Efficient and Generic Monte-Carlo Methods for Computing Sensitivities of Stochastic Systems

Dan Zhu

Submitted in total fulfillment of the requirements of the degree of Doctor of Philosophy

May, 2016

Centre for Actuarial Studies
Department of Economics
The University of Melbourne

Produced on archival quality paper

Abstract

This thesis introduces new Monte-Carlo methods for sensitivity analysis in stochastic dynamical systems. Simulation is an efficient tool that provides solutions to problems with multi-dimensional stochastic set-ups. It is a method for propagating model inputs into model outputs through statistical sampling. Since the outputs inherently depend on the inputs, it is important to assess the impact of their changes, i.e. the derivatives of the output with respect to the model inputs. The ability to successfully perform derivative estimation via simulation is highly dependent on the mathematical design of the underlying program and the dimensionality of the problem. Accordingly, there is a need for the improved methods in this thesis.

In Chapter 3, we first present a new method to compute second-order sensitivities of financial products with discontinuous or angular payoffs by Monte Carlo simulation. The idea is to use a sequence of measure changes such that the limits' dependence on the parameter of interest is eliminated. The methodology is optimal in terms of minimizing the variance of the likelihood ratio weight. Applications are presented for both equity options and interest-rate products with discontinuous payoff structures.

In Chapter 4, we consider a special class of exotic derivatives, i.e. products with early-exercise features. Given the exercise strategy, the pathwise angularities are removed by a sequence of measure changes. The first-order optimal partial proxy method is implemented to calculate the Hessians of Bermudan Swaptions and cancellable Swaps, the resulting pathwise estimates are unbiased and accurate.

The numerical schemes in Chapter 3 and 4 are determined by the discontinuities of the pathwise performance measure. That is the critical point where the performance measure passes the discontinuity. In Chapter 5, we address the issue of inexplicit critical points. Numerical techniques are used to approximate these points, and the bias introduced vanishes at the limit when computing first- and second-order derivatives.

While Chapter 3 and Chapter 4 address the issue associated with irregular payoff functions, we consider the irregularities of the underlying distribution in Chapter 6. We introduce new methods to computing first- and second-order sensitivities of functions simulated by rejection techniques. Measure changes are introduced to ensure the pathwise discontinuities in the acceptance-rejection decisions are removed. Applications are presented for computing Greeks of equity options with certain L*é*vy-driven underlyings and to finding sensitivities of performance measures in queuing systems.

In Chapter 7, we use derivative estimation methods developed in previous chapters to determine the level of capital required to maintain a certain level of ruin probability. We first present a stratified sampling algorithm for estimating extremely small finite-time ruin probabilities. We further compute unbiased first- and second-order derivative estimates of the finite-time ruin probabilities with respect to both distributional and structural parameters. We then estimate the regulatory capital and its sensitivities. These estimates provide information to insurance companies for meeting prudential regulations as well as for designing risk management strategies.

Declaration

This is to certify that:

1. *the thesis comprises only my original work towards the PhD except where indicated in the Preface,*
2. *due acknowledgment has been made in the text to all other material used,*
3. *the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.*

Signed,

Dan Zhu

Preface

This thesis was completed under the supervision of Professor Mark Joshi in the Centre for Actuarial Studies, The University of Melbourne. Chapter 3 to 7 contain the original research of this thesis, except as otherwise noted.

Chapter 3 and 5 are based on one paper, titled "*Optimal Partial Proxy Method for Computing Gammas of Financial Products with Discontinuous and Angular Payoffs*", which is co-authored by Mark Joshi. The paper is to appear in *Applied Mathematical Finance*. The research was carried out by Dan Zhu under the supervision of Mark Joshi. The paper was written by Dan Zhu, with proofreading and editing by Mark Joshi.

Chapter 4 is based on the paper "*An Exact and Efficient Method for Computing Cross-Gammas of Bermudan Swaptions and Cancellable Swaps under the LIBOR Market Model*" , co-authored by Mark Joshi. The paper is to appear in the *Journal of Computational Finance*. The research was carried out by Dan Zhu under the supervision of Mark Joshi. The paper was written by Dan Zhu, with proofreading and editing by Mark Joshi.

Chapter 6 is based on the paper "*An Exact Method for the Sensitivity Analysis of Systems Simulated by Rejection Techniques*", which is co-authored by Mark Joshi. The paper is to appear in the *European Journal of Operational Research*. The research was carried out by Dan Zhu under the supervision of Mark Joshi. The paper was written by Dan Zhu, with proofreading and editing by Mark Joshi.

Chapter 7 is based on the paper "*The Efficient Computation and Sensitivity Analysis of Finite-time Ruin Probabilities and the Estimation of Risk-Based Regulatory Capital*", which is co-authored by Mark Joshi. The paper is to appear in *ASTIN Bulletin*. The research was carried out by Dan Zhu under the supervision of Mark Joshi. The paper was written by Dan Zhu, with proofreading and editing by Mark Joshi.

None of the work appearing here has been submitted for any other qualifications, nor was it carried out prior to PhD candidature enrollment.

Acknowledgments

My path towards this thesis started in my childhood. My parents always told me to dream about things worth pursuing and to achieve things most people would not dream about. As a typical Chinese girl, I have followed their guidance, and worked through my life with ambition and persistence. Without this foundation, I probably would not have dared to study an intellectually intense subject, Actuarial Science, and then challenge myself to do a PhD in Financial mathematics.

While working on a PhD is a lonely endeavour, it is also not possible without others. Hence, my deepest and warmest gratitude goes to my supervisor, Professor Mark Joshi, for his kind support and guidance throughout the degree between March 2012 and February 2016. Just like the famous Chinese saying, one can only strike high if standing on a great man's shoulder. I have to thank Professor Joshi for his belief that eventually something valuable would emerge from fuzzy pictures presented in various drafts, for his questions that forced me to define and defend my choices, and for his guidance that lead me through the process of developing confidence as an academic.

To the academic members of the Centre for Actuarial Studies at the University of Melbourne, and in particular the academic staff, I express my sincere thanks for their teaching and support over my undergraduate and graduate years. I would also like to thank my colleagues at the Centre for Actuarial Studies for creating an extremely friendly environment.

Of course, I feel extremely grateful for my amazing family, extraordinary in many ways. Their support has been unconditional all these years, and they have given up many things for me to be here. In particular, my mum has always believed in my talent and my ability to achieve, and my dad provided me with support both emotionally and financially. Lastly, I have to thank my husband, Zhi, and my lovely boys, for their love and stimulating encouragements. I am forever indebted to them!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation of the monograph

Since the beginning of electronic computing, people have been interested in carrying out random experiments on a computer. Monte Carlo simulation was named after the city in Monaco, as games of chance are played in its famous casino. Although there were some early development of simulation principles, modern application of Monte Carlo methods dates from the 1940s for works on the atomic bomb experiments. Nowadays, it has become an essential tool for pricing of derivative securities, risk management, operations research and economics. Monte Carlo simulation is, in essence, the simulation of random processes by means of a computer, which arise naturally as part of the modelling of a real-life system, such as the evolution of the stock market, the arrival and departure of customers, and the surplus process of an insurance company.

Simulation should be used when the consequences of a proposed design cannot be immediately observed and when there are no analytical solutions. It is particularly valuable when there is significant uncertainty of the outcome. Probabilistic simulations provide means to quantify uncertainties such as sample standard deviations, skewness and kurtosis. Perhaps most importantly, when the system under consideration involves complex interactions of multiple stochastic processes, simulation should be used instead of other numerical techniques. Derivative pricing is one of these complicated problems where Monte Carlo simulation is widely used for valuation and risk management, and it is the main focus of this monograph.

To run a Monte Carlo simulation in practical situations such as pricing derivative products typically involves large numbers of input parameters, so that the outputs are affected by changes in the parameters. Parameter sensitivity analysis can be used to

- quantify the effect of changes in parameters in the output,
- decide which parameters should be optimized or determined more accurately through further experimental studies,
- search for errors in the model by observing unexpected relationships between inputs and outputs,
- find input factors for which the model output is optimized,
- ease the process of calibration by focusing on the sensitive parameters.

Despite its usefulness, sensitivity analysis for Monte Carlo simulations is complicated by its stochastic nature. In this monograph, we present new methods for computing parameter sensitivities of various stochastic systems. Our new approaches allow sensitivity estimations to be easily implemented in most Monte Carlo algorithms given their multi-dimensional integral formulations, even when this formulation is not explicit. We show that starting from a Monte Carlo algorithm for estimating the expected performance measure, it is possible to implement an additional procedure that simultaneously estimates its first- and second-order derivatives with respect to any parameter of interest. The corresponding supplementary cost is very low, as no additional random sampling is required.

1.2 Stochastic systems

A stochastic process is a collection of random variables, representing the evolution of some system of random values over time. Stochastic processes, in particular, Brownian motions were first studied by Bachelier (1900) in his PhD thesis to give a theory for the valuation of financial options. Ever since its first development, it has been used to study problems in various disciplines including financial mathematics, actuarial science, econometrics, queuing systems and many more. Before presenting Monte Carlo methods for computing sensitivities, we shall talk about some interesting underlying stochastic processes, that their sensitivities are studied in later chapters.

1.2.1 Black-Scholes Model

Black and Scholes (1973) and Merton (1973) introduced the famous Black-Scholes-Merton modelling framework for valuing financial derivative products, and the corresponding analytical formula for vanilla call and put options were derived. Many options, however, have multivariate payoffs with path-dependent features. For these exotic derivative products, analytical solutions are not available. The multi-dimensional Black-Scholes model has been widely used to price and hedge exotic equity and commodity derivatives, as well as insurance products depending on baskets of assets. Efficient simulation techniques for evaluating both the expected payoff as well as its sensitivities are critical for the operations of the financial industry.

Here, we briefly recap the famous model of option pricing. First, the simple stochastic differential equation (SDE) of the one-dimensional process, $S(t)$, is

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t),$$

where μ is the real-world drift, σ is the volatility and $W(t)$ is the standard Brownian motion driving the evolution of the process. This one-dimensional process is used for pricing vanilla European- and American-style options. For products with multiple underlyings, the n-dimensional geometric Brownian motion are typically used. They can be specified through a system of SDEs

$$dS_i(t) = u_i S_i(t)dt + \sigma_i dW_i(t), \text{ for } i = 1, 2, ...n,$$

where $W_i(t)$ is the standard one-dimensional Brownian motion such that $W_i(t)$ and $W_j(t)$ have correlation $\rho_{i,j}$. The covariance of $(\sigma_1 W_1, ..., \sigma_n W_n)$ is an $n \times n$ matrix Σ by setting $\Sigma_{i,j} = \sigma_i \sigma_j \rho_{i,j}$. Thus, the solution is given by

$$S_i(t) = S_i(0) \exp^{\left((\mu_i - 0.5\sigma_i^2)t + \sigma_i W_i(t)\right)}, \text{ for } i = 1, 2, ...n.$$

Under various assumptions such as the existence of riskless bond, $B_t = B_0 \exp{(rt)}$, given a constant risk free rate, r, analytical solutions of vanilla European options are derived under the one-dimensional framework. However, the prices of exotic derivative securities generally do not have closed-form solutions. Numerical methods are consequently introduced to estimate them. The partial-differential equation method (Wilmot et al, 1993) and the quadrature method (Andricopoulos,2002) will suffice for low-dimensional problems, but they are subject to the issue of curse of dimensionality, which arises if the number of underlings becomes large. Boyle (1977) developed a Monte Carlo simulation method for solving option valuation problems, which is based on the simulation of the asset returns under the risk neutrality assumption to derive the value of the option. For example, Bermudan put options are among the most liquid derivative products, which do not have analytical solutions for their prices. To value them via simulation, the simple one-dimensional algorithm for evolving the stock process under the risk-neutral measure can be written as

$$S(t) = S(0) \exp^{\left((r - 0.5\sigma^2)t + \sigma\sqrt{t}Z\right)},$$

where Z is the simulated random standard normal variate.

The advantage of Monte Carlo simulation over other numerical methods is its effectiveness in multi-dimensional settings.Various simulation schemes have been introduced to price multidimensional derivative products. The reduced-factor log-Euler scheme is probably the most widely used approach, and it is the method that we used in later chapters. A finite-time horizon, T,is discretized into a set of times, i.e. T_j for $j = 1, 2, ...N$ with $T_N = T$ and $\Delta T_j = T_j - T_{j-1}$. Let $S_i(T_j)$

denote the price of ith stock at time T_j. The scheme simulates the multi-dimensional process with F driving Brownian motions such that $F < n$. By specifying a volatility structure, one can compute the pseudo square-root of the covariance matrix, Σ, using the spectral decomposition method. This gives us the components in order of significance. The evolution across step j for each asset is

$$S_i(T_j) = S_i(T_{j-1}) \exp\Big(r\triangle T + \sum_{f=1}^{F} \big(a_{i,f}(T_{j-1})Z_{j,f} - 0.5a_{i,f}(T_{j-1})^2\big)\Big). \tag{1.1}$$

Here, $Z_{j,f}$ is the fth standard normal random variable simulated at step j and a_{ik} is the ikth element of the pseudo square-root of the covariance matrix.

1.2.2 The LIBOR Market Model

Interest-rate modelling is far more complicated than equity modelling since modelling the yield curve is a multi-dimensional problem. There are, by now, many different types of interest-rates in the market so that it is difficult to unify the pricing approach for all interest-rate related products. The first generation interest-rate models focus on modelling the continuously compounded short-rate process, which characterizes the whole yield curve. Vasicek (1977) introduced an interest-rate model, in which the short-rate process was modelled by a mean-reverting Ornstein-Uhlenbeck process. Heath et al. (1992) proposed a framework, in which the forward curve was directly modelled (known as the HJM framework). The initial term structure of bonds in this model is an input so that calibration to market prices of bonds is automatic. Under the no-arbitrage assumption of the risk-neutral measure, they showed that the drifts of the forward rates are uniquely determined by the volatility functions of the rates.

Unfortunately, instantaneous forward rates are not quoted in the market while the payoff functions of interest-rate derivatives are generally expressed in terms of market observable simple forward rates such as LIBORs and swap-rates, which facilitate the introduction of the *market models* for discrete forward rates (Brace et al., 1997). They explicitly identified the joint arbitrage-free dynamics of all the forward LIBORs under a common forward pricing measure while Jamshidian (1997) studied the forward LIBORs in an alternative fashion and introduced the concept of a spot measure. The model is consistent with the market standard approach for pricing caps using Black's formula, which helps to calibrate the model to current market prices of caplets and European swaptions. it has developed from its standard lognormal basis to more sophisticated versions to produce more realistic market features, for example, stochastic volatilities are incorporated in the model to capture the market-observed volatility smile. The model has been studied extensively in the past; for more details of the model, we refer the reader to Brace (2007), Fries (2007) and Joshi (2011).

In this section, we briefly summarize the LIBOR market model (LMM). Given, the set of tenor dates

$$0 < T_0 < T_1 < T_2 \cdots < T_{n-1} < T_n,$$

where the time difference between two reset dates T_{i+1} and T_i is τ_i. Let $B_i(t)$ denote the value of zero-coupon bonds maturing at T_i, observed at time t. We define the LIBOR spanning over the period $[T_i, T_{i+1})$ at time $t < T_i$ for $i = 0, 1, 2, ..., n-1$ as

$$f_i(t) = \frac{B_i(t) - B_{i+1}(t)}{\tau_i B_{i+1}(t)},$$

with dynamic

$$df_i(t) = u_i(f,t) f_i(t) dt + \sigma_i(t) f_i(t) dW_t^i,$$

where W_t is the F-dimensional standard Brownian motions. For $t > T_i$, we have $f_i(t) = f_i(T_i)$. Typically, the instantaneous volatility curve $\sigma_i(t)$ is chosen to be time-homogeneous and the correlations between the rates are assumed to be constant.

Define a function, $\xi : [0, T_n) \to 0, 1, 2, ..., n-1$, to be the index of the next LIBOR reset date at time t. The LMM has a multi-dimensional setting. Monte-Carlo simulations are used to price and hedge exotic interest-rate derivatives. The modelling dynamic of f_i under the spot measure with the log-Euler scheme is

$$f_i(T_{j+1}) = f_i(T_j) e^{u_i(T_j) + \sum_{k=1}^{F} (a_{ik} Z_k - 0.5 a_{ik}^2)} \tag{1.2}$$

for $T_{j+1} \leq T_i$. Here a_{ik} is the ikth element of the pseudo square-root of the covariance matrix calculated by the spectral decomposition method . Under the spot measure, the drift is

$$u_i(T_j) = \sum_{k=1}^{F} a_{ik} e_{ik}, \tag{1.3}$$

with

$$e_{ik} = \sum_{s=\xi(T_j)}^{i} \frac{\tau_s f_s}{1 + \tau_s f_s} a_{sk}. \tag{1.4}$$

The computational order of implementing the LMM is $O(nF)$ per step by this method (Joshi, 2003). The drift in the model is state-dependent.

The LMM is widely used by practitioners and the academic community for valuing interest-rate derivatives, such as caplets and digital caplets, more importantly exotic derivatives such as target redemption notes, swaps and Bermudan swaptions, among many others. Computing sensitivities of these products via simulation was considered difficult in the past, we present effective methods for computing sensitivities of these products under the LMM in the later chapters. There is also a new modern pricing approach prevailing among practitioners following the global financial crisis. It is based on multiple yield curves reflecting the different credit and liquidity risk of LIBOR rates with different tenors and the overnight discounting of cash flows originated by derivative transactions under collateral with daily margination; for more details of the model, we refer the reader to Henrard (2014) Lee (2013) applied the adjoint algorithmic differentiation to compute pathwise deltas and vegas of interest rate caps and one-way floaters under this new interest-rate modelling framework.The computation of higher derivatives as well as greeks of more complicated products under the multi-curve model is yet to be studied in the future.

1.2.3 Lévy-driven Stock Processes

Although the conventional models used in derivatives pricing assume continuous stock paths, many studies have found evidence of jumps in prices and have advocated the inclusion of jumps in pricing models. Jump models better capture high peaks and heavy tails than the original geometric Brownian motion models, features typical of market data.

A Lévy process has stationary and independent increments, and satisfies the technical requirement that $X(t)$ converges in distribution to $X(s)$ as $t \to s$. The stationarity of the increments means that $X(t+s) - X(s)$ has the distribution independent of $X(t)$. Finite Lévy processes can be represented as the sum of a deterministic drift, a Brownian motion, and a pure-jump process independent of the Brownian motion (Sato, 1999), i.e.

$$X(t) = bt + \sqrt{A}W(t) + \sum_{k=1}^{N(t)} Y_k,$$

where A and b are constants, Y_k are i.i.d. jump sizes and N is a Poisson process.

In applying Monte Carlo simulation, generating paths of the underlying Lévy process can present a challenge, particularly for the ones with infinite activities. For processes with such features, the jumps can no longer be described through a compound Poisson process and must instead be described through a Lévy measure. A general treatment of simulation methods for infinite-activity Lévy processes can be found in Schoutens (2003). For cases where the underlying density functions are difficult to work with, rejection techniques are typically used to simulate the process and the efficiency of the algorithm depends on the parameter set-up of the underlying distributions.

In Chapter 5, we consider first- and second-order sensitivities of vanilla and exotic option prices with two examples of Lévy processes, the Normal-inverse-gaussian process (NIG) and the variance-gamma process (VG). Both processes can be written as time varying Brownian motion, i.e. given the random time, the log price is reduced to a lognormal random variable. It has been shown that these extensions of the Black-Scholes model can better capture the skewness and kurtosis of the stock process. The sensitivities computed can help modellers to identify the optimal set of parameters to capture the true stock dynamics.

1.2.4 $M_t|M|1$ queues

Queuing is common in almost all aspects of life, for example, in telephone exchanges, in a supermarket, at a petrol station, at computer systems, etc. To characterize a queuing system, we have to identify the probabilistic properties of the incoming flow of requests, service times and service disciplines. Arrivals may originate from one or several sources referred to as the calling population.

The calling population can be limited or 'unlimited'. The arrival process can be characterized by the distribution of the inter-arrival times of the customers. The service mechanism of a queuing system is specified by the number of servers, each server having its own queue or a common queue and the probability distribution of customer's service time. The other randomness in the system is the service time, which are usually assumed to be independently and identically distributed random variables. The service times, and inter-arrival times are commonly assumed to be independent random variables.

The discipline of a queuing system means the rule that a server uses to choose the next customer from the queue (if any) when the server completes the service of the current customer. Under a first-in-first-out rule, various performance measures can be calculated to assess the efficiency and effectiveness of the system, e.g.

- the expected minimum (maximum) number of customers in system,
- the expected minimum (maximum) waiting times,
- the expected waiting times,
- the expected time spent by a customer in the system.

An $M_t|M|1$ queuing system is one of the simplest non-trivial queue where the requests arrive according to a non-homogeneous Poisson process with unlimited calling population. The inter-arrival times are exponentially distributed random variables, dependent on the time of the last arrival customer. The service times with only one server are also assumed to be independent and exponentially distributed. Simulating $M_t|M|1$ queues is sometimes difficult due to the non-existence of explicit inverse cumulative density function of the inter-arrival time distribution, and rejection techniques are typically used to simulate non-homogeneous Poisson processes. Given a finite-time horizon for the arrival of customers, the performance measures of an $M_t|M|1$ queuing usually exhibit pathwise discontinuities.

1.2.5 Insurer's surplus process

In order to study ruin-related problems of insurance portfolios, the risk theory community considers the excess of income over claims paid, i.e. the insurer's surplus process. In this monograph, we use Monte-Carlo simulation to generate repeated random samples of this process. The mathematical representation of the surplus process is

$$R_t = u + c(t) - \sum_{i=1}^{N_t} X_i, \tag{1.5}$$

where

- u is the initial surplus, or capital,

- N_t is the number of claims up to time t, i.e. a delayed renewal process generated by a sequence of inter-claim times T_i,
$$N_t = \inf\{j \geq 0 : T_0 + \ldots T_j \geq t\},$$

- X_i's are independent and identically distributed non-negative random variables,

- $c(t)$ is the amount of premiums collected by the insurer up to time t.

The classical risk model was the first model introduced to study the behaviour of the above process, which models T_i as independent and identically distributed exponential random variables. Andersen (1957) extended the model by allowing T_i to have arbitrary distributions. For both models, the premium collection is assumed to be

$$c(t) = (1 + \theta)\mu\lambda t,$$

where $\lambda = 1/\mathbb{E}[T_i]$, $\mu = \mathbb{E}[X_i]$ and θ is the premium loading factor. The independent and identically distributed inter-claim times imply that these two models do not describe situations, such as motor insurances, where claim occurrence epochs depend on the time of the year. The non-homogeneous Poisson process is an alternative for this case, such that a typical periodic intensity function $\lambda(t) = a + b\cos(2\pi ct)$ can be used to describe the arrival rate of claims (P. Čížek et al, 2011). For this model, the premium collection is assumed to be

$$c(t) = (1 + \theta)\mu \int_0^t \lambda(s)ds.$$

The claim arrival process per se does not fully explain the underlying dynamics of insurance risks, the surplus process is also a function of the sum of a random number of claims, its tail distribution depends heavily on the tail of the individual claim size distribution. Thus, it is important to choose a distribution which adequately describes the tail behaviour of the claim size. Various distributions have been used to model X_i, including both light-tailed and heavy-tailed distributions. While most of the literature has focused on light-tailed distributions, the nature of most insurance businesses implies more heavy-tailed claim sizes.

1.3 Sensitivity Analysis in Monte-Carlo Simulation

A simulation exercise takes a set of input parameters, which can be classified into

- distributional parameters which appear in the density function of the random variable of interest(denoted by η)

- structural parameters which are parameters of the performance measure function(denoted by ϕ).

By the Law of Large Numbers, the sample average provides an estimate of the expected performance measure function, $\Upsilon(\phi, \eta)$, such that

$$\Upsilon(\phi, \eta) = \mathbb{E}_\eta[g(X, \phi))] = \int g(x, \phi) f(x, \eta) dx. \tag{1.6}$$

by repeatedly generating random variates, x, from the underlying distribution with density function, f, and computing the pathwise performance measure, g, as a function of the generate random variate. Both η and ϕ are subject to changes and errors. Sensitivity analysis investigates the impact of these changes and errors on the conclusion, Υ. Comprehensive reviews of different methodologies are given in Glasserman (2004), and we present a brief review in Chapter 2 of this monograph.

1.3.1 The importance of sensitivity analysis

Sensitivity analysis is of great importance in almost all simulation exercises. When the parameters are estimated, knowing how inaccuracy of the inputs affects the accuracy of the outputs is crucial for decision making. When the parameters are themselves random, sensitivities are essential tools for risk assessment and designing risk mitigation strategies. When the objective is to optimize some quantity as a function of the inputs, this can be facilitated by good estimates of not just the gradient, but also of the Hessian(the square matrix of second-order partial derivatives).

In financial mathematics, option prices can often be observed in the market, but this is not the case for Greeks. The Greeks are the price sensitivities with respect to the parameters of interest. They play an important role in financial risk management, for instance, delta (the price sensitivity with respect to the underlying indicates the number of units of the underlying securities to hold in the hedged position, and Gamma (the second derivative of option price with respect to the underlying) is used to determine the optimal time interval required for rebalancing under transaction costs (Broadie and Glasserman,1996) as well as the profit and loss in a delta-hedging strategy. Thus, fast and accurate methods for computing first- and second-order Greeks are critical for both trading and risk management of investment banks and hedge funds.

In operations research, sensitivity analysis is also an important tool for decision making. For example, operation managers wish to know how sensitive the average waiting time in the hotline queue is to the changes in parameters of the service time distribution. If the time is relatively insensitive to reasonable changes, the decision maker can have more confidence in the application of his solution. In contrast, if the solution is sensitive to changes, more precise assumptions on the basis of the parameters are required. Many researchers in management science require derivative estimates to determine the optimal set-up for their operation experiments. Mathematically, for twice-differentiable functions, critical points can be found by finding the points where the gradient of the objective function is zero. If the Hessian is positive definite at a critical point, then the point is a local minimum; if the Hessian matrix is negative definite, then the point is a local maximum.When the objective function is convex, then any local minimum will also be a global minimum.

Sensitivity analysis in ruin theory has rarely been studied in my understanding. Nevertheless, it can provide helpful information to insurers for selecting their insurance and investment portfolios as well as designing their risk management strategies for meeting prudential regulations. For example, insurance companies set aside risk-based capital according to prudential regulations to ensure the probability of ruin is under a certain level. Thus, interesting questions to ask are, "how sensitive is the regulatory capital to the arrival rate of claims?", and "is it more sensitive to the investment returns or is it more sensitive to the average claim size?".

1.3.2 The difficulties of computing sensitivities

While sensitivity analysis is of crucial importance in financial mathematics, operations research and ruin theory, the computation of sensitivities is difficult. This is because analytical solutions of these problems do not exist. Although Monte Carlo sensitivity analysis has been explored extensively in the past, some obstacles are still left to be solved.

The finite differencing method is the simplest solution: one simply bumps each parameter one by one and revalues. However, it produces biased estimators and requires balancing between bias and variances. Multiple Monte Carlo simulations with different inputs are then required to estimate the sensitivities to different parameters. When the underlying number of parameters n is large, this method is very time-consuming since we need to run at least $n+1$ simulations. The likelihood ratio and the pathwise methods differentiate the integrand of equation (1.6) directly. Therefore, they are also called the direct methods, and produce unbiased estimators. These methods stand as the conceptual framework for other models developed in the area.

As pointed out by Glasserman(2004), the pathwise method when applicable produces the smallest variances among the three. However, it requires the pathwise performance measure to be Lipschitz continuous everywhere and differentiable almost surely in the parameter of interest in order to compute the first-order derivatives. The application of the pathwise method to compute second-order derivatives is limited to a smaller class of functions. Now, we define this class of functions.

Definition 1.3.1. *Let $\hat{C}^2$ denote the class of functions that*

- *have Lipschitz continuous first-order derivatives everywhere,*
- *are twice differentiable almost surely.*

Thus, for some practical sensitivity estimation problems, the pathwise method is not applicable since it requires the smoothness of the pathwise performance measure function as well as the smoothness of the simulation algorithm for the state variables.

The likelihood ratio method, on the other hand, is often applicable for financial Greek estimations as long as the density function is non-singular. However, the likelihood ratio method often produces derivative estimates with large variances as the number of random variables depending on the parameters of interest increases. For applications in other fields, the density function of the underlying distribution is sometimes unknown or the computation of the score function is not feasible. Consequently, the application of the likelihood ratio method is not applicable for such cases.

Estimating second-order derivatives is far more difficult than estimating first-order derivatives. In particular, the pathwise method is rarely applicable to second-order greek estimations since the payoff functions usually do not fall into the $\hat{C}^2$ class; the likelihood ratio method, on the other hand, results in even larger variance. Glasserman (2004) suggested using a combination of the pathwise method and the likelihood ratio method to compute second-order Greeks, *the pathwise likelihood ratio method*, for a special class of payoffs. Here, we shall define a class of functions.

Definition 1.3.2. *We say a function is angular if*

- *it is Lipschitz continuous everywhere and differentiable almost surely,*
- *and it does not have Lipschitz continuous first-order derivatives.*

The call options are among the most liquidly traded derivatives in the market, and their payoff function is angular. Although its first- and second-order greeks can be derived analytically under the Black-Scholes model, they are not easily computable under other modelling assumptions such as the Variance-Gamma process. For products with angular payoffs, the pathwise likelihood ratio method is better in terms of reducing the standard errors than using the pure likelihood ratio method, but it requires the payoff function to be Lipschitz continuous. This approach also suffers from the same problems as the likelihood ratio method for first-order greeks.

The focus of this monograph is to introduce new simulation methods to compute second-order sensitivities of expected performance measure under various stochastic set-ups. We first address the issue of irregularities of the pathwise performance as a function of the state variables and then consider the case where the underlying distribution has intractable or cumbersome densities. Ultimately, we construct a new simulation algorithm for computing the pathwise performance measure that admits the application the pathwise method to compute unbiased second-order derivative estimates.

In addition, the computational cost of such calculations in a multi-dimensional setting has made it too cumbersome to apply the traditional methods. Consider a multi-asset option depending on 30 different underlying assets, the cross-gammas form a 30×30 Hessian matrix. To compute the Hessian, using the finite differencing method would involve re-calculations for 465 different entries. To ensure that the computational cost is reasonable, we therefore adapt the Algorithmic Hessian approach introduce by Joshi and Yang (2011). The basic idea of their method is explained in Chapter 2 of this monograph.

1.4 Structure of the monograph

This monograph is organized as follows. Chapter 2 reviews the traditional Monte Carlo derivative estimation methods. We introduce efficient measure changes to compute second-order sensitivities of financial products with discontinuous or angular payoffs in Chapter 3. Chapter 4 investigates the pricing and greeking of exotic options with early-exercise features such as Bermudan swaptions and cancellable swaps in the LIBOR market model. While the measures in Chapter 3 are defined in terms of the points of discontinuities, Chapter 5 of this monograph consider the cases where these critical points are inexplicit . We present a generic method for computing first- and second-order derivatives of stochastic systems simulated by rejection techniques in Chapter 6. In particular, we make application of this new approach to computing first- and second-order derivatives with Levy-driven underlyings and queuing expected performance measures. Chapter 7 extends the algorithms developed in Chapter 3 and 5 to the case of computing sensitivities of ruin probabilities. We address the problem of estimating regulatory capital as well as its sensitivities with respect to underlying distributional and structural parameters. We conclude the monograph in Chapter 8.

Chapter 2

Monte-Carlo Sensitivity Review

Derivative estimation in Monte Carlo simulation presents both theoretical and practical challenges. This chapter reviews methods for estimating sensitivities of expectations in the literature. We start with the traditional Monte-Carlo derivative estimation methods: the finite differencing method, the pathwise method and the likelihood ratio method. We then review more advanced techniques such as the partial proxy methods, the optimal partial proxy method, the Algorithmic Hessian approach, the Hong-Liu method, the measure-valued differentiation and the Malliavin calculus approach. Most of these methods focus on computing first-order sensitivities. For the ones that compute second-order derivatives, they generally have strict requirements on the underlying modelling framework.

2.1 Finite Differencing

To illustrate the basic idea of the finite differencing method, we consider an expected performance measure, $\Upsilon(\phi, \eta) : \mathbb{R}^2 \to \mathbb{R}$. Let $X(\eta)$ denote the random variable that depends on the parameter, η. We are interested in the sensitivity of this function with respect to ϕ. For a small bump size, $h \in \mathbb{R}$, the method is to simulate n independent realizations of the performance measure at the parameter value $\phi + h$, and re-simulate n independent additional replications at parameter value $\phi - h$. We can obtain the average from both sets of simulations, $\bar{g}(X(\eta), \phi + h)$ and $\bar{g}(X(\eta), \phi - h)$. The resulting central difference estimator of the first-order derivative,

$$\frac{\bar{g}(X(\eta), \phi + h) - \bar{g}(X(\eta), \phi - h)}{2h}, \tag{2.1}$$

has expectation

$$\frac{\Upsilon(\phi + h, \eta) - \Upsilon(\phi - h, \eta)}{2h}.$$

Similarly, the central-difference estimator of the second-order derivative is given by

$$\frac{\bar{g}(X(\eta), \phi + h) - 2\bar{g}(X(\eta), \phi) + \bar{g}(X(\eta), \phi - h)}{h^2}. \tag{2.2}$$

Assuming third-order differentiability of $\Upsilon(\phi, \eta)$ as a function of ϕ, the bias of the first-order estimator is

$$\frac{1}{6}\frac{\partial^3 \Upsilon(\phi, \eta)}{\partial \phi^3} h^2 + o(h^2).$$

Thus, by choosing a smaller h, we can reduce the bias. However, the variance of the estimator is,

$$\frac{\mathrm{Var}[\bar{g}(X(\eta), \phi + h) - \bar{g}(X(\eta), \phi - h)]}{h^2}.$$

In the above equation, the factor h^2 in the denominator alerts us to possibly disastrous consequences of taking h to be very small. In general, the variance of the estimate does not blow up as h tends to zero if

$$\mathrm{Var}[\bar{g}(X(\eta), \phi + h) - \bar{g}(X(\eta), \phi - h)] = \mathcal{O}(h^2).$$

This requires not only the same random numbers for the two simulations, but also that for almost all values of the random numbers, the pathwise estimate is continuous in the parameter of interest (Glasserman, 2004).

The virtue of the finite differencing method is its simplicity; it requires less examination of the model dependence on the parameter of interest. However, for performance measures with pathwise discontinuities, this method often produces results with very large variances given a small parameter increment. For example, when a digital-call option is close to the expiry, a small bump may cause the path to cross the strike and result in a very different estimate of the payoff. In addition, if we want to further estimate the derivative of Υ with respect to η, we then need to at least perform another re-simulation with a sample of n paths. Therefore, the number of re-simulations increases linearly with the number of parameters of interest. The computational cost is intolerable for cases where a large number of parameters are considered.

2.2 The Direct Methods

Here we discuss the likelihood ratio method and the pathwise method in a unified view as represented by L'Ecuyer (1990). They are both derived from the integral representation of expectations such that

- the pathwise method differentiates the performance function, g, by considering the random variables independent of the parameters of interest such that they are "pushed into" g,

- the likelihood ratio method differentiates the density function by choosing the random variable so that the parameters of interest appear in its density function but not the performance measure.

These direct methods were originally studied for the derivative estimation of discrete-event systems, and was introduced by Broadie and Glasserman (1996) to financial applications. They showed that the main advantages of them over the finite-differencing method is increased computational speed and the ability to produce unbiased derivative estimates.

The likelihood Ratio Method

The derivative price $\Upsilon(\phi, \eta)$ can be taken as the expectation of some function $g(x, \phi)$, and x is the sample outcome with respect to some probability measure space $(\Omega, \sum, P_\eta)$. Therefore we can express $\Upsilon(\phi, \eta)$ as

$$\Upsilon(\phi, \eta) = \int_\Omega g(x, \phi) dP_\eta(x).$$

Here, we consider the derivative of this expectation with respect to η. Typically, the order of differentiation and integration is not interchangeable, since the probability measure, P_η, depends on η. To eliminate this dependence, we can perform a change of measure using a new probability measure, P, which is independent of η and dominates P_η. That is, for every measurable set B, we have

$$P(B) = 0 \Longrightarrow P_\eta(B) = 0.$$

Now using the Radon-Nikodym theorem, we can rewrite the integral as

$$\Upsilon(\phi, \eta) = \int_\Omega g(x, \phi) L(P, \eta, x) dP(x),$$

where $L(P, \eta, x) = \frac{dP_\eta}{dP}$ is the Radon-Nikodym derivative of P_η with respect to P. The sampling is now performed under P, and $g(x, \phi)L(P, \eta, x)$ is the new estimator for the price.

Provided that the new integrand is smooth, we can change the order of differentiation and integration. The price sensitivity with respect to η is

$$\frac{\partial \Upsilon(\phi, \eta)}{\partial \eta} = \int_\Omega g(x, \phi) \frac{L(P, \eta, x)}{\partial \eta} dP(x). \tag{2.3}$$

When the above equation holds, we can estimate the gradient at any value of η. Here, $\frac{L(P,\eta,x)}{\partial \eta}$ is called the score function.

The application of the likelihood ratio method is, however, limited by two features: it requires explicit knowledge of the relevant probability densities and their smoothness, and its estimates often have large variance. For computing greeks of financial products, the likelihood ratio method is often applicable. However, the resulting estimator sometimes exhibit an explosion in variance. For example, consider the simulation of a path $S(0), S(T_1), ..., S(T_N)$ of a geometric Brownian motion, and the estimation of vega. Every transition depends on the volatility parameter, σ. Thus,as the score used to estimate vega has as many terms as there are transitions. Summing a large number of terms in the score produces an likelihood ratio estimator with a large variance. Thus, the variance of the gradient estimator increases linearly with the number of random variables which η has influence upon. In addition, the likelihood ratio method is not applicable when the reduced factor model is used for derivative products with multi-dimensional underlyings.

Example 2.2.1. ***Likelihood Ratio Delta:*** *To demonstrate the idea of the likelihood ratio method, we use the simple example of computing call option deltas under the Black-Scholes model. To estimate the delta using the likelihood ratio method, we need to choose $S(0)$ as the random variable of interest, so $S(0)$ becomes a parameter of the density function. The payoff function is*

$$g(x, S(0))) = e^{-rT}(x - K)_+.$$

The unbiased estimator of delta for a call option is

$$e^{-rT}(S_T - K)_+ \frac{Z}{S(0)\sigma\sqrt{T}},$$

where Z is a simulated standard normal.

The form of the option payoff in this example is actually irrelevant; any other function of $S(T)$ would result in an estimator of the same form. This is a general feature of the likelihood ratio method that the form of the estimator does not depend on the details of the performance measure function, g. Once the score is calculated for a given model, it can be multiplied by many different performance measure functions to estimate their sensitivities.

The Pathwise Method

A key point of L'Ecuyer (1990) concerns the choice of the sample space. In fact, for a given simulation model, there can be different ways of defining the sample space and the random variable. Different choices may lead to different derivative estimators, some being more efficient than others. The pathwise method is another direct approach, which often produces estimators with smaller variances than the likelihood ratio method. The basic idea of the pathwise method is to choose the sample space so that the density function of the random variable is independent of η, e.g. the density function of standard normals. Consequently, the parameter is pushed into the performance measure function

$$\Upsilon(\phi, \eta) = \int_\Omega g(\phi, \eta, x) f(x) dx,$$

where $f(x)$ is the probability density function of the standard normal distribution. Provided that $g(\phi, \eta, x)$ is *differentiable almost everywhere and Lipschitz continuous everywhere* as a function of η, the derivative becomes

$$\frac{\partial \Upsilon(\phi, \eta)}{\partial \eta} = \int_\Omega \frac{\partial g(\phi, \eta, x)}{\partial \eta} f(x) dx. \tag{2.4}$$

This means that not only the performance measure function needs to be Lipschitz continuous as a function of the state variables, but that the simulation algorithm for evolving the state variables is also required to be smooth.

