Importance sampling algorithms for rare event simulation of jump-diffusions based on viscosity solutions of Hamilton-Jacobi equations

Alvaro Guillen Cuevas

A Thesis

for The Department of

Mathematics and Statistics

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Science (Mathematics) at

Concordia University

Montreal, Quebec, Canada

August 2017

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Alvaro Guillen Cuevas**

Entitled: **Importance sampling algorithms for rare event simulation of jump-diffusions based on viscosity solutions of Hamilton-Jacobi equations**

and submitted in partial fulfilment of the requirements for the degree of

Master of Science (Mathematics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Examiner

Dr. Melina Mailhot

_____ Examiner

Dr. Xiaowen Zhou

_____ Thesis Supervisor

Dr. Lea Popovic

Approved by _____

Chair of Department or Graduate Program Director

_____

Dean of Faculty

Date _____

Abstract

Importance sampling algorithms for rare event simulation of jump-diffusions based on viscosity solutions of Hamilton-Jacobi equations

by Alvaro Guillen Cuevas

Consider a marked point process or a jump-diffusion, from which we want to simulate a trajectory of the process or a functional of it. In both cases the magnitude of noise contributors is controlled by a small parameter ε. Raw Monte Carlo methods produce estimators with a large relative error, which increases even more as N increases or ε decreases. Using viscosity sub-solutions of Hamilton-Jacobi equations, we were able to produce importance sampling algorithms with optimal asymptotic behaviour and low relative error across a variety of small values of noise contribution. Some basic stochastic knowledge and means to produce the discretization of a trajectory of the jump-diffusion are needed, both of which are provided in this text. Furthermore, we applied the algorithm we developed to model the bistability in the concentration of certain molecular species.

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor Doctor Lea Popovic for proposing the research topic. Furthermore, her guidance and patience made the preparation of this thesis possible.

I would like to thank as well both members of my examination committee, Doctor Melina Mailhot and Doctor Xiaowen Zhou. Their feedback and suggestions are really appreciated.

I also would like to express my deepest gratitude for all the guidance provided by my professors in Concordia University. Moreover, I would like to take this opportunity as well for thanking the staff at the Department of Mathematics and Statistics at Concordia. Needless to say that I could not have continued pursuing my dreams without Concordia's generous financial support.

My deepest gratitude and love for Diana for her unconditional support and words of encouragement. Moreover, I would like to thank my mother as well for always being there with a cheerful attitude. Finally, I would like to express my appreciation to my friends in Montreal for being so supportive.

Contents

List of Figures

List of Tables

Introduction

Stochastic modelling has come a long way since Robert Brown noted through his microscope an apparently random motion in pollen particles in 1827 (see (Brown, 1828)). Little did he know that many years later his last name would be used to refer the random motion of a wide variety of particles. Brownian Motion was just the beginning. Robert Merton proposed in 1969 an optimal allocation of a budget among several assets, which would maximize the expected utility of the investment portfolio (see (Merton, 1969)). Later, Merton supervised the work of Myron Scholes in collaboration with Fischer Black to yield a pioneering formula to valuate stock options (see (Black and Scholes, 1973)).

Needless to say, there is a broad set of topics which can be analyzed with a probabilistic approach, like systems biology (see (Wilkinson, 2011)) or climate modeling (see (Franzke et al., 2015)). However, quantitative finance is one of the most popular topics which can be addressed by the application of stochastic calculus. After all, it might be vastly profitable to know the value of an unknown asset in a given time horizon. The way stock prices can be described by stochastic differential equations has been studied broadly, usually under the assumption that the object being modelled is a continuous function of time. Nevertheless, this assumption is not always reasonable.

According to (Tankov and Voltchkova, 2009) and additionally to empirical evidence of jumps in observed stock prices (see section 1.2 in (Tankov, 2003)) there are several reasons to prefer jump-diffusion models to their absolutely continuous counterpart, at least in the academic finance context. Firstly, jump-diffusion models do not neglect the probability that the stock price changes by a large amount over a short period of time, something that models with absolutely continuous paths do. Even though the probability of such event is small, it is of great importance to risk management. Secondly, from the point of view of hedging, continuous models of stock prices with a couple of additional instruments yield a complete market, i.e., a position on an instrument in this

market can be replicated using a linear combination of the remaining products. This makes the existence of derivative products redundant. Having jumps in the stock price trajectory produces non-completeness in the market users to replicate any position of assets with derivative products.

A closer study of stochastic differential equation allows us to simulate the trajectory in a fixed time horizon. We might be interested in the value of the process at that time or a function of it. Its drift and diffusion coefficients (and if it is the case of a jump-diffusion, its counting measure) are enough to propose a discretization with good levels of accuracy. It only remains to rely on traditional Monte Carlo methods which, by calculating the sample mean of a large amount of simulations, yield an unbiased estimator of the quantity of interest with acceptable properties.

Among the quantities of interest it is of special relevance to calculate probabilities of events that do not occur that frequently. The most obvious examples of this could be a colossal earthquake or a catastrophic hurricane. We could take into consideration the probability of failure of certain component of a system that is supposed to have an impeccable performance, in the medical or engineering industries perhaps. Moreover, calculating the probability of a large financial crisis might be of special applicability presently. And in a smaller case, we might be trying to price a derivative product whose value depends that it falls out of the money; having an asset with a low diffusion coefficient raises the computational difficulty of this problem since we are sampling from a small part of the sample space. In all of these cases, raw Monte Carlo methods are not a suitable choice since they yield estimators of high relative error, a non-desirable feature in these situations. Biasing our simulations for this small part of the sample space to appear more frequently and then properly re-weighting the summands is what importance sampling is about, which in this case would provide an estimator with a low relative error.

The goal in rare event simulation is to produce importance sampling algorithms with low relative error and optimal asymptotic behavior. For absolutely continuous diffusions we can consider the work of (Vanden-Eijnden and Weare, 2012) and (Djehiche et al., 2014). Both propose that the bias of the diffusion can be presented as the solution of a Hamilton-Jacobi equation. The former suggests to find the solution of such equation as the solution of a specific variational problem; in other words, the change of drift is done in such a way that the process tends towards the trajectory it is most likely to follow. The latter introduces the concept of viscosity sub-solutions to Hamilton-Jacobi equations. It is shown that if the change of drift is based in these, asymptotic optimality is

achieved. Moreover, given a birth-and-death process, we can produce a more efficient estimator by modifying its birth and death rates according to another viscosity sub-solution to a Hamilton-Jacobi equation. The goal of this thesis is to give an intuitive primer of the probabilistic tools required in rare event simulation and important sampling, then to review discretization schemes for stochastic differential equations in order to actually carry out the simulations, and finally to extend the work of (Djehiche et al., 2014) for jump-diffusions.

In Chapter 1 we give the probabilistic bases that will sustain the theoretical tools for the importance sampling algorithms. We will review simple Itô integrals with respect to a Brownian Motion and we will extend that review to integrals with respect to a counting measure, which will lead us to a review of marked point processes. These two parts are vital to understand jump-diffusions and a modification to Itô's formula for these kind of processes. Finally, we will present how Girsanov's theorem is used to propose a modification in the drift of the diffusions.

Chapter 2 represents the skeleton of our simultations. In here we will study the work of (Giesecke et al., 2015) where a discretization scheme for jump-diffusions is proposed. Briefly, the mark inter-arrival times are constructed using a transformation of Poisson inter-arrival times, and the absolutely continuous part between jumps is approximated using Euler discretization.

For the main core of this work, Chapter 3 shows an explanation of the main theoretical ideas of (Djehiche et al., 2014). First, we introduce the concept of Hamilton-Jacobi equations and their viscosity subsolutions. Then, we introduce the concept of the Mane potential, which is a fundamental part of the construction of said sub-solutions. All the concepts will be treated simultaneously for diffusions and jump-diffusions to make their similitudes evident. The chapter will culminate with the design of an importance sampling algorithm for a jump-diffusion which will be then tested using the discretization scheme reviewed in Chapter 2.

Throughout the first chapters, the applications of our algorithms are evidently related to financial modelling. However, in Chapter 4 we investigate how to simulate the concentration of a molecular species, how it changes according to a marked point process (whose probability of transition between two main stable equilibrium points will be simulated efficiently using the algorithm proposed in Chapter 3), and a valuable jump diffusion approximation proposed by (Leite and Williams, 2017) that not also implies a faster way to perform simulations, but it also addresses the domain problems an absolutely continuous diffusion approximation may have when the process leaves $[0, 1]$.

Chapter 1

Preliminary concepts

The goal of our first chapter is to review the probabilistic tools that will set the basis of our work. The majority of these references are found in (Shreve, 2004) and (Runggaldier, 2003). Unless otherwise stated, we are working in a probability space consisting of a probability measure $\mathbb{P}$ defined over a sample space Ω and with an associated filtration $\mathcal{F}_t$. With this in mind we need to develop models which have both a continuous and a discrete dynamic.

1.1 Itô integrals

When integrating with respect to a Brownian motion, naively we could think that we are dealing with a Riemann-Stieltjes integral, i.e., an integral of an integrand with respect to the differential of a function. However, we would be wrong since the differential of our function is almost nowhere differentiable. Hence, in Riemann-Stieltjes terms this does not make sense. On top of that the integrand might be non-deterministic, requiring to impose certain restrictions to it; for instance we need for it to be $\mathcal{F}_t$-adapted, meaning that the information available up to time t is sufficient to evaluate it at that time. For these reasons stochastic calculus provides us with a wide set of theory and techniques to help us model these kind of processes.

Definition 1.1.1. *Itô process.* Consider a Brownian Motion $W(t)$, $t \geq 0$, the deterministic value $X(0)$ and the adapted stochastic processes $b(u, X(u))$ and $\sigma(u, X(u))$. We can call an *Itô process* a

stochastic process of the form

$$X(t) = X(0) + \int_0^t b\left(u, X(u)\right) du + \int_0^t \sigma\left(u, X(u)\right) dW(u). \tag{1.1.0.1}$$

According to (Iacus, 2009) for (1.1.0.1) to have a unique, continuous, adapted strong solution such that $\mathbb{E}\left[\int_0^T |X(s)|^2 ds\right] < \infty$ we need to have the following conditions:

- *Global Lipschitz.* For all $x, y \in \mathbb{R}$ and $t \in [0, T]$ there exists a finite constant K such that

$$|b(t,x) - b(t,y)| + |\sigma(t,x) - \sigma(t,y)| < K|x - y|.$$

- *Linear growth.* For all $x \in \mathbb{R}$ and $t \in [0, T]$ there exists a finite constant C such that

$$|b(t,x)| + |\sigma(t,x)| < C(1 + |x|).$$

There might be cases where local versions of these conditions are enough.

Often in the literature, an Itô process is written in its differential form, i.e.

$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t). \tag{1.1.0.2}$$

Alongside this text the dependence of t in both $b(t, X(t))$ and $\sigma(t, X(t))$ might be omitted for notation simplicity.

The following concept is important since it makes a difference between the differentiability of a real-valued function and an Itô process. To every function, in particular to a random variable, we can define several kinds of variations. We are particularly interested in the variation of order two, since it is usually non-zero in Itô processes, which makes it different from real-valued functions.

Define a partition of the interval $[0, T]$ as $\Pi = \{t_0, t_1, ..., t_n\}$. The *size of the partition* is $||\Pi|| = \max_{j=0,...,n-1} (t_{j+1} - t_j)$. We then proceed to the next set of definitions.

Definition 1.1.2. *Quadratic variation.* Let $f(t)$ be a function defined for $0 \leq t \leq T$ and define a partition of the interval $[0, T]$ as $\Pi = \{t_0, t_1, ..., t_n\}$. The quadratic variation of f up to time T is

$$[f, f](T) = \lim_{||\Pi|| \to 0} \sum_{j=0}^{n-1} \left[f(t_{j+1}) - f(t_j)\right]^2.$$

Definition 1.1.3. *Quadratic variation of an Itô process.* We can define the quadratic variation of an Itô Process as

$$[X, X](t) = \int_0^t \sigma^2(X(u), u)du.$$

One could interpret this as having an infinitesimally small amount of time dt which accumulates $\sigma^2(t, X(t))$ units of change of the process $X(t)$. Informally, the quadratic variation can be written in a differential way as

$$dX(t)dX(t) = \sigma^2(t, X(t))dt. \tag{1.1.0.3}$$

1.2 Marked Point Processes

The next essential part in the foundation of our theory is describing counting events that occur randomly over time. Intuitively, such processes can be found in day-to-day applications of probability theory in the form of Poisson Processes.

To describe a Poisson Process, consider the sequence $\{T_n\}_{n\geq 1}$ of independent exponential random variables, each one of them being a waiting time for a specific event, which can be referred to as a jump. In other words, the first jump happens at time T_1, then the second one happens T_2 time units after the first one, etc. We can also account for the cumulative version of such inter-arrival times. Defining $S_n = \sum_{n=1}^{N} T_n$ we are describing the time the N-th jump happened. A Poisson process $N(t)$ counts the number of jumps that have happened up to time t. It is well known that this quantity has a Poisson distribution with parameter λt. We refer to λ as the intensity of the process. The following definition is a generalization of the dynamic described above.

Definition 1.2.1. *Marked point process.* Let $\{T_n\}_{n\geq 1}$ be a sequence of non-negative random variables describing the inter arrival time of a random event. Also consider the sequence $\{Y_n\}_{n\geq 1}$ of random variables taking values in the set $\mathcal{M} = \{1, 2, ..., K\}$. We call the double sequence (T_n, Y_n) an $\mathcal{M}-$marked point process.

We should also consider the following representation of a univariate marked process, for k $\in \mathcal{M}$:

$$N_k(t) := \sum_{n\geq 1} \mathbb{1}_{\{T_n \leq t\}} \mathbb{1}_{\{Y_n = k\}}. \tag{1.2.0.4}$$

And if we consider the vector $N(t) = (N_1(t), N_2(t), ..., N_k(t))$, we have a *multivariate point process* (k-variate). A slight generalization can be done when we want to deal with a subset $A \subseteq \mathcal{M}$, in which case we can re-write (1.2.0.4) as:

$$N_A(t) := \sum_{n \geq 1} \mathbb{1}_{\{T_n \leq t\}} \mathbb{1}_{\{Y_n \in A\}}. \tag{1.2.0.5}$$

To such marked point process we can define the following *random counting measure*:

$$p((0,t], A) := N_A(t), \quad t \geq 0, \quad A \in \mathcal{M}. \tag{1.2.0.6}$$

which can be thought of as the number of events of the class A that occurred between time 0 and t and in fact allows us to interpret integrals of given functions $H(s,y)$ with respect to a random counting measure as follows:

$$\int_0^t \int_A H(s,y) p(ds, dy) = \sum_{n \geq 1} H(T_n, Y_n) \mathbb{1}_{\{T_n \leq t\}} = \sum_{n=1}^{N(t)} H(T_n, Y_n). \tag{1.2.0.7}$$

We will often see a generalization of (1.2.0.7) where the integrand of $\int_0^t \int_{\mathcal{M}} H(s,y) p(ds,dy)$ can also depend on another stochastic process and not only on s, i.e.,

$$\int_0^t \int_{\mathcal{M}} H(X(s), y) p(ds, dy).$$

Each point process $N(t)$ has an intensity λ. In the simplest case, the one of a Poisson process, it describes the rate at which a number of random events occur in a finite amount of time, modelling as well the arrival time between two events. In a more general framework, we can describe the intensity of a point process as $\lambda_t(dy) = \lambda_t \nu(dy)$ where λ_t refers to the intensity of the point process $N(t)$ and $\nu(dy)$ is a probability measure on $\mathcal{M}$, i.e., it represents the different values the sequence Y_n can take. The pair $(\lambda_t, \nu(dy))$ is called *local characteristics* of $p(ds, dy)$. Certain processes admit an intensity depending itself on the current state of the process (or the value of the process one instant before). In such cases we say that we have a *state-dependent intensity*, which is usually denoted as $\Lambda(X(t))$.

A common generalization of (1.2.0.6) is $q(ds, dy) = p(ds, dy) - \lambda_s \nu(dy) ds$. We call it a *martingale measure* since integrals of the form

$$\int_0^t \int_{\mathcal{M}} H(s,y) q(ds, dy)$$

are $\mathbb{P}-$martingales. The part $\lambda_s \nu(dy)ds$ is called the *compensator of the counting measure.*

If the marks $\{Y_n\}$ take one of finitely many possible zero values $y_1, y_2, ..., y_K$, it is possible to decompose the marked point process into K independent constant-marked point processes.

Theorem 1.2.1 (Decomposition of a marked point process)**.** *Let $y_1, ..., y_K$ be a finite set of non-zero numbers such that $\mathbb{P}[Y_i = y_k] := p(y_k) > 0$ for every k and $\sum_{k=1}^{K} p(y_k) = 1$. Let λ_t be given and let $N_1(t), N_2(t), ..., N_K(t)$ be independent Poisson processes, each $N_k(t)$ having intensity $\lambda_t p(y_k)$. Define*

$$N(t) = \sum_{k=1}^{K} y_k N_k(t).$$

Then $N(t)$ is a marked point process with intensity λ_t and a probability distribution $p(y_k)$ over the set of marks $y_1, ..., y_K$.

For ease of notation we leave out any further dependence of another process in the intensity λ_t, but there will be some cases where it is dependent with another process and time t.

1.3 Jump-Diffusions

We need to define now a way to add up both characteristics defined in the previous two sections: the continuous part and the jumping process.

Definition 1.3.1. *Jump- diffusion.* Given a non-random starting point $X(0)$, a jump-diffusion is a stochastic process of the form

$$X(t) = X(0) + \int_0^t b(s, X(s))ds + \int_0^t \sigma(s, X(s))dW(s) + \int_0^t \int_{\mathcal{M}} \gamma(X(s-), y)p(ds, dy).$$

where $b(s, X(s))$ and $\sigma(s, X(s))$ are integrable, $\mathcal{F}_t-$adapted processes and $\gamma(X(s-), y)$ is a process being $\mathcal{F}_t-$predictable and larger than -1. The first two integrals are defined in (1.1.1). As for $p(ds, dy)$, it is a random counting measure whose integral is interpreted in (1.2.0.7), whose random events occur according to a state-dependent point process with intensity $\Lambda(X(t))$ and whose marks take values in the set $\mathcal{M}$ according to the probability distribution $\nu(dy)$. The notation $t-$ refers to the value of the process just before the jump event.

The differential form of the equation in (1.3.1) is:

$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t) + \int_{\mathcal{M}} \gamma(X(t-), y)p(dt, dy). \tag{1.3.0.1}$$

According to (1.2.0.6) and (1.2.0.7) we may re-write (1.3.0.1) as

$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t) + \gamma(X(t-), Y(t))dN(t). \tag{1.3.0.2}$$

Example 1.3.1. When it comes to a particular case of (1.3.0.2) one could think of $dN(t)$ representing a point process able to visit finitely many states. Using theorem 1.2.1 it is easy to see that a Point process with local characteristics $(\Lambda(X(t)), \nu(dy))$ visiting K states in the state space can be decomposed into K different Point processes, each one of them with intensity $\Lambda_k(X(t)) := \Lambda_k$ over the state k (for $k = 1, ..., K$). In such case (1.3.0.2) becomes

$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t) + \sum_{k=1}^{K} \gamma_k(X(t-))dN_k(t).$$

□

Example 1.3.2. In order to make (1.3.0.2) even more specific, consider a special case of example 1.3.1, when the jump part corresponds to a birth-and-death process. In such case $K = 2$; the jump size is $\gamma_1(X(t-)) = 1$, $N_1(t)$ is a point process that only jumps up by 1 according to the intensity $\Lambda_1(X(t))$. Analogously $\gamma_2(X(t-)) = -1$, $N_2(t)$ is a point process that only jumps down by -1 according to the intensity $\Lambda_2(X(t))$, giving the differential equation

$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t) + dN_1(t) - dN_2(t).$$

□

A number of different combinations of the parameters in 1.3.0.1 yield to different models widely reviewed in the literature.

