are heading towards the rare event, and demoting those which do not. In this way, computational effort is concentrated on trajectories for which the rare event is in fact observed. When the corresponding statistical estimator is finally constructed, the weighting towards these favoured trajectories is accounted for "post hoc". We consider several methods in this class in this thesis.

On the other hand, methods of simulating continuous-time processes which mitigate or avoid the bias and error introduced by conventional time-discretisation methods have received wide attention recently. Unbiasedness is an attractive property for several reasons. For example, it removes the effort of analysing the magnitude of the approximation error in an approximate scheme. It is also possible to run unbiased estimators in parallel and average them to obtain a lower-variance estimator for a given computing time.

Many approaches to achieving unbiased estimation for continuous-time processes have been considered. One can attempt to devise methods to simulate without approximation the finite-dimensional marginals of a diffusion process, leading to (for example) the *exact* methods following Beskos and Roberts [2005], Beskos et al. [2006]. Exact methods have been successfully combined with SMC algorithms in the context of particle filtering in the work of Fearnhead et al. [2008].

To estimate functionals which depend on more than finite-dimensional marginals - for example, functionals of a whole continuous-time path - it may be useful to simulate further information about the process, as with the *ε-strong* methods for diffusions and other processes following Beskos et al. [2012]; Pollock et al. [2016].

Alternatively, one might accept the convenience of numerical discretisation while attempting to minimise the resulting error for a given computational cost, leading to the multilevel Monte Carlo methods following Heinrich [2001], Giles [2008]. These methods, and other biased methods, may sometimes be de-biased entirely, for example following the randomisation methods of McLeish [2011]; Rhee and Glynn [2012].

In this thesis we consider the application of appropriate unbiased and exact simulation methods to certain rare event estimation methods. *Multilevel splitting*, a popular algorithm for estimating the probability of rare events, is formulated as a sequence of problems of deciding whether a process has crossed a given barrier. Since deciding whether a process has crossed a barrier depends on its whole path rather than finite dimensional marginals, the class of ε-strong methods are an attractive proposition for removing approximation error in this setting.

On the other hand, the large-deviations inspired genealogical particle method of Del Moral et al. [2015] for estimating rare event probabilities has a less discrete internal structure: it is a Sequential Monte Carlo method for which particles closer to the rare event are strongly promoted for resampling. A binary assessment of whether the process has reached the rare event is made only at the end of the simulation process. In this case, we consider the possibility of a de-biased Multilevel Monte Carlo algorithm, whose variance can then be quantified (the corresponding analysis would be more challenging in the splitting setting).

1.1 Thesis structure

The main example of continuous-time processes considered in this thesis are diffusion processes. Chapter 2 provides a brief overview of this class of processes, together with a short overview of conventional numerical discretisation techniques and their properties. The rest of the chapter is devoted to ε-strong methods, which are presented first in an abstract framework, followed by several detailed examples. We give space for a longer exposition in particular of the method of Blanchet et al. [2017], which uses a different idea from the earlier literature. Rather than concentrating on its technical details, we attempt to motivate and explain its broader construction in relatively elementary

terms.

Chapter 3 introduces first the multilevel splitting algorithm for rare event estimation in a direct fashion, together with several closely-related variants. Subsequently, connections are drawn to the Sequential Monte Carlo (SMC) framework, which provides a substantial body of established results and literature to which appeal can be made. We close the chapter by describing the mathematical framework of Feynman-Kac models, of which SMC algorithms can be considered particle approximations.

In chapter 4, we describe how to combine ε-strong simulation with multilevel splitting and its SMC variant to obtain new implementable exact algorithms, requiring no numerical discretisation of the continuous-time process. We consider both the one-dimensional and multi-dimensional cases, and establish the unbiasedness of the resulting algorithms together with numerical examples in simple settings.

Chapters 5 and 6 consider a different rare event algorithm, and a different method of addressing approximation error. In chapter 5, we introduce the Multilevel Monte Carlo method and its associated de-biased estimator, and describe how to apply it to the rare event setting of Del Moral et al. [2015]. In Chapter 6, we investigate the variance of this estimator, and discuss the computational challenges of implementing it. We finally provide some numerical investigations and discuss following Jasra et al. [2020] how an approximate implementation can still be useful.

Chapter 2

Simulating diffusion processes

2.1 Diffusion processes and their approximations

This thesis is concerned with methods for the unbiased estimation and simulation of rare events for continuous-time processes. The central example of a continuous-time process to which we repeatedly refer is a *diffusion process*. We set down here some basic definitions and properties of this class of processes.

We are also concerned with the problem of approximation error in simulating these processes. Traditionally simulation has been carried out using methods which divide the time dimension into a discrete grid, upon which (approximate) values of the process are simulated. The properties of these *discretisation methods* are of some concern to us in chapters 5 and 6, so we also set down some definitions, simple examples and basic results relating to these schemes.

Recent developments in the *exact* and *ε-strong* simulation of diffusions do away entirely with the issue of approximation error, usually at the cost of a more computationally expensive algorithm. We consider the application of these methods to rare event problems in chapter 4. For this reason, we take the time to describe in some detail a few important examples at the end of this section.

2.1.1 Motivation

Following the treatment of Øksendal [1992], the motivation for developing a theory of stochastic integration (that is, a theory of integrals against random variables) comes from the desirability of solving certain problems which are most naturally formulated as differential equations containing random or noise terms. A fairly general form that such a model might take is

$$dX(t) = a(t, X(t))dt + b(t, X(t))W(t)$$

on the time interval $t \in [0, T]$, where $X(t)$ is the quantity being modelled, a, b are deterministic functions, and $W(t)$ is allowed to be a random variable.

One way to proceed with constructing a solution to this equation is to postulate some desirable properties for W. In particular, the requirement that $W(t)$ is a mean-0, stationary process with $W(s)$ independent of $W(t)$ for $s \neq t$ suggests the following reformulation as a discrete summation on the partition $0 = t_0 < \ldots < t_n = T$: given $\hat{X}(t_0) = x_0$, for $k = 1, \ldots, n$,

$$\hat{X}(t_k) = \hat{X}(t_0) + \sum_{j=1}^{k} a(t_{j-1}, \hat{X}(t_{j-1}))(t_j - t_{j-1}) + \sum_{j=1}^{k} b(t_{j-1}, \hat{X}(t_{j-1}))(B(t_j) - B(t_{j-1})) \tag{2.1}$$

where $B(t)$ is Brownian motion, the unique mean-0, stationary process with independent increments and continuous paths. Here we assume B is defined on the probability space $(\Omega, \mathcal{F})$, and a, b are functions $\mathbb{R}_{\geq 0} \times \Omega \to \mathbb{R}^d$ for some $d > 0$.

As long as a is of bounded variation (see Section 2.1.2 below), one may then take a limit of $\sum_{j=1}^{k} a(t_{j-1}, \hat{X}(t_{j-1}))(t_j - t_{j-1})$ over increasingly fine time grids to get the Riemann integral $\int_0^t a(s, X(s))ds$. The question is then how to take a similar limit for the second term above, and more generally for sums of the form $\sum_{j=1}^{k} b(t_{j-1}, \omega)(B(t_j, \omega) - B(t_{j-1}, \omega)$.

Before settling this question, we begin by recalling some important results from the deterministic theory of integration. These will be directly useful in motivating one of the ε-strong constructions in section 2.2, and are also useful in clarifying by contrast some important properties of integrals against random variables.

2.1.2 Deterministic integration

Let $g : [0, 1] \to \mathbb{R}$ be a function. A *partition* P of $[0, 1]$ is defined from a sequence $0 = t_0 < t_1 < \ldots < t_n = 1$ by taking $P = \{[t_{j-1}, t_j] : j = 1, \ldots, n\}$. The *mesh size* of P is $|P| = \sup_{[s,t] \in P}(t - s) > 0$. Write $\mathcal{P}$ for the set of all partitions of $[0, 1]$. To define integrals, we need a sequence $(P^{(n)})_{n=1}^{\infty}$ of partitions of $[0, 1]$ with mesh $|P^{(n)}| \to 0$. (There is nothing special about the interval $[0, 1]$, which we use for notational convenience.)

Given another function $f : [0, 1] \to \mathbb{R}$, the *Riemann-Stieltjes integral* of f against g is the limit of the following *Riemann sums*, when this limit exists:

$$\int_0^1 f dg := \lim_{n \to \infty} \sum_{[s,t] \in P^{(n)}} f(s)(g(t) - g(s))$$

The following theorem gives the required condition on g for a Riemann-Stieltjes integral to exist for every continuous integrand f:

Theorem 1. *[Burkill and Burkill, 1970, Section 6.5] Say that g is of* bounded variation

if $\sup_{P\in\mathcal{P}}\sum_{[s,t]\in P}|g(t)-g(s)|<\infty$. *Then the Riemann sums*

$$S_n=\sum_{[s,t]\in P^{(n)}} f(s)(g(t)-g(s))$$

converge to a limit independent of $(P^{(n)})$ *for every continuous* f *if and only if* g *is of bounded variation.*

Informally speaking, the requirement that g is of bounded variation is a strong restriction on the "roughness" of g. Many functions against which one would like to define integrals are not of bounded variation. In particular, it is well-known that paths of Brownian motion are (almost surely) not of bounded variation (see for example [Mörters and Peres, 2010, Theorem 1.35]), so it is not possible to define integrals against Brownian paths as Riemann-Stieltjes integrals for all continuous f.

One way to try to get past the above result is to weaken the requirement that we are able to integrate all continuous f against g. A classical result in this line is a theorem of Young. For a paramater $\alpha>0$, say f is *α-Hölder continuous* if $|f(s)-f(t)|<C|s-t|^\alpha$ for some constant C, and all $s,t\in[0,1]$.

Theorem 2. *Young [1936] Let* $f,g:[0,1]\to\mathbb{R}$ *be* α,β*-Hölder continuous respectively, with* $\alpha+\beta>1$, *and* $(P^{(n)})$ *as above. Then the Riemann sums*

$$S_n=\sum_{[s,t]\in P^{(n)}} f(s)(g(t)-g(s)) \tag{2.2}$$

converge to a limit independent of $(P^{(n)})$. *The* Young integral $\int_0^1 f dg$ *may then be defined as the limit as* $n\to\infty$ *of the sums* S_n.

It is immediately clear from this definition that the Young integral coincides with the Riemann integral of f against g when the latter exists. For the particular case with f,g both α-Hölder continuous, this means that the integral of f against g can be constructed if $\alpha>1/2$. Brownian motion just misses this condition, being α-Hölder continuous for all $\alpha<1/2$ ([Mörters and Peres, 2010, Corollary 1.20]). The Young integral is therefore inadequate for defining the integral of a Brownian path against itself, which is too restrictive for our purpose.

2.1.3 The Ito integral and stochastic differential equations

A further step to get around Theorem 1 is to weaken the required mode of convergence, as well as restricting the class of integrands. The *Ito integral* against Brownian motion may be constructed as a limit *in* L^2 of Riemann sums as follows.

Let $\mathcal{B}$ be the Borel σ-algebra on $\mathbb{R}$, and let B be Brownian motion in $\mathbb{R}$ defined on the probability space $(\Omega,\mathcal{F})$. For all $t\in[0,1]$, let $\mathcal{F}_t$ be the sigma algebra generated by $\{B(s):\ s\le t\}$. Say $H:[0,\infty)\times\Omega\to\mathbb{R}$ is *adapted* to $(\mathcal{F}_t)_{t=0}^1$ if the maps $\omega\to H(t,\omega)$ are $\mathcal{F}_t$-measurable for all $t\in[0,1]$.

Theorem 3. *[Øksendal, 1992, Chapter 3] Suppose that H is adapted, that $(t,\omega) \to H(t,\omega)$ is $\mathcal{B} \times \mathcal{F}$-measurable for each $t \in [0,1]$, and that $\mathbb{E}\int_0^1 H(t,\omega)^2 dt < \infty$ for all $\omega \in \Omega$. Then there is a sequence of functions*

$$\phi_n(t,\omega) = \sum_j e_{n,j}(\omega)\mathbb{1}_{[t_j^n, t_{j+1}^n)}(t)$$

satisfying

$$\mathbb{E}\int_0^1 (H(t,\omega) - \phi_n(t,\omega))^2 dt \to 0 \text{ as } n \to \infty,$$

and for these ϕ_n it holds that

$$\sum_j e_{n,j}(\omega)(B(t_{j+1}^n,\omega) - B(t_j^n,\omega)) \tag{2.3}$$

converges in L^2 as $n \to \infty$.

The *Ito integral* $\int_0^1 H(s)dB(s)$ of H against B is then defined to be this L^2 limit. In a sense, one gets around the problem of pointwise convergence of the Riemann sums by defining the limiting random variable on all of Ω at once.

The Ito definition may be further extended to a wider class of functions H than that specified in this result (see [Øksendal, 1992, Chapter 3] for details). One important point is that in the Riemann sums specified in Theorem 3, $H(s)$ cannot be replaced by $H(r)$ for any $r \in (s,t]$ without altering value of the limit, and so changing the definition of the integral. This is one respect in which the Ito integral fundamentally differs from the Riemann integral.

Armed with the Ito integral, we can then define the following *stochastic differential equation* (SDE) against Brownian motion

$$dX(t) = a(t, X(t)) + b(t, X(t))dB(t) \tag{2.4}$$

given the initial value $X(0) = X(0)$ as equivalent to the integral equation

$$X(t) = X(0) + \int_0^t a(s, X(s))ds + \int_0^t b(s, X(s))dB(s),$$

where the final integral is an Ito integral. The functions a and b are usually known as the *drift* and *volatility* respectively.

Sufficient conditions on a, b and $X(0)$ for the existence of a unique solution X with continuous sample paths to this equation may be found in [Øksendal, 1992, Theorem 5.2.1]. We assume the existence of such a solution throughout. Of particular interest are SDEs whose drift and volatility do not depend explicitly on the time parameter:

$$dX(t) = a(X(t)) + b(X(t))dB(t). \tag{2.5}$$

A solution X to such an equation is called a *diffusion process* (or an *Ito diffusion process* if it is necessary to emphasise the choice of $H(s)$ in the Riemann sums).

An important identity for the Ito integral is *Ito's formula* (or *Ito's lemma*), which we state below in its less common integral form (we reference it only once in section 2.2.3, using this form):

Theorem 4. *[Kloeden and Platen, 2013, Theorem 3.3.2] Let X be the Ito solution to the stochastic differential equation (2.4), and $g(t,x)$ a function $g : \mathbb{R}_{\geq 0} \times \mathbb{R} \to \mathbb{R}$ continuously differentiable in t and twice continuously differentiable in x. Then writing $Y(s) = g(s, X(s))$,*

$$Y(t) - Y(0) = \int_0^t \left(\frac{\partial g(s, X(s))}{\partial s} + a(s, X(s)) \frac{\partial g(s, X(s))}{\partial x} + \frac{1}{2} b^2(s, X(s)) \frac{\partial^2 g(s, X(s))}{\partial x^2} \right) ds + \int_0^t \frac{\partial g(s, X(s))}{\partial x} b(s, X(s)) dB(s).$$

By rearranging, this makes it possible to write an Ito integral against Brownian motion in terms of Riemann integrals.

A fact which we rely on heavily throughout is that an Ito diffusion X is a *Markov process*: the future behaviour of X conditional on its history up to time t depends on that history only through $X(t)$. A formal statement of this fact is the following:

Theorem 5. *[Øksendal, 1992, Theorem 7.1.2] Let f be a bounded Borel function $\mathbb{R} \to \mathbb{R}$, and let $\mathbb{E}^x$ denote expectation with respect to the measure defined by the SDE (2.5) with $X_0 = x$. Then*

$$\mathbb{E}^x \left[f(X(t+h)) | \mathcal{F}_t \right](\omega) = \mathbb{E}^{X(t,\omega)} \left[f(X(h)) \right].$$

Finally, we note that the integration theory for deterministic and stochastic integrals in dimension one can of course be extended to theories over $\mathbb{R}^d$ for any integer d, which is covered in the sources cited earlier in this section. We take this for granted throughout.

2.1.4 Numerical schemes

A diffusion process X is an infinite-dimensional random variable. That is, for each time t in some interval $[0, T]$ it takes a value $X(t)$. This means it does not (except in trivial cases) admit a finite-dimensional representation, and so it cannot be fully simulated or stored on a computer. In order to carry out simulations relating to these processes, it is usual to make a finite-dimensional approximation by dividing $[0, T]$ into a regularly spaced grid $0 = t_0 < \ldots < t_n = T$ with $t_j - t_{j-1} = \Delta$ for all j and some (small) constant Δ.

An approximate sample $\hat{X} = (\hat{X}(t_0), \ldots, \hat{X}(t_n))$ from the law of $(X(t_0), \ldots, X(t_n))$ is then constructed on this discrete grid. (Since X is Markov, it often makes sense to do

this recursively.) A continuous-time path $\hat{X}(0:T)$ is then constructed on the whole interval $[0,T]$ by interpolating between the $\hat{X}(t_k)$.

A natural example is to return to the discretisation (2.1) used to define the Ito integral, which we now write in a form relating $\hat{X}(t_k)$ to $\hat{X}(t_{k-1})$:

$$\hat{X}(t_k) = \hat{X}(t_{k-1}) + a(t_{k-1}, \hat{X}(t_{k-1}))\Delta + b(t_{k-1}, \hat{X}(t_{k-1}))\Delta B_k, \tag{2.6}$$

where $\Delta B_k = B(t_k) - B(t_{k-1}) \sim \mathcal{N}(0, \Delta)$ are independent and identically distributed. This implies a numerical scheme for simulating X approximately, known as the *Euler-Maruyama scheme*: initiate $\hat{X}(t_0) = x_0$, and then in sequence sample ΔB_k, followed by $\hat{X}(t_k)$ given $\hat{X}(t_{k-1})$. Finally, a continuous-time path may be obtained by taking the path to be piecewise constant on intervals $[t_{k-1}, t_k)$. This is detailed in Algorithm 1 below.

Algorithm 1 Euler-Maruyama scheme

Given $\hat{X}(t_0) = x_0$:

1. For $k = 1, \ldots, n$:
   (a) Simulate $\Delta B_k \sim \mathcal{N}(0, \Delta)$
   (b) Calculate
   $$\hat{X}(t_k) = \hat{X}(t_{k-1}) + a(t_{k-1}, \hat{X}(t_{k-1}))\Delta + b(t_{k-1}, \hat{X}(t_{k-1}))\Delta B_k.$$
2. Form $\hat{X}(0:T)$ by defining for all $t \notin \{t_0, \ldots, t_n\}$:
   $$\hat{X}(t) = \hat{X}(t_j) \text{ for } j = \{\max k : t_k < t\}.$$

The Euler-Maruyama scheme is sufficiently simple and attractive as to be very widespread in the practical simulation of diffusion paths, especially when no detailed study of the induced approximation error is to be made. Often, however, it is desirable to be more selective about the accuracy of a chosen scheme. Two useful characterisations of the quality of the resulting approximation are the *strong convergence order* and *weak convergence order* of the scheme. (We here follow [Kloeden and Platen, 2013, Section 9.6-9.7]).

Let $\hat{X}^\Delta(0:T)$ be a discrete approximation to $X_{0:T}$ generated by a recursive method such as the Euler-Maruyama scheme on the regular time grid $t_0, \ldots, t_n$ with $t_j - t_{j-1} = \Delta$. Then we say that $\hat{X}^\Delta$ converges strongly to X with strong order $s > 0$ at time T if there are constants $\Delta_0, C > 0$ such that for all $\Delta < \Delta_0$,

$$\mathbb{E}\left|X(T) - \hat{X}^\Delta(T)\right| \leq C\Delta^s.$$

That is, the strong convergence rate is a measure of the absolute error in the approximation of the path X itself.

Sometimes we are not interested so much in approximating X itself as the expected value of some functional φ of X. We say that $\hat{X}^\Delta$ converges weakly to X with respect

to φ with weak order $w > 0$ at time T if there are constants $\Delta_0, C > 0$ such that for all $\Delta < \Delta_0$,

$$\left|\mathbb{E}[\varphi(X(T))] - \mathbb{E}[\varphi(\hat{X}^{\Delta}(T))]\right| \leq C\Delta^w.$$

We set down here as a matter of record that the Euler-Maruyama scheme of Algorithm 1 has weak order of convergence $w = 1$, and strong order $s = 0.5$. Many schemes with superior convergence rates have been constructed and studied; [Kloeden and Platen, 2013, Chapters 10-15] describes many alternatives in some detail. In general, these higher-order schemes are designed to ensure either good strong or weak convergence properties, but not both.

One weakness of discrete numerical schemes in general is the approximation error they induce and the high computational cost of decreasing this error by reducing the grid size Δ. A second, less obvious problem is that interpolation between the discrete samples $(\hat{X}^{\Delta}(t_0), \ldots, \hat{X}^{\Delta}(t_n))$ is entirely insufficient for evaluating certain interesting functionals φ of the entire sample path $X(0:T)$.

An example of such a functional which is of importance in this thesis is the "barrier crossing" functional $\varphi(X(0:T)) = \mathbb{1}(\sup_{t\in[0,T]} V(X(t)) \geq a)$, where $V : \mathbb{R}^d \to \mathbb{R}$ is continuous and $a \in \mathbb{R}$.

If it happens that $V(\hat{X}^{\Delta}_k) \geq a$ for some k, then it is clear that $\varphi(\hat{X}^{\Delta}(0:T)) = \varphi(X(0:T)) = 1$. On the other hand, if $V(\hat{X}^{\Delta}_k) < a$ for all k, it remains undetermined which value $\varphi(X(0:T))$ takes for the underlying process X. Figure 2.1 illustrates the trouble one can run into when using discretisation methods to approximate barrier-type functions. Chapter 4 considers this problem in more detail.

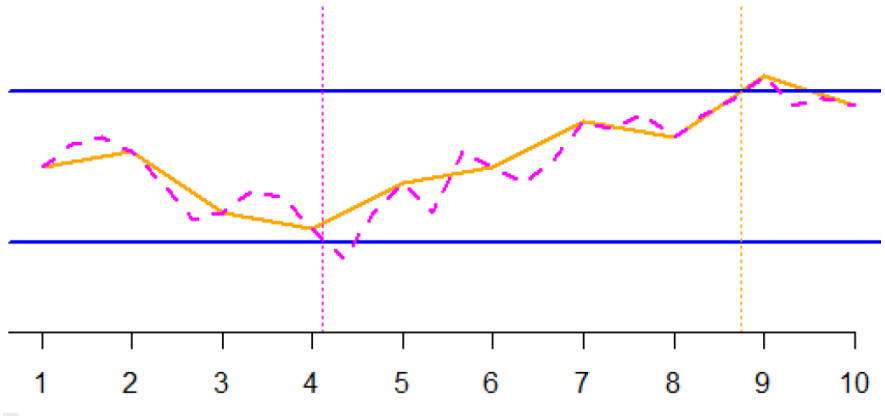


Figure 2.1: Same underlying diffusion path approximated by discretely at two different levels of fineness (the coarser in solid orange, and the finer in dashed pink). The question of *which barrier is crossed first* is answered differently by each, as shown by the vertical dotted lines. Using the finer discretiation it would be concluded that the lower barrier was hit first at around time 4, whereas using the coarser the conclusion would be that the upper barrier was hit first and at around time 9.

2.2 ε-strong methods

We have mentioned the shortcomings associated with numerical discretisation schemes in certain situations. In this section, we describe two classes of methods which address these shortcomings, both of which have been the focus of much recent work. *Exact simulation* methods ensure that samples of a diffusion process X on some discrete grid $0 = t_0 < \ldots < t_k = T$ obey the marginal joint law of $(X(t_0), \ldots, X(t_k))$ induced by the stochastic differential equation 2.4.

These first received attention in the work of Beskos and Roberts [2005] and the simplified construction of Beskos et al. [2006], and have received continued further attention (for example Chen and Huang [2013]; Pollock et al. [2016]). We do not focus on these methods in this chapter since we do not use them directly in this thesis. However, we do note that the method of Beskos et al. [2006] is closely related to that of the second ε-strong algorithm we describe below in section 2.2.3.

As we noted earlier, exact methods are not always sufficient to evaluate the expectation $\mathbb{E}[\varphi(X(0:t))]$ of a functional f of a complete diffusion path $X(0:t)$ (for example, the barrier-crossing functional described previously). *ε-strong methods* provide instead information constraining the range of the path between the discrete grid times $t_0, \ldots, t_k$. This is not a fully general solution to the problem of evaluating expectations of functionals. However, it is sufficient to address barrier-crossing problems, and it is these which we will consider in chapter 4.

2.2.1 ε-strong simulation

Formally, following the definition given in Blanchet et al. [2017], an ε-strong algorithm is a joint construction of X together with a family of processes $\tilde{X}^\varepsilon$ indexed by $\varepsilon > 0$ (defined on the same probability space) over an interval $[s, t]$ such that the following four properties hold:

1. Almost surely, $\sup_{r \in [s,t]} \|X(r) - \tilde{X}^\varepsilon(r)\| \leq \varepsilon$ for an appropriate norm $\|\cdot\|$;
2. $\tilde{X}^\varepsilon$ is piece-wise constant and left-continuous on $[s, t]$, taking only finitely many values and so can be fully stored on a computer;
3. $\tilde{X}^\varepsilon$ can be simulated exactly. (That is, to sample $\tilde{X}^\varepsilon$ it is necessary to sample certain intermediate random variables, and this criterion requires that this can be done without approximations); and
4. Given any finite sequence of tolerances $\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_m > 0$, for $1 \leq k < \ell \leq m$ it holds almost surely for all $r \in [s, t]$ that

$$\{x : \|\tilde{X}^{\varepsilon_\ell}(r) - x\| \leq \varepsilon_\ell\} \subset \{x : \|\tilde{X}^{\varepsilon_k}(r) - x\| \leq \varepsilon_k\},$$

   and moreover it is possible to sample explicitly $\tilde{X}^{\varepsilon_\ell}$ conditional on $\tilde{X}^{\varepsilon_k}$.

An ε-strong algorithm produces a chain (in time) of finitely many $\|\cdot\|$-balls, each of

which almost surely constrains the sample path of X over the corresponding interval of time. Moreover, by applying Property 4 the radius of these balls can be iteratively reduced, constraining X progressively more tightly by employing a greater number of balls. An example (in two spatial dimensions) of how ε-strong sampling may be used is given in Figure 2.2.

It is often advantageous to apply Property 4 selectively in order to get tight constraints on X at certain locations of interest, while allowing looser constraints elsewhere. For example, Figure 2.2(c) shows the result of applying Property 4 to the first two ε_1-balls of the initial ε_1-sample in 2.2(b).

The choice of a weak inequality in 1) differs slightly from the presentation in Blanchet et al. [2017]. The reason is simply that in chapter 4 our application requires calculating suprema and infima of a continuous function ξ over regions $C(t) = \{x : \|\tilde{X}(t) - x\| \leq \varepsilon\}$, and the weak inequality ensures that these extrema are attained in the regions $C(t)$.

The insistence in Property 2 that $\tilde{X}^\varepsilon$ be piece-wise constant is not strictly necessary since other processes which admit finite-dimensional representations could fill the same role. For example, continuous and piece-wise linear/polynomial $\tilde{X}^\varepsilon$ are possible alternatives. However, we will assume throughout that $\tilde{X}^\varepsilon$ is piece-wise constant.

We now briefly describe some important examples, some of which we used in the simulations of Chapter 4. These cover the cases of i) Brownian motion (Section 2.2.2), ii) diffusions with volatility which can be transformed to unity via the Lamperti transform (Section 2.2.3), and iii) diffusions in multiple dimensions (Section 2.2.4).

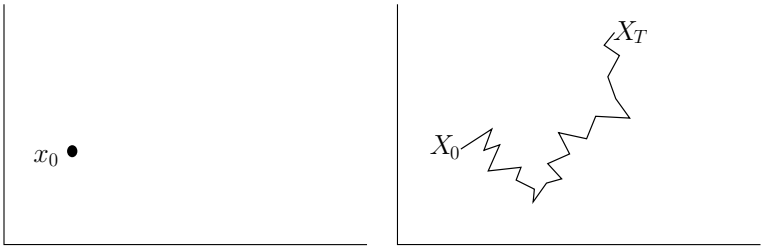
Several other examples of ε-strong exist, and their construction is an active area of research. Examples include methods for SDEs driven by fractional Brownian motion in Chen et al. [2019], for the Bessel process in Deaconu and Herrmann [2021], and for the convex minorants of stable processes in González Cázares et al. [2020].

2.2.2 Example: Brownian motion

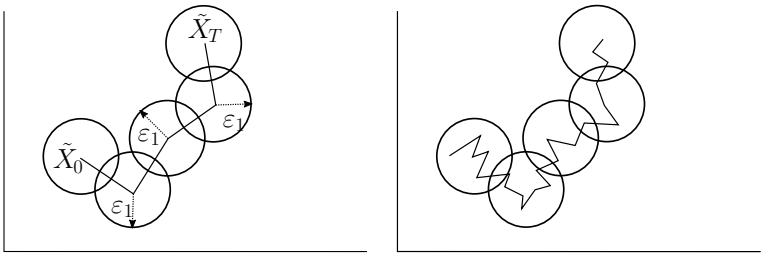
The first example of an ε-strong algorithm was the construction for Brownian motion given in Beskos et al. [2012]. Let B denote a standard Brownian motion in $\mathbb{R}$, and let $0 \leq s < t$ with $B(s) = x, B(t) = y$. Let $M_{s,t} = \sup_{r\in[s,t]} B_r$ and $m_{s,t} = \inf_{r\in[s,t]} B_r$. Finally, take $L \leq x \wedge y \leq x \vee y \leq U$. Then the probability p that B escapes the interval $[L, U]$ before time t admits a certain alternating series representation

$$p(t-s, x, y, L, U) = \mathbb{P}\left(m_{s,t} < L \text{ or } U < M_{s,t} | B(s) = x, B(t) = y\right) = \sum_{i=1}^{\infty} (\sigma_i - \tau_i) \ ,$$

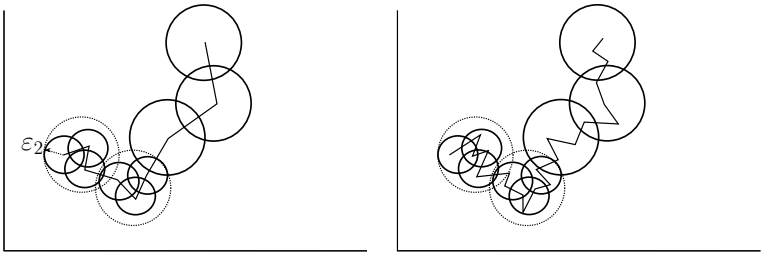
where $\sigma_i(x, y, t-s, L, U), \tau_i(x, y, t-s, L, U) > 0$, and have explicit forms in terms of standard functions (see [Beskos et al., 2012, Section 4], and Pötzelberger and Wang [2001] for a full derivation. We omit these expressions and other details which are not directly relevant for the contributions of this thesis in the interest of brevity.) Note that for any L, U which do not fall into the specified range, this probability is 0.



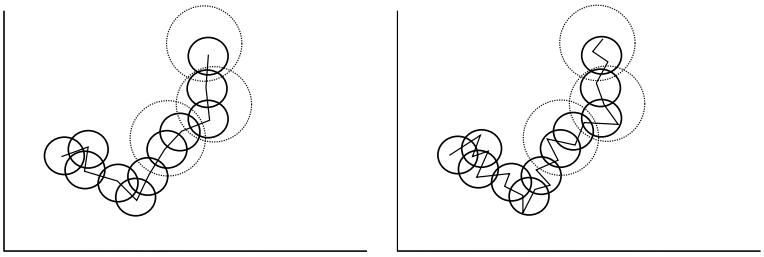
(a) Initial value $X_0 = x_0$, and target path X



(b) ε_1-strong sample and constraining regions



(c) Partial ε_2-strong sample



(d) Full ε_2-strong sample

Figure 2.2: An illustration of ε-strong simulation. The left column shows shows the ε-strong process $\tilde{X}$ developing as conditional samples are made first with tolerance ε_1, followed with $\varepsilon_2 < \varepsilon_1$. The right column shows the fixed target path X, and how the ε-strong constraints relate to it. Pale circles indicate superseded constraints from the previous step.

The partial sums S_j are given by adding each σ_i and subtracting each τ_i alternately, so that $S_0 = 0$, and for $k \geq 1$, $S_{2k-1} = S_{2k-2} + \sigma_k$ and $S_{2k} = S_{2k-1} - \tau_k$. Writing $p := p(t-s, x, y, L, U)$, these sums satisfy

$$S_{2k} \leq S_{2k+2} \leq p \leq S_{2k+1} \leq S_{2k-1}. \tag{2.7}$$

This expansion can be be used to sample p exactly: first, sample a standard uniform random variable V, and then calculate sufficient terms in the convergent series to determine whether V is greater than or less than p. This technique is known as the alternating series method (Devroye [1981]) and elsewhere as *retrospective Bernoulli sampling*. It is given in Algorithm 2:

Algorithm 2 Retrospective Bernoulli sampling

Given a probability p and an alternating series $(S_n)_{n\in\mathbb{N}}$ satisfying (2.7), returns 1 with probability p and 0 otherwise.

1. Simulate $V \sim \mathcal{U}[0,1]$.
2. Take initial lower and upper bounds $p_0 = 0$, $q_0 = 1$ for p, and set $n = 0$.
3. While $V \in [p_n, q_n]$: set $p_{n+1} = S_{2n} \vee 0, q_{n+1} = S_{2n+1} \wedge 1$, and update $n \leftarrow n+1$.
4. If $V < p_n$ return 1; else return 0.

The alternating series representation S_n for p can be used to derive similar alternating representations for the probabilities of more complex events. For example, writing $\mathbb{P}^{x,y}_{s,t}$ for the law of Brownian motion conditional on $B(s) = x, B(t) = y$, and writing $q(r, x, y, L, U) = 1 - p(r, x, y, L, U)$, the probability that the maximum and minimum of B lie within specified intervals is

$$\begin{aligned}\mathbb{P}^{x,y}_{s,t}\left(m_{s,t} \in [L^{\downarrow}, L^{\uparrow}], M_{s,t} \in [U^{\downarrow}, U^{\uparrow}]\right) \\ = q(L^{\downarrow}, U^{\uparrow}) - q(L^{\downarrow}, U^{\downarrow}) - q(L^{\uparrow}, U^{\uparrow}) + q(L^{\uparrow}, U^{\downarrow})\end{aligned}$$

(where the dependence of q on $(t-s), x, y$ has been suppressed). Therefore

$$R_n = S_n(L^{\downarrow}, U^{\uparrow}) - S_{n+1}(L^{\downarrow}, U^{\downarrow}) - S_{n+1}(L^{\uparrow}, U^{\uparrow}) + S_n(L^{\uparrow}, U^{\downarrow})$$

is an alternating series for $\mathbb{P}^{x,y}_{s,t}\left(m_{s,t} \in [L^{\downarrow}, L^{\uparrow}], M_{s,t} \in [U^{\downarrow}, U^{\uparrow}]\right)$, and so events of this probability may be sampled exactly using Retrospective Bernoulli Sampling (Algorithm 2).

Again, to avoid reproducing details incidental to the primary purpose of this thesis we refer the reader to Beskos et al. [2012] for explicit constructions of the sampling schemes for various other events. For our purpose it is sufficient to know that the following three sampling operations can be carried out exactly using developments along the above lines:

1. *Initial layers* ([Beskos et al., 2012, Section 5.3]): given $B(s)$, it is possible to sample $B(t)$ together with the initial constraints $(L^{\downarrow}_{s,t}, L^{\uparrow}_{s,t}, U^{\downarrow}_{s,t}, U^{\uparrow}_{s,t})$.
2. *Refine layers* ([Beskos et al., 2012, Section 5.3]): given $s, t, B(s) = x, B(t) =$

y, and $(L^{\downarrow}_{s,t}, L^{\uparrow}_{s,t}, U^{\downarrow}_{s,t}, U^{\uparrow}_{s,t})$, it is possible to narrow the constraints: that is, to sample an indicator random variable for the event $L^{\downarrow}_{s,t} < m_{s,t} < L^{*}_{s,t}$ for any $L^{*}_{s,t} \in [L^{\downarrow}_{s,t}, L^{\uparrow}_{s,t}]$, and likewise for $M_{s,t}$.

3. *Bisect layers* ([Beskos et al., 2012, Sections 5.1, 5.2]): given $s, t, B(s) = x, B(t) = y$, and $(L^{\downarrow}_{s,t}, L^{\uparrow}_{s,t}, U^{\downarrow}_{s,t}, U^{\uparrow}_{s,t})$, it is possible to sample the mid-point[1] $B(s^*)$ for $s^* = \frac{1}{2}(s+t)$, together with new constraints $(L^{\downarrow}_{s,s^*}, L^{\uparrow}_{s,s^*}, U^{\downarrow}_{s,s^*}, U^{\uparrow}_{s,s^*})$ and $(L^{\downarrow}_{s^*,t}, L^{\uparrow}_{s^*,t}, U^{\downarrow}_{s^*,t}, U^{\uparrow}_{s^*,t})$. In fact, it is possible to do this for all $r \in (s,t)$.

These sampling procedures are made clearer by Figure 2.3.

These procedures are combined as follows to obtain an ε-strong sample for B: after sampling the initial layers, alternate between *bisecting* each available interval $[s,t]$; and iteratively *refining* to tighten each existing constraint until each is smaller than a chosen threshold. A more formal description is given in Algorithm 3 below.

Algorithm 3 ε-strong sampling for Brownian motion

Given $B(0) = 0$ and $\varepsilon > 0$:

1. Simulate $B(T) \sim \mathcal{N}(0,T)$ and $L^{\downarrow}_{0,T}, L^{\uparrow}_{0,T}, U^{\downarrow}_{0,T}, U^{\uparrow}_{0,T}$. Store all these variables together with $B(0) = 0$ in $\mathcal{I}_{0,T}$, and let $\mathcal{I} = \{\mathcal{I}_{0,T}\}$.
2. While $\sup_{\mathcal{I}} \left(U^{\uparrow}_{s,t} - L^{\downarrow}_{s,t}\right) > 2\varepsilon$:
   (a) Bisect the layers of each $\mathcal{I}_{s,t} \in \mathcal{I}$ (procedure 3), and remove $\mathcal{I}_{s,t}$ from $\mathcal{I}$ and replace it with $\mathcal{I}_{s,s^*}$ and $\mathcal{I}_{s^*,s}$.
   (b) Refine the layers of each new $\mathcal{I}_{s,t} \in \mathcal{I}$ (procedure 2) until

$$\max\left(L^{\uparrow}_{s,t} - L^{\downarrow}_{s,t}, U^{\uparrow}_{s,t} - U^{\downarrow}_{s,t}\right) < \sqrt{\frac{1}{2}(t-s)}.$$

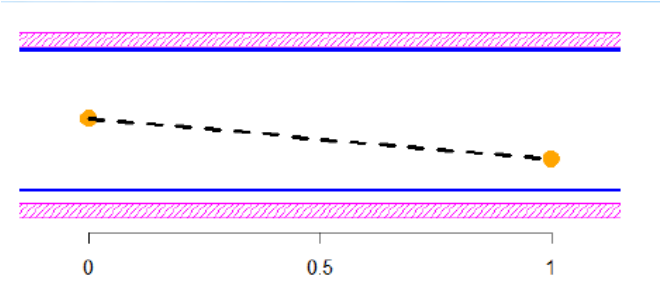
Procedure 3 allows B to be sampled on an increasingly fine dyadic time-grid conditional on all the constraints on its maxima and minima. Procedure 2 ensures (probabilistically) that the fluctuations of B between the sampled points become more and more tightly constrained. It is established in [Beskos et al., 2012, Section 3] that this procedure converges in the sense that $\sup_{\mathcal{I}}(U^{\uparrow}_{s,t} - L^{\downarrow}_{s,t}) \to 0$ almost surely as $n \to \infty$, and that the choice $\sqrt{\frac{1}{2}(t-s)}$ of threshold is optimal for fast convergence.