Example 2.2.2. ***Pathwise Delta****: For call option deltas under the Black-Scholes world, we push the parameter of interest into the payoff function,*

$$g(u, S_0) = e^{-rT}(S_0 e^{(r-0.5\sigma^2)T+\sigma\sqrt{T}\Phi^{-1}(u)} - K)_+,$$

where Φ denote the cumulate density function of a standard normal random variable. The pathwise estimator becomes

$$\frac{S_T}{S_0}e^{-rT}\mathbb{I}\{S_T > K\}.$$

For financial products, the payoff function is often discontinuous or angular, so that the application of the pathwise method is limited. The likelihood ratio method, on the other hand, is often applicable as the underlying density functions usually satisfy the smoothness conditions. In fact,we can always choose the random outcome to be the performance measure function itself, where all the parameters of interest are pushed into the density for the application the likelihood ratio method. However, it is often difficult to find the density function of such a complex random variable.

It is also important to notice that the pathwise method can also be viewed as the limit of the finite difference method, therefore they produce the same variance at the limit. Numerical results (L'Ecuyer, 1990) have been used to compare the three traditional methods of derivative estimation. The finite differencing method can be as good as the pathwise method when the parameter of interest has only one dimension. However, the computational cost of the finite differencing method increases linearly with the number of parameters of interest.

2.3 Algorithmic Gradients and Hessians

Computing fast and accurate sensitivities i of central importance to large organizations. In recent years, much progress has been made by using the automatic differentiation method to compute first- and second-order sensitivities. These developments are based on the assumption that the underlying performance measure is smooth such that it satisfies the application of the pathwise method discussed in Chapter 1.

The core idea of this method is to apply a chain-rule based technique to very simple mappings.

Definition 2.3.1. *Elementary operations refer to a vector mapping that is an identity mapping in all coordinates except one; in that coordinate, it only depends on one or two coordinates.*

There is a general result that if a function of n variables can be computed with L elementary operations, then all of its first-order derivatives can be computed with a maximum of $4L$ operations, see Griewank and Walther (2008) for a detailed review of these techniques. This result is much better than approximating using finite-differencing. The adjoint version of automatic differentiation relies on doing a backwards computation, repeatedly applying the chain rule to each elementary operation in the program.

Giles and Glasserman (2006) introduced automatic differentiation to the financial mathematics community for computing first-order derivatives. Joshi and Yang (2011) extended these methods for computing second-order derivatives in an algorithmic fashion. In this monograph, we use their algorithmic Hessian Approach to reduce the computational cost of computing second-order sensitivities. The numerical examples demonstrate that the speed of computation is consistent with the Joshi-Yang result.

2.3.1 Pathwise Delta in LMM

Giles and Glasserman (2006) studied the hedging of exotic interest-rate derivatives using the lognormal LMM. In a model with N steps and n rates, they were able to demonstrate that all the deltas (the first derivative with respect to the initial rates) could be computed with order $\mathcal{O}(Nn)$ computations per Monte Carlo path. Here, we briefly illustrate the idea. The algorithm for computing the value of the derivative under the LMM can be decomposed as follow

$$g \circ K_{N-1} ... \circ K_1 \circ K_0$$

where g is the discounted payoff function at time T_N, and $K_j : R^M \to R^M$ is the evolution of the n rates from time T_j to T_{j+1}. To compute the pathwise deltas, we start from the payoff function,

$$\frac{\partial g}{\partial f(0)} = \frac{\partial g}{\partial f(T_N)} \triangle(N), \tag{2.5}$$

where $\triangle(N)$ is the $n \times n$ matrix such that

$$\triangle(N) = \frac{\partial f(T_N)}{\partial f(0)}.$$

We can decompose the matrix $\triangle(N)$ as the product of N matrices $D(j)$ where

$$D(j) = \frac{\partial f(T_{j+1})}{\partial f(T_j)},$$

then equation (2.5) becomes

$$\frac{\partial g}{\partial f(0)} = \frac{\partial g}{\partial f(T_N)} D(N-1) D(N-2) \cdots D(0) \triangle(0).$$

The computation of delta starts from time T_N, and update the pathwise derivative backwards. Thus, to apply this method for computing first-order derivatives, it requires the smoothness of both the function g as well as K_j's.

2.3.2 Extension to The Computation of Hessians

The method of Giles and Glasserman (2006) greatly improved the speed of computation for first-order greeks, but their second-order results are less satisfactory. Their application of the automatic differentiation method for computing gammas (i.e., the second-order derivatives with respect to the initial forward rates) is of order $\mathcal{O}(Nn^3)$ for interest-rate derivatives under the LMM. Their first-order results with order $\mathcal{O}(Nn)$ computations per Monte Carlo path suggested the possibility of a better second-order algorithm.

Meanwhile, the computation of Hessians by algorithmic means has long been of interest to the automatic differentiation community. The most popular method currently appears to be that of Christianson (1992). This method involves n separate computations; one for the gradient of each first-order partial derivative. The essential idea is to differentiate the algorithm for computing the value in a forward fashion by differentiating each elementary operation. This yields an algorithm for computing the first-order partial derivative which can then be differentiated in adjoint fashion to give its gradient. Much research has addressed the question of how to automatically prune the computational graph in order to take account of the sparseness of the Hessian for particular cases. See Griewank and Walther (2008) for a recent overview.

A breakthrough was made by Joshi and Yang (2011). They computed the whole Hessian matrix in an efficient way using algorithmic differentiation. In contrast to the previous methods, their method computes all first- and second-order derivatives simultaneously in an adjoint fashion. They show that if the Hessian with respect to the M state variables of a function $G : R^M \times R^p \to R$ is known, it is possible to compute the Hessian of $G \circ F$ with an elementary operation, K, by overwriting the Hessian of G with $AM + B$ additional operations. Here A and B are constants depending on the class of elementary operations. As a consequence of such elementary operations, the Hessian can be obtained from the old one by updating one or two rows without recalculation.

As the speed of this algorithm depends the number of state variables, a crucial part of the effective use of this approach is to write algorithms in such a way that the number of state variables is minimized. Essentially, they are variables that depend on the initial varying inputs, and necessary to support the evolution at each step. For example the reduced factor LIBOR market model has $M = n + F + 2$ state variables, where the first n elements are the forward rates, then F elements of e_i, and another two state variables to assist in the evolution process. After defining the state variables, the Hessian is initiated at time T. The Hessian $\underline{H}_g$ can then be initiated at time T_n. The fundamental idea of the algorithmic Hessian lies in the fact that once the Hessian is initialized, the evolution of state variables at each time T_j is a computer program function, that can always be decomposed into a string of elementary operations. The overall derivative can then be decomposed using the chain rule, evolving backwards one elementary operation at a time.

The main achievement of this method is reducing the order of computation per step from

$O(n^3)$ by Giles and Glasserman(2006), to $O(n^2F)$; typically n the number of evolving rates is 40 and F is just 5. The computational burden is greatly reduced. Numerical tests have been done to demonstrate the effectiveness of the approach on various interest-rate derivatives, that computing n(n+1)/2 gammas in the LIBOR market model takes roughly n/3 times as long as computing the price.

The algorithmic Hessian approach was designed for smooth mappings. For options with discontinuous payoffs, they adopted the smoothing method to approximate the discounted payoff function. This function is used at the point of discontinuity as an approximation of the Indicator function. The smoothing approach is related to the method of Liu and Hong (2011), where the gaussian kernel smoothing is used at the point of discontinuity.

2.4 The Hong-Liu Method and the Liu-Hong Method

The automatic differentiation method for reducing computational cost requires the pathwise estimator of the expected performance measure to be Lipschitz continuous for computing first-order derivatives. This is not the cases for some practical problems where derivatives estimates are required. Hong and Liu (2010) considers the probability of the random performance exceeding a certain value. This expected performance measure is the expectation of an indicator function, which fails to satisfy the conditions for the application of the pathwise method. They introduced a new approach for computing their first-order sensitivities. They show that the estimator is consistent and follows a central limit theorem for simulation outputs from both terminating and steady-state simulations. Here, we briefly summarize the method.

Let $L(\eta, X)$ denote the random performance of interest, that depends on the parameter of interest, η. Define the probability

$$p_y(\eta) = \mathbb{P}\big(L(\eta, X) \leq y\big).$$

We are interested in estimating $p_y'(\eta)$. Under the assumptions that $\partial_\eta \mathbb{E}[L(\eta, X)] = \mathbb{E}[\partial_\eta L(\eta, X)]$ holds and p_y is differentiable, they obtained

$$p_y'(\eta) = -f(y, \eta)\mathbb{E}[\partial_\eta L(\eta, X)|L(\eta, X) = y],$$

where f is the probability density function of the random variable,$L(\eta, X)$.

This result is easily computable if the pathwise estimate of $\partial_\eta L(\eta, X)$ is obtained and the density function of the random variable L is known. However, this is often not the case. For most financial context, the underlying performance of interest is often too complex to identify its density function. For example, when computing the sensitivities of a digital-basket option, the density of the underlying basket may be too cumbersome to compute. The authors further addressed this problem by using a combination of the finite differencing method and the pathwise method. That is, the finite-differencing method is used to approximate the density function and the pathwise method for computing $\partial_\eta L(\eta, X)$. Thus, the derivative estimates are biased.

Liu and Hong (2011) further suggested an kernel estimation approach for computing greeks of financial products with discontinuous payoffs. They considered the payoffs in the form of $k(X,\eta)\mathbb{I}_{L(\eta,X)\geq 0}$ where k is a smooth mapping in both X and η. By extending the results of Hong and Liu (2010), they first convert the greek into the sum of an ordinary expectation and a derivative with respect to an auxiliary parameter,

$$p'_y(\eta) = \mathbb{E}[\partial_\eta k(X,\eta)\mathbb{I}_{L(\eta,X)\geq 0}] - \partial_y \mathbb{E}[k(X,\eta)\partial_\eta L(\eta,X)\mathbb{I}_{L(\eta,X)\geq 0}]_{y=0}$$

They noticed the fact that the second term on the right-hand side of the above equation can be estimated by the finite-difference method, which motivated the use of the kernel smoothing. They suggested to use the standard normal density function, Φ, as the kernel such that

$$p'_y(\eta) = \mathbb{E}[\partial_\eta k(X,\eta)\mathbb{I}_{L(\eta,X)\geq 0}] - \lim_{\delta\to 0}\frac{1}{\delta}\mathbb{E}\left[k(X,\eta)\partial_\eta L(\eta,X)\Phi\Big(\frac{L(\eta,X)}{\delta}\Big)\right]_{y=0}$$

Unlike the likelihood ratio method, this kernel smoothing approach does not require explicit knowledge of the underlying density function, which makes it easy to implement in practice. However, the bias depends on the chosen δ with the order of convergence of less than $\frac{1}{2}$.

2.5 Partial Proxy Methods

Partial proxy methods were introduced to cope with cases involving singularities and limited supports. In these methods bumped paths are simulated using schemes that are designed so that the bump does not cause the crossing of a singularity. Instead, some constraints are imposed and the bumped scheme is modified to leave these invariant. As long as the discontinuity arises from a function of the constrained quantity, jumps are eliminated.

Fries and Joshi (2008) defined a proxy constraint function $p_i : \mathbb{R}^n \to \mathbb{R}$ at each step that represents the quantity that will give raise to pathwise discontinuities at T_i. They suggested to perform a measure change on the normal random variables for evolving the state variables, so that the proxy constraint function of the bumped path in the new scheme finishes on the same side of discontinuity as the unbumped path.

Chan and Joshi (2013) pointed out that a naive measure change may result in an unstable Monte-Carlo weight and hence, this may increase the variance of the derivative estimate. Based on this observation, they introduced a new regularization scheme. The essential difference between the two schemes is that the Chan-Joshi method, known as minimal partial proxy, chooses the measure change to minimize the variance of the score function amongst all Gaussian changes of density that do not cause crossing of the discontinuity, whereas the Fries-Joshi approach is to use the likelihood ratio along the normal line to the proxy constraint's level sets.

Since proxy methods lead to stable behaviour for small bumps, it is possible to analyze the behaviour as the bump size goes to zero and derive the limit proxy method. This is in similar vein to how the pathwise method is related to finite differencing in the continuous case. These limit proxy methods were studied in Chan and Joshi (2013) and shown to be effective.

The minimal partial proxy method is still not satisfactory. For example, the minimal partial proxy scheme requires there to be only one discontinuity per step and so cannot always be used. Further more, the limit version of the minimal partial proxy scheme was not derived in the case that the discontinuities were determined by a non-linear function. It therefore does not apply to a basket option in the Black-Scholes model nor to a TARN when the reference rate is a swap-rate in the LMM.

2.6 Optimal Partial Proxy Method

Chan and Joshi (2015) introduced a new approach to compute sensitivities of discontinuous integrals. The minimal partial proxy scheme is minimal amongst Gaussian schemes but need not be so amongst all schemes. The variance of the likelihood ratio under the optimal partial proxy method is minimal amongst all measure changes and that it is therefore in a certain sense optimal. It is generic and applicable to a wide range of problems. For simplicity, we only present the one-step and one-dimensional case and the general case follows from this basic example. Consider the parameter of interest, θ, which can be either a distributional parameter or a structural parameter. The one-step and one-dimensional integral by choosing $x \in U(0,1)$ can be expressed as

$$\Upsilon(\theta) = \int_0^1 I_{[a_0,a_1]}(x(\theta,u))g(x(\theta,u))du.$$

Here x is the algorithm for turning a random uniform into the state variable, and g is the performance measure function. There is a mild condition on x that there exist differentiable functions $a_j(\theta)$'s such that

$$x(\theta,u) \in [a_0,a_1] \Rightarrow u \in [a_0(\theta),a_1(\theta)], \tag{2.6}$$

then the above integral becomes

$$\Upsilon(\theta) = \int_{a_0(\theta)}^{a_1(\theta)} k(\theta,u)du, \tag{2.7}$$

where

$$k = g(x(\theta,u)).$$

At this point, if we apply the Leibniz rule to equation(2.7), the derivative becomes

$$\Upsilon'(0) = a_1'(0)k(0,a_1(0)) - a_0'(0)k(0,a_0(0)) + \int_{a_0(0)}^{a_1(0)} \frac{\partial k}{\partial \theta}(0,u)du.$$

The first two terms represent the points where the finite-differencing method passing the discontinuities. The authors suggest using a change of variable to eliminate the dependence on θ at the limit up to the second order, by replacing u with $U(v, \theta)$ in equation(2.7),

$$U(v, \theta) = v + \theta\gamma(v), \tag{2.8}$$

with conditions of

1. $v \in [a_0(0), a_1(0)] \Leftrightarrow U \in [a_0(\theta), a_1(\theta)]$,
2. $U(v, 0) = v$,
3. $a'_j(0) = \gamma(a_j(0))$for $j = 1, 2$.

We can now rewrite the integral

$$\Upsilon(\theta) = \int_{a_0(0)}^{a_1(0)} k(\theta, U(v, \theta))\frac{\partial U}{\partial v}du,$$

so the derivative evaluated at zero becomes

$$\Upsilon'(0) = \int_{a_0(0)}^{a_1(0)} (\frac{\partial k}{\partial \theta}\frac{\partial U}{\partial v} + k\frac{\partial^2 U}{\partial \theta \partial v})du. \tag{2.9}$$

Now the derivative is expressed as a combination of a pathwise part and a likelihood ratio part. In the above equation $\frac{\partial U}{\partial v}$ is the pathwise coefficient, while $\frac{\partial^2 U}{\partial \theta \partial v}$ is the LR score function.

The mild conditions give us the flexibility to choose the function in a way which minimizes the variance of the Monte-Carlo implementation. Since the pathwise method when applicable produces a lower variance than the likelihood ratio method, it is preferable to use as much pathwise method as possible. Although this can only be done in a case by case basis, Chan and Joshi (2015) present a special case where the continuous part of the payoff function is constant, and the optimal choice for γ is a piecewise linear function to minimize the variance of the likelihood ratio part.

2.7 The measure-value differentiation method

The likelihood ratio method differentiates the density function, while the pathwise method differentiates the performance measure. Another approach of derivative estimate is to differentiate the underlying measure, called measure-valued differentiation (Heidergott et al, 2010). Their result for estimating first-order derivative is unbiased. Here, we briefly talk about this method.

Measure-valued differentiation inspired by the concept of weak differentiation (Pflug, 1996) critically depends on the differentiability of the measure, μ_η, that the parameter has influence upon. If μ_η is $\mathcal{D}$-differentiable, then

$$\frac{d}{d\eta}\int g(x)\mu_\eta(dx) = c_\eta\Big(\int g(x)\mu_\eta^+(dx) - \int g(x)\mu_\eta^+(dx)\Big),$$

where μ_η^+ and μ_η^- are the Hahn-Jordan decomposition of μ_eta, and they are also probability measure. The representation as the difference of two probability measures $\mu_\eta' = c_\eta(\mu_\eta^+ - \mu_\eta^-)$ is not unique. Although the constant c_η is usually minimal in terms of minimizing the variance of Monte Carlo estimates if the two parts are orthogonal, this is not guaranteed for all performance measures.

This measure-valued derivative is applicable to wide range of distributions such as normal, exponential and gamma. However, this method requires sampling from different versions of the distribution for each parameters of interest. For example, to compute the sensitivity of a call option with respect of the risk free rate, it requires

$$\frac{\partial\mathbb{E}\Big[\exp^{-rT}\big(S(T)-K\big)_+\Big]}{\partial r} = c_r\left(\mathbb{E}^+\Big[\exp^{-rT}\big(S(T)-K\big)_+\Big] - \mathbb{E}^-\Big[\exp^{-rT}\big(S(T)-K\big)_+\Big]\right),$$

such that $\mathbb{E}^+$ and $\mathbb{E}^-$ requires sampling different versions of $S(T)$ under different distributions depending on the riskfree interest rate. Thus, it exhibit a considerable amount of computational effort in order to compute the sensitivities with respect to all parameter inputs.

A particular implementation of the estimation is to use the same underlying random variable to drive the evolution of the pair, $S(T)^+$ and $S(T)^-$ at each sample path. Numerical examples showed that coupling via common random numbers, however, is not necessarily optimal (in terms of variance reduction) for every performance function g. Furthermore, as the number of random variables depending on the parameter increases, the derivative estimate can also suffer the problem of large variances.

2.8 Malliavin calculus for Greek computation

Another important approach is the Malliavin calculus, that extends the calculus of variations from functions to stochastic processes. It allows the computation of derivatives of random variables. Fournie et. al (2001) pioneered in the field and showed how to use the Malliavin calculus to compute greeks for a small number of European options. Benhamou (2003) extended the method and applied the Malliavin calculus to compute greeks of more financial products.

To illustrate the idea, we work with a one-dimensional process, and it is of the form

$$X_T(\eta) = X_0(\eta) + \int_0^T \mu(X_t(\eta),\eta,t)dt + \int_0^T \sigma(X_t(\eta),\eta,t)dW(t),$$

where $X_t(\eta) \in \mathbb{R}$. The greek estimator can be derived from the integration by parts formula (Malliavin, 1999),that is the product of the payoff, g, and the Malliavin weighting, W,

$$\frac{d}{d\eta}\mathbb{E}\big[g(X(\eta))\big] = \mathbb{E}[g(X(\eta))W].$$

There are many weighting functions that satisfy the conditions for the above equation to hold. However, the general optimal weighting function for reducing the variance of theMonte Carlo implementation is not found in the literature. Benhamou (2003) introduced the weighting function generator defined as its Skorohod integrand, and showed that these weighting functions have to satisfy necessary and sufficient conditions expressed as conditional expectations with respect to the stochastic trajectory. The weighting function with minimal variance was also presented. It is worth noticing that the set of the Malliavin weighting function is defined deliberately independent from the payoff function. The optimal solution in the sense of the total variance crucially depends from the payoff and the state variables.

In the case of an explicit density function, the likelihood ratio method showed that the greeks can be written as the expectation of the payoff function times the likelihood ratio terms. However, the derivation of the likelihood ratio weight requires the explicit knowledge of the density function while the Malliavin calculus method enables us to derive the weighting with much less information. Like the likelihood ratio method, it can only applied to full-factor models.

2.9 Conclusion

While computing first-order derivatives via Monte Carlo simulation has been studied extensively in the past, computing second-order sensitivities is a much more complicated issue. For the pathwise method to work for second-order derivatives, it requires the pathwise performance measure function to be $\hat{C}^2$. Options with discontinuous and angular functions fall out of this category. The application of the likelihood ratio method for second-order derivatives is sometimes valid; however, they often exhibit large variances. In the following chapter, we construct alternative $\hat{C}^2$ algorithms for computing performance measures, which consequently admit the application of the pathwise method to computing first- and second-order derivatives.

Chapter 3

Optimal Partial Proxy Method for Computing Gammas of Financial Products with Discontinuous and Angular Payoffs

3.1 Introduction

Financial institutions have significant exposures to derivative products with complex structures. For example, insurance companies sell products with embedded options as part of their guarantee to the insured, banks use derivatives to protect themselves from interest-rate changes, and hedge funds use derivatives to manage investment risks or to speculate on the future direction of markets. Fluctuations in the underlying asset prices will affect derivative prices directly, which may materially impact on the balance sheet of the financial institution. Therefore, it is important for the financial institutions to manage the risks underlying their derivative exposures.

In this chapter, we extend the OPP method (Chan and Joshi, 2015) to computing second-order greeks. That is, a sequence of measure changes is performed to eliminate the limits dependence on the parameter of interest, so pathwise discontinuities of the discounted payoff function and its first-order derivatives are eliminated. The resulting discounted payoff function is $\hat{C}^2$. We can then change the order of differentiation and integration to calculate the Hessian. The change of variable is also optimal in terms of minimizing the variance of the likelihood ratio weight. We call this method, the second-order optimal partial proxy method for calculating Hessians (HOPP(2)).

Unlike discontinuous payoffs, angular payoff functions are Lipschitz continuous everywhere and differentiable almost surely. Thus, the pathwise method is applicable to calculate the first-order greeks for financial products with angular payoffs but not the second-order ones. A different type of measure change function is suggested to remove the pathwise discontinuities of the first-order derivatives. We call this method, the first-order optimal partial proxy method for calculating Hessians (HOPP(1)).

We apply the HOPP(2) and the HOPP(1) schemes to calculate the Hessian of both equity and interest-rate derivatives. For high-dimensional products, the computational burden is another important issue to consider when calculating price sensitivities. The LMM of interest-rate derivatives is one such example. Here, HOPP(2) and HOPP(1) are combined with the algorithmic Hessian approach (Joshi and Yang, 2011) to produce a simulation scheme for calculating the Hessian of derivative products with discontinuous or angular payoffs. The computational order for a path is $\mathcal{O}(nFMI)$ for calculating the Hessian of Target redemption notes, where M is the number of state variables and I is the number of observation dates up to and including the redemption.

The remaining sections of the chapter are organized as follows. The basic idea of HOPP(2) is presented in Section 3.2, and examples of the method applied to the Black-Scholes model are presented. In Section 3.3, we apply the model to calculate the Hessian of financial products with angular payoffs using HOPP(1). In Section 3.4, we present the general framework of the HOPP(2) method. We then apply HOPP(2) to the LMM in Section 3.5, to calculate the second-order greeks of various vanilla and exotic style interest rate derivatives.

3.2 One-dimensional case

3.2.1 One-step and one-dimensional case

In this section, we present the basic idea of the HOPP(2) method. The simple discontinuous payoff we considered here is the same as Section 2.6, which can be expressed as

$$I_{\alpha_0 \leq S \leq \alpha_1} g(S),$$

where g is the twice-differentiable discounted payoff function of S.

The one-step and one-dimensional integral can be viewed as a one-dimensional Monte-Carlo simulation over $u \in U(0,1)$,

$$\Upsilon(\theta) = \int_0^1 I_{[\alpha_0,\alpha_1]}\big(x(\theta,u)\big) g\big(x(\theta,u)\big)\, du.$$

where x is the algorithm for turning a standard uniform u and a parameter θ into the state variable S. Here, θ can be either distributional or structural. We assume that there exist twice-differentiable functions, $a_j(\theta)$, for $j = 0$, 1, such that

$$x(\theta, u) \in [\alpha_0, \alpha_1] \Leftrightarrow u \in \big[a_0(\theta), a_1(\theta)\big]. \tag{3.1}$$

The above integral becomes

$$\Upsilon(\theta) = \int_{a_0(\theta)}^{a_1(\theta)} g\big(x(\theta, u)\big)\, du. \tag{3.2}$$

To remove the limits' dependence on θ for both Υ and Υ', we introduce a change of variable by replacing u with $U(v, \theta)$ in the above equation, where

$$U(v, \theta) = v + (\theta - \theta_0)\gamma(v) + \frac{1}{2}(\theta - \theta_0)^2\delta(v), \tag{3.3}$$

with conditions

- for Υ, we require $\gamma\big(a_j(\theta_0)\big) = a_j'(\theta_0)$ for $j = 0, 1$,
- for Υ', we require $\delta\big(a_j(\theta_0)\big) = a_j''(\theta_0)$ for $j = 0, 1$.

The change of variable function is quadratic in θ, since we consider only the first- and second-order derivatives of the function. The conditions above guarantee that the limits do not move with θ up to the third-order, i.e.

$$\frac{\partial U(a_j(\theta_0), \theta_0)}{\partial \theta} = a_j'(\theta_0),$$

$$\frac{\partial^2 U(a_j(\theta_0), \theta_0)}{\partial \theta^2} = a_j''(\theta_0).$$

We can now expand the integral using Taylor's theorem up to the third-order

$$\Upsilon(\theta) = \int_{a_0(\theta_0)}^{a_1(\theta_0)} g\Big(x\big(\theta, U(v, \theta)\big)\Big) \frac{\partial U(v, \theta)}{\partial v}\, dv.$$

The derivative evaluated at θ_0 is

$$\Upsilon'(\theta_0) = \int_{a_0(\theta_0)}^{a_1(\theta_0)} g'\big(x(\theta_0, v)\big)L(\theta_0, v) + g(x(\theta_0, v)\big)\Omega(\theta_0, v)\, dv, \tag{3.4}$$

with

$$L(\theta_0, v) = \frac{\partial}{\partial \theta}\Big[x\big(\theta, U(v, \theta)\big)\Big]_{|\theta = \theta_0} = \frac{\partial x(\theta_0, U)}{\partial \theta} + \frac{\partial x(\theta_0, U)}{\partial U}\gamma(v),$$

$$\Omega(\theta_0, v) = \frac{\partial^2 U(\theta_0, v)}{\partial \theta \partial v} = \gamma'(v).$$

The expression agrees with the one derived for first-order derivatives in Section 2.6. We further differentiate this expression to obtain the second-order derivative

$$\Upsilon''(\theta_0) = \int_{a_0(\theta_0)}^{a_1(\theta_0)} g''\Big(x\big(\theta_0, U(v, \theta_0)\big)\Big)L^2 + g'\Big(x\big(\theta_0, U(v, \theta_0)\big)\Big)h(\theta_0, v) + g\Big(x\big(\theta_0, U(v, \theta_0\big)\Big)\omega(\theta_0, v)\, dv, \tag{3.5}$$

with

$$\omega(\theta_0, v) = \frac{\partial^3 U(v, \theta_0)}{\partial \theta^2 \partial v} = \delta'(v).$$

and

$$h(\theta_0, v) = \lambda(\theta_0, v) + \upsilon(\theta_0, v),$$

where

$$\lambda(\theta_0, v) = \frac{\partial^2 x(\theta_0, u)}{\partial \theta^2} + 2\frac{\partial^2 x(\theta_0, u)}{\partial \theta \partial u}\gamma(v) + \frac{\partial^2 x(\theta_0, u)}{\partial u^2}(\gamma(v))^2 + \frac{\partial x\big(\theta_0, U(v, \theta_0)\big)}{\partial u}\delta(v),$$

$$\upsilon(\theta_0, v) = 2\frac{\partial x\big(\theta_0, U(v, \theta_0)\big)}{\partial \theta}\frac{\partial^2 u(\theta_0, v)}{\partial \theta \partial v} = 2L(\theta_0, v)\gamma'(v),$$

$$\omega(\theta_0, v) = \frac{\partial^3 U(v, \theta_0)}{\partial \theta^2 \partial v} = \delta'(v).$$

The term $\frac{\partial u(\theta_0, v)}{\partial v}$ is dropped from equations (3.4) and (3.5) since it is one at the limit as discussed in Section 2.6.

The unity of the pathwise and the likelihood ratio methods is discussed by L'Ecuyer (1990). Our approach can be viewed as a combination of the pathwise method and the likelihood ratio method. The change of variable performed is to shift some dependence of θ from the discounted payoff function to the probability density function, so we can apply the pathwise method to the payoff function away from the discontinuities and the likelihood ratio method near the discontinuities. The pathwise method when applicable produces a lower variance than the likelihood ratio method. Therefore, it is preferable to use as much pathwise method as possible.

3.2.2 Optimization

The mild conditions on $\gamma(v)$ and $\delta(v)$ give us the flexibility to choose these functions in a way that minimizes the variance of the Monte-Carlo implementation. The functions γ and δ can be chosen to minimize the variance of the likelihood ratio part, i.e. the variances of η and ω . This is however dependent on the structure of the discounted payoff function, Chan and Joshi (2015) presented a special case where the continuous part of the payoff function is constant, and the optimal choice for γ is a piecewise-linear function for computing first-order derivatives.

Here, we show that affine functions are also optimal for computing second-order derivatives of the expectation of piecewise-constant functions. Under the HOPP(2) scheme, when the integrand is a function being constant between α_0 and α_1, and zero otherwise, the second derivative of the integral can be expressed as

$$\Upsilon''(\theta_0) = \int_{a_0(\theta_0)}^{a_1(\theta_0)} g\Big(x\big(\theta_0, U(v, \theta_0)\big)\Big)\omega(\theta_0, v)dv.$$

Letting K be the constant value that g takes between α_0 and α_1, the equation can be simplified to

$$\Upsilon''(\theta_0) = K \int_{a_0(\theta_0)}^{a_1(\theta_0)} \delta'(v)dv.$$

Our objective is to choose $\delta(v)$ in such a way so that the variance of the second-order derivative estimator is minimized, that is to minimize

$$\int_{a_0(\theta_0)}^{a_1(\theta_0)} \delta'(v)^2 dv.$$

Chan and Joshi (2015) showed that, given that δ' is continuously differentiable on $[a_0(\theta_0), a_1(\theta_0)]$ except at a finite number of points where it is continuous, this expression is minimized globally when δ is linear between $a_0(\theta_0)$ and $a_1(\theta_0)$. With given values of the function δ at $a_0(\theta_0)$ and $a_1(\theta_0)$, we have that the optimal δ is

$$\delta(v) = a_0''(\theta_0) + \frac{a_1''(\theta_0) - a_0''(\theta_0)}{a_1(\theta_0) - a_0(\theta_0)}\big(v - a_0(\theta_0)\big). \tag{3.6}$$

Here, we adopt the same linear function for γ as the first-order OPP scheme,

$$\gamma(v) = a_0'(\theta_0) + \frac{a_1'(\theta_0) - a_0'(\theta_0)}{a_1(\theta_0) - a_0(\theta_0)}\big(v - a_0(\theta_0)\big). \tag{3.7}$$

In general, we may have a discounted payoff function with several discontinuities. For m discontinuities, we assume that there exist a set of twice-differentiable functions, $a_j(\theta)$, for $j = 0, 1, \ldots, m+1$, such that

$$a_0(\theta) = 0 \text{ and } a_{m+1}(\theta) = 1,$$

and for $j = 0, 1, \ldots, m$,

$$a_j(\theta) < a_{j+1}(\theta),$$

where $a_j(\theta)$ is the jth critical value function corresponding to the jth discontinuity of the payoff. We assume the discounted payoff function is $\hat{C}^2$ on (a_j, a_{j+1}) and is $\hat{C}^2$ extendible to $[a_j, a_{j+1}]$. HOPP(2) is applicable to discounted payoff functions with multiple discontinuities, by having different δ and γ functions for each interval $(a_j(\theta_0), a_{j+1}(\theta_0)]$ as shown in equation (3.6) and (3.7).

3.2.3 One-dimensional and one-step examples

The first example we present here is the digital call option in the Black-Scholes model. The analytical formula for the Gamma is:

$$-\frac{e^{-rT}\phi(d_2)}{S_0^2\sigma\sqrt{T}}\Big(1 - \frac{d_2}{\sigma\sqrt{T}}\Big),$$

where ϕ is the standard normal probability density function, and

$$d_2 = \frac{\log(\frac{S_0}{K}) + (r - 0.5\sigma^2)T}{\sigma\sqrt{T}}.$$

From equation (3.5), the HOPP(2) method produces an unbiased estimator of the Gamma, i.e.

$$\mathbb{E}\big[I_{S_T>K} e^{-rT} \omega(S_0, v)\big],$$

with

$$\omega(S_0, v) = \delta'(v) = -\frac{\Phi''(d_2)(\frac{\partial d_2}{\partial S_0})^2 + \Phi'(d_2)\frac{\partial^2 d_2}{\partial S_0^2}}{\Phi(d_2)},$$

where Φ is the standard normal cumulative density function.

The HOPP(2) Gamma of a digital call option becomes

$$e^{-rT}\delta'(v)\mathbb{P}(S_T > K) = \Big[\Phi''(d_2)\Big(\frac{\partial d_2}{\partial S_0}\Big)^2 + \Phi'(d_2)\frac{\partial^2 d_2}{\partial S_0^2}\Big]e^{-rT},$$

this expression agrees with the analytical formula.

We present applications of the HOPP(2) method to vanilla equity products, namely, the digital put option, the put option and the parabolic put option. Here, the payoff of a parabolic put option is defined as

$$\Big((K - S_T)_+\Big)^2$$

where K is the strike of the option and S_T is the stock price at the expiry. Th payoff function of this hypothetical product is $\hat{C}^2$ as a function of the underlying stock value. The pathwise method is applicable to calculate its Gammas, which allows the comparison of the HOPP(2) and the pure pathwise methods. For numerical demonstration, we set $S_0 = 100$, $K = 100$, $\sigma = 0.2$ and $r = 0.05$ per annum. The expiry time is 0.2 years.

The standard errors generated by the HOPP(2) method are compared to the standard errors from the likelihood ratio, the pathwise likelihood ratio and the pure pathwise methods based on 20,000 paths samples, see Table 3.1 and 3.2.

Option	HOPP(2) Mean	HOPP(2) S.E.	LR S.E	PWLR S.E.	PW S.E.
Parabolic Delta	-5.5505	0.0788	0.1248		0.04
Parabolic Gamma	0.8221	0.0099	0.0326	0.012	0.0044
Put Delta	-0.4377	0.0044	0.0069		0.0033
Put Gamma	0.044	0.0004	0.0016	0.0004	
Digital Delta	-0.0439	0.0003	0.0005		
Digital Gamma	0.0008	2.61594E-06	1.77493E-05		

Table 3.1: Vanilla options: means and standard errors of the Delta and Gamma estimates by various methods with 20,000 paths samples

Option	LR/HOPP(2)	PWLR/HOPP(2)	PW/HOPP(2)
Parabolic Delta	1.585		0.5074
Parabolic Gamma	3.289	1.211	0.4406
Put Delta	1.583		0.7477
Put Gamma	4.015	1.083	
Digital Delta	1.397		
Digital Gamma	15.631		

Table 3.2: Vanilla options: standard errors of the Delta and Gamma estimates by various methods divided by standard errors by the HOPP(2) with 20,000 paths samples

We observe that the HOPP(2) method outperforms the likelihood ratio method and the pathwise likelihood ratio method. We perform further investigations on the performance of the four methods with different volatilities. We keep all the parameters the same as the above example, except the volatility. The volatility is varying from 0.1 to 0.5. The comparison of the standard errors of the four methods are illustrated in Figure 3.1.

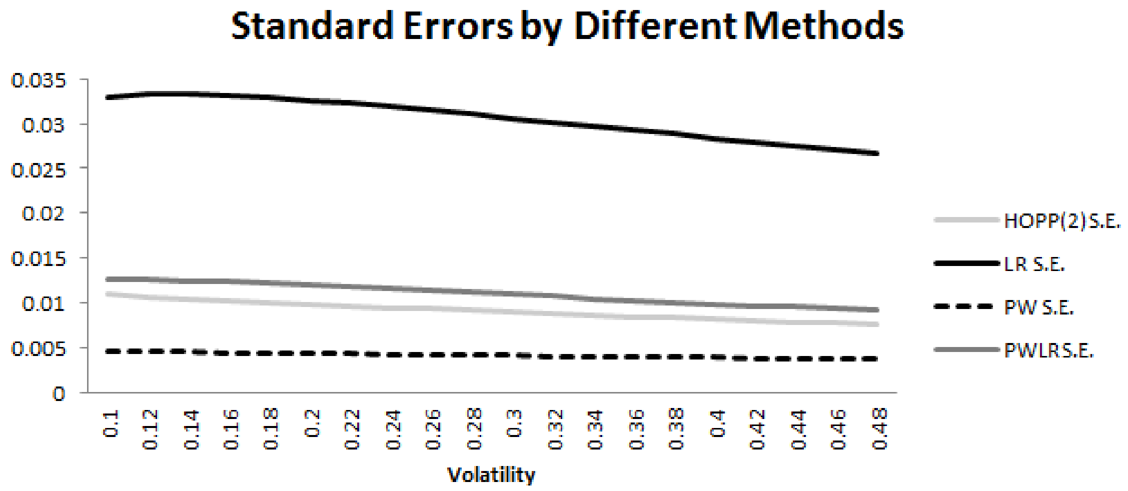

Figure 3.1: Parabolic Put Option: Comparison of standard errors of the Gamma calculated by various methods with varying volatility using 20,000 paths samples

3.2.4 Extension to multi-step cases

To calculate second-order derivatives of the price for products with path-dependent features by HOPP(2), we perform a change of variable at each step where there are pathwise discontinuities. Here, we have a set of N time steps, $T_1, T_2, \ldots, T_N$, and $S_i \in \mathbb{R}$ is the value of the state variable observed at T_i. The change of variable performed on each step is to remove pathwise discontinuities of both the discounted payoff function and its second-order derivatives. We replace the standard uniform, v_i, of step i, with the change of variable function, $u_i(v_i, \theta)$, as shown in Section 3.2.1, which is:

- continuously twice-differentiable as a function of θ,
- and piecewise-differentiable as a function of v_i.

The resulting discounted payoff of the path is,

$$\hat{P}(\theta) = \prod_{i=1}^{N} \frac{\partial u_i(\theta, V_i)}{\partial v} P\Big(S(\theta, U(\theta, V))\Big), \tag{3.8}$$

with the properties:

- $\hat{P}(\theta_0) = P(S(\theta_0, V))$,
- $\mathbb{E}[\hat{P}(\theta)] = \mathbb{E}[P(S(\theta, V))]$,
- $\hat{P}$ is $\hat{C}^2$ as a function of θ.