Example 1.3.3. In (Merton, 1976) the premises of the Black-Scholes option pricing formula are challenged by questioning the assumption of the underlying stock being described by a stochastic process with a continuous path. The parameter set in this case is $b(x) = bx$, $\sigma(x) = \sigma x$, $\Lambda(x) = \lambda$, $\gamma(x, y) = x(e^y - 1)$ and the marks are drawn from a normal distribution with mean m and variance s^2, with $b, \sigma, \lambda > 0$. □

Example 1.3.4. In a similar manner (Kou, 2002) proposes an alternative to the Black-Scholes model, trying to circumvent the fact that the log returns of the price of an asset have heavier tails than those of a normal distribution. This is done with the same combination of parameters as in example 1.3.3 but changing the distribution of the marks to the double exponential distribution, which is described as $\nu(dy) = (p\eta_1 e^{\eta_1 y}\mathbb{1}_{y\geq 1} + q\eta_2 e^{\eta_2 y}\mathbb{1}_{z<0})dy$. This replacement has also the advantage of giving closed analytical results due to the memoryless property of the exponential distribution. □

Example 1.3.5. In order to tackle the positive relationship between default and equity volatility, as well as the negative relationship between volatility and stock price, (Carr and Linetsky, 2006) proposed a model with $b(x) = (r + \Lambda(x))x$, $\sigma(x) = ax^{\beta+1}$, $\Lambda(x) = b + ca^2x^{2\beta}$ and $\gamma(x, y) = -x$, with $r, a > 0$, $b \geq 0$, $c > 0.5$ and $\beta < 0$. □

1.4 Itô's formula

During the development of several formulae, we need to find the differential of a time-dependent function of a stochastic process. This can be done using the following result:

Theorem 1.4.1 (Itô's formula)**.** *Let $X(t)$ be a solution to $dX(t) = b(t, X(t))dt + \sigma(t, X(t))dW(t)$ be an Itô process and let $f(t, x)$ be a function with $f_t(t, x)$ and $f_x(t, x)$ defined and continuous. Then for $T > 0$*

$$f(T, X(T)) = f(0, X(0)) + \int_0^T f_t(t, X(t))dt + \int_0^T f_x(t, X(t))dX(t) + \frac{1}{2}\int_0^T f_{xx}\left(t, X\left(t\right)\right) d\left[X, X\right](t). \tag{1.4.1.1}$$

Re-writing the quadratic variation part in (1.4.1.1) yields:

$$\begin{aligned} f(T, X(T)) = f(0, X(0)) &+ \int_0^T f_t(t, X(t))dt + \int_0^T f_x(t, X(t))b(t, X(t))dt \\ &+ \int_0^T f_x(t, X(t))\sigma(t, X(t))dW(t) + \frac{1}{2}\int_0^T f_{xx}(t, X(t))\sigma^2(t, X(t))dt, \end{aligned} \tag{1.4.1.2}$$

or it also can be expressed in the differential notation

$$df(t,X(t)) = f_t(t,X(t))dt + f_x(t,X(t))dX(t) + \frac{1}{2}f_{xx}(t,X(t))dX(t)dX(t).$$

Since our main interest is to work with jump diffusions, we need to present the following generalization to (1.4.1.1).

Theorem 1.4.2 (Itô's formula for jump processes)**.** *Let $X(t)$ be such that it satisfies (1.3.0.2). Then for a twice-differentiable function f we have*

$$\begin{aligned} df(t,X(t)) &= f_t(t,X(t))dt + f_x(t,X(t))b(t,X(t))dt + \frac{1}{2}f_{xx}(t,X(t))\sigma^2(t,X(t))dt \\ &\quad + f_x(t,X(t))\sigma(t,X(t))dW(t) \\ &\quad + [f\,(t,X(t-) + \gamma(X(t-),Y(t)) - f(t,X(t-))]dN(t). \end{aligned} \tag{1.4.2.1}$$

Also to find the differential of a product of stochastic processes we might need the following corollary from (Shreve, 2004).

Corollary 1.4.2.1 (Itô's Product Rule)**.** *Let $X_1(t)$ and $X_2(t)$ be jump processes.Then*

$$X_1(t)X_2(t) = X_1(0)X_2(0) + \int_0^t X_2(s-)dX_1(s) + \int_0^t X_1(s-)dX_2(s) + [X_1,X_2](t).$$

If we consider that said jump processes have two parts: an Itô process $(X_i^c(t))$, say, and a marked point process $J_i(t)$, say, then we can rewrite (1.4.2.1) as:

$$\begin{aligned} X_1(t)X_2(t) &= X_1(0)X_2(0) + \int_0^t X_2(s)dX_1^c(s) + \int_0^t X_1(s)dX_2^c(s) + [X_1^c,X_2^c](t) \\ &\quad + \sum_{0<s\leq t}[X_1(s)X_2(s) - X_1(s-)X_2(s-)]. \end{aligned} \tag{1.4.2.2}$$

The first application of Theorem 1.4.2 is finding a solution to equations of the form (1.3.0.2). Indeed, if we consider $f(t,x) = ln(x)$, and $dX(t) = X(t-)[b(X(t))dt + \sigma(X(t))dW(t) + \gamma(X(t-),Y(t))dN(t)]$:

$$
\begin{aligned}
d(ln(X(t))) &= 0 * dt + \frac{1}{X(t)} b\left(X(t), t\right) X(t) dt + \frac{1}{2} \frac{-1}{X(t)^2} \sigma^2(X(t), t) X(t)^2 dt \\
&+ \frac{1}{X(t)} X(t) \sigma(X(t), t) dW(t) \\
&+ [ln(X(t-)(1 + \gamma(X(t-), Y(t)))) - ln(X(t-))] dN(t) \\
&= b(X(t), t)dt - \frac{1}{2}\sigma^2(X(t), t)dt + \sigma(X(t), t)dW(t) + ln(1 + \gamma(X(t-), Y(t)))dN(t).
\end{aligned} \tag{1.4.2.3}
$$

As before, this equation may be commonly written in integral form

$$
\begin{aligned}
ln(X(t)) - ln(X(0)) = \int_0^t b(X(s), s) - \frac{1}{2}\sigma^2(X(s), s)ds + \int_0^t \sigma(X(s), s)dW(s) \\
+ \int_0^t ln(1 + \gamma(X(s-), Y(s)))dN(s).
\end{aligned}
$$

Exponentiating both sides and solving for $X(t)$ we get

$$
X(t) = X(0) e^{\int_0^t b(X(s),s) - \frac{1}{2}\sigma^2(X(s),s)ds + \int_0^t \sigma(X(s),s)dW(s) + \int_0^t ln(1+\gamma(X(s),Y(s-)))dN(s)},
$$

and using (1.2.0.7) we re-write the jump part, which gives us what we will call the *exponential formula*:

$$
X(t) = X(0) e^{\int_0^t b(X(s),s) - \frac{1}{2}\sigma^2(X(s),s)ds + \int_0^t \sigma(X(s),s)dW(s)} \prod_{n=1}^{N(t)} \left(1 + \gamma(T_n, Y_n)\right). \tag{1.4.2.4}
$$

1.5 Change Of measure

Let us provide an intuitive notion on how to describe the dynamics of a stochastic process under an equivalent probability measure, which will be formalized later. Being $\mathbb{P}$ the original measure we will need a non-negative random variable L such that $\mathbb{E}[L] = 1$. Then the new probability measure $\tilde{\mathbb{P}}$ is defined for every $A \in \mathcal{F}$ as:

$$
\tilde{\mathbb{P}}(A) = \int_A L d\mathbb{P}.
$$

The variable L is usually referred to as the *Radon-Nykodym derivative* of $\tilde{\mathbb{P}}$ with respect to $\mathbb{P}$, which we can refer intuitively as

$$L = \frac{d\tilde{\mathbb{P}}}{d\mathbb{P}}. \tag{1.5.0.5}$$

For illustrative purposes, consider a stochastic process satisfying (1.1.0.2), which we are interested in expressing under a new probability measure, having drift term $\tilde{b}(t, X(t))$ instead of $b(t, X(t))$. Then if we define the Radon-Nykodym derivative as (1.5.0.5) and we define a new process $\Theta(t, X(t))$ such that

$$\Theta(t, X(t)) = \frac{\tilde{b}(t, X(t)) - b(t, X(t))}{\sigma(t, X(t))}.$$

It turns out that $d\tilde{W}(t) = dW(t) - \Theta(t, X(t))dt$ is a $\tilde{\mathbb{P}}-$Brownian Motion. This implies that under $\tilde{\mathbb{P}}$ the process $X(t)$ satisfies the stochastic differential equation:

$$\begin{aligned}
dX(t) &= b(t, X(t))dt + \sigma(t, X(t))dW(t) \\
&= b(t, X(t))dt + \sigma(t, X(t))(d\tilde{W}(t) + \Theta(t, X(t))dt) \\
&= b(t, X(t))dt + \sigma(t, X(t))d\tilde{W}(t) + \sigma(t, X(t))\Theta(t, X(t))dt \\
&= b(t, X(t))dt + \sigma(t, X(t))d\tilde{W}(t) + (\tilde{b}(t, X(t)) - b(t, X(t)))dt \\
&= \tilde{b}(t, X(t))dt + \sigma(t, X(t))d\tilde{W}(t).
\end{aligned} \tag{1.5.0.6}$$

In the same way, we need to define a derivative of change of measure for any kind of process, not only for Itô processes. To such end, we have the following theorem:

Theorem 1.5.1 (Girsanov's Theorem)**.** *For every t in the finite time interval $[0, T]$ let ψ_t be an $\mathcal{F}_t-$predictable process and $h_t(y) \geq 0$ an $\mathcal{F}_t-$predictable $\mathcal{M}-$indexed process such that $\int_0^t \psi_s \lambda_s ds < \infty$ and $\int_{\mathcal{M}} h_t(y)\nu(dy) = 1$. Let $L(t) = L_c(t) * L_j(t)$ such that $L_c(t)$ satisfies*

$$dL_c(t) = L_c(t)\Theta_t dW_t, \quad L(0) = 1,$$

and $L_j(t)$ satisfies

$$dL_j(t) = \int_{\mathcal{M}} (\psi_t h_t(y) - 1)\, L_j(t-) q(dt, dy), \tag{1.5.1.1}$$

with $q(dt,dy) = p(dt,dy) - \lambda_t \nu(dy)dt$ *(i.e., the martingale measure associated with* $p(dt,dy)$*). If* $\mathbb{E}[L_j(t)] = 1 = \mathbb{E}[L_c(t)]$ *for all* t *then there exists a probability measure* $\widetilde{\mathbb{P}}$ *equivalent to* $\mathbb{P}$ *such that*

$$dW(t) = \Theta(t)dt + d\tilde{W}(t),$$

where $\tilde{W}(t)$ *is a* $\widetilde{\mathbb{P}}-$*Brownian Motion. Also, under* $\widetilde{\mathbb{P}}$*,* $p(dt,dy)$ *has local characteristics* $(\psi_t\lambda_t, h_t(y)\nu(dy))$.

We want of course to have an explicit expression for $L(t)$. A simple of (1.4.2.1) yields:

$$\begin{aligned} dL(t) &= d[L_c(t) * L_j(t)] \\ &= L_c(t-)dL_j(t) + L_j(t)dL_c(t) \\ &= L(t)\Theta(t)dW(t) + L(t-)\int_{\mathcal{M}} (\psi_t h_t(y) - 1)q(dt,dy), \quad L(0) = 1, \end{aligned} \tag{1.5.1.2}$$

whose solution using (1.4.2.4) is:

$$\begin{aligned} L(t) = \exp\left\{-\frac{1}{2}\int_0^t \Theta(s)^2 ds + \int_0^t \Theta(s)dW(s)\right\} \\ * \exp\left\{\int_0^t \int_{\mathcal{M}} (1 - \psi_s h_s(y))\lambda_s m_s(dy)ds\right\} \prod_{n=1}^{N(t)} \left(\psi_{T_n} h_{T_n}(Y_n)\right). \end{aligned} \tag{1.5.1.3}$$

Example 1.5.1. *(Examples 1.3.1 and 1.3.2 continued)* If we are dealing with a multivariate point process, (1.5.1.1) may be re-written as

$$L_j(t) = \sum_{k=1}^{K} (\psi_t(k) - 1)L_j(t-)(dN_k(t) - \lambda_k(t)dt), \tag{1.5.1.4}$$

and in such case (1.5.1.3) is

$$L(t) = \exp\left\{-\frac{1}{2}\int_0^t \Theta(s)^2 ds + \int_0^t \Theta(s)dW(s)\right\} * \prod_{k=1}^{K} \left[exp\left\{\int_0^t (1 - \psi_k(s))\lambda_k(s)ds\right\} \prod_{n=1}^{N_k(t)} \psi_{T_n}(k)\right]. \tag{1.5.1.5}$$

It is important to note a difference between (1.5.1.3) and (1.5.1.5): in the latter we lose the term $h_{T_n}(Y_n)$ that appeared in the former. This is because the decomposed point process does not involve

explicitly a probability distribution for the marks, since each value the Y_i can take is reflected in a factor of the product $\prod_{n=1}^{N_k(t)} \psi_{T_n}(k)$.

Note also that (1.5.1.5) may be used to give the derivative of change of measure of the jump-diffusion proposed in example 1.3.2, with $K = 2$. $\square$

Chapter 2

Numerical methods for Markov chains and jump-diffusions

Consider a process X(t) either behaving like a marked point process or solving a stochastic differential equation as (1.3.0.1). We might be interested in a whole trajectory up to time T or in the expected value of some function of the process (e.g., moments, distributions). Both requirements are equally pertinent since they arise in important applications such as option valuation. To name a couple: pricing an American option involves knowing the expected value of the payoff at expiration date, or perhaps finding the value of an Asian option which needs to have the underlying stock option price tracked during the whole period of its validity. Overall, such quantities are not usually available in an explicit analytical form, which is why we will first discuss the principles of solving this problem via Monte Carlo simulation, and we will give intuitive ideas of why it works, as described in (Iacus, 2009). More importantly, we will give an explanation on some discretization schemes, which will allow us to draw a random sample of the distribution of X(t).

2.1 Monte Carlo methods

Say we are interested in estimating an integral of the form $\mathbb{E}[g(X)]$. This expected value might not have a closed form, due for instance, to its complexity or because the distribution of X might not be analytically tractable. Despite this difficulty, if a random sample $X_1, X_2, ..., X_n$ drawn from the distribution of X can be made, a fairly reliable tool to use as an estimator of $\mathbb{E}[g(X)]$ is the sample

mean of $g(X_1), g(X_2), ..., g(X_n)$, i.e.,

$$\widehat{\mathbb{E}[g(X)]} = \frac{1}{n}\sum_{i=1}^{n} g(X_i). \tag{2.1.0.1}$$

Let $g_n = \widehat{\mathbb{E}[g(X)]}$. It can be shown that g_n is unbiased, i.e., $\mathbb{E}[g_n] = \mathbb{E}[g(X)]$. A first idea on how to evaluate the performance of an estimator would be using $Var(g_n)$; however, this measure is scale dependent and would not present a standardized notion of the estimator's execution. For this reason a common practice to asses whether or not an estimator accomplishes its purpose is through its *relative error*. In a group of estimators, the one with the smallest relative error is usually a good choice. Such quantity is defined as:

$$R = \frac{\sqrt{Var(g_n)}}{\mathbb{E}[g_n]}.$$

Even though Monte Carlo methods seem unsophisticated, they are backed up by the next three transparent concepts from frequentist statistics, which can be consulted in (Grimmett and Stirzaker, 2001).

Theorem 2.1.1 (Law of large numbers)**.** *Given a random sample $X_1, X_2, ..., X_n$, as $n \longrightarrow \infty$*

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \xrightarrow{D} \mathbb{E}[g(X)]$$

Theorem 2.1.2 (Ergodic theorem for stationary sequences)**.** *Under mild conditions on the joint distribution of the random sample $X_1, X_2, ..., X_n$, almost surely as $n \longrightarrow \infty$*

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \longrightarrow \mathbb{E}[g(X)].$$

Theorem 2.1.3 (Central limit theorem)**.** *Let $X_1, X_2, ...$ be a sequence of independent identically distributed random variables with mean μ and standard deviation σ. Also consider Y a standard normal random variable. Then as $n \longrightarrow \infty$*

$$\frac{1}{n}\frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma} \xrightarrow{D} Y.$$

These three theorems guarantee that as $n \longrightarrow \infty$

$$\widehat{\mathbb{E}[g(X)]} \xrightarrow{D} N\left(\mathbb{E}[g(X)], \frac{Var[g(X)]}{n}\right).$$

Here the abuse of notation $N(m, s^2)$ refers to a normal random variable with mean m and variance s^2.

Given that the .95 quantile of a standard normal random variable is 1.96, we can construct a 95% confidence interval for the theoretical value of $\mathbb{E}[g(X)]$ as follows:

$$\left(\widehat{\mathbb{E}[g(X)]} - 1.96\frac{\hat{\sigma}}{\sqrt{n}}, \widehat{\mathbb{E}[g(X)]} + 1.96\frac{\hat{\sigma}}{\sqrt{n}}\right), \tag{2.1.3.1}$$

where $\hat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n}\left(g(X_i) - \widehat{\mathbb{E}[g(X)]}\right)$ is the unbiased estimator of $Var[g(X)]$.

2.2 Birth-and-death process simulation

A simple particular case of the marked point processes, which were introduced in section 1.2, is a birth-and-death process, whose jumps are only equal to 1 (i.e., *birth*) or -1 (i.e., *death*), i.e., $\mathcal{M} = \{\pm 1\}$. Said events arrive respectively according to the state-dependent rates $\lambda(x)$ and $\mu(x)$. In order to design an algorithm for simulating a birth-and-death process we can take some key concepts on continuous time Markov chains from (Allen, 2010). It can be shown that in a birth-and-death process the jump intensity of any jump event is $\lambda(x) + \mu(x)$, therefore the interarrival times have an exponential distribution with parameter $\lambda(x) + \mu(x)$. Furthermore, being in a state x a birth-and-death process jumps to $x + 1$ with probability $\lambda(x)/(\lambda(x) + \mu(x))$ and it jumps to $x - 1$ with probability $\mu(x)/(\lambda(x) + \mu(x))$. Hence, we propose Algorithm 1 for a time horizon T, an initial time t_0 and an initial state x_0.

2.3 Discretization methods for diffusions and jump-diffusions

The challenge about simulating a trajectory of X up to time T satisfying (1.3.0.1) may appear as early as when the stochastic differential equation does not have a discrete part, i.e., cases like (1.1.0.2). A considerable amount of cases may be solved using (1.4.2.4), but when this fails we need to approximate the solution (e.g., using a discretization process like the Euler scheme).

The next level of complexity is when $p(dt, dy)$ has a constant intensity λ. Clearly this is the case of a Compound Poisson Process and we can generate independently a discretization of the continuous part and the jump times.

Algorithm 1 Birth-and-death process simulation

Set T, t_0 and x_0
Assign $x = x_0$
Assign $t = t_0$
while $t < T$ **do**
    Calculate the birth rate $\lambda(x)$
    Calculate the death rate $\mu(x)$
    Set $dt = random(exp(mean(\lambda(x) + \mu(x)))$
    Set the probability of a birth $p(birth) = \lambda(x)/(\mu(x) + \lambda(x)))$
    Set the probability of a death $p(death) = \mu(x)/(\mu(x) + \lambda(x)))$
    Sample a jump event $jump = random(p(birth), p(death))$
    $x = x + jump$
    $t = t + dt$
end while

When instead of having a Poisson process drive the jump times, we are dealing with a state-dependent intensity $\Lambda(X_t)$, the discretization of the continuous part is no longer independent of the jump times, which is why more elaborate approximation methods are needed. We will review a procedure proposed by (Giesecke et al., 2015).

2.3.1 Euler approximation

This scheme of approximation was originally intended to give solutions to deterministic differential equations, which makes it quite intuitive to use.

Definition 2.3.1. *Euler approximation of X.* Suppose we have an Itô process of the form (1.1.0.2) with initial value $X(0) = x_0$. We can propose a partition to the interval $[0, T]$, $0 = t_0 < t_1 < ... < t_N$ and we will call it $\Pi_N([0, T])$. The Euler approximation of X is a continuous stochastic process Z satisfying for $i = 0, 1, ..., N - 1$,

$$Z(t_{i+1}) = Z(t_i) + b(Z(t_i))(t_{i+1} - t_i) + \sigma(Z(t_i))(W(t_{i+1}) - W(t_i)).$$

For a time $t \in [t_i, t_{i+1})$ the process is defined using linear interpolation

$$Z(t) = Z(t_i) + \frac{t - t_i}{t_{i+1} - t_i} \left(Z(t_{i+1}) - Z(t_i) \right).$$

Using this approximation is very straightforward since we know the distribution of $W(t_{i+1}) - W(t_i)$, which is normal with mean 0 and variance $t_{i+1} - t_i$.

The following definition proposes a way to classify different discretization schemes.

Definition 2.3.2. *Weak order of convergence.* The discretization Z is said to converge weakly of order β to X if for a fixed time T and a continuous function g being differentiable $2(\beta + 1)$ times and having polynomial growth, the following statement holds for independent constants C and δ:

$$|\mathbb{E}[g(X(T))] - \mathbb{E}[g(Z(T))]| \leq C\delta^{\beta}.$$

The Euler scheme is weakly convergent of order $\beta = 1$ (see Section 2 in (Iacus, 2009)). The following proposed scheme is arbitrarily convergent to 1, where "arbitrarily" stands for "as close to 1 as small is the time discretization step" (see Theorem 4.6 in (Giesecke et al., 2015)).