It will be observed that Algorithm 3 samples more detail about the Brownian path than is strictly needed for an ε-strong algorithm as defined in Section 2.2.1. To extract an ε-strong sample $\tilde{B}^{\varepsilon}$ in this sense from Algorithm 3, we take

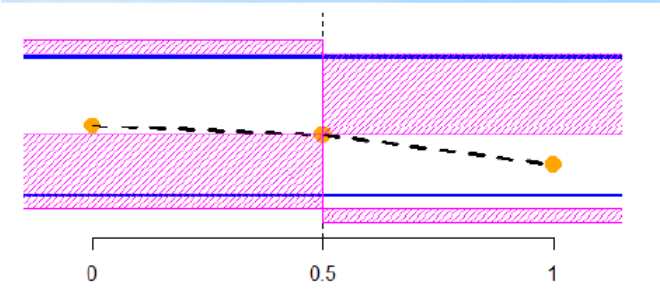
$$\tilde{B}^{\varepsilon}(r) = \sum_{(s,t)\in\mathcal{I}} 1(r \in [s,t)) \frac{1}{2}(U^{\uparrow}_{s,t} + L^{\downarrow}_{s,t})$$

where $\mathcal{I}$ is as defined in Algorithm 3 (see figure).

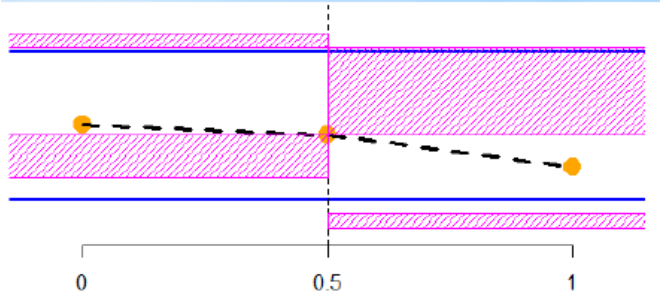
[1] The method described in Section 5.1 of Beskos et al. [2012] to sample the mid-point is not in fact quite exact, since it involves sampling from a density by numerically inverting its (analytically derived) CDF. Several genuinely exact methods for carrying out this particular procedure are described in [Pollock, 2013, Section 6.3.2], designed to be efficient in various regimes. These methods were used for the simulations in Chapter 4.



(a) $B(0)$, $B(1)$ and initial layers.



(b) Bisection at $t = 0.5$, with new layers on either side



(c) Refinement of layer

Figure 2.3: An illustration of ε-strong simulation of one-dimensional Brownian motion. Orange dots are exactly sampled points of the Brownian motion, which the black dashed line interpolates linearly. Pink shaded regions are layers $[L^{\downarrow}, L^{\uparrow}], [U^{\downarrow}, U^{\uparrow}]$. The blue line define a region of interest - we might wish to know whether the Brownian motion escapes the region, and if so on which side it first escapes.

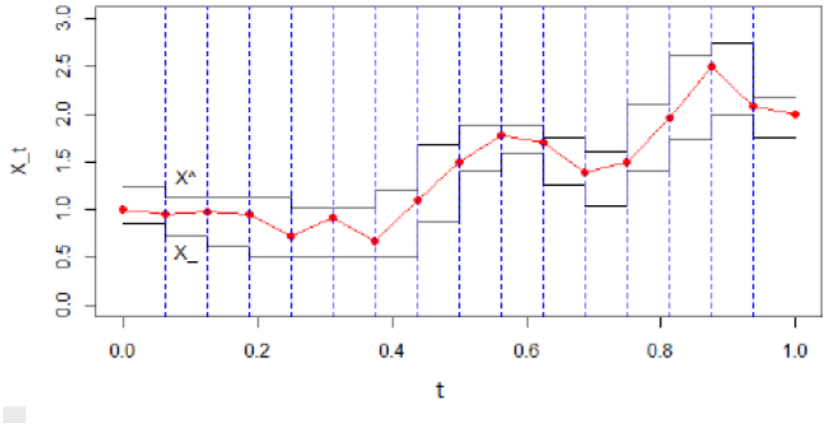


Figure 2.4: After several iterations of Algorithm 3. Red dots are exactly sampled values of Brownian path, red lines are linear interpolation between them. Upper and lower processes in black are sequences of $U^{\uparrow}$ and $L^{\downarrow}$ respectively, which define upper and lower bounding processes for the underlying path.

It is then clear that all the required properties detailed in Section 2.2.1 are available.

However, this construction is rather coarse in throwing away the extra information given by the $L^{\uparrow}$s and $U^{\downarrow}$s, and for the problems we will consider in chapter 4, something closer the full algorithm will be found useful.

Finally, we note that Brownian motion in $\mathbb{R}^d$ may be simulated in an ε-strong fashion by carrying out the procedures detailed in this section for each dimension independently. This is the example we actually use for the 2D simulations in Chapter 4, due to its relative computational simplicity.

2.2.3 Example: Diffusions with unit volatility

The next important contribution to the ε-strong sampling literature was a method for diffusion processes, described in Pollock et al. [2016]. The technique is to use Brownian-type ε-strong proposals, which are accepted or rejected according to an appropriate scheme. Consider the one-dimensional diffusion process $X(0) = X(0)$,

$$dX(t) = a(X(t))dt + b(X(t))dB(t).$$

The crucial result which is exploited in order to make the rejection sampling scheme tractable is the *Lamperti transform* ([Kloeden and Platen, 2013, Chapter 4.4]).

Theorem 6. Lamperti transform*: Suppose a is continuously differentiable, b is strictly positive and twice continuously differentiable, and for some $C > 0$ and every $x \in \mathbb{R}$*

$$|a(x)|^2 + |b(x)|^2 \leq C(1 + |x|^2).$$

Let $Y(t) = \eta(X(t)) = \int_x^{X(t)} \frac{du}{b(u)}$, *where* x *is arbitrary. Then*

$$dY(t) = \left[\frac{a(\eta^{-1}(X(t)))}{b(\eta^{-1}(X(t)))} - \frac{b'(\eta^{-1}(X(t)))}{2}\right] dt + dB(t),$$

ie. Y *is a diffusion process with unit volatility.*

This allows us, when the given assumptions are met and the transform and its inverse can be calculated, to work with stochastic differential equations with unit volatility $b(X(t)) = 1$:

$$dX(t) = a(X(t))dt + dB(t). \tag{2.8}$$

The benefits of doing so are made immediately apparent by the following observation: writing $\mathbb{Q}^x_{0:T}$ for the law induced by 2.8 over $[0, T]$ with $X(0) = x$, and $\mathbb{Z}^x_{0:t}$ for the law of Brownian motion over $[0, T]$ with initial value x, the Radon-Nikodym derivative $d\mathbb{Q}/d\mathbb{Z}$ has the form

$$\begin{aligned}\frac{d\mathbb{Q}^x_{0:t}}{d\mathbb{Z}^x_{0:t}}(X) &= \exp\left(\int_0^T a(X(s))dB(s) - \frac{1}{2}\int_0^T a^2(X(s))ds\right) \\ &= \exp\left(\int_x^{X(t)} a(u)du - \frac{1}{2}\int_0^T \left(a^2(X(s)) + a'(X(s))\right) ds\right)\end{aligned}$$

where the second line follows from applying Ito's formula.

The difficult term in this expression is $\int_x^{X(t)} a(y)dy$. It may be removed by taking the RN-derivative of $\mathbb{Q}$ with respect to a slightly different measure. Define the probability density h by

$$h(y; x, T) \propto \exp\left(\int_0^y a(u)du - \frac{(y-x)^2}{2T}\right),$$

and let $\mathbb{W}^x_{0:T}$ be the path-space measure of the Brownian bridge W with $W_0 = x$ and $B(t) \sim h(\cdot)$. Then

$$\begin{aligned}\frac{d\mathbb{Q}^x_{0:t}}{d\mathbb{W}^x_{0:t}}(X) &= \frac{d\mathbb{Q}^x_{0:t}}{d\mathbb{Z}^x_{0:t}}(X)\left(\frac{d\mathbb{W}^x_{0:t}}{d\mathbb{Z}^x_{0:t}}(X)\right)^{-1} \\ &= \exp\left(-\frac{1}{2}\int_0^T a^2(X(s)) + a'(X(s))ds\right) \\ &=: \exp\left(-\int_0^T \phi(X(s))ds\right)\end{aligned}$$

where $\phi(X(s)) = (a^2(X(s)) + a'(X(s)))/2$.

Suppose $\phi(x) > \Phi$ for all x. Then the beginnings of a rejection sampler for the solution path associated with $\mathbb{Q}$ becomes apparent (we explain how this extends to a ε-strong sampler shortly). Writing $\mathbb{Z}^{x,y}_{0:T}$ for the law of Brownian motion over the interval $(0, T)$ with starting value x and terminal value y, that is

1. Draw $X(t) \sim h(\cdot)$

2. Draw $X_{(0:T)} \sim \mathbb{Z}_{0:T}^{x,y}$

3. Accept path X with probability

$$p(X) = \exp\left(-\int_0^T \phi(X(s))ds\right)\exp\left(\Phi T\right).$$

The first step can often be carried out by constructing an appropriate rejection sampling procedure to obtain samples from h. The problem with this sketch is that neither the second nor third step is tractable: it is not possible to sample an entire continuous-time path, nor to calculate the requisite integral. However, one solution to these problems also provides a way to draw ε-strong samples rather than simply finite-dimensional marginals.

The insight is that if step 2 in the above is replaced with an ε-strong proposal of the Brownian bridge using Algorithm 3, then the ε-strong bounds can be used to construct upper and lower bounds $\phi^{\uparrow}, \phi^{\downarrow}$ for $p(X)$. For example, having sampled the proposal path

$$X^{\varepsilon}_{\text{prop}} = \sum_{[s,t) \text{ in } \mathcal{I}} 1(r \in [s,t))X^{\varepsilon}_{s,t},$$

one has immediately the upper and lower bounds for ϕ of

$$\phi^{\downarrow} = \exp\left(-\sum_{[s,t) \text{ in } \mathcal{I}} \sup_{v \in X^{\varepsilon}_{s,t} \pm \frac{\varepsilon}{2}} (\phi(v) - \Phi)(t-s)\right),$$
$$\phi^{\uparrow} = \exp\left(-\sum_{[s,t) \text{ in } \mathcal{I}} \inf_{v \in X^{\varepsilon}_{s,t} \pm \frac{\varepsilon}{2}} (\phi(v) - \Phi)(t-s)\right).$$

By defining a sequence of tolerances $\varepsilon_n \to 0$, we obtain in the same way sequences $(\phi_n^{\downarrow}), (\phi_n^{\uparrow})$ of upper and lower bounds.

$$\phi_n^{\downarrow} = \exp\left(-\sum_{[s,t) \text{ in } \mathcal{I}_n} \sup_{v \in X^{\varepsilon_n}_{s,t} \pm \frac{\varepsilon_n}{2}} (\phi(v) - \Phi)(t-s)\right),$$
$$\phi_n^{\uparrow} = \exp\left(-\sum_{[s,t) \text{ in } \mathcal{I}_n} \inf_{v \in X^{\varepsilon_n}_{s,t} \pm \frac{\varepsilon_n}{2}} (\phi(v) - \Phi)(t-s)\right),$$

each converging to $p(X)$. These sequences can be used to accept or reject the ε-strong path with the exact probability $p(X)$ using retrospective Bernoulli sampling (Algorithm 2). Algorithm 4 illustrates this procedure.

[Pollock et al., 2016, Algorithm 4] is also suggested to be employed as an ε-strong algorithm, and they observe that numerically it is superior in convergence speed to Algorithm 4.

Algorithm 4 ε-strong rejection sampler (Algorithm 10 of Pollock et al. [2016])

Given $X(0) = x$:

1. Simulate $y = X(T) \sim h(\cdot)$.
2. Simulate initial layers

$$(L^{\downarrow}_{0,T}, L^{\uparrow}_{0,T}, U^{\downarrow}_{0,T}, U^{\uparrow}_{0,T}) \sim \mathbb{W}^{x,y}_{0,T}$$

   according to procedure (1) of Section 2.2.2, and calculate $\phi^{\downarrow}_1, \phi^{\uparrow}_1$.
3. Simulate $V \sim U[0,1]$.
4. Then while $\phi^{\downarrow}_n \leq V \leq \phi^{\uparrow}_n$:
   (a) Find

$$[s,t) = \arg\max_{[s',t')\in\mathcal{I}} \left(\left(\sup_{v \in X^n_{s',t'} \pm \frac{1}{2}\varepsilon^n_{s',t'}} \phi(v) - \inf_{v \in X^n_{s',t'} \pm \frac{1}{2}\varepsilon^n_{s',t'}} \phi(v) \right) (t' - s') \right).$$

   (b) Simulate a midpoint and new layers over the interval $[s,t]$ according to procedure (2) of Section 2.2.2.
   (c) Refine this layer according to procedure (3) of Section 2.2.2.
   (d) Set $n \leftarrow n+1$, update X^{n+1} and calculate $\phi^{\downarrow}_n, \phi^{\uparrow}_n$.
5. Accept X^n if $V \leq \phi^{\downarrow}_n$, else reject and return to step 1.

We note that despite the use of the Lamperti transform, as for the Brownian motion example this method is not entirely restricted to one-dimensional diffusion processes. It is also applicable to multi-dimensional processes whose diffusion coefficients can be transformed to unity. A class of such processes are, for example, those with diffusion coefficient a multiple of the identity matrix.

2.2.4 Example: Multidimensional diffusions via rough paths

A very general algorithm for ε-strong simulation of solutions to multi-dimensional stochastic differential equations has been constructed in Blanchet et al. [2017]. It is based on quite different ideas, derived from the theory of rough paths. Rough path theory provides a comprehensive theory of integration against Hölder-continuous functions which embraces much of the older principle theories of integration, including the Riemann-Stieltjes and Young integrals discussed in 2.1.2, and the Ito integral discussed in 2.1.3. Since we are mainly concerned with SDEs driven by Brownian motion, we will try to present the results we require in fairly elementary terms and avoid any more abstract formalism.

The attractiveness of rough path theory as a tool for constructing ε-strong algorithms for SDEs is that its definition of an integral both i) takes the form of a limit of a Riemann-like sum, and ii) makes the integral a *continuous* function of the driving Brownian motion (in a sense to be made precise below). By sampling the driving Brownian motion on a fine discrete grid to obtain a close approximation to the whole Brownian path, and then passing this discrete sample through a discretisation scheme derived from the the Riemann-like sum (analogously to (2.3) for the Ito integral),

one might hope that the continuity guarantees a result "close" to the SDE solution. Remarkably, the paper under discussion shows this idea can be made to work: in particular, it provides the necessary quantitative estimates of continuity constants to establish how fine a discrete grid is necessary, and provides an algorithm for carrying out the simulations.

It is not proposed to address much of the algorithmic detail in the paper itself here, since it is quite involved, but rather to give a short, self-contained account motivating the use of rough paths for ε-strong estimation, and briefly detailing some background material in rough paths useful for understanding a crucial definition in it.

Unfortunately, computational considerations meant we were unable to use this algorithm directly in chapter 4. The issue we encountered is briefly touched on in Section 7, item 2 on pg. 43 of the paper, and relates to the trade-off between two tuning parameters α and β which must be chosen under the constraints that $\alpha \in (1/3, 1/2)$ and $\beta \in (1-\alpha, 2\alpha)$. Theorems 2.1 and 2.2 quoted below imply that the further both α and β are from $1/2$, the finer the dyadic grid on which the Brownian path must be simulated. Conversely, the upper bound required by [Blanchet et al., 2017, Lemma 5.2, condition (5.9)] is forced to be small by values of α and β near $1/2$, and is largest (and therefore weakest) for α near $1/3$ and β near $2/3$. Thus values of α, β nearer $1/2$ require sampling on a very fine grid $\mathcal{D}_n$ to meet condition (5.9), whereas values of α nearer $1/3$ and of β nearer $2/3$ also require sampling on a very fine grid $\mathcal{D}_n$ to ensure that $G\Delta_n^{2\alpha-\beta}$ in Theorems 2.1 and 2.2 is smaller than the chosen ε.

In practice, it was found that by choosing α very near to $1/3$, it was possible to meet condition (5.9) reasonably easily, but that ensuring the resulting $G\Delta_n^{2\alpha-\beta}$ was sufficiently small required a prohibitively large n for the extensive simulation required by our multilevel splitting algorithms. However, increasing α towards $1/2$ resulted in difficulties meeting condition (5.9) arising before a sufficient compensating decrease in $G\Delta_n^{2\alpha-\beta}$ was obtained. It is possible that a more careful computation implementation could mitigate this issue.

However, we hope the account below complements the exposition of Blanchet et al. [2017] and is of some independent interest.

Some motivational remarks

Here we begin with the stochastic differential equation

$$dX(t) = a(X(t))dt + b(X(t))dB(t) \tag{2.9}$$

on the interval $t \in [0, 1]$, where $B(t)$ is d-dimensional Brownian motion, $b : \mathbb{R}^{d'} \to \mathbb{R}^{d' \times d}$, $a : \mathbb{R}^{d'} \to \mathbb{R}^{d'}$. We take the initial condition $X(0) = x_0$, and write (for example) B_i for the i-th component of the vector B.

Rather than the usual Ito solution, X is defined to be a solution to this equation under

the following condition (see Blanchet et al. [2017, p. 31]). Let $A_{i,j} : [0,\infty)^2 \to \mathbb{R}$ for $i, j = 1, \ldots, d$ be functions satisfying

$$A_{i,j}(r,t) = A_{i,j}(r,s) + A_{i,j}(s,t) + (B_i(s) - B_i(r))(B_j(t) - B_j(s)), \tag{2.10}$$

for any $s \in (r,t)$. More succinctly, this may be written

$$A(r,t) = A(r,s) + A(s,t) + (B(s) - B(r)) \otimes (B(t) - B(s)) \tag{2.11}$$

where $\otimes$ is the tensor product $\otimes : (\mathbb{R}^d)^2 \to \mathbb{R}^{d \times d}$. Then X is a solution to the SDE if $X(0) = X(0)$, and for each $i = 1, \ldots, d$, almost surely

$$\begin{aligned} |X_i(t) - X_i(s) - a_i(X(s))(t-s) - \sum_{j=1}^{d'} b_{i,j}(X(s))(B_j(t) - B_j(s)) \\ - \sum_{j=1}^{d'} \sum_{\ell=1}^{d} \sum_{m=1}^{d'} \partial_\ell b_{i,j}(X(s)) b_{\ell,m}(X(s)) A_{m,j}(s,t)| = o(t-s). \end{aligned} \tag{2.12}$$

The restriction (2.10) should be viewed in light of the important case with

$$A_{i,j}(r,t) = \int_r^t (B_i(u) - B_i(r)) dB_j(u), \tag{2.13}$$

where $\int \cdot dB_j$ is the Ito integral, for which the relation is easily verified:

$$\begin{aligned} \int_r^t (B_i(u) - B_i(r)) dB_j(u) &= \int_r^s (B_i(u) - B_i(r)) dB_j(u) + \int_s^t (B_i(u) - B_i(r)) dB_j(u) \\ &= A_{i,j}(r,s) + \int_s^t (B_i(u) - B_i(s)) dB_j(u) + \int_s^t (B_i(s) - B_i(r)) dB_j(u) \\ &= A_{i,j}(r,s) + A_{i,j}(s,t) + (B_i(s) - B_i(r))(B_j(t) - B_j(s)). \end{aligned}$$

It may be observed that with this choice of $A_{i,j}$, the definition (2.12) takes on the appearance of a higher-order local approximation resulting from a Taylor expansion. A detailed exposition of this notion of a solution may be found in [Davie, 2008; Friz and Victoir, 2010, Ch. 8], both of which assume some familiarity of the reader with Terry Lyons' theory of rough paths. We provide some motivation below.

The resulting Euler scheme

Before developing this mathematical background, we note the Euler scheme implied by the above definition. Just as the conventional definition of the Ito integral implies the following Euler scheme for an SDE, transforming an approximation $B^n(\omega)$ of a Brownian path on the dyadic grid $\mathcal{D}_n$ into an approximate solution path $\hat{X}^n(\omega)$:

$$\hat{X}_i^n(t_{k+1}^n) = \hat{X}_i^n(t_k^n) + a_i(\hat{X}^n(t_k^n)) + \sum_{j=1}^{d'} b_{i,j}(\hat{X}^n(t_k^n))(B_j^n(t_{k+1}^n) - B_j^n(t_k^n)),$$

the rough path-inspired definition implies the following embellished scheme:

$$\begin{aligned}\hat{X}_i^n(t_{k+1}^n) =& \hat{X}_i^n(t_k^n) + a_i(\hat{X}^n(t_k^n)) + \sum_{j=1}^{d'} b_{i,j}(\hat{X}^n(t_k^n))(B_j^n(t_{k+1}^n) - B_j^n(t_k^n)) \\ &+ \sum_{j=1}^{d'}\sum_{\ell=1}^{d}\sum_{m=1}^{d'} \partial_\ell b_{i,j}(\hat{X}^n(t_k^n)) b_{\ell,m}(\hat{X}^n(t_k^n)) A_{m,j}^n(t_k^n, t_{k+1}^n).\end{aligned}$$

This scheme is intractable if we take $A_{m,j}^n$ to be the first iterated integrals described above, since they cannot be calculated for $m \neq j$. For the case $m = j$, we do however have that

$$\begin{aligned}A_{m,j}^n(r,t) = A_{m,m}^n(r,t) &= \int_r^t (B_m^n(u) - B_m^n(r)) dB_m^n(u) \\ &= \frac{1}{2}((B_m^n(t) - B_m^n(r))^2 - (t-r)).\end{aligned}$$

In order to obtain a tractable scheme therefore, one can define

$$\tilde{A}_{i,j}^n = \begin{cases} \frac{1}{2}\left((B_i^n(t) - B_i^n(r))^2 - (t-r)\right) & \text{if } i = j; \\ 0 & \text{otherwise.} \end{cases}$$

and use the modified scheme

$$\hat{X}_i^n(t_{k+1}^n) = \hat{X}_i^n(t_k^n) + a_i(\hat{X}^n(t_k^n) + \sum_{j=1}^{d'} b_{i,j}(\hat{X}^n(t_k^n))(B_j^n(t_{k+1}^n) - B_j^n(t_k^n)) \tag{2.14}$$

$$+ \sum_{j=1}^{d'}\sum_{\ell=1}^{d}\sum_{m=1}^{d'} \partial_\ell b_{i,j}(\hat{X}^n(t_k^n)) b_{\ell,m}(\hat{X}^n(t_k^n)) \tilde{A}_{m,j}^n(t_k^n, t_{k+1}^n). \tag{2.15}$$

This in fact is what is used (see [Blanchet et al., 2017, Eqn 2.4]). Replacing the intractable off-diagonal terms with 0 introduces an approximation error in $\hat{X}^n$ beyond that which is to be controlled by the following rough path considerations. This error is a technical detail which can be controlled, but the analysis is a departure from our outline of the role of rough path theory in constructing ε-strong approximations and we refer the reader to the discussions of the parameter Γ_R in Blanchet et al. [2017, Section 6], in particular Proposition 6.2.

Pathwise integration, the Ito integral and rough paths

Our procedure falls into four steps. First, we motivate and define a new type of integral, known as a *rough integral*. We will see that this definition allows integrating a very large class of functions against Brownian paths. Next, we quote a result demonstrating that in fact the new definition coincides exactly with the Ito integral when both exist.

Thirdly we show that the new integration map is in some sense a continuous function of the driving noise. We will see that although the integral cannot be a continuous function

of the driving Brownian path $B(\omega)$ alone, it *is* a continuous function of Brownian motion together with its first iterated integrals. The final step is to apply this integration theory to the stochastic differential equation.

Recall that the more common Ito solution to (2.9) is defined using the Ito integral with respect to Brownian motion. An alternative to the construction presented in Section 2.1.3 is the following: for a function $f : \mathbb{R}_{\geq 0} \times \Omega \to \mathbb{R}^d$ which is right-continuous, adapted and locally bounded, and a sequence of partitions (P_n) with mesh $\to 0$ and $n \to \infty$, the Ito integral may be constructed as the limit *in probability* of Riemann sums of random variables [Friz and Hairer, 2014, Proposition 5.1]:

$$\int_s^t f(r)dB(r) = \operatorname*{plim}_{n\to\infty} \sum_{P_n} f(t_{j-1})(B(t_j) - B(t_{j-1})). \tag{2.16}$$

This definition is usually extended beyond the class of right-continuous f, but this is sufficient to note that the limit is *not* constructed pointwise for each $\omega \in \Omega$ as the limit of the real sequence

$$\text{“} \lim_{n\to\infty} \sum_{P_n} f(t_{j-1}, \omega)(B(t_j, \omega) - B(t_{j-1}, \omega))\text{”}, \tag{2.17}$$

which in general does not exist.

Write ϕ for the solution map (Ito map) defined by Equation 2.16, taking a driving Brownian path $B(\omega)$ to its corresponding Ito solution path $X(\omega)$. Then a consequence of using the weaker probabilistic limit (rather than a pointwise real limit) is that ϕ cannot be made continuous for any choice of norm on the space of Brownian paths (see [Lyons, 1991, Theorem 3] and [Friz and Hairer, 2014, Proposition 1.1]). The question of how $[\int f dB](\omega)$ is related to the Riemann series $\sum_{P_n} f(t_{j-1}, \omega)(B(t_j, \omega) - B(t_{j-1}, \omega))$ along any *particular* sequence of partitions $P^{(n)}$ is not at all clear.

Suppose it were possible to define an integral against Brownian paths pointwise, with a continuous solution map. Then an application for ε-strong simulation suggests itself. Write Φ for this new continuous solution map. Then by simulating a sufficiently close approximation $\hat{B}^n(\omega)$ to the Brownian path $B(\omega)$ (ie. $\hat{B}^n$ is simulated on a sufficiently fine dyadic grid $\mathcal{D}_n$), the continuity of Φ guarantees that the solution paths $\Phi(B^n(\omega))$ and $\Phi(B(\omega))$ can be made arbitrarily close in an appropriate sense. Rough path theory provides a construction for exactly such a Φ, known as the Ito-Lyons map, in a rather general setting. Here we are concerned only with constructing Φ for SDEs driven by Brownian motion, which turns out to be the simplest case.

The reason for this, together with an informal justification for the solution definition (2.12), should be made apparent by the following discussion. Recall that when discussing the Young integral, we quoted the fact that for $f, g : [0, 1] \to \mathbb{R}$ which are

α, β-Hölder continuous respectively with $\alpha + \beta > 1$, the Riemann sums

$$S_n = \sum_{[s,t] \in P^{(n)}} f(s)(g(t) - g(s)) \tag{2.18}$$

converge to a limit independent of $(P^{(n)})$.

The insight of rough path theory in this setting is that the Riemann sums in Equation (2.18) fail to converge for $\alpha, \beta < 1/2$ because the "rectangular approximation" $f(s)(g(t) - g(s))$ for the contribution to the integral between s and t is too loose. Moreover, convergence *can* be ensured for these "rougher" paths by using higher order approximations for the local contributions. The case in which f, g are α-Hölder continuous for some $1/3 < \alpha \leq 1/2$, as holds for two paths of Brownian motion, turns out to be the simplest. It requires only a second order approximation to obtain convergent Riemann-type sums. The reason for this may be illustrated informally as follows:

Suppose initially that $f(t) = h(g(t))$ for some continuously differentiable function h first. Then we might want

$$\begin{aligned} \text{“}\int_s^t f(u)dg(u)\text{”} &= \text{“}\int_s^t \big[f(s) + (f(u) - f(s))\big]dg(u)\text{”} \\ &\approx h(g(s))(g(t) - g(s)) + h'(g(s))\text{“}\int_s^t (g(u) - g(s))dg(u)\text{”} \end{aligned}$$

If we are able to define the *iterated integral* of g against itself, $\int_s^t (g(u) - g(s))dg(u)$, this suggests an altered Riemann-type sum to define the LHS integral. In the case of Brownian paths with $g(u) = B(u)$, it is easy to define this integral: we can simply take

$$\mathbb{B}(\omega; s, t) = \int_s^t (B(u, \omega) - B(s, \omega))dB(u, \omega)$$

to be the usual Ito integral. But instead of defining $\int_0^T h(B(\omega))dB(\omega)$ in the Ito sense, we instead try to define it as the limit as $n \to \infty$ of the sequence

$$S_n(\omega) = \sum_{[s,t]} \Big[h(B(s, \omega))(B(t, \omega) - B(s, \omega)) + h'(B(s, \omega))\mathbb{B}(\omega; s, t)\Big] .$$

The intuition here[2] is that since Brownian motion is α-Hölder continuous for all $1/3 < \alpha < 1/2$, neither $B(t) - B(s) \sim |t - s|^\alpha$ nor $\mathbb{B}(s, t) \sim |t - s|^{2\alpha}$ vanish as the mesh size $|P^{(n)}| \to 0$; hence including the additional iterated integral term is necessary. On the other hand, all higher order terms $\sim |t - s|^{3\alpha}, \sim |t - s|^{4\alpha}, \ldots$ do vanish in the limit, so this single additional term should be sufficient.

[2] made formal in Lemma 9 below; see also [Friz and Hairer, 2014, p. 16].

The rough integral, and rough SDEs

We'll now formally define the rough integral against Brownian motion that was informally described above. In order to do this, we will rely on two crucial facts: first, the already noted α-Hölder continuity of Brownian paths for all $\alpha \in (1/3, 1/2)$; and second, an analogous property of the iterated integrals, which we now establish. We denote the space of α-Hölder functions $[0,T] \to \mathbb{R}^d$ by $C^\alpha([0,T], \mathbb{R}^d)$. Say a function $F : [s,t]^2 \to \mathbb{R}^d$ is in $C_2^{2\alpha}([0,T]^2, \mathbb{R}^d)$ if

$$||F||_{2\alpha} := \sup_{s<t} \frac{||F(s,t)||}{|t-s|^{2\alpha}} < \infty \ ,$$

which can be interpreted as a "diagonal" Hölder condition, in that F is required not to be Hölder continuous in s or t individually, but between s and t.

Recall the classical Kolmogorov continuity theorem:

Theorem 7. *Suppose that the random variable $X : [0,T] \to \mathbb{R}^d$ is such that*

$$\mathbb{E}|X_{s,t}|^\gamma \leq K|t-s|^\beta$$

for some $\gamma > 0$, $\beta > 1$, constant $K > 0$, and all $s, t \in [0,T]$. Then X has a continuous modification, and in particular a modification which is α-Hölder continuous for all $\alpha < (\beta - 1)/\gamma$.

A more general form of this theorem can be used to establish the "diagonal Hölder" condition.

Theorem 8. *Extended Kolmogorov continuity theorem ([Friz and Hairer, 2014, Theorem 3.1]): suppose that $X : [0,T] \to \mathbb{R}^d$, $\mathbb{X} : [0,T]^2 \to \mathbb{R}^d$ are such that*

$$\begin{aligned} \mathbb{E}|X_{s,t}|^\gamma &\leq K_1|t-s|^\beta \ , \\ \mathbb{E}|\mathbb{X}_{s,t}|^{\frac{\gamma}{2}} &\leq K_2|t-s|^{2\beta} \end{aligned}$$

for some $\gamma > 0$, $\beta > 1$, $K_1, K_2 > 0$, and all $s, t \in [0,T]$. Then $(X, \mathbb{X})$ has a modification such that X is α-Hölder continuous, and $\mathbb{X}$ is diagonal 2α-Hölder continuous, for all $\alpha < (\beta - 1)/\gamma$.

The following lemma establishes that Theorem 8 applies to the pair $(B, \mathbb{B})$ for all $\alpha \in (1/3, 1/2)$ almost surely.

Lemma 9. *[Friz and Hairer, 2014, Theorem 3.4] The extended Kolmogorov continuity theorem applies to $(B, \mathbb{B})$ for all $\beta > 1$, with $\gamma = 2(\beta - 1)$.*

$$\begin{aligned} \mathbb{E}|X_{s,t}|^\gamma &\leq K_1|t-s|^\beta \ , \\ \mathbb{E}|\mathbb{X}_{s,t}|^{\frac{\gamma}{2}} &\leq K_2|t-s|^{2\beta} \end{aligned}$$

Writing $\mathcal{C}^\alpha([0,T], \mathbb{R}^d)$ for the space of pairs $(X, \mathbb{X})$ with $X \in C^\alpha([0,T], \mathbb{R}^d)$, $\mathbb{X} \in$

$C_2^{2\alpha}([0,T]^2,\mathbb{R}^d)$, this establishes that, almost surely, rough Brownian paths are in $\mathcal{C}^\alpha$ for all $1/3 \leq \alpha < 1/2$. We now quote a result which avers the existence of the rough integral against Brownian motion for suitable paths.

Again taking $1/3 \leq \alpha < 1/2$, say that $X(\omega) \in C^\alpha([0,T],\mathbb{R}^{d'})$ is *controlled* by $B(\omega)$ if there exists an $X'(\omega) \in C^\alpha([0,T],\mathbb{R}^{d'\times d})$ and $R(s,t;\omega)$ with $||R(s,t;\omega)||_{2\alpha} < \infty$ such that

$$X(t,\omega) - X(s,\omega) = X'(s,\omega)(B(t,\omega) - B(s,\omega)) + R(s,t;\omega).$$

Here X' is an extended notion of the derivative of X. The case $X = F(B)$, $X' = (DF)(B)$ is a concrete example. In general, X' is known as a *Gubinelli derivative* of X and may not be unique. We write $\mathcal{D}^{2\alpha}_{B(\omega)}$ for the space of pairs $(X,X')(\omega)$ controlled by $B(\omega)$.

Theorem 10. *[Friz and Hairer, 2014, Prop. 5.1] Let H_1 be the null set on which Lemma 9 fails, and let $(X,\mathbb{X})(\omega) \in \mathcal{D}^{2\alpha}_{B(\omega)}$ for all ω outside the null set H_2. Let also $(P^{(n)})$ be a sequence of partitions with $|P^{(n)}| \to 0$. Then the sums*

$$S_n(\omega) = \sum_{[s,t]\in P^{(n)}} \big(X(s,\omega)(B(t,\omega) - B(s,\omega)) + X'(s,\omega)\mathbb{B}_{s,t}(\omega)\big)$$

converge to a limit independent of the sequence $(P^{(n)})$ for all $\omega \in (H_1 \cup H_2)^c$. The rough integral *of X against $\boldsymbol{B} = (B,\mathbb{B})$ is then defined by*

$$\int_0^T X(u)d\boldsymbol{B}(u) = \lim_{n\to\infty} S_n.$$

Moreover, we have almost surely the equality

$$\int_0^T f(B(u)))d\boldsymbol{B}(u) = \int_0^T f(B(u))dB(u),$$

where the integral on the left is the rough integral, and that on the right is the Ito integral.

We sketch here the short proof of the almost sure equality of the rough and Ito integrals, since this is the main point of our discussion:

Proof. For any sequence of partitions $P^{(n)}$, the convergence in probability of $\sum_{[s,t]} X(s)(B(t) - B(s))$ implies the convergence almost surely along a sub-sequence of the same, where the particular domain of almost sure convergence depends on the sequence of partitions.

It is sufficient then to show that $\mathbb{E}[(\sum_{[s,t]} X'(s)\mathbb{B}(s,t))^2] \to 0$ as $|P^{(n)}| \to 0$. Then for $P^{(n)} = (t_0,\ldots,t_m)$, writing

$$S_k = \sum_{i=0}^{k-1} X'(t_i)\mathbb{B}(t_i,t_{i+1}))$$

then S_k is a martingale, so we have

$$\begin{aligned}
\mathbb{E}[(\sum X'(s)\mathbb{B}(s,t))^2] &= \mathbb{E}[(\sum (S_k - S_{k-1}))^2] \\
&= \sum \mathbb{E}[(S_k - S_{k-1})^2] \\
&\leq \sup_{s,\omega} |X(s,\omega)|^2 \sum \mathbb{E}[\mathbb{B}(s,t)^2] \\
&\leq C \sum (s-t)^2 \\
&\to 0 \ .
\end{aligned}$$

□

We now quote the main theorem characterising rough solutions to stochastic differential equations (we do not concern ourselves here with the second component of the rough solution described in this result):

Theorem 11. *[Friz and Hairer, 2014, Theorem 9.1] Let $a : \mathbb{R}^d \to \mathbb{R}^d$ be Lipschitz continuous, $b : \mathbb{R}^d \to (\mathbb{R}^d)^2$ three times continuously differentiable. Let also $X(0) \in \mathbb{R}^d$. Then almost surely there is a unique rough solution $(X, \mathbb{X})$ to*

$$dX(t) = a(X(t))dt + b(X(t))dB(t)$$

with the initial condition $X(0) = X(0)$, and its first component X is a solution to the corresponding Ito stochastic differential equation. Moreover, $X(\omega)$ is a continuous function of the driving Brownian motion (see [Friz and Hairer, 2014, Theorem 8.5]).

Finally, the connection to the Euler scheme of Equation 2.12 is drawn in [Friz and Hairer, 2014, Section 8.7]: suppose $(X, f(X)) \in \mathcal{D}_B^{2\alpha}$; then by an analysis of the size of the error in the rough approximation, it can be shown ([Friz and Hairer, 2014, Eqn 8.13]) that

$$X(s) - X(t) = f(X(s))(B(t) - B(s)) + (Df)(X(s))f(X(s))\mathbb{B}(s,t) + o(|t-s|).$$

This idea is developed in Davie [2008] for numerical approximations.

The continuity argument

Recall that earlier in this section, we suggested that by simulating a "sufficiently close" approximation $B^n(\omega)$ to the Brownian path $B(\omega)$, continuity of the Ito-Lyons solution map Φ guarantees that the solution paths $\Phi(B^n(\omega))$ and $\Phi(B(\omega))$ can be made close in an appropriate sense. We now give a little detail on this point.