We differentiate equation (3.8) to obtain the first-order derivative estimator under the HOPP(2) simulation scheme,

$$
\begin{aligned}
\hat{P}'(\theta_0) =& \sum_{k=1}^{N} \frac{\partial P\Big(S\big(\theta_0, U(\theta_0, V)\big)\Big)}{\partial S_k} \prod_{i=1}^{k} \Big(\frac{\partial S_i(S_{i-1}(\theta_0), V_i)}{\partial S_{i-1}} + \frac{\partial S_i\big(S_{i-1}(\theta_0), u_i(\theta_0, V_i)\big)}{\partial u_i} \frac{\partial u_i(\theta_0, V_i)}{\partial \theta} \Big) \prod_{j=1}^{N} \frac{\partial u_j(\theta_0, V_j)}{\partial v} \\
&+ P\big(S(\theta_0, V)\big) \sum_{i=1}^{N} \frac{\partial^2 u_i(\theta_0, V_i)}{\partial v \partial \theta} \prod_{j \neq i} \frac{\partial u_j(\theta_0, V_j)}{\partial v}.
\end{aligned}
$$

We further differentiate this expression to obtain the second-order derivative estimator,

$$
\begin{aligned}
\hat{P}''(\theta_0) =& \sum_{s=1}^{N} \sum_{k=1}^{N} \frac{\partial^2 P\Big(S\big(\theta_0, U(\theta_0, V)\big)\Big)}{\partial S_k \partial S_s} \prod_{i=1}^{k} L_i(\theta_0, V_i) \prod_{i=1}^{s} L_i(\theta_0, V_i) \\
&+ \sum_{k=1}^{N} \frac{\partial P\Big(S\big(\theta_0, U(\theta_0, V)\big)\Big)}{\partial S_k} \Big[\sum_{i=1}^{k} \lambda_i(\theta_0, V_i) \prod_{j \neq i} L_j(\theta_0, V_j) + 2 \prod_{i=1}^{k} L_i(\theta_0, V_i) \sum_{i=1}^{k} \Omega_i(\theta_0, V_i) \Big] \\
&+ P\big(S(\theta_0, V)\big) \Big[\sum_{i=1}^{N} \omega_j(\theta_0, V_j) + \sum_{i=1}^{N} \Omega_i(\theta_0, V_i) \sum_{j \neq i} \Omega_j(\theta_0, V_j)].
\end{aligned}
$$

The functions L_i, λ_i, Ω_i and ω_i are as defined in Section 3.2.1.

We apply HOPP(2) to calculate the Gamma of the discrete barrier option and the digital barrier option. For the Monte-Carlo simulation, we compute Gammas and Deltas for the down-and-out put option with initial stock price and strike 100, and barrier 80. The volatility of the stock is 0.5 and the interest rate is 0.05 per annum. The expiry time is in 1 year with 10 observation dates equally-spaced across $[0, T]$. The discrete barrier option payoffs are path-dependent and discontinuous, as a consequence, the pathwise method is not applicable to these products.

The results from HOPP(2) and the likelihood ratio method converge. The standard errors and the ratio of standard errors are presented in Table 3.3 based on 500,000 paths samples.

Option	HOPP(2) Mean	LR Mean	HOPP(2)S.E.	LR S.E.	LR S.E./HOPP(2) S.E.
Digital Barrier Delta	0.002	0.002	1.45028E-05	2.33001E-05	1.607
Digital Barrier Gamma	-9.359E-05	-9.184E-05	9.52616E-07	1.86469E-06	1.957
Barrier Delta	0.027	0.027	0.000240	0.0003799	1.582
Barrier Gamma	-0.001	-0.001	1.4734E-05	3.00469E-05	2.039

Table 3.3: Down-and-Out Put options: Mean and standard errors of the Delta and Gamma estimates by various methods on samples with 500,000 paths.

One important drawback of the likelihood ratio method is its dependence on the volatility of the underlying stock. The next numerical experiment is based on 50,000 paths samples with the parameters the same as above except

- the number of observation dates is 90,
- and the volatility is varying from 0.05 to 0.5.

The ratio of standard errors computed by HOPP(2) to the standard errors computed by the likelihood ratio method is plotted in Figure 3.2. The ratio is 22.063 when the volatility is 0.05 and 2.823 when the volatility is 0.5.

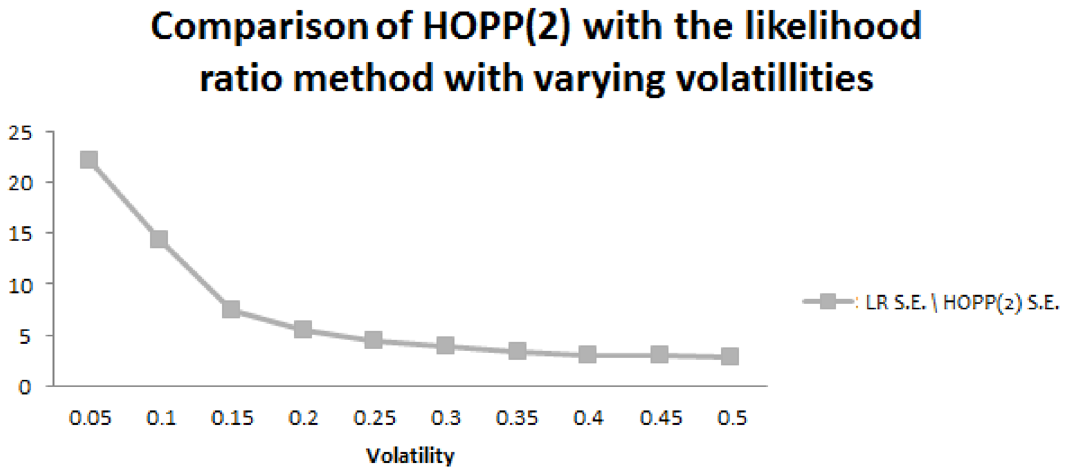

Figure 3.2: Ratio of the standard error computed by the likelihood ratio method to the standard error computed by HOPP(2) for changing volatilities with 50,000 paths samples.

3.3 Application to Derivative Products with Angular Payoffs

For financial products with angular payoffs, both HOPP(1) and HOPP(2) are applicable to compute the second-order derivatives. For illustration,we apply the two methods to calculate the Gamma of a vanilla put option, and compare the standard errors and the computational times. For HOPP(2), the change of variable function is

$$u^{(2)}(\theta, v) = v + (\theta - \theta_0)\gamma(v) + \frac{1}{2}(\theta - \theta_0)^2\delta(v).$$

The HOPP(2) estimator for the Gamma of a vanilla put option is

$$\exp(-rT)\left(- 2\frac{S_T\sigma\sqrt{T}}{\phi(Z)S_0}\gamma(v) - \frac{S_T\sigma\sqrt{T}}{\phi(Z)^2}[\sigma\sqrt{T} - Z] - \frac{S_T\sigma\sqrt{T}}{\phi(Z)}\delta(v) - 2\left[\frac{S_T}{S_0} + \frac{S_T\sigma\sqrt{T}}{\phi(Z)}\gamma(v)\right]\right) I_{S_T<K}$$

$$+\exp(-rT)(K - S_T)_+\delta'(v),$$

where v is the simulated standard uniform, Z is the simulated standard normal random variable and ϕ is the standard normal density function. The functions γ and δ are defined as in Section 3.2.1.

For HOPP(1), the change of variable function is

$$u^{(1)}(\theta, v) = v + (\theta - \theta_0)\gamma(v).$$

The HOPP(1) estimator for the Gamma has a simpler form

$$\exp(-rT)\Bigg(-2\frac{S_T\sigma\sqrt{T}}{\phi(Z)S_0}\gamma(v) - \frac{S_T\sigma\sqrt{T}}{\phi(Z)^2}[\sigma\sqrt{T} - Z] - 2\Big[\frac{S_T}{S_0} + \frac{S_T\sigma\sqrt{T}}{\phi(Z)}\gamma(v)\Big]\Bigg) I_{S_T<K}.$$

The function γ is the same as the one in $u^{(2)}(\theta, v)$.

The two estimators are both unbiased, and converge to the analytic value. Since the HOPP(1) estimator of the Gamma does not involve the term with the payoff and the function δ, we can expect a smaller standard error and a shorter computational time. We perform a Monte-Carlo simulation to calculate the Gamma of an at-the-money put option to compare the two methods. We set $S_0 = K = 100$, $T = 0.2$, $\sigma = 0.2$ and $r = 0.05$ per annum. For a sample of 4,000,000 paths, the standard error is 2.45×10^{-5} calculated by HOPP(2) and 2.34×10^{-5} by HOPP(1). The computation time is 0.239 seconds by HOPP(2) and 0.2 seconds by HOPP(1). Therefore, it appears that for financial products with angular payoffs, it is slightly more efficient to use HOPP(1).

3.4 The multi-step and multi-dimensional case

In our general framework, we allow multiple parameters of interest. The parameter $\theta \in \mathbb{R}^m$ lies within a small open set U_θ about the base point θ_0. We have a set of N time steps, $T_1, T_2, T_3, \ldots, T_N$. At T_j, the evolution of state variables, $S_j \in \mathbb{R}^n$, depends on the parameter θ and F standard uniforms, $V_j \in (0,1)^F$. We divide the evolution across each step into two phases. In the first phase, we perform the ordinary evolution of the state variables except the first standard uniform. In the second phase, we perform the change of variable in a similar way to the one-dimensional case to remove pathwise discontinuities of both the discounted payoff function and the first-order derivatives of it.

The resulting HOPP(2) discounted payoff $\hat{P}$ is

$$\hat{P}(\theta) = P\big(K_N(\theta)\big)\prod_{i=1}^{N}\frac{\partial U_i\big(K_j^*(\theta), V_{i,1}\big)}{\partial v}, \tag{3.9}$$

where P is the discounted payoff function, depending on the evolution of state-variables up to time T_N, and K_j is the composition of a sequence of mappings, i.e.

$$K_j(\theta) = Q_j \circ Q_{j-1} \circ Q_{j-2} \cdots \circ Q_1(\theta).$$

In this section, we introduce a general framework of constructing $\hat{P}$ under HOPP(2), which is applicable to most cases.

Financial products often have payoffs which are constant or linear away from their discontinuities, we want to make use of this regularity property in our HOPP(2) scheme. We first establish a class of functions which HOPP(2) is applicable to. The functions are defined in terms of strata sets.

Definition 3.4.1. *We shall say that disjoint sets $E_{j,k}$ with $j = 1, 2, \ldots, N$, and $k = 0, 1, \ldots, M_j$, stratify the state-space if $E_{j,k} \subset \mathbb{R}^n$ and*

$$\bigcup_k E_{j,k} = \mathbb{R}^n.$$

Here M_j is the number of discontinuities at step j, and we have $S_j \in E_{j,k}$ for some $k = 0, 1, \ldots, M_j$. Our objective here is to apply HOPP(2) to a function which has regular properties away from the discontinuities, i.e. it is $\hat{C}^2$ if $S_j \in int(E_{j,k})$, and has pathwise discontinuities if $S_j \in \partial(E_{j,k})$. Now, we define the class of discontinuous functions with these strata sets.

Definition 3.4.2. *Let $\mathcal{E}((E_{j,k}))$ denote the class of functions*

$$\sum_a \alpha_a \beta_a$$

with $\beta_a(K_N)$ a $\hat{C}^2$ function and α_a a product of indicator functions of sets $E_{j,k}$.

This class includes a function which is equal to a general twice-differentiable function if a sequence of strata, $E_{1,k_1}, E_{2,k_2}, \ldots, E_{N,k_N}$, are all visited by the state variables, and zero otherwise. We also include all finite linear combinations of such functions. Now, we proceed to define the HOPP(2) evolution across the steps.

Definition 3.4.3. *Let Q_j be the evolution of state variables over the time step $[T_{j-1}, T_j]$, for $j = 1, 2, \ldots, N$. The evolution of each step is decomposed again into two phases such that*

$$Q_j = G_j \circ F_j,$$

with

$$F_j : \mathbb{R}^n_{S_{j-1}} \times U_\theta \times (0,1)^{F-1} \to \mathbb{R}^n_{S_j^*},$$

and

$$G_j : \mathbb{R}^n_{S_j^*} \times U_\theta \times (0,1) \to \mathbb{R}^n_{S_j}.$$

Since the composition of differentiable functions is differentiable, we shall say Q_j is C^2 as a function of S_{j-1} and θ if

- *F_j is C^2 with respect to both S_{j-1} and θ for fixed V_i^{F-1},*
- *G_j is C^2 as a function of (S_j^*, θ) for fixed $v_{i,1}$.*

Here, we assume that the standard uniforms, $v_{j,2}, v_{j,3}, \ldots, v_{j,F}$, in F_j are observed before $v_{j,1}$, so the occurrence of $E_{j,k}$ is then determined by $v_{j,1}$. Therefore, although the problem here is multi-dimensional, the change of variable is still one-dimensional, i.e, we only perform the change of variable on the first standard uniform of each step in G_j.

Following Chan and Joshi (2015), we assume that the evolution is adopted to the strata. That is, there exists a sequence of twice-differentiable critical value functions, $a_{j,i}\big(K_j^*(\theta)\big)$, with $j = 1, 2, \ldots, N$, and $i = 0, 1, \ldots, M_j + 1$ such that

$$a_{j,0} = 0 \text{ and } a_{j,M_j+1} = 1,$$

and

$$a_{j,i-1} < a_{j,i},$$

with the property that $G_j(S_j^*, \theta, v) \in E_{j,k}$ if and only if

$$a_{j,i-1}\Big(K_j^*(\theta)\Big) < v_{j,1} \leq a_{j,i}\Big(K_j^*(\theta)\Big).$$

Unlike equation (3.3) where one parameter of interest needs to be considered, our change of variable in the general framework needs to remove the limit dependence on m parameters of interest up to third order. Here, we introduce a change of variable function, $U_j\big(K_j^*(\theta), v_{j,1}\big)$, to replace $v_{j,1}$ at step j. It depends on the evolution of the state variable up to time T_{j-1} and the evolution F_j, i.e. $K_j^* = F_j \circ K_{j-1}$, with conditions that

- it is twice-differentiable as a function of θ and piecewise-differentiable as a function of $v_{j,1}$;
- it is bijective on $[0, 1]$ for fixed $K_j^*(\theta)$;
- $U_j\big(K_j^*(\theta_0), v_{j,1}\big) = v_{j,1}$;
- $U_j\big(K_j^*(\theta_0), a_{j,i}(K_j^*(\theta_0))\big) = a_{j,i}(K_j^*(\theta_0))$ for i=0, 1, …, $M_j + 1$;
- for it to remove the limit dependence of $\hat{P}$ on all parameters, we require for k=1, 2, …, m,

  $$\frac{\partial}{\partial \theta_k}\Big[U_j\big(K_j^*(\theta), a_{j,i}\big)\Big]_{|\theta=\theta_0} = \frac{\partial}{\partial \theta_k}\Big[a_{j,i}\big(K_j^*(\theta)\big)\Big]_{|\theta=\theta_0},$$

  for all $i = 0, 1, \ldots, M_j + 1$;
- for it to remove the limit dependence of $\hat{P}'$ on all parameters, we require for k=1, 2, …, m and

$s = 1, 2, \ldots, m$,

$$\frac{\partial^2}{\partial\theta_k\partial\theta_s}\Big[U_j\big(K_j^*(\theta), a_{j,i}\big)\Big]_{|\theta=\theta_0} = \frac{\partial^2}{\partial\theta_k\partial\theta_s}\Big[a_{j,i}\big(K_j^*(\theta)\big)\Big]_{|\theta=\theta_0}$$

for all $i = 0, 1, \ldots, M_j + 1$.

There are many functions which satisfy the conditions listed above. One example is the unique continuous function which is linear on the intervals $(a_{j,i}(\theta_0), a_{j,i+1}(\theta_0)]$ with the above conditions satisfied,

$$U_j\big(K_j^*(\theta), v_{j,1}\big) = a_{j,i}(K_j^*(\theta)) + \frac{a_{j,i+1}(K_j^*(\theta)) - a_{j,i}(K_j^*(\theta))}{a_{j,i+1}(K_j^*(\theta_0)) - a_{j,i}(K_j^*(\theta_0))}(v_{j,1} - a_{j,i}(K_j^*(\theta_0))), \tag{3.10}$$

if $a_{j,i}(\theta_0) < v_{j,1} \leq a_{j,i+1}(\theta_0)$. By construction, the first- and second-order partial derivatives of U_j with respect to θ satisfy the last two conditions above. As a result of such changes of variable, the strata do not move up to third order, i.e.

$$a_{j,i-1}\Big(K_j^*(\theta_0)\Big) < v_{j,0} < a_{j,i}\Big(K_j^*(\theta_0)\Big) \Leftrightarrow$$

$$a_{j,i-1}\Big(K_j^*(\theta)\Big) + \mathcal{O}\Big(||\theta-\theta_0||^3\Big) < U_j\big(K_j^*(\theta), v_{j,1}\big) < a_{j,i}\Big(K_j^*(\theta)\Big) + \mathcal{O}\Big(||\theta-\theta_0||^3\Big).$$

Another objective of HOPP(2) is to minimize the variance of the Monte-Carlo weight, i.e. to minimize the expression

$$\int_0^1 \Big(\frac{\partial U_j\big(K_j^*(\theta), v\big)}{\partial v}\Big)^2 dv.$$

The Monte-Carlo weight from the linear change of variable function in equation (3.10) is

$$\frac{a_{j,i+1}(K_j^*(\theta)) - a_{j,i}(K_j^*(\theta))}{a_{j,i+1}(K_j^*(\theta_0)) - a_{j,i}(K_j^*(\theta_0))},$$

if $a_{j,i}(\theta_0) < v_{j,1} \leq a_{j,i+1}(\theta_0)$. Chan and Joshi (2015) showed that, this is optimal in terms of minimizing the variance of the Monte-Carlo weight.

The change of variable using equation (2.10) performed on each step has removed the limit dependence on θ of both $\hat{P}$ and $\hat{P}'$, so the small bump size does not cause the bumped path and unbumped path to finish on different sides of discontinuities. Define $\theta_k(h)$ to be a vector of size m, which is equal to zero except in the kth coordinate, and the kth coordinate of it equals h, for $h > 0$. The finite differencing HOPP(2) Gamma estimator of θ_k for $k = 1, 2, \ldots, m$, is

$$\frac{\hat{P}(\theta + \theta_k(h)) - 2\hat{P}(\theta) + \hat{P}(\theta - \theta_k(h))}{h^2}.$$

The finite differencing HOPP(2) Cross-Gamma estimator of θ_k and θ_s , for k=1, 2, . . . , m, and s=1, 2,

$\dots, k-1, k+1, \dots, m$, is

$$\frac{\hat{P}(\theta + \theta_k(h)) - \hat{P}(\theta - \theta_k(h)) - \hat{P}(\theta - \theta_s(h)) + \hat{P}(\theta - \theta_s(h))}{4h^2}.$$

We are interested in the case as $h \to 0$, i.e. the pathwise estimator of the Hessian. Glasserman (2004) showed that the main obstacle to the applicability of the pathwise method to calculate the Hessian is discontinuities in the first-order derivatives of the discounted payoff function, so we have made the first-order derivatives of the discounted payoff function under HOPP(2) Lipschitz continuous. Our HOPP(2) scheme also satisfies other conditions of the pathwise method.

1. The discounted payoff under HOPP(2) is twice-differentiable almost surely since
   - the Monte-Carlo weight $\prod_{i=1}^{N} \frac{\partial U_i\left(K_j^*(\theta), V_{i,1}\right)}{\partial v}$ and the evolution of state variables $K_N(\theta)$ are twice-differentiable functions of θ provided that $a_{j,i}(\theta)$'s are twice-differentiable functions of θ;
   - the probability of the event that $\hat{P}$ is twice-differentiable as a function of θ at θ_0 is 1.
2. The finite differencing estimators of Gammas and Cross-Gammas are uniformly integrable.

The discounted payoff function $\hat{P}$ is now $\hat{C}^2$, we can apply the pathwise method to $\hat{P}$ to calculate pathwise estimators of the Hessian. The HOPP(2) estimator of Cross-Gamma is given by

$$\mathbf{E}\left[\frac{\partial^2\left(P\big(K_N(\theta)\big) \prod_{i=1}^{N} \frac{\partial U_i\left(K_j(\theta), V_{i,0}\right)}{\partial v}\right)}{\partial \theta_k \partial \theta_s}\right]. \tag{3.11}$$

Here, the change of variable functions are as shown in equation (3.10).

3.5 Application to the LIBOR Market Model

3.5.1 LIBOR market model product specifications

Digital Caplet: A digital caplet pays off 1 at T_1 when the value of the underlying forward rate, f_0, is above the strike at T_0, or zero otherwise. The payoff as observed at T_0 is

$$\frac{I_{f_0(T_0)>K}}{1 + f_0(T_0)\tau_0}.$$

Caplet: A caplet pays off $(f_0(T_0) - K)\tau_0$ at T_1 when the value of the underlying forward rate, f_0, is above the strike at T_0, or zero otherwise. The payoff as observed at T_0 is

$$\frac{(f_0(T_0) - K)_+ \tau_0}{1 + f_0(T_0)\tau_0}.$$

For the Monte-Carlo simulation, we set $f_0 = K = 0.04$, $T_0 = 0.5$ and $T_1 = 1$. The volatility of the forward is 0.2. On a sample of 200,000 paths, the results are summarized in Table 3.4.

Option	Analytical Mean	HOPP(2) Mean	HOPP(2) S.E.	LR S.E	S.E Ratio
Caplet Delta	0.256	0.25589	0.00019	0.00031	1.591
Caplet Gamma	33.892	33.88921	0.022	0.103	4.700
Digital Caplet Delta	68.061	68.08076	0.036	0.050	1.390
Digital Caplet Gamma	-920.293	-919.91974	0.486	14.050	28.92

Table 3.4: Vanilla interest rate product: means and standard errors of the Delta and Gamma estimates by HOPP(2) and the likelihood ratio method on a 200,000 paths sample

Target redemption note

Here, we consider the definition of TARN in Piterbarg(2004), where the note knocks out when the total structured coupon reaches a target level, R. The structured coupon of each time, T_j, is $C_j = (K - 2.0 f_j(T_j))_+$, paid at T_{j+1}. With an initial coupon, C^*, the payoff at T_0 is

$$CF_0 = \frac{\Big(C^* + I_{C_0<R}C_0 + I_{C_0>R}R - f_0(T_0)\Big)\tau_0}{1 + f_0(T_0)\tau_0},$$

the payoff at T_j is

$$CF_j = \frac{\Big((I_{(\sum_{k=0}^{j} C_k<R)}C_j + I_{(C_j>R-\sum_{k=0}^{j-1}C_k)}(R - \sum_{k=0}^{j-1} C_k) - f_j(T_j)I_{(\sum_{k=0}^{j-1}C_k<R)}\Big)\tau_j}{1 + f_j(T_j)\tau_j},$$

the payoff at T_{n-1} is

$$CF_{n-1} = \frac{I_{(\sum_{k=0}^{n-2}C_k<R)}(R - \sum_{k=0}^{n-2} C_k - f_{n-1}(T_{n-1}))\tau_{n-1}}{1 + f_{n-1}(T_{n-1})\tau_{n-1}}.$$

For the Monte-Carlo simulations, we implement a five-rate LIBOR model for the TARN to compare the results from HOPP(2) and the likelihood ratio method. The likelihood ratio method is only applicable if the full-factor model is used, therefore it will be computationally intensive if a large number of forward rates is involved. The forward times are equally-spaced across $[0.5, 3]$. The initial forward rates increase linearly from $f_0 = 2.5\%$ to $f_4 = 4.5\%$. We set $K = 0.12$ and $R = 0.09$. The volatilities are 0.2 and the correlation ρ_{ij} is $0.5 + 0.5\exp(-0.2|T_i - T_j|)$. We compare the results calculated by HOPP(2) on a sample of 200,000 paths to the result calculated by the likelihood ratio method on a sample of 200,000,000 paths, so the standard errors of the first two Gammas and the first Cross-Gamma by the two methods are of similar precision, see Table 3.6 and 3.7 in the chapter appendix.

We also compare the ratio of the sum of standard errors of the Hessian calculated by the likelihood ratio method to the sum of standard errors of the Hessian calculated by HOPP(2) with the same sample size, 200,000 paths, see Table 3.8.

We analyze the performance of HOPP(2) and the likelihood ratio method, with different volatilities and different values of T_0. We plot the sum of the standard error of the Hessian calculated by the two methods on the same graph, the parameters are the same as in the above example, except we vary the volatility from 0.1 to 0.5 in Figure 5 and T_0 from 0.05 to 0.5 in Figure 6. The sample sizes for these two experiments are 500,000 paths. The results are represented in the Figures 3.3 and 3.4.

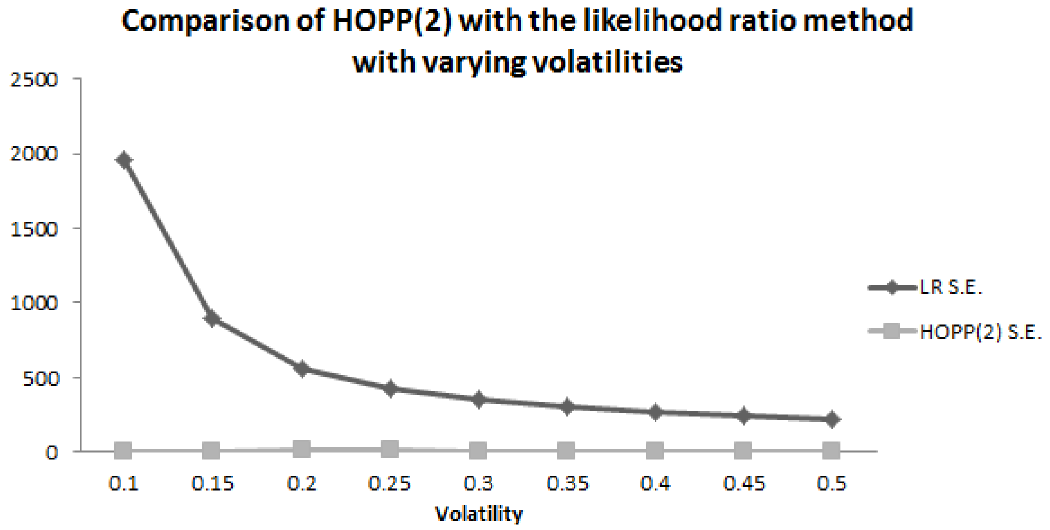

Figure 3.3: Hessian of five-rates TARN with varying volatility: the sum of standard errors of the Hessian calculated by HOPP(2) and the likelihood ratio method with 500,000 paths

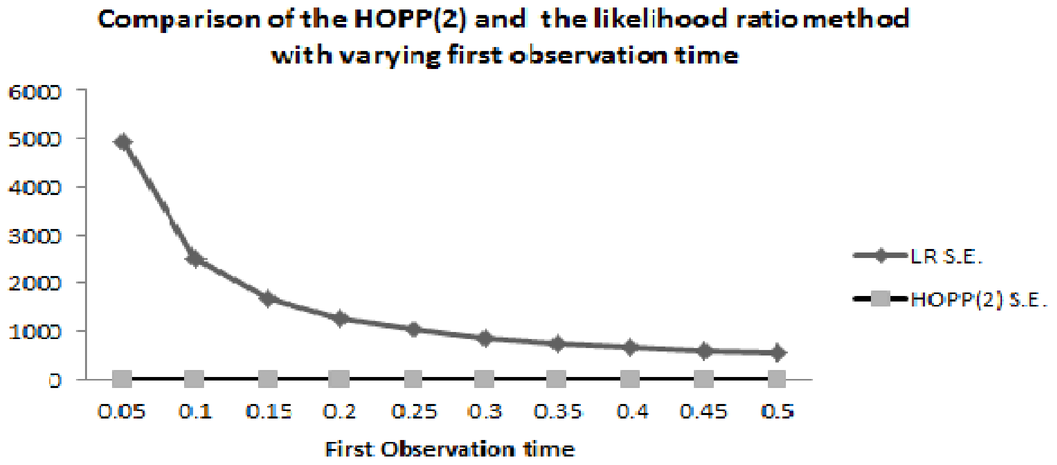

Figure 3.4: Hessian of five-rates TARN with varying first observation date: the sum of standard errors of the Hessian calculated by HOPP(2) and the likelihood ratio method with 500,000 paths

The HOPP(2) method always outperforms the likelihood ratio method with smaller standard errors. The standard errors of the Hessian calculated by HOPP(2) unlike the likelihood ratio method are very stable. The maximum sum of standard errors by HOPP(2) for the set of volatilities is 9.88 when $\sigma = 0.2$, compared to 555.89 by the likelihood ratio method; the minimum is 2.07 when $\sigma = 0.1$, compared to 1962.38 by the likelihood ratio method. The maximum sum of standard errors by HOPP(2) for the set of T_0 is 9.88 when $T_0 = 0.5$, compared to 555.89 by the likelihood ratio method; the minimum is 3.69 when $T_0 = 0.05$, compared to 4929.00 by the likelihood ratio method.

Computational timing for TARN

Finally, we investigate the computational timing of HOPP(2) for calculating Greeks. The algorithmic Hessian method by Joshi and Yang(2011) is able to compute all the Gammas with order $O(nFNM)$ with $M = n + F + 2$ if the order of computation for the price is $O(nFN)$ which is the case for the LMM.

For this experiment, we compute the price and the Greeks (the Hessian and the Deltas are computed simultaneously) with sample size of 50,000 paths. Since we are using n steps with n rates, the time taken for computing price should be with an order of n^2, and the Hessian should be n^3. The results are shown in Table 3.5. We also divide the times by n^2 and n^3 to compare the results.

n	5	10	15	20	25
Price Time	0.218	0.53	0.936	1.451	2.059
Additional HOPP(2) Greek Time	0.749	1.748	2.964	4.508	6.287
Price $\times n^{-}2$	0.00872	0.0053 7	0.00416	0.0036275	0.0032944
Greeks $\times n^{-}3$	0.005992	0.001748	0.000878222	0.0005635	0.000402368

Table 3.5: Time(in seconds) of pricing and calculating the Deltas and the Hessian simultaneously of a TARN

It is interesting to see the ratio of the additional HOPP(2) Greek times divided by the time of calculating the price is not increasing as the number of assets increases. This is possibly because we have used the same target rate for the TARNs with different number of forward rates, so the TARN with 25 rates has a much higher possibility of early redemption than the one with 5 rates. The HOPP(2) method only initializes the Hessian at the redemption date, the computational order is then $O(nFMI)$ for $I \leq N$, where I the number of observation dates up to and including the redemption date. The average value of I is likely to be similar for the 5-rate TARN and the 25-rate TARN, since the same target rate is used.

3.6 Conclusion

We have presented a new method for computing second-order Greeks of financial products. The idea of our method is originated from the OPP method (Chan and Joshi, 2015) for computing first-order derivatives. A sequence of measure changes are introduced, which removes the pathwise discontinuities of both the discounted payoff function and its first-order derivatives. The measure change is selected so that the variance of the likelihood ratio part is minimized. The simulated discounted payoff function under the new scheme is $\hat{C}^2$, and the algorithmic Hessian method is used to calculate the pathwise Hessian of the discounted payoff under the new scheme. This method allows multiple discontinuities per step, which is applicable for a wide range of products with discontinuous or angular payoffs, where the pure pathwise method is not applicable. Our numerical results suggest that the new method is significantly better than the pathwise-likelihood ratio and the pure likelihood ratio method in terms of reducing the standard error of the Hessian estimators, as well as being more widely applicable.

Appendix One: Tables of Numerical Results

	f_1	f_2	f_3	f_4	f_5
f_1	-29.46 vs -29.36	-36.17 vs -34.89	-13.82 vs -16.48	-1.88 vs -2.5	-0.82 vs -0.48
f_2	-36.17 vs -34.89	-43 vs -41.71	-16.81 vs -16.58	-2.13 vs -1.27	-1.02 vs -2.23
f_3	-13.82 vs -16.48	-16.81 vs -16.58	-0.87 vs 1.16	-2.12 vs -2.29	-0.59 vs -0.19
f_4	-1.88 vs -2.5	-2.13 vs -1.27	-2.12 vs -2.29	1.22 vs -1.14	-0.12 vs 0.77
f_5	-0.82 vs -0.48	-1.02 vs -2.23	-0.59 vs -0.19	-0.12 vs0.77	0.01 vs -0.46

Table 3.6: TARN: Mean of the Hessian computed by HOPP(2) on a 200,000 paths sample and by the likelihood ratio method on a 200,000,000 paths sample

	f_1	f_2	f_3	f_4	f_5
f_1	1.88 vs 1.7	1.72 vs 1.64	1.01 vs 1.17	0.26 vs 1.02	0.03 vs 0.67
f_2	1.72 vs 1.64	2.84 vs 2.2	1.29 vs 1.48	0.34 vs1.16	0.04 vs 0.76
f_3	1.01 vs 1.17	1.29 vs 1.48	1.13 vs 1.61	0.27 vs1.11	0.03 vs 0.65
f_4	0.26 vs 1.02	0.34 vs 1.16	0.27 vs 1.11	0.16 vs1.23	0.01 vs 0.69
f_5	0.03 vs 0.67	0.04 vs 0.76	0.03 vs 0.65	0.01 vs 0.69	0.00 vs 0.52

Table 3.7: TARN: standard errors of the Hessian computed by HOPP(2) on a 200,000 paths sample and by the likelihood ratio method on a 200,000,000 paths sample

	f_1	f_2	f_3	f_4	f_5
f_1	28.31	30.06	35.95	121.57	690.56
f_2	30.06	24.53	36.36	106.65	585.63
f_3	35.95	36.36	44	124.86	703.82
f_4	121.57	106.65	124.86	232.28	1857.18
f_5	690.56	585.63	703.82	1857.18	66540.6

Table 3.8: TARN: standard errors of the Hessian computed by the likelihood ratio method divided by standard errors of the Hessian computed by HOPP(2) on 200,000 paths

Chapter 4

Computing Cross-Gammas of Bermudan Swaptions and Cancellable Swaps

4.1 Introduction

Bermudan swaptions and cancellable swaps are among the most liquidly traded exotic interest rate derivative contracts. Consequently, their pricing and risk management are of high practical importance. Such derivatives are typically delta-hedged, the Hessian (or gamma-matrix) of the price measures the rates of change in the deltas with respect to changes in the underlying forward rates, and thus provides an explanation for the profit-and-loss in a delta-hedging strategy. The objective of this chapter is to develop an efficient and accurate algorithm to computing the Hessians of Bermudan swaptions and cancellable swaps which can be used in practice for hedging purposes.

The pricing of Bermudan swaptions and cancellable swaps is often implemented under the LMM (Brace et al, 1997) framework. Monte Carlo simulation is the most common method of its implementation in practice. Until recently, the problem of pricing early-exercisable interest-rate derivatives by Monte Carlo simulation was regarded as very hard. The difficulties arise because the exercise decisions embedded in Bermudan-type products require comparisons between the value received upon exercise and the continuation value. The continuation value is the expected value of the unexercised product, which is not easily obtainable in a simulation.

A number of break-throughs have been made to address this problem. Most of these methods focus on producing a lower bound algorithm, which produce a lower-biased estimate of the price. We shall focus on lower bounds in this paper. The regression approach (typically least-squares) first introduced by Carriere (1996) and then Longstaff and Schwartz (2001), is widely used to approximate the continuation values in calculating the lower bound. The approach is usually adequate to make the correct exercise decisions when options are deeply in or out of the money, but is less effective when the decision is not so clear. Based on this observation, Beveridge et al (2013) used two regressions at each exercise time. Joshi (2014) recently adopted this idea, but took it further. The author suggested using multiple regressions, typically about 5, discard a fixed fraction of paths farthest from the boundary each time and regress the remaining ones. The major advantage of the multiple regression over the double regression is that by only discarding a small fraction of paths, we are unlikely to be affected by substantial mis-estimation of the continuation value close to the boundary. In these regression-based techniques, a second pass is typically used to compute a lower-biased estimate of the price using the exercise strategy developed in this first pass; a better exercise strategy via the multiple regression method provides a tighter lower bound of the price, i.e. the downward bias is smaller in the second pass. We shall use the multiple regression approach in this paper to estimate the price.

Along with pricing, another important task in practice for quants is to compute accurate and fast greeks. Piterbarg (2004) showed that given the optimal exercise strategy which maximizes the value of the option over all exercise times, one can apply the pathwise method to produce unbiased estimates of deltas whilst fixing the optimal exercise time. However, Korn and Liang (2015) emphasized that due to the lack of Lipschitz continuous first-order derivatives, the basis for applying the pathwise method to calculating gammas of Bermudan-type product is not justified. Previously, they were only estimated with some smoothing techniques; for example, Joshi and Yang (2011) used a localized smoothing of the Monte-Carlo algorithm at each exercise time and the estimated gammas for cancellable swaps were consequently biased.

The computation of greeks of financial products with discontinuous and angular payoffs has been studied extensively. The PWLR method computes the Hessian of products with angular payoffs, but suffers the problem of producing estimates with large standard errors, and it is not applicable if the reduced-factor LMM is used. In the previous chapter, we used a twice-differentiable change of measure function to produce a $\hat{C}^2$ algorithm for computing the price.For products with angular payoffs, a differentiable change of measure function is sufficient to remove the pathwise discontinuities of the first-order derivatives. In this chapter, we modify the HOPP(1) method to produce an unbiased estimate of the Hessian for Bermudan-type products. The difficulties of applying the HOPP(1) method here is discussed in Section 4.3. To address the problems, at each exercise time, we replace one standard uniform for simulating the state variables with a new change of measure function, which is a weighted average of the original standard uniform and the HOPP(1) change of measure function. The weight depends on the location of the unbumped path. In particular, the weight of the HOPP(1) change of measure is zero if the unbumped path is far from the boundary. The pathwise method is then applicable to the new $\hat{C}^2$ pricing algorithm to produce unbiased estimates of the Hessian, we shall call it the Hessian by Optimal Measure Changes(HOMC). The virtue of our method for computing sensitivities is its wide applicability, since it only requires the smoothness of cash flow functions and monotonic approximated continuation values.

The paper is organized as follows. In Section 4.2, we first give product descriptions of some LIBOR exotics, as well as a brief discussion of the multiple regression method. We then in Section 4.3 present the basic idea of the HOMC algorithm. Numerical results on cancellable Swaps are presented in Section 4.4.

4.2 The LIBOR market model and the multiple regression algorithm

4.2.1 LIBOR product descriptions

A payer swap is a swap in which the holder pays the fixed rates, K, and receives the floating rates. The payoff of a payer swap is

$$\sum_{i=0}^{n-1} \frac{(f_i(T_i) - K)\tau_i}{1 + \tau_i f_i(T_i)} P_i(T_i).$$

One can compute the price as well as the Greeks of a swap analytically. Payer swaps increase in value as forward rates rise, and vice versa. The gammas of a payer swap are negative due to the interaction of the swap rate and the discounting bonds.

A receiver Bermudan swaption is where the holder of the option has the right but not the obligation to enter into a receiver swap at a set of pre-determined reset times. The swap always terminates at time T_n, so the length of the swap decreases with time. Thus we have the payoff of a receiver Bermudan swaption,

$$(K - \mathrm{SR}_j)_+ \sum_{i=j}^{n-1} \tau_i P_{i+1},$$

where SR_j denotes the swap rate starting at T_j and ending at T_n.

A cancellable swap is where one of the counter parties has the right but not the obligation to terminate the swap on one or more pre-determined dates during the life of the swap. A payer cancellable swap can be viewed as a combination of a vanilla swap and a receiver Bermudan swaption. We concentrate on cancellable swaps throughout this paper. The exercise value of a cancellable swap at exercise time, T_j for $j = 0, 1, ...n - 1$, is zero, and we have the accumulated cash flows from T_0 to T_j. Once we compute the Hessian of the price for a cancellable swap, it is trivial to obtain the Hessian for its Bermudan swaption counterpart. This is because we can compute the Hessian of the price of vanilla swaps analytically.

4.2.2 The multiple regression algorithm

Pricing products with early-exercisable features via Monte-Carlo simulation is one of the hard problems in financial engineering. This is because exercise decisions require knowledge of continuation values, which are not easily obtainable in simulations. Regression-based techniques are popular methods to obtain approximations of the continuation values from simulated paths, we refer the reader to Joshi (2011) for detailed explanations of these methods. Here, we present a brief sketch of the least-squares algorithm. Before we start our description, consider the following notation. For $i = T_0, T_1, T_2, ..., T_{n-1}$, let

- F_i denote the vector of the current basis functions and current state variables,
- PC_i denote the pathwise discounted continuation value,
- $C_i(F_i)$ denote the approximated continuation value as a function of the current basis,
- $E_i(F_i)$ denote the current exercise value, which depends on the current state variables.

The first pass of the algorithm is based on a backwards induction procedure starting at T_{n-1}, and repeat and move backwards until T_0. At each exercise time, T_i, the continuation value, C_i, is estimated by a regression of PC_i on F_i. Typically, it is approximated by a linear combination of current basis functions. Least-squares regression is then used to estimate the coefficients. The approximated continuation value from the least-squares regression is compared to E_i; the value of the product at each exercise time is set to be E_i if it is greater, and to the pathwise discounted continuation value, PC_i, otherwise. We then move backwards to step T_{i-1}. We discount the value obtained from T_i and add any additional cash flows between the steps to obtain PC_{i-1}. Then, we regress again and repeat the same procedure until the first step.