2.3.2 Giesecke/ Teng/ Wei approximation

Since the jump times of X cannot be generated independently of the process because the intensity measure is state-dependent, we will construct another jump diffusion, Z, using time scaling, i.e., we will approximate the jump times using a standard Poisson process.

Consider first $\{(\epsilon_n, Y_n)\}_{n \geq 1}$ where the ϵ_i are exponential random variables with mean 1 and the Y_n are drawn from the distribution $\nu(dy)$. Define then the sequence of random variables $0 = \tau_0 < \tau_1 < \tau_2 < ...$ such that

$$\tau_{n+1} = \inf \left\{ t > \tau_n : \int_{\tau_n}^{t} \Lambda(Z_s) ds \geq \epsilon_{n+1} \right\}.$$

These are the jump times of Z. It is clear that for $t \in [\tau_n, \tau_{n+1})$ the process Z behaves as a diffusion described by

$$Z(t) = Z(\tau_n) + \int_{\tau_n}^{t} b(Z(s)) ds + \int_{\tau_n}^{t} \sigma(Z(s)) dW(s). \tag{2.3.0.2}$$

And during a jump time

$$Z(\tau_{n+1}) = Z(\tau_{n+1}-) + \gamma(Z(\tau_{n+1}-), Y_{n+1}).$$

(Giesecke et al., 2015) shows that the way Z is constructed, i.e. X and Z having the same intensity measure and being a diffusion like (2.3.0.2) between jumps, X and Z have the same distribution and therefore Z is a good candidate to approximate $\mathbb{E}[g(X)]$.

Define the compensator process as:

$$A(t) = \int_0^t \Lambda(Z(s))ds.$$

Having shown a process Z which has the same distribution as X, we need now to propose a discretization for Z. Let Z^h be that discretization. We will approximate the process A by its discretization A^h.

Let us define first the size of each stem, $h = T/N_{steps}$. In the interval $[0, T]$ there are several points relevant to our discretization: the fixed grid points jh for $j = 0, 1, ..., N_{steps}$ and the approximate jump times $\{\tau_n^h\}_{n\geq 1}$. Combining both sets we have a sequence of discretization times which will be refered as $\{t_i\}$.

Accordingly to Definition 2.3.1, let the continuous approximations Z^h and A^h at the discretization times be:

$$Z^h(t_{i+1}-) = Z^h(t_i) + b(Z^h(t_i))(t_{i+1} - t_i) + \sigma(Z^h(t_i))(W(t_{i+1}) - W(t_i))$$

$$A^h(t_{i+1}) = A^h(t_i) + \Lambda(Z^h(t_i))(t_{i+1} - t_i).$$

While for any point between discretization times, i.e., for $t \in [t_i, t_{i+1})$

$$Z^h(t) = Z^h(t_i) + b(Z^h(t_i))(t - t_i) + \sigma(Z^h(t_i))(W(t) - W(t_i))$$

$$A^h(t) = A^h(t_i) + \Lambda(Z^h(t_i))(t_- t_i).$$

If we define the $n-th$ jump time as $E_n = \sum_{k\leq n} \epsilon_k$, we can approximate the τ_n by

$$\tau_n^h = \inf\{t : A^h(t) \geq E_n\}$$

which, given the fact that the discretization A^h is increasing, can be inverted to get

$$\tau_n^h = \eta_n^h + \frac{E_n - A^h(\eta_n^h)}{\Lambda(Z^h(\eta_n^h))},$$

where $\eta_n^h = \inf\{t_i : A^h(t_i) + \Lambda(Z(t_i))[(\lfloor \frac{t_i}{h} \rfloor + 1)h - t_i] > E_n\}$ is the last discretization time before the jump.

When an approximate jump happens at τ_n^h, the process Z^h must be updated as

$$Z^h(\tau_n^h) = Z^h(\tau_n^h -) + \gamma(Z^h(\tau_n^h -), Y_n).$$

As we stated at the beginning of this explanation, we must construct a set of discretization times as follows: $t_0 = 0$, $t_{i+1} = \inf_{jh, \tau_n^h > t_i}\{jh, \tau_n^h, T\}$. Also, we set $Z^h(t_i) = Z^h(t_i -)$ if $t_i \in \{jh : j = 1, ..., N_{steps}\}$.

This procedure is summarized in Algorithm 2.

2.4 Numerical examples

2.4.1 Call option on a short-interest rate

We will estimate the price of a call option on a short-interest rate, i.e., we will estimate $\mathbb{E}[(X(T) - K)_+]$, where X is the short term interest rate satisfying

$$dX(t) = \kappa(\theta - X(t))dt + \sigma dW(t) + \int_{\mathcal{M}} yp(dt, dy)$$

with $X(0) = x_0$ and κ, θ and σ are non-negative constants and $p(dt, dy)$ is a counting measure with intensity measure $\Lambda(X(t))\nu(dy)$. Specifically $\Lambda(x) = \Lambda_0 + \Lambda_1 x$, for $\Lambda_0 \geq 0$ and $\Lambda_1 > 0$ and $\nu(dy)$ a discrete distribution m.

The constants we will use are $x_0 = 0.1$, $\theta = 0.1$, $\kappa = 2$, $\sigma = 0.02$, $\Lambda_0 = 5$, $\Lambda_1 = 50$ and $K = 0.1$. The jump size distribution m takes values in the set $\{0.01, 0.015, 0.02, 0.025, 0.03\}$. This particular example is studied in (Giesecke et al., 2015), where the true value of the option, 0.126435, was found using analytical methods.

Each trajectory of Z^h was discretized using $N_{steps} = 400$, whereas the sample mean (2.1.0.1) is computed using $n = 160000$ trials. The estimated value of the call option is 0.1262217 and we can

Algorithm 2 Giesecke/Teng/Wei approximation to a jump diffusion

Set T, decide N_steps

Set time step $h = T/N_{steps}$

Initialize $i = j = n = 0$

Initialize $s = 0$

Set $Z^h(s) = x_0$

Set $A^h(s) = 0$

Set $E_n = random(exp(mean1))$

while $s < T$ **do**

    Compute $A^h_{temporary} = A^h(s) + \Lambda(Z^h(s))[(i+1)h - s]$

    if $A^h_{temporary} \geq E_n$, i.e. there is a jump in the interval $[s, (i+1)h]$ **then**

        Compute $\tau^h_n = s + \frac{E_n - A^h(s)}{\Lambda(Z^h(s))}$

        Compute $Z^h(\tau^h_n -) = Z^h(s) + b(Z^h(s))(\tau^h_n - s) + \sigma(Z^h(s))\sqrt{\tau^h_n - s} * random(N(0,1))$

        Compute $Z^h(\tau^h_n) = Z^h(\tau^h_n -) + \gamma(Z^h(\tau^h_n -), random(Y_n))$

        Accumulate time $s = \tau^h_n$

        Update $A^h(s) = E$

        Update $n = n + 1$

        Update $E_n = E_n + random(exp(mean1))$

    end if

    if There is no jump in the interval $[s, (i+1)h]$ **then**

        Compute $Z^h((i+1)h) = Z^h(s) + b(Z^h(s))((i+1)h - s) + \sigma(Z^h(s))\sqrt{(i+1)h - s} * random(N(0,1))$

        Update $s = (i+1)h$

        Update $A^h(s) = A^h_{temporary}$

        Update $i = i + 1$

    end if

end while

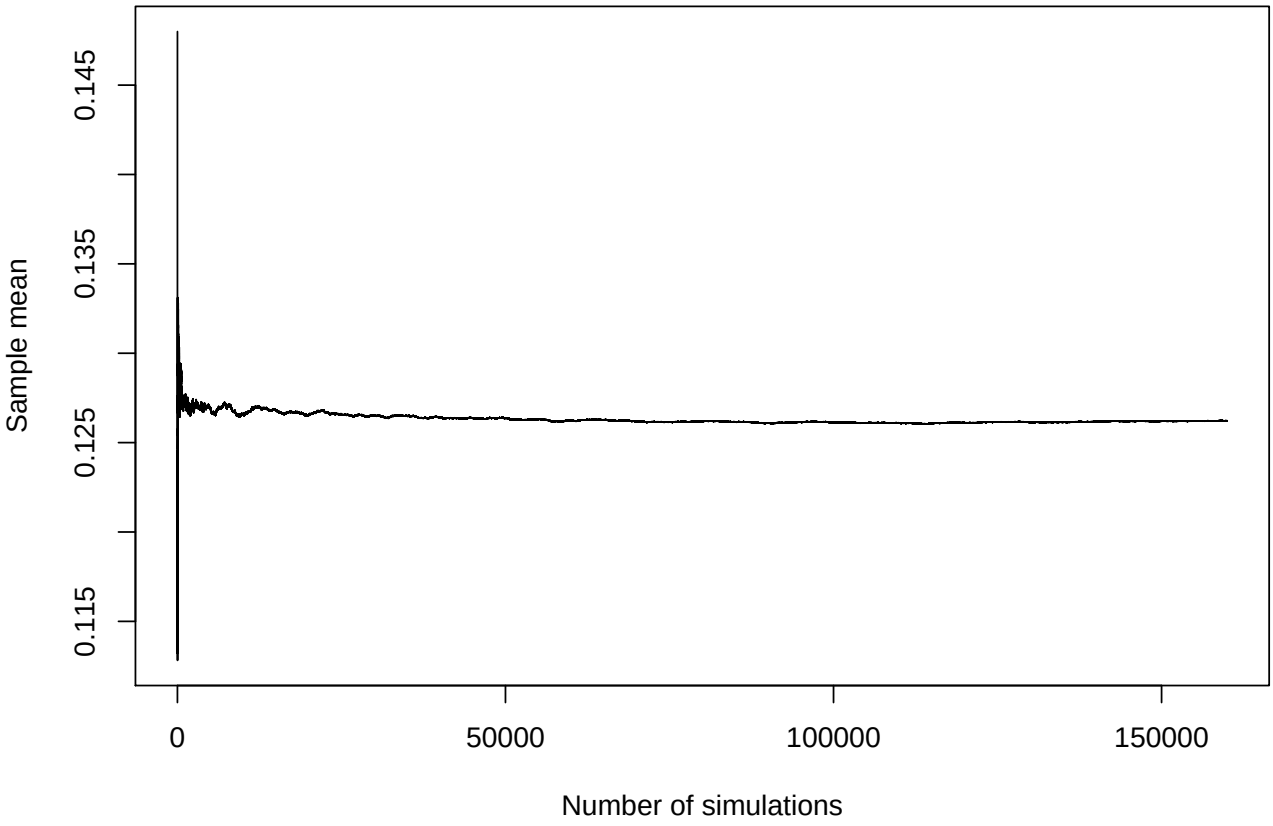

Figure 2.1: Convergence of the sample mean for estimating the price of a call option on a short-interest rate

see in figure (2.1) that it approximates asymptotically to the true value of the call option. The code in R to replicate this example may be found in Appendix A.

2.4.2 Brownian motion with a birth-and-death process

We will try our algorithm to approximate the value of $X(T)$, where X solves the equation

$$dX(t) = \theta dt + \sigma dW(t) + \int_{\mathcal{M}} yp(dt, dy)$$

with initial value $X(0) = x_0$. Here θ and σ are non-negative constants and $p(dt, dy)$ is a counting measure with intensity measure $\lambda(X(t))\nu(dy)$. Concretely, $\Lambda(x) = \Lambda_0 + \Lambda_1 x$, for $\Lambda_0 \geq 0$ and $\Lambda_1 > 0$

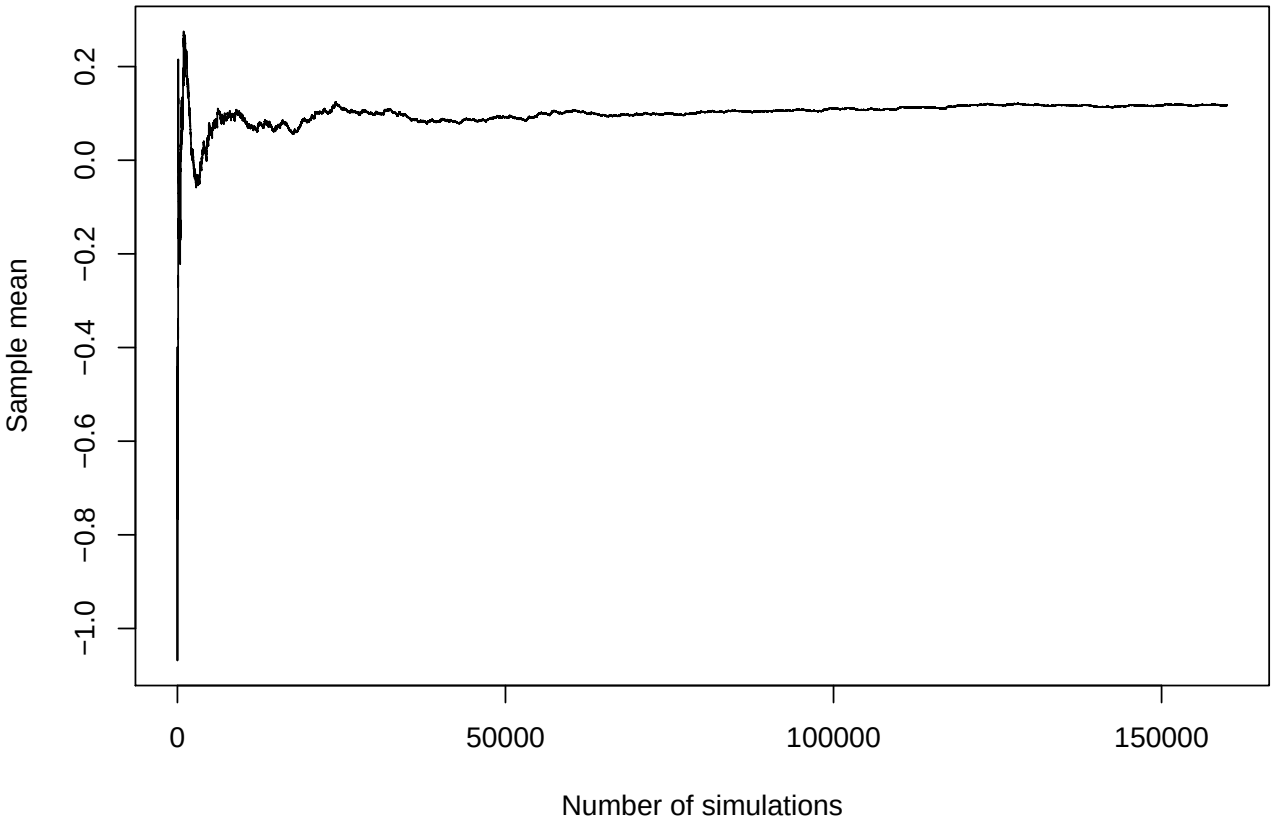

Figure 2.2: Convergence of the sample mean for estimating $X(1)$ for a Brownian Motion with a birth-and-death process

and $\nu(dy)$ takes the values -1 and 1 with equal probability.

The constants we will use are $x_0 = 0$, $\theta = 0.1$, $\sigma = 1$, $\Lambda_0 = 5$ and $\Lambda_1 = 50$.

Again, each trajectory of Z^h was discretized using $N_{steps} = 400$, whereas the sample mean (2.1.0.1) is computed using $n = 160000$ trials. The estimated value of $X(1)$ is 0.117543 and the graph on figure (2.2) shows again nice convergence behaviour.

Chapter 3

Rare event simulation

We will focus our attention in processes like (1.3.0.1), particularly jump-diffusions with small diffusion coefficients and whose magnitude of jumps is controlled by a parameter ε, i.e.,

$$dX(t) = b(X(t))dt + \sqrt{\varepsilon}\sigma(X(t))dW(t) + \int_{\mathcal{M}} \varepsilon\gamma(X(t-), Y(t))dN(t). \tag{3.0.0.1}$$

Alternatively the jump part could be visiting exclusively finitely many states. In that case the last equation becomes

$$dX(t) = b(X(t))dt + \sqrt{\varepsilon}\sigma(X(t))dW(t) + \sum_{k=1}^{K} \varepsilon\gamma_k(X(t-))dN_k(t). \tag{3.0.0.2}$$

Say we want to estimate the probability that the time τ^ε a process X leaves the domain $\Omega = (a, b)$ is less than or equal to fixed time T, i.e.,

$$p_\varepsilon := \mathbb{P}[\tau^\varepsilon \leq T] = \mathbb{P}[X(T) \notin \Omega]. \tag{3.0.0.3}$$

Some procedures to simulate a process' trajectory (and implicitly some related concepts, e.g., the exit time of a domain) were explored in chapter 2. We consider $[\tau^\varepsilon \leq T]$ as a rare event since having a small ε would mean the process takes longer to leave Ω. Raw Monte Carlo methods are not appropriate since few simulations of the process will end up in the relevant interval, resulting in a large relative error. A way to go around this is to lead the simulation towards the rare event, and then to properly re-weight the sample mean. This is known as *importance sampling*.

It is clear that (3.0.0.3) is a particular case of $\mathbb{E}[exp\{-ng(X^n(T))\}\mathbb{1}_{\{X^n(T)\notin\Omega\}}]$. We will explain the techniques developed mainly in (Djehiche et al., 2014), as well as some other references, where importance sampling algorithms are designed to approximate these kind of integrals.

3.1 Viscosity subsolutions to Hamilton Jacobi equations

In order to apply importance sampling we need to define a new sampling measure $\tilde{\mathbb{P}}$, from which we will draw a random sample of size n of $X(T)$, which solves (3.0.0.1) [or its particular case (3.0.0.2)]. To reflect the transformation from one measure to another, we will take into consideration Theorem 1.5.1. The drift in equation (3.0.0.1) will be modified choosing $\Theta(t, X(t))$ in (1.5.1.2) in a way the process is forced out of Ω faster.

Let $\hat{p_\varepsilon}$ be the estimator of p_ε. A first approach would be to consider

$$\hat{p_\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{\tau_i \leq T\}}.$$

Clearly, $\hat{p_\varepsilon}$ is unbiased since

$$\begin{aligned}
\mathbb{E}[\hat{p_\varepsilon}] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{\tau_i \leq T\}}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\mathbb{1}_{\{\tau_i \leq T\}}\right] \\
&= \frac{np_\varepsilon}{n} \\
&= p_\varepsilon. \qquad (3.1.0.4)
\end{aligned}$$

Analogously, the variance of $\hat{p_\varepsilon}$ is:

$$
\begin{aligned}
Var[\hat{p_\varepsilon}] &= Var\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{\tau_i \leq T\}}\right] \\
&= \frac{1}{n^2}\sum_{i=1}^{n} Var\left[\mathbb{1}_{\{\tau_i \leq T\}}\right] \\
&= \frac{np_\varepsilon(1-p_\varepsilon)}{n^2} \\
&= \frac{p_\varepsilon(1-p_\varepsilon)}{n}.
\end{aligned} \tag{3.1.0.5}
$$

Hence the relative error of $\hat{p_\varepsilon}$ is

$$
\begin{aligned}
R &= \frac{\sqrt{Var(\hat{p_\varepsilon})}}{\mathbb{E}[\hat{p_\varepsilon}]} \\
&= \frac{\sqrt{\frac{p_\varepsilon(1-p_\varepsilon)}{n}}}{p_\varepsilon} \\
&= \sqrt{\frac{1}{n}\left(\frac{1}{p_\varepsilon}-1\right)}.
\end{aligned} \tag{3.1.0.6}
$$

Given that p_ε is precisely the quantity we are interested in estimating, we will estimate it by its unbiased estimator. Therefore, the relative error of $\hat{p_\varepsilon}$ can be approximated as:

$$
R \approx \sqrt{\frac{1}{n}\left(\frac{1}{\hat{p_\varepsilon}}-1\right)}.
$$

Given the aforementioned disadvantages of using a plain sample mean to estimate p_ε, let us propose another an alternative sampling measure. We will use $d\mathbb{P}/\tilde{\mathbb{P}}$ as in Girsanov's Theorem. It turns out that if we generate X_i and consequently τ_i according to $\tilde{\mathbb{P}}$ and then define $\tilde{p}_\varepsilon$ as

$$
\tilde{p}_\varepsilon = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{\tau_i \leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i). \tag{3.1.0.7}
$$

we get another unbiased estimator for p_ε, this time unbiased under $\tilde{\mathbb{P}}$, i.e.,

$$\begin{aligned}\tilde{\mathbb{E}}[\tilde{p}] &= \tilde{\mathbb{E}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right]\\ &= \frac{1}{n}\sum_{i=1}^{n}\tilde{\mathbb{E}}\left[\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right]\\ &= \frac{1}{n}np_\varepsilon\\ &= p_\varepsilon. \end{aligned} \tag{3.1.0.8}$$

We can also compute the variance of $\tilde{p}_\varepsilon$, again, with respect to

$$\begin{aligned}\widetilde{Var}(\tilde{p}_\varepsilon) &= \widetilde{Var}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)\\ &= \frac{1}{n^2}\widetilde{Var}\left(\sum_{i=1}^{n}\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)\\ &= \frac{1}{n^2}\sum_{i=1}^{n}\widetilde{Var}\left(\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)\\ &= \frac{1}{n^2}\sum_{i=1}^{n}\left(\tilde{\mathbb{E}}\left[\left(\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)^2\right]-\left(\tilde{\mathbb{E}}\left[\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right]\right)^2\right)\\ &= \frac{1}{n^2}\sum_{i=1}^{n}\left(\tilde{\mathbb{E}}\left[\mathbb{1}_{\{\tau_i\leq T\}}\left(\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)^2\right]-p_\varepsilon^2\right). \end{aligned} \tag{3.1.0.9}$$

We have discussed how a good estimator is chosen via the relative error. Taking a closer look at this quantity, it is minimized if the variance of the estimator is minimized. Reviewing the last equation and using the fact that we are working with a random sample, we may draw conclusions about the critical points of (3.1.0.9) by studying closely a single summand $\tilde{\mathbb{E}}[\mathbb{1}_{\{\tau_i\leq T\}}(\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i))^2]$.