Let $\mathcal{D}_n = \{t_k^n : k = 0, \ldots, 2^{n-1}+1\}$ be the dyadic rationals of order n, and $W_{i,k}^n$ a system of independent standard normal random variables for $i = 1, \ldots, d'$, $n = 1, \ldots, \infty$, $k = 1, \ldots, 2^{n-1}$. The Lévy construction of Brownian motion ([Mörters and Peres,

2010, Section 1.1.2]) uses the approximating sequence (B^n) where

$$B^n(t^n_{2k+1}) = \frac{1}{2}\left(B^{n-1}(t^{n-1}_k) + B^{n-1}(t^{n-1}_{k+1})\right) + 2^{-\frac{n+1}{2}} W^n_{k+1},$$

(together with linear interpolation between these points, which we need not worry about). The right notion of "sufficiently close" then turns out to be "using B^n with n sufficiently large that the constants K_1, K_2 in the Extended Kolmogorov continuity theorem can be sampled explicitly". Constructing these upper bounds K_1 and K_2 is where much of the detailed algorithmic constructions fall, which we do not develop here. [Blanchet et al., 2017, Sections 4-5] contain the relevant details.

The main theorem may be summarised in the following way:

Theorem 12. *[Blanchet et al., 2017, Theorems 2.1, 2.2] i) It is possible to sample explicitly and exactly the constants $K_1(\omega)$, $K_2(\omega)$ in the extended Kolmogorov criterion so that*

$$\sup_{s,t\in[0,T]} ||B_{s,t}(\omega)||_\infty \leq K_1(\omega)|t-s|^\alpha \text{ and}$$
$$\sup_{s,t\in[0,T]} ||\mathbb{B}_{s,t}(\omega)||_\infty \leq K_2(\omega)|t-s|^{2\alpha},$$

jointly *with an n-th dyadic approximation $(B^n(s,\omega))_{0\leq s\leq T}$ to a Brownian path for any n sufficiently large.*
ii) Let $\hat{X}^n(\omega)$ be the corresponding approximation to $X(\omega)$ obtained by using the Euler scheme, and let $\beta \in (1-\alpha, 2\alpha)$. Suppose that

$$||a||_\infty, ||\nabla a||_\infty, ||b||_\infty, ||b'||_\infty, ||b''||_\infty, ||b'''||_\infty \leq M$$

*for some constant M. Then it is possible to find a constant $G(\omega)$ in terms of K_1, K_2 and M, which does **not** depend on n, such that*

$$\sup_{t\in[0,T]} ||X(t,\omega) - \hat{X}^n(t,\omega)||_\infty \leq 2^{-n(2\alpha-\beta)} G(\omega)$$

iii) As an immediate corollary to ii), given $\varepsilon > 0$ and having sampled $G(\omega)$, it is possible to choose an n so that

$$\sup_{t\in[0,T]} ||X(t,\omega) - \hat{X}^n(t,\omega)||_\infty \leq \varepsilon.$$

As an immediate consequence, given any finite sequence $\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_m > 0$, and having sampled $G(\omega)$, it is possible to find a sequence of positive integers $n_1, \ldots, n_m$ so that condition iii) is satisfied with each pair (ε_k, n_k), so this theorem outlines a true ε-strong algorithm in the sense of section 2.2.1.

The full procedure is described in Algorithm 5.

Algorithm 5 Epsilon-strong simulation via rough paths (based on Blanchet et al. [2017]

Given $\varepsilon > 0$, and M as above:

1. Simulate N sufficiently large that the Hölder continuity constants $||X||_\alpha, ||A||_{2\alpha}$ can be bounded by K_1, K_2 respectively, together with K_1, K_2 themselves, and also $\{W_k^n : n = 1, \dots, N\}$.
2. Calculate $G(K_1, K_2)$, and if necessary simulate $N_2 \geq N$ such that

$$G \cdot (2^{-N_2})^{2\alpha - \beta} < \varepsilon,$$

   along with extra $W_{i,k}^n$ for $n = 1, \dots, N_2$.
3. Form the discrete Brownian path B^{N_2} using the W_k^n, via the Levy construction.
4. Pass B^{N_2} through the discretisation scheme (2.14) to obtain the ε-strong approximation approximation $\tilde{X}^\varepsilon$.

Chapter 3

Multilevel splitting and Sequential Monte Carlo

3.1 Introduction

This chapter begins by introducing rare event estimation, and in particular the *multilevel splitting* (MLS) family of algorithms, which provide algorithmic framework for rare events we consider in Chapter 4. The unbiasedness of the algorithms is discussed, along with variants and some properties of interest.

We note that one variant of MLS, *fixed effort* splitting, is a Sequential Monte Carlo (SMC) algorithm. Since in chapter 5 we consider a different rare event algorithm which falls into this framework, and since the SMC literature provides various guarantees of useful properties such as unbiasedness and central limit theorems, we take some space to describe SMC algorithms more generally.

SMC algorithms may be viewed as particle approximations of mathematical models known as Feynman-Kac models. Since we use the Feynman-Kac framework extensively in our exposition and analysis in chapters 5 and 6, we conclude by introducing these models and connecting them to SMC and rare event estimation.

3.2 Splitting for Rare Events

3.2.1 Overview

Rare events are those which have (very) low probability of occurrence. Estimating the probability of rare events is important, among other places, throughout the natural and social sciences; see, for example, Rubino and Tuffin [2009, Part II] for a broad range of applications. The case of interest in this thesis is that where the rare event corresponds to a continuous-time Markov process hitting a particular "rare" set of interest before (or at) a specified stopping time. (Such rare event problems are sometimes known as *dynamic* rare events, as distinguished from *static* rare events for which one considers

whether the whole path of X falls inside a specified rare set; this distinction is drawn for example in Johansen et al. [2006].)

We consider two types of stopping time. In chapter 4, the stopping time is the hitting time for the process for another positive recurrent set. In chapters 5 and 6, we also consider the case where the stopping time is simply a chosen constant. These settings, especially the former, have attracted considerable attention in the literature, and general solutions centre around simulation-based methods.

The principal approaches for estimation in these settings fall into two broad categories: *importance sampling* and *splitting*. In importance sampling, one simulates from a process for which the event of interest is more likely to occur, and corrects for the change of sampling distribution using importance weights. With splitting methods, trajectories which approach the rare set (in an appropriate sense) are replicated (or *split*) to allow lower-variance estimation of the target probability. This chapter and the following are concerned with splitting methods, in particular with implementing such methods with no bias for a broad class of continuous-time processes. Existing methods depend upon time-discretisation and hence introduce a difficult to quantify bias. We show in chapter 4 that the adaptation of ideas from the field of ε-strong simulation to this context allows this bias to be avoided.

Let $(X(t) : t \geq 0)$ be a continuous-time Markov process in $\mathbb{R}^d$, and let $A, B \subset \mathbb{R}^d$ be disjoint sets, with A positive recurrent for X. The problem of interest is that of efficiently estimating the probability that X reaches set B before set A when this probability is very small. That is, writing τ_S for the first hitting time of a set S, the objective is to estimate

$$p = \mathbb{P}(\tau_B < \tau_A) \ll 1.$$

The assumption that $p \ll 1$ rules out direct Monte Carlo estimation, since the computational cost of generating the event $\{\tau_B < \tau_A\}$ enough times to get a reliable estimate will be impractically high. As noted in 3.3.1, the relative variance of the naive estimator obtained from N Monte Carlo simulations is $p(1-p)/N \cdot p^{-2} \approx 1/(Np)$ for p sufficiently close to 1, suggesting that a large multiple of $p^{-1} \gg 1$ trials is needed to get a reasonable variance.

Multilevel splitting (MLS) is a popular algorithm based on targeting the rare event via a sequence of more likely events. The idea goes back to the 1951 paper Kahn and Harris [1951] (which in turn attributes the idea to von Neumann), discussing an application to the transmission of particles through an impeding barrier in the context of nuclear shielding. The method is to choose a sequence of nested sets $B_1 \supset B_2 \supset \cdots \supset B_m = B$, all disjoint from A, and use a particle system to sequentially estimate $\mathbb{P}(\tau_{B_i} < \tau_A)$. Starting with a particle system of a large enough size, N, a reasonable fraction will reach B_1, allowing an estimate of $p_1 = \mathbb{P}(\tau_{B_1} < \tau_A)$. Then, by branching (or "splitting") those which do into R_i copies, a healthy population can be maintained to estimate the subsequent probabilities (as explained below).

Splitting algorithms have been independently rediscovered many times and in many variants since the work of Kahn and Harris [1951]. Prominent examples include the *repetitive simulation trials after reaching thresholds* (RESTART) algorithm of Villén-Altamirano and Villén-Altamirano [1994], developed for modelling packet loss probabilities in telecommunications, and the *pruning-enriched Rosenbluth method* (PERM) of Grassberger [1997] for simulating polymer chains.

The basic MLS algorithm we present as Algorithm 6 in Section 3.2.2 is that found in Garvels [2000], in which various implementation issues such as the choice of levels and importance function are also addressed. The unbiasedness of the algorithm for discrete-time processes is shown rigorously in Amrein and Künsch [2011], which identifies and resolves an issue in the original argument of Garvels [2000]. The construction of confidence intervals and the optimal choice of tuning parameters under cost constraints are addressed in Lagnoux-Renaudie [2006, 2008]. A characterization of the asymptotic properties of this algorithm, including a central limit theorem, are given in Del Moral and Lezaud [2006].

Choosing the nested sets and other parameters of the algorithm to maintain a particle population of stable size, rather than one which dies out or explodes, can be difficult. One practical variant which removes the difficulty of choosing the *splitting ratios* R_i in advance is that of Lagnoux-Renaudie [2009], in which a first particle system is used to estimate the R_i, and a second system uses these estimated values to estimate p. An alternative idea is to construct the levels B_i adaptively, for example via the scheme of Cérou and Guyader [2007]. A generalisation of this scheme has recently been shown be unbiased in Bréhier et al. [2016].

The proof of unbiasedness in Amrein and Künsch [2011] also holds for a variant in which the initial system of N particles is kept at fixed size by sampling new trajectories uniformly at random (with replacement) from the surviving trajectories at each level. This variant is a type of Sequential Monte Carlo method and can be understood within the framework of Del Moral [2004]; this approach is also discussed in Garvels [2000] and elsewhere under the name of *fixed effort splitting*. It is presented as Algorithm 7 in Section 3.2.2.

In this version the sets B_i are still chosen in advance, but the number of particles is fixed at N for the duration of the algorithm. Rather than independently "splitting" each path which survives to B_i into a pre-determined number of offspring, exactly N particles are resampled (i.e. sampled with replacement) from among the surviving particles. This removes the difficulty of choosing a suitable splitting ratio in order to arrive at a stable population size, but it is more difficult to understand the variance properties of this algorithm and even the asymptotic variance expression is somewhat more complex than that of the simple algorithm.

Other resampling schemes have also been considered. For example, Amrein and Künsch [2011] describes a variant in which at each level, new trajectories are sampled uniformly

at random from the successful pool and propagated until fixed number of successes at the next level are achieved (in this case, they demonstrate that unbiasedness continues to hold). A very similar idea is the *Alive particle filter* for SMC algorithms with $\{0, 1\}$-valued potential functions of Del Moral et al. [2015]. These also provide unbiased estimates of normalizing constants the rare event probability. We did not experience difficulties with extinction of the particle system in our experiments, and in this thesis we concentrate only on the basic fixed-splitting and fixed-resampling schemes already described. However, incorporating this approach within the exact MLS framework that we present would be interesting because it would automatically mitigate the influence of poorly chosen intermediate levels, though at the cost of further randomizing the computational cost.

Related to this last point, we observe explicitly that all these splitting-type rare event algorithms have a random computational cost, since trajectories are sampled not over a fixed time interval but for as long as necessary until one or another set is hit. (Indeed this is true also of the naive Monte Carlo rare event estimate described at the beginning of this section). Several approaches to reducing the cost associated with *long excursions* (particle trajectories which take a very long time to reach $B_i \cup A$) have been developed.

One group of techniques known as *truncation* methods, neatly summarised in [L'Ecuyer et al., 2007, Section 1.2.2], relies on killing trajectories which reach B_i for some i and then return back to $B_{i-\beta}$ for some integer β. The most straightforward implementation simply kills all trajectories which fall back down β levels. Naturally this introduces a bias into the estimator. More sophisticated variations, known as *Russian roulette* methods, restore unbiasedness by a combination of killing trajectories which fall back β levels with a certain probability, and assigning appropriate weights to the particles. Detailed examples of such approaches, together with proofs of their unbiasedness, may be found in [L'Ecuyer et al., 2007, Section 2.7].

The trade-off in using truncation methods is increased variance in the estimator, either from introducing random particle weights, from inducing more variance in the number of simulated chains, or by inducing stronger correlations between the particle trajectories. An additional limitation is that truncation methods remain vulnerable to trajectories which take long excursions without crossing either up or down levels.

An alternative approach is to impose a strict limit $t^{\max}$ on the length of time for which the user is willing to wait for a trajectory to reach $B_i \cup A$, having reached B_{i-1}. Again, the simplest approach of simply killing trajectories which fail to achieve this before time $t^{\max}$ introduces bias into the estimator. This is known as *user-impatience bias*, and is discussed for example in Thönnes [1999] in the context of the coupling-from-the-past technique for perfect simulation of Markov chains.

In the wider SMC context (see section 3.3), a more careful alternative to killing trajectories which have run for too long may be found in *Anytime* Monte Carlo techniques, described for Sequential Monte Carlo in Paige et al. [2014] and Murray et al. [2021].

Rather than being assigned a fixed number of particles and requiring a random runtime, such anytime algorithms start with a fixed computing budget. They maintain a running estimate, which improves as more computation is carried out within the fixed budget. Moreover, like the conventional MLS Algorithm 6 but unlike its SMC alternative Algorithm 7, there is no need to wait for all particles to reach a certain level before resampling.

Paige et al. [2014] modifies the standard Sequential Monte Carlo method (Algorithm 10) in such a way that there is no need to wait for all N sampling steps to terminate at steps (1) and (4a). Instead, samples are split into multiple copies immediately as they arrive, with the number of copies and assignation of weights to each copy chosen in such a way as to ensure unbiasedness of the resulting estimator. Applied to SMC-Multilevel Splitting (Algorithm 7), this results in something similar to conventional MLS (Algorithm 6), although the details of splitting numbers differs considerably.

Murray et al. [2021] shows that if simulating from q_k in Algorithm 10 (SMC) can be too expensive, any maximum simulation time $t^{\max}$ may be stipulated, and approximate simulations from q_k obtained at $t^{\max}$, with exact simulation from q_k guaranteed in the limit $t^{\max} \to \infty$. However, this does not appear to be directly applicable to splitting for rare events, since the unnormalised importance weights at level i for Algorithm 7 are simply 0 or 1 according to whether the particle reached level i or not, whereas the approximate samples from q_k are not guaranteed to lie in $A \cup B_i$.

3.2.2 Multilevel splitting

Multilevel splitting requires the specification of a sequence of nested events $\mathbb{R}^d \supset B_1 \supset B_2 \supset \cdots \supset B_m = B$ in such a way that the probabilities

$$p_i := \begin{cases} \mathbb{P}(\tau_{B_1} < \tau_A) \ , & i = 1 \\ \mathbb{P}(\tau_{B_i} < \tau_A \mid \tau_{B_{i-1}} < \tau_A) \ , & 2 \leq i \leq m \end{cases}$$

are large relative to p, and may consequently be estimated more efficiently.

In order to do this, it is convenient to assume the existence of a continuous function $\xi : \mathbb{R}^d \to \mathbb{R}$ of which the boundaries of A and B_i are *level sets*. That is, we suppose that there are real numbers $z_A < z_1 < z_2 < \cdots < z_m = z_B$ such that

$$A = \xi^{-1}\left((-\infty, z_A]\right) \ , \ B_i = \xi^{-1}\left([z_i, \infty)\right) \ ,$$

with boundaries $\partial A = \xi^{-1}(\{z_A\}), \partial B_i = \xi^{-1}(\{z_i\})$. The function ξ has numerous names in the literature, including the *reaction co-ordinate*, and is typically defined such that higher values represent locations closer to B, as it is throughout this paper.

We take $(X(t) : t \geq 0)$ to be a continuous-time Markov process with almost surely continuous sample paths (the central example of diffusion processes will be the focus of our algorithmic constructions). We take $X(0)$ to be distributed according to an initial

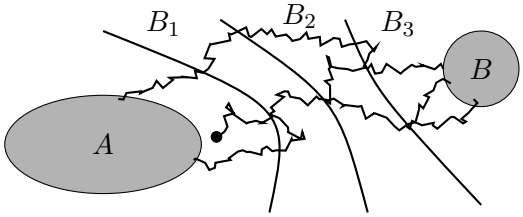


Figure 3.1: An illustration of multilevel splitting for a single particle system. The particle begins at the black node, and each branch splits into two i.i.d copies upon reaching a set B_i for the first time. Branches which reach A terminate there. Those which reach B are used to form an estimate of the rare event probability $\mathbb{P}(\text{process hits } B \text{ before } A)$.

distribution λ, where the support of λ is contained in $\xi^{-1}([z_A, z_B))$ (so X may begin on the boundary ∂A of A; this choice is made for notational convenience, and the support of λ may be taken instead to be all of $\mathbb{R}^d$ with only minor modifications). Except where necessary, dependence upon λ will be suppressed from our notation.

Write τ_i for τ_{B_i}, and define $\sigma_i = \tau_A \wedge \tau_i$ to be the first hitting time of $A \cup B_i$ for X. Note that τ_A, τ_i, σ_i are equivalently the hitting times for the one-dimensional process $[\xi(X)]_s = \xi(X_s)$ of $\{z_A\}, \{z_i\}$ and $\{z_A, z_i\}$ respectively. In an algorithmic implementation, it is often more convenient to use $\xi(X)$ to decide when a level has been crossed if X has dimension greater than one. We remark that the process $\xi(X)$ is not in general a Markov process, so it is not possible to reduce all problems of this form to the univariate Markovian process setting.

The idea of MLS is to run a particle system in which each particle splits into several, say R_i, copies immediately upon first reaching a set boundary ∂B_i; alternatively, it is terminated upon reaching ∂A. The splitting is managed so that a healthy system of particles is available for estimating each p_i. Since $p = \prod_{i=1}^m p_i$, if we have an estimator $\hat{p}_i$ for each p_i, then $\prod_{i=1}^m \hat{p}_i$ is a natural estimator for p. The $\hat{p}_i$ are defined as follows: suppose that we begin with N_0 particles, of which N_1 reach level B_1 before A. Then we estimate p_1 with

$$\hat{p}_1 = \frac{N_1}{N_0}.$$

Suppose that each of these N_1 surviving particles is split into a constant number R_1 of copies immediately upon reaching B_1, and that N_2 of the branched trajectories reach B_2 before A. Then we estimate p_2 with

$$\hat{p}_2 = \frac{N_2}{R_1 N_1}.$$

Continuing in this way, one obtains a sequence of estimators $\hat{p}_1, \ldots, \hat{p}_m$ of $p_1, \ldots, p_m$ which may be multiplied together to give the estimate

$$\hat{p} = \prod_{i=1}^m \hat{p}_i = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}$$

for p.

This estimator is unbiased; a proof may be found in [Amrein and Künsch, 2011, Section 3, Proposition 3.1]. We quote this result and its proof shortly, since we use a similar argument in Proposition 4.4.3 in Section 4.4.3 to establish the unbiasedness of the splitting-type algorithm which we develop later.

Provided that z_i and R_i are well-chosen, this estimate can be much more efficient than naïve Monte Carlo estimation. For instance, choosing z_i such that $p_i = p^{1/m}$, and choosing R_i with small variance (but i.i.d for each particle) such that $\mathbb{E}[R_i] = p_i^{-1}$, the relative variance of $\hat{p}$ is reduced to approximately $\mathcal{O}(p^{-1/m})$. See Glasserman et al. [1999]; Lagnoux-Renaudie [2006], for more detail on parameter choice and asymptotic variance calculations.

It is convenient to work with the discrete-time pair-process $U_i = (\sigma_i, X(\sigma_i))$, $i = 1, \dots, m$, i.e. the values of X at its splitting times together with the splitting times themselves. (Note that if X hits A before B_i, then $\sigma_j = \tau_A$ for every $j \geq i$, so $(\sigma_i, X(\sigma_i))$ is an absorbing state). Let $\mathcal{S}$ be the Borel sigma algebra associated with $\mathbb{R}_{\geq 0} \times \mathbb{R}^d$, and let $M_i : (\mathbb{R}_{\geq 0} \times \mathbb{R}^d) \times \mathcal{S} \to [0, 1]$ denote the Markov kernels of this discrete-time process. Finally, we define potential functions $G_i : \mathbb{R}_{\geq 0} \times \mathbb{R}^d \to \{0, 1\}$ on the state space of this pair process as indicators of the sets B_i for the process (U_i):

$$G_i(t, x) = \begin{cases} 1, & \text{if } \xi(x) \geq z_i, \\ 0, & \text{otherwise.} \end{cases}$$

A full description of multilevel splitting is given in Algorithm 6.

Algorithm 6 Idealised Multilevel Splitting

Given λ together with G_i, M_i for $i = 1, \dots, m$, an initial number of particles N_0, and splitting ratios $R_1, \dots, R_{m-1}$:

1. For each $j = 1, \dots, N_0$, draw independently $X_1^j(0) \sim \lambda$ and $U_1^j \sim M_1\left(\left(0, X_1^j(0)\right), \cdot\right)$.
2. Let $S_1 = \{U_1^j : G_1(U_1^j) = 1\}$ be a list of the the surviving paths, and set $N_1 = |S_1|$.
3. For $i = 2, \dots, m$:
   (a) If $N_{i-1} = 0$, return $\hat{p} = 0$.
   (b) Else given $S_{i-1} = \{\bar{U}_{i-1}^j\}_{j=1}^{N_{i-1}}$, for all $(j, k) \in \{(j', k') : 1 \leq j' \leq N_{i-1}, 1 \leq k' \leq R_{i-1}\}$ sample independently $U_i^{j,k} \sim M_i(\bar{U}_{i-1}^j, \cdot)$.
   (c) Let $S_i = \{U_i^{j,k} : G_i(U_i^{j,k}) = 1\}$, and $N_i = |S_i|$.
4. Return
$$\hat{p} = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}.$$

Proposition 1. *[Amrein and Künsch, 2011, Appendix B.2]*

$$\mathbb{E}[\hat{p}] = p.$$

Proof. We consider the discrete-time Markov process $(U_i)_{i=0}^m$. Let M_i denote the transition kernels of this process at time i. Let also $M_{i:j} = M_j \circ M_{j-1} \circ \cdots \circ M_{i+1}$ denote the

elements of the associated two-parameter dynamic semigroup, describing the evolution of the process from time i to j. Note that for every i, $\Delta = \mathbb{R} \times A$ is an absorbing set for M_i since if $X(\sigma_i) \in A$, $\sigma_j = \sigma_i$ for all $j > i$.

Define $\mathcal{F}_0$ to be the sigma algebra generated by $\{U_0^j : j = 1, \ldots, N_0\}$, $\mathcal{F}_k = \mathcal{F}_{k-1} \vee \Sigma(U_k^j : 1 \leq j \leq N_k)$ for $k = 2, \ldots, m$, so that $(\mathcal{F}_k)_{k=1}^m$ is the natural filtration of the process $(U_i)_{i=0}^m$.

The following recursion holds for the number of particles successfully reaching B_i given $\mathcal{F}_{i-1}$, which is immediate from the definition of M_i: that for any function $h_k : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ which is equal to 0 on Δ:

$$\mathbb{E}\left[\sum_{j=1}^{R_{i-1}N_{i-1}} h_i\left(U_i^j\right) \middle| \mathcal{F}_{i-1}\right] = R_{i-1} \sum_{k:\tilde{G}_{i-1}(U_{i-1}^k)=1} \int h_i(u) M_i\left(U_{i-1}^k, du\right). \tag{3.1}$$

(taking $R_0 = 1$) which follows since each U_i^j which is a descendent of any particular U_{i-1}^k has the same marginal law.

Recall the estimators $\hat{p}_i$ for p_i:

$$\hat{p}_i = \begin{cases} \frac{N_1}{N_0}, & i = 1, \text{ and} \\ \frac{N_i}{R_{i-1}N_{i-1}}, & 2 \leq i \leq m. \end{cases}$$

It holds that for $1 \leq k$,

$$\mathbb{E}\left[\prod_{i=k}^m \hat{p}_i \middle| \mathcal{F}_{k-1}\right] = \frac{1}{N_{k-1}} \sum_{j:\tilde{G}_{k-1}(U_{k-1}^j)=1} \left(1 - M_{(k-1):m}\left(U_{k-1}^j, \Delta\right)\right),$$

by backwards induction on k, starting with the case $k = m$. Note that for the case $k = 1$, each term in the sum on the RHS is the probability that a particle with a given starting value successfully reaches B before A. The result is then obtained upon taking a further expectation over the starting value.

The case $k = m$:

$$\begin{aligned} \hat{p}_m &= \frac{N_m}{R_{m-1}N_{m-1}} \\ &= \frac{1}{R_{m-1}N_{m-1}} \sum_{j=1}^{R_{m-1}N_{m-1}} \tilde{G}_m\left(U_m^j\right) \end{aligned}$$

and the result then follows from taking the conditional expectation, and applying (3.1) with $h_i\left(\sigma, X\left(\sigma\right)\right) = \mathbb{I}_{B_j}\left(X(\sigma)\right)$.

Now supposing the result holds for $k+1$, we show that it holds also for k. We have

the following chain of equalities (with the convention that $G_0(U_0^j) = 1$):

$$\mathbb{E}\left[\prod_{i=k}^{m}\hat{p}_i\middle|\mathcal{F}_{k-1}\right] = \mathbb{E}\left[\hat{p}_k\mathbb{E}\left[\prod_{i=k+1}^{m}\hat{p}_i\middle|\mathcal{F}_k\right]\middle|\mathcal{F}_{k-1}\right] \tag{3.2}$$

$$= \mathbb{E}\left[\frac{N_k}{R_{k-1}N_{k-1}}\cdot\frac{1}{N_k}\sum_{j:\tilde{G}_k(U_k^j)=1}\left(1 - M_{k:m}\left(U_k^j, \Delta\right)\right)\middle|\mathcal{F}_{k-1}\right] \tag{3.3}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(U_{k-1}^j)=1}\left(\int\left(1 - M_{k:m}\left(u, \Delta\right)\right)M_k\left(U_{k-1}^j, du\right)\right) \tag{3.4}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(U_{k-1}^j)=1}\left(1 - \int M_{k:m}\left(u, \Delta\right)M_k\left(U_{k-1}^j, du\right)\right) \tag{3.5}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(U_{k-1}^j)=1}\left(1 - M_{(k-1):m}\left(U_{k-1}^j, \Delta\right)\right) \tag{3.6}$$

where (3.2) follows from the tower rule, and noting that $\hat{p}_k$ is $\mathcal{F}_k$-measurable; (3.3) from the induction hypothesis; (3.4) from (3.1); (3.5) from expanding and integrating over $M_k(U_{k-1}^j, du)$; and (3.6) from the semigroup property.

Moreover, putting $k = 1$ we have

$$(3.6) = \frac{1}{N_{k-1}}\sum_{j=1}^{N_0}\left(1 - M_{0:m}\left(U_0^j, \Delta\right)\right)$$

Since $\left(1 - M_{0:m}\left(U_0^j, \Delta\right)\right) = \mathbb{P}\left(\tau_B < \tau_A | X(0)\right)$, we see that

$$\begin{aligned}\mathbb{E}\left[\hat{p}\right] &= \mathbb{E}\left[\prod_{i=1}^{m}\hat{p}_m\right] \\ &= \mathbb{P}(\tau_B < \tau_A)\end{aligned}$$

as desired. □

A small modification to Algorithm 6 gives a variant commonly known as *fixed effort splitting*, which can be viewed as a Sequential Monte Carlo algorithm. This connection has been exploited previously by Cérou et al. [2006] (note that these algorithms are distinct from those which use SMC to approximate static rare events which depend upon the trajectory of a process only over a fixed time interval, see Cérou et al. [2012]; Del Moral and Garnier [2005]; Johansen et al. [2006]). In this variant, a number of particles N to be maintained throughout is chosen in advance, and the splitting of each individual surviving particle is replaced with resampling from the set of surviving particles. This is useful in that the procedure does not requite the specification of tuning parameters $R_1, \ldots, R_{m-1}$. A full description is given in Algorithm 7.

The particle system in Algorithm 7 has the structure of an SMC sampler (see sec-

Algorithm 7 Idealised MLS via SMC

Given λ together with G_i, M_i for $i = 1, \ldots, m$, and a fixed number of particles N:

1. For each $j = 1, \ldots, N$, draw independently $X_1^j(0) \sim \lambda$ and $U_1^j \sim M_1\left(\left(0, X_1^j(0)\right), \cdot\right)$.
2. Record $N_1 = \sum_{j=1}^N G_1(U_1^j)$.
3. For $i = 2, \ldots, m$:
   (a) If $N_{i-1} = 0$, return $\hat{p}^{\mathrm{SMC}} = 0$.
   (b) For $j = 1, \ldots, N$ sample independently
   $$U_i^j \sim \frac{\sum_{k=1}^N G_{i-1}(U_{i-1}^k) M_i(U_{i-1}^k, \cdot)}{\sum_{k=1}^N G_{i-1}(U_{i-1}^k)}.$$
   (c) Record $N_i = \sum_{j=1}^N G_i(U_i^j)$.
4. Return
   $$\hat{p}^{\mathrm{SMC}} = \prod_{i=1}^m \left(\frac{N_i}{N}\right).$$

tion 3.3.3 below), with step 3b combining multinomial resampling with propagation via M_i. The unbiasedness proof of Amrein and Künsch [2011] also applies to Algorithm 7. An alternative but more general point of view from which this derives is the general theory of SMC estimators in the Feynman-Kac framework described above; see in particular [Del Moral, 2004, Theorem 7.4.2].

The rare event probability of interest can be interpreted as the normalizing constant of an excursion-valued Feynman-Kac flow in the sense of [Del Moral, 2004, Section 12.2.6]. This flow has transition densities specified in terms of the underlying dynamics and stopping times, and zero-one-valued potential functions indicate whether crossing occurs into B_i or A at each level. Consequently, the SMC variant of MLS admits an interpretation as a mean field approximation of this flow and the estimator benefits from the usual theoretical analysis of these, see Del Moral [2004]. This includes inheriting a strong law of large numbers, a central limit theorem and a proof of unbiasedness. This theory does not apply directly, however, to the estimator of Algorithm 6.

3.3 Sequential Monte Carlo

As we have noted, many (though not all) popular algorithms for rare event estimation can be understood as examples of Sequential Monte Carlo methods. These methods have been closely studied since their introduction in Gordon et al. [1993], and their properties have been well-described across a large body of literature. Moreover, analysis of rare event algorithms which fall under the SMC framework is often presented in the language and context of this wider SMC literature. For this reason, we begin with a brief description of these methods, and a statement of some important results concerning them. The material in this section follows the treatments in Doucet and Johansen [2011] and the recent book Chopin and Papaspiliopoulos [2020].

3.3.1 Monte Carlo methods

Let η be the density function of a distribution over the sample space $\mathcal{X}$, and let X be distributed according to η. Our object of interest is the expected value of a function $\varphi : \mathcal{X} \to \mathbb{R}$ of X:

$$I(\varphi) = \mathbb{E}_\eta[\varphi(X)] = \int \varphi(x)\eta(x)dx.$$

Often it is not possible to calculate this integral directly, but it is possible to sample i.i.d. instances $X^1, \ldots, X^N \sim \eta$ for an integer N. In this case, the Monte Carlo estimator of $I(\varphi)$ is

$$\hat{I}^{\mathrm{MC}}(\varphi) = \frac{1}{N}\sum_{i=1}^{N}\varphi(X^i) = \int \varphi(x)\hat{\eta}(x)dx \tag{3.7}$$

where

$$\hat{\eta}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X^i}(x).$$

This can be understood as approximating η by the discrete mass function $\hat{\eta}$ using equally-weighted point masses at each sampled location X^i, and taking an expectation with respect to $\hat{\eta}$. The attractive properties of the estimator $\hat{I}^{\mathrm{MC}}$ are that it is unbiased, and that its variance

$$\mathrm{Var}[\hat{I}^{\mathrm{MC}}(\varphi)] = \frac{1}{N}\left(\mathbb{E}[\varphi(X)^2] - I(\varphi)^2\right)$$

is of order $\mathcal{O}(1/N)$ regardless of the dimension of the sample space $\mathcal{X}$.

Algorithm 8 Monte Carlo estimation

1. For $i = 1, \ldots, N$, sample $X^i \sim \eta$.
2. Estimate $\hat{I}^{\mathrm{MC}}(\varphi) = \frac{1}{N}\sum_{i=1}^{N}\varphi(X^i)$

It is immediately evident, however, that this type of estimator is not suited to a rare event problem such as estimating the probability $p = \mathbb{P}(X \in A \subset \mathcal{X})$ where $p \ll 1$, for example. (This corresponds to the choice $\varphi(x) = \mathbb{1}(A)$.) The relative variance (ie. the ratio of the variance to the squared mean) of the estimator is $p(1-p)/N \cdot p^{-2} \approx 1/(Np)$ for p sufficiently close to 1. This suggests that a prohibitively large multiple of $p^{-1} \gg 1$ trials is needed for an estimate with a reasonable variance.

This is one example among several of the difficulties posed by this type of straightforward Monte Carlo estimation. Other difficulties may be that η requires too much computing power to sample from in a reasonable algorithm, or is intractable to sample from at all. All these issues may be dealt with by Sequential Monte Carlo methods.

3.3.2 Sequential Importance Sampling

One way to begin motivating Sequential Monte Carlo is to consider targeting the "inaccessible" density η (in one of the senses described above) using a sequence of bridging

densities $(\eta_k)_{k=1}^n$, with $\eta_k : \mathbb{R}^k \to \mathbb{R}^+$, such that the n-th marginal of η_n coincides with η:

$$\int_{x_1,\ldots,x_{n-1}} \eta_n(x_{1:n}) = \eta(x_n),$$

and the η_k have the following sequential structure:

$$\begin{aligned}\eta_n(x_{1:n}) &= \eta_n(x_n|x_{1:n-1})\eta_{n-1}(x_{1:n-1}) \\ &= \eta_1(x_1)\prod_{k=2}^{n}\eta_k(x_k|x_{1:k-1}).\end{aligned} \tag{3.8}$$

We could then obtain samples from η by drawing $X_1^1, \ldots, X_1^N \sim \eta_1$, and sequentially $X_k^i \sim \eta_k(\cdot|X_{1:k-1}^i)$ for $i = 1, \ldots, N$ and $k = 1, \ldots, n$. Finally, the X_n^i can be plugged into the estimator defined in (3.7). This is still an inflexible algorithm, however.

One generalisation is to allow that it may not be possible to sample directly from the conditional densities $\eta_k(\cdot|x_{1:k-1})$. Instead, we relax the requirement (3.8) on the η_k, and instead specify a sequence of proposal distributions $(q_k)_{k=1}^n$, $q_k : \mathbb{R}^k \to \mathbb{R}^+$ such that

$$(q_k(x_{1:k}) = 0) \implies (\eta_k(x_{1:k}) = 0)$$

and which satisfy the same sequential factorisation

$$q_n(x_{1:n}) = q_1(x_1)\prod_{k=2}^{n} q_k(x_k|x_{1:k-1}).$$

Sampling directly from the conditional densities $\eta_k(\cdot|x_{1:k-1})$ can then be replaced by *importance sampling* from the $q_k(\cdot|x_{1:k-1})$. That is, at time 1 one draws a sample $X_1^1, \ldots, X_1^N \sim q_1$ and calculates the associated unnormalised and normalised *importance weights* $w(X_1^i) = \frac{\eta_1}{q_1}(X_1^i)$, $W_1^i = w(X_1^i)/\left(\sum_{j=1}^N w(X_1^j)\right)$. The motivation here is that since

$$I_1(\varphi) = \mathbb{E}_{\eta_1}[\varphi(X)] = \mathbb{E}_{q_1}[\varphi(X)\frac{\eta_1}{q_1}(X)],$$

then $\sum_{i=1}^N W_1^i\varphi(X_1^i)$ is an unbiased estimator for $I_1(\varphi)$.

Likewise, one could continue to sample $X_k^i \sim q_k(\cdot|X_{1:k-1}^i)$ and calculate the associated unnormalised weights

$$w(X_{1:k}^i) = \frac{\eta_k}{q_k}(X_{1:k}^i) = \frac{\eta_k(X_{1:k}^i)}{\eta_{k-1}(X_{1:k-1}^i)q_k(X_k^i|X_{1:k-1}^i)}w(X_{1:k-1}^i)$$

and the normalised weights $W_k^i = w(X_k^i)/\left(\sum_{j=1}^N w(X_k^j)\right)$ for $i = 1, \ldots, N$ and $k = 1, \ldots, n$. Finally, one would so obtain the unbiased estimate $\sum_{i=1}^N W_n^i\varphi(X_n^i)$ for $I_n(\varphi)$.

Since doing so requires being able to evaluate η_k pointwise, we gain further flexibility by allowing the densities η_k to be known only up to a normalising constant. That is, rather than $(\eta_k)_{k=1}^n$, we use a bridging sequence $(\gamma_k)_{k=1}^n$ with $\gamma_k = Z_k\eta_k$, where $Z_k =$

$\int_{x_{1:k}} \gamma_k(x_{1:k})$ is the normalising constant. This more general algorithm is presented as *Sequential Importance Sampling* in Algorithm 9. It provides estimates for both $I_n(\varphi)$ and Z_n.

Algorithm 9 Sequential Importance Sampling

1. For $i = 1, \ldots, N$, sample $X_1^i \sim q_1(\cdot)$ and calculate the (unnormalised) importance weights $w_1(X_1^i) = \gamma_1(X_1^i)/q_1(X_1^i)$.
2. Calculate also the normalised weights $W_1^i = \frac{w_1(X_1^i)}{\sum_{j=1}^N w_1(X_1^j)}$.
3. For $k = 2, \ldots, n$:
   (a) For $i = 1, \ldots, N$, sample $X_k^i \sim q_k(\cdot|X_{1:k-1}^i)$ and calculate the (unnormalised) importance weights

$$w_k(X_{1:k}^i) = \frac{\gamma_k}{q_k}(X_{1:k}^i) = \frac{\gamma_k(X_{1:k}^i)}{\gamma_{k-1}(X_{1:k-1}^i)q_k(X_k^i|X_{1:k-1}^i)} w_{k-1}(X_{1:k-1}^i).$$

   (b) Calculate also the normalised weights $W_k^i = \frac{w_k(X_{1:k}^i)}{\sum_{j=1}^N w_k(X_{1:k}^j)}$
4. Estimate

$$\hat{I}_n^{\mathrm{SIS}}(\varphi) = \sum_{i=1}^N W_n^i \varphi(X_n^i),$$

$$\hat{Z}_n^{\mathrm{SIS}} = \frac{1}{N} \sum_{i=1}^N w(X_n^i).$$

We note here that there is nothing special about the final time step n in this algorithm. Estimators $\hat{I}_k^{\mathrm{SIS}}(\varphi)$ and $\hat{Z}_k^{\mathrm{SIS}}$ may be constructed sequentially for each $k \leq n$, and in some applications these intermediate estimators may be of interest in their own right.