A second independent pass is then generated. At each exercise time, T_i, we compare the current exercise value and the estimated continuation value with the coefficients determined in the first pass. If $E_i > C_i$, we exercise the product and compute the corresponding discounted cash flows, otherwise, we move forward to the next exercise time. The result gives an unbiased estimate of the lower bound given the exercise strategy determined in the first pass.

The least-squares regression method is widely used for calculating lower bounds. However, the accuracy of the output is heavily dependent on the choice of basis in the regression, see Beveridge et al (2013). What to use for basis functions is generally not obvious, and a significant amount of time is required to investigate the appropriate set of basis functions for a particular product. For Bermudan swaptions and cancellable swaps, second-order polynomials in forward rates and swap-rates are popular choices following Piterbarg (2004).

One enhancement of the least-squares method is the multiple regression method introduced by Joshi (2014), which is an efficient algorithm with less dependence on the choices of basis functions. It is based on the observation that the least-squares method is usually adequate to make the correct exercise decisions when options are deeply in or out of money, i.e. when $|C_i - E_i|$ is large. He used multiple regressions to obtain a better fit to continuation values when the option is close to the boundary. The approximated continuation value is dependent on the distance of the path from the boundary. Let $C_{i,j}(F_i)$ denote the approximated continuation value from the jth regression at T_i as a function of the current basis. For a fixed fraction $\alpha \in (0, 1)$ and regression depth d, the regression step in the first pass at each time T_i in the least-squares algorithm is instead performed as follow:

- Calculate the pathwise discounted continuation value PC_i.

- Perform an initial regression using the least-squares method with these PC_i's, and produce the first estimate of the expected discounted continuation value, $C_{i,1}(F_i)$.

- Use these estimates, $C_{i,1}(F_i)$, to determine the critical value, $L_{i,1}$, such that the fraction $1-\alpha$ of the paths have $|C_{i,1}(F_i)-E_i(F_i)|$ greater than it.

- Discard the paths which have $|C_{i,1}(F_i)-E_i(F_i)|$ greater than $L_{i,1}$, and perform a second regression with the remaining PC_i's to obtain another set of coefficients and a critical value $L_{i,2}$.

- Perform similar exercises for other regressions until the dth regression.

The approximated continuation value from the sequence of regressions at each time T_i is

$$C_i(F_i)=\sum_{j=1}^{d-1}\Big(C_{i,j}(F_i)\mathbb{I}_{|C_{i,j}(F_i)-E_i(F_i)|>L_{i,j}}\prod_{k=1}^{j-1}\mathbb{I}_{|C_{i,k}(F_i)-E_i(F_i)|<L_{i,k}}\Big)+C_{i,d}(F_i)\prod_{k=1}^{d-1}\mathbb{I}_{|C_{i,k}(F_i)-E_i(F_i)|<L_{i,k}}. \tag{4.1}$$

Given the parameter input $\theta\in\mathbb{R}^m$ and a sequence of standard uniform random variates $V\in\mathbb{R}^{n\times F}$, the pathwise estimate of the price for cancellable products can now be expressed as a function $P:\mathbb{R}^m\times\mathbb{R}^{n\times F}\to\mathbb{R}$ such that

$$P(\theta,V)=N(0)\sum_{i=0}^{n-1}\frac{CF\big(T_i,F_i(\theta,V)\big)}{N\big(T_i,F_i(\theta,V)\big)}\prod_{j=0}^{i}\mathbb{I}_{C_i\big(F_i(\theta,V)\big)>E_i\big(F_i(\theta,V)\big)}, \tag{4.2}$$

where N is the numeraire and CF's are the cash flows of the product. The exercise value E_i of a cancellable swap is zero.

4.3 The HOMC algorithm

In this section, we present the HOMC algorithm. The first pass of the algorithm is exactly the same as explained in the previous section, which determines the exercise strategy. In the second pass, we perform measure changes to ensure the pathwise estimate of the price has Lipschitz continuous first-order derivatives. We only need to ensure that small bumps in the parameters of interest do not lead to the bumped path finishing on a different side of the angularity, since the pathwise estimate of the price is smooth away from the pathwise angularities.

The HOPP(1) algorithm has been made generic, which is applicable to a wide range of practical situations. The only two requirements for it to apply are

- the underlying pathwise estimate of the price, $P(\theta,V)$, is smooth away from the angularities,

- at T_i given the state variables evolved up to T_i except the first standard uniform, F_i^*, with standard uniforms, $v_{i,j}$ for $j>1$, observed, we can either compute or approximate(via numerical techniques such as the Newton-Raphson method) the differentiable critical value function of the first standard uniform, $a_i(F_i^*)$, for the function $P(\theta, V)$ to cross the angularity.

The pathwise price function in equation (2.6) satisfies the above requirements, since the cash flow functions of Bermudan swaptions and cancellable swaps are smooth and the continuation value of these products are monotonic as a function of the first standard uniform at each step given all other standard uniforms. Thus, we obtain the HOPP(1) change of measure

$$U_i^{HOPP(1)}(\theta, v_{i,1}) = \begin{cases} \frac{1-a_i\left(F_i^*(\theta)\right)}{1-a\left(F_i^*(\theta_0)\right)}\Big(v_{i,1} - a_i\big(F_i^*(\theta_0)\big)\Big) + a_i\big(F_i^*(\theta)\big), & \text{if } v_{i,1} > a_i\big(F_i^*(\theta_0)\big) \\ \frac{a_i\left(F_i^*(\theta)\right)}{a_i\left(F_i^*(\theta_0)\right)} v_i, & \text{if otherwise.} \end{cases}$$

Here, we modify the HOPP(1) method: measure changes are only performed if the unbumped path is within the inner-most region and a smooth function is introduced to provide the transition of the change of measure function between the inner-most and the outer regions. Firstly, if the unbumped path is not within the inner-most region, it is impossible for the bumped path with a small bump size to cross the angularity. Thus, we shall only use measure changes if the approximated continuation value is determined by the inner-most regression, i.e. $C_i(F_i) = C_{i,d}(F_i)$. Secondly, if the unbumped path is within the inner-most region, the possibility of crossing the angularity reduces as the unbumped path moves away from the boundary. Thus, we shall only use the full HOPP(1) algorithm close to the boundary.

The following change of measure function is used,

$$U_i(\theta, v_{i,1}) = g(\theta_0, v_{i,1})U_i^{HOPP(1)}(\theta, v_{i,1}) + \big(1 - g(\theta_0, v_{i,1})\big)v_{i,1}, \tag{4.3}$$

to replace the first generated standard uniform $v_{i,1}$ at step i. This function is a weighted average of the original standard uniform $v_{i,1}$ and the HOPP(1) change of measure function $U_i^{HOPP(1)}$. The weight g is a function depending on the location of the unbumped path. In particular, we want to use more of the HOPP(1) change of measure near the boundary, since the possibility of the bumped path with a small bump size crossing the angularity is significant, and vice versa. We require the function g to satisfy the following conditions,

- $g(\theta_0, v_{i,1}) = 0$ if $|C_{i,d-1} - E_i| > L_{i,d-1}$,
- $g(\theta_0, v_{i,1}) = 1$ if $C_{i,d} = E_i$,
- $\frac{\partial g}{\partial C_{i,d}} > 0$ if $C_{i,d} < E_i$ and $\frac{\partial g}{\partial C_{i,d}} < 0$ if $C_{i,d} > E_i$,
- continuously differentiable as a function of $v_{i,1}$.

Thus, we need to choose a function, g, which depends on both $C_{i,d-1}$ and $C_{i,d}$.

There are many functions which satisfy the above conditions; for computing the Hessians of Bermudan swaptions and cancellable swaps, we employ the following

$$g(\theta_0, v_{i,1}) = \begin{cases} 0 & \text{if } |C_{i,d-1} - E_i| > L_{i,d-1} \\ \exp\left(- \frac{\left(C_{i,d}\left(F_i(\theta_0, v_{i,1})\right) - E_i\left(F_i(\theta_0, v_{i,1})\right)\right)^2}{L_{i,d-1}^2 - \left(C_{i,d-1}(F_i\left(\theta_0, v_{i,1})\right) - E_i\left(F_i(\theta_0, v_{i,1})\right)\right)^2} \right) & \text{if otherwise} \end{cases} . \tag{4.4}$$

It is easy to verify that the above equation satisfies the conditions on the function g for it to be used in equation (4.3). In particular, we have constructed the function such that it is smooth in $v_{i,1}$.

Since, we are in the inner-most region, the function $U_i^{HOPP(1)}$ is determined by the inner-most approximated continuation value, i.e. a_i is the critical value of $v_{i,1}$ such that $C_{i,d} = E_i$. This change of measure function is continuously differentiable in the parameter of interest θ, since $U_i^{HOPP(1)}$ is a differentiable function, and it is optimal in terms of minimizing the variance of the likelihood ratio terms.

The resulting likelihood ratio weight is

$$W_i(\theta, v_{i,1}) = g(\theta_0, v_{i,1}) \frac{\partial U^{HOPP(1)}(\theta, v_{i,1})}{\partial v_{i,1}} + \left(1 - g(\theta_0, v_{i,1})\right) + \left(U^{HOPP(1)}(\theta, v_{i,1}) - v_{i,1}\right) \frac{\partial g(\theta_0, v_{i,1})}{\partial v_{i,1}}. \tag{4.5}$$

The HOMC pathwise estimate of the price is

$$\hat{P}(\theta) = P(\theta, U) \prod_{i=0}^{n-1} W_i(\theta, v_{i,1}), \tag{4.6}$$

where U is the set of modified standard uniforms, i.e. the first standard uniform at each step, $v_{i,1}$, is replaced by equation (
HOMC change). The new algorithm satisfies the following conditions

- it is an unbiased estimate of the price, i.e. $\mathbb{E}[P(\theta, V)] = \mathbb{E}[\hat{P}(\theta)]$,

- it is $\hat{C}^2$, since the sequence of measure changes has removed the pathwise discontinuities of the first-order derivatives,

- it is twice-differentiable almost surely, since the composite of differentiable functions is differentiable,

- the finite-differencing estimate (Glasserman, 2004) of the Hessian is uniformly integrable.

Thus, it satisfies the conditions for applying the pathwise method to compute second-order derivatives.

4.4 Application to cancellable swaps and numerical results

In this section, we shall perform numerical experiments to demonstrate the efficacy and speed of the HOMC algorithm. Since a cancellable swap is the sum of a vanilla swap and a Bermudan swaption, we shall only present the algorithm for a cancellable swap.

Our numerical examples are presented for cancellable swaps, however, we believe that there are no particular barriers to implementing them for more complicated Bermudan-type products. Here, we provide a brief discussion of the application of HOMC for snowballs. The exercise value of a callable snowball is not analytically computable. Piterbarg (2004) suggested a regression approach to estimate the exercise value. Beveridge and Joshi (2008) showed numerous simplifications if one instead works with the cancellable product,. See also Joshi (2011) and Amin(2003). In particular, one only needs to use regressions to approximate the continuation values of cancellable snowballs, which is the same for our cancellable swap example. One subtlety associated with snowballs is however their payoff functions are also angular. To apply HOMC, we need to modify the change of measure function, U_i, at T_i, i.e. incorporate an additional critical point, to ensure that the bumped path is on the same side as the unbumped paths for both the point where the exercise value equals to the approximated continuation value and the point where the payoff at T_i hits to zero.

4.4.1 Product description and LMM set-up

We consider an n-period cancellable swap, which can be cancelled at each reset date. The estimated pathwise price of the product is as shown in equation (4.2), with the cash flow function

$$CF(T_i, f_i) = \frac{\tau_i(f_i(T_i) - K)}{1 + \tau_i f_i(T_i)},$$

where K is the strike rate.

We consider the 6-months LIBORs, i.e. $T_i - T_{i-1} = 0.5$. A flat volatility structure with volatility $\sigma_i = 0.1$ is used. The instantaneous correlations are given by

$$\rho_{i,j} = \exp(-0.1|T_i - T_j|).$$

The rates are driven by a five-dimensional Brownian motion. Korn and Liang (2015) studied similar examples with a one-dimensional Brownian motion set-up.

For the multiple regression method in the first pass, we use three basis variables, $\log(f_i)$, $\log(f_{i+1})$ and $\sum_{j=i+2}^{n-1} \log(f_j)$ at each T_i, and the estimated $C_{i,j}$'s are affine functions of these basis variables. Quadratic functions are typical examples used in the literature for the basis. However, the multiple regression algorithm makes the choice of basis functions for the least-squares regression much less important. Linear basis functions of the $\log$ forward rates are also linear in the first standard normal random variable at each step. Thus we can easily compute the critical value function, a_i, of the first standard uniform in equation (4.3). This property helps to reduce the computational cost of implementing the HOMC algorithm.

4.4.2 Comparison with the PWLR method

We first consider an at-the-money cancellable swap, with the first reset date $T_0 = 0.5$ and $K = 0.05$. The first pass is conducted using 50,000 paths sample and a regression depth of $d = 5$ for the multiple regression algorithm to fix the exercise strategy. The regression is conducted so that the inner-most regression has 10,000 paths. Since we are working with multiple regressions, a significant number of paths is required for the inner-most region to produce an accurate approximation of the continuation value close to the boundary. This approach was developed in GPU programs, where a large number of paths can be rapidly computed; Joshi (2014) used 327,680 paths for the first pass to develop the exercise strategy. In the second pass, we use 5,000 paths for the HOMC algorithm. We shall use the PWLR method to benchmark our results. The PWLR method is not applicable for the reduced-factor LMM. Therefore we only benchmark our method against the PWLR method for a five-rate cancellable, due to the computational cost of PWLR. In order to achieve a similar level of precision to HOMC, we need to run 500,000 paths in the second pass for PWLR. The results demonstrate that the HOMC method produces unbiased estimates of the Hessian, and it is more efficient than the PWLR method, see Table 4.1 and 4.2.

HOMC vs PWLR	f_0	f_1	f_2	f_3	f_4
f_0	19.79 vs 19.22	8.37 vs 8.44	6.51 vs 6.39	5.08 vs 5.21	3.17 vs 3.25
f_1	8.37 vs 8.44	15.61 vs 16.22	8.52 vs 8.62	6.65 vs 6.50	3.69 vs 3.58
f_2	6.51 vs 6.39	8.52 vs 8.62	13.61 vs 13.66	7.65 vs 7.52	4.3 vs 4.97
f_3	5.08 vs 5.21	6.65 vs 6.50	7.65 vs 7.52	12.31 vs 11.91	4.42 vs 4.3
f_4	3.17 vs 3.25	3.69 vs 3.58	4.3 vs 4.97	4.42 vs 4.3	14.55 vs 14.06

Table 4.1: The mean of the Hessian computed by the HOMC algorithm using 5,000 paths in the second pass and the PWLR method using 500,000 paths in the second pass for an at-the-money five-rate cancellable swap when $T_0 = 0.5$. 50,000 paths are used in the first pass for both methods

HOMC vs PWLR	f_0	f_1	f_2	f_3	f_4
f_0	1.34 vs 0.77	1.02 vs 0.72	0.97 vs 0.7	1.06 vs 0.84	0.95 vs 1.1
f_1	1.02 vs 0.72	1.19 vs 0.99	1.02 vs 0.95	1.05 vs 1.15	0.96 vs 1.51
f_2	0.97 vs 0.7	1.02 vs 0.95	1.24 vs 0.95	1.29 vs 1.15	1.08 vs 1.54
f_3	1.06 vs 0.84	1.05 vs 1.15	1.29 vs 1.15	1.73 vs 1.15	1.43 vs 1.53
f_4	0.95 vs 1.1	0.96 vs 1.51	1.08 vs 1.54	1.43 vs 1.53	3.82 vs 1.12

Table 4.2: The standard error of the Hessian computed by the HOMC algorithm using 5,000 paths in the second pass and the PWLR method using 500,000 paths in the second pass for an at-the-money five-rate cancellable swap when $T_0 = 0.5$. 50,000 paths are used in the first pass for both methods

As pointed out in the previous chapter, the PWLR approach can produce very large standard errors when the initial reset time is small. In our next numerical experiment, using the same set-up as the first experiment except setting $T_0 = 0.1$, we need to use 2,000,000 paths in the second pass for the PWLR method to produce estimates with a similar level of precision to our HOMC with 5,000 paths. The results are summarized in Table 4.3 and 4.4.

HOMC vs PWLR	f_0	f_1	f_2	f_3	f_4
f_0	10.62 vs 11.11	4.26 vs 4.07	2.98 vs 2.27	1.39 vs 1.43	1.32 vs 1.96
f_1	4.26 vs 4.07	21.89 vs 23.73	9.85 vs 10.54	5.88 vs 5.97	5.38 vs 5.36
f_2	2.98 vs 2.27	9.85 vs 10.54	21.43 vs 20.76	8.15 vs 8.93	5.37 vs 5.21
f_3	1.39 vs 1.43	5.88 vs 5.97	8.15 vs 8.93	17.42 vs 17.41	5.23 vs 5.85
f_4	1.32 vs 1.96	5.38 vs 5.36	5.37 vs 5.21	5.23 vs 5.85	11.68 vs 11.77

Table 4.3: The mean of the Hessian computed by the HOMC algorithm using 5,000 paths in the second pass and the PWLR method using 2,000,000 paths in the second pass for an at-the-money five-rate cancellable swap when $T_0 = 0.1$. 50,000 paths are used in the first pass for both methods

HOMC vs PWLR	f_0	f_1	f_2	f_3	f_4
f_0	1.65 vs 0.75	1.36 vs 0.81	1.16 vs 0.75	1.09 vs 0.83	0.92 vs 1.08
f_1	1.36 vs 0.81	1.36 vs 1.12	1.1 vs 1.02	1.08 vs 1.14	0.97 vs 1.49
f_2	1.16 vs 0.75	1.1 vs 1.02	1.12 vs 1.02	1.14 vs 1.14	1.03 vs 1.49
f_3	1.09 vs 0.83	1.08 vs 1.14	1.14 vs 1.14	1.59 vs 1.14	1.28 vs 1.48
f_4	0.92 vs 1.08	0.97 vs 1.49	1.03 vs 1.49	1.28 vs 1.48	1.72 vs 1.07

Table 4.4: The standard errors of the Hessian computed by the HOMC algorithm using 5,000 paths in the second pass and the PWLR method using 2,000,000 paths in the second pass for an at-the-money five-rate cancellable swap when $T_0 = 0.1$. 50,000 paths are used in the first pass for both methods

We plot the sum of standard errors produced by the two methods using the same set-up and 50,000 paths for both passes, but varying the first reset date from 0.05 to 0.5, see Figure 4.1.

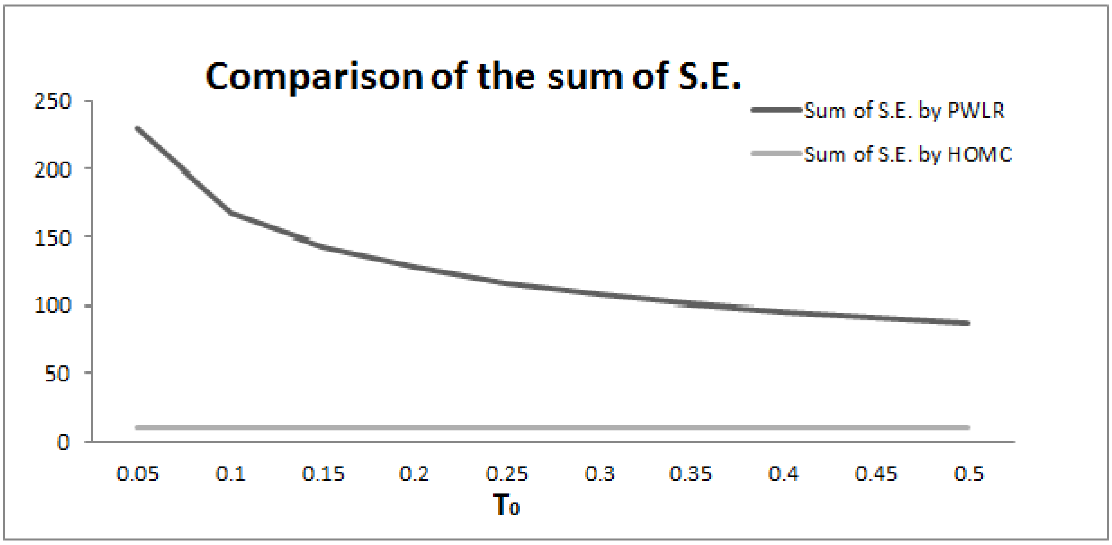

Figure 4.1: Comparison of the sum of standard error with varying T_0 computed by the HOMC algorithm and the PWLR method using 50,000 paths for both passes.

The maximum sum of standard errors computed by the PWLR method is 230.0428 when $T_0 = 0.05$, and the corresponding sum of standard errors computed by HOMC is 9.982. The minimum sum of standard errors computed by the PWLR method is 87.3943 when $T_0 = 0.5$, and the corresponding sum of standard errors computed by HOMC is 10.137. Overall the PWLR method produces large standard errors as the initial reset time approaches zero, and our HOMC produces estimates of the Hessian with stable standard errors.

We further plot the sum of standard errors produced by the two methods using the same set-up and 50,000 paths for both passes, but varying the flat volatility, σ_i, from 0.02 to 0.2, see Figure 4.2.

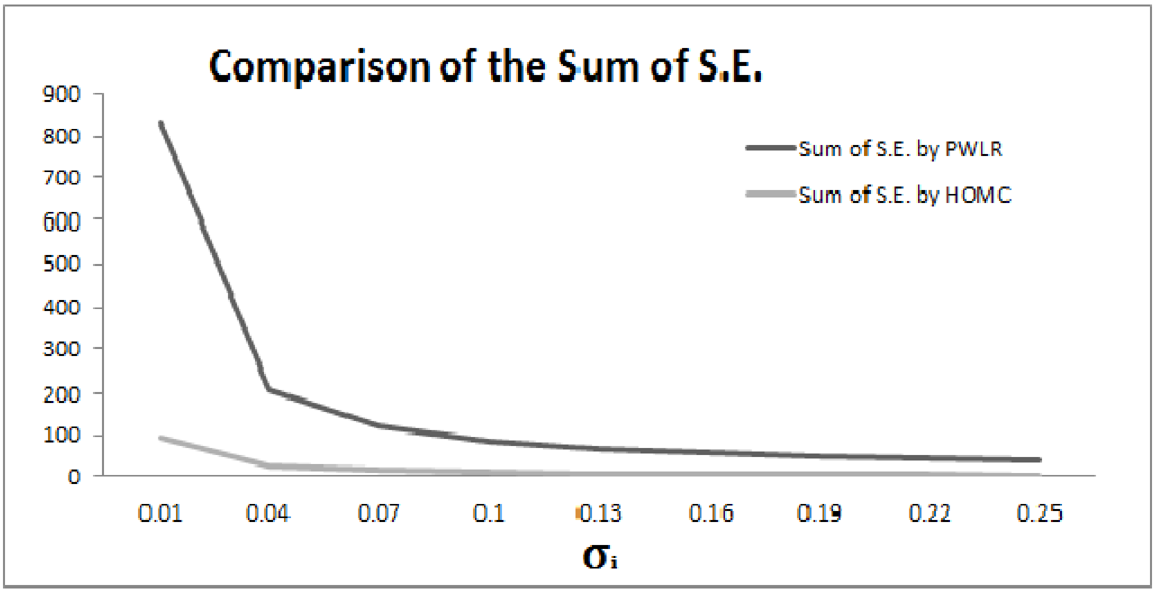

Figure 4.2: Comparison of the sum of standard error with varying σ_i computed by the HOMC algorithm and the PWLR method using 50,000 paths for both passes.

The maximum sum of standard errors computed by the PWLR method is 417.573 when $\sigma_i = 0.02$, and the corresponding sum of standard errors computed by HOMC is 47.473. The minimum sum of standard errors computed by the PWLR method is 45.605 when $\sigma_i = 0.2$, and the corresponding sum of standard errors computed by HOMC is 5.071. Overall both PWLR method and the HOMC method produce large standard errors as the flat volatility, σ_i, approaches zero, but our HOMC method always outperforms the PWLR method significantly.

In the next experiment, we consider a ten-rate at-the-money cancellable swap. Beveridge and Joshi (2014) computed the deltas and gammas of this product with the same set-up as our numerical experiment. They used a smoothing technique to deal with the angularity. Thus the estimated gammas are biased. For this particular product, the price we computed is 105.03 basis points(105 basis points in Joshi and Yang (2011)) with 7.428E-05 standard error using 50,000 paths for both the passes. We compute the Hessian and the deltas of this product, and the results are summarized in Table 6 and 7 in the Appendix. The results show that the delta and Hessian computed by HOMC is within standard error to the results by Joshi and Yang (2011).

4.4.3 The shape of the gamma and the price

Gamma is an estimate of how much the delta of an option changes when the underlying parameter shifts. As a tool, gamma describes how stable the current delta is. A big gamma means that the delta can change dramatically for even a small move in the underlying forward rate. For a delta-hedged position, constantly monitoring gammas is especially important to ensure the adequacy of the current delta position. We shall examine the second-order derivative of the price with respect to the first forward rate, i.e. the first gamma, by the HOMC algorithm, as the first forward rate varies.

In Figures 4.3, 4.4, and 4.5, we plot the price and the corresponding first gammas for a ten-rate cancellable swap with the same set-up as before with varying f_0. In order to produce a stable plot, we use 65,535 paths for the first pass and 50,000 paths for the second pass for these experiments. Since the PWLR method also produces unbiased estimates of the Hessian, one can use it to produce these plots. However, due to the significant standard errors of the PWLR estimates(shown in the previous numerical examples), it is computationally expensive using PWLR for producing plots with the same shape as our results. We consider the different first reset dates with different ranges for the first forwards, and present the plots for $T_0 = 0.01$ in Figure 3, $T_0 = 0.05$ in Figure 4, and $T_0 = 0.5$ in Figure 4.5.

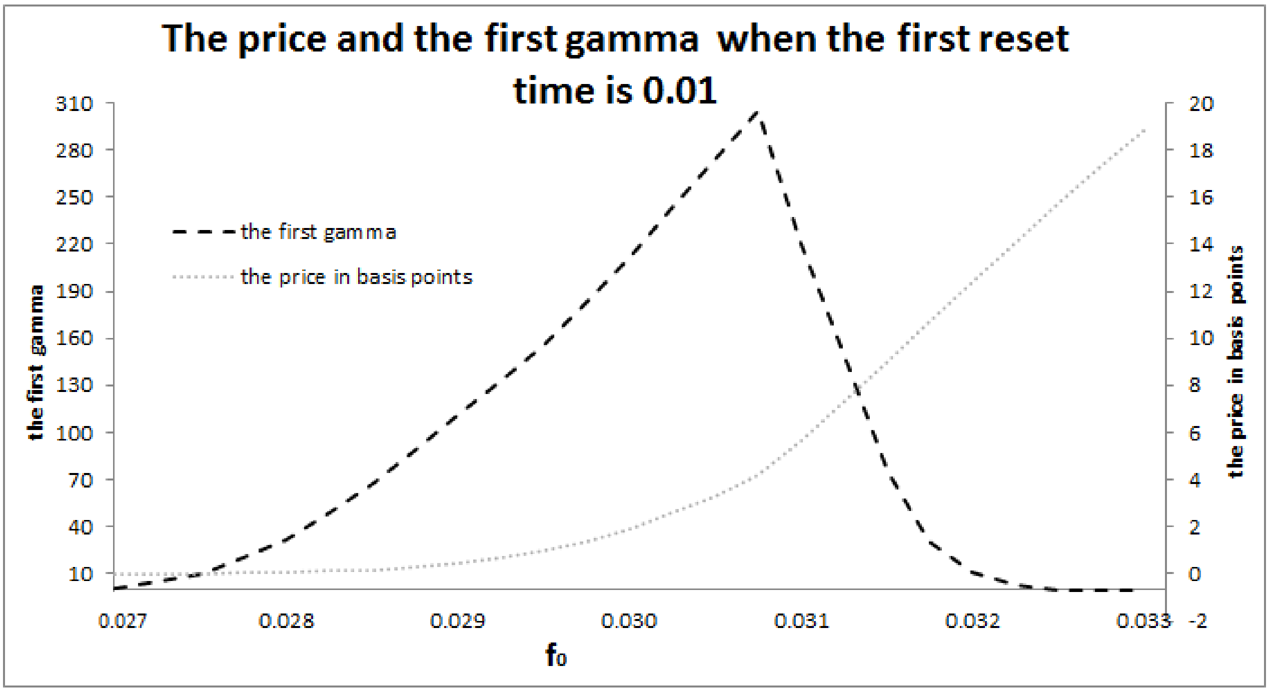

Figure 4.3: The price and the first gamma of a ten-rate cancellable swap with varying the initial value of f_0 from 0.027 to 0.033, with 65,535 paths for the first pass and 50,000 paths for the second, when $T_0 = 0.01$.

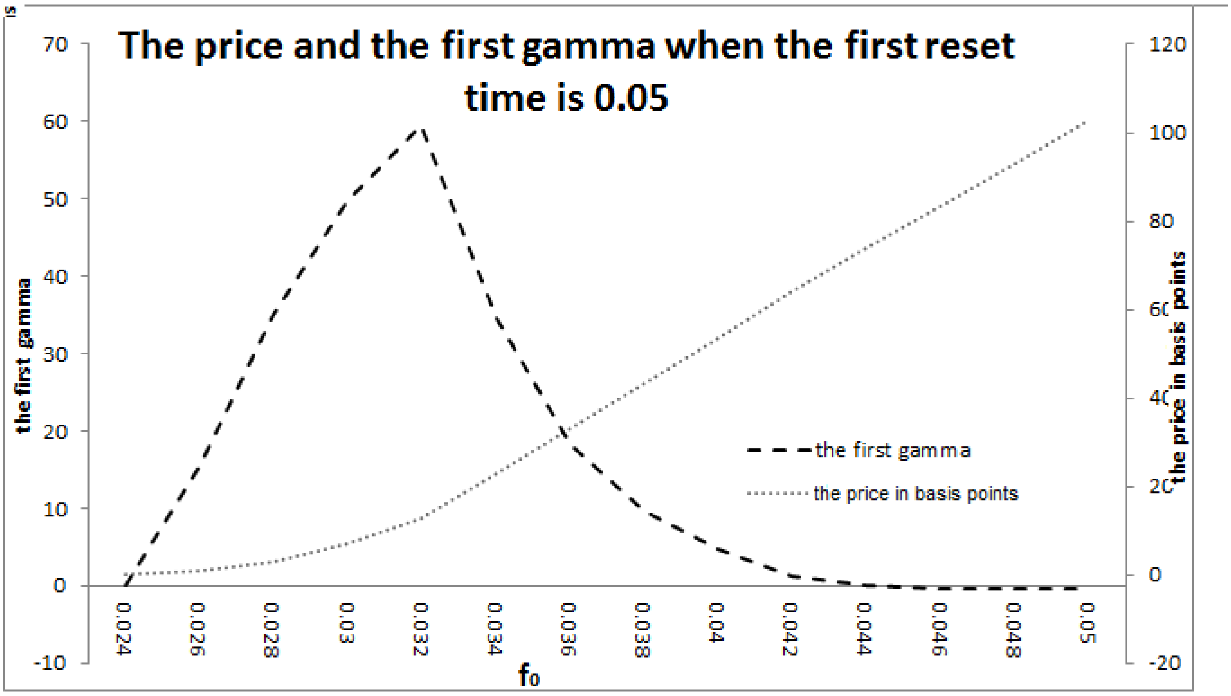

Figure 4.4: The price and the first gamma of a ten-rate cancellable swap with varying the initial value of f_0 from 0.024 to 0.05, with 65,535 paths for the first pass and 50,000 paths for the second, when $T_0 = 0.05$.

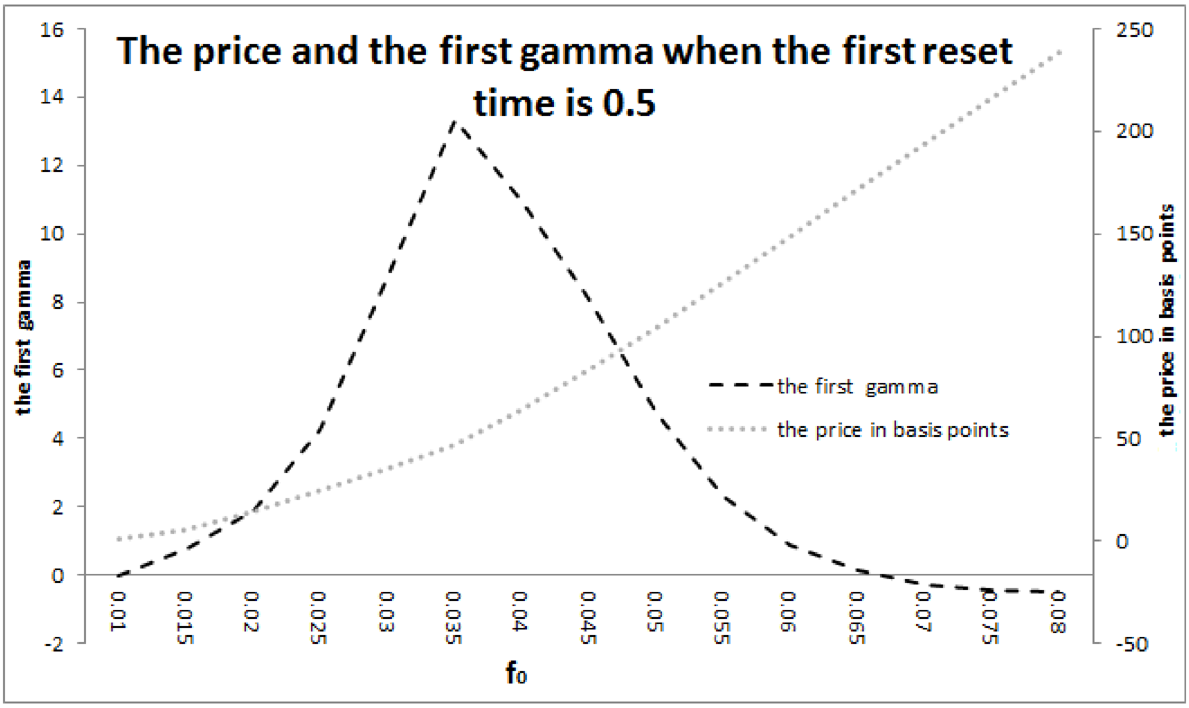

Figure 4.5: The price and the first gamma of a ten-rate cancellable swap with varying the initial value of f_0 from 0.01 to 0.08, with 65,535 paths for the first pass and 50,000 paths for the second, when $T_0 = 0.5$.

T_0	Max Gamma	Max price	Min Gamma	Min Price
0.01	304.60 (3.9853 S.E.) when $f_0 = 0.0308$	18.885 bps (6.571E-05 S.E.) when $f_0 = 0.033$	-0.4635 (7.403E-04 S.E.) when $f_0 = 0.033$	0.00011bps (1.549E-07 S.E.) when $f_0 = 0.027$
0.05	59.61 (1.77468 S.E.) when $f_0 = 0.032$	102.683 bps (7.563E-05 S.E) when $f_0 = 0.05$	-0.4708 (0.001601 S.E.) when $f_0 = 0.05$	0.19176 bps(5.409E-06 S.E.) when $f_0 = 0.024$
0.5	13.34 (0.4235 S.E.) when $f_0 = 0.035$	238.303 bps (7.712E-05 S.E.) when $f_0 = 0.08$	-0.4876 (0.007823 S.E.) when $f_0 = 0.08$	1.481 bps (1.229E-05 S.E.) when $f_0 = 0.01$

Table 4.5: Summary of Figure 4.3-4.5

The three graphs demonstrate that gammas are big near the exercise boundary, and reduce away from it. The big gammas are consistent with the plot of the price demonstrating a greater curvature near the boundary. This is because a small change in the first forward is likely to shift the path across the angularity close to the boundary. As the time to the first reset date draws nearer, gammas of at-the-boundary options increase. We also observe negative gammas for deeply in-the-money cancellable swaps, which is consistent with the structure of the underlying product. The cancellable swap behaves very much like a vanilla swap when the product is deeply in-the-money, and the vanilla swap has negative gammas.

4.4.4 Computational cost

Finally, we investigate the computational cost of the HOMC algorithm. For this experiment, we compute the price and the Greeks (the Hessian and the Deltas are computed simultaneously) with a sample size of 50,000 paths for both the first and second passes for an n-rate at-the-money cancellable swap using the same set-up as before. Since we are using n steps with n rates, the time taken for computing price should have order of n^2, and the Hessian should be n^3. The results are shown in Table 4.6. We also divide the times by n^2 and n^3 to compare the results. These are bounded as expected.

	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
Price	0.0653	0.173	0.327	0.521	0.769	1.044	1.386	1.776
Price $\times n^{-2}$	0.00261	0.00174	0.00145	0.00130	0.00123	0.00116	0.00113	0.00111
Greeks	0.230	1.007	2.462	4.758	8.070	12.695	18.600	27.462
Greeks$\times n^{-3}$	0.00184	0.00101	0.00073	0.00059	0.00053	0.00047	0.00043	0.00043

Table 4.6: Times(in seconds) for pricing and additional times for Greeks, of an n-rate cancellable swap, followed by the same times divided by n^2 and n^3 respectively, using 50,000 paths for both passes

The results show that the measure change performed at each step does not greatly increase the computational burden. Therefore, the HOMC algorithm can be implemented in practice to compute fast and accurate estimates of the Greeks for Bermudan swaptions and cancellable swaps.

4.5 Conclusion

We have successfully derived a methodology for computing the Hessians of Bermudan swaptions and cancellable swaps. The key to our approach is to perform a measure change at each exercise point if the path is near the boundary to ensure that the first-order derivatives of the pathwise estimate of the price are Lipschitz continuous. The measure change is selected so that the variance of the likelihood ratio part is minimized. The pathwise estimate of the price under the new scheme is $\hat{C}^2$, and the algorithmic Hessian method is then applied to calculate the pathwise estimate of the Hessian. Our numerical results suggest that the HOMC algorithm outperforms the PWLR method. The exact and efficient Hessian of the price computed by HOMC would be used in practice for hedging Bermudan swaptions and cancellable swaps.

Chapter 5

Inexplicit Critical Value Functions

The simulation schemes in the previous chapters are defined in terms of the critical value functions, which can be described as some proxy constraint function equaling a certain level. The change of variable is performed at every step on the first standard uniform, $v_{i,1}$; it is replaced by a function, $Ui(\theta, v_{i,1})$, defined in terms of the critical values of $v_{i,1}$ for the proxy constraint function to equal the critical level. Therefore, we need to find these critical values of $v_{i,1}$ explicitly. For example, for a digital basket option, the proxy constraint function is the average price of stocks; the critical value is the value of the first uniform such that the average price equals the strike. For cases where we are unable to find closed-form solutions, we use two iterations of the Newton-Raphson method to approximate these critical values numerically.

In this chapter, we apply the HOPP(2) and the HOPP(1) schemes to calculate the Hessian of both equity and interest-rate derivatives with inexplicit critical value functions. In addition to derivative products, we also apply our simulation scheme to computing second-order derivatives of insurance contracts, since sensitivity analysis is one of the key risk management practices in pricing and reserving of insurance contracts. There is an extensive amount of insurance literature focusing on the first-order sensitivities. Insurance contracts often provide guaranteed minimum benefits in an event of death, ill, or withdrawal, these guarantees represent embedded derivatives in the liabilities that are often complex, path dependent options. A pathwise approach was introduced by Hobbs et al (2009) to provide unbiased sensitivity estimators. Valuation of pension funds are subject to actuarial assumptions such as mortality rates, interest rates and levels of salary inflation, Joshi and Pitt (2010) applied the adjoint method to assess sensitivities of valuation results for pension funds to these key inputs. However, the Global Financial Crisis that started in 2008 alerted the insurance industry to the crucial importance of higher order sensitivities in their hedging programs. Here, we apply the HOPP(1) scheme to calculate the Hessian of a portfolio insurance product, the North Guarantee, which has an angular payoff as a function of the underlying stocks.