Let us develop an important concept for importance sampling. Following the explanation of (Dupuis and Wang, 2007)and (Dupuis et al., 2012) we will suppose that our probability of interest p_ε satisfies the following large deviation principle

$$\lim_{\varepsilon\to 0}\varepsilon\ln p_\varepsilon = -I,$$

where I is called the *large deviation rate*. Combining this with Jensen's inequality

$$\tilde{\mathbb{E}}\left[\left(\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)^2\right]\geq\left(\tilde{\mathbb{E}}\left[\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right]\right)^2=p_\varepsilon^2$$

we get

$$\limsup_{\varepsilon\to 0}-\varepsilon\ln\tilde{\mathbb{E}}\left[\left(\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}\left(X_i\right)\right)^2\right]\leq 2\limsup_{\varepsilon\to 0}\varepsilon\ln p_\varepsilon=2I.$$

We denote an estimator as *asymptotically optimal* (or we say that its relative error is *logarithmically efficient*) if the following upper bound is achieved

$$\limsup_{\varepsilon\to 0}-\varepsilon\ln\tilde{\mathbb{E}}\left[\left(\mathbb{1}_{\{\tau_i\leq T\}}\frac{d\mathbb{P}}{d\tilde{\mathbb{P}}}(X_i)\right)^2\right]\geq 2I.$$

In such case we call $2I$ the *optimal decay rate.* It can be shown (see (Dupuis et al., 2015)) that I can be represented with the variational problem

$$I=\inf_{\psi\in AC[t,T]}\left\{\int_t^T L(\psi(s),\dot{\psi}(s))+g(\psi(T)),\psi(t)=x,\psi(T)\in\partial\Omega\right\}$$

where $L(x,v)=\sup_p\{pv-H(x,p)\}$ is denoted as the *local rate function.* The function $H(x,p)$ stands for the *Hamiltonian*, which will be specified later. Given that I is the value function of a variational problem, it is the only viscosity solution to the Hamilton-Jacobi equation

$$\begin{cases}V_t(t,x)-H(x,-DV(t,x))=0, & (t,x)\in[0,T)\times\Omega\\ V(t,x)=g(x), & (t,x)\in[0,T]\times\partial\Omega\end{cases}\tag{3.1.0.10}$$

where H is the Hamiltonian, which is given by the Fenchel-Legendre transform of the local rate function, i.e., $H(x,p)=\sup_v[\langle p,v\rangle-L(x,v)]$.

Consider now the Mane potential, which is given by

$$S^c(x,y)=\inf_{\psi,t}\left\{\int_0^t c+L(\psi(s),\dot{\psi}(s))ds,\psi(0)=x,\psi(t)=y\right\}.\tag{3.1.0.11}$$

From (Djehiche et al., 2014), $y\mapsto S^c(x,y)$ is a viscosity solution to the homogeneous Hamilton Jacobi equation $H(y,DS(y))=c$ for every $y\in(a,b), y\neq x$. The *Mane critical value* is the infimum real value for which said stationary Hamilton-Jacobi equation admits a global viscosity subsolution.

Each continuous viscosity solution to (3.1.0.10) admits a min-max representation given by:

$$V(t,x) = \inf_{x\in\partial\Omega} \sup_{c>c_H} \{g(y) + S^c(x,y) - c(T-t)\}. \tag{3.1.0.12}$$

where $S^c(x,y)$ is the Mane potential and c_H is the Mane critical value, which can also be computed as:

$$c_H = \sup_x \inf_p H(x,p). \tag{3.1.0.13}$$

Our priority now is to explore some expressions for the Hamiltonian. In (Djehiche et al., 2014) we can find that the formula for the such functional corresponding to an Itô process with drift and diffusion coefficients $b(x)$ and $\sqrt{\varepsilon}\sigma(x)$ respectively is

$$H(x,p) = b(x)p + \frac{\sigma^2(x)p^2}{2} \tag{3.1.0.14}$$

and it can be shown that the Hamiltonian of a Levy process with increments of size $\gamma(X(t-),Y(t))$ and intensity $\lambda_s\nu(dy)ds$ is

$$H(x,p) = \int_{\mathcal{M}} e^{p\gamma(x,y)} - 1 - p\gamma(x,y)\nu(dy). \tag{3.1.0.15}$$

It is important to note that (3.0.0.1) has an Itô process part and a jump part corresponding to a Levy process. Since the Hamiltonian is additive in terms of small noise contributions, the Hamiltonian associated with the equation we are working with is given by

$$H(x,p) = b(x)p + \frac{\sigma^2(x)p^2}{2} + \int_{\mathcal{M}} e^{p\gamma(x,y)} - 1 - p\gamma(x,y)\nu(dy). \tag{3.1.0.16}$$

If our goal is to solve for p in $H(x,p) = c$ then we need to specify more details about $\nu(dy)$. We will describe how to do it for an expression like (3.1.0.16) for a process having finitely many jumps. Let us show some examples in advance to illustrate this idea.

Example 3.1.1. Let us start with the simplest case, the one of an Itô process with drift and diffusion coefficients $b(x)$ and $\varepsilon\sigma(x)$ respectively, whose Hamiltonian is (3.1.0.14). Thus we need to solve the equation

$$b(x)p + \frac{\sigma(x)^2 p^2}{2} = c$$
$$\frac{\sigma(x)^2}{2}p^2 + b(x)p - c = 0, \tag{3.1.0.17}$$

which is clearly a quadratic equation in p, having the following two roots:

$$\begin{aligned} p &= \frac{-b(x) \pm \sqrt{b(x)^2 - 4(\frac{\sigma(x)^2}{2})(-c)}}{2\frac{\sigma(x)^2}{2}} \\ &= \frac{-b(x) \pm \sqrt{b(x)^2 + 2c\sigma(x)^2}}{\sigma(x)^2}. \end{aligned} \tag{3.1.0.18}$$

□

Example 3.1.2. The Hamiltonian for a birth-and-death process with birth rate $\lambda(x)$ and death rate $\mu(x)$, according to (Djehiche et al., 2014), is

$$H(x,p) = \mu(e^{-p} - 1) + \lambda(x)(e^p - 1).$$

And in this case we need to solve for p in

$$\mu(x)(e^{-p} - 1) + \lambda(x)(e^p - 1) = c,$$

which we can rewrite in terms of $q = e^p$ as follows

$$\mu(x)\left(\frac{1}{q} - 1\right) + \lambda(x)\,(q - 1) = c.$$

If we multiply both sides of the equation by q we get

$$\mu(x)(1 - q) + \lambda(x)(q^2 - q) = qc.$$

Regrouping terms we can get another quadratic equation in q:

$$\mu(x) - \mu(x)q + \lambda(x)q^2 - \lambda q - cq = 0$$
$$\mu(x)q^2 - (\lambda(x) + \mu(x) + c)q + \mu = 0 \tag{3.1.0.19}$$

which has the two solutions

$$q = \frac{\lambda(x) + \mu(x) + c \pm \sqrt{(\lambda(x) + \mu(x) + c)^2 - 4\lambda(x)\mu(x)}}{2\lambda(x)},$$

and we can re-write it back in terms of p

$$p = ln\left(\frac{\lambda(x) + \mu(x) + c \pm \sqrt{(\lambda(x) + \mu(x) + c)^2 - 4\lambda(x)\mu(x)}}{2\lambda(x)}\right).$$

□

Example 3.1.3. *(Example 1.5.1 continued).* If the jump part of our jump-diffusion is a Levy process which can take finitely many jump values, say K, the jump size function becomes $\gamma(x, y) := \gamma_k(x)$ (for $= 1, ..., K$) and the Hamiltonian (3.1.0.15) becomes

$$H(x, p) = \sum_{k=1}^{K} \left(e^{p\gamma_k(x)} - 1 - p\gamma_k(x)\right).$$

Then the Hamiltonian of (3.0.0.1) becomes:

$$H(x, p) = b(x)p + \frac{\sigma^2(x)p^2}{2} + \sum_{k=1}^{K} \left(e^{p\gamma_k(x)} - 1 - p\gamma_k(x)\right). \tag{3.1.0.20}$$

Likewise we had previously established how the jump size function for a birth-and-death process is $\gamma_1(x) = 1$ for the births and $\gamma_2(x) = -1$ for the deaths. Let us refer in a similar way to their intensities as $\Lambda_1(x) := \mu(x)$ for the births and $\Lambda_2(x) := \lambda(x)$ for the deaths. Hence a Hamiltonian for such process is given by

$$H(x, p) = \mu(x)(e^{-p} - 1) + \lambda(x)(e^{p} - 1).$$

Then if the jump part of (3.0.0.1) is a birth-and-death process, its Hamiltonian is

$$H(x, p) = b(x)p + \frac{\sigma^2(x)p^2}{2} + \mu(x)(e^{-p} - 1) + \lambda(x)(e^{p} - 1). \tag{3.1.0.21}$$

Clearly in order to solve for p in $H(x, p) = c$ we cannot apply the same procedure as we did when the presented problem was a quadratic equation or when we could get to one using some transformation of the variables. Not to mention that the level of complexity escalates when the degree of the polynomial is higher than or equal to 5, since there is not an algebraic general formula

for the zeroes of such equations (for details see (Abel, 1826)). In consequence we need to use some type of numerical method for finding the zeroes of a polynomial, e.g., the *bisection method*.

For the upcoming sequence of examples on this topic, a numerical solution for p in $H(x,p)=c$ will be needed. For reference we will call $p_{num}(x)$ that numerical solution.

□

3.1.1 Use of the Mane potential to propose a change of measure

Once an expression for p is obtained, it is important to note, as stated in (Djehiche et al., 2014) that the Mane potential $S^c(x,\cdot)$ is a primitive function of p, meaning that we only need to integrate the latter to get the former. Inside the integral with respect to a dummy variable z, the sign $\pm$ in the root obtained from the quadratic equation must be selected as $sign(z-x)$. This method of finding the Mane potential comes in handy since solving (3.1.0.11) is complicated.

(Djehiche et al., 2014) explain the high level of complexity of solving (3.1.0.10), therefore they suggest that we can base the appropriate drift alteration in

$$\Theta(t,x) = -\sigma(x)DU(t,x) \tag{3.1.0.22}$$

where U is a classical (or piecewise classical) subsolution to the Hamilton-Jacobi equation

$$\begin{cases} U_t(t,x) - \bar{H}(x,-DU(t,x)) \geq 0, & (t,x) \in [0,T) \times \Omega \\ U(T,x) \leq g(x), & x \in \partial\Omega \end{cases} \tag{3.1.0.23}$$

Another point worth noting is that if we consider the min-max representation (3.1.0.12) we can propose a family of functions being subsolutions to (3.1.0.23). Once we have found a c that maximizes (3.1.0.11), such family (see (Djehiche et al., 2014)) is described by

$$\begin{aligned} U^c(t,x) &= \inf_{y\in\partial\Omega}\{S^c(x,y) - c(T-t)\} \\ &= \min\{S^c(x,a), S^c(x,b)\} - c(T-t). \end{aligned} \tag{3.1.0.24}$$

The following theorem, taken directly from Proposition 5.1 in (Djehiche et al., 2014) will allow us to guarantee that the simulation algorithms based on U are asymptotically optimal.

Theorem 3.1.1. *A simulation algorithm is asymptotically optimal if* $U(0,x_0) = V(0,x_0)$. *A sufficient condition for this to happen is that there exists a saddle point (c,y) for the min-max representation at the initial point* $(0,x_0)$, *i.e. if:*

$$\inf_{y\in\partial\Omega}\sup_{c>c_H}\left\{S^c(x_0,y)-cT\right\} = \sup_{c>c_H}\inf_{y\in\partial\Omega}\left\{S^c(x_0,y)-cT\right\}.$$

Example 3.1.4. *(Examle 3.1.1 continued).* Consider once more an Itô process with drift coefficient $b(x)$ and diffusion coefficient $\sqrt{\varepsilon}\sigma(x)$. Its Mane potential is:

$$S^c(x,y) = \int_x^y \frac{-b(z)+sign(z-x)\sqrt{b(z)^2+2c\sigma(z)^2}}{\sigma(z)^2}dz.$$

Recall that p is the solution to a quadratic equation and it takes into consideration both signs of the square root. Proposition 2.1 in (Djehiche et al., 2014) helps us decide between both options using $sign(z-x)$, given the fact that the Mane potential is the maximal of all subolutions to the homogeneous Hamilton Jacobi equation $H(y,DS(y)) = c$ that vanish at x.

It follows from (3.1.0.24) and (3.1.0.22) that given an optimal choice of c and a starting point x_0, an ideal change of measure for this process is given by:

$$\Theta(t,x) = \frac{-b(x)+sign(x-x_0)\sqrt{b(x)^2+2c\sigma(x)^2}}{\sigma(x)}. \tag{3.1.1.1}$$

□

Example 3.1.5. *(Example 3.1.2 continued).* On the other hand, if we are working with a birth-and-death process with birth rate $\lambda(x)$ and death rate $\mu(x)$, its Mane potential is

$$S^c(x,y) = \int_x^y ln\left(\frac{\lambda(z)+\mu(z)+c+sign(z-x)\sqrt{(\lambda(z)+\mu(z)+c)^2-4\lambda(z)\mu(z)}}{2\lambda(z)}\right)dz.$$

We use the same argument as in Example (3.1.4) to incorporate $sign(z-x)$ in the integrand.

It can be shown (see Section 5.3 from (Djehiche et al., 2014)) that given an optimal choice of c a good sampling measure $\mathbb{P}$ would be one based on the following modified rates:

$$\begin{aligned}\tilde{\lambda}(x) &= \lambda(x)\phi(x)\\ \tilde{\mu}(x) &= \mu(x)/\phi(x)\end{aligned} \tag{3.1.1.2}$$

where

$$\phi(x) = \frac{c + \lambda(x) + \mu(x)}{2\lambda(x)} + sign(S^c(x,a) - S^c(x,b))\sqrt{(\frac{c + \lambda(x) + \mu(x)}{2\lambda(x)})^2 - \frac{\mu(x)}{\lambda(x)}}. \tag{3.1.1.3}$$

□

Example 3.1.6. *(Example 3.1.3 continued).* The Mane potential of a jump-diffusion whose jump part is driven by a process with finitely many jumps and the one of a jump-diffusion whose jump part is driven by a birth-and-death process are very similar in structure, that is represented by

$$S^c(x,y) = \int_x^y p_{num}(z)dz. \tag{3.1.1.4}$$

Then we may use the fundamental theorem of calculus to obtain an expression for $DU^c(x,y)$ as follows:

$$DU^c(t,x) = -p_{num}(x). \tag{3.1.1.5}$$

And using (3.1.0.22) we get

$$\Theta(t,x) = \sigma(x)p_{num}(x). \tag{3.1.1.6}$$

□

Let us elaborate on how one could represent a jump-diffusion like (3.0.0.2) under a new sampling measure. Considering (1.5.1.5) let $\psi_t = 1$ and $h_t(y) = 1$. Then the derivative of change of measure is

$$L(t) = \exp\left\{\frac{1}{\sqrt{\varepsilon}}\int_0^t \Theta(t, X(s))dW(s) - \frac{1}{2\varepsilon}\int_0^t \Theta(s, X(s))^2 ds\right\} \tag{3.1.1.7}$$

which can be presented in the following differential form using Itô's formula:

$$dL(t) = \frac{L(t)\Theta(t, X(t))}{\sqrt{\varepsilon}}dW(t). \tag{3.1.1.8}$$

Given that under the new sampling measure

$$\tilde{W}(t) = W(t) - \frac{1}{\sqrt{\varepsilon}} \int_0^t \Theta(s, X(s))ds$$

is a $\tilde{\mathbb{P}}-$Brownian Motion and considering that typically we write $d\tilde{W}(t) = dW(t) - \frac{1}{\sqrt{\varepsilon}}\Theta(t, X(t))dt$, under the updated measure the jump diffusion is

$$\begin{aligned}
dX(t) &= b\left(X(t)\right)dt + \sqrt{\varepsilon}\sigma(X(t))\left(d\tilde{W}(t) + \frac{1}{\sqrt{\varepsilon}}\Theta(t, X(t))dt\right) + \int_{\mathcal{M}} \varepsilon\gamma(X(t-), Y(t))dN(t) \\
&= \left(b\left(X(t)\right) + \frac{\sqrt{\varepsilon}\sigma(X(t))}{\sqrt{\varepsilon}}\Theta(t, X(t))\right)dt + \sqrt{\varepsilon}\sigma(X(t))d\tilde{W}(t) + \int_{\mathcal{M}} \varepsilon\gamma(X(t-), Y(t))dN(t) \\
&= \left(b(X(t)) + \sigma(X(t))\Theta(t, X(t))\right)dt + \sqrt{\varepsilon}\sigma(X(t))d\tilde{W}(t) + \int_{\mathcal{M}} \varepsilon\gamma(X(t-), Y(t))dN(t) \\
&= \tilde{b}(X(t))dt + \sqrt{\varepsilon}\sigma(X(t))d\tilde{W}(t) + \int_{\mathcal{M}} \varepsilon\gamma(X(t-), Y(t))dN(t) \qquad (3.1.1.9)
\end{aligned}$$

where $\tilde{b} = b(X(t)) + \sigma(X(t))\Theta(t, X(t))$.

This procedure to generate a trajectory of $X(T)$ is very similar to the one developed in Algorithm 2. We only need to choose the drift coefficient accordingly to the one proposed by the change of measure. Moreover, the diffusion and jump size coefficients are modified as well since they are controlled by the parameter ε. Finally, recalling that when the jump-diffusion is constant (between jump times) it can be approximated by the Euler scheme, the derivative of change of measure $L(t)$ (which is a solution to (3.1.1.8)) will be updated in parallel using the same discretization scheme. The detailed outline of how to generate a single trajectory of $X(T)$ is presented in Algorithm 3.

Algorithm 3 Giesecke/Teng/Wei approximation to a jump diffusion applied to the technique proposed by Djehiche, et.al.