However, the variance of the SIS estimators is known to scale extremely poorly with n. The following section describes how a simple modification to SIS results in an algorithm with considerably better variance properties.

3.3.3 Sequential Monte Carlo

The main reason for the poor variance behaviour of SIS is that particles which attain low importance weights at any step k are propagated through to step n. The fix is to insert an extra step known as *resampling* between the iterations of SIS, in which the particle system $(X_k^i)_{i=1}^N$ with normalised weights $(W_k^i)_{i=1}^N$ is replaced by a new system $(\bar{X}_k^i)_{i=1}^N$ with equal weights $1/N$.

In the simplest form of resampling, this system simply consists of N i.i.d draws from the empirical measure $\sum_{i=1}^N W_k^i \delta_{X_k^i}(\cdot)$. In other words, the importance samples and their weights are leveraged to form an approximate sample from η_k. It should be noted that other resampling schemes from the SMC literature can also be used; see Gerber et al. [2019] for a detailed analysis of schemes which might be expected to reduce estimator variance without introducing any bias.

The interleaving of SIS and resampling is known as Sequential Monte Carlo, and is presented in Algorithm 10.

Algorithm 10 Sequential Monte Carlo

1. For $i = 1, \dots, N$, sample $X_1^i \sim q_1(\cdot)$ and calculate the (unnormalised) importance weights $w_1(X_1^i) = \gamma_1(X_1^i)/q_1(X_1^i)$.
2. Calculate also the normalised weights $W_1^i = \frac{w_1(X_1^i)}{\sum_{j=1}^N w_1(X_1^j)}$.
3. *Resampling*: resample $\{X_1^i, W_1^i\}_{i=1}^N$ to obtain $\{\bar{X}_1^i, \frac{1}{N}\}_{i=1}^N$.
4. For $k = 2, \dots, n$:
   (a) For $i = 1, \dots, N$, sample $X_k^i \sim q_k(\cdot|X_{1:k-1}^i)$ and calculate the (unnormalised) importance weights

$$w_k(X_{1:k}^i) = \frac{\gamma_k}{q_k}(X_{1:k}^i) = \frac{\gamma_k(X_{1:k}^i)}{\gamma_{k-1}(X_{1:k-1}^i) q_k(X_k^i|X_{1:k-1}^i)} w_{k-1}(X_{1:k-1}^i).$$

   (b) Calculate also the normalised weights $W_k^i = \frac{w_k(X_{1:k}^i)}{\sum_{j=1}^N w_k(X_{1:k}^j)}$
   (c) *Resampling*: resample $\{X_k^i, W_k^i\}_{i=1}^N$ to obtain $\{\bar{X}_k^i, \frac{1}{N}\}_{i=1}^N$.

3.3.4 Feynman-Kac formalism

Feynman-Kac models are mathematical models of which Sequential Monte Carlo algorithms such as Algorithm 10 are particle approximations. The exposition of the theory and practice of Sequential Monte Carlo methods from the point of view of Feynman-Kac models is the subject of several comprehensive books, in particular Del Moral [2004, 2013]; Chopin and Papaspiliopoulos [2020]. In this thesis it is of relevance mainly in chapters 5 and 6, in which all the analysis is carried out in terms of these models and their particle approximations.

Following [Del Moral, 2013, Section 1.4.2], Feynman-Kac models can be motivated as an extensions of the notion of "change of measure". Suppose one has a Markov chain X_p taking values in a sequence E_p of state spaces, and write $\mathbb{P}_n$ for the measure of the path $(X_0, \dots, X_n)$. Suppose one also has a sequence $G_p : E_p \to \mathbb{R}_+$ of non-negative *potential functions*. The *Feynman-Kac measure* $\mathbb{Q}_n$ is the measure which accounts for 'twisting" or "re-weighting" the chain X_p by the potential G_p after each transition ([Del Moral, 2013, Equation 1.37]):

$$d\mathbb{Q}_n = \frac{1}{\mathcal{Z}_n} \prod_{p=0}^{n-1} G_p(X_p) d\mathbb{P}_n,$$

where $\mathcal{Z}_n$ is the *normalising constant* ensuring that $\mathbb{Q}_n$ is a probability measure. The corresponding unnormalised Feynman-Kac measure is $d\Gamma_n = \mathcal{Z}_n d\mathbb{Q}_n$. Of more interest to us are the n-th time marginals: for a bounded test function f, the n-th marginal

measure γ_n of Γ_n satisfies

$$\gamma_n(f) = \mathbb{E}\left[f(X_n)\prod_{p=0}^{n-1} G_p(X_p)\right], \tag{3.9}$$

and the corresponding marginal η_n of $\mathbb{Q}_n$ satisfies $\eta_n(f) = \gamma_n(f)/\gamma_n(1)$. If X_n has transition densities M_n and $X_0 \sim \lambda(\cdot)$, then (3.9) can equally be written

$$\gamma_n(f) = \int f(x_n)\lambda(dx_0)G_0(x_0)\prod_{p=1}^{n-1} M(x_{p-1}, dx_p)G_p(x_p).$$

For γ_n we have the immediate recursion

$$\gamma_n(dx_n) = \gamma_{n-1}(dx_{n-1})G_{n-1}(x_{n-1})M_n(dx_{n-1}, x_n),$$

from which it immediately follows ([Del Moral, 2013, Lemma 3.2.1, Equation (1.48)]) that

$$\gamma_n(f) = \eta_n(f)\prod_{p=1}^{n-1}\eta_p(G_p).$$

The corresponding recursion for η_n is

$$\eta_n(dx_n) = \frac{\eta_{n-1}((G_{n-1}M_n(\cdot, dx_n))}{\eta_{n-1}(G_{n-1})},$$

(These basic properties will all be repeatedly used in chapters 5 and 6.)

The close relation between Feynman-Kac models and SMC may be briefly sketched as follows. Suppose, as in section 3.3.2, one wishes to estimate $\eta_n(\varphi)$ for some test function φ. Take a selection of particles $\xi_0 = (\xi_0^1, \ldots, \xi_0^N) \sim \eta_0$, and a sequence of potential functions G_p. For time step $p = 1, 2, \ldots, n$, given ξ_p one resamples a new population $\hat{\xi}_p = (\hat{\xi}_p^1, \ldots, \hat{\xi}_p^N)$ independently at random from ξ_p, choosing ξ_p^j proportional to $G_p(\xi_p^j)$. One then samples $\xi_{p+1}^i \sim M_{p+1}(\hat{\xi}_p^i, \cdot)$ for $i = 1, \ldots, N$. At each time p, one defines the empirical measure

$$\eta_p^N(\cdot) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\xi_p^i}(\cdot).$$

Then at time n, one estimates $\eta_n(\varphi)$ using the empirical measure $\eta_n^N(\varphi)$. On the one hand, it is clear that this is a particle approximation of the Feynamn-Kac model defined in this section. On the other hand, it coincides with Algorithm 10, with the potential functions corresponding to the unnormalised weights w_k. Hence SMC may be used for inference wherever an underlying Feynman-Kac model is specified. A more detailed exposition of the relation between Feynamn-Kac formula and SMC algorithms can be found in [Del Moral, 2013, Section 4.3], and throughout the book Chopin and Papaspiliopoulos [2020].

In connection with rare event estimation and multilevel splitting, we note following

[Del Moral, 2013, Section 1.4.2.2] that with the choice $G_p = \mathbb{1}_{B_p}$, it follows that $\gamma_n(1) = \prod_{p=0}^{n-1} \eta_p G_p = \mathbb{P}(X_p \in B_p,\ p = 0, \ldots, n)$. In other words, the SMC variant of multilevel splitting targets the normalising constant $\gamma_n(1)$ of the corresponding Feynman-Kac model. More details on the connection between rare event estimation and Feynman-Kac models can be found in [Del Moral, 2004, Section 12.2.5], [Del Moral, 2013, Section 2.7] (including a treatment of multilevel splitting), and [Chopin and Papaspiliopoulos, 2020, Section 17.2.4].

Chapter 4

Exact multilevel splitting

4.1 Introduction

In this chapter, we describe several algorithms for exact, unbiased estimates of rare event probabilities for continuous-time Markov processes in the splitting setting. Each is based on combining the ε-strong sampling procedures introduced in Section 2.2 with one of the rare event algorithms described in chapter 3.

In 4.1.1, we begin with a slightly fuller discussion of discretisation error and the barrier crossing problem introduced in Section 2.1.4. In 4.4, we set down the notational scheme that we will employ to describe the action of ε-strong algorithms applied to existing ε-strong paths. In 4.2, we describe efficient multilevel splitting algorithms in one dimension which leverage the properties of the ε-strong algorithms of 2.2.2 and 2.2.3 in one dimension. In 4.3, we set down algorithms for the more general multi-dimensional setting, and establish their unbiasedness. Finally, we illustrate both the one-dimensional and higher-dimensional algorithms with numerical examples in 4.5.

4.1.1 Discretisation error for continuous-time processes

In the description of multilevel splitting in section 3.2.2, we assumed that we are able to simulate without approximation the pair $U_i = (\sigma_i, X(\sigma_i))$. Since these depend upon full sample paths of X, it is not apparent that this can be done except when X has an exceptionally simple form, for example if X is a piece-wise deterministic process. In practice it is usual to resort to a discretisation scheme (see section 2.1.4). For example, suppose that X is described by $X(0) \sim \lambda$ and

$$dX(t) = a\left(X(t)\right) dt + b\left(X(t)\right) dB(t) \tag{4.1}$$

for $t \in [0, T]$, where B is e-dimensional Brownian motion for some $e \in \mathbb{N}$, and $a : \mathbb{R}^d \to \mathbb{R}^d, b : \mathbb{R}^d \to \mathbb{R}^{d\times e}$ are sufficiently regular to guarantee the existence of a strong Itô solution (see for example, [Kloeden and Platen, 2013, Section 4.5] for suitable conditions).

We might use an Euler-Maruyama scheme such as the following, defined on a chosen time-grid $t_j \in \mathcal{P}$ for a partition $\mathcal{P}$ of $[0, T]$:

$$\hat{X}(t_{j+1}) = \hat{X}(t_j) + a\left(\hat{X}(t_j)\right)(t_{j+1} - t_j) + b\left(\hat{X}(t_j)\right)(B(t_{j+1}) - B(t_j)) \ . \tag{4.2}$$

with $\hat{X}(0) \sim \lambda$. Such a scheme can then be used to implement an approximation of Algorithm 6 as follows: rather than drawing samples from M_i in Steps 1 and 3b, one runs the discrete scheme until a crossing into A or B is observed at time $\hat{\sigma}_i = \min_j\{t_j : \hat{X}(t_j) \in A \cup B\}$, and approximates U_i using $(\hat{\sigma}_i, \hat{X}(\hat{\sigma}_i))$. Writing $\hat{M}_i$ for the Markov transition kernels of the sequence $(\hat{U}_i)_{i=0}^m = \left((\hat{\sigma}_i, \hat{X}(\hat{\sigma}_i))\right)_{i=0}^m$, this is described in Algorithm 11.

Algorithm 11 Discretised Multilevel Splitting (Euler-Maruyama method)

Given λ together with G_i, $\hat{M}_i$ for $i = 1, \ldots, m$, an initial number of particles N_0, and splitting ratios $R_1, \ldots, R_{m-1}$:

1. For each $j = 1, \ldots, N_0$, draw independently $\hat{X}_1^j(0) \sim \lambda$ and $\hat{U}_1^j \sim \hat{M}_1\left(\left(0, \hat{X}_1^j(0)\right), \cdot\right)$.
2. Let $S_1 = \{\hat{U}_1^j : G_1(\hat{U}_1^j) = 1\}$ be a list of the the surviving paths, and set $N_1 = |S_1|$.
3. For $i = 2, \ldots, m$:
   (a) If $N_{i-1} = 0$, return $\hat{p} = 0$.
   (b) Else given $S_{i-1} = \{\bar{U}_{i-1}^j\}_{j=1}^{N_{i-1}}$, for all $(j,k) \in \{(j', k') : 1 \le j' \le N_{i-1}, 1 \le k' \le R_{i-1}\}$ sample independently $\hat{U}_i^{j,k} \sim \hat{M}_i(\bar{U}_{i-1}^j, \cdot)$.
   (c) Let $S_i = \{\hat{U}_i^{j,k} : G_i(U_i^{j,k}) = 1\}$, and $N_i = |S_i|$.
4. Return
$$\hat{p} = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}.$$

This practice is very common in the splitting literature, and more generally when simulating diffusion processes. For example, in Cérou and Guyader [2007] a formal algorithm is developed in a continuous-time setting, but the numerical example is discretised "[finely] enough to avoid clipping the process, which could introduce a bias in the estimation". Lagnoux-Renaudie [2009] acknowledges explicitly the bias induced by discretisation in their application, and proposes a small modification to reduce, but not eliminate, it. Even Bréhier et al. [2016], which focuses on establishing the unbiasedness of a particular adaptive multilevel splitting framework in some generality ultimately invokes time-discretisation to apply the framework to continuous-time processes such as over-damped Langevin diffusions.

With such a numerical scheme, one is forced to assess the level-crossing problem according to the discrete sample paths of $\hat{X}$. But the law $\hat{P}$ of $\hat{X}$ will not in general coincide with the true finite-dimensional marginal law P_k of $(X_{t_1}, \ldots, X_{t_k})$ induced by (4.1). And even if it were possible to get a finite-dimensional sample from P restricted to the times of the partition, for example by exact simulation, this would give no information about the sample path over the open intervals (t_j, t_{j+1}), during which a crossing may (or may not) occur.

The quantities that are needed to carry out Algorithms 6 and 7 are $U_i^j = (\sigma_i^j, X_i^j(\sigma_i^j))$ and $G_i(U_i^j)$. Using ε-strong simulation as described in section 2.2, it is possible to sample exactly $G_i(U_i^j)$ without access to U_i^j itself. Later, we show also that the modification to Algorithms 6 and 7 made necessary by using this approximation does not affect the unbiasedness of the resulting estimates.

4.2 In one dimension

Before describing a general algorithm for exact rare event simulation, we first briefly deal with the special case of dimension one. As detailed in sections 2.2.2 and 2.2.3, in this case we have access to samplers which produce more information about the process than is generated by general ε-strong samplers. This additional information may be exploited to construct considerably more efficient rare event estimators.

For clarity of exposition, and since our main concern is the multi-dimensional case, we concentrate on the use of the algorithm of section 2.2.2. The extension of the methods to this section to the second algorithm may be accomplished by leveraging, for example, [Pollock et al., 2016, Algorithm 23]; see also the accompanying examples.

In one dimension, the distinction drawn in 3.2 between the process X and its projection under a reaction co-ordinate is not meaningful. Without loss of generality we take ξ to be the identity function, and simply consider the probabilities

$$\mathbb{P}(X \text{ hits } z_i \text{ before } z_A | X \text{ hit } z_{i-1} \text{ before } z_A).$$

4.2.1 Crossing a single barrier

We begin by considering the simpler problem of sampling exactly an indicator random variable for the event that X hits some target z_D during the fixed time interval $[0, t]$, given that $X_0 = x_0$ for some $x_0 \in (-\infty, z_D)$. Using the properties described in Section 2.2.2, we can immediately write down pseudocode for a procedure which does this, given in Algorithm 12.

Algorithm 12 Single barrier crossing, 1-D Brownian motion

1. Give $X_s = x_s$, sample $X_t \sim \mathcal{N}(x_s, t - s)$.
2. If $X_t \geq z_D$, return $+1$. Otherwise, sample bounds
$$L_{s,t}^{\downarrow} \leq m_{s,t} \leq L_{s,t}^{\uparrow},\ U_{s,t}^{\downarrow} \leq M_{s,t} \leq U_{s,t}^{\uparrow}$$
for X conditional on X_s, X_t.
3. (a) If $U_{s,t}^{\downarrow} \geq z_D$, return $+1$.
   (b) If instead $U_{s,t}^{\uparrow} \leq z_D$, return -1.
   (c) If neither hold, apply Step 2 of 2.2.2 with $U^* = z_D$ and return ± 1 accordingly.

In fact, even this algorithm is redundant in that step 2 can be modified to check the crossing directly. To explain this, we need to look more closely at Step 1 of 2.2.2:

- Given $B_0 = 0$, it is possible to sample B_t together with initial layers $(L^{\downarrow}_{0,t}, L^{\uparrow}_{0,t}, U^{\downarrow}_{0,t}, U^{\uparrow}_{0,t})$.

In fact, one has substantial control over the choice of constraint. Given two increasing unbounded sequences $(a_n)_{n=0}^{\infty}, (b_n)_{n=0}^{\infty}$ with $a_0 = b_0 = 0$, the set

$$\mathbb{R} \setminus (B_0 \wedge B_t, B_0 \vee B_t)$$

is the disjoint union of all sets of the form

$$((B_0 \wedge B_t) - a_{i+1}, (B_0 \wedge B_t) - a_i] \cup [(B_0 \vee B_t) + b_j, (B_0 \vee B_t) + b_{j+1}),$$

$i, j = 0, \ldots, \infty$.

By placing an ordering $I_1 = (i_1, j_1), I_2 = (i_2, j_2), \ldots$ on the pairs (i, j), the layers can be initialised by sampling exactly the events

$$\begin{aligned}\mathbb{P}_{0,t}^{B_0,B_t}(m_{s,t} \in [(B_0 \wedge B_t) - a_{i_n+1}, (B_0 \wedge B_t) - a_{i_n}],\\ M_{s,t} \in [(B_0 \vee B_t) + b_{j_n}, (B_0 \vee B_t) + b_{j_n+1}])\end{aligned}$$

in sequence $n = 1, 2, \ldots$ until a success is obtained. (Recall that we demonstrated how to sample this type of event by retrospective Bernoulli sampling in Section 2.2.2.) Sampling the first success at I_n is equivalent to sampling the initial layers

$$\begin{aligned}(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}) =\\ ((B_0 \wedge B_t) - a_{i_n+1}, (B_0 \wedge B_t) - a_{i_n}, (B_0 \vee B_t) + b_{j_n}, (B_0 \vee B_t) + b_{j_n+1})\,.\end{aligned}$$

Then by choosing $b_1 = z_D - (B_0 \vee B_t)$, the initial upper layer immediately determines whether $M_{0,t} \geq z_D$. In this case, step 3c of Algorithm 12) is never reached and the algorithm terminates in a single step.

4.2.2 Crossing a two-sided barrier

Next, we need a procedure to decide which of two given barriers is crossed first. Let $z_A < z_B$ be the barriers, and assume $X_0 = x_0$ for $x_0 \in (z_A, z_B)$. Again, we get a more efficient algorithm by targeting the barriers when sampling the initial layers. In the notation of the previous section, we take $a_1 = z_A + (X_0 \wedge X_t), b_1 = z_D - (X_0 \vee X_t)$. The initial layers then immediately determine whether X crosses into A or B or both or neither over the interval $[0, t]$.

4.2.3 Exact splitting in one dimension

Algorithm 13 can be used to specify an exact multilevel splitting algorithm in dimension one, which is given in Algorithm 14. We assume here a recurrent set $(\infty, z_A]$, and a sequence of barriers $z_A < z_1 < \ldots < z_m$, where $\mathbb{P}(X \text{ hits } z_m \text{ before } z_A)$ is the rare event in question.

Algorithm 13 Double barrier crossing, 1-D Brownian motion

Given barriers $z_A < z_B$ and $X_s = x_s \in (z_A, z_B)$,

1. Sample $X_t \sim \mathcal{N}(X_s, t-s)$.
2. Sample bounds
$$L^{\downarrow}_{s,t} \le m_{s,t} \le L^{\uparrow}_{s,t},\ U^{\downarrow}_{s,t} \le M_{s,t} \le U^{\uparrow}_{s,t}$$
for X conditional on X_s, X_t.
3. (a) If $z_A \le L^{\downarrow}_{s,t}, U^{\uparrow}_{s,t} \le z_B$, set $(s,t) \leftarrow (t, 2t-s)$ and return to 2.
   (b) If $L^{\uparrow}_{s,t} \le z_A$ and $U^{\uparrow}_{s,t} \le z_B$, return 0.
   (c) If $L^{\downarrow}_{s,t} \ge z_A$ and $U^{\downarrow}_{s,t} \ge z_B$, return +1.
   (d) If $L^{\uparrow}_{s,t} \le z_A$ and $U^{\downarrow}_{s,t} \ge z_B$, apply property 3 of section 2.2.2. Using the new information over the interval $[s, t^*]$, check conditions (b)-(d) of step 4 again, following the listed consequences. If condition (a) holds, repeat with the information from $[t^*, t]$. If (a) holds again, set $(s,t) \leftarrow (t, 2t-s)$, and return to 2.

A significant simplification which fails to hold in multiple dimensions is that since the barrier z_D is a single number, knowing that X hits the barrier immediately entails knowing *where* it hits the barrier, namely at the only possible location z_D itself. (By contrast, simply knowing that a 2D process Y hits a line, say, gives no information about *where* it hits the line.)

Since X is assumed to be a Markov process, this allows us to dispense with saving most of the details of the path of X. Starting from each level z_i, it suffices to determine whether X hits z_A or z_{i+1} first using Algorithm 13, and to split the successful paths at the "correct" location z_{i+1}. This is exactly what is exploited below.

Algorithm 14 Exact multilevel splitting, 1-D Brownian motion

Given barriers $z_A < z_1 < \ldots < z_m$, splitting numbers $R_1, \ldots, R_{m-1}$, and $x_0 \in (a, b_1)$:

1. For $j = 1, \ldots, N$, set $X_0^j = x_0$. Initiate $N_1 = 0$.
2. For $j = 1, \ldots, N_0$, sample $G_1^j \sim$ Algorithm.13(z_A, z_i, X_0^j), and update $N_1 \leftarrow N_1 + G_1^j$.
3. For $i = 2, \ldots, m$:
   (a) For $j = 1, \ldots, R_{i-1}N_{i-1}$, set $X_i^j = z_i$. Initiate $N_i = 0$.
   (b) For $j = 1, \ldots, N$, sample $G_j^i \sim$ Algorithm.13(z_A, z_i, X_i^j), and update $N_i \leftarrow N_i + G_i^j$.
4. Estimate
$$\hat{p} = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}.$$

4.3 Exact simulation of rare events in multiple dimensions

In multilevel splitting, one tracks the progress of X towards A and B using the reaction co-ordinate ξ, declaring a crossing at level i when the process $\xi(X)$ reaches either z_A or z_i. In this section we describe methods for sampling such *barrier crossing* events exactly, for Markov processes X with almost surely continuous sample paths for which an ε-strong method for sampling X exists. Combining these with a slight modification

of Algorithms 6 & 7 provides a method of obtaining unbiased estimates of rare event probabilities. Much of what follows is geometrically intuitive, though notationally cumbersome, and Figures 2.2 and 4.1 are intended to illustrate the intuition which motivates the accompanying specifications.

Throughout the rest of this chapter we take X to be a diffusion

$$dX_t = a(X_t)dt + b(X_t)dB_t$$

together with the conditions assumed in Section 2.1.3. For simplicity, we assume further that X has volatility bounded away from 0, which ensures that X crosses any given boundary with positive probability over any time interval. We assume also that an ε-strong algorithm as described in 2.2 has been chosen and is used to carry out the sampling in Algorithms 15, 16 and 18.

4.4 Notation and operations on ε-strong paths

With Property 4 of Section 2.2 in mind, let $(\varepsilon_\ell)_{\ell=1}^\infty$ be a decreasing sequence of tolerances converging to 0. Write $\tilde{X}^\ell[s:t]$ for the ε-strong path $\tilde{X}^{\varepsilon_\ell}[s:t]$. Let $(t_1, \dots, t_K)$ be the jump-times of this path, and let $t_0 = s$, $t_{K+1} = t$. It is useful to define the associated discrete-time process $(\tilde{X}_k^\ell)_{k=0}^K$ where $\tilde{X}_k^\ell = \tilde{X}^\ell(t_k)$. We define also an augmented process called the *skeleton* of $\tilde{X}^\ell$ to be the discrete-time process $(\tilde{Z}_k^\ell)_{k=0}^K$ such that

$$\tilde{Z}_k^\ell = (t_k, t_{k+1}, \tilde{X}_k^\ell, \ell).$$

(This is, in a way, rather a backwards definition since an ε-strong path itself is typically constructed from its skeleton.) Given the skeleton $\tilde{Z}$ as defined above, the (almost) unique ε-strong path associated with it is defined by

$$\tilde{X}^\ell(u) = \sum_{k=0}^{K} \mathbb{I}_{[t_k, t_{k+1})}(u)\tilde{X}_k^\ell.$$

for $u \in [s,t)$, and we may take $\tilde{X}^\ell(t) = \tilde{X}_K^\ell$. The skeleton is somehow a more computationally-motivated object than its associated path, and we will refer primarily to the paths themselves outside of our algorithmic pseudo-code. It is useful to define also $C_k = \{x \ : \|x - \tilde{X}_k^\ell\| \le \varepsilon_\ell\}$, the constraining region for X over $[t_k, t_{k+1}]$.

Say two skeletons $(\tilde{Z}_k^{i,1})_{k=0}^K, (\tilde{Z}_m^{j,2})_{m=0}^L$ defined on $[r,s]$ and $[s,t]$ respectively, and with constraining regions $(C_k^1)_{k=0}^{K-1}, (C_m^2)_{m=1}^{L-1}$ respectively, are *compatible* if $C_K^1 \cap C_0^2 \neq \emptyset$. For two compatible skeletons, we define their concatenation $\tilde{Z}^3 = \tilde{Z}^{i,1} \oplus \tilde{Z}^{j,2}$ to be the process $(\tilde{Z}_n^3)_{n=1}^{K+L+1}$ with

$$\tilde{Z}_n^3 = \begin{cases} \tilde{Z}_n^{i,1}, & 0 \le n \le K, \\ \tilde{Z}_{n-K-1}^{j,2}, & K+1 \le n \le L+K+1. \end{cases}$$

Analogously, for two compatible ε-strong paths $\tilde{X}^{i,1}, \tilde{X}^{j,2}$ defined on $[r,s]$, $[s,t]$ respectively, and with skeletons $(\tilde{Z}_k^{i,1})_{k=0}^K, (\tilde{Z}_m^{j,2})_{m=0}^L$, we define their concatenation as follows. Take the skeleton $\tilde{Z}^3 = \tilde{Z}^{i,1} \oplus \tilde{Z}^{j,2}$, and writing $\tilde{Z}_n^3 = (t_n, t_{n+1}, \tilde{X}_n^{\ell(n)}, \ell(n))$ set

$$(\tilde{X}^{i,1} \oplus \tilde{X}^{j,2})(u) = \sum_{n=0}^{K+L+1} \mathbb{I}_{[t_n, t_{n+1})}(u) \tilde{X}_n^{\ell(n)}$$

for $u \in [s,t)$. Define the end-point by $(\tilde{X}^{i,1} \oplus \tilde{X}^{j,2})(t) = \tilde{X}_{L+K+1}^{\ell(L+K+1)}$.

Two paths which are themselves concatenations of ε- strong paths may be concatenated analogously. We will exploit the fact that the binary concatenation operation is associative to allow us to write concatenations of more than two processes without ambiguity. We also find it convenient to adopt the convention that for $k_1 \geq k_2$, the degenerate sub-skeleton $(\tilde{Z}_k)_{k=k_1}^{k_2}$ acts as the identity for this binary operation, so that for any skeleton $\tilde{Y}$, $(Z_k)_{k=k_1}^{k_2} \oplus \tilde{Y} = \tilde{Y} \oplus (Z_k)_{k=k_1}^{k_2} = \tilde{Y}$.

Typically ε-strong algorithms require more information about the process than we have made explicit in our definition of a skeleton, for example those of Pollock et al. [2016]. For ease of exposition, we have suppressed this since we do not need to refer to it for the development of the algorithms in this paper, but it should be understood that our skeletons contain any extra information required for the stated ε-strong conditions to hold.

4.4.1 Crossing a single barrier

We begin with the simpler problem of sampling exactly an indicator random variable for the event that X crosses into a set $D = \xi^{-1}([z_D, \infty))$ when started from its complement $D^c = \xi^{-1}((-\infty, z_D))$, over the fixed time interval $[0,t]$. To this end, we suppose that $X(0) \sim \lambda$ where the support of λ is contained in D^c. Assume that X is sufficiently regular that (almost surely) a path $X[0:t]$ which crosses into D attains a maximum distance $d^{\max}(X, D^c) > 0$ from D^c, and conversely, a path $X[0:t]$ which does not cross into D has (almost surely) minimum distance $d^{\min}(X, D) > 0$ from D. A sufficient condition to ensure that this is true is $P_t(x, B) > 0$ for (almost) all $x \in \mathbb{R}^d$, since A is positive recurrent for X.

Consider the ε-strong path $\tilde{X}^{\varepsilon_1}(0:t)$ for a tolerance ε_1, and let $0 = t_0 < \cdots < t_{K+1} = t$ be its jump-times. For $k = 0, \ldots, K$, inside each time interval $[t_k, t_{k+1}]$ the ball $C_k = \{x \ : \|x - \tilde{X}^{\varepsilon_1}(t_k)\| \leq \varepsilon_1\}$ almost surely constrains the path of X associated with $\tilde{X}^{\varepsilon_1}$. So if $\varepsilon_1 < \max(d^{\max}(X, D^c), d^{\min}(X, D))$, then either i) $C_k \subset D$ for some k (if X does make a crossing), or ii) $C_k \subset D^c$ for all k (if X does not make a crossing). By checking each C_k in turn, we can determine which of these conditions holds, and thereby construct the desired indicator random variable.

Of course, it is not possible to choose a suitable ε_1 in advance, since the underlying path X and its minimum and maximum distances from D and D^c are not known. Instead, we can specify a sequence $(\varepsilon_\ell)_{\ell=1}^\infty$ of tolerances with $\varepsilon_\ell \to 0$. If X^ℓ turns out

to be insufficient to determine the crossing, we can apply Property 4 of section 2.2.1 to sample $X^{\ell+1}$ conditional on X^ℓ as necessary until a sufficiently small tolerance is found.

It is very wasteful, however, to construct a finitely-representable path $\tilde{X}[0:t]$ which is very close to $X[0,t]$ on the whole interval $[0:t]$. It is likely that even when X crosses into D, much of the time X is not near the boundary ∂D, and we need only approximate X closely where it is near ∂D. For this reason, as suggested in 2.2.1, it useful to work instead with paths of mixed tolerance

$$\tilde{X} = \bigoplus_{j=1}^{J} \tilde{X}^{\ell(j)}[s_{j-1}, s_j],$$

where $(0 = s_0, s_1, \ldots, s_J)$ is a partition of $[0,t]$ with $s_J = t$, and $(\varepsilon_{\ell(j)})_{j=1}^J$ is a selection from $(\varepsilon_\ell)_{\ell=1}^\infty$. Such a path is the result, for example, of applying Property 4 with $\varepsilon_2 < \varepsilon_1$ to a constant segment $\tilde{X}^1(t_{k-1}, t_k)$ of $\tilde{X}^1$, and the result in this case would be

$$\tilde{X} = \tilde{X}^1[0:t_{k-1}] \oplus \tilde{X}^2[t_{k-1}:t_k] \oplus \tilde{X}^1[t_k:t].$$

For later convenience, our formalisation in Algorithm 15 of the algorithm under description takes an $\tilde{X}$ of this kind, or rather the skeleton of such a path, as input.

The three possible relationships between C_k and D, D^c can be described in terms of ξ, the reaction co-ordinate: $\sup_{x\in C_k} \xi(x) < z_D$ is equivalent to $X[t_k, t_{k+1}] \subset D^c$, and similarly $\inf_{x\in C_k} \xi(x) \geq z_D$ is equivalent to $X[t_k, t_{k+1}] \subset D$. The third possibility, that

$$\inf_{x\in C_k} \xi(x) < z_D, \quad \sup_{x\in C_k} \xi(x) \geq z_D \ ,$$

gives no definite information about the location of $X[t_k, t_{k+1}]$ with respect to D. It is consistent with $X[t_k, t_{k+1}]$ falling entirely in D, entirely in D^c, or partially in both. We categorise the behaviour of the process in this time interval by defining:

$$n_k := \begin{cases} -1, & \text{if } \sup_{x\in C_k} \xi(x) < z_D \\ 0, & \text{if } \inf_{x\in C_k} \xi(x) < z_D,\ \sup_{x\in C_k} \xi(x) \geq z_D \\ +1, & \text{if } \inf_{x\in C_k} \xi(x) \geq z_D \end{cases} . \tag{4.3}$$

Algorithm 15 samples exactly an indicator for the event that X crosses into D.

It may be noted that in Step 4a) of Algorithm 15, it is not strictly necessary to choose k minimal. There may be computational advantages to using a different system, such as attempting choose an k for which $C_k \cap D$ is large (indicating a high probability of crossing). This can be computationally preferable, at the expense of providing less information about τ_A (see Section 4.4.3).

The assumption that $\sup_{x\in C_k} \xi(x)$ and $\inf_{x\in C_k} \xi(x)$ can be calculated is rather strong,

Algorithm 15 Single barrier crossing

function$((\tilde{Z}_k)_{k=0}^K, D)$:

1. Calculate the sequence $(n_k)_{k=0}^K$.
2. If $n_k = -1$ for all $k = 1, \ldots, K$, return $(0, \tilde{Z})$ to indicate no crossing into D.
3. If $n_k = +1$ for some k, return $(+1, \tilde{Z})$ to indicate a crossing into D.
4. Else:
   (a) Set $j = \min\{k \in \{0, \ldots, K\} : n_k = 0\}$, and consider $\tilde{\mathcal{Z}}^{\ell(j)} := \tilde{Z}_j^{\ell(j)} = (t_j, t_{j+1}, \tilde{X}_j, \ell(j))$. Use the refining Property 4 of section 2.2.1 to sample $\tilde{\mathcal{Z}}^{\ell(j)+1}$ conditional on $\tilde{\mathcal{Z}}^{\ell(j)}$.
   (b) Update
   $$\tilde{Z} \leftarrow (\tilde{Z}_k)_{k=0}^{j-1} \oplus \tilde{\mathcal{Z}}^{\ell(j)+1} \oplus (\tilde{Z}_m)_{m=j+1}^{K},$$
   and update $K \leftarrow (\#\text{jump-times of } \tilde{Z}) + 1$. Return to Step 1.

but holds for many realistic scenarios. For example, supposing X takes values in $\mathbb{R}^d$ and, taking the norm $\|x\| = \max_{i \in 1,\ldots,d} \mid x_i \mid$, these quantities can be calculated if ξ is monotonic in each argument. As a specific example, in Section 4.5, we take $d = 2$, $\xi(x, y) = \min(x, y)$. Another example of a tractable reaction coordinate, which illustrates that monotonicity is not necessary, is $\xi(x, y) = \mid x - y \mid$.

4.4.2 Crossing a two-sided barrier

We consider next a two-sided barrier problem, with regions $A = \xi^{-1}((-\infty, z_A])$, $B = \xi^{-1}([z_B, \infty))$, with $X(0) \sim \lambda$ such that $z_A \leq \xi(X(0)) < z_B$, and the problem of sampling an indicator random variable for the event that X crosses into B before A. Here we work over over a random interval $[0, \sigma]$ where σ is the hitting time for $A \cup B$ of X, rather than over a fixed interval as in the previous section.

In this case, we can declare a level crossing into A (for example) at the first k for which

$$\sup_{x \in C_k} \xi(x) < z_A \text{ and } \max_{j<k} \sup_{x \in C_k} \xi(x) < z_B,$$

if such an k exists.

Informally, we can declare the crossing into A when i) some ε-ball lies entirely in set A, which guarantees that X has reached A; and ii) no preceding ε-ball intersects set B, which guarantees that X has not reached B. (The conditions for a crossing into B are analogous). If there is no such k, it is necessary to carry out further simulations using Property 4 of 2.2.1.

As in the previous section, we associate a number $n_k \in \{-2, -1, 0, 1, 2\}$ with each ball C_k, according to the categorisation in Table 4.1. In order to simplify the categorisation and presentation of the algorithm, we make the assumption that our initial tolerance ε_1 is sufficiently small that i) C_k intersects at most one of A, B; and that ii) if $C_k \cap A \neq \emptyset$, then $C_{k+1} \cap B = \emptyset$ (likewise with A,B interchanged); this can be assured by refining the initial tolerance until it is satisfied. With this assumption, the categorisation in Table 4.1 is complete, and the sequence (n_k) satisfies $n_{k+1} \in \{n_k - 1, n_k, n_k + 1\}$.

n_k	Condition	Meaning
-2	$\sup_{x\in C_k}\xi(x) \le z_A$	X remains within A on $[t_k, t_{k+1}]$
-1	$\inf_{x\in C_k}\xi(x) \le z_A,\ \sup_{x\in C_k}\xi(x) > z_A$	X may enter A in $[t_k, t_{k+1}]$
0	$\inf_{x\in C_k}\xi(x) > z_A,\ \sup_{x\in C_k}\xi(x) < z_B$	X does not enter A or B in $[t_k, t_{k+1}]$
1	$\sup_{x\in C_k}\xi(x) \ge z_B,\ \inf_{x\in C_k}\xi(x) < z_B$	X may enter B in on $[t_k, t_{k+1}]$
2	$\inf_{x\in C_k}\xi(x) \ge z_B$	X remains within B on $[t_k, t_{k+1}]$

Table 4.1: Enumeration of possible C_k-locations with respect to a two-sided barrier.