5.1 The Newton Raphson method with one-step

The Newton Raphson method is typically used to find the solution of $\beta(v) = D$, i.e. the critical value of the parameter, v, such that the proxy constraint function, β, equals a certain level. Given an initial guess, v_0, one can approximate the proxy constraint function

$$\hat{\beta}(x) = \beta(v_0) + \beta'(v_0)(v - v_0),$$

then equate this approximation to D to obtain an estimate of the solution,

$$v_1 = v_0 + \frac{D - \beta(v_0)}{\beta'(v_0)}.$$

For the first-order OPP, Chan and Joshi (2015) used this approach to approximate $a_{j,i}$. Kreyszig(1988), Chapter 18 shows that the convergence of this one-iteration approximation is quadratic, thus the bias resulted from using unbumped strata for the bumped path is of order $\mathcal{O}(||\theta - \theta_0||^2)$. The bias disappears at the limit after division by $||\theta - \theta_0||$. However, this is not sufficient for calculating second-order derivatives using HOPP(2), since the bias does not disappear after division by $||\theta - \theta_0||^2$.

5.2 Numerical approximation of the critical value functions

Our solution is to use two iterations of the Newton-Raphson method to approximate the critical value functions. That is, given the estimate from the first iteration, v_1, we can approximate the function as

$$\hat{\beta}(v) = \beta(x_1) + \beta'(v_1)(v - v_1),$$

then obtain a second estimate of the solution,

$$v_2 = v_1 + \frac{D - \beta(v_1)}{\beta'(v_1)}.$$

The convergence rate of this two-iteration approximation is fourth order (Press et al, 2007, Chapter 9). Thus, it is sufficient for our purpose of estimating second-order derivatives. In this section, we explain the exact implementation of the two-iteration Newton-Raphson subroutine.

Assume that there exists a proxy constraint function, β_i, at T_i, such that for $k = 0, 1, ..., M_i$, the kth point of discontinuity arise from

$$D_{i,k}(\theta) = \beta_i\Big(S_i^*, \theta, v_{i,1}\Big),$$

where $D_{i,k}$ is the kth discontinuity level of the discounted payoff function. For example, the proxy constraint function for the digital basket option is the average of the basket and the discontinuity level is the strike; the proxy constraint function determines a strike crossing of the discounted payoff function. Our objective is to find the solution of this problem given S_i^* and θ, i.e. the critical value function $a_{i,j}$.

We perform the linearization using standard normals, that is to find the critical value of the first standard normal, $Z_{i,1}^*$, such that

$$D_{i,k}(\theta) = \beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1}^*)\Big),$$

to ensure that the approximation of $a_{i,j} = \Phi(Z_{i,1}^*)$ lies in the range $(0, 1)$. We also assume that the function, $\beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1})\Big)$, has a positive derivative with respect to $Z_{i,1}$. The approximated critical values, $\hat{a}_{i,j}(\theta, v_{i,1})$, are calculated with the following procedures.

1. *We approximate the proxy constraint function, β_i, about the initial guess, i.e. $Z_{i,1} = \Phi^{-1}(v_{i,1})$, the simulated first standard normal,*

$$\hat{\beta}_i^1\Big(S_i^*, \theta, \Phi(Z)Big) = \beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1})\Big) + \frac{\partial\beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1})\Big)}{\partial Z}(Z - Z_{i,1}).$$

2. *We equate this linear approximation, $\hat{\beta}_i^{(1)}\Big(S_i^*, \theta, \Phi(Z)\Big)$, to the discontinuity level, $D_{i,k}(\theta)$, to obtain the first estimate, $\hat{Z}_{i,k}^1$, of the critical standard normal,*

$$\hat{Z}_{i,k}^{(1)}(\theta, v_{i,1}) = \Bigg(D_{i,k}(\theta) - \beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1})\Big)\Bigg)\Bigg(\frac{\partial\beta_i\Big(S_i^*, \theta, \Phi(Z_{i,1})\Big)}{\partial Z}\Bigg)^{-1} + Z_{i,1}.$$

3. *We apply this method again to approximate the proxy constraint function, β_i, about $\hat{Z}_{i,k}^{(1)}$'s and then equate to the discontinuity level. We obtain the second approximation of the critical value,*

$$\hat{Z}_{i,k}^{(2)}(\theta, v_{i,1}) = \Bigg(D_{i,k}(\theta) - \beta_i\Big(S_i^*, \theta, \Phi\big(Z_{i,k}^1(\theta, v_{i,1})\big)\Big)\Bigg)\Bigg(\frac{\partial\beta_i\Big(S_i^*, \theta, \Phi\big(Z_{i,k}^1(\theta, v_{i,1})\big)\Big)}{\partial Z}\Bigg)^{-1} + Z_{i,k}^1(\theta, v_{i,1}).$$

4. *The approximated critical value function is given by*

$$\hat{a}_{i,j}(\theta, v_{i,1}) = \Phi[\hat{Z}_{i,k}^{(2)}(\theta, v_{i,1})].$$

The approximated critical value function, $\hat{a}_{i,j}(\theta, v_{i,1})$, is twice-differentiable with respect to θ, provided that the proxy constraint function, β_i, is twice-differentiable with respect to the parameter

θ. The change of variable function is

$$U_i(\theta, v_{i,1}) = \left(\frac{\hat{a}_{i,j+1}(\theta, v_{i,1}) - \hat{a}_{i,j}(\theta, v_{i,1})}{\hat{a}_{i,j+1}(\theta_0, v_{i,1}) - \hat{a}_{i,j}(\theta_0, v_{i,1})}\right)\Big(v_{i,1} - \hat{a}_{i,j}(\theta_0, v_{i,1})\Big) + \hat{a}_{i,j}(\theta, v_{i,1}), \tag{5.1}$$

when $\hat{a}_{i,j}(\theta_0, v_{i,1}) < v_{i,1} \leq \hat{a}_{i,j+1}(\theta_0, v_{i,1})$. The resulting Monte-Carlo weight is more complicated than the linear case, since the critical value functions, $\hat{a}_{i,j}(\theta, v_{i,1})$, depend on both θ and the first standard uniform. The Monte-Carlo weight is the derivative of the function, $U_i(\theta, v_{i,1})$, with respect to $v_{i,1}$.

5.3 The smallness of the bias arising from using the Newton-Raphson method

The HOPP(2) simulation scheme has the following objectives:

1. When $\theta = \theta_0$, we have the unbumped paths exactly the same as the bumped paths.
2. The expectation of the discounted payoff from the HOPP(2) simulation scheme is equal to the expectation of the discounted payoff under the original scheme.
3. The discounted payoff as a function of θ from the HOPP(2) scheme is $\hat{C}^2$, so we can apply the pathwise method to compute the Hessian. This resulting Hessian computed is equal to the Hessian of the discounted payoff under the original scheme.

Since the critical values, $\hat{a}_{i,j}$'s, are approximations, we still have a small possibility of having the bumped and unbumped paths to finish on different sides of discontinuities.Thus, the HOPP(2) scheme with approximated critical value functions has the first two properties but not the third one. We remove this remaining discontinuity by using the unbumped strata, $E_{j,k}$, for the bumped path. We shall call the resulted pathwise estimator of the price, $\hat{P}^{NR}$. This modified HOPP(2) scheme with approximated critical values has the first and the third properties, but fails to have the second one. In the following theorem, we shall show that the bias created in this modified scheme vanishes when computing first- and second-order derivative estimates.

Theorem 5.3.1. *Using the unbumped strata, $E_{j,k}$, for the bumped path creates a bias in the bumped path's expectation that vanishes to fourth order, i.e.*

$$\mathbb{E}[\hat{P}^{NR}(\theta)] - \mathbb{E}[P(\theta)] = \mathcal{O}\Big(||\theta - \theta_0||^4\Big).$$

Proof. We focus on the behaviour of the bumped path close to the critical points, this is because

- when $v_{i,1}$ equals the actual critical value $a_{i,j}$, the approximated critical value is correct, i.e, $\hat{a}_{i,j} = a_{i,j}$;

- both $\hat{a}_{i,j}$ and $a_{i,j}$ are smooth functions of θ;
- the change of variable function is a smooth function between the critical points and will be extendible to the closed interval, and it is also the identity mapping in $v_{i,1}$ when $\theta = \theta_0$.

Therefore, the change of variable function will not map the bumped path across a critical point for θ lying in a small neighbourhood of θ_0, unless $v_{i,1}$ is in a small neighborhood of the true critical value.

We only show the one-step case with a single discontinuity, and v is just above or on a critical point since the case of below follows by symmetry. Without loss of generality, we assume that $a_0(\theta_0) = 0$, and the support of the first random uniform is $[-\frac{1}{2}, \frac{1}{2}]$. We need to show that, the probability of the unbumped path is above the discontinuity while the bumped path actually finishes below the discontinuity is $\mathcal{O}(||\theta - \theta_0||^4)$, that is

$$\mathbb{P}[v > 0, u(v,\theta) < a_0(\theta)] = \mathcal{O}\Big(||\theta - \theta_0||^4\Big).$$

We adopt the change of variable as equation (5.1)

$$u(v,\theta) = \hat{a}_0(\theta, v) + \frac{\hat{a}_1(\theta, v) - \hat{a}_0(\theta, v)}{\hat{a}_1(\theta_0, v) - \hat{a}_0(\theta_0, v)}\big(v - \hat{a}_0(\theta_0, v)\big).$$

We rewrite as

$$u(v,\theta) = v + [\hat{a}_0(\theta, v) - \hat{a}_0(\theta_0, v)] + \frac{(\hat{a}_1(\theta, v) - \hat{a}_0(\theta, v)) - \hat{a}_1(\theta_0, v) - \hat{a}_0(\theta_0, v)}{\hat{a}_1(\theta_0, v) - \hat{a}_0(\theta_0, v)}(v - \hat{a}_0(\theta_0, v)).$$

We first consider the third term

- it is a smooth as a function of both θ and v;
- it vanishes at both $\theta = \theta_0$ and $v = 0$.

We can then express this term as $v(\theta - \theta_0)^T \chi(v, \theta)$ by Taylor's Theorem, where χ is a smooth vector-valued function. So, we have

$$u(v,\theta) - a_0(\theta) = v + \big[\hat{a}_0(\theta, v) - \hat{a}_0(\theta_0, v) - a_0(\theta)\big] + v(\theta - \theta_0)^T \chi(v, \theta).$$

Now consider the second term

- it is a smooth function of both v and θ;
- when $\theta = \theta_0$, it vanishes;
- when $v = 0$, we have $\hat{a}_0(\theta_0, 0) = 0$;
- when $v = 0$, $\hat{a}_0(\theta, 0) - a_0(\theta)$ vanishes to fourth order in $\theta - \theta_0$, since two iterations of the Newton-Raphson method are used.

Applying Taylor's theorem, the expression becomes

$$u(v,\theta) - a_0(\theta) = v + (\theta-\theta_0)^T v\varpi(v,\theta) + v(\theta-\theta_0)^T\chi(v,\theta) + \vartheta(v,\theta)$$

$$= v + v(\theta-\theta_0)^T\kappa(v,\theta) + \vartheta(v,\theta),$$

where $\kappa = \chi + \varpi$, and $\vartheta(v,\theta) = \mathcal{O}\Big((||\theta-\theta_0||)^4\Big)$ is a smooth function of both v and θ. Now κ is smooth, and in a small enough neighbourhood in θ_0, we have $v(\theta-\theta_0)^T\kappa > -v/2$, so on that set, our event is contained in

$$v/2 + \vartheta(v,\theta) < 0,\ v > 0.$$

Since $\vartheta(v,\theta) = \mathcal{O}\Big(||\theta-\theta_0||^4\Big)$, the event is smaller than

$$v < C(||\theta-\theta_0||)^4,\ v > 0,$$

for some constant C, which is $\mathcal{O}\Big(||\theta-\theta_0||^4\Big)$. We have shown that the set on which $\hat{P}^{NR}(\theta)$ disagrees with the version using the bumped strata has probability vanishing to fourth order. Since it is a finite linear combination of integrable functions which are C^2 times indicator functions of strata, this shows that their difference of expectations also vanishes to fourth order. Since the expectation of $P(\theta)$ equals that of the bumped strata version, the result is now immediate.

□

For our HOPP(2) scheme, i.e. $\theta \to \theta_0$, the bias disappears after division by $||\theta-\theta_0||^2$. Hence, the HOPP(2) estimator of Hessian is unbiased. However, the measure change determined by the approximated critical values is no longer fully optimal. Therefore, it is preferable to first apply a simple mapping to β_i, so it is closer to linear, and then the measure change with the Newton-Raphson subroutine will be closer to optimal. For example, instead of linearizing β_i, it is sometimes better to linearize $\log(\beta_i)$.

5.4 Application to the multi-dimensional Black-Scholes model

We consider

1. a digital basket option, with payoff as follows

$$H\Bigg(\sum_{i=1}^{n}\frac{1}{n}S_i(T) > K\Bigg);$$

2. a parabolic Asian basket option, with payoff as follows

$$\Bigg(\Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n}\frac{1}{n}S_i(T_j) - K\Big)_+\Bigg)^2,$$

where H is the Heaviside function.

For every step of the HOPP(2) scheme, we first evolve all the stocks values ordinarily except the first standard uniform,

$$S_i^*(T_j) = S_i(T_{j-1}) \exp\Big(r\triangle T - 0.5a_{i,1}(T_{j-1})^2 + \sum_{f=2}^{F} \big(a_{i,f}(T_{j-1})Z_{j,f} - 0.5a_{i,f}(T_{j-1})^2\big)\Big),$$

then in the second phase, the change of variable is performed as described in Section 3.1,

$$S_i(T_j) = S_i^*(T_j) \exp\left(a_{i,1}(T_{j-1})\Phi^{-1}\Big(U_j^*\big(K^*(\theta), v_{j,1}\big)\Big)\right),$$

for $i = 1, 2, \ldots, n$.

Since the likelihood ratio and the pathwise likelihood ratio methods are not applicable when the density function is singular, i.e. $F < n$, the numerical experiments are conducted using full-factor models to compare the performance of HOPP(2) with the other methods.

To show the convergence of the Hessian computed by HOPP(2), we apply it to calculate the Hessian of the parabolic Asian basket option. We present the hypothetical parabolic Asian basket option here, so we can compare the Hessians calculated by HOPP(2) and the pathwise method. We use a five-asset model with $\rho_{i,j} = 0.6$, $\sigma_i = 0.5$ for $i, j = 1, 2, \ldots, n$, and $r = 0.05$ per annum. The initial values of the stocks are 100 and the barrier is 100. The expiry time is 0.5, with 15 observation dates equally space across $[0, T]$, the sample size is 20,000 paths for both HOPP(2) and the pathwise method. We present the results for only the first Gamma and Cross-Gamma in Table 5.1, since the product and the parameter inputs are symmetric for all stocks. The mean values calculated by the two methods converge; the pure pathwise method produces slightly smaller standard errors, but would not be applicable for non-parabolic payoffs.

We compare the HOPP(2) method with the pure likelihood ratio method, by comparing the mean and the standard errors of the Hessian calculated by the two methods on a digital basket option. For numerical experiment, we implement a five-asset basket with initial stocks equal to 88, the strike is 100. We set the volatility to be 0.1 and the correlation to be 0.6. The interest rate is 0.05 and the maturity time is 0.7. The sample size is 20,000 paths for HOPP(2) and 6,000,000 paths for the likelihood ratio method, since the values calculated are of similar degree of precision in terms of the standard error, see Table 5.1.

	HOPP(2) Mean	HOPP(2) S.E.	PW Mean	PW.S.E
Parabolic Asian first Gamma	0.7896	0.0065	0.7886	0.0063
Parabolic Asian first Cross-Gamma	0.7744	0.0062	0.7733	0.0060
	HOPP(2) Mean	HOPP(2) S.E.	LR Mean	LR.S.E
Digital basket first Gamma	2.06×10^{-4}	4.916×10^{-6}	2.161×10^{-4}	6.534×10^{-6}
Digital basket first Cross-Gamma	2.055×10^{-4}	4.890×10^{-6}	1.98×10^{-4}	4.737×10^{-6}

Table 5.1: The Comparison of HOPP(2) with the pathwise method on a parabolic Asian basket option both with a 20,000 paths sample, and the comparison of HOPP(2) with a 20,000 paths sample with the likelihood ratio method with a 6,000,000 paths sample on a digital basket option

5.4.1 Computation times

In this section, we investigate the computational timing of HOPP(2) for calculating Greeks under the multi-dimensional Black-Scholes model. In particular, we focus on digital basket options to demonstrate the efficacy of the method with Newton-Raphson subroutines. The algorithmic Hessian method by Joshi and Yang (2011) is able to compute all the Gammas with order $O(nFNM)$. For the following examples of digital basket options, we have $N = 1$, set $F = 4$ and use $M = n + 6$ for evolving the processes. The other parameters are the same as in the previous subsection. We vary the number of assets in the basket, n, to assess the computational burden of our HOPP(2) method. For these experiments, we compute the price and the Greeks (the Hessian and the Deltas are computed simultaneously) with 20,000 paths samples. To demonstrate the timing difference, we also provide the ratio of the additional time of computing all Greeks using HOPP(2) to the time of computing the price.

In the first example, we consider an out-of-the-money digital basket option, i.e. the current price level is 88 and the strike is 100. The timing results in Table 5.2 demonstrate that HOPP(2) is very efficient; the additional time for computing Greeks is less than n times of that for computing the price. This is because HOPP(2) only needs to compute pathwise sensitivities if the path finishes in the money, and the pathwise Greek estimates are zero otherwise. For an out-of-the-money option, it is unlikely for a path to finish in the money, thus it does not require much additional effort to compute the Greeks.

n	5	10	15	20	25
Price times	0.025	0.034	0.043	0.051	0.061
Additional HOPP(2) Greek times	0.023	0.058	0.118	0.176	0.254
The ratio of Greek times to price times	0.92	1.71	2.74	3.45	4.16

Table 5.2: Time(in seconds) of pricing and calculating the Deltas and the Hessian simultaneously of an out-of-the-money digital basket with 20,000 paths samples

In the second example, we consider an in-the-money digital basket option, i.e. the current price level is 110 and the strike is 100. For these in-the-money options, an additional effort is required to compute the Greeks on almost all paths. That is, we need to apply the change of measure on every non-zero path with the Newton-Raphson's search to ensure the algorithm is $\hat{C}^2$. The timing results in Table 5.3 demonstrate that the additional computational cost is not significant; the additional time for computing Greeks is only slightly more than n times of that for computing the price. In particular, for the case of $n = 25$, the additional time of computing 325 second-order derivatives and 25 Deltas is only about 27 times more than computing the price alone.

n	5	10	15	20	25
Price times	0.024	0.033	0.042	0.053	0.062
Additional HOPP(2) Greek times	0.213	0.438	0.773	1.149	1.675
The ratio of Greek times to price times	8.88	13.27	18.40	21.68	27.02

Table 5.3: Time(in seconds) of pricing and calculating the Deltas and the Hessian simultaneously of an in-the-money digital basket with 20,000 paths samples

5.5 Application to Derivative Products with Angular Payoffs

The North Guarantee is an insurance product, which is dependent on a basket of assets. It provides investors capital guarantees over their investments while still benefiting from any market growth in their selected investments. There are many portfolio insurance products available in the market with similar features, here we refer to the one provided by the insurer, AMP (AMP, 2009). An investment strategy (from two or three options) and the term of the contract (from 6, 8, 10 or 20 years) are chosen at the commencement, the investor cannot change the term or the investment strategy during the term of the contract, but can switch investment options within the chosen investment strategy. However, we do not model this option in our simulation exercises.

The payoff to the insured is defined in terms of the account value and the protected balance of the insured's account.

- The account value is the current market value of the underlying assets, after insurance premiums deducted as a percentage of the value monthly. It is subject to ordinary market fluctuations.
- The protected balance locks in any growth as a result of positive investment performance each year or every two years, provided that the account value exceeds the protected balance. For example, the initial market value of the basket is worth one million, at the first observation date after the commencement, the account value is the current value of the basket net of the insurance premiums
  - if the account value is more than one million, the protected value then locked in as the current account value,

– else, the protected value is one million.

The protected balance is not affected by negative market performances. At maturity, it is the maximum account value observed on the annual or biennial observation dates before the maturity.

Without the guarantee, only the account value at maturity is paid to the insured. With the guarantee, the terminal amount paid to the insured is the maximum of the protected balance and the account value observed at the maturity. Therefore the payoff to the insured is the difference between the two, i.e.

$$(\text{Protected Balance} - \text{Account Value})_+.$$

The discounted payoff function of this product as shown above is angular, so we apply HOPP(1) to calculate the Hessian of this product with respect to the initial asset values.

For numerical experiments, we implement a 20-year product with a five-asset basket, and $19+t_0$ years to the maturity (for $0 < t_0 < 1$, this is after its commencement). The observation dates are at t_0, $1 + t_0$, $2 + t_0$, $\dots$, $19 + t_0$. At t_0, the product is exactly one year from the commencement. The account values are charged with insurance premiums 0.3$\%$ monthly in arrears, so the account value at t_0 is $0.997^{12}\sum_{i=1}^{5} S_i(t_0)$. The premium is charged as a percentage of the account value, so we only evolve the assets over the observation dates. Since the investors are given the right to choose different investment options with different degrees of riskiness within each investment strategy, the volatilities are set to be 0.02, 0.06, 0.2, 0.3 and 0.4 per annum, the correlations are 0.6. The interest rate is 0.05 per annum.

We plot the sum of standard errors of the Hessian calculated by the pathwise likelihood ratio method and that by HOPP(1) with varying first observation date t_0 from 0.01 to 0.1 in Figure 5.1, the sample size is 20,000 paths. Here, we set the current stocks to be 80 and the original stocks to be 100. We observe the pathwise likelihood ratio method produces very large standard errors when the first observation date is small. The HOPP(1) method unlike the pathwise likelihood ratio method produces very stable standard errors as the first observation date changes. For the same set of t_0, the maximum sum of standard errors calculated by HOPP(1) is 0.01505 when $t_0 = 0.1$ compared to 0.44294 by the pathwise likelihood ratio method; the minimum is 0.01470 when $t_0 = 0.015$, compared to 1.13244 by the pathwise likelihood ratio method.

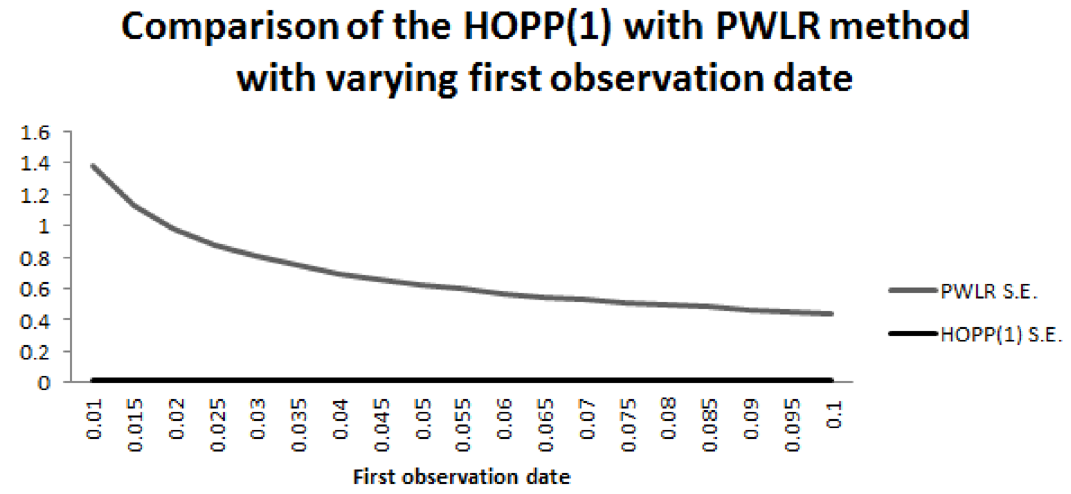

Figure 5.1: North Guarantee: The sum of standard errors of the Hessian calculated by the PWLR method and HOPP(1) with varying initial observation date, with 20,000 paths samples

We plot the sum of standard errors of the Hessian calculated by the pathwise likelihood ratio method and that by HOPP(1) with varying current stock price level from 100 to 50 in Figure 5.2, the sample size is 20,000 paths. Since when the current stock level is above the initial stock prices, the Gamma estimates are very small, we only consider the cases where the current stock is below the original stock level. Here the original stock prices are 100 and the first observation time is 0.1. The maximum sum of standard errors by HOPP(1) is 0.117088 when the stock level is 90, compared to 0.421071 by the pathwise likelihood ratio method; the minimum by HOPP(1) is 0.01463 when the stock level is 78, compared to 0.45681 by the pathwise likelihood ratio method.

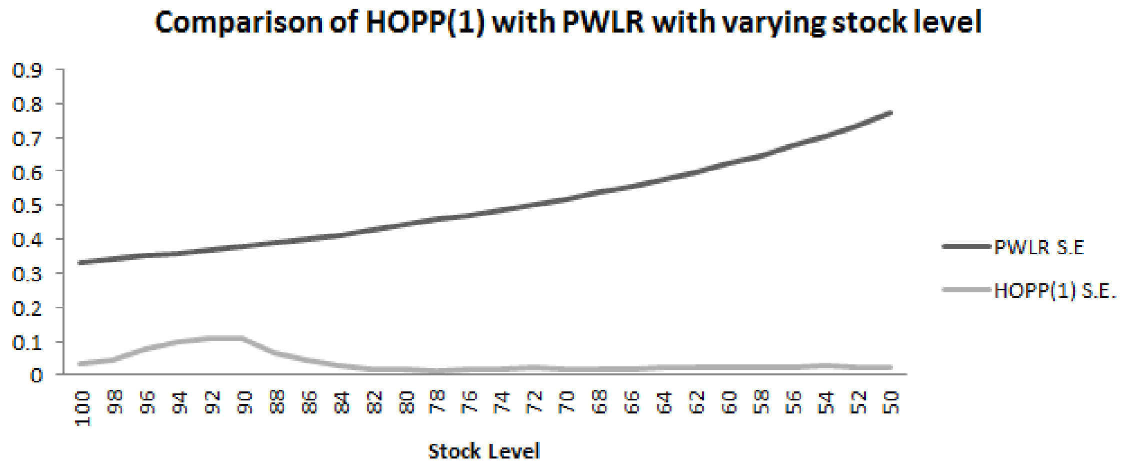

Figure 5.2: North Guarantee: The sum of standard errors of the Hessian calculated by the PWLR method and HOPP(1) with varying current stock values, with 20,000 paths samples

When the first observation date is small and the current stock level is far below the original stock level, the pathwise likelihood ratio method produces very large standard errors. We set the current stock price to be 55 and the first observation date to be 0.01. Since the pathwise likelihood ratio method produces also very large standard errors of Cross-Gammas when the volatilities are different, for this particular example, we use volatility 0.1 for all assets in order to compare the Hessians calculated by the two methods. We compare the result by HOPP(1) on a 20,000 paths sample to the result calculated by the pathwise likelihood ratio method on a 14,000,000 paths sample, so the Hessians calculated by the two methods have similar precision, see Table 5.4

	HOPP(1) Mean	HOPP(1) S.E.	PWLR Mean	PWLR.S.E
North Guarantee first Gamma	2.0257×10^{-3}	2.594×10^{-4}	2.418×10^{-3}	2.972×10^{-4}
North Guarantee first Cross-Gamma	1.836×10^{-3}	2.412×10^{-4}	2.0187×10^{-3}	2.972×10^{-4}

Table 5.4: North Guarantee: The comparison of the Hessian calculated by HOPP(1) with a 20,000 paths sample and by the pathwise likelihood ratio method with a 14,000,000 paths sample

When the first observation date is long and the current stock level is close to the original stock level, the pathwise likelihood ratio method achieves its best result. For the next numerical example, we set the first observation date to be 1 and the current stock equal to the original stock level at 100, see Table 5.5, 5.6 and 5.7 in the chapter appendix.

Unlike HOPP(1), the pathwise likelihood ratio method generates very large standard errors when

- the volatility is very small,
- the differences between the volatilities are large,
- the first observation time is small.

5.6 Conclusion

In this chapter, we explored the issue of inexplicit proxy constraint functions. We used the Newton-Raphson method with two steps for products with discontinuous payoffs and with one step for products with angular payoffs. As we are no longer working with the original critical value functions, this might lead to bias in estimating the true expected price. However, we demonstrated that there is no bias when computing price sensitivities. Numerical results showed that the computational cost associated with approximating the critical values is small.

Chaper Appendix: Tables of Numerical Results

HOPP(1) vs PWLR	S_1	S_2	S_3	S_4	S_5
S_1	0.8069 vs 0.0207	0.8237 vs -0.3237	0.761 vs -0.3672	0.6349 vs -3.8794	0.525 vs 5.4115
S_2	0.8237 vs -0.3237	0.8705 vs 1.3621	0.7693 vs 1.1033	0.642 vs 0.6253	0.5222 vs -2.0246
S_3	0.761 vs -0.3672	0.7693 vs 1.1033	1.0118 vs 1.1948	0.504 vs 0.3869	0.3745 vs 1.7844
S_4	0.6349 vs -3.8794	0.642 vs 0.6253	0.504 vs 0.3869	1.0092 vs 1.4486	0.1535 vs 0.4795
S_5	0.525 vs 5.4115	0.5222 vs -2.0246	0.3745 vs 1.7844	0.1535 vs 0.4795	1.0664 vs 0.2248

Table 5.5: North-Guarantee: Mean×1000 of the Hessian calculated by the pathwise likelihood ratio method and by HOPP(1) using 20,000 paths samples

HOPP(1) vs PWLR	S_1	S_2	S_3	S_4	S_5
S_1	0.1037 vs 0.9191	0.104 vs 0.8414	0.1177 vs 1.5667	0.1239 vs 3.0328	0.1009 vs 6.4483
S_2	0.104 vs 0.8414	0.1066 vs 0.283	0.1182 vs 0.522	0.1238vs 1.101	0.1003 vs 2.2495
S_3	0.1177 vs 1.5667	0.1182 vs 0.522	0.1631 vs 0.1596	0.1331vs 0.3033	0.1097 vs 0.6264
S_4	0.1239 vs 3.0328	0.1238 vs 1.101	0.1331 vs 0.3033	0.182vs 0.2171	0.1085 vs 0.5039
S_5	0.1009 vs 6.4483	0.1003 vs 2.2495	0.1097 vs 0.6264	0.1085 vs 0.5039	0.1703 vs 0.3053

Table 5.6: North-Guarantee: Standard Errors×1000 of the Hessian calculated by the pathwise likelihood ratio method and by HOPP(1) using 20,000 paths samples

HOPP(1) vs PWLR	S_1	S_2	S_3	S_4	S_5
S_1	8.866	8.091	13.31	24.473	63.877
S_2	8.091	2.655	4.416	8.896	22.427
S_3	13.31	4.416	0.978	2.279	5.708
S_4	24.473	8.896	2.279	1.193	4.643
S_5	63.877	22.427	5.708	4.643	1.793

Table 5.7: North-Guarantee:Ratio of the standard error of the Hessian calculated by the pathwise likelihood ratio method divided by the standard errors of the Hessian calculated by HOPP(1) using 20,000 paths samples

Chapter 6

First- and second-order sensitivities of functions simulated by rejection techniques

6.1 Introduction

Simulation is a widely-used tool in Operations Research. It can be viewed as a way to compute statistics of complex systems evolving under randomness or as a methodology for evaluating high-dimensional integrals. However, it is not enough just to compute values, for many applications the sensitivities of these values with respect to model inputs is equally important. When the parameters are estimated, knowing how inaccuracy in inputs affects the accuracy of outputs is crucial for decision making. When the parameters are known precisely but themselves random, sensitivities are an essential tool for risk assessment and designing risk mitigation strategies. This is particularly the case when using Monte Carlo simulation to price complex financial derivative products. For many OR applications, the objective is optimize some quantity as a function of the inputs, this is facilitated by good estimates of not just the gradient but also of the Hessian.

A widely-used intrinsically-discontinuous simulation method is rejection sampling and there has been surprisingly little work on how to implement the pathwise method for it. Rejection techniques such as the acceptance-rejection method (Neumann, 1951), the ratio-of-uniforms method (Kinderman and Monahan, 1977) and the transformations-with-multiple-root method (Michael et al, 1976), are powerful alternatives for simulating random variates when the conventional inverse-transform method is not applicable. For example, various rejection methods are introduced to simulate the gamma and inverse-gaussian variates, which are essential for pricing exotic derivatives with the variance-gamma (VG) and the normal-inverse-gaussian (NIG) stock process. Furthermore, in the study of queuing systems, the rejection method "thinning" (Lewis and Shedler, 1979) is commonly used for simulating non-homogeneous Poisson arrival processes. The fundamental issue for all these methods is that a small change in parameter values may alter the decision at each acceptance test and so render a very different outcome for a given sample. None of the adaptations of the pathwise method in the literature appear to be applicable to rejection sampling.

One alternative approach is to differentiate the underlying density and this yields the likelihood ratio method. However, this requires easy computation of the density and also can lead to high variances so pathwise techniques are preferable when available. For many cases where rejection sampling is natural, the density is intractable. In recent work Volk-Makarewicz (2014) considers a similar problem for derivatives prices using the Variance Gamma process via the measure-valued differentiation method. but does not compute sensitivities of the shape parameter for the gamma random variable. There does not seem to have been any work on the NIG process using MVD.Other related approaches are the vibrato technique of Giles (2009) and Malliavin techniques (Benhamou, 2003). Whilst all of these approaches are interesting, none of them are obviously adaptable to the case of rejection sampling.

An important aspect of pathwise methods is that the simulation algorithm is also differentiated. This implies a dependency on its choice. This can cause issues both in terms of applicability when the output of the algorithm is not continuous, and also of variance since the estimates depend on the derivatives of terms in the algorithm. In Chapter 3, we extended the Optimal Partial Proxy method (Chan and Joshi, 2015) to computing Hessians for financial products with discontinuous payoffs (HOPP). Here, we further introduce a change of measure function at each acceptance test, so the small bump in the parameter of interest does not alter the acceptance-rejection decision rendering the pathwise method applicable at the cost of a likelihood ratio term which we show is in a certain sense of minimal variance.

Practical problems in OR often involve multiple parameters of interest. Here, we adopt the Algorithmic Hessian approach introduced by Joshi and Yang (2011) for convenience. We name the simulation algorithm resulting from combining our discontinuity removed rejection algorithm with the JoshiYang method, the Optimal Sensitivities for Rejection Sampling (OSRS) method.

We emphasize the widespread applicability of OSRS: it does not require the simulation algorithm nor the payoff function to be continuous; the marginal density is not explicitly needed; and the measure changes are benign and result in much smaller variances than LR when both are applicable. We present applications to both financial engineering and queuing systems to illustrate the methods breadth. We present results for risk management of call and barrier options using VG and NIG processes: we are able to compute first- and second-order sensitivities to all parameters including the Gamma processs shape parameter. In addition, we consider queuing systems that fail to satisfy the conditions in Glasserman (1991) for the application of the pathwise method. We present sensitivities of such discrete-event systems with respect to all parameter inputs.

The remaining sections of this paper are organized as follows. The basic idea of OSRS is presented in Section 6.2. In Section 6.3, we apply our OSRS to computing sensitivities of call options and barrier options with L$\acute{e}$vy-driven underlyings. In Section 6.4, we apply the model to compute parameter sensitivities of the average time spent of a finite-time horizon $M_t|M|1$ queue, where the interarrival time is simulated by the thinning technique.

6.2 The Basic idea of OSRS

In this section, we provide a general framework for computing sensitivities of expectations of performance measures calculated via rejection techniques. Computing sensitivities is considered particularly hard when the underlying state variables can only be obtained via rejection sampling. The LR method is problematic since the underlying distributions for such cases do not have known and tractable probability density functions, and it has a tendency to produce high variances. The FD method produces biased estimates and it is not feasible for rejection sampling in that a small bump in the parameter of interest can cause a switch between acceptance and rejection, thus a significant change in the simulated outcome. The pathwise method as the limit case of the FD method cannot be applied here due to this inherent discontinuity of the algorithm.

6.2.1 Notations

To fix the notations, let

- $\theta \in \mathbb{R}^m$ denote the parameters of interest within a small neighbourhood Θ about the base point θ_0,
- V_j denote the standard uniforms for generating the jth simulated outcome and $s(\theta, V_j)$ denote the algorithm for generating an outcome,
- V_j^D denote the jth decision standard-uniform variate,
- V denote the standard uniforms for generating a random variate from the target distribution including both V_j's and V_j^D's,
- $N(\theta, V)$ denote the number of simulated outcomes up to and including the accepted one,
- $S(\theta, V)$ denote the rejection algorithm for turning θ and V into the target random variable, so that

$$S(\theta, V) = s(\theta, V_{N(\theta,V)}). \tag{6.1}$$

The value $N(\theta, V) \in \mathbb{N}$ is itself a discrete random variable which depends on the parameter of interest. For example, in acceptance-rejection sampling, each outcome is generated from a related distribution with density function f^*, and accepted as a random variable from the target distribution with density function f, if an independent decision variate, V_j^D, satisfies,

$$V_j^D \leq \frac{f\big(s(\theta, V_j)\big)}{c(\theta) f^*\big(s(\theta, V_j)\big)},$$

for $c(\theta)$ which depends on the distributional parameters. The probability of acceptance is $\frac{1}{c(\theta)}$, and the number of iterations needed follows a geometric distribution.

Let $g(S(\theta, V), \theta)$ denote the performance measure. Our objective is to compute the sensitivities of its expectation,

$$\Upsilon(\theta) = \mathbb{E}[g(S(\theta, V), \theta)],$$

with respect to all parameters of interest. We will assume that g is a C^2 function of θ for fixed S and a piecewise C^2 function of S. See Chapter 3 for the assumptions on the general performance measure.

6.2.2 One-step-one-dimensional case with an identity performance measure

To illustrate the basic idea of OSRS, we consider the case where the performance measure is the random variable itself and $\theta \in \mathbb{R}$. We are interested in the sensitivities of its expectation,

$$\Upsilon(\theta) = \mathbb{E}[S(\theta, V)].$$

Thus, the only discontinuity arises from the simulation algorithm. Glasserman (2004) states that the critical condition for the application of the pathwise method to computing second-order derivatives is for the function to have Lipschitz continuous first-order derivatives. In this section, we construct a new simulation algorithm, $\hat{S}(\theta, V)$, such that

- $\hat{S}(\theta, V)$ is C^2 everywhere,

- $\mathbb{E}[\hat{S}(\theta, V) - S(\theta, V)] = \mathcal{O}(|\theta - \theta_0|^3)$.

The first condition above permits the application of the pathwise method, and the second condition ensures that the results are unbiased estimators of $\Upsilon'(\theta)$ and $\Upsilon''(\theta)$.

We now establish a class of rejection algorithms that OSRS is applicable to.

Assumption 6.2.1. *The original rejection algorithm, $S(\theta, V)$, satisfies:*

- *$s(\theta, V_j)$ is a C^2 function of θ,*

- *there exists a sequence of critical-value functions, $a_j(\theta)$, which are C^2,*

- *the jth outcome is accepted if the decision variate $V_j^D < a_j(\theta)$ and rejected otherwise.*

Given $V_j^D < a_j(\theta_0)$, it is possible to have $V_j^D > a_j(\theta)$ for $\theta \neq \theta_0$. To remove this inherent discontinuity, we perform a measure change, (really just a change of variables,) and replace each V_j^D by a function, $U_j(\theta, V_j^D)$, such that

1. *it is twice differentiable as a function of* θ;
2. *it is piecewise differentiable as a function of* V_j^D *and bijective on* $[0,1]$ *for fixed* θ;
3. $U_j\Big(\theta, a_j(\theta_0)\Big) = a_j(\theta)$;
4. $U_j(\theta_0, V_j^D) = V_j^D$;
5. $\frac{\partial}{\partial\theta} U_j(\theta, V_j^D)_{|\theta=\theta_0, v_j^D=a_j(\theta_0)} = a_j'(\theta_0)$;
6. $\frac{\partial^2}{\partial\theta^2} U_j(\theta, V_j^D)_{|\theta=\theta_0, v_j^D=a_j(\theta_0)} = a_j''(\theta_0)$.