Set T, decide N_steps

Set time step $h = T/N_{steps}$

Choose c that maximizes the min-max representation (3.1.0.24)

Initialize $i = j = n = s = 0$

Set $Z^h(s) = x_0$

Set $A^h(s) = 0$

Set $E_n = random(exp(mean1))$

while $s < T$ **do**

    Compute $A^h_{temporary} = A^h(s) + \Lambda(Z^h(s))[(i+1)h - s]$

    if $A^h_{temporary} \geq E_n$, i.e. there is a jump in the interval $[s, (i+1)h]$ **then**

        Compute $\tau^h_n = s + \frac{E_n - A^h(s)}{\Lambda(Z^h(s))}$

        Compute $Z^h(\tau^h_n -) = Z^h(s) + \tilde{b}(Z^h(s))(\tau^h_n - s) + \sqrt{\varepsilon}\sigma(Z^h(s))\sqrt{\tau^h_n - s} * random(N(0,1))$

        Compute $L^h(\tau^h_n -) = L^h(s) + \left(L^h(s)\Theta(Z^h(\tau^h_n -)\right)/\sqrt{\varepsilon}) * \sqrt{\tau^h_n - s} * random(N(0,1))$

        Compute $Z^h(\tau^h_n) = Z^h(\tau^h_n -) + \varepsilon\gamma(Z^h(\tau^h_n -), random(\nu(dy)))$

        Accumulate time $s = \tau^h_n$

        Update $A^h(s) = E$

        Update $n = n + 1$

        Update $E_n = E_n + random(exp(mean1))$

    end if

    if There is no jump in the interval $[s, (i+1)h]$ **then**

        Compute $Z^h((i+1)h) = Z^h(s) + b(Z^h(s))((i+1)h - s) + \sqrt{\varepsilon}\sigma(Z^h(s))\sqrt{(i+1)h - s} * random(N(0,1))$

        Compute $L^h((i+1)h) = L^h(s) + \left(L^h(s)\Theta(Z^h(s)\right)/\sqrt{\varepsilon}) * \sqrt{(i+1)h - s} * random(N(0,1))$

        Update $s = (i+1)h$

        Update $A^h(s) = A^h_{temporary}$

        Update $i = i + 1$

    end if

end while

3.2 Analogy with an Itô process

Along with the simulation of a rare event of a specific type of a jump-diffusion, a number of numerical approximations need to be performed. In order to reinforce these ideas, they will be replicated for an Itô process, where a lot of explicit formulas are available. For this section we must consider that we are dealing with a diffusion without any jumps, i.e., an Itô process of the form $dX(t) = b(X(t))dt + \sqrt{\varepsilon}\sigma X(t)dW(t)$.

3.2.1 Numerical approximation to the Mane potential

As it has been described before, one of the main challenges is to overcome the fact that it results complicated to obtain an analytical expression for p in $H(x, p) = c$ in order to obtain the Mane potential of a jump diffusion. However, such problem can be addressed by numerically finding the zeroes of the equation $f(p) = H(x, p) - c$ with satisfactory results.

In the case of an Itô process we have an explicit, analytical formula for p given by (3.1.0.18). Having established routines for the drift and diffusion coefficients, the value of p can be coded in `R` as

```
p.explicit<-function(z,c_){
  (-b(z)+sign(z-x0)*sqrt(b(z)^2+2*c_*sigma_fn(z)^2))/(sigma_fn(z)^2)
}
```

Finding zeroes of a function is not an uncommon task. In this case we may use something as simple as the *bisection method*. If we have prior knowledge of the whereabouts of a zero and we can enclose it in an interval, this procedure repeatedly bisects said interval bounding the zero until a desired level of accuracy is reached (for more details on the bisection method see (Brent, 2013)). A classical function which performs said procedure in R is `uniroot`. The routine executes this method with the restriction that the function of interest evaluated in both ends of the interval needs to have opposite signs. This inconvenience can be circumvent by using the function `uniroot.all` instead, since it divides the interval into a specified number of sub intervals and then performs the bisection method on each one.

Paying closer attention to (3.1.0.24) we will need to compute the Mane potential in two different cases: for values $z \in [a, x_0]$ and for values $z \in [x_0, b]$. `p.explicit` already makes that distinction

using `sign(z-x0)`. However, we need to make a special distinction for the numerical version of p. For such case we can define a function which will numerically find the zeroes of $f(p) = H(z,p) - c$ for $z \in [x_0, b]$

```
p.fn.max<-function(z,c_){
  H.xpc<-function(p,xx){
    b(xx)*p+(0.5*sigma_fn(xx)^2)*p^2-c_
  }
  pp<-sapply(X = z,
             FUN=function(x) uniroot.all(f = H.xpc,interval = c(-10,10),xx=x))
  if (is.null(dim(pp))){
    return(pp)
  }
  else{
    return(apply(X = pp,MARGIN = 2,FUN = max))
  }
}
```

with its analogous function which will numerically find the zeroes of $f(p)$ in $z \in [a, x_0]$.

```
p.fn.minnn<-function(z,c_){
  H.xpc<-function(p,xx){
    b(xx)*p+(0.5*sigma_fn(xx)^2)*p^2-c_
  }
  pp<-sapply(X = z,
             FUN=function(x) uniroot.all(f = H.xpc,interval = c(-10,10),xx=x))
  if (is.null(dim(pp))){
    return(pp)
  }
  else{
    return(apply(X = pp,MARGIN = 2,FUN = min))
  }
}
```

A reasonable way to show that both the algebraic function and its numerical copy produce the same output would be by showing that the area between both curves is virtually zero. Let us then define a function that calculates the difference between both quantities for $z \in [x_0, b]$, which we will then integrate in said interval.

Numerical integration methods begin with producing a partition of the interval of interest and then reducing the integral to a simplified sum. There is a number of methods that can be applied to this situation, e.g., the rectangle rule, simpson rule, trapezoidal rule. In the majority of these methods the elements of the partition are equidistant. However, there is a relatively simple function in R which performs *adaptative quadrature*, which refines the step sizes where the function changes rapidly (see (Piessens et al., 2012)).

```
area.max<-function(z,c_){
  p.thesis(z,c_)-p.fn.max(z,c_)
}
integrate(f = area,lower = x0,upper = Omega_b,c_=2.84)
```

With a result of `7.24975e-05 with absolute error < 8e-19` which is close to zero, we can conclude that both the analytical and the numerical version of p produce the same output for

$z \in [x_0, b]$.

Analogously, let us define a function that calculates the difference between both functions for $z \in [a, x_0]$, which we will then integrate in said interval:

```
area.min<-function(z,c_){
  p.thesis(z,c_)-p.fn.min(z,c_)
}


integrate(f = area.min,lower = Omega_a,upper = x0,c_=2.84)
```

with a result of `-3.46059e-05 with absolute error < 3.8e-19` which is close to zero, we can conclude that both the analytical and the numerical version of p produce the same output for $z \in [a, x_0]$.

3.2.2 Numerical approximation to a solution for p in H(x,p)=c

As stated previously, to any root-finding algorithm we must provide an initial value of the roots. The accuracy of such educated guess is vital for the convergence of the algorithm. Up to this moment in the functions described in Section 3.2.1 we have overlooked such initial guess.

In Figure 3.1 we can see some plots of $f(p) = H(z, p) - c$, made for $z = 90, 95, 100, 105, 110$ respectively (all of these values corresponding to elements within Ω), it is easy to see that the roots may be found in close proximity to 0.

3.2.3 Computation of the Mane critical value

When verifying if an algorithm is asymptotically optimal we need to find a saddle point (c, y) in the min-max representation at the initial point $(0, x_0)$ as described in Theorem 3.1.1. It is useful then to know the value of c_H, the Mane critical value, which can be computed using (3.1.0.13). We can start with the fact that for an Itô process the Hamiltonian is given by 3.1.0.14. Obtaining the first derivative of H with respect to p, making it equal to zero and solving for p to find critical points results in:

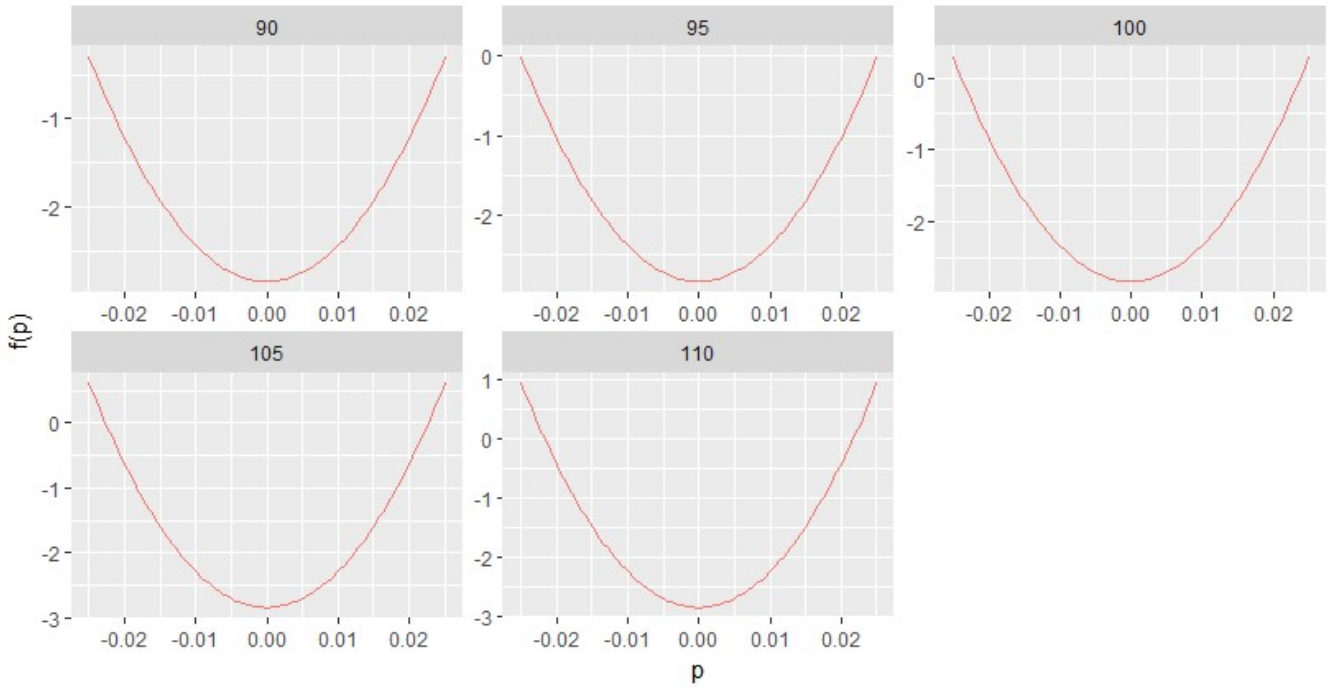

Figure 3.1: Plots of $f(p) = H(z,p) - c$ for the case of an Itô process, made for $z = 90, 95, 100, 105, 110$ respectively

$$\begin{aligned} \frac{\partial H}{\partial p} &= b(x) + \sigma(x)^2 p \\ \sigma(x)^2 p &= -b(x) \\ p &= \frac{-b(x)}{\sigma(x)^2}. \end{aligned} \tag{3.2.0.10}$$

After that, we need to find the second derivative of H with respect to p to verify the nature of the critical point $p = \frac{-b(x)}{\sigma(x)^2}$:

$$\frac{\partial^2 H}{\partial p^2} = \sigma(x)^2.$$

Given the fact that $\sigma^2(x) > 0$, we can conclude that located at $p = \frac{-b(x)}{\sigma(x)^2}$ there is a local minimum of H equal to

$$H\left(x, \frac{-b(x)}{\sigma(x)^2}\right) = b(x)\left(\frac{-b(x)}{\sigma(x)^2}\right) + \frac{\sigma^2}{2}\left(\frac{-b(x)}{\sigma(x)^2}\right)^2$$
$$= -\frac{b(x)^2}{\sigma(x)^2} + \frac{b(x)^2}{2\sigma(x)^2}$$
$$= -\frac{b(x)^2}{2}. \tag{3.2.0.11}$$

Only one aspect of (3.1.0.13) is missing, the supremum part. Hence we can conclude that the Mane critical value for an Itô process is

$$c_H = \sup_x \left\{-\frac{b(x)^2}{2}\right\}. \tag{3.2.0.12}$$

It is clear that the Mane critical value can be obtained analytically when H is simple enough, but for more complicated Hamiltonians the solution might not be that straightforward, which is why we can easily program a function in `R` to find c_H for us. To a clarifying end let us program the following two simple functions: the first one to compute the value of the Hamiltonian, and the second one to perform the minimization over all the values of p of H for a fixed x, and immediately performing the maximization over all the values of x.

```
H.px<-function(p,x){
  b(x)*p+(0.5*sigma_fn(x)^2)*p^2
}


c_H<-function(){
  inside<-function(x){
    optimize(f=H.px,interval = c(-10000,10000),maximum = FALSE,x=x)\$objective
  }
  optimize(f=inside,interval = c(-10000,10000),maximum = TRUE)\$objective
}
```

Example 3.2.1. Consider an Itô process $X(t)$ being a solution of $dX(t) = b(X(t))dt + \varepsilon\sigma X(t)dW(t)$, where $b(x) = \sigma(x) = 1$. We can conclude that $c_H = -1/2$ using (3.2.0.12). And in order to verify that with the function we just proposed in `R`, we need to define functions for the drift and diffusion coefficients as follows:

```
b<-function(x){
  1
}

sigma_fn<-function(x){
  1
}
```

Which yields a Mane critical value consistent to the $-1/2$ previously obtained.

```
> c_H()
[1] -0.5
```

□

Example 3.2.2. Consider another Itô prcess $X(t)$ being a solution of $dX(t) = b(X(t))dt + \varepsilon\sigma X(t)dW(t)$, with $b(x) = -2x(x^2-1)$ and $\sigma(x) = 1$. Using (3.2.0.12) we can find the Mane critical value as follows:

$$\begin{aligned} c_H &= \sup_x \left\{ -\frac{b(x)^2}{2} \right\} \\ &= \sup_x \left\{ -\frac{(-2x(x^2-1))^2}{2} \right\} \\ &= \sup_x \left\{ \frac{-4x^2(x^2-1)^2}{2} \right\} \\ &= \sup_x \left\{ -2x^2(x^4-2x^2+1) \right\} \\ &= \sup_x \left\{ -2x^6+4x^4-2x^2 \right\}. \end{aligned} \tag{3.2.0.13}$$

Only the maximization of $-2x^6+4x^4-2x^2$ remains outstanding:

$$\begin{aligned} \frac{d(-2x^6+4x^4-2x^2)}{dx} &= -12x^5+16x^3-4x \\ -12x^5+16x^3-4x &= 0 \\ 3x^5-4x^3+x &= 0 \\ x(3x^4-4x^2+1) &= 0 \end{aligned} \tag{3.2.0.14}$$

From the equation above, we can infer that $x = 0$ is a critical point thus we need to verify its nature as follows:

$$\frac{d^2(-2x^6 + 4x^4 - 2x^2)}{dx^2} = -60x^4 + 48x - 4.$$

Evaluating the second derivative in $x = 0$ gives a negative value. Hence the maximum is located at $x = 0$ and consequently $c_H = 0$. Such statement can be verified numerically with the function in `R` stated previously. Indeed, if we define the drift and diffusion coefficients as follows:

```
b<-function(x){
  -2*x*(x^2-1)
}


sigma_fn<-function(x){
  1
}
```

the result is essentially zero, corresponding with the result obtained analytically.

```
> c_H()
[1] -4.135903e-25
```

□

3.3 Simulation of a birth-and-death process

Let us exemplify the ideas we have explained so far with a simple example of a birth-and-death process. We used the settings in Section 5.3 of (Djehiche et al., 2014).

The process evolves as follows: it jumps up by $1/N$ with rate $N\lambda(x)$ and it jumps down by $1/N$ with rate $N\mu(x)$, where

$$\begin{aligned} \lambda(x) &= 3x(1-x) \\ \mu(x) &= . \end{aligned} \tag{3.3.0.15}$$

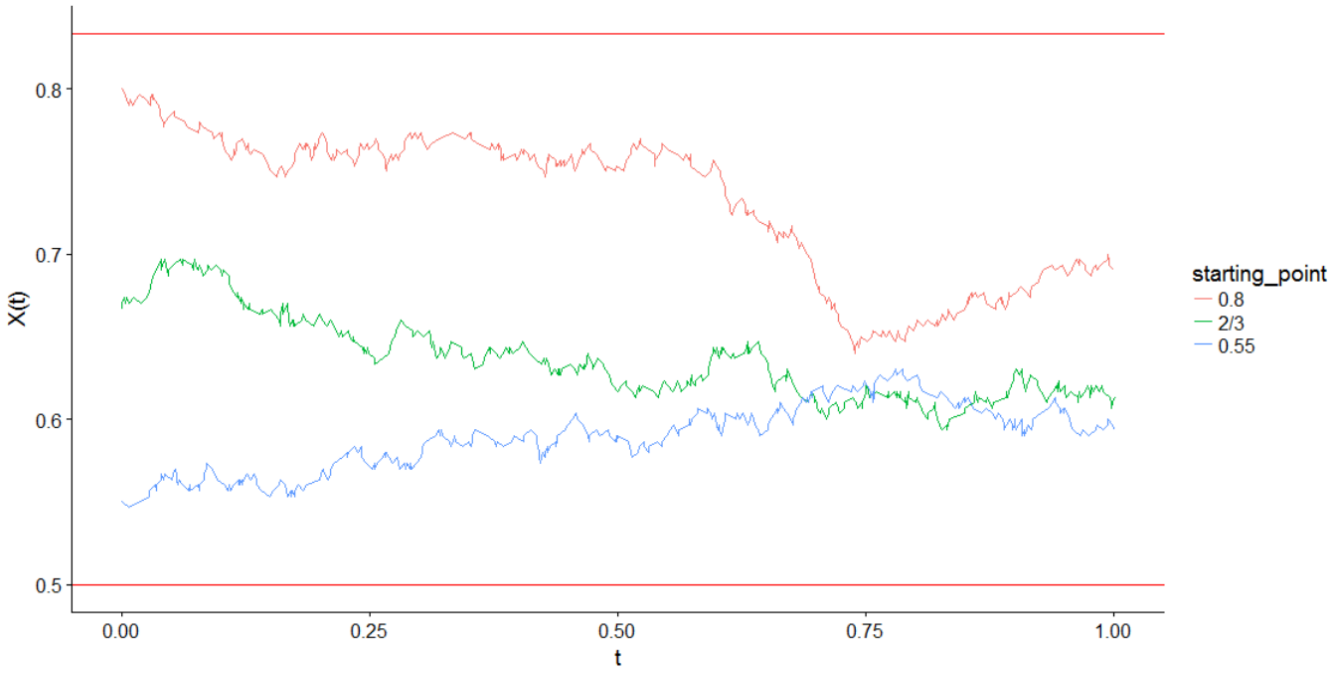

Figure 3.2: Sample paths of a birth-and-death process with different starting points

Figure 3.2 shows three different trajectories for the process, starting at $X(0) = 4/5$, $2/3$ and $11/20$. We are interested in analysing the probability that the process $X(t)$ leaves the domain $\Omega = (1/2, 5/6)$ before time $T = 1$, i.e., $p := \mathbb{P}[\tau < 1]$. Therefore it is of great utility to modify the measure of the process to make it exit Ω faster since, as it can be seen in Figure 3.2, relying on the current measure for calculating a Monte Carlo estimator for p of the form (2.1.0.1) would lead to almost none of the particles hitting the relevant event $\{\tau < 1\}$.

3.3.1 Numerical results

For each starting point $X(0)$ two thousand simulations were performed, half of them under the original rates (3.3.0.15) and half of them by modifying the rates as in Example 3.1.5. Table 3.1 shows the results of said simulations, from which the following conclusions can be drawn: very few particles hit the relevant region before the change of measure, i.e., respectively 2, 0 and 4 simulations meet the event $\{\tau < T\}$. Said estimators also have a high variance. On the contrary it can be seen that the change of measure decreases significantly the variance of the estimators.