Suppose we have calculated the sequence $(n_k)_{k=0}^K$ associated with the path $\tilde{X}[s:t]$. In order to determine which of A and B has been crossed first, it is necessary to consider segments of $\tilde{X}[s:t]$ in which a crossing of one or the other barrier may have occurred, but a crossing of both barriers *cannot* have occurred. By checking each of these segments in turn, the decision can be made. In terms of the sequence (n_k), these segments are constructed as follows. We write J for the number of segments, where the definition of J is contained in the construction. We define recursively the sequence of indices which mark the beginning of a new segment in which a crossing may occur, as the sequence $(\kappa(j))_{j=0}^J$. Let $\kappa(0) = 0$, and while $\kappa(j-1) < K+1$, set

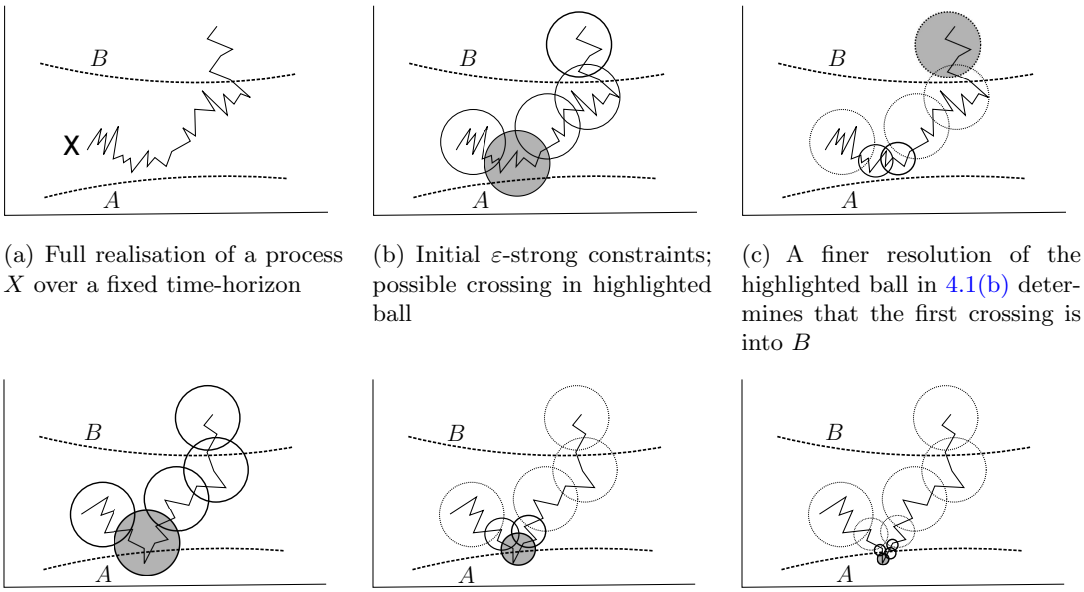
$$\kappa(j) = \begin{cases} \min\{k > \kappa(j-1) : n_k = 0\} \wedge (K+1) \text{ if } n_{\kappa(j-1)} \neq 0 \\ \min\{k > \kappa(j-1) : n_k \neq 0\} \wedge (K+1) \text{ if } n_{\kappa(j-1)} = 0. \end{cases}$$

Each element of this sequence is taken to denote the beginning of a block $\mathcal{B}_j \subset (n_k)$ of consecutive elements, so $\mathcal{B}_j = \{n_k : \kappa(j-1) \le k < \kappa(j)\}$. By construction, each $\mathcal{B}_j$ consists of a string of elements of exactly one of the sets $\{-2,-1\}, \{0\}, \{1,2\}$. Each block therefore corresponds to a segment of $\tilde{X}[s:t]$ in which X crosses into at most one of A and B. For example, in the case that $(n_i) = (1,1,0,0,-1,-2,-1)$, $J = 3$ and the blocks are $\mathcal{B}_1 = (1,1), \mathcal{B}_2 = (0,0), \mathcal{B}_3 = (-1,-2,-1)$, which in this case correspond to "possible crossing into B", "no crossing" and "definite crossing into A", respectively.

The two-sided barrier crossing procedure is given in Algorithm 16, in which the output is an indicator random variable for the event that X hits B before A. An illustration is given in Figure 4.1.

In general, it may be computationally inefficient to use a sufficiently small initial ε for the whole sample path of X, for example when X crosses a barrier at a very early time. We note that there are many variations of Algorithm 16 which will also sample the outcome correctly and may avoid doing so for computational efficiency. Our choice has been made for clarity of exposition.

We have assumed throughout for convenience that the sample paths with which we deal are almost surely continuous. Relaxing this requirement is straightforward but slightly complicates the implementation. Given an ε-strong algorithm for a jump-diffusion or similar piece-wise-continuous process, an appropriate alteration to the rule for beginning a new block will give an equally correct algorithm.



(a) Full realisation of a process X over a fixed time-horizon

(b) Initial ε-strong constraints; possible crossing in highlighted ball

(c) A finer resolution of the highlighted ball in 4.1(b) determines that the first crossing is into B

(d) Alternative realisation of X, together with initial ε-strong constraints

(e) One-stage resolution indeterminate

(f) Further resolution determines first crossing into A

Figure 4.1: Two illustrations of Algorithm 16. The first row shows: 4.1(a) a realisation of X over a finite time horizon, 4.1(b) an initial ε-strong simulation and 4.1(c) a refinement which is sufficient to show the process crossing into B. The second row shows 4.1(d) an alternative sample path consistent with the same initial ε-strong simulation, 4.1(e) an inconclusive refinement and 4.1(f) a further refinement sufficient to conclude that the process has crossed into A.

Algorithm 16 Two-sided barrier crossing

function$((\tilde{Z}_k)_{k=0}^K, A, B)$, for $\tilde{Z}$ a skeleton over the interval $[s,t]$:

1. Initialise $\tilde{Z}^{\text{full}}$ as an empty skeleton.
2. Calculate the sequence $(n_k)_{k=0}^K$ associated with $\tilde{Z}$.
3. Divide $(n_k)_{k=0}^K$ into blocks $\mathcal{B}_1, \dots, \mathcal{B}_J$ using the sequence $(\kappa(j))_{j=0}^{J+1}$ as described above.
4. For $j = 1, \dots, J$:
   (a) If $(-2) \in \mathcal{B}_j$, set $D = (-1)$ to indicate a crossing into A, and skip to 6.
   (b) If $(+2) \in \mathcal{B}_j$, set $D = (+1)$ to indicate a crossing into B, and skip to 6.
   (c) If $(-1) \in B_j$, sample $(I, \tilde{Z}') \leftarrow$ Algorithm.15 $((\tilde{Z}_k)_{k=\kappa(j-1)}^{\kappa(j)-1}, A)$ to decide if the first crossing is into A, and set
   $$\tilde{Z} \leftarrow (\tilde{Z}_k)_{k=0}^{\kappa(j-1)-1} \oplus \tilde{Z}' \oplus (\tilde{Z}_k)_{k=\kappa(j)}^{K},$$
   and $K \leftarrow (\#\text{jump-times of } \tilde{Z}) + 1$. If $I = (+1)$, set $D = (-1)$ and skip to 6.
   (d) If $(+1) \in B_j$, sample $(I, \tilde{Z}') \leftarrow$ Algorithm.15 $((\tilde{Z}_k)_{k=\kappa(j-1)}^{\kappa(j)-1}, B)$ to decide if the first crossing is into B, and update $\tilde{Z}, K$ as in c). If $I = (+1)$, set $D = (+1)$ and skip to 6.
5. Here, we know that no crossing is made in the interval $[s,t]$ spanned by $\tilde{Z}$, so it is necessary to sample a continuation of the path. Let $\tilde{Z}^{\text{full}} \leftarrow \tilde{Z}^{\text{full}} \oplus \tilde{Z}$. Writing $(t_K, t, x, \varepsilon) = \tilde{Z}_K$, update $(s,t) \leftarrow (t, 2t-s)$, and sample a new $\tilde{X}^{\varepsilon_1}(s:t]$. Record its skeleton $(\tilde{Z}_k)_{k=1}^K$, and return to 2.
6. Set $\tilde{Z}^{\text{full}} \leftarrow \tilde{Z}^{\text{full}} \oplus \tilde{Z}$, and return $(D, \tilde{Z}^{\text{full}})$.

4.4.3 Exact MLS

Finally, we turn to an exact implementation of multilevel splitting. The main point of difference with Algorithm 6 is that since an ε-strong sample $\tilde{X}[s,t]$ merely constrains the corresponding path $X[s,t]$, there is no easy way to determine the hitting location and time of any given barrier. In particular, we will not have access to the hitting times σ_i, τ_i nor the hitting locations $X(\tau_i)$ defined in Section 3.2.

Suppose we use Algorithm 16 to sample an indicator random variable for the event that X hits B_1 before A, for instance, with initial simulation interval $[0,T]$. Suppose that a positive result is returned over the interval $[0, cT]$, for some random $c \in \mathbb{N}$ corresponding to the number of passes through Algorithm 16. We must then choose when and where to split this path of $\tilde{X}$. In this section, we show that if the splitting is carried out at time cT, this does not affect the unbiasedness of the MLS estimate.

In general, write $\tilde{\sigma}_i$ for a random time which serves as an upper bound on the first hitting time of $A \cup B_i$ for $\tilde{X}$, which is defined as:

$$\tilde{\sigma}_i = T \cdot \min\{c \in \mathbb{N} : cT \geq \sigma_i\},$$

i.e. the time to which $\tilde{X}$ is sampled in Algorithm 16 (so $\tilde{\sigma}_i$ is a multiple of T) in order to establish that a crossing into A or B_i has occurred. Similarly, let $\tilde{\tau}_i$ be the

corresponding upper bound on the first hitting time of B_i. To understand the exact MLS we describe later in this section, it is helpful to have in mind an idealised splitting scheme slightly different from MLS as presented in Section 3.2, which we call *idealised splitting with coupling*.

As in idealised MLS, we assume that it is possible to sample complete continuous paths of X up to a given stopping time. But rather than split these paths into *independent* copies at times τ_i, the split paths are coupled in the following way: from time τ_i until time $\tilde{\tau}_i$, the "split" paths are set to be identically equal, and after this time they evolve conditionally independently given $X_{\tilde{\tau}_i}$.

For $i = 1, \ldots, m$, let $\tilde{M}_i$ denote the transition kernels for the discrete time quadruple process $V_i = (\sigma_i, \tilde{\sigma}_i, X(\sigma_i), \tilde{X}(\tilde{\sigma}_i))$. Note that these are Markov kernels: in particular, $(\sigma_i, X(\sigma_i))$ depends only on either $(\tilde{\sigma}_{i-1}, \tilde{X}(\tilde{\sigma}_{i-1}))$ if $\tilde{\sigma}_{i-1} \leq \sigma_i$, or on $(\sigma_{i-1}, \tilde{\sigma}_{i-1}, X(\sigma_{i-1}), \tilde{X}(\tilde{\sigma}_{i-1}))$ if $\tilde{\sigma}_{i-1} > \sigma_i$ (since in this last case, for all $j < i-1$ it holds that either $\tilde{\sigma}_j = \tilde{\sigma}_i$ or $\tilde{\sigma}_j < \sigma_{i-1}$ - intuitively, "$\tilde{X}$ doesn't move until X catches up to it").

Define also $\tilde{G}_i(V_i) = \mathbb{I}_{B_i}(X(\sigma_i))$. The details are given in Algorithm 17. Call the estimator for p resulting from this algorithm $\tilde{p}$.

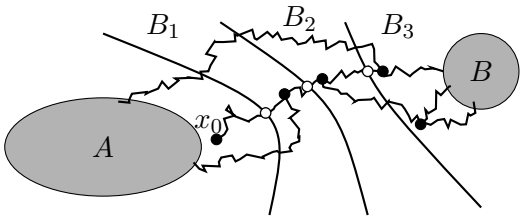


Figure 4.2: An illustration of Idealised Splitting with Couplings for a single particle system. The particle begins at the node labelled x_0. Level crossings are indicated by empty nodes, whereas splittings occur at the filled nodes. Between any empty node and the following filled node, the particle trajectories are coupled identically. Compare with Figure 3.1.

Of course, it is not possible to implement Algorithm 17 as written, since we cannot simulate full paths of X, nor make splits at times τ_i. But the construction of MLS with couplings means that an algorithm which splits paths at the tractable time $\tilde{\tau}_i$ instead (and allows them to propagate independently from that point) produces identical estimators $\tilde{q}_i$ for p_i.

It is possible that a particle crosses several barriers $z_i, z_{i+1}, \ldots$ before time $\tilde{\tau}_i$. In this case, the particle splits as normal at each barrier, each new copy remaining identically coupled, until the splitting time $\tilde{\tau}_i$ after which they proceed independently.

The following proposition establishes that this framework gives rise to unbiased estimates. In order to simplify the analysis, we make a further assumption about the ε-strong method being used: that over any interval $[s, t]$, we have $\left(\tilde{X}(t) \mid \tilde{X}(s) = x_s\right)$ is equal in distribution to $(X(t) \mid X(s) = x_s)$. In other words, we assume that the

Algorithm 17 Idealised Splitting with Coupling

Given λ together with $\tilde{G}_i$, $\tilde{M}_i$ for $i = 1, \ldots, m$, an initial number of particles N_0, and splitting ratios $R_1, \ldots, R_{m-1}$:

1. For $j = 1, \ldots, N_0$:
   (a) Draw $X_1^j(0) \sim \lambda$, $V_1^j \sim \tilde{M}_1\left(\left(0, 0, X_1^j(0), X_1^j(0)\right), \cdot\right)$.
2. Let $S_1 = \{V_1^j : \tilde{G}_1(V_1^j) = 1\}$ be a list of the the surviving paths, and $N_1 = |S_1|$.
3. For $i = 2, \ldots, m$:
   (a) If $N_{i-1} = 0$, return $\tilde{p} = 0$.
   (b) Otherwise given $S_{i-1} = \{V_{i-1}^j\}_{j=1}^{N_{i-1}}$, for each $(j,k) \in \{(j',k') : 1 \leq j' \leq N_{i-1}, 1 \leq k' \leq R_{i-1}\}$:
      i. Sample $V_i^{j,k} \sim \tilde{M}_i(V_{i-1}^j, \cdot)$
   (c) Let $S_i = \{V_i^{j,k} : \tilde{G}_i(V_i^{j,k}) = 1\}$, and set $N_i = |S_i|$.
4. Estimate
$$\tilde{p} = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}.$$

end-points of ε-strong samples are from the true distribution of X. This holds in the schemes developed in Beskos et al. [2012] and Pollock et al. [2016], for example, but not for the scheme of Blanchet et al. [2017]. But provided there is *some* way of extending an ε-strong path from the interval $[s,t]$ to $[s,t']$ for $t' > t$, a similar analysis should continue to hold.

Given the MLS with couplings scheme, it is now possible to establish this result with only minor modification of the existing arguments of Amrein and Künsch [2011]; this result paves the way for the methodology which we introduce and in principle allows for unbiased estimation of rare event probabilities in continuous time whenever ε-strong simulation of the process of interest is possible.

Proposition 2. *$\tilde{p}$ is an unbiased estimator for p: $\mathbb{E}[\tilde{p}] = p$.*

Proof. We consider the discrete-time Markov process $(V_i)_{i=0}^m$. Let $\tilde{M}_i$ denote the transition kernels of this process at time i. Let also $\tilde{M}_{i:j} = \tilde{M}_j \circ \tilde{M}_{j-1} \circ \cdots \circ \tilde{M}_{i+1}$ denote the elements of the associated two-parameter dynamic semigroup, describing the evolution of the process from time i to j. Note that for every i, $\Delta = \mathbb{R}^2 \times A \times \mathbb{R}^d$ is an absorbing state for $\tilde{M}_i$ since if $X(\sigma_i) \in A$, $\sigma_j = \sigma_i$ for all $j > i$.

Define $\mathcal{G}_0$ to be the sigma algebra generated by $\{V_0^j : j = 1, \ldots, N_0\}$, $\mathcal{G}_k = \mathcal{G}_{k-1} \vee \Sigma(V_k^j : 1 \leq j \leq N_k)$ for $k = 2, \ldots, m$, so that $(\mathcal{G}_k)_{k=1}^m$ is the natural filtration of the process $(V_i)_{i=0}^m$.

We observe the following recursion for the number of particles successfully reaching B_i given $\mathcal{G}_{i-1}$, which is immediate from the definition of M_i: that for any function $h_i : \mathbb{R}^2 \times (\mathbb{R}^d)^2 \to \mathbb{R}$ which is equal to 0 on Δ:

$$\mathbb{E}\left[\sum_{j=1}^{R_{i-1}N_{i-1}} h_i\left(V_i^j\right)\middle|\mathcal{G}_{i-1}\right] = R_{i-1}\sum_{k:\tilde{G}_{i-1}(V_{i-1}^k)=1}\int h_i(u)\tilde{M}_i\left(V_{i-1}^k, du\right). \tag{4.4}$$

(taking $R_0 = 1$) which follows since each V_i^j which is a descendent of any particular V_{i-1}^k has the same marginal law.

Define estimators $\tilde{p}_i$ for p_i in Algorithm 17, namely

$$\tilde{p}_i = \begin{cases} \frac{N_1}{N_0}, & i = 1, \text{ and} \\ \frac{N_i}{R_{i-1}N_{i-1}}, & 2 \leq i \leq m. \end{cases}$$

We show that for $1 \leq k$,

$$\mathbb{E}\left[\prod_{i=k}^{m}\tilde{p}_i\middle|\mathcal{G}_{k-1}\right] = \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(V_{k-1}^j)=1}\left(1-\tilde{M}_{(k-1):m}\left(V_{k-1}^j, \Delta\right)\right),$$

by backwards induction on k, starting with the case $k = m$. Note that for the case $k = 1$, each term in the sum on the RHS is the probability that a particle with a given starting value successfully reaches B before A. The result is then obtained upon taking a further expectation over the starting value.

The case $k = m$:

$$\begin{aligned}\tilde{p}_m &= \frac{N_m}{R_{m-1}N_{m-1}} \\ &= \frac{1}{R_{m-1}N_{m-1}}\sum_{j=1}^{R_{m-1}N_{m-1}}\tilde{G}_m\left(V_m^j\right)\end{aligned}$$

and the result then follows from taking the conditional expectation with respect to $\mathcal{G}_m$, and applying (4.4) with $h_i\left(\sigma, \tilde{\sigma}, X\left(\sigma\right), \tilde{X}(\tilde{\sigma})\right) = \mathbb{I}_{B_i}\left(X(\sigma)\right)$.

Now supposing the result holds for $k+1$, we show that it holds also for k. We have

the following chain of equalities (with the convention that $G_0(U_0^j) = 1$):

$$\mathbb{E}\left[\prod_{i=k}^{m}\tilde{p}_i\middle|\mathcal{G}_{k-1}\right] = \mathbb{E}\left[\tilde{p}_k\mathbb{E}\left[\prod_{i=k+1}^{m}\tilde{p}_i\middle|\mathcal{G}_k\right]\middle|\mathcal{G}_{k-1}\right] \tag{4.5}$$

$$= \mathbb{E}\left[\frac{N_k}{R_{k-1}N_{k-1}}\cdot\frac{1}{N_k}\sum_{j:\tilde{G}_k(V_k^j)=1}\left(1-\tilde{M}_{k:m}\left(V_k^j,\Delta\right)\right)\middle|\mathcal{G}_{k-1}\right] \tag{4.6}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(V_{k-1}^j)=1}\left(\int\left(1-\tilde{M}_{k:m}\left(u,\Delta\right)\right)\tilde{M}_k\left(V_{k-1}^j,du\right)\right) \tag{4.7}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(V_{k-1}^j)=1}\left(1-\int\tilde{M}_{k:m}\left(u,\Delta\right)\tilde{M}_k\left(V_{k-1}^j,du\right)\right) \tag{4.8}$$

$$= \frac{1}{N_{k-1}}\sum_{j:\tilde{G}_{k-1}(V_{k-1}^j)=1}\left(1-\tilde{M}_{(k-1):m}\left(V_{k-1}^j,\Delta\right)\right) \tag{4.9}$$

where (4.5) follows from the tower rule, and noting that $\tilde{p}_k$ is $\mathcal{G}_k$-measurable; (4.6) from the induction hypothesis; (4.7) from (4.4); (4.8) from expanding and integrating over $M_k(V_{k-1}^j, du)$; and (4.9) from the semigroup property.

Moreover, putting $k = 1$ we have

$$(4.9) = \frac{1}{N_{k-1}}\sum_{j=1}^{N_0}\left(1-M_{0:m}\left(V_0^j,\Delta\right)\right)$$

Since $\left(1-M_{0:m}\left(V_0^j,\Delta\right)\right) = \mathbb{P}\left(\tau_B < \tau_A|X(0)\right)$, we see that

$$\begin{aligned}\mathbb{E}\left[\tilde{p}\right] &= \mathbb{E}\left[\prod_{i=1}^{m}\tilde{p}_m\right]\\ &= \mathbb{P}(\tau_B < \tau_A)\end{aligned}$$

as desired. □

In this algorithm we have taken the simple choice to split paths of $\tilde{X}$ at the first (random) multiple of T after which a crossing is guaranteed, $\tilde{\tau}_i$. Since this could result in a long gap between the crossing time and the splitting time, i.e. a large $(\tilde{\tau}_i - \tau_i)$, this may introduce some unwanted variance into the estimation and various other approaches to splitting could be implemented. One simple alternative is to choose in advance a reasonably fine deterministic time-grid, and to split at the first location on the time-grid after which crossing is guaranteed.

The estimate p^{ex} given by Exact Multilevel Splitting (Algorithm 18) is exactly the same as that given by Idealised Splitting with Coupling (Algorithm 17), since the particle systems defined by these algorithms are essentially identical.

Corollary 1. *$\mathbb{E}[\hat{p}^{ex}] = p$ for all $N \geq 1$.*

Algorithm 18 Exact Multilevel Splitting

Given λ together with $\tilde{G}_i$, $\tilde{M}_i$ for $i = 1, \ldots, m$, an initial number of particles N_0, and splitting ratios $R_1, \ldots, R_{m-1}$:

1. Initialise $S_1 = \cdots = S_m = \emptyset$.
2. For $j = 1, \ldots, N_0$:
   (a) Draw $\tilde{X}_0^j(0) \sim \lambda$, and simulate $\tilde{X}_0^j[0:T]$ together with its skeleton $\tilde{Z}_0^j$ as per section 2.2.1.
   (b) Sample $(\tilde{G}_1(\tilde{Z}_1^j), \tilde{Z}_1^j) \sim$ Algorithm.16$(\tilde{Z}_0^j, A, B_1)$. If $\tilde{G}_1(\tilde{Z}_1^j) = 1$, add $\tilde{Z}_1^j$ to S_1.
3. Record $N_1 = |S_1|$.
4. For $i = 2, \ldots, m$:
   (a) if $N_{i-1} = 0$, return the estimate $\hat{p}^{\text{ex}} = 0$ of p.
   (b) Otherwise, given $S_{i-1} = \{\tilde{Z}_{i-1}^j\}_{j=1}^{N_{i-1}}$, for all pairs $\{(j,k)\}_{1 \le j \le N_{i-1}}^{1 \le k \le R_{i-1}}$ sample independently
   $$(\tilde{G}_i(\tilde{Z}_i^{j,k}), Z_i^{j,k}) \sim \text{ Algorithm.16}(\tilde{Z}_{i-1}^j, A, B_i),$$
   and if $\tilde{G}_i(\tilde{Z}_i^{j,k}) = 1$, add $\tilde{Z}_i^{j,k}$ to S_i.
   (c) Set $N_i =\mid S_i \mid$.
5. Return
$$\hat{p}^{\text{ex}} = \frac{N_m}{N_0 \prod_{i=1}^{m-1} R_i}.$$

Similarly to Algorithm 7, a Sequential Monte Carlo variant of Algorithm 18 may be constructed by replacing the splitting step with resampling: this is illustrated in Algorithm 19. Its advantages over Algorithm 18 are the same as the advantages of Algorithm 7 in Section 3.2, namely that there are no splitting ratios R_i which need to be calibrated to ensure a stable particle system.

Algorithm 19 Exact Multilevel Splitting via SMC

Given λ together with $\tilde{G}_i$, $\tilde{M}_i$ for $i = 1, \ldots, m$, and a fixed number of particles N:

...

4$(b)'$. Otherwise, given $S_{i-1} = \{\tilde{Z}_{i-1}^j\}_{j=1}^{N_{i-1}}$, sample $a_1, \ldots, a_N$ independently and uniformly at random (with replacement) from $1, \ldots, N_{i-1}$, and sample independently $(\tilde{Z}_i^j, \tilde{G}_i(\tilde{Z}_i^j)) \sim$ Algorithm.16$(\tilde{Z}_{i-1}^{a_j}, A, B_i)$.

...

5. Estimate
$$\hat{p}^{\text{SMC}} = \frac{\prod_{i=1}^m N_i}{(N)^m} = \prod_{i=1}^m \left(\frac{N_i}{N}\right).$$

4.5 Illustrations

The examples in this section were carried out using a single core on an Intel Xeon E5-2440 processor with an advertised clock speed of 2.40GHz.

4.5.1 Brownian motion in one dimension

Our first illustrative example uses the ε-strong scheme for Brownian motion of Beskos et al. [2012]; Pollock et al. [2016], in a setting in which the exact solution is known. In one dimension, the reaction co-ordinate may be taken to be the identity function. We choose $A = (-\infty, 0]$, $B = [3^{18}, \infty)$, $B_i = [3^i, \infty)$ for $i = 1, \ldots, 17$, with initial point $x_0 = 1$. It is well-known that for real $0 < a < b$, the probability that a Brownian path started at a reaches b before 0 is a/b — as can be verified with a simple optional stopping argument. Therefore the target probability is $3^{-18} \approx 2.58 \times 10^{-9}$.

The ε-strong scheme in question has some additional features which allow a substantial improvement in speed to that given in Algorithm 19 in this exceptionally simple setting. See sections 2.2.2 and 4.2; we are in fact implementing Algorithm 14. Over a given time interval $[s, t]$, for a Brownian motion $(X(r))_{r\in[s,t]}$ it is possible to sample bounds

$$L^{\downarrow}(s,t) \leq \inf_{r\in[s,t]} X(r) \leq L^{\uparrow}(s,t) \ , \ U^{\downarrow}(s,t) \leq \sup_{r\in[s,t]} X(r) \leq U^{\uparrow}(s,t).$$

An ε-strong algorithm over $[0, T]$ in the sense given in section 2.2.1 can be recovered by choosing partitioning $[0, T]$ into suitably small time intervals $[s_k, t_k]$, and taking

$$\tilde{X}^{\varepsilon}(r) = \frac{1}{2}(U^{\uparrow}(s_k, t_k) - L^{\downarrow}(s_k, t_k))$$

for $r \in [s_k, t_k]$. This corresponds to taking $C_k = [L^{\downarrow}(s_k, t_k), U^{\uparrow}(s_k, t_k)]$ in the notation of Section 4.4.1. This does not make use of the extra information available in $(L^{\uparrow}, U^{\downarrow})$. In particular, to assess whether X has crossed above the point b over $[s_k, t_k]$, it is sufficient to find that $U^{\downarrow}(s_k, t_k) > b$, since this guarantees the maximum of X is large enough. This is easier to check than the stricter condition that $C_k \subset (b, \infty)$, or equivalently that $L^{\downarrow}(s_k, t_k) > b$.

We compare the exact estimator to Euler-Maruyama-type schemes (see (4.2)) with three levels of resolution. The initial step sizes for the schemes are taken to be $0.01, 0.005$ and 0.001. In this example we exploit the simplicity of the problem at hand and the time-scaling property of Brownian motion to allow the Euler-Maruyama scheme to maintain a constant level of relative error over time: in each scheme, when level B_i is reached the step size is multiplied by $3^2 = 9$. This scaling which depends upon analytical techniques which would not be available in more realistic problems was essential in order to achieve a reasonable calculation time (this choice ensures the expected number of Euler-Maruyama steps until a crossing is decided remains constant as the scale of the problem grows). 500 estimators were produced for each procedure, each with population of 1000 particles. The results are shown in Figure 4.3.

Typical run-times for a single sample were 19s for the exact-MLS algorithm, 32s for the Euler-Maruyama-0.005 scheme, 157s for the Euler-Maruyama-0.001 scheme. This demonstrates that in favourable circumstances exact MLS can yield estimates of rare event probabilities at significantly lower computational cost than that at which

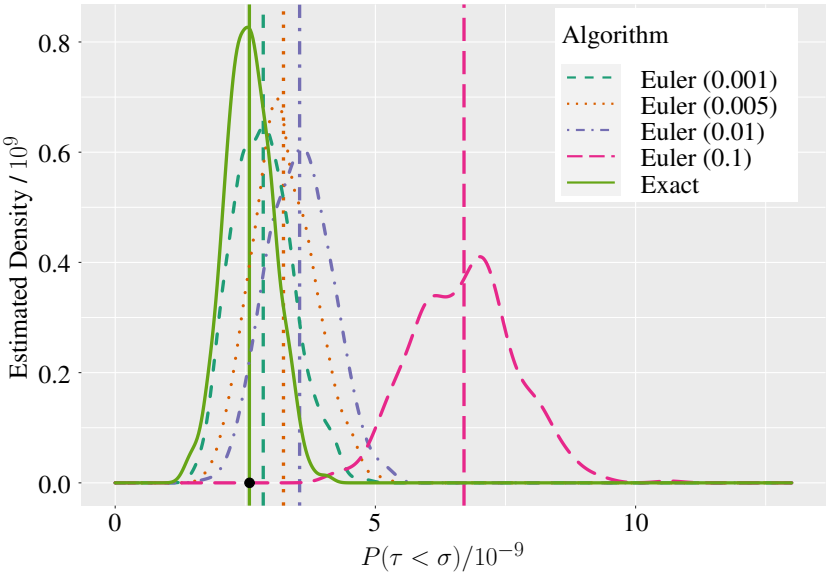


Figure 4.3: Kernel density estimates for the various schemes. The mean of each density estimate is shown as a coloured vertical line. The black dot on the ordinal axis indicates the true probability.

discretisation-based methods can reach an acceptable level of bias. In other settings the cost of exact methods can be somewhat higher, as the next example will demonstrate.

4.5.2 A bivariate example

Our second example illustrates Algorithm 19 in a pure form, is a two-dimensional problem. The random process is again taken to be Brownian motion initialised at $W_0 = (\frac{1}{2}, \frac{1}{2})$. The reaction co-ordinate is chosen to be $\xi(x, y) = \min(x, y)$, and the levels are chosen to be $A = \xi^{-1}((-\infty, 0)), B = \xi^{-1}((2^{\frac{20}{2}}, \infty)), B_i = \xi^{-1}(2^{(\frac{i}{2}+1)}, \infty))$ for $i = 1, \ldots, 18$. We are not aware of any simple means by which the rare event probability can be analytically obtained in this case.

The ε-strong algorithm used is that of Beskos et al. [2012]; Pollock et al. [2016], in the same fashion as Section 4.5.1. Again, the exact estimator is compared to three Euler-Maruyama schemes of (4.2) with increasing degrees of fineness, in this case with initial step-sizes 0.1, 0.05 and 0.01. The step-sizes were again re-scaled at each new level to ensure an approximately constant relative error, this time by a factor of $2^2 = 4$. For each scheme, 250 trials were simulated, each using 100 particles. Typical running times were 13s for the Euler-Maruyama-0.1 scheme, 21s for the Euler-Maruyama-0.05 scheme, 1m 45s for the Euler-Maruyama-0.05 scheme and 534m for the exact scheme.

Although the running time for exact MLS is significantly longer than that of the discrete schemes in this case, we believe that a more careful attempt to tune and adapt its parameters could substantially reduce the difference. However, the optimal choice of parameters will depend on the particular application, and since our aim is to provide

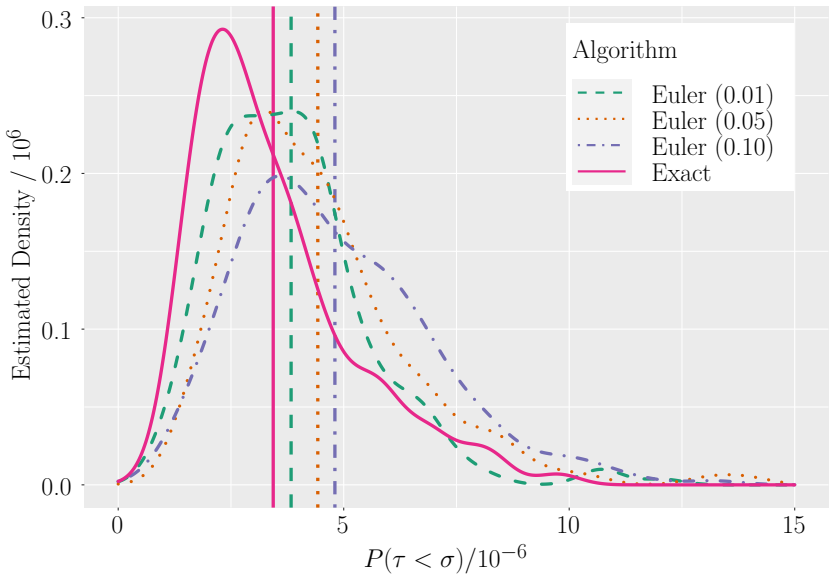


Figure 4.4: Kernel density estimates for exact MLS and for an Euler-Maruyama-discretisation-based method with three different discretisation step sizes. The mean of each density estimate is shown as a coloured vertical line.

proof-of-concept validation of the generic methodology developed in this paper we have not made extensive attempts to do so.

Figure 4.4 shows a kernel density estimate of the sampling distribution of the resulting estimators obtained using the `geom_density` function of ggplot2 Wickham [2016], using its default choice of bandwidth. As the Euler-Maruyama scheme increases in fineness, the resulting estimate appears closer to the estimate from exact MLS.

4.6 Discussion

We have presented the first algorithm for the exact estimation of a class of rare event probabilities for continuous-time processes. Our method has been to directly replace discrete approximations of continuous-time sample paths with ε-strong samples of the same paths, in order to obtain the sequence $\hat{p}_i$ of conditionally unbiased estimates required for multilevel splitting.

There is considerable ongoing effort in the development of ε-strong simulation methods for a broad class of stochastic processes, and their application (see for instance Mider et al. [2019]). It is likely that further development in this area will allow the approach described within this paper to be applied with greater efficiency to a broader class of stochastic processes. Naturally, it is likely that the exactness of this approach will always come at the expense of a computational cost that exceeds that of discretisation-based schemes. This cost can be partially offset against the need to assess the bias inherent in such schemes.

One important assumption that we have made is the ability to calculate infima and

suprema of the reaction co-ordinate over the sets C_k which constrain a sample path X. The problem of assessing whether these C_k intersect the MLS sets B_i has links to the problem of *collision detection* studied in computer graphics (see eg Kockara et al. [2007]). Insight from this field might allow a more careful classification of suitable ξ for a given C_k, or for the assumption to be weakened in certain circumstances.

As we noted in section 3.2, one strength of this method is that other quantities relating to the distribution of *paths* which reach the rare event set B may also be estimated. These require the sort of detailed information given by ε-strong simulation. However, if the rare event probability is the only quantity of interest, it is possible that approaches to obtaining unbiased estimates of rare event probabilities *without* requiring the full machinery of unbiased MLS implementations are sufficient.

The construction of computable unbiased estimators via a sequence of asymptotically biased estimators has received much attention in recent years, following Glynn and Rhee [2014]. This technique has been directly extended to estimating expectations of functionals of SDE paths in Rhee and Glynn [2015], and recently this approach has been further extended to non-linear filtering problems in Jasra et al. [2020]. It is to this idea that we turn in the subsequent chapters.

Chapter 5

A de-biased algorithm for rare events

5.1 Introduction

So far, we have relied on exact and ε-strong sampling of diffusion processes to address the discretisation bias described in section 4.1.1. An alternative to these methods is to make use of recent techniques for de-biasing statistical estimators - specifically, de-biasing estimators whose bias arises from the numerical discretisation of diffusion processes.

In this chapter, we describe an algorithm for de-biasing a certain rare event estimator using these techniques. In section 5.2, we describe the de-biasing method in the context of *multilevel Monte Carlo* (MLMC) estimation for functionals of diffusion processes. In section 5.3, we describe following Del Moral and Garnier [2005] the rare event setting we will work in, for which the estimator falls in the SMC framework described in terms of Feynman-Kac formulae. In section 5.4, we describe following Jasra et al. [2017] and Jasra et al. [2020] the application of MLMC and de-biasing to this context, and establish a de-biased estimation procedure for this setting.

5.2 Unbiased estimation for SDEs

De-biasing techniques may be illustrated by the example of modifying the (biased) *multilevel Monte Carlo* (MLMC) methods introduced in Heinrich [2001] and Giles [2008]. We describe this in the context of the discrete simulation of diffusions. As usual, let

$$dX(t) = a(X(t))dt + b(X(t))dB(t) \tag{5.1}$$

be the defining equation for the diffusion process X over the time interval $[0, T]$, assuming $X_0 = x_0$, and taking the usual conditions on a, b. For $\ell = 0, 1, \ldots$ let X^ℓ denote the process defined by an Euler-Maruyama discretisation of (5.1) using the step-size

$h_\ell = 2^{-\ell}$. Let π be the law of X, and π^ℓ the law of X^ℓ. Suppose that $\varphi : \mathbb{R}^d \to \mathbb{R}$ is a function for which $\pi^\ell(\varphi) \to \pi(\varphi)$ (as a real sequence) as $\ell \to \infty$.

Naive Monte Carlo estimation of $\pi(\varphi)$ might consist of choosing a discretisation level $L \in \mathbb{N}$ and generating N samples $X^{(i),L} \sim \pi^L$, $i = 1, \dots, N$ to produce the estimator $\frac{1}{N}\sum_{i=1}^N \varphi(X^{(i),L})$. The insight of MLMC is that the seemingly innocuous observation

$$\pi^L(\varphi) = \pi^0(\varphi) + \sum_{\ell=1}^{L} \left(\pi^\ell(\varphi) - \pi^{\ell-1}(\varphi)\right) \tag{5.2}$$

offers a way to more carefully design an estimator so as to minimise the cost required to obtain a given mean-squared error.

The idea is to produce independent Monte Carlo estimates of each term on the right side of (5.2), using N_ℓ Monte Carlo samples for the term $\pi^\ell - \pi^{\ell-1}$ where the N_ℓ are tuning parameters of the algorithm. An individual Monte Carlo estimate for one of these difference terms is constructed by producing *coupled* samples (X_1^ℓ, X_2^ℓ) whose left and right marginals are distributed according $\pi^\ell, \pi^{\ell-1}$ respectively, using two discretisations of *the same* driving Brownian motion on the dyadic grid $\mathcal{D}_n$ (see Algorithm 20 below):

Algorithm 20 Coupled Euler scheme

Given $X_1^\ell(t_0), X_2^\ell(t_0) = x_0$, and independent standard normal random variables W_k:

1. For $k = 0, \dots, (2^n T - 1)$:
   (a) If k even, set

$$X_1^\ell(t_{k+1}) = X_1^\ell(t_k) + a(X_1^\ell(t_k))h_\ell + b(X_1^\ell(t_k))\sqrt{h_\ell}W_k.$$

   (b) If k odd, set

$$\begin{aligned} X_1^\ell(t_{k+1}) &= X_1^\ell(t_k) + a(X_1^\ell(t_k))h_\ell + b(X_1^\ell(t_k))\sqrt{h_\ell}W_k, \\ X_2^\ell(t_{k+1}) &= X_2^\ell(t_{k-1}) + a(X_2^\ell(t_{k-1})) \cdot (2h_\ell) + b(X_2^\ell(t_{k-1})) \cdot \sqrt{h_\ell}(W_k + W_{k-1}) \end{aligned}$$

We shall sometimes refer to X_1^ℓ as the *fine* process and X_2^ℓ as the *coarse*.