See Section 6.2.3 for discussion of the existence of such a function. As a result of the measure change, the new algorithm adopts the acceptance-rejection decision of the unbumped path for the bumped path, and the point of discontinuity here does not move up to the third order. (See Chan–Joshi (2012) and Joshi–Zhu (2014) for further discussions). There will, of course, be a likelihood ratio weight to compensate for the measure change.

For N acceptance tests in a sample path, we need to perform the change of measure N times. The resulting performance measure is

$$\hat{S}(\theta, V) = s(\theta, V_N(\theta_0, V))W(\theta, V) \tag{6.2}$$

where

$$W(\theta, V) = \prod_{j=1}^{N(\theta_0, V)} \frac{\partial}{\partial V_j^D} U_j(\theta, V_j^D).$$

By (4), we have $U_j(\theta_0, V_j^D) = V_j^D$ so $W(\theta_0, V) = 1$. The likelihood ratio, $\frac{\partial}{\partial V_j^D} U_j(\theta, V_j^D)$, is created from the measure change at the jth test. The final weight is their product; the total number of likelihood ratios depends on the number of acceptance tests in the unbumped path.

Theorem 6.2.1. *The pathwise first-order derivative*

$$\frac{\partial s(\theta, V_N(\theta_0))}{\partial\theta} + s(\theta, V_N(\theta_0))\frac{\partial W(\theta_0, V)}{\partial\theta} \tag{6.3}$$

is unbiased. The pathwise second-order derivative

$$\frac{\partial^2 s(\theta, V_N(\theta_0))}{\partial\theta^2} + 2\frac{\partial s(\theta, V_N(\theta_0))}{\partial\theta}\frac{\partial W(\theta_0, V)}{\partial\theta} + s(\theta, V_N(\theta_0))\frac{\partial^2 W(\theta_0, V)}{\partial\theta^2} \tag{6.4}$$

is unbiased.

See Section 6.A for the proof.

6.2.3 Optimization

Our next objective is to choose the optimal change of measure function such that the variance of the Monte-Carlo implementation is minimized. In Chapter 3, we have shown that the optimal choice is

$$U_j(\theta, V_j^D) = \begin{cases} \frac{a_j(\theta)}{a_j(\theta_0)} V_j^D, & \text{if } V_j^D < a_j(\theta_0), \\ \frac{1-a_j(\theta)}{1-a_j(\theta_0)} \Big(V_j^D - a_j(\theta_0) \Big) + a_j(\theta), & \text{otherwise.} \end{cases} \tag{6.5}$$

for minimizing the variance of the likelihood ratio weight. the cases we consider here, V_j^D is only used for making the acceptance-rejection decision and so the pathwise estimate is a product of the likelihood ratio weight and a function with no dependence on V_j^D. In consequence, since there is no covariance, minimizing the variance of the likelihood ratio weight using(6.5) will also minimize the variance of the resultant pathwise estimator in this class.

6.2.4 The generalization

We allow the state variables, $S \in \mathbb{R}^n$, to be evolved over N^s steps, which may be a random variable, $N^s(\theta) \in \mathbb{N}$, depending on θ. The evolution of S up to step i is

$$\mathbb{K}_i = K_i \circ K_{i-1} \circ ... \circ K_1, \text{ for } i = 1, 2, ...N^s.$$

The evolution of S over one step comprises of the following two sub-mappings:

$$K_i = O_i \circ R_i, \text{ for } i = 1, 2, ..., N^s,$$

such that R_i is a rejection algorithm for simulating one state variable and O_i is an ordinary vector function for evolving the other state variables. We also allow the possibility of more than one state variable to be evolved by rejection algorithms with different R_i's.

We first fix the following notations for R_i, and let

- S_i denote the state variables after each $\mathbb{K}_i$ and S_i^* denote the state variables just before O_i;
- M_i denote the number of critical-value functions in R_i that define the discontinuities of the rejection tests;
- $N_i(\theta)$ denote the number of acceptance test performed in R_i;
- $V_{i,j}^D$ denote the decision variate of the jth test and $V_{i,j}$ denote the standard uniforms for simulating the jth outcome;
- V_i denote the sequence of $V_{i,j}$'s and V_i^D denote the sequence of $V_{i,j}^D$'s.

We now establish the set of rejection functions that our OSRS algorithm is applicable to.

Assumption 6.2.2. *$R_i(S_{i-1}, \theta, V_i, V_i^D) = S_i^*$ satisfies:*

- *It evolves one state variable by a rejection algorithm and is an identity mapping in all other state variables.*
- *It is of the form of equation* (6.1) *and satisfies the first condition of Assumption 6.2.1. Further, there exists a sequence of C^2 critical-value functions such that the jth outcome is accepted if*

  $$a_{i,j}^{(f-1)}(\theta) < V_{i,j}^D \leq a_{i,j}^{(f)}(\theta), \text{ for some } f = 1, 2, ...M_i,$$

  and rejected otherwise.
- *The evolutions, R_i and R_j for $j \neq i$, takes independent sequences of uniforms.*

Thus, the only discontinuities in R_i arise from the acceptance-rejection decisions.

For the ordinary evolution, let V_i^o denote the vector of random uniforms in O_i with a known dimension.

Assumption 6.2.3. *$O_i(S_i^*, \theta, V_i^o) = S_i$ is a C^2 function of both S_i^* and θ for a fixed V_i^o.*

We also assume the performance measure function is consistent with the assumptions made in Section 3.4.

With the above assumptions satisfied, we have obtained a class of functions that have regular behaviours away from the points of discontinuities, and discontinuities occur either at the acceptance test of R_i or $S_i \in \partial E_{i,k}$ defined by g. We now proceed to define a general OSRS algorithm.

A. Removing the discontinuities in acceptance-rejection decisions

First, we consider the discontinuities in R_i. Each decision variate $V_{i,j}^D$ of R_i, for $j = 1, 2, ...N_i$, is replaced by a change of variable function such that

$$U_i(\theta, V_{i,j}^D) = \frac{a_{i,j}^{(f)}(\theta) - a_{i,j}^{(f-1)}(\theta)}{a_{i,j}^{(f)}(\theta_0) - a_{i,j}^{(f-1)}(\theta_0)} \left(V_{i,j}^D - a_{i,j}^{(f-1)}(\theta_0) \right) + a_{i,j}^{(f-1)}(\theta), \tag{6.6}$$

with the corresponding likelihood ratio weight

$$\frac{\partial U_i(\theta, V_{i,j}^D)}{\partial v} = \frac{a_{i,j}^{(f)}(\theta) - a_{i,j}^{(f-1)}(\theta)}{a_{i,j}^{(f)}(\theta_0) - a_{i,j}^{(f-1)}(\theta_0)},$$

if $a_{i,j}^{(f)}(\theta_0) \leq V_{i,j}^D < a_{i,j}^{(f-1)}(\theta_0)$, for $f = 1, 2, ..., M_i$. Here, we shall call the OSRS version of the rejection algorithm, $\hat{R}_i$, which is C^2 as a function of both θ and S_{i-1}.

The simulation algorithm for state variables is C^2 after replacing the R_i's by $\hat{R}_i$'s, because the composition of C^2 functions is C^2. Let $\hat{S}^*(\theta)$ denote the simulated state variables with $\hat{R}_i$'s replacing

R_i's. The resulting algorithm for computing the performance measure is

$$\hat{g}^*(\theta) = g(\hat{S}^*(\theta), \theta) \prod_{i=1}^{N^s} \prod_{j=1}^{N_i(\theta_0)} \frac{\partial U_i(\theta, V_{i,j}^D)}{\partial v}.$$

B.Removing the discontinuities of the performance measure function

If g is a $\hat{C}^2$ function, the pathwise method is applicable to g^* to compute the unbiased pathwise derivatives. However, we have allowed g to be non-smooth in our general framework. The functions' properties in this section are consistent with the ones defined in chapter 3 for the application of the HOPP algorithm. Here, we adopt their idea, and perform one additional change of measure at each time step to remove the pathwise discontinuities of both g and g'.

This additional change of measure on each step is performed on a standard uniform. The appropriate choice of which uniforms depends on the evolution algorithm as well as on the discontinuities of the performance measure function. The chosen V_i^* is replaced by another change of variable function, $U_i^*(\theta, V_i^*)$, of it, so that the points of discontinuities do not move with the parameters of interest θ up to the third order.

Finally, we have a new simulation algorithm for the performance measure function that is $\hat{C}^2$ such that

$$\hat{g}(\theta) = g(\hat{S}(\theta), \theta) W(\theta, V),$$

where $\hat{S}(\theta)$ is the simulation algorithm for the state variables under the new scheme and W is the product of the likelihood ratio weights resulting from the measure changes performed, i.e.

$$W(\theta, V) = \prod_{i=1}^{N^s(\theta_0)} \left\{ \frac{\partial U_i^*(\theta, V_i^*)}{\partial v} \prod_{j=1}^{N_i(\theta_0)} \frac{\partial U_i(\theta, V_{i,j})}{\partial v} \right\}.$$

The resulting function, $\hat{g}(\theta)$, now satisfies Glasserman's conditions for applying the pathwise method to computing first and second-order sensitivities

1. the first-order derivatives of $\hat{g}(\theta)$ are Lipschitz continuous everywhere;
2. the performance measure $\hat{g}(\theta)$ is twice-differentiable almost surely;
3. the FD version of Gamma and Cross-Gamma estimators (Glasserman, 2004) under the new scheme are uniformly integrable.

We can now apply the pathwise method to construct unbiased estimates of the Hessian under OSRS.

6.2.5 Inexplicit critical-value functions

Our simulation algorithm is defined in terms of the discontinuities of the rejection technique, i.e. the critical value functions. Up to this point, we have assumed that they can be found explicitly.

For situations where this is not the case, we adopt Newton-Raphson's method with two steps to numerically approximate these critical-value functions. First, we make the following assumption.

Assumption 6.2.4. *There exists a differentiable proxy-constraint function, $\eta_i(V_{i,j}^D, \theta)$, for each rejection algorithm with strictly positive or negative derivatives with respect to the decision variate such that the simulated jth outcome is accepted if*

$$C^* > \eta_i(V_{i,j}^D, \theta) > C, \textit{ for some } C \in \mathbb{R} \textit{ and } C^* \in \mathbb{R},$$

and rejected otherwise.

Here, we use the same procedure as in Section 5.2 for approximating the critical points. We apply this method again to approximate the proxy constraint function about this new base Since we are now working with approximated critical-value functions, there is still a small possibility of having the bumped path and the unbumped path making different acceptance-rejection decisions. To resolve such problems, we use the unbumped path events for the bumped path, i.e. the acceptance-rejection decisions of the bumped paths are always determined by the unbumped path decisions, and this results in a bias of order $\mathcal{O}\Big(||\theta - \theta_0||^4\Big)$ as shown in chapter 5. It is clear that such a bias vanishes for the first- and second-order derivative estimates as $\theta \to \theta_0$.

6.3 Application to computing Greeks of financial products with Lévy-driven underlying

In this section, we apply OSRS to computing Greeks of derivative prices with VG and NIG underlyings. Black and Scholes (1973) demonstrated pricing European options using geometric Brownian motions, but this model is inconsistent with empirical option data. The NIG process (Barndorff-Nielsen,1998) and the VG process (Madan and Seneta,1990) are then introduced to better capture the skewness and kurtosis. The computation of exotic option prices with these models often requires Monte-Carlo simulations by expressing them as time-changed Brownian motions. Various rejection techniques are used to simulate the inverse-gaussian and gamma random variables. Conditioning on the gamma random variable for VG and the inverse-gaussian random variable for NIG, $\log(S_t)$ is distributed normally and can be generated easily.

For Lévy-driven models, Glasserman and Liu (2011) present various exact and approximate methods for simulating first-order Greeks. Their exact pathwise method requires computing parametric derivatives of a random variable, whose distribution depends on the parameter θ. Let F_X denote the cumulative distribution function of X and f the associated density. Their results are based on the relationship

$$X_\theta = F_X^{-1}(U, \theta) \text{ for } U \sim U[0,1],$$

and differentiating

$$F(F^{-1}(U, \theta), \theta) = U.$$

The resulting derivative estimates are

$$\frac{\partial}{\partial\theta}X_\theta = -(f(X_\theta))^{-1}\frac{\partial}{\partial\theta}F(X_\theta,\theta),$$

and

$$\frac{\partial^2}{\partial\theta^2}X_\theta = -(f(X_\theta))^{-1}\Big(\frac{\partial^2}{\partial\theta^2}F(X_\theta,\theta) + 2\frac{\partial}{\partial\theta}f(X_\theta,\theta)\frac{\partial}{\partial\theta}X_\theta + \frac{\partial}{\partial X}f(X_\theta,\theta)\big(\frac{\partial}{\partial\theta}X_\theta\big)^2\Big).$$

The computation of $\frac{\partial}{\partial\theta}F(X_\theta,\theta)$ and $\frac{\partial^2}{\partial\theta^2}F(X_\theta,\theta)$ in the above two equations is often challenging. For the VG process, it requires efficient algorithms for approximating digamma and trigamma functions. Thus, the performance of the resulting pathwise derivative estimator depends on the accuracy of the approximating algorithms used for these functions.

Here, for the VG process and the NIG process, we can apply the LR method to their time-changed representation. In general, to apply the method for L$\acute{e}$vy-driven models requires closed-form solutions to $\frac{\partial}{\partial\theta}\log\big(f(X,\theta)\big)$. As we shall see below, for cases where it is applicable, the method often leads to large variances.

We apply our OSRS algorithm to compute first and second-order sensitivities of both call options and barrier options, where $\log(S_t)$ follows the VG and NIG processes simulated by rejection techniques. For comparison purposes, we extend Glasserman and Liu (2011)'s exact methods to computing second-order sensitivities.

6.3.1 The NIG process

In this section, we consider the application of OSRS to compute sensitivities of a barrier option where $\log(S_t)$ follows the NIG process. We can express the evolution of NIG stock process over the observation dates $[T_{i-1}, T_i]$ under the risk neutral measure as

$$S_i = S_{i-1}\exp\Big((r+\omega+u)(T_i - T_{i-1}) + \beta I_i + \sqrt{I_i}\Phi^{-1}(V_i^o)\Big), \tag{6.7}$$

where I_i is an inverse-gaussian random variable with parameters $\rho = \delta(T_i - T_{i-1})$ and $\lambda = \sqrt{\alpha^2-\beta^2}$ and V_i^o is a standard uniform random variable, and

$$\omega = \delta\big(\sqrt{\alpha^2-(\beta+1)^2} - \sqrt{\alpha^2-\beta^2}\big).$$

Here, δ, α and β are parameters of the inverse-gaussian process (Schoutens, 2003). Michael et al (1976) introduced a transformations-with-multiple-roots method to simulate an inverse-gaussian random variable with parameters ρ and λ, the algorithm is shown below:

1. *Generate a random variate from a standard normal distribution, i.e.* $Z = N(0,1)$.

2. *Set* $\chi = Z^2$ *and then calculate*

$$Y = \frac{\rho}{\lambda} + \frac{\chi}{2\lambda^2} - \frac{\sqrt{4\rho\lambda\chi + \chi^2}}{2\lambda^2};$$

3. *Generate* $V^D = U(0,1)$.

4. *If* $V^D \le \frac{\rho}{\rho+\lambda Y}$*, return* Y *as the* $IG(\rho,\lambda)$ *random number, else return* $\frac{\rho^2}{\lambda^2 Y}$*.*

Here, the critical-value function is

$$a(\rho,\lambda,Y) = \frac{\rho}{\rho+\lambda Y},$$

which is a C^2 function.

The discounted payoff function of a down-and-out call barrier option is

$$g(S,\theta) = \exp(-rT_{N^s})(S_{T_{N^s}} - K)_+ \prod_{i=1}^{N^s-1} \mathbb{I}_{S_{T_i}>B},$$

where $T_1, T_2, ..., T_{N^s}$ are the deterministic observation dates. The payoff function is discontinuous when $S_{T_i} = B$. For benchmark purposes, we apply the LR method to compute the first and second-order Greeks.

We can compute the sensitivities of its price with respect to the parameters of interest, S_0, r, u, δ, α and β using OSRS. Define the payoff critical-value function for $i = 1, 2, ...N^s - 1$

$$a_i^g(\theta) = \Phi\left[\frac{\log\left(\frac{B}{S_{i-1}}\right) - (r+\omega+\mu)(T_i - T_{i-1}) - \beta I_i}{\sqrt{I_i}}\right],$$

and

$$a_{N^s}^g(\theta) = \Phi\left[\frac{\log\left(\frac{K}{S_{Q-1}}\right) - (r+\omega+\mu)(T_i - T_{i-1}) - \beta I_i}{\sqrt{I_i}}\right]. \tag{6.8}$$

We replace V_i^o in equation (6.7) by a change of variable function

$$U_i^g(\theta, v_i) = \begin{cases} \frac{1-a_i^g(\theta)}{1-a_i^g(\theta_0)}(V_i^o - a_i^g(\theta_0)) + a_i^g(\theta) & \text{if } V_i^o > a_i^g(\theta_0) \\ \frac{a_i^g(\theta)}{a_i^g(\theta_0)} V_i^o & \text{otherwise.} \end{cases} \tag{6.9}$$

to remove the pathwise discontinuities of both g and g'.

6.3.2 The VG process

In this section, we consider the application of OSRS to compute sensitivities of a call option where $\log(S_t)$ follows the VG process. We can express the VG stock price under the risk-neutral measure as

$$S_T = S_0 \exp\Big((r+\omega)T + \beta G_T + \sigma\sqrt{G_T}\Phi^{-1}(V^o)\Big), \tag{6.10}$$

where G_T is a gamma random variable with the shape parameter $\alpha = \frac{T}{v}$ and the scale parameter v, V^o is a standard uniform random variable, and

$$\omega = \frac{1}{v}\log(1 - \theta v - \frac{\sigma^2 v}{2}).$$

We refer the reader to Schoutens (2003) for a detailed explanation of the VG process.

Rejection techniques for simulating $\gamma(\alpha, 1)$ random variates have been studied extensively and vary according to the value of α since the qualitative properties of the distribution vary greatly with it (Tanizaki, 2008). We use 3 different algorithms accordingly. The Ahrens-Dieter method (1974) simulates gamma random variables with $\alpha \leq 1$ by an acceptance-rejection approach, and the expected number of re-calculations required per accepted sample increases as α approaches 1 from below. On the other hand, the GKM1 method (Cheng and Feast, 1980) deals with the set of gamma random variables with $\alpha > 1$ by a ratio-of-uniforms approach, an excessive number of regenerations is needed to obtain an pair of uniforms which lies within the target region. The efficacy of the OSRS algorithm depends on the performance of the rejection algorithm, so an exorbitant number of re-calculations in the rejection algorithms is clearly not desirable and both of these algorithms break down as α approaches 1. We therefore use the Tanizaki (2008) ratio of uniforms method near 1.

Computing sensitivities with respect to the shape parameter of a gamma random variable is particularly hard. Here, we present an efficient way of computing the first, and more importantly the second-order derivatives with respect to all parameters of the variance-gamma process, including the shape parameter of the gamma random variables.

Ahrens and Dieter(1974)

We use this approach for simulating gammas with $\alpha < 0.75$.

```
1. Setup: $b = \frac{\alpha+e}{e}$.
2. Repeat: generate $V_1^D = U(0,1)$ and $V_2^D = U(0,1)$ independently; set $\chi = bV_1^D$.
3. If $\chi > 1$,
   Option One: set $Y = \chi^{\frac{1}{\alpha}}$,
   if $V_2^D < \exp(-Y)$, accept,
   otherwise,
   Option Two: set $Y = -\log(\frac{b-\chi}{\alpha})$,
   if $V_2^D \leq Y^{\alpha-1}$, accept;
4. until accept;
5. return Y.
```

To apply OSRS, we perform change of measures on both V_1^D and V_2^D, since V_1^D determines the choice of two options, and V_2 makes the acceptance-rejection decision. The critical-value function for the first measure change on V_1^D is

$$a_1(b) = \frac{1}{b}.$$

The critical-value function for the second measure change on V_2^D is

$$a_2(Y, \alpha) = \begin{cases} \exp(-Y) & \text{Option One,} \\ Y^{\alpha-1} & \text{Option Two.} \end{cases}$$

We replace V_1^D and V_2^D with change of measure functions as shown in equation refmeasure AR with the above corresponding critical value functions.

The $GKM1$ approach (Cheng and Feast, 1980)

We use this ratio-of-uniforms method for simulating gammas with $\alpha \geq 3$.

1. *Setup:* $\hat{\alpha} = \alpha - 1, b = \frac{\alpha - \frac{1}{6\alpha}}{\hat{\alpha}}$, $m = \frac{2}{\hat{\alpha}}$ *and* $d = m + 2$.
2. *Repeat: generate* $V = U(0, 1)$ *and* $V^D = U(0, 1)$ *independently, set* $Y = b\frac{V^D}{V}$.
3. *If* $m_1 V - d + Y + \frac{1}{Y} \leq 0$ *accept,*
   else if $m \log(V) - \log(Y) + Y - 1 \leq 0$ *accept,*
4. *until accept,*
5. *return* $Z = \hat{\alpha}Y$.

For OSRS, the change of measure is only performed to ensure that the bumped path makes the same decision as the unbumped path in the second test. This is because the region defined by the first acceptance test is a subset of the region defined by the second acceptance test. Consider the following two cases

1. *The unbumped path passes the first test*: it would also pass the second test and accept $Y(\theta_0)$, the measure change in the second test ensures that the bumped path passes the second test accordingly and accepts $Y(\theta)$.
2. *The unbumped path fails the first test*: the measure change in the second test ensures that the bumped path accepts $Y(\theta)$ if the unbumped path accepts $Y(\theta_0)$ in the second test and rejects it if the unbumped path rejects $Y(\theta_0)$.

Regardless of the decision in the first test, a change of measure is always performed on V^D whilst V is left unchanged. Observe that given $V \in (0, 1)$,

$$\eta(V^D) = m \log(V) - \log(Y(V^D)) + Y(V^D) - 1,$$

we do not have an explicit solution for the critical values such that $\eta(V^D) = 0$. Some simple analysis on $\eta(V^D)$ shows that for a given V,

- as $V^D \to 0$, it approaches ∞;
- as $V^D \to 1$, it is always greater than or equal to zero;
- its minimum occurs when $V^D = \frac{V}{b} \in (0, 1)$, and is less than zero;
- it has strictly negative derivatives in $(0, \frac{V}{b})$ and strictly positive derivatives in $(\frac{V}{b}, 1)$.

Consequently, there are two critical values

$$a^{(1)} \in \left(0, \frac{V}{b}\right) \text{ and } a^{(2)} \in \left(\frac{V}{b}, 1\right).$$

that we need to approximate by the method in Section 2.5. We initialize the approximation

- for $a^{(1)}$ with $Z_0 = \Phi^{-1}(\frac{V}{2b})$,
- for $a^{(2)}$ with $Z_0 = \Phi^{-1}(\frac{V}{2b} + \frac{1}{2})$.

Tanizaki (2008) Approach

This ratio-of-uniforms approach caters for arbitrary shape parameters. We use it for shape parameters $0.75 \le \alpha < 3$.

1. *Setup:* $n = \frac{1}{\alpha} + \frac{\alpha - 0.4}{3.6\alpha}$, $b_1 = \alpha - \frac{1}{n}$, $b_2 = \alpha + \frac{1}{n}$, $C_1 = b_1\left(\frac{\log(b_1)-1}{2}\right)$ *and* $C_2 = b_2\left(\frac{\log(b_2)-1}{2}\right)$*;*
2. *Repeat: generate* V *and* V^D *independently from* $U(0, 1)$*, set* $w_1 = C_1 + \log(V)$*,* $w_2 = C_2 + \log(V^D)$*, and* $Y = n(b_1 w_2 - b_2 w_1)$*;*
3. *if* $Y < 0$*, reject*
   else set $X = n(w_2 - w_1)$*, and if* $\log(Y) \ge X$*, accept;*
4. *until accept, return* $\exp(X)$.

In the above algorithm

$$\log(Y) \ge X \Rightarrow Y > 0,$$

thus we only need to focus on the test of $\log(Y) \ge X$. Given V, we have

$$\eta(V^D) = \log(b_2 w_1 - b_1 w_2(V^D)) - n w_2(V^D) + n w_1 + \log(n),$$

such that

$$\eta(V^D) \ge 0 \Rightarrow \text{ accept.}$$

Like the GKM1 method, we adopt the method in Section 6.2.5 to numerically approximate the critical value functions. Some simple analysis on the function η shows

- $\eta(\exp(\frac{b_2}{b_1} w_1 - C_2)) = -\infty$, let $V_{-\infty} = \exp(\frac{b_2}{b_1} w_1 - C_2)$;

- $\eta(1) \leq 0$;
- $\eta(\exp(\frac{1}{n} + \frac{b_2}{b_1}w_1 - C_2)) \geq 0$ and it is at its maximum, let $V_{max} = \exp(\frac{1}{n} + \frac{b_2}{b_1}w_1 - C_2)$;
- it has strictly positive derivatives in $(V_{-\infty}, V_{max})$ and negative derivatives in $(V_{max}, 1)$.

Consequently, there are two critical values

$$a^{(1)} \in (V_{-\infty}, V_{max}) \text{ and } a^{(2)} \in (V_{max}, 1).$$

We initialize the approximation

- for $a^{(1)}$ with $Z_0 = \Phi^{-1}(0.5(V_{-\infty} + V_{max})))$,
- for $a^{(2)}$ with $Z_0 = \Phi^{-1}(0.5(V_{-\infty} + V_{max}))$.

The discounted payoff of a call option is

$$g(S_T, \theta) = \exp(-rT)(S_T - K)_+.$$

It is Lipschitz continuous everywhere and differentiable almost surely, the pathwise method is therefore applicable to computing the first-order Greeks. The first-order derivatives of the payoff function, however, are discontinuous when $S_T = K$, so we use a combination of the pathwise and the LR method (PWLR) to compute the second-order Greeks. For a call option with $\log(S_t)$ following the VG process, we perform a change of measure on V^o in equation (6.10) to remove the pathwise discontinuities in the first-order derivatives of the payoff function. It is replaced by the change of variable function in equation (6.9), with the following critical-value function

$$a^g(\theta) = \Phi\left[\frac{\log(\frac{K}{S_0}) - (r+\omega)(T) - \beta G_T}{\sigma\sqrt{G_T}}\right]. \tag{6.11}$$

6.3.3 Comparison of OSRS and Traditional Sensitivity Computation Approaches

For the down-and-out call option with NIG underlying, we set the parameters $S_0 = 100$, $K = 100$, $r = 0.1$, $\delta = 0.31694$, $\alpha = 28.42$, $\beta = -15, 0862$, $u = 0.05851$ and $T = 1$, the barrier level equals 80 and 12 observation dates are equally spaced. We compare the results computed by the OSRS algorithm with the gradient and Hessian computed by the exact LR method suggested by Glasserman and Liu (2011), shown in Table 6.4 and 6.5. The results computed by OSRS converge to the ones by the LR method, which demonstrates that the OSRS algorithm indeed produce unbiased estimates. OSRS, however, produces smaller standard errors for both the first and second-order sensitivities.

In Figures 6.1 and 6.2, we plot the sum of standard errors of the gradient and the Hessian computed by OSRS method and the LR method, with varying number of observation dates from 4 to 12 over a one-year horizon.

	OSRS minimum	LR minimum	OSRS maximum	LR maximum
Gradient	0.7015	5.8954	0.7835	7.47359
Hessian	15.2412	317.8014	29.2063	516.9784

Table 6.1: Summary of Results: Sum of Standard Errors by OSRS and LR

We observe that the LR method produce a larger standard errors as the number of observation dates increase. In Table 6.1, the minimums are obtained when the number of observation dates is 4 and the maximums are obtained when the number of observation dates is 12. This is because

- the number of likelihood score functions increases with the number of time steps, as more random variables I_i depend on the parameters of interest;

- we have fixed T, the inter-observation times $T_i - T_{i-1}$ reduces as the number of observation dates increases, and hence the simulated I_i is more likely to be small and the individual score function of I_i has larger variance.

OSRS, on the other hand, produces estimators for the gradient and the Hessian with relatively stable standard errors.

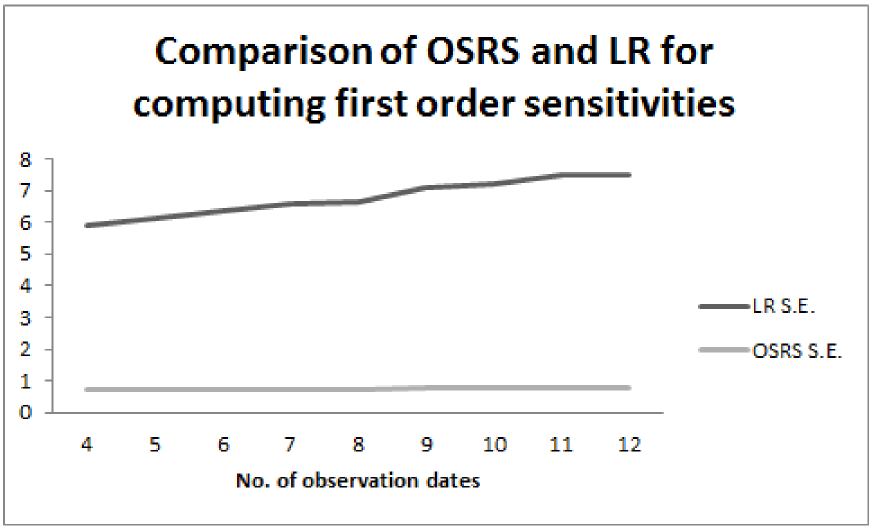

Figure 6.1: Barrier option with NIG underlying: the sum of standard errors of the first-order sensitivities computed by OSRS and LR with 20,000 path sample

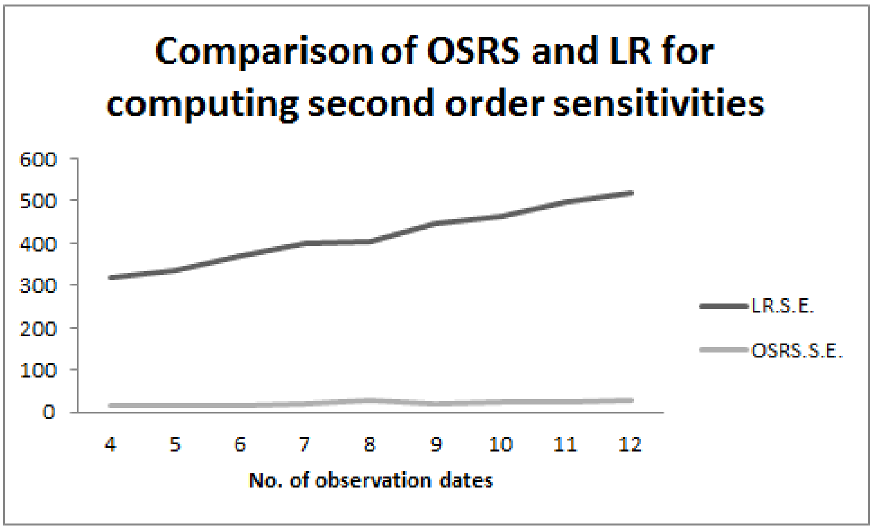

Figure 6.2: Barrier option with NIG underlying: the sum of standard errors of the Hessian computed by OSRS and LR with 20,000 path sample

For the call option with VG underlyings, we set $S = 100$, $K = 100$, $\beta = -0.15$, $r = 0.05$, $\sigma = 0.2$ and $v = 1$. We then combine their pathwise and LR approaches to compute the Hessian, and compare it to our OSRS results. We consider the case with

- a small shape parameter, i.e. $\alpha = 0.7$, that is $T = 0.7$, the gamma random variate is simulated by the Ahrens-Dieter method and the results are summarized in Table 6.6 and 6.7;
- a large shape parameter, i.e. $\alpha = 4.5$, that is $T = 4.5$, the gamma random variate is simulated by the GKM1 method and the results are summarized in Table 6.8 and 6.9;
- the shape parameter α equals one, that is $T = 1$, the gamma random variate is simulated by Tanizaki's method and the results are summarized in Table 6.10 and 6.11.

The numerical results show that, the OSRS algorithm produces unbiased estimators of first- and second-order Greeks of a call option with $\log(S_t)$ following the VG process. Observe that the gradient computed by OSRS converges to the value computed by the pathwise method and the Hessian computed by OSRS converges to the results computed by PWLR. OSRS outperforms the PWLR method in computing second-order sensitivities for the three shape parameters chosen.

In Figure 6.3, we vary the maturity date from 0.5 to 5.5 to compare the sum of standard errors of the Hessian calculated by OSRS and PWLR. For the same set of maturity dates, the maximum sum of standard errors of the Hessian calculated by OSRS is 91.8616 when $T = 5.5$ compared to 207.0194 by the PWLR method; the minimum is 17.8921 when $T = 0.5$, compared to 456.2566 by the PWLR method. The results show that as α gets smaller than 1, the sum of standard errors produced by PWLR increases dramatically. This is because extremely small G_T's in the denominator of the score function are wreaking havoc for gamma random variables with $\alpha < 1$. The OSRS method, however, does not have such problems, and produces relatively stable standard errors.

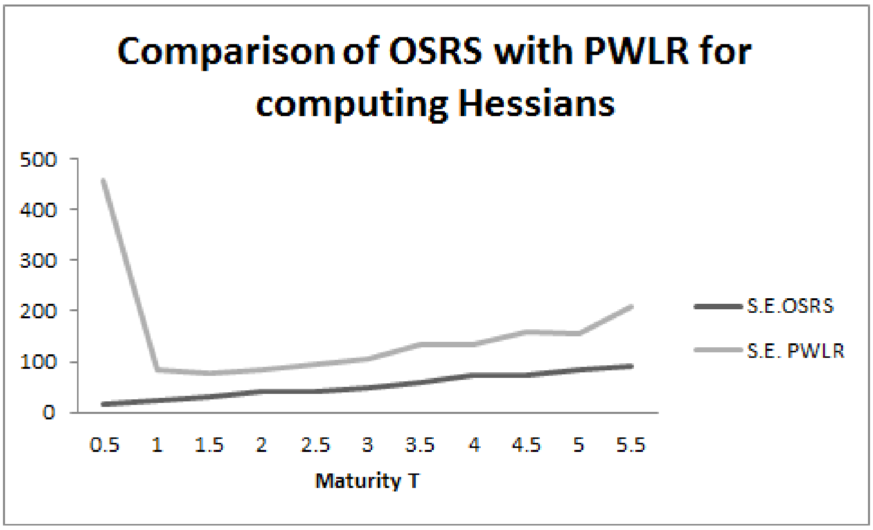

Figure 6.3: Call option with VG underlying: the sum of standard errors of the Hessian computed by OSRS and PWLR with 20,000 path sample

It is evident that OSRS does produce a smaller standard error than LR for computing first-order sensitivities. It also outperforms both the LR method and the PWLR method for computing second-order sensitivities. Although the pathwise method outperforms the OSRS method for computing first-order sensitivities, it would not be applicable when the payoff function is discontinuous.

We also demonstrate the speed of the OSRS algorithm for computing Greeks for the same numerical set-ups as the previous examples summarized in Table 6.2.

	Micheal et al(NIG)	Ahrens-Dieter method(VG)	GKM1 method(VG)	Tanizaki's method(VG)
Price time	0.215	0.119	0.122	0.134
OSRS time	1.418	0.260	0.532	0.61

Table 6.2: Time in seconds taken to compute the price and all the Greeks via OSRS for a sample of 20,000 paths

6.4 Application to queues

In this section, we apply OSRS to computing sensitivities of performance measures of queuing systems. The pathwise method, or IPA, has been introduced to computing parameter sensitivities of discrete-event systems. Glasserman (1991) provided a general formulation of IPA for a broad class of discrete-event systems, and stated sufficient conditions for these estimates to be unbiased. For computing first-order sensitivities, the pathwise performance measure is required to be a continuous function of θ and differentiable with probability one. Here we look at queues associated to discontinuous algorithms.

One interesting example is the non-homogeneous Poisson process. It has wide applications in modelling queuing systems such as the arrival process at an intensive care unit (Lewis, 1972), the transaction processing in a data base management system (Lewis and Shedler, 1979), and bulk arrivals under parallel and series configuration (Suhasini et al, 2013). We assume that there exists λ such that $\lambda(t) \leq \lambda$ for all $t \geq 0$. Given V, a simulated random uniform, obtaining an interarrival time random variate from such distributions by the inverse-transform amounts to finding x satisfying

$$\Lambda(x) = \int_{t}^{t+x} \lambda(y)dy = -\log(1 - V). \tag{6.12}$$

In general, the above problem can only be solved by an extensive amount of computational effort through techniques such as numerical quadrature (Ortega and Rheinboldt, 1970). The intuitive idea behind the thinning method (Lewis and Shedler,1979) is to first find a rate function $\lambda^*(t)$, which dominates the desired rate function $\lambda(t)$ and has a tractable solution to equation (6.12); next, solve x in equation (6.12) with the rate $\lambda^*(t)$, and reject an appropriate fraction of them so that the desired rate $\lambda(t)$ is achieved.

6.4.1 Interarrival time

Our first numerical example is computing sensitivities of the interarrival time with respect to the parameters of $\lambda(t)$. For benchmarking purposes, we have chosen the example so that the first- and second-order sensitivities can be derived from the analytical expected value. Define the random variable X_t as the arrival time of the next customer provided a customer just arrived at time t. Here we take the intensity function

$$\lambda(t) = \frac{\lambda}{1+t}, \text{ for } t \geq 0,$$

which is a continuously decreasing function bounded by λ.

To simulate X_t, we adopt the following thinning procedure

1. *generate* $V = U(0, 1)$, *set* $x_t = t - \frac{1}{\lambda}log(V)$;

2. *generate* V^D;

3. *if* $V^D \leq \frac{1}{1+x_t}$, *set* $X_t = x_t$ *and accept, else go back to step 1.*

For this simple case, the only discontinuities arise from the acceptance tests. The critical-value function is

$$a(x_t, \lambda) = \frac{1}{1 + x_t}.$$

The numerical experiment is performed by assuming $t = 2$ and $\lambda = 3$, and the comparison with the analytical solution is shown in Table 6.3.

	Analytical λ	Analytical t	OSRS λ	OSRS t	OSRS S.E. λ	OSRS S.E. t
λ	0.75	-0.25	0.725	-0.246	0.045	0.019
t	-0.25	0	-0.246	8.3E-03	0.019	0.010
First-order	-0.75	1.5	-0.742	1.495	0.024	0.017

Table 6.3: Sensitivities of the expected interarrival time with a time-dependent intensity function: OSRS results with 20,000 paths sample compared with analytical value

The result illustrates that the OSRS applied to the thinning algorithm produces unbiased estimators, as the derivative estimates calculated by OSRS converge to the analytical solutions.

6.4.2 Average time spent in an $M_t|M|1$ queue with a cyclical arrival intensity

We now consider a more realistic example, an $M_t|M|1$ queue with the arrival intensity

$$\lambda(t) = \exp(\alpha_0 + \alpha_1 t + K \sin(w_0 t + \omega)), \text{ for } t \geq 0. \tag{6.13}$$

This is a log-linear function with cyclical behaviours. This model is found by Lewis (1972) for arrivals at an intensive care unit. The computation of the corresponding $\Lambda^{-1}(x)$ is computationally demanding. Instead, we use the thinning technique with the following majoring function,

$$\lambda^*(t) = \exp(\alpha_0 + \alpha_1 t + K), \text{ for } t \geq 0,$$

for which $\Lambda^{-1}(x)$ can be calculated analytically.