X(0)	4/5	2/3	11/20
$\hat{p}$	0.002	0	0.004
Var($\hat{p}$)	0.001997998	0	0.003987988
Relevant particles	2	0	4
$\tilde{p}$	0.001710637	9.33E-06	0.00467098
Var($\tilde{p}$)	1.76E-07	3.21E-10	1.95E-06
Relevant particles	998	227	990

Table 3.1: Descriptive statistics of the sampling distribution of $\tau_{1,2}$ for different N

3.4 Simulation of a combination of a geometric Brownian motion and jumps

Let us verify the ideas we have developed with an example where we will estimate the probability that the process $X(t)$ leaves the domain Ω before time T, i.e., $\mathbb{P}[\tau_\varepsilon < T]$ which we had previously referred to as p_ε. Consider a process $X(t)$ being the solution to the stochastic differential equation

$$dX(t) = \mu X(t)dt + \sqrt{\varepsilon}\sigma X(t)dW(t) + \varepsilon \int_{\mathcal{M}} yp(dt, dy). \tag{3.4.0.16}$$

Let us identify the components of this equation: the drift coefficient $b(x) = \mu x$, the diffusion coefficient $\sigma(x) = \sigma x$ and the counting measure $p(dt, dy)$ having intensity measure $\Lambda(X(t))\nu(dy)$ where $\Lambda(x) = \Lambda_0 + \Lambda_1 x$ and $\nu(dy)$ is a discrete probability distribution over the finite, countable set $\mathcal{M}$. More specifically, each element $y_k \in \mathcal{M}$ has probability $p(y_k)$ of being chosen. As it has been done before, relabelling the random counting measure as N(t), (3.4.0.16) may be rewritten as

$$dX(t) = \mu X(t)dt + \sqrt{\varepsilon}\sigma X(t)dW(t) + \varepsilon Y(t)dN(t). \tag{3.4.0.17}$$

Since $\mathcal{M}$ is finite countable (say it has K elements) we can use theorem 1.2.1 to re-write (3.4.0.17) as a particular case of (3.0.0.2).

$$dX(t) = \mu X(t)dt + \sqrt{\varepsilon}\sigma X(t)dW(t) + \varepsilon \sum_{k=1}^{K} y_k dN_k(t) \tag{3.4.0.18}$$

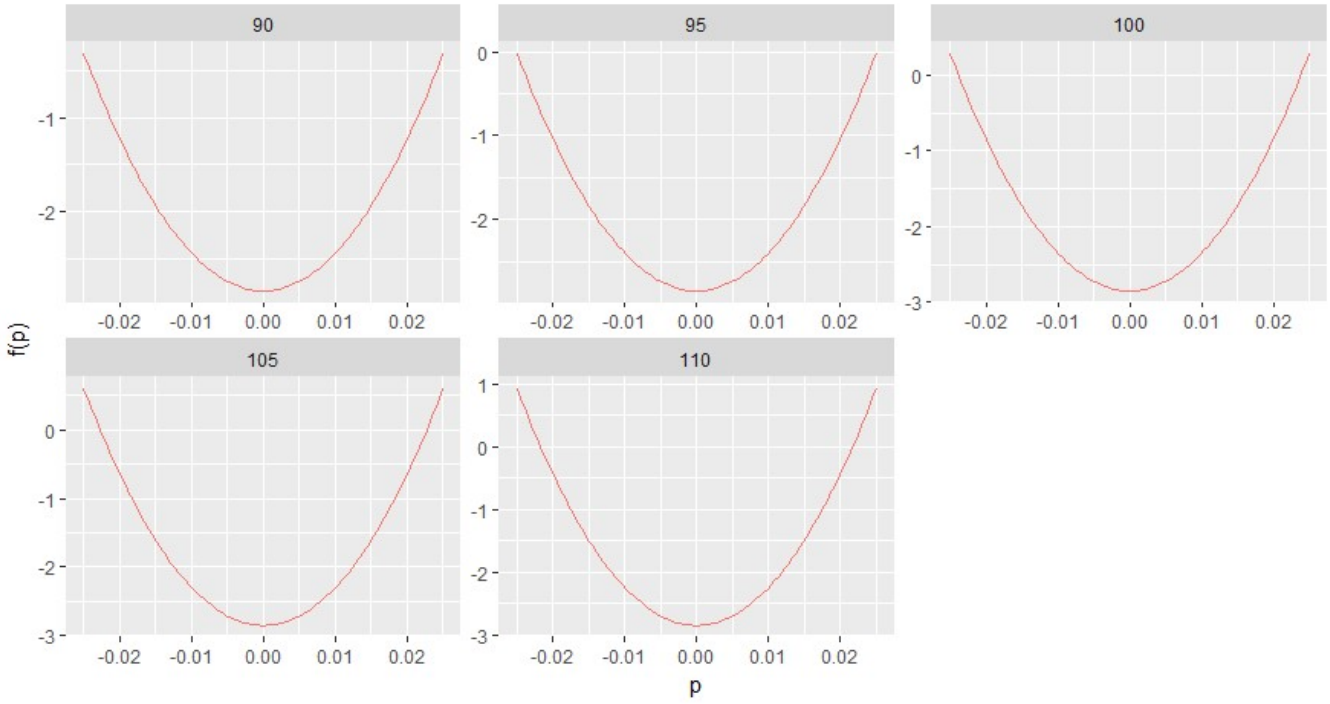

Figure 3.3: Plots of $f(p) = H(z,p) - c$ for the case of a jump-diffusion, made for $z = 90, 95, 100, 105, 110$

where each $N_k(t)$ $(k = 1, ..., K)$ is a Point Process with intensity $\Lambda(X(t))p(y_k)$. Having different representations of a jump-diffusion is useful. We use (3.4.0.16) for identifying the elements of Algorithm (2) and (3.4.0.18) is vital to perform calculations with respect to the Hamiltonian and the Mane potential.

Consider the parameters $\Omega = (90, 110)$, $X(0) = 100$, $\mu = 0$, $\sigma = 1$, $\Lambda_0 = 5$, $\Lambda_1 = 50$ and $\nu(dy)$ is a uniform distribution among the set $\mathcal{M} = \{0.01, 0.015, 0.02, 0.025, 0.03\}$. Which means that each $N_k(t)$ $(k = 1, ..., 5)$ is a Point process with intensity $\Lambda(X(t))/5$. Note also that $(\gamma_1(x), \gamma_2(x), \gamma_3(x), \gamma_4(x), \gamma_5(x)) = (y_1, y_2, y_3, y_4, y_5) = (0.01, 0.015, 0.02, 0.025, 0.03)$.

3.4.1 Numerical results

Note that in the case of of the geometric Brownian Motion with jumps, $H(x,p)$ is given by (3.1.0.20). Analogously to section (3.2.2) we need an initial guess for the numerical approximation to p in $f(p) = H(x,p) - c$. As we can see in figure 3.3, its roots can also be found in the vicinity of 0.

As performed in section (3.2.1) we need to define the Mane potential as (3.1.1.4) which helps us

to find a c maximizing (3.1.0.24). This is a crucial part of the algorithm since the right choice of c will determine whether or not the algorithm is asymptotically optimal. For such end we will use Theorem 3.1.1 and we will show that:

$$\inf_{y\in\partial\Omega}\sup_{c>c_H}\{S^c(x_0,y)-cT\}=\sup_{c>c_H}\inf_{y\in\partial\Omega}\{S^c(x_0,y)-cT\}\,. \tag{3.4.0.19}$$

In order to find the Mane critical value c_H, we will use (3.1.0.13) and proceed analogously as in Section 3.2.3. Let us write a function in `R` that calculates the Hamiltonian (3.1.0.20) and another one that performs the optimization.

```
H.px<-function(p,x){
  exponent<-diag(x=p,
          nrow=length(p),
          ncol=length(p))%*%t(replicate(length(p),mark_set))
  b(x)*p+(0.5*sigma_fn(x)^2)*p^2+rowSums(exp(exponent)-1-exponent)
}


c_H<-function(){
  inside<-function(x){
    optimize(f=H.px,interval = c(-10000,10000),maximum = FALSE,x=x)$objective
  }
  optimize(f=inside,interval = c(-10000,10000),maximum = TRUE)$objective
}
```

which yields a Mane critical value of

```
> c_H()
[1] 1.017209e-06
```

Starting with the left hand side of (3.4.0.19), we can program the following function for a fixed $y\in\partial\Omega$

$$y\longmapsto\sup_{c>c_H}\{S^c(x_0,y)-cT\}$$

which for code purposes is called max_min:

```
max_min<-function(y,cap_t,low_t){
  max.part<-function(c_,cap_t,low_t,y){
    mane(c_,x0,y)\$value-c_*(cap_t-low_t)
  }
    max.value<-optimize(max.part,interval = c(c_H(),100000),maximum=TRUE,
                        cap_t=cap_t,low_t=low_t,y=y)}
```

Then we can find the infimum of max_min for $y \in \partial\Omega$:

```
min.part<-min(max_min(y = Omega_a,cap_t = T_vec,low_t = low_t)\$objective,
              max_min(y = Omega_b,cap_t = T_vec,low_t = low_t)\$objective)
```

which yields an infimum value of

```
> min.part
[1] 0.1134783
```

Simultaneously for the right hand side of (3.4.0.19) we define a function that for each $c > 0$

$$c \longmapsto \inf_{y \in \partial\Omega} \{S^c(x_0, y) - cT\}$$

which for coding purposes is called min_max

```
min_max<-function(c_,cap_t,low_t,a,b_){
  min(mane(c_,x0,a)\$value,mane(c_,x0,b_)\$value)-c_*(cap_t-low_t)
}
```

Then we can find the supremum of min_max for all $c > c_H$.

```
max.part<-optimize(min_max,interval = c(c_H(),100000),maximum =TRUE ,
                   cap_t=cap_T, low_t=low_t,a=Omega_a,b=Omega_b)\$objective
```

which yields a supremum value of

```
> max.part
[1] 0.1134783
```

Since the left hand side of 3.4.0.19 is equal to its right hand side, the point $(c, y) = (2.85153, 110)$ is a saddle point and the algorithm we will design using c will be asymptotically optimal.

Now we can define (3.1.1.6) and then draw a random sample of size n of $X(T)$ being a solution of (3.1.1.9). Hence we can estimate p_ε with $\tilde{p_\varepsilon}$ defined in (3.1.0.7), i.e., we will re-weight the sample mean using $L(t)$ as in (3.1.1.7).

Four different values of ε were chosen. For each one of them 1000 simulations were performed with 100 time discretization steps. The results are summarized in tables (3.2)-(3.5).

$\tilde{p}_{0.05} \approx 0.06803$	R
Before change of measure	0.01823
After change of measure	0.00370

Table 3.2: Estimate of $p_{0.05}$ and its relative error

$\tilde{p}_{0.06} \approx 0.06788$	R
Before change of measure	0.01191
After change of measure	0.00370

Table 3.3: Estimate of $p_{0.06}$ and its relative error

$\tilde{p_{0.07}} \approx 0.11398$	R
Before change of measure	0.01113
After change of measure	0.00278

Table 3.4: Estimate of $p_{0.07}$ and its relative error

$\tilde{p}_{0.08} \approx 0.19108$	R
Before change of measure	0.00810
After change of measure	0.00205

Table 3.5: Estimate of $p_{0.08}$ and its relative error

These results have some properties worth noting. Consistently among every case the relative error is lower when we apply the change of measure than when the simulations are performed without using importance sampling. The detailed R code needed to replicate this example may be found in Appendix B.

Chapter 4

Applications to bistability of molecular species

The importance sampling algorithm we have studied can be applied in a number of different fields. One of them is the dynamics modelling of certain molecular species, e.g., a cell or a gene-expression system. To this setting, cell-to-cell variability can be attributed to two different sources:

- Noise happening within each cell.
- Fluctuations in molecular composition due to cell division, splitting and resampling.

Both types of noise are interconnected vastly and have very direct applications in systems biology. For instance, it is of great interest to determine the role of the noise in creating phenotypic heterogeneity. Another concept of particular interest is *bistability*: the alternation between two different stable states for a molecular species. Let us present a stochastic model for reaction dynamics.

Without loss of generality, suppose that we have a system with two chemical species, A and B. Inside the system, k different types of biochemical reactions can happen. A reaction R_i, $(i = 1, ...k)$ explaining a_i *units of* A *and* b_i *units of* B *react and produce* a_i' *units of* A *and* b_i' *units of* B can be written as:

$$\{a_i A + b_i B \to a_i' A + b_i' B\} \tag{4.0.0.1}$$

Each R_i happens according to a state dependent rate λ_i. When denoting the number of molecules of A and B with the vector $(X_A(t), X_B(t))$, it can be seen that it evolves as a Markov jump process

with jump sizes $\{(a_i' - a_i, b_i' - b_i), i = 1, ..., k\}$ occurring at rates $\lambda_i((X_A(t), X_B(t)))$. By designating N as the capacity of the system, we imply that we might be dealing with limited space or resources, hence we will consider a conservatory relationship between the two species, i.e., $X_B(t) = N - X_A(t)$ and the number of molecules of A and B is $(X_A(t), N - X_A(t))$. For convenience we can drop the subscript, i.e., $X(t) := X_A(t)$. Also, we will work with a scaled process $X_N(t) := X(t)/N$ which clearly lies in the interval $[0, 1]$.

Throughout this section we propose two approaches. (McSweeney et al., 2014) suggests generating exactly the marked point process, as well as improving computational efficiency using a diffusion approximation. However, said approximation implies some domain problems when the process is close to the boundaries of $[0, 1]$, which is why we propose a jump-diffusion approximation developed in (Leite and Williams, 2017).

4.1 Marked point process

Let us suppose for a moment that the marked point process is indeed a birth-and-death process evolving according to some state-dependent rates $r_-(x)$ and $r_+(x)$. Furthermore, let us assume that the process has two stable equilibrium points, say x_1 and x_3, and one unstable equilibrium point, say x_2. Since we are interested in modelling bistable behaviours in a molecular species we could refer without loss of generality to the average time the process takes to leave a neighbourhood of x_1, cross x_2 and arriving to a neighbourhood of x_3. We may call this $\tau_{1,2}$, the mean exit time of (x_1, x_2). Analogously $\tau_{3,2}$ is the mean exit time of (x_3, x_2). By studying $\tau_{1,3}$ and $\tau_{3,2}$ we gain insight into how much time the process spends in either of the stable equilibrium points. According to (McSweeney et al., 2014) the deviations of X_N away from said neighbourhoods are described by the large deviation rate function given by the *quasipotential*, for x_j, $(j \in \{1, 3\})$,

$$l_{j,2} = \int_{x_j}^{x_2} \ln\left(\frac{r_-(x)}{r_+(x)}\right) dx. \tag{4.1.0.2}$$

Moreover, for $j \in \{1, 3\}$ the mean transition times $\beta_{j,2} = \mathbb{E}[\tau_{j,2}]$ can be related to the quasipotential $l_{j,2}$ with the following relationship:

$$\lim_{N\to\infty} \frac{1}{N} \ln \beta_{j,2} = l_{j,2}. \tag{4.1.0.3}$$

Consider the following reaction system presented in (McSweeney et al., 2014).

$$\begin{aligned} A &\xrightarrow{\kappa_{-1}^{10}} B \\ B &\xrightarrow{\kappa_{1}^{01}} A \\ A + B &\xrightarrow{\kappa_{-1}^{11}} 2B \\ 2A + B &\xrightarrow{\kappa_{1}^{21}} 3A \end{aligned} \tag{4.1.0.4}$$

On top of the arrows we have the constants $(\kappa_{-1}^{10}, \kappa_{1}^{01}, \kappa_{-1}^{11}, \kappa_{1}^{21})$. We are interested in reflecting both types of variability in molecular concentration using a marked point process. In order to do so, the process will have two types of jumps:

- Jumps up by $1/N$ and jumps down by $1/N$ modelled by the marked point processes $M^+(t)$ and $M^-(t)$ with respective intensity rates $Nr_+(x)$ and $Nr_-(x)$ where:

$$\begin{aligned} r_+(x) &= \kappa_1^{01}(1-x) + \kappa_1^{21}x^2(1-x) \\ r_-(x) &= \kappa_{-1}^{10}x + \kappa_{-1}^{11}x(1-x). \end{aligned} \tag{4.1.0.5}$$

- Jumps by $\pm 1/N$ with equal probability modelled by the marked point process $N(t)$ with intensity rate $\Lambda(x)$. In other words, the jumps $Y(t)$ (say) take values in the mark set $\mathcal{M} = \{-1/N, 1/N\}$ with equal probabilities, i.e., $\mathbb{P}[Y(t) = -1/N] := p(1/N) = 0.5$ and $\mathbb{P}[Y(t) = 1/N] := p(-1/N) = 0.5$. The intensity $\Lambda(x)$ is described as follows

$$\Lambda(x) = \frac{1}{2}\varepsilon N^2 x(1-x). \tag{4.1.0.6}$$

Summarizing, the marked point process follows the following differential equation:

$$dX_N(t) = Y(t)dN(t) + \frac{1}{N}dM^+(t) - \frac{1}{N}dM^-(t). \tag{4.1.0.7}$$

In this setting, ε should be of order $O(1/N)$, guaranteeing that both types of jumps are equally weighted.

Note that by using Theorem 1.2.1 we can decompose $N(t)$ into two particular cases of the sub-process, say $N_1(t)$ and $N_2(t)$:

- $N_1(t)$ is a marked point process with intensity $\Lambda(x)p(1/N) = 0.5\Lambda(x)$ whose jumps are equal to $1/N$.
- $N_2(t)$ is a marked point process with intensity $\Lambda(x)p(-1/N) = 0.5\Lambda(x)$ whose jumps are equal to $-1/N$.

Hence we can combine the sub-processes as follows:

$$\begin{aligned}\frac{1}{N}N_1(t) + \frac{1}{N}M^+(t) &= \frac{1}{N}\left(N_1(t) + M^+(t)\right) \\ &:= \frac{1}{N}N_1^+(t) \end{aligned} \tag{4.1.0.8}$$

where $N_1^+(t)$ is a marked point process with intensity $\tilde{r}_+(x)$ whose jumps are equal to $1/N$. Analogously

$$\begin{aligned}\frac{1}{N}N_2(t) + \frac{1}{N}M^-(t) &= \frac{1}{N}\left(N_2(t) + M^-(t)\right) \\ &:= \frac{1}{N}N_2^-(t). \end{aligned} \tag{4.1.0.9}$$

where $N_2^-(t)$ is a marked point process with intensity $\tilde{r}_-(x)$ whose jumps are equal to $-1/N$. Note that the intensities are equal to

$$\begin{aligned}\tilde{r}_+(x) &= Nr_+(x) + \frac{1}{2}\Lambda(x) \\ \tilde{r}_-(x) &= Nr_-(x) + \frac{1}{2}\Lambda(x). \end{aligned} \tag{4.1.0.10}$$

Wrapping everything up, we can express all the sub-processes in a single marked point process by defining the marked point process $\bar{N}(t)$ as

$$\bar{N}(t) = \frac{1}{N}N_1^+(t) - \frac{1}{N}N_2^-(t) \tag{4.1.0.11}$$

whose intensity is equal to:

$$0.5\Lambda(x) + r^+(x) + 0.5\Lambda(x) + r^-(x) = \Lambda(x) + r^+(x) + r^-(x) \tag{4.1.0.12}$$

and whose jumps $Y(t)$ take values in the mark set $\mathcal{M} = \{-1/N, 1/N\}$ with the following probabilities

$$\mathbb{P}[Y(t) = 1/N] := p(1/N) = \frac{0.5\Lambda(x) + r^+(x)}{\Lambda(x) + r^+(x) + r^-(x)} \tag{4.1.0.13}$$

$$\mathbb{P}[Y(t) = -1/N] := p(-1/N) = \frac{0.5\Lambda(x) + r^-(x)}{\Lambda(x) + r^+(x) + r^-(x)}. \tag{4.1.0.14}$$

Hence (4.1.0.7) becomes

$$dX(t) = Y(t)d\bar{N}(t). \tag{4.1.0.15}$$

The choice of the constants $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21})$ plays a crucial role for the design of the process. If we want $X_N(t)$ to model a system having stable equilibria points $\{\alpha, 1-\alpha\}$ then $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21})$ may be obtained by solving the following equation:

$$-\kappa_{-1}^{10}x + \kappa_1^{01}(1-x) - \kappa_{-1}^{11}x(1-x) + \kappa_1^{21}x^2(1-x) = C(x-\alpha)(x-0.5)(x-(1-\alpha)) \tag{4.1.0.16}$$

where C is a constant which needs to be chosen so that the $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21})$ are of the same order of magnitude. Expanding the left hand side of (4.1.0.16) we get:

$$-\kappa_1^{21}x^3 + (\kappa_{-1}^{11} + \kappa_1^{21})x^2 - (\kappa_{-1}^{10} + \kappa_1^{01} + \kappa_{-1}^{11})x + \kappa_1^{01}. \tag{4.1.0.17}$$

Expanding the right hand side of (4.1.0.16) we get:

$$cx^3 - \frac{3c}{2}x^2 + c\left(\alpha - \alpha^2 + \frac{1}{2}\right)x - c\frac{\alpha(1-\alpha)}{2}. \tag{4.1.0.18}$$

Equating the coefficients of (4.1.0.17) and (4.1.0.18) gives the following system of linear equations, which can be easily solved for $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21})$:

$$\begin{aligned}
\kappa_1^{21} &= -c \\
\kappa_{-1}^{11} + \kappa_1^{21} &= -\frac{3c}{2} \\
-(\kappa_{-1}^{10} + \kappa_1^{01} + \kappa_{-1}^{11}) &= c\left(\alpha - \alpha^2 + \frac{1}{2}\right) \\
\kappa_1^{01} &= -c\frac{\alpha(1-\alpha)}{2}.
\end{aligned} \tag{4.1.0.19}$$

We can apply the algorithms developed in Chapter 2 to generate useful sample information of $\tau_{j,2}$ and $\beta_{j,2}$. On top of that by defining $p_{j,2} = \mathbb{P}[\tau_{j,2} < T]$ we can apply the algorithms from Chapter 3 to improve the accuracy and efficiency of estimates of $p_{j,2}$. We only need to define a starting point $X_N(0)$ and a domain Ω from which the process will exit faster using a change of measure; if $X_N(0) = \alpha$ then $\Omega = [0, 0.5)$ and if $X_N(0) = 1 - \alpha$ then $\Omega = (0.5, 1]$.