These coupled samples can be used to obtain low-variance estimates of each term $\left(\pi^\ell(\varphi) - \pi^{\ell-1}(\varphi)\right)$. Using N_ℓ such samples for each term on the right side of (5.2), we define the MLMC estimator[1]

$$\hat{\pi}^{\mathrm{MLMC}}(\varphi) = \frac{1}{N_0}\sum_{i=1}^{N_0} \varphi(X^{0,i}) + \sum_{\ell=1}^{L} \frac{1}{N_\ell}\sum_{j=1}^{N_\ell} \left(\varphi(X_1^{\ell,j}) - \varphi(X_2^{\ell,j})\right).$$

One then chooses L and $N_0, \dots, N_L$ to minimise the MSE for a given cost. More

[1] The term *multilevel* has an entirely different meaning in this context than it does in *multilevel splitting*; in the latter case the "levels" are nested intermediate sets surrounding a rare set of interest, whereas in the former "levels" refers to the discretisation levels $h_\ell = 2^{-\ell}$.

precisely, writing

$$\Xi_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}\varphi(X^{0,i}),\ \Xi_\ell = \frac{1}{N_\ell}\sum_{j=1}^{N_\ell}\left(\varphi(X_1^{\ell,j}) - \varphi(X_2^{\ell,j})\right),$$

the MSE can be decomposed as

$$\begin{aligned}\mathbb{E}\left|\hat{\pi}^{\mathrm{MLMC}}(\varphi) - \pi(\varphi)\right|^2 &= \sum_{l=0}^{L}\mathbb{E}\left|\Xi_\ell - (\pi^\ell(\varphi) - \pi^{\ell-1}(\varphi))\right|^2 + (\pi^L(\varphi) - \pi(\varphi))^2 \\ &:= \sum_{\ell=0}^{L}\frac{V_\ell}{N_\ell} + B_L^2 \end{aligned} \tag{5.3}$$

(directly from the definition; see eg. [Jasra et al., 2017, Eqn (8)]). Here the first term on the right hand side is a sum of variances over the levels $\ell \leq L$, and the second term is the squared bias governed purely by the final level L. To obtain an MSE of roughly ε^2, one can tune the bias and variance terms to contribute roughly $\varepsilon^2/2$ each.

If B_L is of order $\mathcal{O}(h_L^\alpha)$, it is immediate that $L = -\left(\log\varepsilon/\sqrt{2}\right)/(\alpha\log 2) + \log(c_1)$ gives the correct contribution for the bias term, for some constant c_1. Then for any fixed $\mathcal{V}_L := \sum_{\ell=0}^{L} V_\ell/N_\ell$, supposing $V_\ell = \mathcal{O}(h_\ell^\beta)$ and the cost to sample X^ℓ is $\mathcal{O}(h_\ell^{-\gamma})$, one can minimise the total cost $\mathcal{C} = \sum_{\ell=0}^{L} N_\ell h_\ell^{-\gamma}$ subject to the variance using (for example) Lagrange multipliers, yielding $N_\ell \propto N_0 2^{-\ell(\gamma+\beta)/2}$. Finally, tuning N_0 to obtain the desired $\mathcal{V}_L = \varepsilon^2/2$ gives $N_0 \propto \varepsilon^{-2}\sum_{l=0}^{L} 2^{\ell(\gamma-\beta)/2}$. This defines three natural regimes for the total cost $\mathcal{C}$, depending on the sign of $(\gamma - \beta)$ (see for example [Jasra et al., 2017, Table 1]).

The value of γ depends only on the discretisation scheme. For the Euler-Maruyama scheme, it is known that $\gamma = 1$. The values of α, β depend both on the discretisation scheme and the nature of the test function φ. A case of special interest which is frequently referenced is the Euler-Maruyama scheme for Lipschitz test functions of the terminal value of the SDE: $\varphi(X) = \varphi(X_T)$. Here $\alpha = \beta = 1$. Also of interest with reference to the calculations in Chapter 6 is where φ is allowed a point of discontinuity, in which case it remains true that $\alpha = 1$, but β is reduced to $1/2$. All these results may be found in [Giles, 2008, Section 3], alongside a more detailed exposition of the above calculations. A more comprehensive overview of the properties, variants and applications of MLMC estimators may be found in Giles [2015].

Our interest here is not, however, in cost-variance trade-offs but in genuinely unbiased methods estimation methods. An insight of McLeish [2011], which has been applied to the discretisation of SDEs in Rhee and Glynn [2012] and Rhee and Glynn [2015], is that an MLMC estimator may be modified into an unbiased estimator as follows. Let now X_0 denote any unbiased estimator for $\pi^0(\varphi)$, and for each $\ell \in \mathbb{N}$ let Ξ_ℓ similarly be unbiased for $\pi^\ell(\varphi) - \pi^{\ell-1}(\varphi)$. Let L be a sample from some strictly positive density

$P(\cdot)$ over the non-negative integers. Then the estimator

$$\hat{p}^{\text{ub}} = \frac{\Xi_L}{P(L)}$$

is unbiased for p since

$$\begin{aligned}\mathbb{E}[\hat{p}^{\text{ub}}] &= \sum_{\ell=0}^{\infty} P(\ell) \frac{\mathbb{E}[\Xi_\ell]}{P(\ell)} \\ &= \pi^0(\varphi) + \sum_{\ell=1}^{\infty} \left(\pi^\ell(\varphi) - \pi^{\ell-1}(\varphi)\right) \\ &= \pi(\varphi)\end{aligned}$$

by the assumption that $\pi^\ell(\varphi) \to \pi(\varphi)$.

As suggested by our prior description of MLMC, an appropriate choice of Ξ_ℓ for this setting might be

$$\Xi_\ell = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} \varphi(X^{0,i}) & L = 0, \\ \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} \varphi(X_1^{\ell,j}) - \varphi(X_2^{\ell,j}) & \ell > 0, \end{cases}$$

where the $(X_1^{\ell,j}, X_2^{\ell,j})$ are coupled Euler samples as above. (The reason we use the more abstract Ξ_ℓ notation is that in more complex settings such as particle filtering, an unbiased estimator of the difference terms remains all that is required, though it might take a more complicated specific form.)

These de-biased MLMC estimators have been the subject of considerable further study. Conditions for finite variance of the estimators have been established in Rhee and Glynn [2015] and Blanchet et al. [2019], alongside numerical experiments in a wide range of applications. We quote the following useful result from Vihola [2018]:

Proposition 3. *[Vihola, 2018, Theorem 3] If*

$$\sum_{\ell=0}^{\infty} \frac{\mathbb{E}[\Xi_\ell^2]}{P(\ell)} < \infty,$$

then $\hat{p}^{ub}$ is a finite-variance unbiased estimator for p.

It is worth noting that in contrast to the exact estimator of the previous chapter, the de-biased estimator above can be negative, though it estimates a probability which is necessarily non-negative. In our context this causes no concern, but in some applications (eg. for use in the pseudo-marginal algorithms introduced in Andrieu and Roberts [2009]), one wishes to avoid negative values of the estimator. The problem of constructing unbiased estimators taking only non-negative values is addressed for example in Jacob and Thiery [2015].

MLMC methods have been applied to Sequential Monte Carlo algorithms in several

places: for example, to the filtering problem in Jasra et al. [2017], to the estimation of normalising constants in Jasra et al. [2018], and to Bayesian inverse problems in Beskos et al. [2017]. The de-biasing technique has also been applied in several related recent papers Jasra et al. [2020b,a]; Heng et al. [2021], and in particular to the filtering setting in Jasra et al. [2020]. This is closest in spirit to the problem analysed in this chapter and the following. The distinction is that the rare event setting we study (introduced below in section 5.3) involves estimating the unnormalised measure of a discontinuous function of an historical process, rather than the normalised (probability) measure of a Lipschitz test function of the time-marginal process. We expand on this point in Section 5.4 after introducing the appropriate notation.

5.3 Setting

Let $Z = (Z(t))_{t\in[0,T]}$ be the solution to a stochastic differential equation in $\mathbb{R}^d$ for some $T \in \mathbb{R}$, subject to the initial condition $X_0 \sim \eta_0$:

$$dZ(t) = a(Z(t))dt + b(Z(t))dB(t). \tag{5.4}$$

Here B_t is d-dimensional Brownian motion, $a : \mathbb{R}^d \to \mathbb{R}^d, b : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, and the following regularity conditions are satisfied:

1. *Ellipticity*: $b(z)b(z)^T$ is positive definite.
2. *Lipschitz continuity*: for some $C > 0$ it holds that

$$\max(||a(z) - a(w)||, ||b(z) - b(w)||) < C||z - w||.$$

3. *Bounded initial moments*: $\mathbb{E}||Z_0||^p < \infty$ for every $p \geq 1$.

Fix $n \in \mathbb{N}$. Let $K : (\mathbb{R}^d)^2 \to \mathbb{R}_{\geq 0}$ denote the density of the transition kernel for Z over time increments of length T/n, so writing $X_p = Z_{\frac{pT}{n}}$ we have that $(X_p)_{p=0}^n$ is a discrete-time chain with initial distribution η_0 and transition kernel K.

Let $V : \mathbb{R}^d \to \mathbb{R}$ be a reaction co-ordinate which we require to be Lipschitz continuous, and to satisfy for some $c > 0$ that $c^{-1} < V(x) < c$ for all x. Let $a \in \mathbb{R}$, and $A = V^{-1}([a, \infty)) \subset \mathbb{R}^d$ a rare set of interest. Our concern is to estimate the probability of the rare event

$$p_n = \mathbb{P}(X_n \in A).$$

A scheme for efficiently estimating p_n for the discrete-time chain (X_p) is described in Del Moral and Garnier [2005]. It works as follows: writing $\mathbb{P}_n$ for the path measure of

$(X_p)_{p=o}^n$, consider for $\lambda \in \mathbb{R}_+$ the twisted probability measure

$$\begin{aligned} d\mathbb{P}_n^\lambda &\propto \exp(\lambda V(X_n)) d\mathbb{P}_n \\ &= \prod_{p=1}^n \exp\{\lambda(V(X_p) - V(X_{p-1}))\} d\mathbb{P}_n \end{aligned}$$

assuming that $V(X_0) = 0$. This corresponds to the twisting of section 3.3.4 with the sequence of potentials $G_p(X_0, \ldots, X_p) = \exp(\lambda(V(X_p) - V(X_{p-1})))$. It is immediately evident that the rare event is much more likely under $\mathbb{P}_n^\lambda$ than $\mathbb{P}_n$, since the potentials G_p give very high weight to high values of V. This intuition may be quantified as follows:

$$\begin{aligned} p_n = \mathbb{P}_n(X_n \in A) &= \mathbb{E}_n[\mathbb{1}_A(V(X_n))] = \mathbb{E}_n^\lambda[\mathbb{1}_A(V(X_n)) \frac{d\mathbb{P}_n}{d\mathbb{P}_n^\lambda}] \\ &= \mathbb{E}_n^\lambda \left[\mathbb{1}_A(V(X_n)) \frac{\mathbb{E}[\exp(\lambda V(X_n))]}{\exp(\lambda V(X_n))} \right] \\ &\leq \exp(-\lambda a) \mathbb{E}[\exp(\lambda V(X_n))] \mathbb{E}_n^\lambda [\mathbb{1}_A(V(X_n))] \\ &= \exp(-\lambda a) \mathbb{E}[\exp(\lambda V(X_n))] \mathbb{P}_n^\lambda(X_n \in A). \end{aligned}$$

Define the *historical process* $(Y_p)_{p=0}^n$ associated to (X_p) by

$$Y_p = (X_0, \ldots, X_p).$$

We denote the transition kernel from $Y_{p-1} \to Y_p$ by M_p, $p = 1, \ldots, n$. Since the historical process is defined on the sequence of spaces $\mathbb{R}^{d \times (p+1)}$, M_p is a function $\mathbb{R}^{d \times (p+1)} \to \mathbb{R}^{d \times (p+2)}$ (and so really depends on p), and we have the identification

$$M_p\left((y_1, \ldots, y_{p-1}), (z_1, \ldots, z_{p-1}, z_p)\right) = \begin{cases} K(y_{p-1}, z_p) & (z_1, \ldots, z_{p-1}) = (y_1, \ldots, y_{p-1}), \\ 0 & \text{otherwise.} \end{cases}$$

For any sequence of potential functions $G_p : \mathbb{R}^{d \times p} \to \mathbb{R}^+$, define the Feynman-Kac measure (see section 3.3.4) for a test function $f \in \mathcal{B}_b(\mathbb{R}^{d \times n})$ by

$$\gamma_p(f) = \mathbb{E}\left[f(Y_p) \prod_{q=1}^{p-1} G_q(Y_q) \right].$$

Define also $\eta_p(f) = \frac{\gamma_p(f)}{\gamma_p(1)}$.

The rare event probability may then be written as

$$P(V(X_n) \geq a) = \gamma_n(S_n^{(a)}), \tag{5.5}$$

where $S_n^{(a)}(y_n) = \mathbb{1}(V(x_n) \geq a) \prod_{p=1}^{n-1} G_p(y_p)^{-1}$. We observe also the equivalent formula

$$P(V(X_n) \geq a) = \eta_n(S_n^{(a)}) \prod_{p=1}^{n-1} \eta_p(G_p),$$

(this is one of the basic Feynman-Kac relations given in 3.3.4).

To estimate $P(V(X_n) \geq a)$, we use a system of N particles $X_p = (X_p^1, \ldots, X_p^N)$, for $p = 0, \ldots, n$. We will abuse notation slightly to write

$$X_p^i = (X_{p,0}^i, \ldots, X_{p,p}^i)$$

where $X_{p,q}^i$ is the location of ancestor at time q of particle labelled "i" at time p. Then X_p^i is the history of the i-th particle up to time p. The transitions $X_{p-1} \to X_p$ consist of a resampling step from the distribution

$$\hat{X}_{p-1}^i \sim \frac{1}{\sum_{i=1}^N G_{p-1}(X_{p-1}^i)} \sum_{i=1}^N G_{p-1}(X_{p-1}^i) \delta_{X_{p-1}^i}(\cdot),$$

and a mutation step

$$X_p^i \sim M_p(\hat{X}_{p-1}^i, \cdot).$$

This may also be interpreted as sampling from the composite distribution

$$\mathbb{P}(X_p \in d(y_p^1, \ldots, y_p^N) | X_{p-1}) = \prod_{i=1}^N \psi_p(m(X_{p-1})) d(y_p^i),$$

where $m(x) = \frac{1}{N} \sum_{i=1}^N \delta_{x^i}(\cdot)$ and

$$\psi_p(\mu) = \frac{1}{\mu(G_{p-1})} \int \mu(dy_{p-1}) G_{p-1}(y_{p-1}) M_p(y_{p-1}, \cdot).$$

The resampling step here is exactly the multinomial sampling scheme described in section 3.3; this may be substituted for any of the lower-variance unbiased resampling schemes as suggested there, without compromising the unbiasedness of the resulting estimator.

At each time p, we have the empirical distribution associated with the particle system

$$\eta_p^N(\cdot) = \frac{1}{N} \sum_{i=1}^N \delta_{X_p^i}(\cdot)$$

and the corresponding unnormalised measures

$$\gamma_p^N(\cdot) = \eta_p^N(\cdot) \prod_{q=1}^{p-1} \eta_q^N(G_q).$$

Our estimator for the rare event is then

$$\hat{p}_n = \gamma_n^N(S_n^{(a)}).$$

Pseudocode for this procedure is given in Algorithm 21. Importantly, the estimate is unbiased: $\mathbb{E}[\gamma_n^N(S_n^{(a)})] = \gamma_n(S_n^{(a)}) = P(X_n \in A)$. This is established in [Del Moral et al., 2015, Lemma 2.1]. It is also shown to satisfy a central limit theorem.

Algorithm 21 Estimator with single system [Del Moral and Garnier, 2005]

Given initial distribution λ, potential functions $G_1, \ldots, G_{n-1}$ and transition kernels $M_1, \ldots, M_n$ for the historical process:

1. For each $i = 1, \ldots, N$, draw independently $X_0^i \sim \lambda$, and $X_1^i \sim M_1(X_0^i, \cdot)$.
2. For $p = 2, \ldots, n$:
   (a) For each i, calculate $G_{p-1}^i = G(X_{p-1}^i)$, and
   $$\bar{G}_{p-1} = \frac{1}{N}\sum_{i=1}^N G_{p-1}^i.$$
   (b) *Resampling*: sample $\tilde{X}_p^i$, $i = 1, \ldots, N$ according to the Boltzmann-Gibbs measure
   $$\frac{1}{N\bar{G}_{p-1}}\sum_{i=1}^N G_{p-1}^i \delta_{X_{p-1}^i}(\cdot).$$
   (c) *Mutation*: for each i sample $X_p^i \sim M_p(\tilde{X}_p^i, \cdot)$.
3. Estimate
   $$\hat{p}_n = \frac{1}{N}\sum_{i=1}^N \left(\mathbb{1}(V(X_n^i) \geq a) \prod_{p=1}^{n-1} G_p(X_p^i)^{-1} \right) \prod_{q=1}^{n-1} \bar{G}_q.$$

In practice, it is not possible to sample exactly from M. Instead one must make use of a numerical discretisation scheme, such as the following Euler-Maruyama method on a discrete grid, which we assume for convenience is the dyadic grid $\mathcal{D}_\ell = \{\frac{k}{2^\ell}; k = 0, \ldots, 2^\ell T\}$: initiate $X_{0,0}^\ell \sim \lambda$, and for $p = 0, \ldots, n$ and $k = 0, \ldots, \left(2^\ell T\right)/n$ define recursively

$$\begin{aligned} X_{p,k+1}^\ell &= X_{p,k}^\ell + a(X_{p,k}^\ell)h_\ell + b(X_{p,k}^\ell)h_\ell^{\frac{1}{2}} W_{p,k} \\ X_{p+1,0}^\ell &= X_{p,2^\ell}^\ell \end{aligned} \tag{5.6}$$

where $h_\ell = 2^{-\ell}$ is the mesh size of $\mathcal{D}_\ell$, and the $W_{p,k}$ are independent standard normal random variables. (Note that working on the dyadic grid assumes that $2^\ell T/n$ is an integer, which can always be guaranteed by a time rescaling of the original diffusion defined by (5.4), so there is no loss of generality in this assumption.) We identify $X_p^\ell := X_{p,0}^\ell$, and write K^ℓ for the transition kernel of the process X_p^ℓ. Thus if Q^ℓ denotes the transition kernel of a single step of the Euler scheme (5.6), K^ℓ is the $2^\ell T/n$-fold composition of Q^ℓ with itself.

Analogous to the previous example, we define $Y_p^\ell = (X_1^\ell, \ldots, X_p^\ell)$, with transitions M_p^ℓ,

and

$$\gamma_p^\ell(f) = \mathbb{E}\left[f(Y_p^\ell)\prod_{q=1}^{p-1} G_q(Y_q^\ell)\right],$$
$$\eta_p^\ell(f) = \frac{\gamma_p^\ell(f)}{\gamma_p^\ell(1)}.$$

As above, to estimate $P(V(X_n) \geq a))$ we use a system of N particles $X_p^\ell = (X_p^{\ell,1}, \ldots, X_p^{\ell,N})$. The transitions $X_{p-1}^\ell \to X_p^\ell$ use the mutation step

$$X_p^{\ell,i} \sim M^\ell(\hat{X}_{p-1}^{\ell,i}, \cdot), \tag{5.7}$$

and the combined resampling plus mutation distribution is

$$\mathbb{P}(X_p^\ell \in d(y_p^1, \ldots, y_p^N)|X_{p-1}^\ell) = \prod_{i=1}^N \psi_p^\ell(m(X_{p-1}^\ell))d(y_p^i),$$

where $m(x) = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}(\cdot)$ and

$$\psi_p^\ell(\mu) = \frac{1}{\mu(G_{p-1})}\int \mu(dy_{p-1})G_{p-1}(y_{p-1})M_p^\ell(y_{p-1}, \cdot).$$

At each time p, we have the empirical distribution associated with the particle system

$$\eta_p^{\ell,N} = \frac{1}{N}\sum_{i=1}^N \delta_{X_p^{\ell,i}}(\cdot)$$

and the corresponding unnormalised measures

$$\gamma_p^{\ell,N}(\cdot) = \eta_p^{\ell,N}(\cdot)\prod_{q=1}^p \eta_q^{\ell,N}(G_q).$$

The our estimator for the rare event is then $\hat{p}_n^\ell = \gamma_n^{\ell,N}(S_n^{(a)})$.

Since the Euler-Maruyama scheme (or any other discretisation scheme) gives only an approximation of the target diffusion process (5.4), if one runs Algorithm 21 with the choice of transition kernels $M_p = M_p^\ell$, then the resulting estimator $\hat{p}_n^\ell$ is no longer unbiased for p_n. In fact, the probability that it unbiasedly targets is

$$p_n^\ell = P(X_n^\ell \in A).$$

In the next section we explain how to alter the algorithm to target p_n instead.

5.4 De-biased algorithm

In the case of a continuous-time solution to an SDE, carrying out Algorithm 21 with the chain (X_p^ℓ) resulting from a numerical discretisation scheme (see section 2.1.4) will not produce an unbiased estimate of the desired rare event.

We propose a scheme to remove the bias which consists of randomising over the choice of discretisation level ℓ, and using a coupled pair of particle systems to target $p_n^\ell, p_n^{\ell-1}$ respectively.

First, we observe that the required weak convergence of (γ^ℓ) holds for the function $S_n^{(a)}$:

Lemma 13. *For all $n \in \mathbb{Z}_+$, $\gamma_n^\ell(S_n^{(a)}) \xrightarrow{w} \gamma_n(S_n^{(a)})$ as $\ell \to \infty$.*

Proof. We establish this result assuming the Euler-Maruyama method is used for the discretisation. We can write

$$\begin{aligned}\gamma_n^\ell(S_n^{(a)}) - \gamma_n(S_n^{(a)}) &= P(X_n^\ell \in A) - P(X^\ell \in A) \\ &= \mathbb{E}\left[\mathbb{1}_A(X_n^\ell) - \mathbb{1}_A(X_n)\right].\end{aligned}$$

[Giles et al., 2009, Section 3] gives an explicit convergence rate for the mean-squared error:

$$\mathbb{E}\left|\mathbb{1}_A(X_n^\ell) - \mathbb{1}_A(X_n)\right|^2 \leq h_\ell^{1/2-\delta}$$

for any $\delta > 0$. Hence

$$\mathbb{E}\left|\mathbb{1}_A(X_n^\ell) - \mathbb{1}_A(X_n)\right| \leq h_\ell^{1/4-\delta'}$$

for all $\delta' > 0$ and so (since $\mathbb{1}_A \geq 0$)

$$\mathbb{E}\left[\mathbb{1}_A(X_n^\ell)\right] \to \mathbb{E}\left[\mathbb{1}_A(X_n)\right]$$

□

(The weak error rate of 1 noted in Section 2.1.4 for the Euler scheme does not apply here, since it holds only for Lipschitz test functions. Convergence rates for non-Lipschitz functions of SDEs under the Euler scheme have been studied in other places: for example, functions of bounded variation in Avikainen [2007], and bounded measurable function in Bally and Talay [1994], under considerably stronger assumptions on the coefficients of the SDE.)

Therefore we can write

$$\gamma_n(S_n^{(a)}) = \gamma_n^0(S_n^{(a)}) + \sum_{\ell=1}^{\infty}\left(\gamma_n^\ell(S_n^{(a)}) - \gamma_n^{\ell-1}(S_n^{(a)})\right).$$

Then following section 5.2, if Ξ_ℓ is such that

$$\mathbb{E}[\Xi_\ell] = \gamma_n^\ell(S_n^{(a)}) - \gamma_n^{\ell-1}(S_n^{(a)}) \tag{5.8}$$

(here we use the convention that $\gamma_n^{-1}(f) = 0$ for any f), and $P(\cdot)$ is a strictly positive distribution over $\mathbb{N} \cup \{0\}$, then the estimator defined by sampling $L \sim P(\cdot)$, and taking

$$\hat{p}_n^{\mathrm{ub}} = \frac{\Xi_L}{P(L)}$$

is unbiased. Of course it is necessary to require of Ξ_ℓ that it produces an estimator with finite variance, for example using the condition of Proposition 3.

The problem of constructing such Ξ_ℓ in the context of Sequential Monte Carlo algorithms has been taken up in several places. For example, a de-biased method for filtering for diffusion processes, targeting $\eta_n(\varphi)$ for a bounded Lipschitz test function φ, was taken up in Jasra et al. [2020]. Analogously to the coupled Euler discretisation scheme we described for estimating functionals of diffusions, it is possible to use *coupled interacting particle systems* to construct suitable Ξ_ℓ. Our problem is not a filtering problem but does fall in the same Sequential Monte Carlo framework, so a similar design can be applied. Crucially, we exploit the fact that the estimator $\hat{p}_n^\ell = \gamma_n^{\ell,N}(S_n^{(a)})$ is unbiased for $p_n^\ell = \gamma_n^\ell(S_n^{(a)})$ as described in section 5.3, which allows unbiased Ξ_ℓ to be constructed fairly directly, without the complications necessitated estimating $\eta_n^\ell(\varphi)$ for which $\eta_n^{\ell,N}(\varphi)$ is *not* unbiased.

Note that Algorithm 21, being an instance of a Sequential Monte Carlo algorithm, consists of alternating between i) propagating a particle according to its transition kernel (or in practice, a discrete approximation of this kernel), and ii) a resampling procedure.

Denote by $\{(X_{p,1}^{\ell,i}, X_{p,2}^{\ell,i}) : \ i = 1, \ldots, N, \ p = 0, \ldots, n\}$ the particles in a coupled system where the particles $\{X_{p,1}^{\ell,i}\}$ marginally obey Algorithm 21 using transitions $M_1^\ell, \ldots, M_n^\ell$, and $\{X_{p,2}^{\ell,i}\}$ obey the same algorithm using transitions $M_1^{\ell-1}, \ldots, M_n^{\ell-1}$. In order to minimise the variance of Ξ_ℓ then, it must be ensured both that the pair $(X_{p,1}^{\ell,i}, X_{p,2}^{\ell,i})$ are propagated using highly correlated coupled transition kernels, but also that at each resampling step the j-th resampled indices I_1^j, I_2^j are chosen to be equal with high probability.

Coupling of the transition kernels may be dealt as in the case of plain MLMC for SDEs as in section 5.2, for example by using the coupled Euler scheme of Algorithm 20 (which we recast slightly to fit the present context in Algorithm 22). We denote by $\bar{M}_p^\ell$ the transition kernel associated to any coupled discretisation scheme for the historical process, of which Algorithm 22 is an example. That is, given the transition kernel M_p^ℓ of a discretisation scheme on a grid of mesh $2^{-\ell}$ over unit time intervals, then

$\bar{M}_p^\ell : \mathbb{R}^{d\times p} \times \mathbb{R}^{d\times(p+1)} \to \mathbb{R}_+$ is such that

$$\bar{M}_p^\ell \left((y, y') , B \times \mathbb{R}^{d\times(p+2)} \right) = M_p^\ell(y, B),$$
$$\bar{M}_p^\ell \left((y, y') , \mathbb{R}^{d\times(p+1)} \times B \right) = M_p^{\ell-1}(y', B).$$

We write $\bar{M}_{p,1}^\ell$ and $\bar{M}_{p,2}^\ell$ for these two marginals of the coupled kernel, so that $\bar{M}_{p,1}^\ell = M_p^\ell$ and $\bar{M}_{p,2}^\ell = M_p^{\ell-1}$.

As for the coupled resampling scheme, it should ensure a high probability of choosing identical indices I_1^j, I_2^j in each system subject to having the correct marginal distributions. The scheme described in Algorithm 23 is the *maximal coupling* for the two multinomial distributions in question, due to Jasra et al. [2017]. That is, it maximises the probability of sampling identical indices subject to preserving the relevant marginal distributions.

Although this simple scheme maximises the chance of sampling identical indices, it also enforces independent sampling from the marginal multinomial distributions with some probability (step 4). This adds substantially to the variance of multilevel schemes for Sequential Monte Carlo. Referring back to the bias-variance decomposition of Equation (5.3), the rate of decay $\mathcal{O}(h_\ell^\beta)$ for the ℓ-level variance V_ℓ for the Euler scheme and a Lipschitz test function is reduced to $\beta = 1/2$, compared to $\beta = 1$ for plain MLMC ([Jasra et al., 2017, Section 3.2]).

This simple method is not the only possible choice, and other coupled resampling schemes have been suggested to improve this rate. A good overview may be found in [Jacob et al., 2016, Section 2]. An example of an alternative is the *optimal Wasserstein coupling*, whose behaviour for the MLPF has been considered in Ballesio et al. [2020].

Pseudocode for the coupled estimation procedure for fixed ℓ, combining Algorithms 21 to 23, is given in Algorithm 24. A single iteration of the de-biased algorithm, including randomising over ℓ, is given in Algorithm 25. Finally, we note that the unbiasedness of the estimator allows M independent iterations of Algorithm 24 to be run in parallel, and their estimates averaged; this is described in Algorithm 26.

5.5 Feynman-Kac formalism

Marginally, the coupled particle systems of Algorithm 25 are particle approximations to the same Feynman-Kac model described in section 5.3. That is, using the subscript $j = 1, 2$ for the finer and coarser particle system respectively, the underlying target Feynman-Kac models for the marginal distributions are defined by

$$\gamma_{p,j}^\ell(\varphi) = \int \varphi(y_p) \prod_{q=1}^{p-1} G_q(y_q) \eta_0(dy_0) \prod_{q=1}^{p-1} M_{q,j}^\ell(y_{q-1}, dy_q)$$

Algorithm 22 Coupled Euler scheme for $\bar{M}^\ell$

To sample from $\bar{M}^\ell((X^\ell_{0,1}, X^\ell_{0,2}), \cdot)$, given independent standard normal random variables W_k:

1. For $k = 0, \ldots, 2^\ell - 1$:
   (a) If k even, set
   $$X^\ell_{k+1,1} = X^\ell_{k,1} + a(X^\ell_{k,1})h_\ell + b(X^\ell_{k,1})\sqrt{h_\ell}W_k.$$
   (b) If k odd, set
   $$X^\ell_{k+1,1} = X^\ell_{k,1} + a(X^\ell_{k,1})h_\ell + b(X^\ell_{k,1})\sqrt{h_\ell}W_k,$$
   $$X^\ell_{\frac{k+1}{2},2} = X^\ell_{\frac{k-1}{2},2} + a\left(X^\ell_{\frac{k-1}{2},2}\right)\cdot(2h_\ell) + b\left(X^\ell_{\frac{k-1}{2},2}\right)\cdot\sqrt{h_\ell}(W_k + W_{k-1})$$
2. Output $(X^\ell_{2^\ell,1}, X^\ell_{2^{\ell-1},2})$.

Algorithm 23 Coupled resampling (Jasra et al. [2017])

Given $G_1 = (G^1_1, \ldots, G^N_1), G_2 = (G^1_2, \ldots, G^N_2)$ where $G^i_j = G(X^i_j)$:

1. Set for $i = 1, \ldots, N$
   $$w^i_1 = \frac{G^i_1}{\sum_{j=1}^N G^j_1},\ w^i_2 = \frac{G^i_2}{\sum_{j=1}^N G^j_2}.$$
2. Letting $\tilde{w}^i = w^i_1 \wedge w^i_2$, calculate
   $$a = \sum_{i=1}^N \tilde{w}^i.$$
3. With probability a, for $q = \frac{\tilde{w}}{a}$ sample
   $$I_1 \sim \mathrm{Cat}\left(\{1, \ldots, N\}, q\right)$$
   and set $I_2 = I_1$.
4. Alternatively, with probability $(1 - a)$ define for $i = 1, \ldots, N$
   $$q^i_1 = \frac{w^i_1 - \tilde{w}^i}{\sum_{j=1}^N (w^j_1 - \tilde{w}^j)},\ q^i_2 = \frac{w^i_2 - \tilde{w}^i}{\sum_{j=1}^N (w^j_2 - \tilde{w}^j)}$$
   and sample independently
   $$I_1 \sim \mathrm{Cat}\left(\{1, \ldots, N\}, q_1\right),\ I_2 \sim \mathrm{Cat}\left(\{1, \ldots, N\}, q_2\right).$$
5. Return (I_1, I_2).

Algorithm 24 Coupled estimator

1. For each $i = 1, \dots, N$, draw independently $X_{0,1}^{\ell,i} \sim \eta_0$, and set $X_{0,2}^{\ell,i} = X_{0,1}^{\ell,i}$.
2. Sample independently $(X_{1,1}^{\ell,i}, X_{1,2}^{\ell,i}) \sim \bar{M}_1^{\ell}((X_{0,1}^{\ell,i}, X_{0,2}^{\ell,i}), \cdot)$ (for example, according to Algorithm 22).
3. For $p = 2, \dots, n$:
   (a) For each i, j calculate $G_{p-1,j}^{i} = G_{p-1}(X_{p-1,j}^{\ell,i})$, and

$$\bar{G}_{p-1,j} = \frac{1}{N} \sum_{i=1}^{N} G_{p-1,j}^{i}.$$

   (b) Resample $\tilde{X}_{p,j}^{\ell,i}$ for $i = 1, \dots, N$ and $j = 1, 2$, according to Algorithm 23.
   (c) For each i, sample $(X_{p,1}^{\ell,i}, X_{p,2}^{\ell,i}) \sim \bar{M}_p^{\ell}\left((\tilde{X}_{p,1}^{\ell,i}, \tilde{X}_{p,2}^{\ell,i}), \cdot\right)$.
4. Estimate

$$\begin{aligned}\hat{q}_n^{\ell} = & \frac{1}{N} \sum_{i=1}^{N} \Big(\mathbb{1}(V(X_{n,n,1}^{\ell,i}) \geq a) \prod_{p=1}^{n-1} \bar{G}_p(X_{p,1}^{\ell,i})^{-1} \\ & - \mathbb{1}(V(X_{n,n,2}^{\ell,i}) \geq a) \prod_{p=1}^{n-1} \bar{G}_p(X_{p,2}^{\ell,i})^{-1} \Big).\end{aligned}$$

Algorithm 25 Unbiased estimator

1. Sample $L \sim P(\cdot)$
   (a) If $L = 0$, run Algorithm 21 to obtain $\hat{q}_n^0$, and estimate

$$\hat{p}_n^{\mathrm{ub}} = \frac{\hat{q}_n^0}{P(0)}.$$

   (b) Else, estimate run Algorithm 24 with $\ell = L$ to obtain $\hat{q}_n^L$, and estimate

$$\hat{p}_n^{\mathrm{ub}} = \frac{\hat{q}_n^L}{P(L)}.$$

Algorithm 26 Unbiased estimator, multiple samples

1. For $i = 1, \dots, M$, sample $\hat{p}_n^{\mathrm{ub},(\mathrm{i})} \sim$ Algorithm 25
2. Estimate

$$\hat{p}_n^{\mathrm{ub},M} = \frac{1}{M} \sum_{i=1}^{M} \hat{p}_n^{\mathrm{ub},(i)}.$$

and

$$\eta_{p,j}^{\ell}(\varphi) = \frac{\gamma_{p,j}^{\ell}(f)}{\gamma_{p,j}^{\ell}(1)}.$$

Likewise, write $\eta_{n,1}^{\ell,N}$ and $\eta_{n,2}^{\ell,N}$ for the unnormalised empirical measures corresponding respectively to the finer and coarser marginals of the coupled scheme, and define for $j = 1, 2$

$$\gamma_{p,j}^{\ell,N}(\cdot) = \eta_{p,j}^{\ell,N}(\cdot) \prod_{q=1}^{p} \eta_{q}^{\ell,N}(G_q).$$

We also define a Feynman-Kac model for the coupled scheme. In particular, we define $\bar{\eta}_0^{\ell} = \eta_0^{\ell}$, and $\bar{\eta}_p^{\ell} = \bar{\Phi}_p^{\ell}(\bar{\eta}_{p-1}^{\ell})$ where $\bar{\Phi}_p^{\ell}$ corresponds to the combined coupled mutation plus coupled sampling step, for example an iteration of Algorithm 22 followed by an iteration of Algorithm 23:

$$\begin{aligned}\bar{\Phi}_p^{\ell}(\mu)(dv) = \mu(\bar{G}_{p-1,\mu}) \frac{\mu(\bar{G}_{p-1,\mu} K^{\ell}(\cdot, dv))}{\mu(\bar{G}_{p-1,\mu})} \\ - (1 - \mu(\bar{G}_{p-1,\mu}))\, (\mu \otimes \mu) \left([H_{p-1,1,\mu} \otimes H_{p-1,2,\mu}]\, \bar{K}_p^{\ell}(\cdot, dv) \right),\end{aligned}$$

where

$$\bar{G}_{p,\mu}(u) = \frac{G_p(u)}{\mu_1(G_p)} \wedge \frac{G_p(u)}{\mu_2(G_p)}, \quad H_{p,j,\mu} = \frac{\frac{G_p(u)}{\mu_1(G_p)} - \bar{G}_{p,\mu}(u)}{\mu\left(\frac{G_p(u)}{\mu_1(G_p)} - \bar{G}_{p,\mu}(u)\right)}$$

for $j = 1, 2$.

It may easily be verified then that $\bar{\eta}_{p,j}^{\ell} = \eta_{p,j}^{\ell}$ ie. the marginal of the joint distribution is correct (see [Jasra et al., 2017, Proposition A.1]).

We now turn to the analysis of the estimator defined in this chapter.

Chapter 6

Variance and numerical experiments

6.1 Introduction

We now turn our attention to the mathematical properties of the estimator described in Algorithm 25, assuming the use of the coupled Euler method (Algorithm 22) and the maximal coupled resampling scheme (Algorithm 23).

6.1.1 Assumptions

We begin by listing in one place all of the notable assumptions we place on the stochastic differential equation and the Feynman-Kac measures.

Assumptions on diffusion

1. $b(x)b(x)^T$ is positive definite.
2. a, b Lipschitz continuous.
3. $\mathbb{E}|X_0|^p < \infty$ for every $p \geq 1$.

Assumptions on G

1. There is a $c > 0$ such that $c^{-1} < G_p(y_p) < c$.
2. $G_p : \mathbb{R}^{d \times p} \to \mathbb{R}$ is Lipschitz continuous: there is a $C > 0$ such that

$$\left|G_p(y_p) - G_p(y'_p)\right| \leq C||y_p - y'_p||.$$

6.1.2 Useful results

Here we sketch some results from Jasra et al. [2017], Jasra et al. [2018] which we will make use of analysing the variance of our procedure. These results are established for the chain (X_p) rather than the historical process (Y_p), but continue to hold in our setting with near-identical proofs. We give one example proof (of Proposition 4) to illustrate the validity in the historical process setting.