The performance measure function we consider here is the average time spent by a customer in the system within a finite-time horizon T,

$$g = \mathbb{I}_{A_{N^s(\theta)} > T} \frac{1}{N^s(\theta) - 1} \sum_{i=1}^{N^s(\theta)-1} (D_i(\theta) - A_i(\theta)) \mathbb{I}_{A_i(\theta) < T}, \text{ for } N^s(\theta) \in \mathbb{N},$$

where A_i is the arrival time and D_i is the departure time of the ith customer, $N^s - 1$ is the number of customers arrived between $[0, T]$. The service time is distributed exponentially with intensity μ. To simulate such discrete-event systems, we adopt the methodology suggested by Ross (2006). The subroutine for generating A_i is

1. *let* $t = A_{i-1}$;
2. *generate* V, *obtain* x *by solving equation (4.1) with* $\lambda^*(t)$ *and let* $x_t = t + x$;
3. *generate* V^D, *if* $V^D < \exp(K(\sin(w_0 x_t + \omega) - 1))$, *accept* $A_i = x_t$;
4. *else return to step* 2.

While IPA is not applicable here for computing sensitivities, the application of the LR method is neither feasible due to the intractable score functions (Rubinstein, 1992). To resolve the problem, we remove the pathwise discontinuities caused by the rejection technique by performing measure changes on every V^D. The change of variable function is defined by the critical-value function

$$a(K, w_0, t_{i,j}, \omega) = \exp(K(\sin(w_0 t_{i,j} + \omega) - 1)),$$

where $t_{i,j}$ is the jth simulated outcome of A_i.

We also consider the irregularities of the performance measure function. First, the pathwise discontinuities of g arise from A_i passing across T. In addition, the pathwise discontinuities of g' arise from A_i passing across D_{i-1}, that when the ith customer arrives before D_{i-1}, the time spent by him in the system equals the queuing time plus the service time, otherwise, it equals the service time. Given A_{i-1} and D_{i-1}, define two critical-value functions for simulating A_i,

$$a^g_{i,1}(\theta) = 1 - \exp\Big\{ - \int_{A_{i-1}}^{T} \lambda^*(y)dy\Big\},$$

and

$$a^g_{i,2}(\theta) = 1 - \exp\Big\{ - \int_{A_{i-1}}^{D_{i-1}} \lambda^*(y)dy\Big\}.$$

If $D_{i-1} < T$, we replace V in step 2 of the simulation algorithm by the change of measure function

$$U^g_i(\theta, U) = \begin{cases} \frac{1-a^g_{i,1}(\theta)}{1-a^g_{i,1}(\theta_0)}(U - a^g_{i,1}(\theta_0)) + a^g_{i,1}(\theta) & \text{if } U > a^g_{i,1}(\theta_0) \\ \frac{a^g_{i,1}(\theta)-a^g_{i,2}(\theta)}{a^g_{i,1}(\theta_0)-a^g_{i,2}(\theta_0)}(U - a^g_{i,2}(\theta_0)) + a^g_{i,2}(\theta) & \text{if } a^g_{i,1}(\theta_0) \geq U > a^g_{i,2}(\theta_0) \\ \frac{a^g_{i,2}(\theta)}{a^g_{i,2}(\theta_0)}U & \text{otherwise.} \end{cases}$$

If $D_{i-1} \geq T$, we have the change of measure function as

$$U^g_i(\theta, U) = \begin{cases} \frac{1-a^g_{i,1}(\theta)}{1-a^g_{i,1}(\theta_0)}(U - a^g_{i,1}(\theta_0)) + a^g_{i,1}(\theta) & \text{if } U > a^g_{i,1}(\theta_0) \\ \frac{a^g_{i,1}(\theta)}{a^g_{i,1}(\theta_0)}U & \text{otherwise.} \end{cases}$$

The numerical experiments are performed by taking $\alpha_0 = 0.01$, $\alpha_1 = 0.001$, $K = 0.654$, $W = 2\pi$, $\omega = 3.519$, $\mu = 5$ and $T = 1$. We benchmark our OSRS results with 100,000 paths against the Hessian and gradient calculated by the FD method with 10,000,000 paths, shown in Table 6.12 and 6.13. We adopt the symmetric FD estimators with different bump sizes, h_θ, for each parameter, see Glasserman(2004) for the detailed explanation. Here, we set $h_{\alpha_0} = 0.001$, $h_{\alpha_1} = 0.0009$, $h_K = 0.01$, $h_W = 0.1$, $h_\omega = 0.1$ and $h_\mu = 0.1$ for the FD estimators. The FD method does not produce as good results as OSRS even with 100 times more paths, this is expected as the performance measure function is discontinuous in the parameters of interest and the bump sizes are very small. We refer the reader to detailed explanations in Glasserman (2004). OSRS produces unbiased estimators with tolerable standard errors for a reasonable number of paths. With the same set of parameter inputs as above and a sample size of 20,000 paths, the time taken for computing the expected time spent in the system is 1.298 seconds, and the corresponding time for computing all the sensitivities is 2.258 seconds by the OSRS method.

6.5 Conclusion

We have introduced a simulation algorithm for computing unbiased estimates of first- and second-order derivative for performance measures simulated by rejection techniques. The method is to perform a measure change at each acceptance test on the decision uniform random variable which ensures that the points of discontinuities do not move with the parameters of interest up to the third order; the change of variable function is also chosen to be optimal in terms of minimizing the variance of the likelihood ratio terms. We applied OSRS to computing sensitivities of options prices with L$\acute{e}$vy-driven underlyings and the average time spent by a customer in an $M_t|M|1$ queue to demonstrate the superiority of OSRS comparing to the traditional methods of estimating derivatives. We saw that is very effective in both cases.

We would like to emphasize breadth of applicability of the new algorithm: it can be applied to situations in which the performance measure function is highly discontinuous, the underlying state variables have intractable distribution functions and none of the traditional methods are feasible.

Chapter Appendix One: Proof of Theorem 1

In this section, we prove that the pathwise estimators of first- and second-order derivatives obtained via the OSRS method are indeed unbiased.

Proof. We consider the following rejection sampling algorithm:

$$S(\theta, V) = \sum_{j=1}^{\infty} s(\theta, v_j)\mathbb{I}_{v_j^D > a(\theta)} \prod_{k=1}^{j-1} \mathbb{I}_{v_k^D \le a(\theta)},$$

where the empty product is 1. Only one of the terms in this infinite sum is non-zero, thus

$$S(\theta, V) = s(\theta, v_j)\mathbb{I}_{v_j^D > a(\theta)} \prod_{k=1}^{j-1} \mathbb{I}_{v_k^D \le a(\theta)} \text{ for some } j \in \mathbb{N}.$$

Given the sequence of standard uniforms for generating the first j outcomes, $\{v_1, ..., v_k, ...v_j\}$, we express its expectation of one outcome as

$$G_j(\theta) = \int_{a_j(\theta)}^{1} ... \int_{0}^{a_1(\theta)} s(\theta, v_j) dv_1^D ... dv_j^D.$$

Assume the sequence of measure change functions, $U(\theta, v_k^D)$, satisfies condition (2), so that

$$\hat{G}_j(\theta) = \int_{U^{-1}\left(\theta, a_j(\theta)\right)}^{1} ... \int_{0}^{U^{-1}\left(\theta, a_1(\theta)\right)} s(\theta, v_j) \prod_{k=1}^{j} \frac{\partial U(\theta, v_k^D)}{\partial v_k^D} dv_1^D ... dv_j^D$$

exist. In addition, condition (3) ensures that given a fixed θ,

$$U^{-1}\big(\theta, a_k(\theta)\big) = a_k(\theta_0), \text{ for all } k = 1, 2, ...j,$$

and

$$\hat{G}_j(\theta) = \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} s(\theta, v_0) \prod_{k=1}^{j} \frac{\partial U(\theta, v_k^D)}{\partial v_k^D} dv_1^D ... dv_j^D.$$

Now, we show that under conditions (4) to (6),

- $G_j(\theta_0 = \hat{G}_j(\theta_0)$,
- $G_j'(\theta_0) = \hat{G}_j'(\theta_0$,
- $G_j''(\theta_0) = \hat{G}_j''(\theta_0)$.

The limit value of the original algorithm and its first two derivatives are shown below.

$$G_j(\theta_0) = s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0),$$

$$\begin{aligned} G_j'(\theta_0) =& \frac{\partial s(\theta_0, v_j)}{\partial \theta}\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) - s(\theta_0, v_j) \frac{\partial a_j(\theta_0)}{\partial \theta} \prod_{k=1}^{j-1} a_k(\theta_0) \\ &+ s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta}, \end{aligned}$$

and

$$\begin{aligned} G_j''(\theta_0) =& \frac{\partial^2 s(\theta_0, v_j)}{\partial \theta^2}\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) - 2\frac{\partial s(\theta_0, v_j)}{\partial \theta} \frac{\partial a_j(\theta_0)}{\partial \theta} \prod_{k=1}^{j-1} a_k(\theta_0) \\ &+ 2\frac{\partial s(\theta_0, v_j)}{\partial \theta}\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} - s(\theta_0, v_j) \frac{\partial^2 a_j(\theta_0)}{\partial \theta^2} \prod_{k=1}^{j-1} a_k(\theta_0) \\ &- 2s(\theta_0, v_j) \frac{\partial a_j(\theta_0)}{\partial \theta} \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} + s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial^2 \log(a\_k(\theta_0))}{\partial \theta^2} \\ &+ 2s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} \sum_{m=1}^{k-1} \frac{\partial \log(a_m(\theta_0))}{\partial \theta}. \end{aligned}$$

First, we have the limit value of the OSRS algorithm

$$\hat{G}_j(\theta_0) = s(\theta_0, v_j) \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} \prod_{k=1}^{j} \frac{\partial U(\theta_0, v_k^D)}{\partial v_k^D} dv_1^D ... dv_j^D.$$

condition (4) implies that $\frac{\partial U(\theta_0, v_k^D)}{\partial v_0^D} = 1$, thus

$$\begin{aligned}\hat{G}_j(\theta_0) &= s(\theta_0, v_j)\int_{a_j(\theta_0)}^{1} ... \int_0^{a_1(\theta_0)} 1 dv_1^D ... dv_j^D \\ &= s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big)\prod_{k=1}^{j-1} a_k(\theta_0)\end{aligned}$$

which agrees with the original $G_j(\theta_0)$.

Its first derivative is

$$\begin{aligned}\hat{G})'_j(\theta_0) = &\frac{\partial s(\theta_0, v_j)}{\partial \theta}\int_{a_j(\theta_0)}^{1} ... \int_0^{a_1(\theta_0)} \prod_{k=1}^{j} \frac{\partial U(\theta_0, v_k^D)}{\partial v_k^D} dv_1^D ... dv_j^D \\ &+ s(\theta_0, v_j)\int_{a_j(\theta_0)}^{1} ... \int_0^{a_1(\theta_0)} \prod_{k=1}^{j} \frac{\partial U(\theta_0, v_k^D)}{\partial v_k^D} \sum_{k=1}^{j} \frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta \partial v_k^D} dv_1^D ... dv_j^D\end{aligned}$$

Due to condition (4), it reduces to

$$\begin{aligned}\hat{G}'(\theta_0, v_j) =& \frac{\partial s(\theta_0, v_j)}{\partial \theta}\int_{a_j(\theta_0)}^{1} ... \int_0^{a_1(\theta_0)} 1 dv_1^D ... dv_j^D + s(\theta_0, v_j)\sum_{k=1}^{j}\int_{a_j(\theta_0)}^{1} ... \int_0^{a_1(\theta_0)} \frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta \partial v_k^D} dv_1^D ... dv_j^D \\ =& \frac{\partial s(\theta_0, v_j)}{\partial \theta}\Big(1 - a_j(\theta_0)\Big)\prod_{k=1}^{j-1} a_k(\theta_0) + s(\theta_0, v_j)\sum_{k=1}^{j-1} a_k(\theta_0)\Big[\frac{\partial U(\theta_0, v_j^D)}{\partial \theta}\Big]_{a_j(\theta_0)}^{1} \\ &+ s(\theta_0, v_j)\Big(1 - a_j(\theta_0)\Big)\sum_{k=1}^{j-1}\prod_{m \neq k}^{j-1} a_m(\theta_0)\Big[\frac{\partial U(\theta_0, v_k^D)}{\partial \theta}\Big]_0^{a_k(\theta_0)}.\end{aligned}$$

Condition (5) implies that

$$\Big[\frac{\partial U(\theta_0, v_j^D)}{\partial \theta}\Big]_{a\big(s(\theta_0, v_j)), \theta_0\big)}^{1} = -\frac{\partial a_j(\theta_0)}{\partial \theta},$$

and

$$\Big[\frac{\partial U(\theta_0, v_k^D)}{\partial \theta}\Big]_0^{a_k(\theta_0)} = \frac{\partial a_k(\theta_0)}{\partial \theta},$$

so $\hat{G}'_j(\theta_0) = G'_j(\theta_0)$.

With condition (4), its second derivative becomes

$$
\begin{aligned}
\hat{G}_j''(\theta_0) =& \frac{\partial^2 s(\theta_0, v_j)}{\partial \theta^2} \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} 1 dv_1^D ... dv_j^D + 2\frac{\partial s(\theta_0, v_j)}{\partial \theta} \sum_{k=1}^{j} \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} \frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta \partial v_k^D} dv_1^D ... dv_j^D \\
&+ s(\theta_0, v_j) \sum_{k=1}^{j} \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} \frac{\partial^3 U(\theta_0, v_k^D)}{\partial \theta^2 \partial v_k^D} dv_1^D ... dv_j^D \\
&+ 2s(\theta_0, v_j) \sum_{k=1}^{j} \sum_{m=1}^{k-1} \int_{a_j(\theta_0)}^{1} ... \int_{0}^{a_1(\theta_0)} \frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta \partial v_k^D} \frac{\partial^2 U(\theta_0, v_m^D)}{\partial \theta \partial v_m^D} dv_1^D ... dv_j^D \\
=& \frac{\partial^2 s(\theta_0, v_j)}{\partial \theta^2} \Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) - 2\frac{\partial s(\theta_0, v_j)}{\partial \theta} \frac{\partial a_j(\theta_0)}{\partial \theta} \prod_{k=1}^{j-1} a_k(\theta_0) \\
&+ 2\frac{\partial s(\theta_0, v_j)}{\partial \theta} \Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} - s(\theta_0, v_j) \prod_{k=1}^{j-1} a_k(\theta_0) \Big[\frac{\partial^2 U(\theta_0, v_j^D)}{\partial \theta^2}\Big]_{a_j(\theta_0)}^{1} \\
&- 2s(\theta_0, v_j) \frac{\partial a_j(\theta_0)}{\partial \theta} \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} \\
&+ s(\theta_0, v_j) \Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \Big[\frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta^2}\Big]_{0}^{a_k(\theta_0)} \\
&+ 2s(\theta_0, v_j) \Big(1 - a_j(\theta_0)\Big) \prod_{k=1}^{j-1} a_k(\theta_0) \sum_{k=1}^{j-1} \frac{\partial \log(a_k(\theta_0))}{\partial \theta} \sum_{m=1}^{k-1} \frac{\partial \log(a_m(\theta_0))}{\partial \theta}.
\end{aligned}
$$

Condition (6) implies that

$$
\Big[\frac{\partial^2 U(\theta_0, v_j^D)}{\partial \theta^2}\Big]_{a_j(\theta_0)}^{1} = -\frac{\partial^2 a_j(\theta_0)}{\partial \theta^2},
$$

and

$$
\Big[\frac{\partial^2 U(\theta_0, v_k^D)}{\partial \theta^2}\Big]_{0}^{a_k(\theta_0)} = \frac{\partial^2 a_k(\theta_0)}{\partial \theta^2},
$$

so $\hat{G}_j''(\theta_0) = G_j''(\theta_0)$.

We have forced the limits to not move with θ up to third order.

Furthermore, condition (1) ensures that the new integrand,

$$
\begin{aligned}
\hat{S}(\theta, V) =& s(\theta, v_j) \prod_{k=1}^{j} \frac{\partial U(\theta, v_k^D)}{\partial v_k^D} \\
=& s(\theta, v_N(\theta_0)) W(\theta, V),
\end{aligned}
$$

is a C^2 function[1] of θ, so we can apply the pathwise method to compute first- and second-order

[1] The construction of the measure changes ensures that the resulted order of differentiability with respect to θ is consistent with $s(\theta, v_0)$ and $a(s, \theta)$.

derivative estimates. The first-order derivative estimate is

$$\frac{\partial \hat{S}(\theta,V)}{\partial\theta}=\frac{\partial s(\theta,v_N(\theta_0))}{\partial\theta}W(\theta,V)+s(\theta,v_N(\theta_0))\frac{\partial W(\theta,V)}{\partial\theta}.$$

Given condition (4), we have

$$\frac{\partial \hat{S}(\theta_0,V)}{\partial\theta}=\frac{\partial s(\theta_0,v_N(\theta_0))}{\partial\theta}+s(\theta_0,v_N(\theta_0))\frac{\partial W(\theta_0,V)}{\partial\theta}.$$

The second-order derivative estimate is

$$\frac{\partial^2 \hat{S}(\theta,V)}{\partial\theta^2}=\frac{\partial^2 s(\theta,v_N(\theta_0))}{\partial\theta^2}W(\theta,V)+2\frac{\partial s(\theta,v_N(\theta_0))}{\partial\theta}\frac{\partial W(\theta,V)}{\partial\theta}+s(\theta,v_N(\theta_0))\frac{\partial^2 W(\theta,V)}{\partial\theta^2}.$$

Given condition (4), we have

$$\frac{\partial^2 \hat{S}(\theta_0,V)}{\partial\theta^2}=\frac{\partial^2 s(\theta_0,v_N(\theta_0))}{\partial\theta^2}+2\frac{\partial s(\theta_0,v_N(\theta_0))}{\partial\theta}\frac{\partial W(\theta_0,V)}{\partial\theta}+s(\theta_0,v_N(\theta_0))\frac{\partial^2 W(\theta_0,V)}{\partial\theta^2}.$$

□

Chapter Appendix Two: Tables of Numerical Results

OSRS vs LR	S	r	u	δ	α	β
S	0.012 vs 0.027	1.172 vs 1.248	2.124 vs 2.195	-0.279 vs -0.28	6.06E-03 vs 4.46E-04	6.24E-03 vs 5.86E-03
r	1.172 vs 1.248	38.411 vs 40.184	117.052 vs 118.829	-32.22 vs -27.779	0.727 vs 0.732	0.748 vs 0.748
u	2.124 vs 2.195	117.052 vs 118.829	212.209 vs 214	-28.108 vs -22.845	0.615 vs 0.608	0.63 vs 0.628
δ	-0.279 vs -0.28	-32.22 vs -27.779	-28.108 vs -22.845	1.041 vs 16.802	-0.324 vs -0.321	-0.332 vs -0.3
α	6.09E03 vs 4.46E-03	0.727 vs 0.732	0.615 vs 0.608	-0.324 vs -0.321	0.018 vs 0.017	0.019 vs 0.018
β	6.24E-03 vs 5.86E-03	0.748 vs 0.748	0.630 vs 0.628	-0.332 vs -0.300	0.019 vs 0.018	0.021 vs 0.023
First Order	0.952 vs 0.947	78.641 vs 78.645	95.156 vs 95.171	4.113 vs 4.934	-0.112 vs -0.124	-0.118 vs -0.12

Table 6.4: Barrier Option with NIG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by LR using a 20,000 paths sample

OSRS vs LR	S	r	u	δ	α	β
S	1.1E-04 vs 0.023	0.014 vs 0.405	0.014 vs 0.417	0.005 vs 0.501	1.73E-04 vs 1.146E-03	1.18E-04 vs 5.34E-03
r	0.014 vs 0.405	1.468 vs 16.468	1.473 vs 16.911	0.383 vs 15.929	0.016 vs 0.151	0.014 vs 0.204
u	0.014 vs 0.417	1.473 vs 16.911	1.478 vs 17.377	0.442 vs 16.321	0.017 vs 0.155	0.015 vs 0.209
δ	4.54E-03 vs 0.501	0.383 vs 15.929	0.442 vs 16.321	0.288 vs 28.309	5.58E-03 vs 0.186	5.74E-03 vs 0.217
α	1.15E-04 vs 4.1E-03	0.016 vs 0.151	0.017 vs 0.155	5.58E-03 vs 0.186	3.1E-04 vs 2.28E-03	3.07E-04 vs 2.282E-03
β	9.39E-04 vs 5.34E-03	0.014 vs 0.204	0.015 vs 0.209	5.74E-03 vs 0.217	3.07E-04 vs 2.82E-03	2.25E-04 vs 3.11E-03
First Order	9.39E-04 vs 0.023	0.075 vs 0.705	0.092 vs 0.72	0.077 vs 0.923	2.25E-03 vs 7.73E-03	2.11E-03 vs 9.29E-03

Table 6.5: Barrier Option with NIG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by LR using a 20,000 paths sample

OSRS vs PWLR	S	r	σ	β	v
S	0.017 vs 0.018	1.206 vs 1.262	-0.251 vs -0.251	-0.129 vs -0.190	0.073 vs 0.075
r	1.206 vs 1.262	52.981 vs 56.435	-30.789 vs -30.246	1.004 vs -3.229	4.93 vs 5.053
σ	-0.251 vs -0.251	-30.789 vs -30.246	34.242 vs 32.052	51.412 vs 51.737	-8.332 vs -8.035
β	-0.129 vs -0.199	1.004 vs -3.229	51.412 vs 51.737	41.812 vs 44.064	-8.398 vs -8.057
v	0.073 vs 0.075	4.93 vs 5.053	-8.332 vs -8.035	-8.398 vs -8.057	-0.169 vs -0.358
First Order	0.731 vs 0.724	44.935 vs 44.516	18.835 vs 18.343	-14.352 vs -14.828	0.288 vs 0.294

Table 6.6: Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 0.7$

OSRS vs PWLR	S	r	σ	β	v
S	1.902E-04 vs 4.02E-03	0.013 vs 0.281	0.016 vs 0.066	0.018 vs 0.248	7.22E-03 vs 6.12E-03
r	0.013 vs 0.281	0.86 vs 17.431	0.875 vs 3.852	0.999 vs 15.358	0.434 vs 0.379
σ	0.016 vs 0.066	0.875 vs 3.852	2.305 vs 7.422	2.502 vs 4.909	0.713 vs 0.36
β	0.018 vs 0.248	0.999 vs 15.358	2.502 vs 4.909	3.136 vs 15.574	0.538 vs 0.453
v	7.22E-03 vs 6.12E-03	0.434 vs 0.379	0.713 vs 0.36	0.538 vs 0.453	0.322 vs 0.304
First Order	5.00E-03 vs 3.73E-03	0.063 vs 0.045	0.147 vs 0.167	0.155 vs 0.132	0.054 vs 0.025

Table 6.7: Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 0.7$

OSRS vs PWLR	S	r	σ	β	v
S	5.586E-03 vs 5.01E-03	2.457 vs 2.529	-0.235 vs -0.238	0.021vs 0.022	0.023 vs 0.023
r	2.457 vs 2.529	104.804 vs 127.357	-303.737 vs -314.878	148.647 vs 152.506	0.492 vs 0.536
σ	-0.235 vs -0.238	-303.737 vs -314.878	100.452 vs 129.214	116.11 vs 128.319	-14.21 vs -15.029
β	0.021 vs 0.022	148.647 vs 152.506	116.11 vs 128.319	114.117 vs 108.343	-22.331 vs -22.727
v	0.023 vs 0.023	0.492 vs 0.536	-14.21 vs -15.029	-22.331 vs -22.727	-1.422 vs -1.695
First Order	0.789 vs 0.803	222.432 vs 225.786	43.952 vs 45.553	-32.273 vs -32.605	2.038 vs 1.915

Table 6.8: Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 4.5$

OSRS vs PWLR	S	r	σ	β	v
S	5.96E-05 vs 2.23E-04	0.027 vs 0.104	0.024 vs 0.077	0.028 vs 0.051	6.54E-03 vs 6.25E-03
r	0.027 vs 0.104	10.044 vs 22.208	6.226 vs 11.84	7.324 vs 10.687	1.813 vs 1.612
σ	0.024 vs 0.077	6.226 vs 11.84	6.679 vs 38.673	7.595 vs 14.788	0.944 vs 1.07
β	0.028 vs 0.051	7.324 vs 10.687	7.595 vs 14.788	10.847 vs 14.228	1.126 vs 1.763
v	6.54E-03 vs 6.25E-03	1.813 vs 1.612	0.944 vs 1.07	1.126 vs 1.763	0.535 vs 0.638
First Order	6.60E-03 vs 4.84E-03	1.807 vs 1.228	1.277 vs 1.562	1.451 vs 1.247	0.261 vs 0.063

Table 6.9: Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 4.5$

OSRS vs PWLR	S	r	σ	β	v
S	0.014 vs 0.013	1.423 vs 1.344	-0.246 vs -0.193	-0.097 vs 2.77E-03	0.056 vs 0.065
r	1.423 vs 1.344	80.77 vs 73.953	-47.8 vs -43.501	7.475 vs 15.473	5.04 vs 5.85
σ	-0.246 vs -0.193	-47.8 vs -43.501	43.294 vs 48.746	63.428 vs 69.915	-8.307 vs -8.622
β	-0.097 vs 2.77E-03	7.475 vs 15.473	63.428 vs 69.915	52.886 vs 52.869	-9.173 vs -10.089
v	0.056 vs 0.065	5.04 vs 5.85	-8.307 vs -8.622	-9.173 vs -10.089	-0.629 vs -0.443
First Order	0.728 vs 0.728	61.553 vs 61.487	23.16 vs 23.851	-17.186 vs -16.9	0.526 vs 0.513

Table 6.10: Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 1$

OSRS vs PWLR	S	r	σ	β	v
S	1.52E-04 vs 1.21E-03	0.015 vs 0.121	0.017 vs 0.053	0.019 vs 0.103	3.5E-03 vs 5.87E-03
r	0.015 vs 0.121	1.371 vs 10.143	1.286 vs 3.841	1.500 vs 8.667	0.296 vs 0.496
σ	0.017 vs 0.053	1.286 vs 3.841	2.854 vs 10.288	3.153 vs 5.668	0.323 vs 0.407
β	0.019 vs 0.103	1.500 vs 8.667	3.153 vs 5.668	3.992 vs 9.613	0.307 vs 0.534
v	3.5E-03 vs 5.87E-03	0.296 vs 0.496	0.323 vs 0.407	0.307 vs 0.534	0.143 vs 0.200
First Order	5.22E-03 vs 3.88E-03	0.433 vs 0.322	0.592 vs 0.600	0.541 vs 0.462	0.057 vs 0.028

Table 6.11: Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 1$

OSRS vs FD	α_0	α_1	K	w_0	ω	μ
α_0	-0.402 vs -0.906	-1.284 vs -23.821	-0.368 vs -0.447	0.123 vs 0.028	0.053 vs 0.044	-0.024 vs -0.036
α_1	-1.284 vs -23.821	-3.693 vs -94.7	-1.173 vs -1.778	0.288 vs 1.246	0.078 vs 0.375	-7.49E-03 vs -0.566
K	-0.368 vs -0.447	-1.173 vs -1.778	-0.269 vs -8.241	0.105 vs 0.044	0.047 vs 0.056	-7.72E-03 vs -0.146
w_0	0.123 vs 0.028	0.288 vs 1.246	0.105 vs 0.044	-0.038 vs -0.061	-0.021 vs -6.59E-03	9.36E-04 vs 1.32E-03
ω	0.053 vs 0.044	0.078 vs 0.375	0.047 vs 0.056	-0.021 vs -6.59E-03	-8.06E-03 vs 0.019	-7.12E-04 vs -1.14E-03
μ	-0.024 vs -0.036	-7.49E-03 vs -0.566	-7.72E-03 vs -0.146	9.36E-04 vs 1.32E-03	-7.12E-04 vs -1.14E-03	0.013 vs 0.037
First Order	0.091 vs 0.103	0.025 vs 0.043	0.023 vs 0.043	-3.86E-03 vs -7.25E-03	2.57E-03 vs 1.60E-03	-0.033 vs -0.035

Table 6.12: Average Waiting time $M_t|M|1$ queue: Mean of the Hessian and first order sensitivities calculated by the OSRS method using a 100,000 path sample and by the FD method using a 10,000,000 path sample

OSRS vs FD	α_0	α_1	K	w_0	ω	μ
α_0	0.293 vs 4.893	0.983 vs 11.101	0.291 vs 0.998	0.061 vs 0.099	0.018 vs 0.098	1.47E-03 vs 0.098
α_1	0.983 vs 11.101	3.844 vs 603.503	0.968 vs 11.084	0.192 vs 1.109	0.041 vs 1.108	3.37E-03 vs 1.109
K	0.291 vs 0.998	0.968 vs 11.084	0.292 vs 4.895	0.060 vs 0.099	0.018 vs 0.099	1.53E-03 vs 0.099
w_0	0.061 vs 0.010	0.192 vs 1.109	0.060 vs 0.099	0.014 vs 0.049	6.99E-03 vs 0.010	3.782E-04 vs 9.98E-03
ω	0.018 vs 0.010	0.041 vs 1.108	0.018 vs 0.099	6.99E-03 vs 0.010	8.29E-03 vs 0.049	2.603E-04 vs 9.99E-03
μ	1.47E-03 vs 0.121	3.37E-03 vs 1.109	1.53E-03 vs 0.099	3.782E-04 vs 9.98E-03	2.603E-04 vs 9.99E-03	5.575E-05 vs 0.049
First Order	6.72E-03 vs 0.014	0.016 vs 0.157	7.04E-03 vs 0.014	1.71E-03 vs 1.41E-03	1.14E-03 vs 1.41E-03	1.39E-04 vs 1.42E-03

Table 6.13: Average Waiting time $M_t|M|1$ queue: Standard Errors of the Hessian and first order sensitivities calculated by the OSRS method using a 100,000 path sample and by the FD method using a 10,000,000 path sample

Chapter 7

The Efficient Computation and Sensitivity Analysis of Finite-time ruin Probabilities and the Estimation of Risk-Based Regulatory Capital

7.1 Introduction

The standard theoretical approach underlying insurance regulations originated in risk theory. The aim of prudential regulations is to ensure that the probability of ruin for insurance portfolios is below some given "acceptable" level. To ensure this, regulators set a mandatory minimum amount of capital that may be used by the insurance company as a buffer. The risk theory community aims to utilize sophisticated stochastic models to estimate

- the probability of ruin within a finite-time horizon given the current level of capital,
- the sensitivities of the current ruin probability with respect to the underlying risk factors,
- the density of the time to ruin given the current portfolio condition,
- the minimum regulatory capital needed to obtain a survival probability of at least 99.9% (for example) over a given possible long time horizon,
- the sensitivities of the regulatory capital with respect to the underlying risk factors.

Traditionally, the risk theory community has focused on deriving analytical solutions of ruin probabilities with restrictions to some specific models. After specifying the theoretical basis, the time to ruin,

$$\tau(u) = \inf\{t > 0 : R_t < 0\},$$

is studied. In particular, the ruin probabilities are defined in terms of $\tau(u)$,

$$\psi(u) = \mathbb{P}(\tau(u) < \infty),$$

and

$$\psi(u,t) = \mathbb{P}(\tau(u) \leq t). \tag{7.1}$$

One can use the Pollaczek-Khinchin formula (Pollaczek,1930 and Khinchin,1967) for calculating the ultimate ruin probabilities. However, it is only tractable when the Laplace transform of the ultimate ruin probability is a rational function. From a practical point of view, the finite-time ruin probability, $\psi(u,t)$, may perhaps be regarded as more interesting than the infinite-time one. The finite-time, t, is related to the planning horizon of the company, and typically long term, for example 10 years. In recent years, there has been a surge of research into methods for computing finite-time ruin probabilities, especially for the classical model:

- Picard and Lefévre (1997) derived the finite-time ruin probabilities with discrete claim size distributions.

- De Vylder (1999) provided numerical approximations of finite-time ruin probabilities for the continuous case by the Picard-Lefévre Formula.

- Dickson and Willmot (2005) derived an expression for the density of the time to ruin in the classical risk model by inverting its Laplace transform. They have shown that finite-time ruin probabilities can be calculated when the individual claim size follows a mixed Erlang distribution.

- Dickson (2007) derived the density function of aggregate claims for joint density functions involving the time to ruin, the deficit at ruin and the surplus prior to ruin.

In general, ruin probabilities can only be calculated analytically for some special light-tailed claim size distributions under the classical risk model and the Sparre Andersen model. These analytical solutions provide helpful insights for the nature of insurance risks. However, due to the inherent complexity of the insurance business, these models are insufficient to explain the underlying dynamics. While the analytical solutions of ruin probabilities under more realistic models are yet to be discovered, we address the above problems by efficient simulation algorithms, that can be used widely for practical purposes.

These analytical formulae can, however, require hours of computation. For more realistic models, there are no analytical results found in the literature. Albeit, one can always use Monte-Carlo simulations to estimate these ruin probabilities.

In the first part of our paper, we develop an efficient simulation algorithm for estimating finite-time ruin probabilities. The natural choice here is the direct simulation algorithm, which provides an unbiased pathwise estimator,

$$g = \mathbb{I}_{\{\tau(u)\leq t\}}. \tag{7.2}$$

This estimator, however, does not provide sensible answers in practical situations, because solvency regulations require insurance companies to hold capital so that ruin in a finite-time horizon is a rare event. For a 99.9$\%$ sufficiency level, one can only expect to observe non-zero pathwise estimates, i.e. $g = 1$, in just five paths of a 5,000 paths sample. To provide reliable estimates via direct simulation would be a time-consuming exercise. Stratified sampling is a method widely used as a variance reduction technique by the Monte-Carlo community. As pointed out by Glasserman et al (1999), stratified sampling can be viewed as a special case of importance sampling in that measure changes are performed on the standard uniforms for simulating the random variables.

Here, we introduce stratified sampling to the field of ruin theory and provide better estimates for finite-time ruin probabilities when they are small. The method is similar to the approach of Joshi and Kainth (2003) for studying credit derivatives. In each claim time, we only require the procedure for simulating X_i to be monotonic in its first standard uniform. Then, given $N_t = n$, for $n = 1, 2, \dots$, we force the probability of ruin at the ith claim to be $\frac{1}{n+1-i}$ provided that ruin has not occurred through the first $i - 1$ claims, by performing a measure change on the first standard uniform in the simulation algorithm for each X_i until ruin occurs. For paths with one or more claims, the probability of ruin becomes certain. The new pathwise estimator is

$$g = \mathbb{I}_{N_t>0} \prod_{i=1}^{n_r} W_i, \tag{7.3}$$

where n_r is the number of the claim at which the surplus level drops below zero and W_i is the likelihood ratio weight resulting from the measure change at the ith claim. The mild assumption and the simple procedure allow the algorithm to have wide applications, which provide an easy and efficient way to compute ruin probabilities. The numerical results show that, the stratified sampling algorithm produces better estimates of finite-time ruin probabilities when they are small.

The second practical problem we address in this paper is to compute derivative estimates of finite-time ruin probabilities with respect to both distributional and structural parameters. Sensitivity analysis is useful for discovering which risk factors are important and which ones are not. Further, the derivative of $\psi(u, t)$ with respect to t gives the density of the time to ruin, which is one important research focus of the risk theory community. Our pathwise estimators of finite-time ruin probabilities clearly fail to satisfy the $\hat{C}^2$ conditions of the pathwise method.

While computing sensitivities has been studied extensively in the Monte-Carlo literature, there has been little progress made for estimating derivatives of ruin probabilities due to practical difficulties of the problem. Asmussen and Rubinstein (1999) estimated the ultimate ruin probability

$\psi(u)$ and its sensitivities, under various distributions of the arrival processes and the claim sizes by simulation. For distributional parameters, they used the LR method; for the structural parameters, they used a combination of the LR and the push-out method. The idea of the push-out method is to rewrite the model so that the parameter of interest appears in the density, and the LR method is then applicable. Privault and Wei (2004) introduced a Malliavin calculus approach to compute sensitivities of the ultimate ruin probability under the classical risk model with a constant interest rate and an arbitrary claim size distribution. Another approach by Vazquez-Abad (2000) is rare perturbation analysis (RPA), which assumes that there exists a σ-field such that the interchange of differentiation and integration is valid. However, for the sensitivities of the finite-time ruin probabilities, only the Malliavin calculus approach has been applied to compute $\frac{\partial \psi(u,t)}{\partial u}$ in the classical model, where the claim sizes were stricted to fixed, exponential and Pareto random variables (Loisel and Privault, 2009).

In this paper, we introduce a new method to compute first- and second-order sensitivities of the finite-time ruin probabilities. Our techniques work under similar assumptions to those for stratified sampling. The pathwise estimator of ruin by direct simulation is shown in equation (7.2), it is clearly a discontinuous function of both the distributional and the structural parameters. To remove the pathwise discontinuities, we perform measure changes to ensure

- the bumped paths and the unbumped paths have the same number of claims, that is $N_t(\eta) = N_t(\eta_0)$, where $N_t(\eta)$ is the number of claims up to time t given the distributional parameters η,
- the surplus level $R_t(\eta, \phi) < 0$ if and only if $R_t(\eta_0, \phi_0) < 0$, where $R_t(\eta, \phi)$ is the surplus level at time t given the distributional and the structural parameters .

We shall call the resulting simulation algorithm, the sensitivity of finite-time ruin by direct simulation (SFRDS). The pathwise estimator by the stratified sampling algorithm as shown in equation (7.3) is $\hat{C}^2$ as a function of the structural parameters as well as the parameters of the claim size distribution. This is because the likelihood ratio weights, W_i, are continuously differentiable in these parameters. However, the function is still discontinuous with respect to the parameters of the inter-claim time distribution. That is, a small bump in them may change the value of the indicator function as well as the value of n_r. We perform changes of measure to ensure such pathwise discontinuities are removed. We shall call the resulting simulation algorithm, the sensitivity of the finite-time ruin by stratified sampling (SFRSS). Numerical experiments are performed to show that the sensitivities computed by the two methods agree. However, when finite-time ruin is rare, the SFRSS method outperforms.

After obtaining efficient algorithms for computing the finite-time ruin probabilities and their sensitivities, we can then address the problem of estimating the regulatory capital. Since the stratified sampling algorithm and the SFRSS method produce better estimates when finite-time ruin is rare, they are used to approximate the critical initial surplus level, i.e. the regulatory capital u^* such that the finite-time ruin probability calculated is at the solvency level α,

$$\psi(u^*, t) = \alpha. \tag{7.4}$$

We use the Newton-Raphson method to provide an approximation, $\hat{u}$, of the regulatory capital given $\alpha = 0.001$ in our numerical examples. Since the value of $\frac{\partial \psi(u,t)}{\partial u}$ is always negative, the initial surplus obtained by the process is guaranteed to approach the critical value.

In addition to the regulatory capital, insurance companies are also interested in its sensitivities to risk factors. For instance, if the claims arrival rate increases, how much more capital h_u, they should raise in order to maintain the solvency level, i.e. $\psi(u + h_u, t) = \alpha$. Sensitivity analysis is required for insurance companies in their financial condition reports in Australia, the UK and Canada. For meeting the prudential regulations, Cocozza and Di Lorenzo (2006) presented various methodologies for solvency assessment of life insurance businesses, but their results only focused on the investment risk; Hardy (1993) has shown that a stochastic simulation method exceeds the traditional deterministic sensitivity tests approach at estimating the relative probabilities of insolvency under different investment strategies and the timing of potential solvency problems. We compute sensitivities of the regulatory capital with respect to all parameters, which provide information for insurance companies in terms of meeting solvency requirements. To demonstrate the validity and efficiency of our simulation algorithm, numerical experiments are performed under the classical risk model, the Sparre Andersen model with interest and the periodic risk model with interest.