Example 4.1.1. We want to model a process having stable equilibria points $\{0.1, 0.9\}$, i.e., with $\alpha = 0.1$. The constants for the reactions may be obtained by letting $c = -32/3$. That way $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21}) = (36/19, 36/19, 400/19, 800/19)$. Additionally, consider $\varepsilon = 0.01$ and therefore $N = 1/100$. A sample path for this process is presented in Figure 4.1.

The values of the quasipotential are not equal for 0.1 and 0.9. Indeed $l_{1,2} = 0.06707331$ and $l_{3,2} = 0.0463878$. Therefore $l_{1,2} > l_{3,2}$, which implies $\beta_{1,2} > \beta_{3,2}$ and hence the process takes longer to leave the equilibrium $x_1 = 0.1$ than it does to leave $x_3 = 0.9$.

In order to perform inference on $\tau_{j,2}$, three thousand simulations of the marked point process were run. Half of them starting at $X_N(0) = 0.1$ and the other half starting at $X_N(0) = 0.9$. The procedure was done once with the jump rates described in (4.1.0.10), and once more modifying the jump rates with the methodology proposed in Chapter 3.

It can be seen at first sight from the sample paths in Figure 4.2 that the process leaves the domain faster after the change of measure. The original pure jump process is shown in red and the one under a modified measure is shown in blue.

In order to monitor the computing time before and after the change of measure, Table 4.1 shows some sampling distribution cutpoints of $\tau_{j,2}$. Note that $\hat{\tau}_{j,2}$ was calculated with the rates (4.1.0.10) and $\tilde{\tau}_{j,2}$ was estimated using the change of measure from Chapter 3. It can be seen that on average the process transitions to the opposite equilibrium point faster after doing the change of measure.

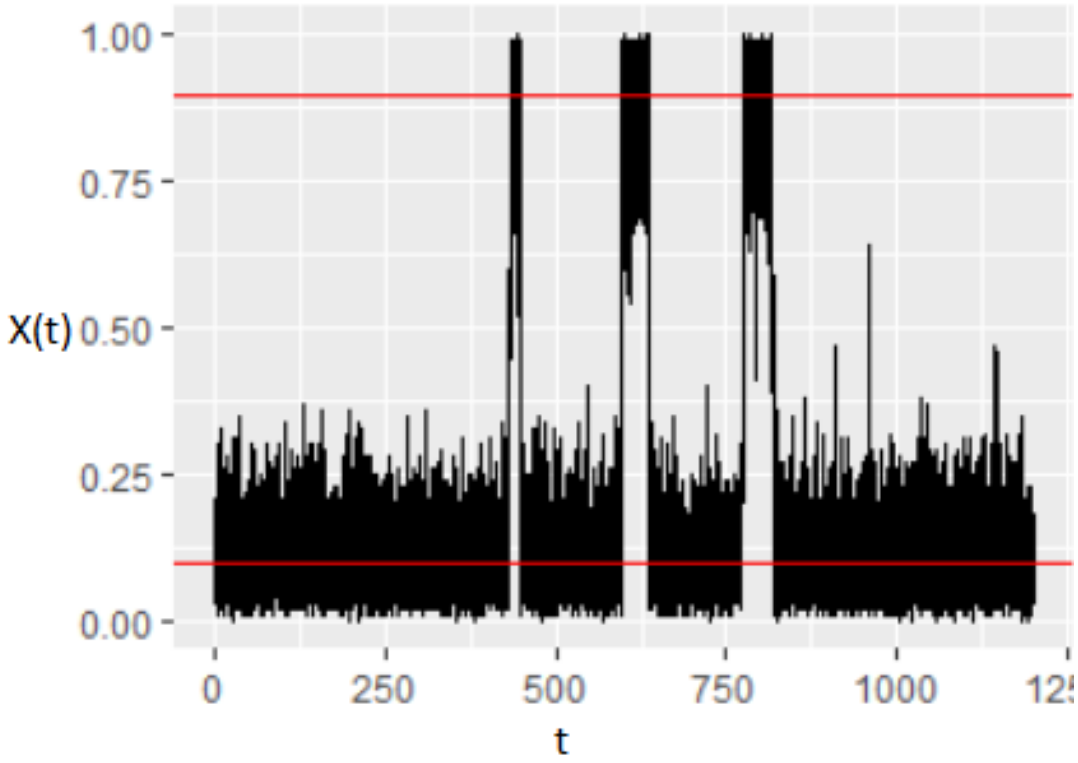

Figure 4.1: Sample path for a pure jump process having $\alpha = 0.1$

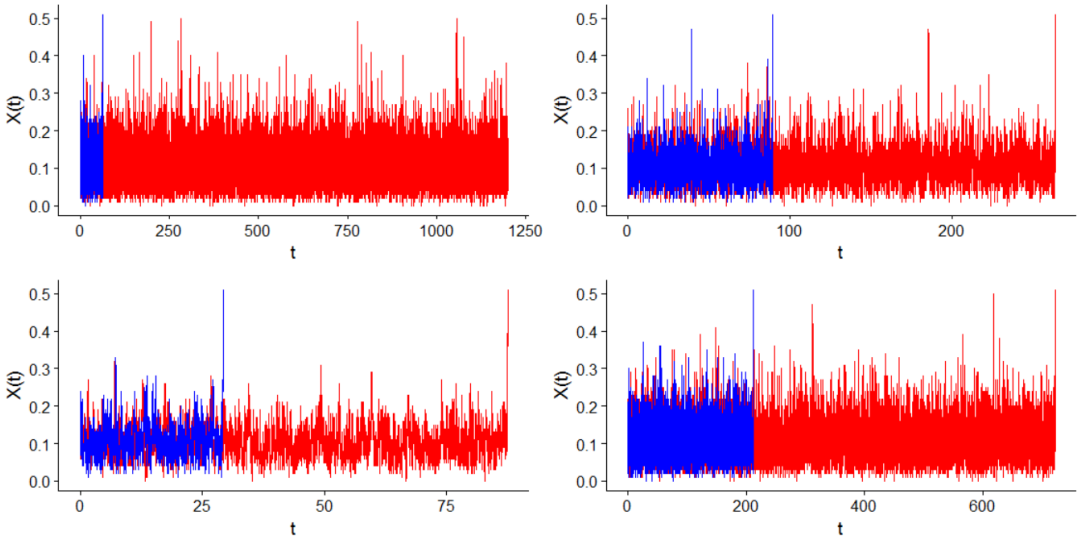

Figure 4.2: Sample paths $X_N(t)$ for $0 < t < \tau_{1,2}$ and $X_N(0) = 0.1$

	Min	1st Quantile	Median	Mean	3rd Quantile	Max
$\hat{\tau}_{1,2}$	0.7928	135.2144	331.1082	440.273	655.547	1200
$\tilde{\tau}_{1,2}$	0.1946	0.6741	1.2900	1.8812	1.4942	22.8849
$\hat{\tau}_{3,2}$	1.1566	16.2111	36.8339	52.1695	70.5986	343.9876
$\tilde{\tau}_{3,2}$	0.0623	0.899	0.6363	1.5515	2.4763	13.9837

Table 4.1: Descriptive statistics of the sampling distribution of $\tau_{j,2}$ for $N = 100$

On the surface the sample mean exit times generated under the original measure, say $\hat{\beta}_{j,2}$, are far from the true mean exit times $\beta_{j,2}$. Indeed $\hat{\beta}_{1,2} = 440.273$ and $\hat{\beta}_{3,2} = 52.1695$ while $\beta_{1,2} \approx 819$ and $\beta_{1,2} \approx 103$. In order to investigate this disparity, three more batches of three thousand simulations each where done for a similar process, for different values of system capacity $N = 110, 120, 130$. The results with the original measure are presented in Table 4.2. Additionally, in Table 4.3 there is a comparison between $\hat{\beta}_{j,2}$ and $\beta_{j,2}$ where it can be seen that the ratio $\hat{\beta}_{j,2}/\beta_{j,2}$ decreases as N increases, which is natural since the results 4.1.0.3 are asymptotic in N.

N	j	Min	1st Quantile	Median	Mean	3rd Quantile	Max
100	1	0.7928	135.2145	331.1083	440.2735	655.5470	1200
110	1	3.7800	256.9934	606.8112	872.8237	1213.8153	1600
120	1	36.4000	504.9914	1213.8200	1746.2227	2428.2752	3130
130	1	32.1250	1050.5878	2512.8970	3617.4700	5023.5485	6121
100	3	1.1566	16.2112	36.8339	52.1696	70.5986	344
110	3	17.3495	34.9768	66.8039	90.8548	127.5979	164
120	3	4.4040	50.2001	108.3208	151.1807	209.9579	262
130	3	5.2756	77.7380	173.0655	247.1901	347.3977	416

Table 4.2: Descriptive statistics of the sampling distribution of $\tau_{j,2}$ for different N

Table 4.4 shows estimates for $p_{j,2}$ as well as its variance. $\mathrm{Var}(\hat{p}_{j,2})$ makes reference to a process generated under the original measure, while $\mathrm{Var}(\tilde{p}_{j,2})$ was calculated after the change of measure. We can see two things: firstly, the process takes longer to leave the stable equilibria $x_1 = 0.1$ since

N	j	$\hat{\beta}_{j,2}$	$\beta_{j,2}$	$\hat{\beta}_{j,2}/\beta_{j,2}$
100	1	440.2735	819.0000	1.8602
110	1	872.8237	1600.4881	1.8337
120	1	1746.2227	3130.0273	1.7925
130	1	3617.4700	6121.3019	1.6922
100	3	52.1696	103.0000	1.9743
110	3	90.8548	164.4585	1.8101
120	3	151.1807	261.5267	1.7299
130	3	247.1901	415.8873	1.6825

Table 4.3: $\hat{\beta}_{j,2}$ versus $\beta_{j,2}$ for the marked point process

overall $p_{2,3} > p_{1,3}$, agreeing with the results from Table 4.2. Secondly, $\text{Var}(\tilde{p}_{j,2})$ is consistently lower than $\text{Var}(\hat{p}_{j,2})$.

T	0.7	0.8	0.9	1	1.1
$p_{1,2}$	0.0014733	0.0016080	0.0016727	0.0024297	0.0017374
$\text{Var}(\hat{p}_{1,2})$	0.0010000	0.0019980	0.0019980	0.0019980	0.0019980
$\text{Var}(\tilde{p}_{1,2})$	0.0000189	0.0000197	0.0000259	0.0000275	0.0000288
$p_{3,2}$	0.0129333	0.0142693	0.0168116	0.0197094	0.0206596
$\text{Var}(p_{3,2})$	0.0128438	0.0138178	0.0147898	0.0157598	0.0167277
$\text{Var}(\tilde{p}_{3,2})$	0.0000842	0.0001975	0.0002005	0.0002227	0.0000227

Table 4.4: Estimates of $p_{j,2}$ and their variance using the marked point process

It follows from Chapter 3 of (Bucklew, 2013) and (4.1.0.3) that the probabilities from Table 4.4 follow an exponential distribution. Indeed, $p_{j,2} = \mathbb{P}[\tau_{j,2} < T] = 1 - e^{-\beta_{j,2}T}$. This can be used as a way to corroborate that we are obtaining equivalent estimators under both the original and the modified sample measures, by fitting a simple homogeneous linear regression to a transformation of the pairs $(T, p_{j,2})$. In other words, if we have the relations $\hat{p}_{j,2} = 1 - e^{-\hat{\beta}_{j,2}T}$ and $\tilde{p}_{j,2} = 1 - e^{-\tilde{\beta}_{j,2}T}$, then $\hat{\beta}_{j,2} \approx \tilde{\beta}_{j,2}$. Certainly for the case $N = 100$, $\hat{\beta}_{1,2} \approx 449$ and $\tilde{\beta}_{1,2} \approx 452$. Likewise $\hat{\beta}_{3,2} \approx 55$ and

$\tilde{\beta}_{3,2} \approx 58$. Details on this comparison are shown in Table 4.5.

N	j	$\hat{\beta}_{j,2}$	$\tilde{\beta}_{j,2}$
100	1	449.1316	452.3304
110	1	873.3606	877.2438
120	1	1746.2227	1746.8112
130	1	3623.5889	3627.2657
100	3	55.4594	58.43321
110	3	93.3566	94.7848
120	3	155.5272	158.1174
130	3	253.2253	256.6878

Table 4.5: Comparison of $\hat{\beta}_{j,2}$ and $\tilde{\beta}_{j,2}$ for the marked point process

□

4.2 Jump-diffusion approximation

A first approach to improve the computation efficiency of any simulation of the process $X_N(t)$ is approximating it by using a diffusion process solving the following stochastic differential equation:

$$\begin{aligned} X_N(t) &= b(X_N(t))dt \\ &+ \sqrt{\varepsilon}\sqrt{X_N(t)(1-X_N(t))}dW(t) + \frac{1}{\sqrt{N}}\sqrt{r_+(X_N(t)) + r_-(X_N(t))}dW(t) \end{aligned} \tag{4.2.0.20}$$

where $W(t)$ is a Brownian Motion. The drift coefficient is written in a way that it reflects both sets of jumps from the marked point process, i.e., (4.1.0.5) and (4.1.0.6):

$$b(x) = -\kappa_{-1}^{10}x + \kappa_1^{01}(1-x) - \kappa_{-1}^{11}x(1-x) + \kappa_1^{21}x^2(1-x). \tag{4.2.0.21}$$

However, while the jump rates (4.1.0.10) take care of X_N when it reaches the boundaries of $[0,1]$ by pushing it back inside (the states are non absorbing), the diffusion approximation could

leave said interval at any point causing some domain issues since, for instance, the domain of the diffusion coefficient $\sqrt{X_N(t)(1-X_N(t))}$ is indeed $[0,1]$. And even if the problem was not a domain restriction, we are modelling a molecular concentration which has to lie between zero and one. Such issues evidence the need to add an extra term to (4.2.0.20)

$$\begin{aligned} X_N(t) &= b(X_N(t))dt \\ &\quad + \sqrt{\varepsilon}\sqrt{X_N(t)(1-X_N(t))}dW(t) + \frac{1}{\sqrt{N}}\sqrt{r_+(X_N(t)) + r_-(X_N(t))}dW(t) \\ &\quad + \sqrt{\varepsilon}\gamma(X_N(t))dL(t). \end{aligned} \tag{4.2.0.22}$$

The process $L(t)$ is, as described in (Leite and Williams, 2017), a one-dimensional, continuous, increasing process and it is called the *reflection term*. It reflects $X_N(t)$ in the following way: when $X_N(t)$ reaches zero it jumps up by $\sqrt{\varepsilon}$ and when $X_N(t)$ reaches one it jumps down by $-\sqrt{\varepsilon}$. This means that $\gamma(X_N(t))$ only takes the values ± 1 when $X_N(t) \in \{0,1\}$. Since (4.2.0.20) and (4.2.0.22) have the same drift and diffusion coefficients, according to (Leite and Williams, 2017), both solutions $X_N(t)$ will have the same distributional behaviour for positive values.

Analogously to (4.1.0.3) there is a large deviation result which characterizes the mean exit time $\beta_{j,2}$ for the diffusion approximation. Let us call $I_{j,2}$ that rate (see section 3 from (McSweeney et al., 2014)). Theorem 3.1 of said reference states that:

$$l_{j,2} \leq I_{j,2} \tag{4.2.0.23}$$

which entails an overestimation of the mean exit time $\beta_{j,2}$ using this approximation, comparing it to the marked point process.

Example 4.2.1. Let us begin by simulating a path starting at $X(0) = 0.1$ until the process reaches $T = 1200$, using the parameters $N = 100$ and $\varepsilon = 0.01$. Once more $(\kappa_{-1}^{10}, \kappa_1^{01}, \kappa_{-1}^{11}, \kappa_1^{21}) = (36/19, 36/19, 400/19, 800/19)$. With this configuration, we expect to simulate a jump-diffusion process approximating the marked point process presented in Example 4.1.1. Certainly, as Figure 4.3 shows, the jump-diffusion behaves at first glance in a similar way to the pure jump process described in Figure 4.1. We can say that the process stays longer around 0.1 than around 0.9.

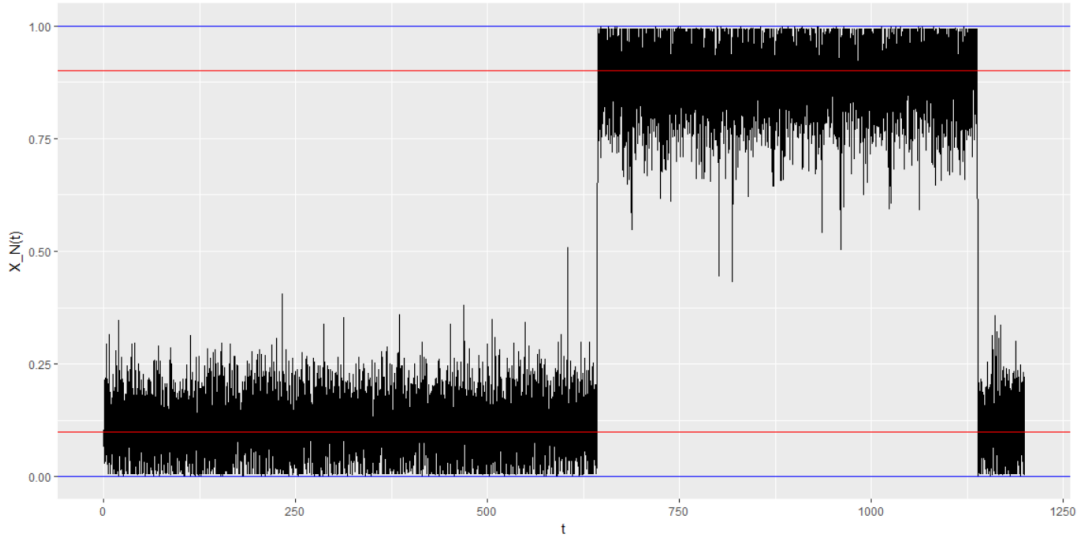

Figure 4.3: Sample path for the jump-diffusion approximation

We are once again interested in measuring $\tau_{j,2}$. To that end, three thousand simulations were performed, half of them with $X_N(0) = 0.1$ and the other half with $X_N(0) = 0.9$. The simulations were done with drift shown in 4.2.0.21 and a second time with the change of drift studied in Chapter 3.

If we look at some of the sample paths shown in Figure 4.4, we can gain some insight on the fact that the process does leave Ω faster after the change of measure. The original pure jump process is shown in red and the one under a modified measure is shown in blue.

Table 4.6 shows some sampling distribution cutpoints of $\tau_{j,2}$. Once more, $\hat{\tau}_{j,2}$ was calculated under the original measure and $\tilde{\tau}_{j,2}$ was estimated using the change of measure from Chapter 3. It can be seen that on average the process leaves Ω faster after the change of measure. Additionally, the exit time seems to be overall higher than the one reported on Table 4.1.

In a similar way to Table 4.3, Table 4.7 shows that the sample exit time $\hat{\beta}_{j,2}$ increases as N increases, accordingly to the theoretical $\beta_{j,2}$.

Table 4.8 shows estimates for $p_{j,2}$. Again $\hat{p}_{j,2}$ was calculated under the original measure and $\tilde{p}_{j,2}$ was done with the change of measure from Chapter 3. Concurring with the marked point process, the jump-diffusion approximation takes longer to leave the stable equilibria $x_1 = 0.1$ since generally speaking $p_{3,2} > p_{1,2}$. Furthermore, the change of measure cuts down the variance of the estimators.