Lemma 14. *C_m-inequality: let $a_1, \ldots, a_k \geq 0$. Then there is a constant $C(k)$ such that*

$$\left(\sum_{i=1}^{k} a_i\right)^m \leq C(k) \sum a_i^m.$$

Lemma 15. *Marcinkiewicz-Zygmund inequality: let $X_1, \ldots, X_N$ be centred independent RVs with finite m-th moments for all $m \geq 1$. Then for a constant $C(m)$ independent of n and of the distribution of the X_i, it holds that*

$$\mathbb{E}\left|\sum_{i=1}^{N} X_i\right|^m \leq C(m)\mathbb{E}\left[\left(\sum_{i=1}^{N} |X_i|^2\right)^{m/2}\right]$$

Lemma 16. *Grönwall inequality (integral form): let $u, \beta : [0,T] \to \mathbb{R}$ be continuous, $\alpha : [0,T] \to \mathbb{R}$ non-decreasing, and suppose that for all $t \in [0,T]$, we have*

$$u(t) \leq \alpha(t) + \int_0^t \beta(s)u(s)ds.$$

Then for all $t \in [0,T]$,

$$u(t) \leq \alpha(t) \exp \int_0^t \beta(s)u(s)ds.$$

We use the following notation: for a bounded function $\varphi : \mathbb{R}^{d\times n} \to \mathbb{R}$, define $||\varphi|| := \sup_{x\in\mathbb{R}^{d\times n}} ||\varphi(x)||$.

Proposition 4. *Jasra et al. [2017], Proposition C.6: Under the listed assumptions, for every $m \geq 1$ and for $j = 1, 2$, there is a constant $C(n, m)$ independent of ℓ, such that for all bounded φ*

$$\left(\mathbb{E}|\eta_{n,j}^{\ell,N}(\varphi) - \eta_{n,j}^{\ell}(\varphi)|^m\right)^{\frac{1}{m}} \leq \frac{C(n,m)||\varphi||}{\sqrt{N}}$$

We include the (short) proof of this result here to demonstrate clearly that the constant C is independent of ℓ, and that it is sufficient to take φ bounded (it need not be Lipschitz continuous, as is assumed in the original):

Proof. Induction on n. Case $n = 0$, by the Marcinkiewicz-Zygmund inequality we have

$$\begin{aligned}
\mathbb{E}|\eta_{0,j}^{\ell,N}(\varphi) - \eta_{0,j}^{\ell}(\varphi)|^m &= \mathbb{E}|\frac{1}{N}\sum_{i=1}^{N}\Big(\varphi(Y_{0,j}^{\ell,i}) - \eta_{0,j}^{\ell}(\varphi)\Big)|^m \\
&\leq \frac{C(m)}{N^m}\mathbb{E}\left[\left(\sum_{i=1}^{N}|\varphi(Y_{0,j}^{\ell,i}) - \eta_{0,j}^{\ell}(\varphi)|^2\right)^{\frac{m}{2}}\right] \\
&\leq \frac{C(m)}{N^m}\left(N||\varphi||^2\right)^{\frac{m}{2}} \\
&= \left(\frac{C(m)}{\sqrt{N}}||\varphi||\right)^m
\end{aligned}$$

For the induction step, by the triangle inequality (Minkowski)

$$\begin{aligned}
\left(\mathbb{E}|\eta_{n,j}^{\ell,N}(\varphi) - \eta_{n,j}^{\ell}(\varphi)|^m\right)^{\frac{1}{m}} \leq &\left(\mathbb{E}\left|\eta_{n,j}^{\ell,N}(\varphi) - [\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)\right|^m\right)^{\frac{1}{m}} \\
&+ \left(\mathbb{E}\left|[\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi) - \eta_{n,j}^{\ell}(\varphi)\right|^m\right)^{\frac{1}{m}}.
\end{aligned}$$

Define $\mathcal{F}_{n-1}^N$ to be the sigma algebra generated by the particle system up to time $n-1$. Then for the first term above, noting that for any test function φ it holds that $\mathbb{E}[\eta_{n,j}^{\ell,N}(\varphi)|\mathcal{F}_{n-1}^N] = [\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)$, by the tower law and the Marcinkiewicz-Zygmund inequality it follows that

$$\begin{aligned}
\mathbb{E}\left|\eta_{n,j}^{\ell,N}(\varphi) - [\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)\right|^m &= \mathbb{E}\left[\mathbb{E}\left[\left|\eta_{n,j}^{\ell,N}(\varphi) - [\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)\right|^m \mid \mathcal{F}_{n-1}^N\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^{N}\Big(\varphi(Y_{n,j}^{\ell,i}) - [\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)\Big)\right|^m \mid \mathcal{F}_{n-1}^N\right]\right] \\
&\leq \left(\frac{C(m)}{\sqrt{N}}||\varphi||\right)^m
\end{aligned}$$

as for $n = 0$. For the second term, we have

$$\begin{aligned}
\bar{\Phi}_n^{\ell}(\bar{\eta}_{n-1}^{\ell,N}) - \eta_{n,j}^{\ell}(\varphi) =& \frac{\eta_{n-1,j}^{\ell,N}(G_{n-1}M_{n,j}^{\ell}(\varphi))}{\eta_{n-1,j}^{\ell,N}(G_{n-1})} - \frac{\eta_{n-1,j}^{\ell}(G_{n-1}M_{n,j}^{\ell}(\varphi))}{\eta_{n-1,j}^{\ell,N}(G_{n-1})} \\
&+ \frac{\eta_{n-1,j}^{\ell}(G_{n-1}M_{n,j}^{\ell}(\varphi))}{\eta_{n-1,j}^{\ell,N}(G_{n-1})} - \frac{\eta_{n-1,j}^{\ell}(G_{n-1}M_{n,j}^{\ell}(\varphi))}{\eta_{n-1,j}^{\ell}(G_{n-1})} \\
=& \frac{1}{\eta_{n-1,j}^{\ell,N}(G_{n-1})}\left[\eta_{n-1,j}^{\ell,N}(G_{n-1}M_{n,j}^{\ell}(\varphi)) - \eta_{n-1,j}^{\ell}(G_{n-1}M_{n,j}^{\ell}(\varphi))\right] \\
&+ \frac{\eta_{n-1,j}^{\ell}(G_{n-1}M_{n,j}^{\ell}(\varphi))}{\eta_{n-1,j}^{\ell,N}(G_{n-1})\eta_{n-1,j}^{\ell}(G_{n-1})}\left[\eta_{n-1,j}^{\ell}(G_{n-1}) - \eta_{n-1,j}^{\ell,N}(G_{n-1})\right].
\end{aligned}$$

Then by the triangle inequality

$$
\begin{aligned}
&\left(\mathbb{E}|\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N}) - \eta_{n,j}^\ell(\varphi)|^m\right)^{\frac{1}{m}} \\
&\qquad \leq \left(\mathbb{E}\left|\frac{1}{\eta_{n-1,j}^{\ell,N}(G_{n-1})}\left[\eta_{n-1,j}^{\ell,N}(G_{n-1}M_{n,j}^\ell(\varphi)) - \eta_{n-1,j}^\ell(G_{n-1}M_{n,j}^\ell(\varphi))\right]\right|^m\right)^{\frac{1}{m}} \\
&\qquad + \frac{\eta_{n-1,j}^\ell(G_{n-1}M_{n,j}^\ell(\varphi))}{\eta_{n-1,j}^\ell(G_{n-1})}\left(\mathbb{E}\left|\frac{1}{\eta_{n-1,j}^{\ell,N}(G_{n-1})}\left[\eta_{n-1,j}^\ell(G_{n-1}) - \eta_{n-1,j}^{\ell,N}(G_{n-1})\right]\right|^m\right)^{\frac{1}{m}} \\
&\qquad \leq \frac{1}{\inf|G_{n-1}|}\left[\left(\mathbb{E}\left|\eta_{n-1,j}^{\ell,N}(G_{n-1}M_{n,j}^\ell(\varphi)) - \eta_{n-1,j}^\ell(G_{n-1}M_{n,j}^\ell(\varphi))\right|^m\right)^{\frac{1}{m}}\right. \\
&\qquad \left. + \frac{\eta_{n-1,j}^\ell(G_{n-1}M_{n,j}^\ell(\varphi))}{\eta_{n-1,j}^\ell(G_{n-1})}\left(\mathbb{E}\left|\eta_{n-1,j}^\ell(G_{n-1}) - \eta_{n-1,j}^{\ell,N}(G_{n-1})\right|^m\right)^{\frac{1}{m}}\right]
\end{aligned}
$$

Since by assumption G_{n-1}, $G_{n-1}M_{n,j}^\ell(\varphi)$ are bounded, the result then follows from the induction hypothesis. (Notice the constant C depends now on n.) □

Proposition 5. *[Jasra et al., 2018, Proposition 5.3]: under assumptions 6.1.1, there is a constant $C(n)$ independent of ℓ such that for $j = 1, 2$,*

$$\mathbb{E}[\left(\gamma_{n,j}^{\ell,N}(1) - \gamma_{n,j}^\ell(1)\right)^2] \leq \frac{C(n)}{N}.$$

To state the next results, we need some more notation. For two probability measures μ, ν defined on the measurable space $(\mathcal{X}, \mathcal{F})$, the *total variation distance* between μ, ν is

$$||\mu - \nu||_{\mathrm{TV}} = \sup_{E\in\mathcal{F}} |\mu(E) - \nu(E)|.$$

For two transition kernels M_1, M_2, let $\mathcal{A} = \{\varphi : ||\varphi|| \leq 1 \text{ and } \varphi \text{ Lipschitz}\}$, and define

$$|||M_1 - M_2||| = \sup_{\varphi\in\mathcal{A}} \sup_x |M_1(\varphi)(x) - M_2(\varphi)(x)|$$

We now define $B_\ell(n)$ as

$$B_\ell(n) = \left(\sum_{p=0}^n \mathbb{E}\left[(||Y_{p,1}^{\ell,1} - Y_{p,2}^{\ell,1}|| \wedge 1)^2\right]^{\frac{1}{2}} + \sum_{p=0}^n ||\eta_{p,1}^\ell - \eta_{p,2}^\ell||_{\mathrm{TV}} + \sum_{p=1}^n |||M_{p,1}^\ell - M_{p,2}^\ell|||\right)^2$$

Proposition 6. *[Jasra et al., 2018, Proposition 5.2]: under assumptions 6.1.1, for all $n \geq 0$ have*

$$\mathbb{E}\left[\left([\gamma_{n,1}^{\ell,N}(1) - \gamma_{n,2}^{\ell,N}(1)] - [\gamma_{n,1}^\ell(1) - \gamma_{n,2}^\ell(1)]\right)^2\right] \leq \frac{\bar{B}_\ell(n)}{N}$$

where $\bar{B}_\ell(n)$ is given explicitly by defining recursively $\bar{B}_\ell(0) = CB_\ell(0)$ for some constant

C independent of ℓ, and

$$\bar{B}_\ell(n) = C(n)\left(B_\ell(n-1) + \bar{B}_\ell(n-1) + ||\eta^\ell_{n-1,1} - \eta^\ell_{n-1,2}||^2_{TV} + (\gamma^\ell_{n-1,1}(1) - \gamma^\ell_{n-1,2}(1))^2\right),$$

We also make use of the following useful table of rates for the Euler scheme and coupled resampling scheme employed here:

Expression	Bound	Source
$\mathbb{E}[(\|\|X^{\ell,1}_{n,1} - X^{\ell,1}_{n,2}\|\| \wedge 1)^m]$	$\mathcal{O}(h_\ell^{1/2})$	[Jasra et al., 2017, Theorem D.5]
$\|\|\eta^\ell_{n,1} - \eta^\ell_{n,2}\|\|_{\text{TV}}$	$\mathcal{O}(h_\ell)$	[Jasra et al., 2017, Appendix D]
$\|\|\|M^\ell_{n,1} - M^\ell_{n,2}\|\|\|$	$\mathcal{O}(h_\ell)$	[Jasra et al., 2017, Appendix D]
$\|\gamma^\ell_{n,1}(1) - \gamma^\ell_{n,2}(1)\|^2$	$\mathcal{O}(h_\ell^2)$	[Jasra et al., 2018, Proposition 5.4]
$B_\ell(n)$	$\mathcal{O}(h_\ell^{1/2})$	[Jasra et al., 2017, Corollary D.6]
$\bar{B}_\ell(n)$	$\mathcal{O}(h_\ell^{1/2})$	[Jasra et al., 2018, Remark 5.1]

Table 6.1: Decay rates for various quantities for the coupled particle filter, assuming the use of the coupled Euler scheme, and the coupled resampling scheme given in Algorithm 23.

6.2 Variance

In Section 5.4, we defined an estimator for the rare event probability in our present setting: for $L \sim P(L)$,

$$\hat{p}^{\text{ub}}_n = \frac{1}{P(L)}\left(\gamma^{N,L}_{n,1}(S^{(a)}_n) - \gamma^{N,L}_{n,2}(S^{(a)}_n)\right).$$

In practice, one would run M independent copies of the particle system and use the averaged estimate $\hat{p}^{\text{ub},M}_n = \frac{1}{M}\sum_{j=1}^M \hat{p}^{\text{ub},(j)}_n$ (see Algorithm 26). This of course has a reduced variance compared to the single estimator, provided the variance of the single estimator is finite.

Here we examine the variance of the single estimator.

Theorem 17. *Suppose that a, b in the SDE equation (5.1) are bounded as well as Lipschitz continuous. Then there is a constant $C = C\left(n, ||S^{(a)}_n||\right)$ independent of ℓ such that*

$$\mathbb{E}\left[\left((\gamma^{\ell,N}_{n,1}(S^{(a)}_n) - \gamma^{\ell,N}_{n,2}(S^{(a)}_n) - (\gamma^\ell_{n,1}(S^{(a)}_n) - \gamma^\ell_{n,2}(S^{(a)}_n))\right)^2\right] \leq \frac{C}{N}h_\ell^{1/4}.$$

This is based on the following heuristic: Proposition 6 due to Jasra et al. [2018], combined with the corresponding rate $\mathcal{O}(h_\ell^{1/2})$ for $\bar{B}_\ell(n)$ given in Table 6.1, establishes a bound proportional to $h_\ell^{1/2}/N$ for the mean-squared error of the fixed-level normalising constant estimator $(\gamma^{\ell,N}_{n,1}(1) - \gamma^{\ell,N}_{n,2}(1))$.

Likewise, [Jasra et al., 2017, Theorem C.4] gives the same $h_\ell^{1/2}/N$ mean-squared error bound for the fixed-level filtering estimator $(\eta^{\ell,N}_{n,1}(\varphi) - \eta^{\ell,N}_{n,2}(\varphi))$, where φ is a Lipschitz

test function.

The exponent $1/2$ in $h_\ell^{1/2}$ appears as half the strong error rate for MLMC using the Euler scheme, known to be $\beta = 1$ for Lipschitz test functions as discussed in section 5.2. It is noted in [Jasra et al., 2017, p. 3075] that the reason for this halving of the strong error rate is the coupled resampling scheme employed in the multilevel particle filter; this conclusion is quantified in [Jasra et al., 2017, Theorem D.5].

In section 5.2, we also noted that MLMC estimation of a test function φ with a point of discontinuity reduces this error rate for the Euler scheme by a factor of $1/2$ as well. Therefore, it seems reasonable to aim for at worst a rate of $h_\ell^{1/4}$ for using a coupled SMC algorithm to estimate a Lipschitz function multiplied by an indicator function, as in the definition of $S_n^{(a)}$:

$$S_n^{(a)}(y_n) = \mathbb{1}(V(x_n) \geq a) \prod_{p=1}^{n-1} G_p(y_p)^{-1}.$$

The assumption of boundedness on a, b is not made in Jasra et al. [2017, 2018], and is used here to extend an argument to the present setting. It is not clear that it is strictly necessary, and our numerical experiment in Section 6.4 for which this assumption does not hold does not seem to perform worse than the theorem would suggest.

Throughout the following, C denotes a general constant which may change value from line to line, but which is always independent of ℓ.

For $n \geq 1$, noting that

$$\gamma_{n,j}^{\ell,N}(S_n^{(a)}) = \eta_{n,j}^{\ell,N}(S_n^{(a)})\gamma_{n,j}^{\ell,N}(1),$$

(see section 3.3.4), and using the elementary equality

$$\begin{aligned}(a_N A_N - b_N B_N) - (aA - bB) =& A_N\left((a_N - b_N) - (a - b)\right) \\ &+ b_N\left((A_N - B_N) - (A - B)\right) \\ &+ (A_N - A)(a - b) + (b_N - b)(A - B)\end{aligned}$$

with

$$\begin{aligned} a_N &= \eta_{n,1}^{\ell,N}(S_n^{(a)}),\ A_N = \gamma_{n,1}^{\ell,N}(1), \\ b_N &= \eta_{n,2}^{\ell,N}(S_n^{(a)}),\ B_N = \gamma_{n,2}^{\ell,N}(1), \\ a &= \eta_{n,1}^{\ell}(S_n^{(a)}),\ A = \gamma_{n,1}^{\ell}(1), \\ b &= \eta_{n,2}^{\ell}(S_n^{(a)})\ ,\ B = \gamma_{n,2}^{\ell}(1), \end{aligned}$$

the C_2-inequality implies that for a constant $C > 0$ which is independent of ℓ,

$$\begin{aligned}\mathbb{E}[((a_N A_N - b_N B_N) - (aA - bB))^2] \leq & C\Big(\mathbb{E}[A_N^2((a_N - b_N) - (a-b))^2] && (6.1)\\ & + \mathbb{E}[b_N^2((A_N - B_N) - (A-B))^2] && (6.2)\\ & + \mathbb{E}[(A_N - A)^2(a-b)^2] && (6.3)\\ & + \mathbb{E}[(b_N - b)^2(A-B)^2]\Big). && (6.4)\end{aligned}$$

We deal with these terms separately. For (6.2), since G_p is bounded below for $p = 1, \ldots, n-1$ (Assumption (1) of 6.1.1),

$$b_N = \eta_{n,2}^{\ell,N}(S_n^{(a)}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(V(X_{n,1}^{\ell,i}) \geq a) \prod_{p=1}^{n-1} G_p(Y_{p,1}^{\ell,i})^{-1} \leq C.$$

Hence applying Proposition 6 and consulting Table 6.1 gives the following bound:

$$\mathbb{E}[b_N^2 \left((A_N - B_N) - (A-B)\right)^2] \leq C\frac{\bar{B}_\ell(n)}{N} \leq \frac{C}{N}h_\ell^{1/2}. \tag{6.5}$$

Now $(a-b)^2, (A-B)^2$ are independent of N, so the corresponding terms (6.3), (6.4) may be bounded as follows:

$$\begin{aligned}\mathbb{E}[(b_N - b)^2(A-B)^2] &= (A-B)^2\mathbb{E}[(b_N - b)^2]\\ &\leq (A-B)^2\frac{C(n)||S_n^{(a)}||^2}{N}\\ &= \frac{C\left(n, ||S_n^{(a)}||\right)}{N}|\gamma_{n,1}^\ell(1) - \gamma_{n,2}^\ell(1)|^2 && (6.6)\\ &\leq \frac{C}{N}h_\ell^2 && (6.7)\end{aligned}$$

(from Proposition 4 and Table 6.1), and

$$\begin{aligned}\mathbb{E}[(A_N - A)^2(a-b)^2] &= (a-b)^2\mathbb{E}[(A_N - A)^2]\\ &\leq (a-b)^2\frac{C(n)}{N}\\ &\leq \frac{C(n)}{N}||\eta_{n,1}^\ell - \eta_{n,2}^\ell||_{\mathrm{TV}}^2 && (6.8)\\ &\leq \frac{C}{N}h_\ell^2 && (6.9)\end{aligned}$$

(from Proposition 5 and Table 6.1).

It finally remains to consider $((a_N - b_N) - (a-b))^2$. We follow the method of [Jasra et al., 2017, Theorem C.4], making departures where necessary to account for the fact

that $S_n^{(a)}$ is not Lipschitz. We have:

$$\mathbb{E}\left[((a_N - b_N) - (a-b))^2\right] \leq 2\mathbb{E}\left[\left((a_N - b_N) - ([\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_1(S_n^{(a)}) - [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_2(S_n^{(a)}))\right)^2\right] \tag{6.10}$$

$$+ 2\mathbb{E}\left[\left(([\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_1(S_n^{(a)}) - [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_2(S_n^{(a)})) - (a-b)\right)^2\right]. \tag{6.11}$$

For the first term (6.10), we condition on $\mathcal{F}_{n-1}^N$, and noting that for any test function φ it holds that $\mathbb{E}[\eta_{n,j}^{\ell,N}(\varphi)|\mathcal{F}_{n-1}^N] = [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_j(\varphi)$, we can apply the Marcinkiewicz-Zygmund inequality to obtain

$$\begin{aligned}
&\mathbb{E}\left[\left([\eta_{n,1}^{\ell,N}(S_n^{(a)}) - \eta_{n,2}^{\ell,N}(S_n^{(a)})] - \left[[\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_1(S_n^{(a)}) - [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_2(S_n^{(a)})\right]\right)^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left(\frac{1}{N}\sum_{i=1}^N \left(S_n^{(a)}(Y_{n,1}^{\ell,i}) - S_n^{(a)}(Y_{n,2}^{\ell,i})\right) - \left[[\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_1(S_n^{(a)}) - [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_2(S_n^{(a)})\right]\right)^2 |\mathcal{F}_{n-1}^N\right] \\
&\leq \frac{C}{N}\left(\mathbb{E}\left|S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right|^2 + \mathbb{E}\left|[\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_1(S_n^{(a)}) - [\bar{\Phi}_n^\ell(\bar{\eta}_{n-1}^{\ell,N})]_2(S_n^{(a)})\right|^2\right)
\end{aligned} \tag{6.12}$$

For the first term of (6.12) above, the fact that $S_n^{(a)}$ is not Lipschitz is crucial. It is, however, simply an indicator function multiplied by a Lipschitz function (by our Lipschitz assumption on the G_p, Assumption (2) of section 6.1.1). Therefore we use the following technique: let δ_ℓ be a small increment to be chosen later. Then we have the decomposition

$$\begin{aligned}
&\mathbb{E}\left[\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right] \\
&\qquad = \sum_{i=1}^8 \mathbb{E}\left[\mathbb{1}\left((V(X_{n,1}^{\ell,1}), V(X_{n,2}^{\ell,1})) \in B_i\right)\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right]
\end{aligned}$$

where the sets B_i are as laid out in Table 6.2. (See also the accompanying Figure 6.1):

These regions are specified for $V(X)$, the projection of X under the reaction co-ordinate. The cases B_2, B_3, B_4, B_5 are all similar in specifying events in which the image of the coarse and fine discretisations under V fall on either side of the rare event boundary a, *and* are at least δ_ℓ apart. Since we make us of these facts only, it is therefore sufficient to analyse these cases together.

Similarly B_6, B_7 specify events in which the image of the coarse and fine discretisations under V fall on either side of the rare event boundary a, and are *at most* $2\delta_\ell$ apart. Again, we analyse these cases together. Thus there are just four cases to consider.

Set	Case of corresponding term in sum
$B_1 = (-\infty, a)^2$	Neither of $V(X_1), V(X_2)$ reaches a
$B_2 = (-\infty, a-\delta_\ell) \times (a, \infty)$	$V(X_2)$ reaches a, $V(X_1)$ fails to reach by at least δ_ℓ
$B_3 = (a, \infty) \times (-\infty, a-\delta_\ell)$	$V(X_1)$ reaches a, $V(X_2)$ fails to reach by at least δ_ℓ
$B_4 = (a+\delta_\ell, \infty) \times (a-\delta_\ell, a)$	$V(X_1)$ clears a by at least δ_ℓ, $V(X_2)$ fails by at most δ_ℓ
$B_5 = (a-\delta_\ell, a) \times (a+\delta_\ell, \infty)$	$V(X_1)$ fails by at most δ_ℓ, $V(X_2)$ clears by at least δ_ℓ
$B_6 = (a-\delta_\ell, a) \times (a, a+\delta_\ell)$	$V(X_1)$ fails by at most δ_ℓ, $V(X_2)$ clears by at most δ_ℓ
$B_7 = (a, a+\delta_\ell) \times (a-\delta_\ell, a)$	$V(X_1)$ clears by at most δ_ℓ, $V(X_2)$ fails by at most δ_ℓ
$B_8 = (a, \infty)^2$	Both $V(X_1), V(X_2)$ reach a

Table 6.2: Eight sets corresponding to different relative positions for the fine and coarse marginals X_1^ℓ, X_2^ℓ

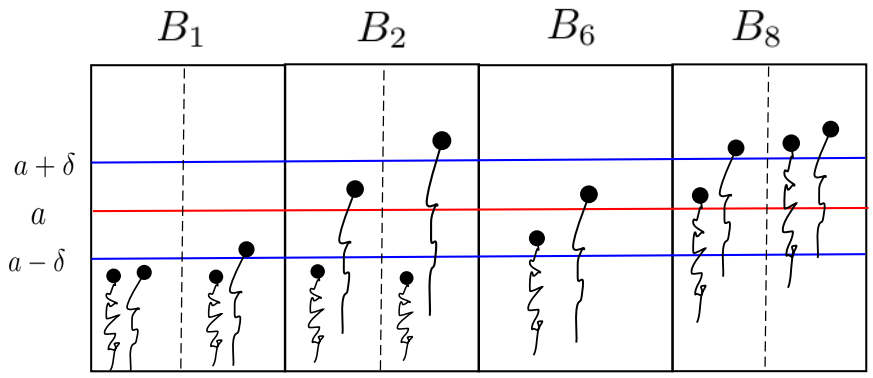


Figure 6.1: Schematic illustration of possible coupled path terminations under the image of the reaction co-ordinate for the sets B_j, $j = 1, 2, 6, 8$. Paths on the left of each pair correspond to the fine discretisation, those on the right to the coarse. Black dots mark the termination points. Two examples are given for $j = 1, 2, 8$ to show different possibilities.

1. $j = 1$:
$$\mathbb{E}\left[\mathbb{1}_{B_j}\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right] = 0.$$

2. For $j = 2, 3, 4, 5$, one of the two terms in the difference is identically 0, so:
$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{B_j}\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right] &\leq \prod_{p=1}^{n-1}\sup_{y_n}|G_p(y_p)|^2\mathbb{P}(B_j)\\ &\leq C\mathbb{P}(|V(X_{n,1}^{\ell,1}) - V(X_{n,2}^{\ell,1})| \geq 2\delta_\ell)\\ &\leq C\frac{\mathbb{E}\left|V(X_{n,1}^{\ell,1}) - V(X_{n,2}^{\ell,1})\right|}{\delta_\ell}\\ &\leq C\frac{\mathbb{E}\left[(|X_{n,1}^{\ell,1} - X_{n,2}^{\ell,1}| \wedge 1)\right]}{\delta_\ell} \qquad (6.13)\end{aligned}$$
where in the third line we have used the Markov inequality, and where the fourth line follows since V is bounded and Lipschitz (recall that we made this assumption in section 5.3).

3. $j = 6, 7$:
$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{B_j}\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right] &\leq 4\prod_{p=1}^{n-1}\sup_{y_n}|G_p(y_p)|^2\mathbb{P}(B_j)\\ &\leq C\ \mathbb{P}(V(X_{n,1}^{\ell,1}) \in (a - \delta_\ell, a))\\ &\leq C\sup|q_n(\cdot)|\delta_\ell \qquad (6.14)\end{aligned}$$
provided $V(X_{n,1}^{\ell,1})$ has a bounded density q_n.

4. For $j = 8$, $S_n^{(a)}$ is Lipschitz on (a, ∞), and also bounded, so we may immediately write
$$\mathbb{E}\left[\mathbb{1}_{B_j}\left(S_n^{(a)}(Y_{n,1}^{\ell,1}) - S_n^{(a)}(Y_{n,2}^{\ell,1})\right)^2\right] \leq \mathbb{E}\left[(|X_{n,1}^{\ell,1} - X_{n,2}^{\ell,1}| \wedge 1)^2\right] \qquad (6.15)$$

Putting these cases together, for the first term $(*)$ of 6.12 we have the bound
$$(*) \leq \frac{C}{N}\left(\delta_\ell + \frac{1}{\delta_\ell}\mathbb{E}\left[(|X_{n,1}^{\ell,1} - X_{n,2}^{\ell,1}| \wedge 1)\right] + \mathbb{E}\left[(|X_{n,1}^{\ell,1} - X_{n,2}^{\ell,1}| \wedge 1)^2\right]\right). \qquad (6.16)$$

Moreover, from Table 6.1 we have that $\mathbb{E}\left[(|X_{n,1}^{\ell,1} - X_{n,2}^{\ell,1}| \wedge 1)^m\right] = \mathcal{O}(h_\ell^{1/2})$, so the optimal choice $\delta_\ell = h_\ell^{1/4}$ gives
$$(*) \leq \frac{C}{N}(\mathcal{O}(h_\ell^{1/4}) + \mathcal{O}(h_\ell^{1/4}) + \mathcal{O}(h_\ell^{1/2})) = \mathcal{O}(h_\ell^{1/4}).$$

For the second term of (6.12), we make use of the decomposition

$$\begin{aligned}
&\bar{\Phi}(\bar{\eta}_{n-1}^{\ell,N})_1(S_n^{(a)}) - \bar{\Phi}(\bar{\eta}_{n-1}^{\ell,N})_2(S_n^{(a)}) \\
&= \left(\frac{1}{\eta_{n-1,1}^{\ell,N}(G_{n-1})} \left(\eta_{n-1,1}^{\ell,N}(G_{n-1} M_{n,1}^{\ell}(S_n^{(a)})) - \eta_{n-1,1}^{\ell,N}(G_{n-1} M_{n,2}^{\ell}(S_n^{(a)})) \right) \right. && (6.17) \\
&+ \frac{1}{\eta_{n-1,1}^{\ell,N}(G_{n-1})} \left(\eta_{n-1,1}^{\ell,N}(G_{n-1} M_{n,2}^{\ell}(S_n^{(a)})) - \eta_{n-1,2}^{\ell,N}(G_{n-1} M_{n,2}^{\ell}(S_n^{(a)})) \right) && (6.18) \\
&+ \left. \frac{\eta_{n-1,2}^{\ell,N}(G_{n-1} M_{n,2}^{\ell}(S_n^{(a)}))}{\eta_{n-1,1}^{\ell,N}(G_{n-1}) \eta_{n-1,2}^{\ell,N}(G_{n-1})} \left(\eta_{n-1,2}^{\ell,N}(G_{n-1}) - \eta_{n-1,1}^{\ell,N}(G_{n-1}) \right) \right) && (6.19)
\end{aligned}$$

The final term (6.19) may be bounded similarly to the calculation in the appendix, see [Jasra et al., 2017, Eqn (29)]: since G_{n-1} is bounded above and below, $S_n^{(a)}$ is bounded above, and G_{n-1} is Lipschitz continuous, for a constant C independent of ℓ

$$\begin{aligned}
&\frac{\eta_{n-1,2}^{\ell,N}(G_{n-1} M_{n,2}^{\ell}(S_n^{(a)}))}{\eta_{n-1,1}^{\ell,N}(G_{n-1}) \eta_{n-1,2}^{\ell,N}(G_{n-1})} \left| \eta_{n-1,2}^{\ell,N}(G_{n-1}) - \eta_{n-1,1}^{\ell,N}(G_{n-1}) \right| \\
&\leq \frac{C}{N} \sum_{i=1}^{N} \left| G_{n-1}(Y_{n-1,1}^{\ell,i}) - G_{n-1}(Y_{n-1,1}^{\ell,i}) \right| \\
&\leq \frac{C}{N} \sum_{i=1}^{N} \left(||Y_{n-1,1}^{\ell,i} - Y_{n-1,2}^{\ell,i}|| \wedge 1 \right). && (6.20)
\end{aligned}$$

The bounded path discrepancy $\mathbb{E}\left[\left(||Y_{n-1,1}^{\ell,i} - Y_{n-1,2}^{\ell,i}|| \wedge 1 \right) \right] = \mathcal{O}(h_\ell^{1/2})$ since

$$\begin{aligned}
\mathbb{E}\left[\left(||Y_{n-1,1}^{\ell,i} - Y_{n-1,2}^{\ell,i}|| \wedge 1 \right) \right] &\leq \mathbb{E}\left[\left(\sum_{p=0}^{n-1} ||X_{p,1}^{\ell,i} - X_{p,2}^{\ell,i}|| \wedge 1 \right) \right] \\
&\leq C\mathbb{E}\left[\sum_{p=0}^{n-1} \left(||X_{p,1}^{\ell,i} - X_{p,2}^{\ell,i}|| \wedge 1 \right) \right] \\
&= C \sum_{p=0}^{n-1} \mathbb{E}\left[\left(||X_{p,1}^{\ell,i} - X_{p,2}^{\ell,i}|| \wedge 1 \right) \right] \\
&= \mathcal{O}(h_\ell^{1/2}),
\end{aligned}$$

by Table 6.1, so the term (6.20) can be controlled.

For (6.17), since G_{n-1} is bounded above and below we have

$$\begin{aligned}
&\frac{1}{\eta^{\ell,N}_{n-1,1}(G_{n-1})}\left|\eta^{\ell,N}_{n-1,1}(G_{n-1}M^{\ell}_{n,1}(S^{(a)}_n)) - \eta^{\ell,N}_{n-1,1}(G_{n-1}M^{\ell}_{n,2}(S^{(a)}_n))\right| \\
\leq &\frac{C}{N}\sum_{i=1}^{N} G_{n-1}(Y^{\ell,i}_{n-1,1})\left|M^{\ell}_{n,1}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1}) - M^{\ell}_{n,2}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1})\right| \\
\leq &\frac{C}{N}\sum_{i=1}^{N}\left|M^{\ell}_{n,1}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1}) - M^{\ell}_{n,2}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1})\right|. \qquad (6.21)
\end{aligned}$$

We deal with this term using a similar eight-fold decomposition to that described previously. In particular, writing $\mathcal{B}_j$ for the pre-image of B_j under the Cartesian product function $(V^{-1}\times V^{-1})$:

$$\begin{aligned}
&M^{\ell}_{n,1}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1}) - M^{\ell}_{n,2}(S^{(a)}_n)(Y^{\ell,i}_{n-1,1}) \\
&\qquad = \int S^a_n(y_1)M^{\ell}_{n,1}(Y^{\ell,i}_{n-1,1}, dy_1) - \int S^a_n(y_2)M^{\ell}_{n,2}(Y^{\ell,i}_{n-1,1}, dy_2) \\
&\qquad = \int (S^a_n(y_1) - S^a_n(y_2))\,\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), d(y_1,y_2)) \\
&\qquad = \sum_{j=1}^{8}\int_{\mathcal{B}_j} (S^a_n(y_1) - S^a_n(y_2))\,\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), d(y_1,y_2)).
\end{aligned}$$

As before, we can divide into four cases:

1. $j = 1$:
$$\int_{\mathcal{B}_j} (S^a_n(y_1) - S^a_n(y_2))\,\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), d(y_1,y_2)) = 0.$$

2. $j = 2,3,4,5$:
$$\begin{aligned}
\int_{\mathcal{B}_j} (S^a_n(y_1) - S^a_n(y_2))\,\bar{M}^{\ell}((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), d(y_1,y_2)) &\leq C\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), \mathcal{B}_j) \\
&\leq C\mathbb{P}(|V(X^{\ell}_{1,1}) - V(X^{\ell}_{1,2})| > 2\delta_\ell) \\
&\leq C\frac{\mathbb{E}\left[(|X^{\ell}_{1,1} - X^{\ell}_{1,2}| \wedge 1)\right]}{\delta_\ell} \qquad (6.22)
\end{aligned}$$

3. $j = 6,7$:
$$\begin{aligned}
\int_{\mathcal{B}_j} (S^a_n(y_1) - S^a_n(y_2))\,\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), d(y_1,y_2)) &\leq C\bar{M}^{\ell}_n((Y^{\ell,i}_{n-1,1}, Y^{\ell,i}_{n-1,1}), \mathcal{B}_j) \\
&\leq C\mathbb{P}(V(X^{\ell,i}_{n,1}) \in (a-\delta_\ell, a)) \\
&\leq C\sup|q^n(\cdot)|\delta_\ell \qquad (6.23)
\end{aligned}$$

4. $j = 8$:

$$\int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\, \bar{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,1}^{\ell,i}), d(y_1, y_2)) \leq C|||M_{n,1}^\ell - M_{n,2}^\ell|||, \tag{6.24}$$

since $S_n^{(a)}$ is Lipschitz continuous over this region.

As before, putting these together we obtain

$$|(6.17)| \leq C\left(\delta_\ell + \frac{1}{\delta_\ell}\frac{1}{N}\sum_{i=1}^N \mathbb{E}\left[(|X_{1,1}^{\ell,i} - X_{1,2}^{\ell,i}| \wedge 1)\right] + |||M_{n,1}^\ell - M_{n,2}^\ell|||\right) \tag{6.25}$$

so choosing $\delta_\ell = h_\ell^{1/4}$ as before we get (6.17) $= \mathcal{O}(h_\ell^{1/4})$ (this is inside an expression to be squared, so will eventually contribute $\mathcal{O}(h_\ell^{1/2})$).

For (6.18), we have

$$\begin{aligned}
&\frac{1}{\eta_{n-1,1}^{\ell,N}(G_{n-1})}\left|\eta_{n-1,1}^{\ell,N}(G_{n-1}M_{n,2}^\ell(S_n^{(a)})) - \eta_{n-1,2}^{\ell,N}(G_{n-1}M_{n,2}^\ell(S_n^{(a)}))\right| \\
\leq& C\frac{1}{N}\sum_{i=1}^N \left|G_{n-1}(Y_{n-1,1}^{\ell,i})M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,1}^{\ell,i}) - G_{n-1}(Y_{n-1,2}^{\ell,i})M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,2}^{\ell,i})\right| \\
=& C\Bigg(\frac{1}{N}\sum_{i=1}^N \Big|\left[G_{n-1}(Y_{n-1,1}^{\ell,i}) - G_{n-1}(Y_{n-1,2}^{\ell,i})\right] M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,1}^{\ell,i}) \\
&+ \frac{1}{N}\sum_{i=1}^N G_{n-1}(Y_{n-1,2}^{\ell,i})\left[M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,1}^{\ell,i}) - M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,2}^{\ell,i})\right]\Big|\Bigg) \\
\leq& C\Bigg(\frac{1}{N}\sum_{i=1}^N \left(||Y_{n-1,1}^{\ell,i} - Y_{n-1,2}^{\ell,i}|| \wedge 1\right) \\
&+ \frac{1}{N}\sum_{i=1}^N \left|M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,1}^{\ell,i}) - M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,2}^{\ell,i})\right|\Bigg)
\end{aligned} \tag{6.26}$$

To bound the term (6.26), we again make use of the eight-fold decomposition. Let $\tilde{M}_n^\ell$ be any coupled transition kernel with left and right marginals both equal to $M_{n,2}^\ell$. Then:

$$\begin{aligned}
&M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,1}^{\ell,i}) - M_{n,2}^\ell(S_n^{(a)})(Y_{n-1,2}^{\ell,i}) \\
&\qquad = \int S_n^a(y_1) M_{n,2}^\ell(Y_{n-1,1}^{\ell,i}, dy_1) - \int S_n^a(y_2) M_{n,2}^\ell(Y_{n-1,2}^{\ell,i}, dy_2) \\
&\qquad = \int (S_n^a(y_1) - S_n^a(y_2))\, \tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)) \\
&\qquad = \sum_{j=1}^8 \int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\, \tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)).
\end{aligned} \tag{6.27}$$

We obtain the desired bound by a suitable choice of coupling $\tilde{M}_n^\ell$. Consider the choice of coupling for which the marginal Euler schemes are defined using common normal random variables, as described in Algorithm 27 below:

Algorithm 27 Coupled Euler scheme for $\tilde{M}_n^\ell$

To sample from $\tilde{M}_n^\ell((Y_{n-1,1}^\ell, Y_{n-1,2}^\ell), \cdot)$, given independent standard normal random variables W_k:

1. Set $Z_{0,1} = X_{n-1,1}^\ell$ and $Z_{0,2} = X_{n-1,2}^\ell$
2. For $k = 0, \ldots, 2^{\ell-1} - 1$, set:

$$Z_{k+1,1} = Z_{k,1} + a(Z_{k,1})h_\ell + b(Z_{k,1})\sqrt{h_\ell}W_k,$$
$$Z_{k+1,2} = Z_{k,2} + a(Z_{k,2})h_\ell + b(Z_{k,2})\sqrt{h_\ell}W_k$$

3. Output

$$\left(X_{n,1}^\ell, X_{n,2}^\ell\right) := \left(Z_{2^{\ell-1},1}, Z_{2^{\ell-1},2}\right).$$

Now for $t \in [0,1]$ and $j = 1, 2$, let $\tilde{Z}_j(t)$ be the continuous time process defined by

$$\tilde{Z}_j(k/2^{\ell-1}) := Z_{k,j}$$

for $k = 0, \ldots, 2^{\ell-1}$, where the $Z_{k,j}$ are as in Algorithm 27, and $\tilde{Z}_j(t)$ given by linear interpolation between these values for all other times t. Then [Rogers and Williams, 2000, Corollary V.11.7] guarantees that there is a constant C independent of ℓ such that

$$\mathbb{E}\left[\sup_{t\in[0,1]} |\tilde{Z}_1(t) - \tilde{Z}_2(t)|\right] \leq C\left(\mathbb{E}|\tilde{Z}_1(0) - \tilde{Z}_2(0)| + \mathbb{E}\left[\int_0^t \sup_{s\in[0,t]} |\tilde{Z}_1(s) - \tilde{Z}_2(s)|\right]\right).$$

As noted in [Jasra et al., 2017, Proposition D.1], combining this result with the Grönwall inequality (Lemma 16) with $u(t) = \mathbb{E}\left[\sup_{t\in[0,1]} |\tilde{Z}_1(t) - \tilde{Z}_2(t)|\right]$, $\alpha(t) = C\mathbb{E}|\tilde{Z}_1(0) - \tilde{Z}_2(0)|$ and $\beta(t) = C$, yields that

$$\begin{aligned}\mathbb{E}\left[\sup_{t\in[0,1]} |\tilde{Z}_1(t) - \tilde{Z}_2(t)|\right] &\leq C\mathbb{E}|\tilde{Z}_1(0) - \tilde{Z}_2(0)| \exp(Ct) \\ &= C\mathbb{E}|\tilde{Z}_1(0) - \tilde{Z}_2(0)|. \end{aligned} \tag{6.28}$$

In particular, taking $t = 1$ provides a useful bound on the distance between the outputs of Algorithm 27 in terms of its inputs; or equivalently, a bound on the distance between samples from $\tilde{M}_n^\ell((Y_{n-1,1}^\ell, Y_{n-1,2}^\ell), \cdot)$ in terms of $||X_{n-1,1}^\ell - X_{n-1,2}^\ell||$.

We need, however, a bound in terms of $||(X_{n-1,1}^\ell - X_{n-1,2}^\ell|| \wedge 1)$, for which Table 6.1 provides a convergence rate. In order to obtain this, we slightly modify [Rogers and Williams, 2000, Corollary V.11.7]. In particular, under the assumptions of that Corol-

lary, and using its notation, we see that

$$\tilde{X}_t - \tilde{Y}_t = \xi - \eta + \int_0^t (b(s,X) - b(s,Y))ds + \int_0^t (\sigma(s,X) - \sigma(s,Y))dB_s.$$

Therefore

$$(\sup_s ||\tilde{X}_s - \tilde{Y}_s|| \wedge 1) \leq (||\xi - \eta|| \wedge 1) + \int_0^t (|b(\tilde{X}_s) - b(\tilde{Y}_s)|ds + \sup_s |\int_0^s \sigma(\tilde{X}_u) - \sigma(\tilde{Y}_u)dB_u|$$

and following the same calculations as [Rogers and Williams, 2000, Lemma V.11.5], we obtain.

$$\mathbb{E}\sup_s(||\tilde{X}_s - \tilde{Y}_s||^p \wedge 1) \leq C\{\mathbb{E}(||\xi - \eta||^p \wedge 1) + \mathbb{E}\int_0^t (|b(\tilde{X}_s) - b(\tilde{Y}_s)|^p + |\sigma(\tilde{X}_s) - \sigma(\tilde{Y}_s)|^p ds\}.$$

Finally, assuming that b, σ are bounded and Lipschitz it follows that

$$\mathbb{E}\sup_s(||\tilde{X}_s - \tilde{Y}_s||^p \wedge 1) \leq C\{\mathbb{E}(||\xi - \eta||^p \wedge 1) + \int_0^t (||\tilde{X}_s - \tilde{Y}_s||^p \wedge 1)ds + \int_0^t (||\tilde{X}_s - \tilde{Y}_s||^p \wedge 1)ds\}$$

and so as for (6.28), the Grönwall inequality implies that

$$\mathbb{E}\sup_s(||\tilde{X}_s - \tilde{Y}_s||^p \wedge 1) \leq C\{\mathbb{E}(||\xi - \eta||^p \wedge 1) \tag{6.29}$$

Returning to the expression (6.27), we again divide into four cases:

1. $j = 1$:
$$\int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\, \tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)) = 0.$$

2. $j = 2, 3, 4, 5$:
$$\begin{aligned}\int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\tilde{M}^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)) &\leq C\bar{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), \mathcal{B}_j) \\ &\leq C\mathbb{P}(|V(X_{n,1}^\ell) - V(X_{n,2}^\ell)| > 2\delta_\ell | X_{n-1,1}^{\ell,i}, X_{n-1,2}^{\ell,i}) \\ &\leq C\frac{\mathbb{E}\left[(||X_{n,1}^{\ell,i} - X_{n,2}^{\ell,i}|| \wedge 1)|X_{n-1,1}^{\ell,i}, X_{n-1,2}^{\ell,i}\right]}{\delta_\ell} \\ &\leq C\frac{\mathbb{E}\left[(||X_{n-1,1}^{\ell,i} - X_{n-1,2}^{\ell,i}|| \wedge 1)\right]}{\delta_\ell}\end{aligned} \tag{6.30}$$

3. $j = 6, 7$:
$$\begin{aligned}\int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\, \tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)) &\leq C\tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,1}^{\ell,i}), \mathcal{B}_j) \\ &\leq C\mathbb{P}(V(X_{n-1,1}^{\ell,i}) \in (a - \delta_\ell, a)) \\ &\leq C\sup|q^n(\cdot)|\delta_\ell\end{aligned} \tag{6.31}$$

4. $j = 8$:

$$\int_{\mathcal{B}_j} (S_n^a(y_1) - S_n^a(y_2))\, \tilde{M}_n^\ell((Y_{n-1,1}^{\ell,i}, Y_{n-1,2}^{\ell,i}), d(y_1, y_2)) \leq ||(X_{n-1,1}^{\ell,i} - X_{n-1,2}^{\ell,i}) \wedge 1||, \tag{6.32}$$

by [Jasra et al., 2017, Assumption 4.1], since $S_n^{(a)}$ is Lipschitz continuous over this region.

Taking again the optimal rate $\delta_\ell = h_\ell^{1/4}$, putting all this together gives

$$\mathbb{E}|(6.18)|^2 = \mathcal{O}(h_\ell^{1/4}).$$

Now combining the three bounds for (6.17), (6.18) and (6.19) gives

$$\mathbb{E}|\bar{\Phi}(\bar{\eta}_{n-1}^{\ell,N})_1(S_n^{(a)}) - \bar{\Phi}(\bar{\eta}_{n-1}^{\ell,N})_2(S_n^{(a)})|^2 = \mathcal{O}(h_\ell^{1/4})$$

and hence we have that (6.10) $= \mathcal{O}(h_\ell^{1/4})$.

It remains to deal with (6.11). Again following [Jasra et al., 2017, Appendix C], this may be broken up into six terms. The analysis of five of these terms (those labelled terms 2, 3, 4, 5, 6) is identical to that in the appendix, and the bounds given there in terms of $||\eta_{n,1}^\ell - \eta_{n,2}^\ell||_{\text{TV}}$, $|||M_{n,1}^\ell - M_{n,2}^\ell|||$ and $B_\ell(n)$ are sufficient for the present result.

It remains only to deal with term 1. Following the expansion in the appendix, we have that

$$\begin{aligned} C\mathbb{E}[(\eta_{n-1,1}^{\ell,N}(G_{n-1}M_{n,1}^\ell(S_n^{(a)}) - G_{n-1}M_{n,2}^\ell(S_n^{(a)})) & \\ - \eta_{n-1,1}^\ell(G_{n-1}M_{n,1}^\ell(S_n^{(a)}) - G_{n-1}M_{n,2}^\ell(S_n^{(a)}))^2] & \\ \leq \frac{C}{N} \sup ||G_{n-1}M_{n,1}^\ell(S_n^{(a)}) - G_{n-1}M_{n,2}^\ell(S_n^{(a)})||^2 & \\ \leq \frac{C}{N} \sup ||M_{n,1}^\ell(S_n^{(a)}) - M_{n,2}^\ell(S_n^{(a)})||^2 & \end{aligned}$$

which may be bounded exactly as in the analysis for 6.17. (In the above the second line follows from Proposition 4, and the third line since G_{n-1} is bounded). The other part of the expansion of Term 1, namely

$$\mathbb{E}\left[\left([\bar{\eta}_{n-1}^{\ell,N} - \bar{\eta}_{n-1}^\ell]([G_{n-1}M_{n,2}^\ell(S_n^{(a)})]_1 - [G_{n-1}M_{n,2}^\ell(S_n^{(a)})]_2)\right)\right],$$

is controlled by the inductive argument of [Jasra et al., 2017, Theorem C.4] since $G_{n-1}M_{n,2}^\ell(S_n^{(a)})$ is Lipschitz-continuous.

6.3 Computational cost

We note that with the rate given in Theorem 17, ensuring Algorithm 25 has finite variance forces it to require an infinite expected computing cost, and vice-versa. The cost of one sample from the coupled Euler scheme of Algorithm 22 at discretisation level $h_\ell = 2^{-\ell}$ is 2^ℓ. Therefore, having chosen the distribution P in Algorithm 25, the expected cost of the algorithm is

$$\mathbb{E}[\text{Cost}] = \sum_{l=0}^{\infty} 2^\ell P(\ell).$$

On the other hand, the variance of the estimator is given by

$$\text{Var}[\hat{p}^{\text{ub}}] \leq C \sum_{\ell=0}^{\infty} \frac{1}{P(\ell)} h_\ell^\beta < \infty,$$

and it is clearly not possible to ensure these are simultaneously finite when $\beta \leq 1$.

This may seem discouraging, but several possibilities remain open. One is to use various methods to try to increase the fixed-level variance rate, as we discuss in section 6.6. A second idea due to Jasra et al. [2020], where the de-biased algorithm similarly cannot achieve finite variance and cost simultaneously, is a truncation technique in which one specifies a maximum acceptable level $L^{\max}$, and samples L from some alternative distribution P^{trunc} over $(0, 1, \ldots, L)$. Although this introduces bias into the algorithm, it retains certain attractive features: in particular, one can in principle achieve a superior mean-squared error to the corresponding multilevel SMC algorithm for a given cost using parallel computation (see the discussion in [Jasra et al., 2020, Section 4]).

6.4 Numerical study: the Ornstein-Uhlenbeck process

The Ornstein-Uhlenbeck process (OU process) with mean 0 is described by the following SDE with mean-reverting drift:

$$dX(t) = -\theta X(t)dt + \sigma dB(t). \tag{6.33}$$

It has stationary distribution $\mathcal{N}(0, \frac{\sigma^2}{\theta})$; moreover, given X_0 one has the explicit formula (see [Protter, 2004, Theorem 42])

$$X(t) = X(0)e^{-\theta t} + \sigma e^{-\theta t} W \sqrt{\frac{1}{2\theta}(e^{2\theta t} - 1)}$$

where $W \sim \mathcal{N}(0, 1)$. Therefore, if we set $X_0 = 0$, it follows that

$$X_t \sim \mathcal{N}\left(0, \frac{\sigma^2}{2\theta}(1 - e^{-2\theta t})\right). \tag{6.34}$$

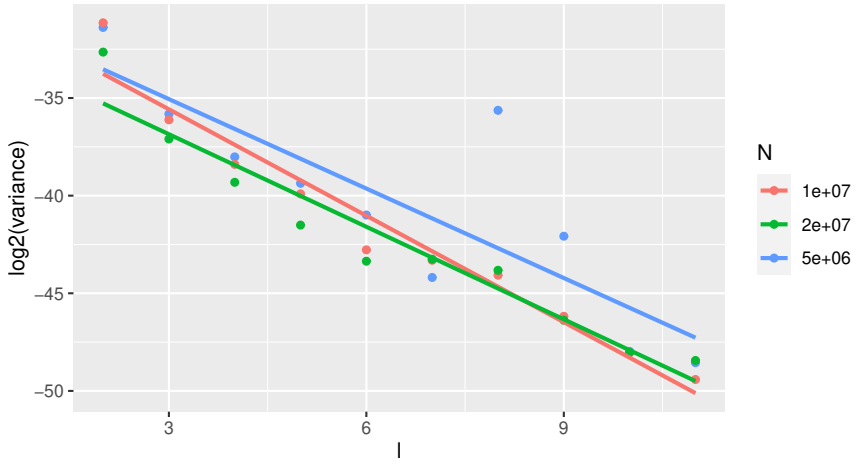


Figure 6.2: Plot of ℓ against the log of the empirical variance of Ξ_ℓ.

Choosing parameters $\theta = 1, \mu = 0, \sigma = 3$, we consider the problem of estimating the event $X_t > a$ for suitable a. For this model example, we consider a modest rare event regime by taking $t = 10$ and $a = 9$. The choice $t = 10$ makes the exact distribution (6.34) to all purposes indistinguishable from the stationary distribution. Either way, it can be checked that $P(X_{10} > 9) \approx 1.105 \times 10^{-5}$.

First, we are interested in numerically verifying Theorem 17. We produce 100 samples of the fixed level estimator Ξ_ℓ, each with $N = 5000000, 10000000, 20000000$ particles, for levels $l = 2, \ldots, 11$. In Figure 6.2, we plot $\log_2$ of the empirical variance of $\Xi_\ell = (\gamma_{10,1}^{\ell,N} - \gamma_{10,2}^{\ell,N})(S_{10}^{(9)})$ over the 1000 samples against ℓ. The gradient ought to give us an estimate of β, supposing that $V[\Xi_\ell] = \mathcal{O}(h_\ell^\beta)$ as Theorem 17 suggests. In fact, although there are a number of outlying points, the gradients look rather optimistic compared to the bound of the theorem.

We also try the truncation method mentioned in section 6.3, and comparing it against an implementation as a multilevel SMC algorithm, following Jasra et al. [2017]. For the multilevel algorithm at level L, we sample independently for each level $\ell = 1, \ldots, L$ the estimator $\Xi_\ell^{N_\ell} \sim$Algorithm.24, and $\Xi_0^{N_0} \sim$Algorithm.21, using N_ℓ particles. We choose the N_ℓ following the guidance in [Jasra et al., 2017, Section 5]. In particular, we set $N_{0,L} = 100 \times L 2^{2L}$, and then $N_\ell = N_{0,L}(h^\ell)^{-3/4}$ for $\ell = 0, \ldots, L$. We then have the ML-SMC estimator

$$\hat{p}_L^{\text{ML-SMC}} = \sum_{l=0}^{L} \Xi_\ell^{N_\ell}$$

Trying 100 samples of $\hat{p}_L^{\text{ML-SMC}}$ with successive values of L in turn, we achieve an MSE of 4.17×10^{-11} at $L = 6$. This MSE is minimal to be sufficient for a rare event of magnitude 10^{-5}, since one wants the root mean-square error small compared to the target probability. The squared bias at this level was found to be 2.63×10^{-12} (so the MSE is somewhat dominated by the variance term and hence not optimally tuned).

The cost to achieve this MSE was then

$$\sum_{\ell=0}^{6}\sum_{i=1}^{N_\ell} 2^\ell = 100 \times 6 \times 2^{2\times 6}\sum_{\ell=0}^{6} 2^{\ell/4} \approx 6.14 \times 10^7.$$

We can compare this against the result of truncating the de-biased algorithm (Algorithm 25), which we term the *truncated-debiased estimator*. We choose the maximum level L to be $L^{\max} = 7$, and following Jasra et al. [2017] take $P(\ell) \propto 2^{-3\ell/2}$ for $\ell = 0, \ldots, 7$, and $P(\ell) = 0$ otherwise. We then generated 75000 samples Ξ_{L_i} with $L_i \sim P$ using 20000 particles each (following the steps of Algorithm 25), and grouped these 75000 samples into subgroups of size M for increasingly large M until the estimators

$$\hat{p}^{\text{T-DB}} = \frac{1}{M}\sum_{i=1}^{M}\frac{\Xi_{L_i}}{P(L_i)}$$

achieved a variance of size comparable to the variance of the ML-SMC algorithm. In this case, the choice $M = 2500$ achieved the MSE of approximately 5.85×10^{-11}. The squared bias turned out to again be $\approx 5.43 \times 10^{-12}$, again an order of magnitude smaller than the variance. The cost in the case can be calculated to be

$$2500 \times 20000 \times \sum_{\ell=0}^{7} 2^\ell 2^{-3\ell/2} \approx 1.60 \times 10^8.$$

This gives a cost ratio of approximately 2.6 between the two algorithms, which compares favourably to the cost ratios of the Multilevel Particle Filter to the truncated-debiased algorithm given in [Jasra et al., 2020, Section 4.3], which range from 2-7.

We summarise all this information in Table 6.3 below.

Algorithm	MSE	Bias^2	Cost
ML-SMC	4.17×10^{-11}	2.63×10^{-12}	6.14×10^7
Truncated-debiased	5.85×10^{-11}	5.43×10^{-12}	1.60×10^8

Table 6.3: Summary of results for estimating the rare event $p \approx 10^{-5}$ with the multilevel-SMC algorithm and the truncated-debiased algorithm.

6.5 De-biasing for Multilevel Splitting

We first considered the de-biasing approach as an alternative to ε-strong simulation for Multilevel Splitting. The numerical results were poor, which we conjecture was due to the sequence of $0-1$ potentials, together with the random trajectory lengths between barrier crossing times.

Moreover, guiding the numerical simulations using an analysis of the type carried out in section 6.2 seemed more difficult in the splitting setting, introducing more violations of the standard Lipschitz assumptions, and also of the boundedness of the potentials away from 0.

However, we record the implementation here in the sequence of Algorithms 28, 29, 30 and 31, as future work may lead to better results in this direction.

Algorithm 28 Discretised MLS

Given λ together with G_i, M_i^Δ for $i = 1, \ldots, m$ and a step-size Δ, and a fixed number of particles N

1. For each $j = 1, \ldots, N$, draw independently $X_0^j \sim \lambda(\cdot)$ and
$$U_1^j \sim M_1^\Delta\left(\left(0, X_0^j\right), \cdot\right).$$
2. Record $N_1 = \sum_{j=1}^N G_1(U_1^j)$.
3. For $i = 2, \ldots, m$:
   (a) If $N_{i-1} = 0$, return $\hat{p} = 0$.
   (b) For $j = 1, \ldots, N$ sample independently $U_i^j \sim \frac{1}{N}\sum_{k=1}^N G_{i-1}(U_{i-1}^k) M_i^\Delta(U_{i-1}^k, \cdot)$.
   (c) Record $N_i = \sum_{j=1}^N G_i(U_i^j)$.
4. Return
$$\hat{p} = \prod_{i=1}^m \left(\frac{N_i}{N}\right).$$

6.6 Discussion

We have described an algorithm for the unbiased estimation of rare event probabilities for continuous time processes, using an MLMC-inspired debiasing technique, and have described and illustrated a conservative bound for its fixed-level estimators.

We have also discussed the computational difficulties of implementing this and similar de-biased algorithms directly, and explored an alternative implementation idea following Jasra et al. [2020], to which we produced comparable results in a simple setting.

It is possible that improvements can be made to our algorithm by using more sophisticated coupling schemes for discretisation of SDEs such as the Milstein method or another scheme with superior convergence rates to the Euler-Maruyama scheme. More promising is the option of looking at alternative resampling schemes such as the transport-based methods discussed in Jacob et al. [2016]; Ballesio et al. [2020], since we have considered only a fairly simple example of such.

Algorithm 29 Coupled Level Crossing

Given $X_1, X_2, \Delta, z_A, z_B, \xi$, and a sequence of independent $\mathcal{N}(0,1)$ random variables $(Z_1, Z_2, \ldots)$:

1. Initialise $n \leftarrow 0$, $t_1 \leftarrow 0$. While $z_A \leq \xi(X_1) \leq z_B$:
   (a) $n \leftarrow n+1$, $t_1 \leftarrow t_1 + \Delta$ and update
   $$X_1 \leftarrow X_1 + a(X_1)\Delta + b(X_1)\sqrt{\Delta}Z_n.$$
   (b) If n even, update
   $$X_2 \leftarrow X_2 + a(X_2)\,(2\Delta) + b(X_2)\sqrt{\Delta}(Z_{n-1} + Z_n).$$
2. Set $g_1 = 1_{\xi(X_1) > z_B}$. Initialise $t_2 \leftarrow t_1$.
3. If $z_A \leq \xi(X_2) \leq z_B$ and n odd, update $t_2 \leftarrow t_2 + \Delta$ and
   $$X_2 \leftarrow X_2 + a(X_2)\,(2\Delta) + b(X_2)\sqrt{\Delta}(Z_n + Z_{n+1}).$$
4. Initialise $n' \leftarrow n+1$. While $z_A \leq \xi(X_2) \leq z_B$:
   (a) $n' \leftarrow n'+1$, $t_2 \leftarrow t_2 + 2\Delta$, and update
   $$X_2 \leftarrow X_2 + a(X_2)(2\Delta) + b(X_2)\sqrt{2\Delta}Z_n.$$
5. Set $g_2 = 1_{\xi(X_2) > z_B}$, and return $(t_1, t_2, X_1, X_2, g_1, g_2)$.

Algorithm 30 Coupled Multilevel Splitting

Given a fixed number of particles N, ξ, initial distribution λ, level markers $z_A, z_1, \ldots, z_m$, $L \in \mathbb{N}$, discretisation level Δ_L and distribution $P(\cdot)$:

1. For each $j = 1, \ldots, N$, draw independently $X_0^{(1,j)}, X_0^{(2,j)} \sim \lambda(\cdot)$ and

$$(t_1^{(1,j)}, t_1^{(2,j)}, X_1^{(1,j)}, X_1^{(2,j)}, g_1^{(1,j)}, g_1^{(2,j)}) \sim \text{Algorithm.29}(X_0^{(1,j)}, X_0^{(2,j)}, \Delta, z_A, z_1, \xi).$$

2. Record $N_1^1 = \sum_{j=1}^N g_1^{(1,j)}$, $N_1^2 = \sum_{j=1}^N g_1^{(2,j)}$. Let

$$g_1^1 = (g_1^{(1,1)}, \ldots, g_1^{(1,N)}), g_1^2 = (g_1^{(2,1)}, \ldots, g_1^{(2,N)}),$$

3. For $i = 2, \ldots, m$:
   (a) If $N_{i-1}^1 = 0$ or $N_{i-1}^2 = 0$, return $\hat{p} = 0$.
   (b) Else, for $j = 1, \ldots, N$:
      i. Sample coupled indices $(I_1^j, I_2^j) \sim$ Algorithm.23(g_{i-1}^1, g_{i-1}^2), and set

$$\tilde{X}_{i-1}^{(1,j)} = X_{i-1}^{(1,I_1^j)}, \tilde{X}_{i-1}^{(2,j)} = X_{i-1}^{(2,I_2^j)}$$

      ii. Sample

$$(t_i^{(1,j)}, t_i^{(2,j)}, X_i^{(1,j)}, X_i^{(2,j)}, g_i^{(1,j)}, g_i^{(2,j)}) \sim \text{Algorithm.29}(\tilde{X}_{i-1}^{(1,j)}, \tilde{X}_{i-1}^{(2,j)}, \Delta, z_A, z_i, \xi)$$

   (c) Record $N_i^1 = \sum_{j=1}^N g_i^{(1,j)}$, $N_1^2 = \sum_{j=1}^N g_i^{(2,j)}$. Let

$$g_i^1 = (g_i^{(1,1)}, \ldots, g_i^{(1,N)}), g_i^2 = (g_i^{(2,1)}, \ldots, g_i^{(2,N)}),$$

4. Return

$$\hat{p} = \frac{1}{P(L)} \left(\prod_{i=1}^m \left(\frac{N_i^1}{N} \right) - \prod_{i=1}^m \left(\frac{N_i^2}{N} \right) \right).$$

Algorithm 31 Unbiased multilevel splitting

Given $P(\cdot)$ supported on $\mathbb{N}$:

1. Draw $L \sim P(\cdot)$, and set $\Delta = 2^{-L}$.
   (a) If $L = 0$, return

$$\hat{p} \sim \text{Algorithm.28}(\ldots)$$

   (b) Else if $L > 0$, return

$$\hat{p} \sim \text{Algorithm.30}(\ldots)$$

Chapter 7

Conclusion

In this thesis, we have examined some possibilities for combining algorithms for efficient rare event estimation in continuous time with exact and unbiased simulation techniques. In Chapter 4, we identified and demonstrated new and practical algorithms for exact splitting in one dimension and multiple dimensions, suitable for events of genuinely small magnitude. We showed that these modifications to traditional methods remain unbiased, while being fully implementable without the use of numerical discretisation. We noted that these algorithms are quite flexible in implementation, for example in the time intervals over which one carries out the ε-strong sampling, in the prioritising of ε-balls for refinement, and in the choice of splitting location. We feel there is real scope for further work in this direction. For example, it may be possible to systematically exploit the geometry of certain reaction co-ordinates and level sets, and likewise of alternative ε-strong methods which continue to be the object of active research. The tricky problem of guaranteeing (expected) finite running-times for these algorithms might also be an interesting question to explore.

In chapters 5 and 6, we specified a new de-biased algorithm in a large-deviations inspired rare events setting, and made preliminary investigations of its properties. Although the the prospects for finite cost unbiased algorithms of this particular type do not seem good, there are many interesting prospects in the general direction of this research. Seeking different settings where finite cost and finite variance of the de-biased algorithms might genuinely be achieved is one possibility. More broadly, improving multilevel Sequential Monte Carlo algorithms by investigating alternative coupled discretisation schemes and coupled resampling schemes to obtain better strong rates of decay for the variance, leading to better MSE/cost trade-offs, may be a promising idea for the future.

Bibliography

M. Amrein and H. Künsch. A Variant of Importance Splitting for Rare Event Estimation: Fixed Number of Successes. *ACM Transactions on Modelling and Computer Simulation*, 21(2):Article 13, 20pp., 2011.

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697 – 725, 2009. doi: 10.1214/07-AOS574.

R. Avikainen. Convergence rates for approximations of functionals of SDEs. *arXiv preprint arXiv:0712.3635*, 2007.

M. Ballesio, A. Jasra, E. von Schwerin, and R. Tempone. A Wasserstein Coupled Particle Filter for Multilevel Estimation, 2020.

V. Bally and D. Talay. The law of the euler scheme for stochastic differential equations: I. convergence rate of the distribution function. *Probability Theory and Related Fields*, 104, 07 1994. doi: 10.1007/BF01303802.

A. Beskos and G.O. Roberts. 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(6):1077–1098, 2006.

A. Beskos, S. Peluchetti, and G.O. Roberts. ε-Strong simulation of the Brownian path. *Bernoulli*, 18(4):1223–1248, 2012.

A. Beskos, A. Jasra, K. Law, R. Tempone, and Y. Zhou. Multilevel Sequential Monte Carlo samplers. *Stochastic Processes and their Applications*, 127(5):1417–1440, 2017.

J. Blanchet, X. Chen, and J. Dong. ε-Strong simulation for multidimensional stochastic differential equations via rough path analysis. *Annals of Applied Probability*, 27(1): 275–336, 2017.

J. Blanchet, P. W. Glynn, and Y. Pei. Unbiased Multilevel Monte Carlo: Stochastic optimization, steady-state simulation, quantiles, and other applications. *arXiv preprint arXiv:1904.09929*, 2019.

C-E. Bréhier, M. Gazeau, L. Goudenège, T. Lelièvre, and M. Rousset. Unbiasedness of some generalized adaptive multilevel splitting algorithms. *Annals of Applied Probability*, 26(6):3559–3601, 2016.

J. C. Burkill and H. Burkill. *A Second Course in Mathematical Analysis*. Cambridge University Press, 1970.

F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007.

F. Cérou, P. Del Moral, F. Le Gland, and P. Lezaud. Genetic genealogical models in rare event analysis. *ALEA: Latin American Journal of Probability and Mathematical Statistics*, 1:181–203, 2006.

F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and computing*, 22(3):795–808, 2012.

N. Chen and Z. Huang. Localization and exact simulation of Brownian motion-driven stochastic differential equations. *Mathematics of Operations Research*, 38(3):591–616, 2013. doi: 10.1287/moor.2013.0585.

Y. Chen, J. Dong, and H. Ni. ε-strong simulation of fractional Brownian Motion and related stochastic differential equations. Mathematics e-print 1902.08824, ArXiv, 2019.

N. Chopin and O. Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

A. M. Davie. Differential equations driven by rough paths: an approach via discrete approximation. *Applied Mathematics Research eXpress*, 2008, 2008. ISSN 1687-1200. doi: 10.1093/amrx/abm009.

M. Deaconu and S. Herrmann. Strong approximation of Bessel processes. *arXiv preprint*, 2021.

P. Del Moral. *Feynman-Kac Formulae*. Springer Verlag, New York, 2004.

P. Del Moral. *Mean field simulation for Monte Carlo integration*. CRC press, 2013.

P. Del Moral and J. Garnier. Genealogical particle analysis of rare events. *Annals of Applied Probability*, 15(4):2496–2534, 2005.

P. Del Moral and P. Lezaud. Branching and interacting particle interpretations of rare event probabilities. In *Stochastic Hybrid Systems*, pages 277–323. Springer, 2006.

P. Del Moral, A. Jasra, A. Lee, C. Yau, and X. Zhang. The Alive Particle Filter and its Use in Particle Markov Chain Monte Carlo. *Stochastic Analysis and Applications*, 33(6):943–974, 2015. doi: 10.1080/07362994.2015.1060892.

L. Devroye. The series method for random variate generation and its application to the Kolmogorov-Smirnov distribution. *American Journal of Mathematical and Management Sciences*, 1(4):359–379, 1981. doi: 10.1080/01966324.1981.10737080.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Oxford Handbook of Nonlinear Filtering*, 12, 2011.

P. Fearnhead, O. Papaspiliopoulos, and G.O. Roberts. Particle filters for partially-observed diffusion. *Journal of the Royal Statistical Society, Series B: Methodology*, 70(4):755–777, 2008.

P. Friz and M. Hairer. *A course on rough paths*. Springer, 2014.

P. Friz and N. Victoir. *Multidimensional stochastic processes as rough paths: theory and applications*, volume 120. Cambridge University Press, 2010.

M. J. J. Garvels. *The splitting method in rare event simulation*. PhD thesis, University of Twente, Twente, 2000.

M. Gerber, N. Chopin, and N. Whiteley. Negative association, ordering and convergence of resampling methods. *Annals of Statistics*, 47(4):2236–2260, 2019. doi: 10.1214/18-AOS1746.

M. Giles, D. Higham, and X. Mao. Analysing multi-level Monte Carlo for options with non-global Lipschitz payoff. *Finance and Stochastics*, 13:403–413, 09 2009. doi: 10.1007/s00780-009-0092-1.

M. B. Giles. Multilevel Monte Carlo path simulation. *Operations research*, 56(3): 607–617, 2008.

M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999.

P. W. Glynn and C-H. Rhee. Exact estimation for Markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389, 2014.

J. I. González Cázares, A. Mijatović, and G. Uribe Bravo. ε-strong simulation of the convex minorants of stable processes and meanders. *Electronic Journal of Probability*, 25, Jan 2020. ISSN 1083-6489. doi: 10.1214/20-ejp503.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F-radar and signal processing*, volume 140, pages 107–113. IET, 1993.

P. Grassberger. Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1000000. *Physical Review E*, 56(3):3682–3693, 1997.

S. Heinrich. Multilevel Monte Carlo Methods. In *International Conference on Large-Scale Scientific Computing*, pages 58–67. Springer, 2001.

J. Heng, A. Jasra, K. J. H. Law, and A. Tarakanov. On unbiased estimation for discretized models. *arXiv preprint*, 2021.

P. E. Jacob and A. H. Thiery. On nonnegative unbiased estimators. *The Annals of Statistics*, 43(2):769–784, 2015. ISSN 00905364.

P. E. Jacob, F. Lindsten, and T. B. Schön. Coupling of particle filters. *arXiv preprint arXiv:1606.01156*, 2016.

A. Jasra, K. Kamatani, K. J. H. Law, and Y. Zhou. Multilevel particle filters. *SIAM Journal on Numerical Analysis*, 55(6):3068–3096, 2017.

A. Jasra, K. Kamatani, P. P. Osei, and Y. Zhou. Multilevel particle filters: normalizing constant estimation. *Statistics and Computing*, 28:47–60, 2018.

A. Jasra, K. J. H. Law, and D. Lu. Unbiased estimation of the gradient of the log-likelihood in inverse problems. *arXiv preprint*, 2020a.

A. Jasra, K. J. H. Law, and F. Yu. Unbiased filtering of a class of partially observed diffusions. *arXiv preprint*, 2020b.

A. Jasra, F. Yu, and J. Heng. Multilevel particle filters for the non-linear filtering problem in continuous time. *Statistics and Computing*, 30(5):1381–1402, 2020.

A.M. Johansen, P. Del Moral, and A. Doucet. Sequential Monte Carlo samplers for rare events. In *Proceedings of the 6th International Workshop on Rare Event Simulation*, pages 256–267, Bamberg, Germany, October 2006.

H. Kahn and T. E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematics Series*, 12:27–30, 1951.

P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 2013.

S. Kockara, T. Halic, K. Iqbal, C. Bayrak, and R. Rowe. Collision detection: a survey. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 4046–4051, 2007. doi: 10.1109/ICSMC.2007.4414258.

A. Lagnoux-Renaudie. Rare event simulation. *Probability in the Engineering and the Informational Sciences*, 20:45–66, 2006.

A. Lagnoux-Renaudie. Effective branching splitting method under cost constraint. *Stochastic processes and their applications*, 118(10):1820–1851, 2008.

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.

P. L'Ecuyer, V. Demers, and B. Tuffin. Rare events, splitting, and quasi-Monte Carlo. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 17(2):9–es, 2007.

Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Rare events, splitting, and quasi-monte carlo. *ACM Trans. Model. Comput. Simul.*, 17(2):9–es, apr 2007. ISSN 1049-3301.

T. Lyons. On the Non-Existence of Path Integrals. *Proceedings: Mathematical and Physical Sciences*, 432(1885):281–290, 1991.

D. McLeish. A general method for debiasing a Monte Carlo estimator. *Monte Carlo Methods and Applications*, 17(4):301–315, 2011.

M. Mider, P.A. Jenkins, M. Pollock, G.O. Roberts, and M. Sørensen. Simulating bridges using confluent diffusions. Mathematics e-print 1903.10184, ArXiv, 2019.

L. M. Murray, S. S. Singh, and A. Lee. Anytime monte carlo. *Data-Centric Engineering*, 2, 2021.

P. Mörters and Y. Peres. *Brownian Motion*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

B. Øksendal. *Stochastic differential equations: an introduction with applications*. Springer, 1992.

B. Paige, F. Wood, A. Doucet, and Y. W. Teh. Asynchronous anytime sequential monte carlo. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

M. Pollock. *Some Monte Carlo methods for jump diffusions*. PhD thesis, University of Warwick, 2013.

M. Pollock, A.M. Johansen, and G.O. Roberts. On the exact and ε-strong simulation of (jump) diffusions. *Bernoulli*, 22(2):794–856, 2016.

Klaus Pötzelberger and Liqun Wang. Boundary crossing probability for Brownian motion. *Journal of Applied Probability*, 38(1):152–164, 2001.

P. E. Protter. *Stochastic Integration and Differential Equations*. Springer-Verlag, 2004.

C-H. Rhee and P. W. Glynn. A new approach to unbiased estimation for SDE's. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–7. IEEE, 2012.

C-H. Rhee and P. W. Glynn. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.

L. C. G. Rogers and D. Williams. *Stochastic Differential Equations and Diffusions*, volume 2 of *Cambridge Mathematical Library*, chapter 5. Cambridge University Press, 2 edition, 2000.

G. Rubino and B. Tuffin, editors. *Rare event simulation using Monte Carlo methods*, volume 73. Wiley Online Library, 2009.

E. Thönnes. Perfect simulation of some point processes for the impatient user. *Advances in Applied Probability*, 31(1):69–87, 1999. ISSN 00018678.

M. Vihola. Unbiased estimators and Multilevel Monte Carlo. *Operations Research*, 66 (2):448–462, 2018.

M. Villén-Altamirano and J. Villén-Altamirano. RESTART: a straightforward method for fast simulation of rare events. In *Proceedings of Winter Simulation Conference*, pages 282–289. IEEE, 1994.

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

L. C. Young. An inequality of the Hölder type, connected with Stieltjes integration. *Acta Mathematica*, 67:251 – 282, 1936. doi: 10.1007/BF02401743.