Finally, in order to show that the algorithm produces the same estimators for the mean exit time

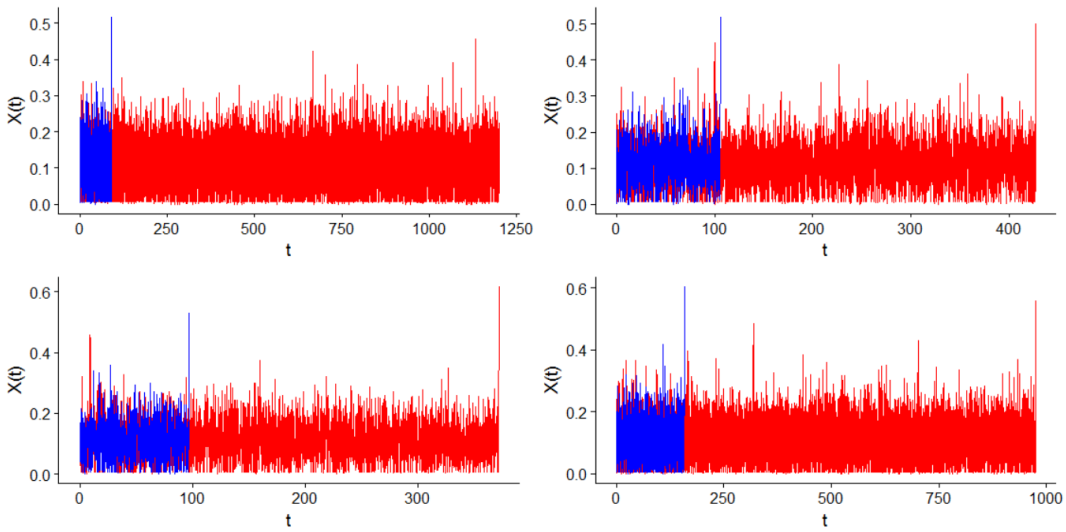

Figure 4.4: Sample paths for a jump diffusion stopped when it reaches 0.5

	Min	1st Quantile	Median	Mean	3rd Quantile	Max
$\hat{\tau}_{1,2}$	4.783333333	167.3	405.125	492.6666667	1415.75	3295
$\tilde{\tau}_{1,2}$	1.759916667	28.24966667	38.66333333	61.33166667	92.63333333	1065.75
$\hat{\tau}_{3,2}$	3.1925	23.9888174	50.51192985	70.2121	99.31019511	993
$\tilde{\tau}_{3,2}$	0.803029055	0.801036371	1.260971999	1.607432468	7.011416595	18.24

Table 4.6: Descriptive statistics of the sampling distribution of $\tau_{j,2}$ for $N = 100$

regardless of the change of measure, Table 4.9 shows a comparison of $\hat{\beta}_{j,2}$ and $\tilde{\beta}_{j,2}$, in an identical way as Table 4.5 was produced.

□

N	j	$\hat{\beta}_{j,2}$	$\beta_{j,2}$
100	1	492.6666	3531.3658
110	1	1218.4718	7993.5000
120	1	1952.8600	18093.8612
130	1	3791.5800	40956.7536
100	3	70.2121	1085.1896
110	3	98.2056	2183.0123
120	3	165.7333	4391.4379
130	3	272.6350	8833.9983

Table 4.7: Comparison of $\hat{\beta}_{j,2}$ and $\beta_{j,2}$ for the jump-diffusion approximation

T	0.7	0.8	0.9	1	1.1
$p_{1,2}$	0.0022030	0.0015626	0.0023434	0.0028945	0.0024226
$\text{Var}(\hat{p}_{1,2})$	0.0063857	0.0029698	0.0031157	0.0069214	0.0031728
$\text{Var}(\tilde{p}_{1,2})$	0.0000066	0.0000014	0.0000050	0.0000111	0.0000107
$p_{3,2}$	0.0107593	0.0112648	0.0134860	0.0144964	0.0164246
$\text{Var}(\hat{p}_{3,2})$	0.0404377	0.0813542	0.0380306	0.0375789	0.0170263
$\text{Var}(\tilde{p}_{3,2})$	0.0000024	0.0000135	0.0003096	0.0006552	0.0007215

Table 4.8: Estimates of $p_{j,2}$ and their variance

N	j	$\hat{\beta}_{j,2}$	$\tilde{\beta}_{j,2}$
100	1	500.2136452	502.2959038
110	1	1219.960567	1223.214005
120	1	1953.172545	1955.81412
130	1	3792.469907	3794.047212
100	3	71.60884605	76.22521347
110	3	99.09976415	100.5505656
120	3	173.1493891	175.6256606
130	3	277.6553545	281.6203918

Table 4.9: Comparison of $\hat{\beta}_{j,2}$ and $\tilde{\beta}_{j,2}$ for the jump-diffusion approximation

Chapter 5

Concluding remarks

Handling simulation of rare events is certainly a challenging task, yet the methodology proposed by (Djehiche et al., 2014) eases the path greatly. Indeed, the contribution of this thesis is first and foremost a thorough dissection of a variance-reducing algorithm of certain Monte Carlo estimators in which we applied important concepts as importance sampling and large deviations. Though in order to get to the core point, we needed to provide a detailed outline of the steps needed to simulate a variety of stochastic processes, e.g., a marked point process, a diffusion process or a combination of both.

The simulation of a diffusion process starts with methods as intuitive as the Euler discretization (2.3.1). Also the method for simulating marked point processes presented in Algorithm 1 follows directly from its underlying definitions. Given that simplicity was a concern at all times, the Giesecke/ Teng/ Wei approximation for jump-diffusions is very valuable. It is indeed more efficient and natural than, for instance, the exact sample simulation studied in (Giesecke and Smelov, 2013). We successfully tested its usefulness by easily and accurately simulating the price of a call option on a short-interest rate, as well as the value at the end of a given time horizon of a simple Brownian motion with jumps.

Inside the core of the algorithm presented in Chapter 3, which was heavily based in solving non-trivial equations, one runs into predicaments like root-finding or optimization, which fortunately can be solved numerically. In addition, parallel computing does wonders for computation time given the iterative nature of Monte Carlo simulations. Throughout the thesis, the programming language of choice was R, given its wide popularity in the statistical world. However, it would be fascinating

to test the performance of the algorithm under another prominent language for scientific computing, such as Python.

In the context of systems biology, our algorithm proved as well to be useful in cutting down the variance of exact processes and their proposed approximations alike. It was certainly fruitful to compare both approaches: the marked point process and its jump-diffusion approximation, whose accuracy consistently improves by raising the value of system's capacity N. More importantly, being the latter a good approximation of the former, we spare some computation resources since simulating a diffusion process is always a much less resource-intensive task, compared to simulating a marked point process.

Bibliography

Abel, N. H. (1826). Démonstration de limpossibilité de la résolution algébrique des équations générales qui passent le quatrieme degré. *Journal für die reine und angewandte Mathematik*, 1:65–96.

Allen, L. J. (2010). *An introduction to stochastic processes with applications to biology*. CRC Press.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *The journal of political economy*, pages 637–654.

Brent, R. P. (2013). *Algorithms for minimization without derivatives*. Courier Corporation.

Brown, R. (1828). Xxvii. a brief account of microscopical observations made in the months of june, july and august 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *Philosophical Magazine Series 2*, 4(21):161–173.

Bucklew, J. (2013). *Introduction to rare event simulation*. Springer Science & Business Media.

Carr, P. and Linetsky, V. (2006). A jump to default extended cev model: an application of bessel processes. *Finance and Stochastics*, 10(3):303–330.

Djehiche, B., Hult, H., and Nyquist, P. (2014). Min-max representations of viscosity solutions of hamilton-jacobi equations and applications in rare-event simulation. *arXiv preprint arXiv:1406.3605*.

Dupuis, P., Spiliopoulos, K., and Wang, H. (2012). Importance sampling for multiscale diffusions. *Multiscale Modeling & Simulation*, 10(1):1–27.

Dupuis, P., Spiliopoulos, K., Zhou, X., et al. (2015). Escaping from an attractor: Importance sampling and rest points i. *The Annals of Applied Probability*, 25(5):2909–2958.

Dupuis, P. and Wang, H. (2007). Subsolutions of an isaacs equation and efficient schemes for importance sampling. *Mathematics of Operations Research*, 32(3):723–757.

Franzke, C. L., O'Kane, T. J., Berner, J., Williams, P. D., and Lucarini, V. (2015). Stochastic climate theory and modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 6(1):63–78.

Giesecke, K. and Smelov, D. (2013). Exact sampling of jump diffusions. *Operations Research*, 61(4):894–907.

Giesecke, K., Teng, G., and Wei, Y. (2015). Numerical solution of jump-diffusion sdes. *Available at SSRN 2298701*.

Grimmett, G. and Stirzaker, D. (2001). *Probability and random processes*. Oxford university press.

Iacus, S. M. (2009). *Simulation and inference for stochastic differential equations: with R examples*, volume 1. Springer Science & Business Media.

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management science*, 48(8):1086–1101.

Leite, S. C. and Williams, R. J. (2017). A constrained langevin approximation for chemical reaction network. *preprint available at the webpage http://www. math. ucsd. edu/˜ williams/biochem/biochem. html*.

McSweeney, J. K., Popovic, L., et al. (2014). Stochastically-induced bistability in chemical reaction systems. *The Annals of Applied Probability*, 24(3):1226–1268.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The review of Economics and Statistics*, pages 247–257.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, 3(1-2):125–144.

Piessens, R., de Doncker-Kapenga, E., Überhuber, C. W., and Kahaner, D. K. (2012). *QUADPACK: A subroutine package for automatic integration*, volume 1. Springer Science & Business Media.

Runggaldier, W. J. (2003). Jump-diffusion models. *Handbook of heavy tailed distributions in finance*, 1:169–209.

Shreve, S. E. (2004). *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media.

Tankov, P. (2003). *Financial modelling with jump processes*, volume 2. CRC press.

Tankov, P. and Voltchkova, E. (2009). Jump-diffusion models: a practitioners guide. *Banque et Marchés*, 99(1):24.

Vanden-Eijnden, E. and Weare, J. (2012). Rare event simulation of small noise diffusions. *Communications on Pure and Applied Mathematics*, 65(12):1770–1803.

Wilkinson, D. J. (2011). *Stochastic modelling for systems biology*. CRC press.

Appendix A

R code: estimation of the price of a call option

```
###################################
# Parameters #
##############

set.seed(1234)
x_0<-0.1
kappa<-2
theta<-0.1
sigma<-0.02
lambda_0<-5
lambda_1<-50
K<-0.1

###################################
# Functions #
#############

b<-function(x){
```

```
  kappa*(theta-x)
}

sigma_fn<-function(x){
  sigma
}

lambda<-function(x){
  lambda_0+lambda_1*x
}

gamma_fn<-function(x,y){
  y
}

random_mark<-function(){
  sample(x = c(0.01,0.015,0.02,0.025,0.03),replace = FALSE,size = 1)
}

##############################
# Program #
############

T_<-1
N_steps<-400
h<-T_/N_steps
cap_trials<-c()
cum_sum<-c()

for(j in 1:160000){
```

```
  i<-0
  n<-0
  s<-0
  Z<-x_0
  A<-0
  E_n<-rexp(n = 1,rate = 1)
  while(s<T_){
    A_temp<-A+lambda(Z)*((i+1)*h-s)
    N<-rnorm(n = 1,mean = 0,sd = 1)
    if(A_temp >= E_n){
      tau_n<-s+(E_n-A)/(lambda(Z)) #time the jump actually happened
      Z_tau_minus<-Z+b(Z)*(tau_n-s)+sigma_fn(Z)*sqrt(tau_n-s)*N
      Z<-Z_tau_minus+gamma_fn(Z_tau_minus,random_mark())
      s<-tau_n
      A<-E_n
      n<-n+1
      E_n<-E_n+rexp(n = 1,rate = 1)
    }
    else{
      Z<-Z+b(Z)*((i+1)*h-s)+sigma_fn(Z)*sqrt((i+1)*h-s)*N
      s<-(i+1)*h
      A<-A_temp
      i<-i+1
    }
  }
  cap_trials<-c(cap_trials,max(Z-K,0))
  cum_sum<-c(cum_sum,mean(cap_trials))
  print(j)
}
```

```
plot(cum_sum,type="l",xlab = "Number of simulations", ylab="Sample mean")
cum_sum[length(cum_sum)]
```

Appendix B

R code: simulation of a Brownian motion with a birth-and-death process

```
#############
# Functions #
#############

b<-function(x){
  0
}

sigma_fn<-function(x){
  x
}

lambda<-function(x){
  lambda_0+lambda_1*x
}

gamma_fn<-function(y){
  1*y
```

```
}

random_mark<-function(){
  sample(x =mark_set,replace = FALSE,size = 1)
}

H.px<-function(p,x){
  exponent<-diag(x=p,nrow=length(p),ncol=length(p))%*%t(replicate(length(p),mark_set))
  b(x)*p+(0.5*sigma_fn(x)^2)*p^2+rowSums(exp(exponent)-1-exponent)
}

c_H<-function(){
  inside<-function(x){
    optimize(f=H.px,interval = c(-10000,10000),maximum = FALSE,x=x)$objective
  }
  optimize(f=inside,interval = c(-10000,10000),maximum = TRUE)$objective
}


p.fn<-function(z,c_,max){
  H.xpc<-function(p,xx,c_){
    exponent<-diag(x=p,nrow=length(p),ncol=length(p))%*%t(replicate(length(p),mark_set))
    b(xx)*p+(0.5*sigma_fn(xx)^2)*p^2+rowSums(exp(exponent)-1-exponent)-c_
  }
  pp<-sapply(X = z,FUN=function(x) uniroot.all(f = H.xpc,interval = c(-10,10),xx=x,c_=c_))
  if (max==TRUE){fff<-"max"}
  else {fff<-"min"}
  if (is.null(dim(pp))){
    return(pp)
  }
```

```
  else{
    return(apply(X = pp,MARGIN = 2,FUN = fff))
  }
}

mane<-function(c_,x,y){
  if(x<y){
    integrate(p.fn,lower = x,upper=y,c_=c_,max=TRUE)
  }
  else{
    integrate(p.fn,lower = x,upper=y,c_=c_,max=FALSE)
  }
}

min_max<-function(c_,cap_t,low_t,a,b_){
  min(mane(c_,x0,a)$value,mane(c_,x0,b_)$value)-c_*(cap_t-low_t)
}

Theta<-function(x,c_){
  if(x>x0){
    sigma_fn(x)*p.fn(x,c_,max=TRUE)
  }
  else{
    sigma_fn(x)*p.fn(x,c_,max=FALSE)
  }
}

b_tilde<-function(x,varepsilon,c_){
  b(x)+sqrt(varepsilon)*sigma_fn(x)*Theta(x,c_)
}
```

```
new_step_original<-function(Z,s,A,E_n,i,L,h,varepsilon,TT){
  A_temp<-A+lambda(Z)*((i+1)*h-s)
  N<-rnorm(n = 1,mean = 0,sd = 1)
  if(A_temp >= E_n){
    tau_n<-s+(E_n-A)/(lambda(Z)) #time the jump actually happened
    Z_tau_minus<-Z+b(Z)*(tau_n-s)+varepsilon*sigma_fn(Z)*sqrt(tau_n-s)*N
    Z<-Z_tau_minus+varepsilon*gamma_fn(y = random_mark())
    s<-tau_n
    A<-E_n
    E_n<-E_n+rexp(n = 1,rate = 1)
  }
  else{
    Z<-Z+b(Z)*((i+1)*h-s)+sqrt(varepsilon)*sigma_fn(Z)*sqrt((i+1)*h-s)*N
    Z_tau_minus<-Z
    s<-(i+1)*h
    A<-A_temp
    i<-i+1
  }
  data.frame(Z=Z,s=s,A=A,E_n=E_n,i=i,L=L)
}

new_step_after<-function(Z,s,A,E_n,i,L,h,varepsilon,TT,c_){
  A_temp<-A+lambda(Z)*((i+1)*h-s)
  N<-rnorm(n = 1,mean = 0,sd = 1)
  if(A_temp >= E_n){
    tau_n<-s+(E_n-A)/(lambda(Z)) #time the jump actually happened
    Z_tau_minus<-Z+b_tilde(Z,varepsilon,c_)*(tau_n-s)+sqrt(varepsilon)*sigma_fn(Z)
    *sqrt(tau_n-s)*N
    L<-L+(Theta(Z,c_)*L*sqrt(tau_n-s)*N)/sqrt(varepsilon)
```

```
    Z<-Z_tau_minus+varepsilon*gamma_fn(y = random_mark())
    s<-tau_n
    A<-E_n
    E_n<-E_n+rexp(n = 1,rate = 1)
  }
  else{
    Z_tau_minus<-Z+b_tilde(Z,varepsilon,c_)*((i+1)*h-s)+sqrt(varepsilon)*sigma_fn(Z)
    *sqrt((i+1)*h-s)*N
    L<-L+(Theta(Z,c_)*L*sqrt((i+1)*h-s)*N)/sqrt(varepsilon)
    Z<-Z_tau_minus
    s<-(i+1)*h
    A<-A_temp
    i<-i+1
  }
  data.frame(Z=Z,s=s,A=A,E_n=E_n,i=i,L=L)
}



chain<-function(x0,t0,T_,epsilon,Nsteps,c_,new){
  h<-(T_-t0)/Nsteps
  Z<-x0
  s<-t0
  A<-0
  E_n<-rexp(n = 1,rate = 1)
  i<-0
  L<-1
  Z_t<-data.frame(Z=Z,s=s,A=A,E_n=E_n,i=i,L=L)
  if(new==FALSE){
    while((s<T_)&&(Omega_a<Z)&&(Z<Omega_b)){
      #while(s<T_){
```

```
      current_step<-new_step_original(Z=Z,s=s,A=A,E_n=E_n,i=i,
      L=L,h=h,varepsilon=epsilon,TT=T_)
      Z<-current_step$Z
      s<-current_step$s
      A<-current_step$A
      E_n<-current_step$E_n
      i<-current_step$i
      Z_t<-rbind(Z_t,current_step)
    }
  }
  else{
    while((s<T_)&&(Omega_a<Z)&&(Z<Omega_b)){
      #while(s<T_){
      current_step<-new_step_after(Z=Z,s=s,A=A,E_n=E_n,i=i,
      L=L,h=h,varepsilon=epsilon,TT=T_,c_=c_)
      Z<-current_step$Z
      s<-current_step$s
      A<-current_step$A
      E_n<-current_step$E_n
      i <-current_step$i
      L<-current_step$L
      Z_t<-rbind(Z_t,current_step)
    }
  }
  return(Z_t)
}

##########
# Script #
##########
```

```
library(rootSolve)
source('C:/Users/User/Google Drive/Research/Git/Djehiche_Giesecke/functions.R')

#library(pushoverr)

set.seed(1234)
x0<-100
lambda_0<-5
lambda_1<-50
varepsilon_vec<-c(0.05,0.06,0.07,0.08)
Omega_a<-90
Omega_b<-110
mark_set<-c(0.01, 0.015, 0.02, 0.025, 0.03)
T_vec<-0.04
parameters<-expand.grid(varepsilon_vec,T_vec)
low_t<-0
N_steps<-100
sample_size<-10^3

#########################################################
# Program #
###########

for(k in 1:length(parameters[,1])){
  time_trials<-indicators<-prom<-c()
  for(j in 1:sample_size){
    ZandL<-chain(x0 = x0,t0 = low_t,T_ = parameters[k,2],epsilon = parameters[k,1],
    Nsteps = N_steps,c_=0,new = FALSE)
    indicators<-c(indicators,1*(ZandL$s[length(ZandL$s)]<parameters[k,2]))
```

```
    print(c(k,j))
  }
  estim<-c(mean(indicators),(1/sample_size)*sqrt((1/mean(indicators))-1))
  write.table(x=c(date(),"## BEFORE ##",sample_size,N_steps,parameters[k,],estim),
  file=paste("C:/Users/User/Google Drive/Research/Code/[Djehiche+Giesecke]
  Epsilon Jump Diffusion v",vers,".txt"), append=TRUE,row.names = F, col.names = F,
   quote=F, sep="//")
}

# time_used.old<-proc.time()-start_time.old

################################################################################################
# Intermediate #
################

c_vec<-rep(0,length(parameters[,2]))
for(k in 1:length(parameters[,2])){
  c_vec[k]<-optimize(min_max,interval =c(c_H(),100000),maximum =TRUE ,
  cap_t=parameters[k,2], low_t=low_t,a=Omega_a,b=Omega_b)$maximum
}

############################
# After that #
##############

for(k in 1:length(parameters[,1])){
  time_trials<-indicators<-prom<-c()
  #c_<-c_vec[k]
  for(j in 1:sample_size){
    ZandL<-chain(x0 = x0,t0 = low_t,T_ = parameters[k,2],epsilon = parameters[k,1],
```

```
    Nsteps = N_steps,c_=c_vec[k],new = TRUE)
    indicators<-c(indicators,
    1*(ZandL$s[length(ZandL$s)]<parameters[k,2])*(ZandL$L[length(ZandL$L)])^-1)
    print(c(k,j))
  }
  estim<-c(mean(indicators),(1/sample_size)*sqrt((1/mean(indicators))-1))
  write.table(x=c(date(),"## AFTER ##",sample_size,N_steps,parameters[k,],estim),
  file=paste("C:/Users/User/Google Drive/Research/Code/[Djehiche+Giesecke]
   Epsilon Jump Diffusion v",vers,".txt"), append=TRUE,row.names = F, col.names = F,
   quote=F, sep="//")
}
```