The remaining sections of the report are organized as follows. The basic idea of the stratified sampling algorithm is presented in Section 7.2. In Section 7.3, we compute first- and second-order sensitivities of finite-time ruin probabilities by both SFRSS and SFRDS. In Section 7.4, we apply the Newton-Raphson method to approximate the regulatory capital and compute its sensitivities.

7.2 Simulating the surplus process

7.2.1 Overview on rare event simulation

Estimating $\psi(u, t)$ via direct simulation of the surplus process R_t would require estimating $\mathbb{P}(E)$, where

$$E = \{\tau(u) \leq t\}.$$

When the initial surplus level u is large, $\psi(u, t) = \mathbb{P}(E)$ is small. The direct Monte-Carlo estimate of $\psi(u, t)$ is

$$\hat{\psi}(u, t)_n = \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}_E^j,$$

where $\mathbb{I}_E^j$'s are independent pathwise estimates and n is the number of paths in the simulation exercise. By the Central Limit Theorem, the relative error in the estimate is described by the approximation,

$$\frac{\hat{\psi}(u,t)_n}{\psi(u,t)} - 1 \approx \sqrt{\frac{(1-\psi(u,t))}{n\psi(u,t)}}Z,$$

where Z is the standard normal random variable. When $\psi(u,t)$ is small, it is unlikely to produce a reliable estimate unless the sample is large. Consequently, naïve rare event simulations are prohibitively expensive for estimating finite-time ruin probabilities.

Due to their importance across different fields of endeavour, fast simulation techniques have been introduced to study the occurrence and impact of rare situations. These methods efficiently reduced relative errors of the estimates in practical situations where crude Monte Carlo is insufficient to provide a sensible answer. Rubino and Tuffin (2009) presented a detailed account of the theoretical basis for modern rare event simulation techniques. They also discussed various applications of these techniques, such as performance measure simulations in queues, nuclear particle transport and biological system simulations. While these advanced techniques have many interesting applications, none of them have been applied to estimate finite-time ruin probabilities. In this paper, we apply one of these methods, i.e. stratified sampling. The idea of stratified sampling is very similar to that of importance sampling. The sample space of the target distribution is divided into K regions, called strata, and then one can compute the sample estimate, g_i, in each stratum. The resulting pathwise estimate is

$$g = \sum_{i=1}^{K} p_i g_i,$$

where p_i is the weight of the ith stratum. Here, we briefly summarize other rare event simulation techniques and leave the application of these methods to the future.

The most commonly used technique for rare event simulation is importance sampling, especially when the underlying state variables are light-tailed. It modifies the direct Monte-Carlo method, and generates random weighted samples from an equivalent probability distribution rather than the distribution of interest. The basic idea is to introduce a probability measure $\mathbb{Q}$ such that the likelihood ratio or the Radon-Nikodym derivative between the original probability measure, $\mathbb{P}$, and $\mathbb{Q}$ is well defined on the event of interest. Then the pathwise unbiased estimator becomes

$$\mathbb{I}_{\omega \in E} \frac{d\mathbb{P}}{d\mathbb{Q}}(\omega).$$

Here, ω is the underlying random outcome simulated according to the probability measure $\mathbb{Q}$. Asmussen and Glynn (2007) showed that the optimal choice of measure change is $\mathbb{P}(\cdot|E)$. Thus, we wish to use a sampling distribution $\mathbb{Q}$ that resembles as closely as possible to the conditional distribution of $\mathbb{P}$ given E. This approach is widely used for estimating performance measures in queuing systems. A typical example is to consider a random variable, $S_n = \sum_{i=1}^n x_i$, where x_i is identically distributed independent random variables with $\mathbb{E}[x_i] < 0$, and estimate the rare event probability

$$\mathbb{P}\Big(\sup_{n\in\mathcal{N}} S_n \geq b\Big),$$

for a deterministic b that is large (Blanchet and Glynn, 2008). This is different from our problem of estimating finite-time ruin probabilities, the value of b in our context is stochastic and depends on the claim arrival process.

Importance sampling has limited application, when the problem dimensionality is high or when the optimal importance sampling density is too complex to obtain. For computing finite-time ruin probabilities, one can apply importance sampling, i.e. perform measure changes on either the claim arrival process or the individual claim size random variable. However, the surplus process here depends critically on the timing of the claims as well as other factors such as investment returns, so it is hard to determine the optimal measure change. It is challenging to implement importance sampling efficiently for computing finite-time ruin probabilities, especially in a general framework that will allow solutions for many classes of problems.

Conditional Monte Carlo is another method for rare event simulation (Asmussen and Glynn, 2007). The idea of the method is that given an auxiliary vector of random variables, Z, we have

$$\mathbb{E}\big[\mathbb{P}(E|Z)\big] = \mathbb{P}(E),$$

so $\mathbb{P}(E|Z)$ is an unbiased estimator of $\mathbb{P}(E)$. More importantly, it has a smaller variance than the crude Monte Carlo estimator. Based on this idea, an efficient method was introduced in Asmussen and Binswanger (1997) to estimate ultimate ruin probabilities with heavy-tailed random claim sizes. Using the Pollaczeck-Khinchine formula, they transformed the problem of computing ultimate ruin probabilities to computing

$$\mathbb{P}\left(\sum_{i=1}^{K} X_i > u\right)$$

where K follows a geometric distribution. It was then estimated using an efficient conditional Monte-Carlo algorithm. This method was improved in the random K case by incorporating control variates and stratification techniques (Asmussen and Kroese, 2006). Their estimators of ultimate ruin probabilities have bounded relative errors. However, we do not have the luxury of the Pollaczeck-Khinchine formula in the finite case. Thus, it is difficult to apply the existing conditional Monte-Carlo method to computing finite-time ruin probabilities.

Another method called “Splitting” was introduced to deal with the problem of estimating rare event probabilities (Villen-Altamirano and Villen-Atamirano, 1991). Basically, it partitions the space-state of the system into a sequence of nested subsets and considers the rare event as the intersection of them. That is, consider

$$E_1 \supset E_2 \cdots \supset E_m = E,$$

the small probability of event E can be decomposed into

$$\mathbb{P}(E) = \mathbb{P}(E_1)\mathbb{P}(E_2|E_1)\ldots\mathbb{P}(E_m|E_{m-1}),$$

with each conditional event being "not rare". When a given subset is entered by a sample path, random sub-simulations are generated from the initial state corresponding to the state of the system at the entry point. Thus the system has been split into a number of new sub-simulations. The final estimate is the product of individual estimates. Splitting is used for estimating the first-entrance probability that a Markov process first enters an unlikely set B before another likely set A, after starting in neither A nor B (Dean and Dupuis, 2009). This is different from estimating finite-time ruin probabilities.

Glasserman et al (1999) analyzed the splitting method, and showed that choosing the degree of splitting correctly produces asymptotically optimal estimates when the state space satisfies some dimensionality conditions. Practically, it implies that too much splitting results in explosive computational requirements, and too little splitting eliminates any reduction in variance. To address this issue, Dean and Dupius (2011) formulated multi-level splitting algorithms for simulating rare probabilities, where the underlying state variable is discrete. In the case of computing finite-time ruin probabilities, the distribution of the underlying surplus process is often continuous and typically difficult to work with. A carefully designed sequence of subsets is required to apply the method, moreover, an efficient algorithm for estimating these conditional probability is crucial.

7.2.2 The idea of the stratified sampling algorithm

Since the rare event simulation techniques of the previous section have not yet been applied to computing finite-time ruin probabilities, we use a simple stratified sampling to deal with the problem. Stratified sampling is widely used for variance reduction purposes by the Monte Carlo community. The idea in this paper is similar to the simulation algorithm in Joshi and Kainth (2003) for pricing credit derivatives. As opposed to direct simulation, it produces non-zero pathwise estimates for all paths with one or more claims.

The conditions for the stratified sampling to apply are

1. we can observe the number of claims, $N_t = n$, for $n = 0, 1, \ldots$, and the exact values of T_i's before the observation of the individual claim sizes;

2. given $R_i^* > 0$, the surplus level just before the ith claim, there exists a twice-differentiable function, $A_{X_i}(R_i^*(\eta, \phi), \eta)$, such that

$$V_{X_i} < A_{X_i} \Leftrightarrow R_i < 0,$$

where V_{X_i} is one of the simulated random uniforms for generating X_i.

This is the actual probability of ruin at the ith claim, X_i, given the surplus level just before the claim. Here we shall only present this case, a similar approach is valid for the opposite situation, i.e $V_{X_i} > A_{X_i} \Leftrightarrow R_i < 0$.

The first step of the algorithm is to simulate the number of claims, n, within a finite-time horizon. If $n = 0$, the pathwise estimate is zero. Otherwise, the surplus level is forced to drop below zero at the ith claim with probability

$$a_i = \frac{1}{n+1-i},$$

provided that ruin has not occurred through the first $i-1$ claims. This number a_i only depends on the total number of claims within the time horizon, not the surplus level just before the claim. This ensures that ruin always occurs and is equidistributed between the n claims. To achieve this, a change of measure is performed on one standard uniform, V_{X_i}, for generating X_i until ruin occurs. The standard uniform, V_{X_i}, is replaced by a change of variable function, $U_i(V_{X_i}, A_{X_i})$, of it, i.e.

$$U_i(V_{X_i}, A_{X_i}) = \begin{cases} \frac{A_{X_i}}{a_i} V_{X_i}, & V_{X_i} < a_i, \\ \frac{1-A_{X_i}}{1-a_i}(V_{X_i} - a_i) + A_{X_i}, & V_{X_i} \geq a_i. \end{cases} \tag{7.5}$$

The resulting likelihood ratio weight from the measure change is

$$W_i(V_{X_i}, A_{X_i}) = \begin{cases} \frac{A_{X_i}}{a_i}, & V_{X_i} < a_i, \\ \frac{1-A_{X_i}}{1-a_i}, & V_{X_i} \geq a_i. \end{cases} \tag{7.6}$$

The pathwise estimator of the finite-time ruin probability under the stratified sampling algorithm is

$$\hat{g}^{SS}(\eta) = \mathbb{I}_{N_t>0} \prod_{i=1}^{n_r} W_i, \tag{7.7}$$

where n_r is the number of the claim at which the surplus level drops below zero.

7.2.3 Numerical examples

We consider three models in our numerical experiments for comparing the results from direct simulation and the stratified sampling algorithm in estimating the finite-time ruin probabilities. The numerical experiments in this section are conducted by 50,000 paths samples.

The classical risk model

We choose the exponential inter-claim time and the exponential claim size for benchmark purposes, see Dickson and Willmot (2005) for the analytical results.

For the stratified sampling algorithm, we need to compute the critical value function, A_{X_i}. Given V_{X_i}, the exponential claim size, X_i, is computed as

$$X_i = -\mu \log(V_{X_i}),$$

where $\mu = \mathbb{E}[X_i]$. The critical value function is

$$A_{X_i}(\mu, R_i^*) = \exp\Big(-\frac{R_i^*}{\mu}\Big).$$

We first set the parameters $\mu = 1$, $\lambda = 1$, $\theta = 0.1$, $u = 10$, and vary the finite-time t. The finite-time ruin probabilities computed by both methods are consistent with the analytical results, the numerical results are shown below.

t	Analytical	DM mean(S.E.)	SS mean (S.E.)
2	0.0013	0.0013(0.000158)	0.0012(0.00013)
4	0.0059	0.0056(0.00333)	0.0065(0.00046)
6	0.0131	0.0140(0.00052)	0.0138(0.00071)
8	0.0220	0.0216(0.00065)	0.0226(0.00099)
10	0.0319	0.0307(0.00077)	0.0322(0.00131)
20	0.0822	0.0801(0.00121)	0.0807(0.00256)
40	0.1573	0.1523(0.00161)	0.1506(0.00408)

Table 7.1: The classical risk model: simulated finite-time ruin probabilities by the direct method and the stratified sampling algorithm with 50,000 paths samples

- The stratified sampling algorithm underperforms direct simulation when $\psi(u,t)$ is large, this is because the change of measure at each X_i introduces excessive likelihood ratio weights.

- The stratified sampling algorithm outperforms direct simulation when ruin is rare and the change of measure induces a faster convergence.

For the classical model, the numerical results suggest that the stratified sampling algorithm is the preferred method when simulating rare finite-time ruin probabilities.

We further consider the computational times of the two methods for estimating the finite-time ruin probabilities. Let $u = 20$ and vary $t = 1, 2, \ldots 15$, in the following figure, we plot the time taken to estimate $\psi(u,t)$ by the two methods with the same set of parameters as above.

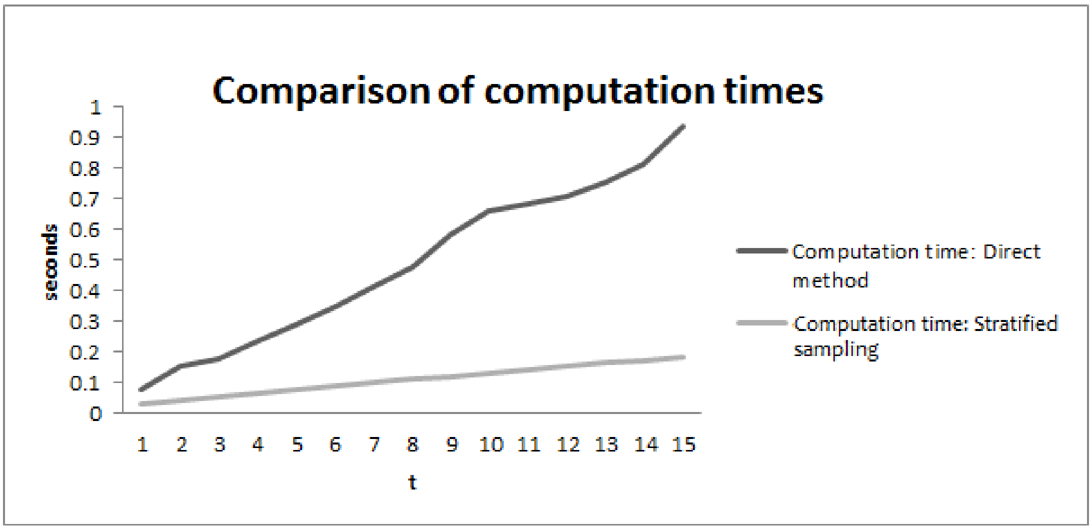

Figure 7.1: The classical model: the computation times of the direct method and the stratified sampling method based on 50,000 paths samples

The plot indicates that the stratified sampling algorithm reduces the computational effort when the probability of ruin is small, because

- the algorithm stops on each path when ruin occurs,
- we have forced ruin to occur at a faster rate, i.e. $a_i > A_{X_i}$.

The Sparre Andersen model with a deterministic investment return on surplus
The traditional surplus process in risk theory assumes that the surplus receives no interest, but a large portion of the surplus of insurance companies comes from investment income. Infinite-time ruin probabilities with a constant interest rate were estimated in Sundt and Teugels (1995) for the classical risk model. In particular, they considered the case with zero initial reserve, and the case with exponential claim sizes. Konstantinides et al (2002) extended the results to allow heavy-tailed claim sizes. These results are restricted to the infinite-time ruin probabilities under the classical model. Despite the practical importance of with-interest models, there has been no progress made in literature on solving the finite-time ruin probabilities with interest, nor the Sparre Andersen model with interest.

Here, we consider the Sparre Andersen model with interest, that is,

$$dR_t = rR_t dt + (1+\theta)\mu\lambda dt - d\sum_{i=1}^{N_t} X_i,$$

where $\mu = \mathbb{E}[X_i]$ and $\lambda = \frac{1}{\mathbb{E}[T_i]}$. While obtaining the analytical solution of the ruin probabilities via Laplace transform is intractable, Monte Carlo simulation provides an alternative solution to the problem. For the numerical experiments,

- the inter-claim times are distributed $Erlang(m, \beta)$, we set $m = 2$ and $\beta = 2$, so $\mathbb{E}[T_i] = 1$,

- the claim size random variables are distributed $Pareto(a, b)$, we set $a = 3$ and $b = 2$, so $\mathbb{E}[X_i] = 1$,
- the finite time horizon is 10,
- the premium loading is 0.1 and the constant force of interest is 0.1.

We set the initial surplus to $1, 5, 10, 25$ and 30, and compare the results simulated by the direct method and the stratified sampling algorithm, see the following table. The numerical results show that the stratified sampling algorithm outperforms the direct method significantly for the Sparre Andersen model with interest for most cases, except when the ruin probability is significantly large. The time taken to compute the finite-time ruin probability with $u = 30$ by the direct method is 0.5208 seconds and the stratified sampling is 0.168 seconds. This indicates a reduction of computational effort due to a faster rate of ruin occurrence for the stratified sampling method.

u	DM mean	DM S.E.	SS mean	SS S.E.
1	0.42712	0.00221	0.42236	0.00263
5	0.09768	0.001328	0.09822	0.000973
10	0.02076	0.000638	0.02150	0.000369
25	0.00174	0.000186	0.00135	2.2011E-05
30	0.00106	0.000146	0.00078	1.2905E-05

Table 7.2: The Sparre Andersen with interest risk model: simulated finite-time ruin probabilities by the direct method and the stratified sampling algorithm with 50,000 paths samples

The periodic risk model with a deterministic investment return on surplus

Another important property of an insurer's risk business is that claims are sometimes caused by periodic phenomena. The previous two models have a constant claim arrival intensity over time, which makes them crude models for insurance portfolios under periodic environments. We allow a periodic intensity for the compound Poisson process in the next experiment, that is the number of claims up to time t, N_t follows a non-homogeneous Poisson process. The general theory for the periodic case has been derived in Asmussen and Rolski (1994), whose discussion relied on the properties of the martingale non-homogeneous Poisson process. A practical simulation algorithm for the surplus process was introduced by Morales(2004). The ultimate ruin probabilities computed using their method were compared with the classical model results, it demonstrated a significant fluctuation depending on the current state of the cycle. However, their discussions were restricted to the ultimate ruin probabilities with exponential claim sizes.

For the numerical experiment, we compute the finite-time ruin probabilities with lognormal claim sizes. This example is of more practical interest to the insurance industry. The following intensity function is considered,

$$\lambda(t) = a + b\cos(2\pi ct), \text{ for } t > 0, \tag{7.8}$$

where $a > 0$, $b > 0$ and $c > 0$ are parameters of the model. This intensity function has a maximum value of $a + b$ and a minimum value of $a - b$. The period of the seasonal behaviour can be modified through the value c in the function. To simulate the time of the next arrival conditional on a claim just arrived at time s, the thinning algorithm is used (Ross, 2006). In particular, the process is simulated by a majoring Poisson process with an intensity function $\lambda^*(t) = a + b$, as follows

1. *Let* $t_s = s$,
2. *generate* $V_{T_i} = U(0, 1)$, *set* $t_s = t_s - \frac{1}{a+b} log(V_{T_i})$;
3. *generate* V_D,
4. *if* $V_D \leq \frac{a+b\cos(2\pi c t_s)}{a+b}$, *set* $T_s = t_s$ *and accept, else go back to step 2.*

We also modify the model to incorporate a constant force of interest r, so that

$$dR_t = rR_t dt + (1+\theta)\mu\lambda(t)dt - d\sum_{i=1}^{N_t} X_i,$$

where N_t is a non-homogeneous Poisson process with intensity $\lambda(t)$.

For the numerical experiments, we choose

- the intensity parameters, $a = 1$, $b = 1$ and $c = 0.1$,
- the claim sizes lognormally distributed with $\nu = -0.5$ and $\sigma = 1$, so that $\mathbb{E}[X_i] = 1$,
- the force of interest $r = 0.1$ and the loading $\theta = 0.1$,
- the finite-time horizon $t = 10$.

We set the initial surplus to $1, 5, 10, 25$ and 30 to show that the stratified sampling algorithm results and the direct method results agree, and the stratified sampling algorithm outperforms the direct method when the probability of ruin is small, see Table 7.3. The time taken to compute the finite-time ruin probability with $u = 30$ by the direct method is 0.643 seconds and by the stratified sampling method is 0.372 seconds.

u	DM mean	DM S.E.	SS mean	SS S.E.
1	0.50542	0.002236	0.50379	0.003752
5	0.12144	0.001461	0.12254	0.001715
10	0.02076	0.000638	0.02304	0.000750
25	0.00038	8.716E-05	0.000473	3.391E-05
30	0.00028	7.482E-05	0.000181	6.422E-06

Table 7.3: The periodic risk with interest risk model: simulated finite-time ruin probabilities by the direct method and the stratified sampling algorithm with 50,000 paths samples

The above numerical results suggest that, the stratified sampling algorithm outperforms the direct method when the initial surplus is large, i.e. the probability of ruin is small. Our ultimate objective is to find the regulatory capital so the finite-time ruin probability over a 10-year period is very small, typically less than 0.1%, thus it is preferable to use the stratified sampling algorithm.

7.3 Sensitivity analysis on finite-time ruin probabilities and density estimation

Sensitivity analysis in risk theory is concerned with estimating derivatives of ruin probabilities with respect to parameters of interest. These derivative estimates serve the following purposes:

- it illuminates the underlying risk process, and identifies the most significant operational parameters;
- one can make use of sensitivities to find the optimal solution with respect to the parameters of interest;
- if the parameter is only partially known, they can be used to estimate the parameter from data.

The pathwise method, or IPA, has been introduced as an efficient way to compute parameter sensitivities of discrete-event systems. Glasserman (1991) provided a general formulation of IPA for a broad class of discrete-event systems, and stated sufficient conditions for these estimates to be unbiased. Pathwise estimators of finite-time ruin probabilities often fail to fall into this restricted class.

7.3.1 Sensitivities with respect to the inter-claim time distribution: remove pathwise discontinuities of N_t

To compute sensitivities of finite-time ruin probabilities with respect to the distributional parameters of the inter-claim time random variable via IPA, we need to ensure that the pathwise estimator of finite-time ruin is a $\hat{C}^2$ function of these parameters. Define

$$t_i = \sum_{j=0}^{i-1} T_j,$$

the arrival time of the ith claim. There is a pathwise discontinuity for each T_i when

$$T_i = t - t_{i-1}, \text{ provided that } t_{i-1} < t, \text{ for } i = 1, 2, \ldots,$$

and at the first claim arrival time when

$$T_0 = t.$$

That is, a small bump in the distributional parameters of T_i will change the number of claims, i.e. $N_t(\eta) \neq N_t(\eta_0)$. We remove such pathwise discontinuities by HOPP of chapter 3, to ensure the unbumped path and the bumped path have the same number of claims. A change of measure performed at each T_i to force the bumped path to finish on the same side of discontinuities as the unbumped paths, that is,

$$T_i > t - t_{i-1} \text{ if and only if } T_i^0 > t^0 - t_{i-1}^0. \tag{7.9}$$

We assume that there exists a twice-differentiable function, $A_{T_i}(\eta)$, such that

$$V_{T_i} < A_{T_i} \Leftrightarrow T_i > t - t_{i-1},$$

where V_{T_i} is one of the simulated standard uniforms for generating T_i. This A_{T_i} is the probability of having i claims conditional on $i-1$ claims already arrived, which is different from A_{X_i} in the previous section. We replace V_{T_i} by a change of variable function, $U_{T_i}(V_{T_i}, \eta)$, of it, such that

$$U_{T_i}(V_{T_i}, \eta) = \begin{cases} \frac{1-A_{T_i}(\eta)}{1-A_{T_i}(\eta_0)}\big(V_{T_i} - A_{T_i}(\eta_0)\big) + A_{T_i}(\eta), & V_{T_i} \geq A_{T_i}(\eta_0), \\ \frac{A_{T_i}(\eta)}{A_{T_i}(\eta_0)} V_{T_i}, & V_{T_i} < A_{T_i}(\eta_0). \end{cases} \tag{7.10}$$

The corresponding likelihood ratio weight is

$$W_{T_i}(V_{T_i}, \eta) = \begin{cases} \frac{1-A_{T_i}(\eta)}{1-A_{T_i}(\eta_0)}, & V_{T_i} \geq A_{T_i}(\eta_0), \\ \frac{A_{T_i}(\eta)}{A_{T_i}(\eta_0)}, & V_{T_i} < A_{T_i}(\eta_0). \end{cases} \tag{7.11}$$

Example: $T_i \sim \exp(\lambda)$
Given a standard random uniform V_{T_i},

$$T_i = -\frac{1}{\lambda}\log(V_{T_i}),$$

where $\lambda = 1/\mathbb{E}[T_i]$. The critical value function for T_i is,

$$A_{T_i}(t_{i-1}, t, \lambda) = \exp\Big(-\lambda(t - t_{i-1})\Big), \text{ for } t_{i-1} < t$$

and the critical value function for T_0 is

$$A_{T_0}(t, \lambda) = \exp\left(-\lambda t\right).$$

7.3.2 Sensitivities with respect to claim size distribution and structural parameters: remove the pathwise discontinuities when $R_s = 0$

For the stratified sampling, when the number of claims $N_t > 0$, simulated ruin is certain. As shown in section 7.2, the decision of ruin is determined by the set of probabilities, $a_i = \frac{1}{n+1-i}$. The changes of measure in equation (7.10) ensures that the bumped path has the same number of claims as the unbumped path, i.e. the value of n is the same. The decision of ruin is then determined by the same value of a_i for both the bumped and unbumped paths, thus the pathwise estimator of the stratified sampling algorithm has no pathwise discontinuities as R_s passing through zero.

However, we need to remove the pathwise discontinuities for the direct method, as small bumps of parameters may cause the surplus level of the bumped path to finish on different sides of zero from the unbumped path at each claim. Given $R_i^*(\eta,\phi) > 0$, assume there exists a twice-differentiable function, $A_{X_i}(R_i^*(\eta,\phi),\eta)$, for $i = 1, 2, \dots$, such that

$$V_{X_i} < A_{X_i} \Leftrightarrow R_i < 0,$$

where V_{X_i} is one of the simulated random uniforms for generating X_i. Notice, this function is exactly the same as the critical value function in Section 7.2. We replace V_{X_i} by a change of measure function of it, $U_{X_i}(V_{X_i},\eta,\phi)$, such that

$$U_{X_i}(V_{X_i},\eta,\phi) = \begin{cases} \frac{1-A_{X_i}(R_i^*(\eta,\phi),\eta)}{1-A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0)}(V_{X_i} - A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0)) + A_{X_i}(R_i^*(\eta,\phi),\eta), & V_{X_i} \geq A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0), \\ \frac{A_{X_i}(R_i^*(\eta,\phi),\eta)}{A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0)}V_{X_i}, & V_{X_i} < A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0). \end{cases} \tag{7.12}$$

The corresponding likelihood ratio weight is

$$W_{X_i}(V_{X_i},\eta,\phi) = \begin{cases} \frac{1-A_{X_i}(R_i^*(\eta,\phi),\eta)}{1-A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0)}, & V_{X_i} \geq A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0), \\ \frac{A_{X_i}(R_i^*(\eta,\phi),\eta)}{A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0)}, & V_{X_i} < A_{X_i}(R_i^*(\eta_0,\phi_0),\eta_0). \end{cases} \tag{7.13}$$

7.3.3 The SFRSS method and the SFRDS method

After the sequence of measure changes, we obtain two new pathwise estimators for finite-time ruin probabilities. For direct simulation, changes of measure are performed at each T_i and X_i, the SFRDS pathwise estimator of finite-time ruin probability is

$$\hat{g}^{SFRDS}(\eta) = \Big(\prod_{i=1}^{n_r-1} \mathbb{I}_{t_{i-1}<t} W_{T_{i-1}} \mathbb{I}_{X_i<R_i^*} W_{X_i}\Big)\Big(\mathbb{I}_{t_{n_r-1}<t} W_{T_{n_r-1}} \mathbb{I}_{X_{n_r}>R_{n_r}^*} W_{X_{n_r}}\Big), \tag{7.14}$$

where $n_r = 1, 2, ...$ is the number of the claim at which the surplus level drop below zero. Similarly, we derive the SFRSS pathwise estimator of finite-time ruin probability,

$$\hat{g}^{SFRSS}(\eta) = \Big(\prod_{i=1}^{N_t} \mathbb{I}_{t_{i-1}<t} W_{T_{i-1}}\Big)\mathbb{I}_{t_{N_t}>t} W_{T_{N_t}} \times \Big(\prod_{i=1}^{n_r-1} \mathbb{I}_{X_i<R_i^*} W_i\Big) I_{X_{n_r}>R_{n_r}^*} W_{n_r}. \tag{7.15}$$

The SFRDS and SFRSS pathwise estimators of finite-time ruin probability have the following properties:

1. $\hat{g}^{SFRDS}(\eta)$ and $\hat{g}^{SFRSS}(\eta)$ are $\hat{C}^2$ since the composition of differentiable functions is differentiable;

2. the finite-differencing estimator of first and second-order derivatives Glasserman (2004) under the new schemes are uniformly integrable.

Now, we can apply the pathwise method to $\hat{g}^{SFRDS}(\eta)$ and $\hat{g}^{SFRSS}(\eta)$ to construct unbiased estimators of the corresponding gradient and Hessian.

7.3.4 Numerical results for the gradient and Hessian of the finite-time ruin probabilities

We use the same set of parameters as in Section 2, and the sample size is 50,000 paths for each simulation.

The classical model
We consider $t = 10$ for $u = 10$ and $u = 20$, see Tables 7.7, 7.8, 7.9, and 7.10. The results show that the gradient and Hessian computed by the two methods agree. When $u = 10$, $\psi(10, 10) = 0.0319$, for a sample of 50,000, the event ruin occurred in 1533 paths by the direct method. The SFRDS method is a better algorithm for estimating the sensitivities in this case; the sum of standard errors produced by the SFRDS method is 2.302782 as opposed to 64.0927 by SFRSS for all first- and second-order sensitivities. When $u = 20$, the probability of ruin is 0.0002776 (by the stratified sampling method), ruin occurred in 17 paths by the direct method. The SFRSS method is better in this case since ruin is a rare event; the sum of standard errors produced by the SFRDS method is 0.695968 as opposed to 0.092013 by SFRSS for all first and second-order sensitivities.

We further consider the computational times of the two methods for estimating sensitivities of the finite-time ruin probabilities. We set $t = 1, 2..., 15$ and $u = 20$ and plot the time taken by the two methods with the same set of parameters as above.

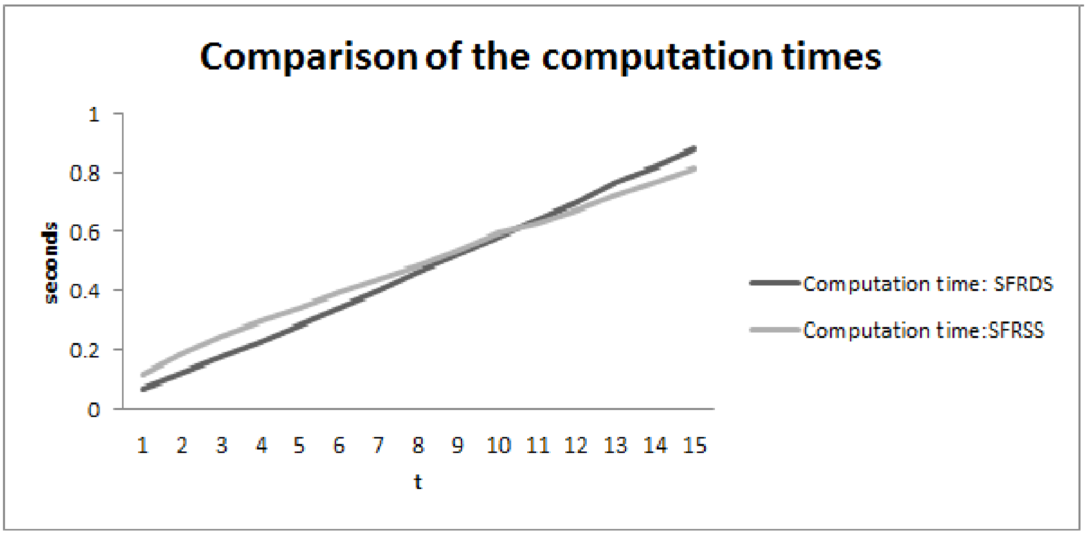

Figure 7.2: The classical model: the computation times of the SFRDS method and the SFRSS based on 50,000 paths samples

The plot suggests that

- the SFRSS method introduces more computational efforts when the time is small and the probability of ruin is extremely small, since the number of non-zero paths is significantly more than the direct method;
- the SFRSS method reduces the computational efforts when the time is big and it forces ruin to occur at a faster rate.

The Sparre Andersen model with interest

We set $u = 30$ and $t = 10$, the probability of ruin is small in this case (ruin occurred in 53 paths by direct simulation), see Tables 7.11 and 7.12 for the results. The SFRSS method outperforms; the sum of standard errors produced by SFRSS is 0.017551 as opposed to 0.1505458 by the SFRDS method. The time taken by the SFRDS method is 0.473 seconds and by the SFRSS method is 1.1415 seconds.

The Periodic risk model with interest

One important subtlety to point out about the thinning algorithm for generating the non-homogeneous Poisson process is its inherent discontinuities at each acceptance-rejection point, i.e. if

$$V_D \leq \frac{a + b\cos(2\pi c t_s)}{a+b},$$

set $T_s = t_s$ and accept, else reject. A small bump in the parameters of interest may alter the acceptance-rejection decision, and consequently results in different final accepted outcomes from the bumped path and the unbumped path. To remove such pathwise discontinuities, we use the OSRS algorithm introduced. A change of measure is performed on each V_D to ensure the bumped path makes the same acceptance-rejection decision as the unbumped path.

We set $t = 10$ and $u = 25$, the probability of ruin is small in this case, see Tables 7.13 and 7.14 for the results. There are only 24 paths in the direct simulation with non-zero estimates. The SFRSS method outperforms; the sum of standard errors produced by SFRSS is 0.287805 as opposed to 1.192388 by the SFRDS method. The time taken by the SFRDS method is 0.707 seconds and by the SFRSS method is 1.416 seconds.

The numerical results show that the SFRSS method is more appropriate for cases where ruin is rare, because it produces unbiased non-zero pathwise estimators for all paths with at least one claim. Unlike the SFRSS method, the SFRDS method only produces non-zero estimates when ruin occurs, thus it suffers the following problems when the event ruin is rare:

- it produces estimates with large sample standard errors;
- the estimates produced by a few significant paths are unlikely to be reliable.

7.3.5 The density of the time to ruin

In recent years, the risk theory community has focused on the actual distribution of the time to ruin. So far, its density can be derived analytically for only a few special cases, and often requires an enormous amount of computational effort.

In this section, we estimate the density function of the time to ruin via Monte-Carlo simulation. Since the finite-time ruin probability,

$$\psi(u,t) = \mathbb{P}(\tau(u) \leq t),$$

is the cumulative density function of $\tau(u)$, the probability density function of $\tau(u)$ is then

$$f_{\tau(u)}(t) = \frac{\partial \psi(u,t)}{\partial t}.$$

Both the SFRSS method and the SFRDS method provide unbiased estimates of the density.

The classical model

We first compare the density estimated by the SFRSS method and the SFRDS method against the analytical results. The density of the time to ruin can be computed explicitly for the classical model with exponential claim size by the following formula,

$$f_{\tau(u)}(t) = \lambda \exp\left(-\mu u - (\lambda + \mu c)t\right)\Big(I_0\big((4\mu\lambda t(u+ct))^{0.5}\big) - \frac{ct}{ct+u} I_2\big((4\mu\lambda t(u+ct))^{0.5}\big)\Big),$$

where I_i is the modified Bessel function of the ith kind (Dickson, 2007). The numerical experiments in this section is performed by setting $u = 5$ and $\theta = 0.25$, the other parameters are the same as in Section 7.2. In order to create a smooth graph, we use 500,000 paths for each simulation and 60 equally spaced-time points over a period of 15 years.

The density estimated by the SFRDS method (the unsmooth dark line) is plotted against the analytic formula (the smooth dotted line), we also included the line representing the density estimated plus one standard error and the line representing the density estimated minus one standard error to show the convergence of the estimated density function, see Figure 7.3. The maximum obtained is 0.01869 with 2.087E-04 standard error when $t = 2.5$, and the corresponding analytical result is 0.01873. The minimum obtained is 0.00613 when $t = 15$ with 1.110E-04 standard error and the corresponding analytical result is 0.00603. The maximum absolute difference between the density estimated by SFRDS and the analytical density is 6.452E-04 when $t = 2.25$, the corresponding Monte Carlo standard error is 4.105E-04. The minimum absolute difference between the density estimated by SFRSS and the analytical density is 4.275E-06 when $t = 13.25$, the corresponding Monte Carlo standard error is 1.119E-04.

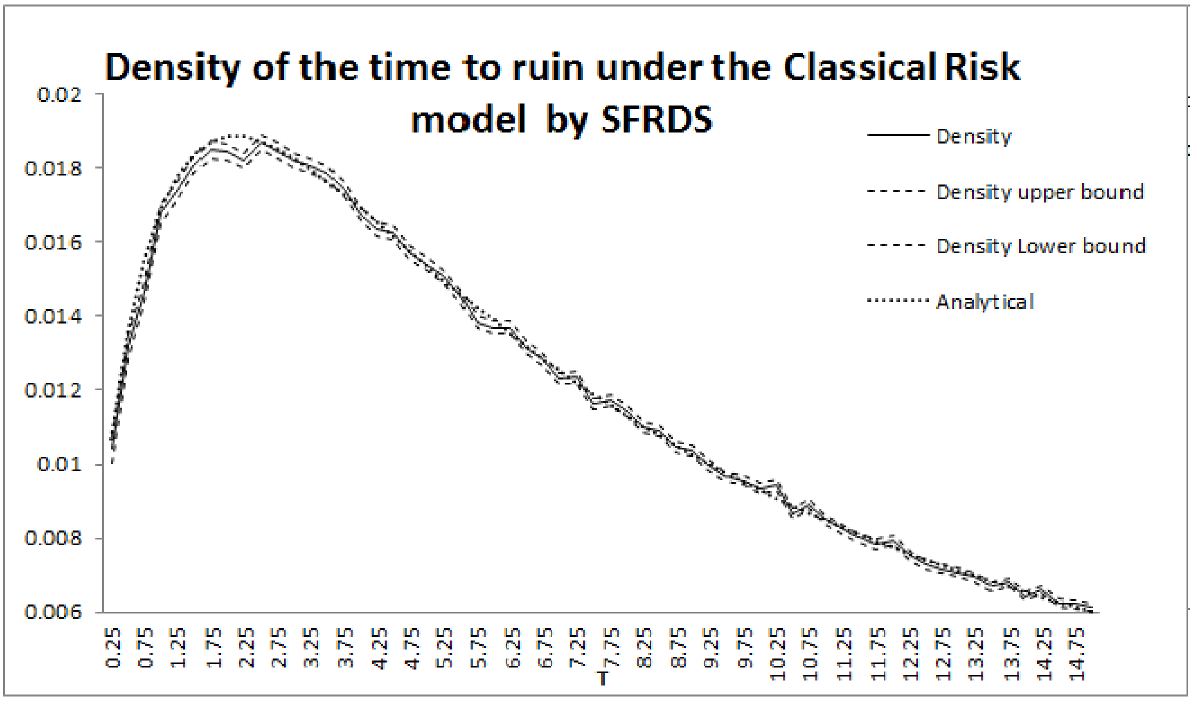

Figure 7.3: The classical risk model: the density of the time to ruin estimated by setting $u = 5$ and $\theta = 0.25$ with a 500,000 paths sample using SFRDS

In Figure 7.4, we plot the density estimated by SFRSS against the analytical results. The maximum obtained is 0.01909 when $t = 2.5$ with 3.407E-04 standard error, and the corresponding analytical result is 0.01873. The minimum obtained is 0.00618 when $t = 15$ with 7.480E-04 standard error and the corresponding analytical result is 0.00603. The maximum absolute difference between the density estimated by SFRSS and the analytical density is 9.116E-04 when $t = 14.25$, the corresponding Monte Carlo standard error is 7.320E-04. The minimum absolute difference between the density estimated by SFRSS and the analytical density is 2.065E-05 when $t = 1.25$, the corresponding Monte Carlo standard error is 3.008E-04.

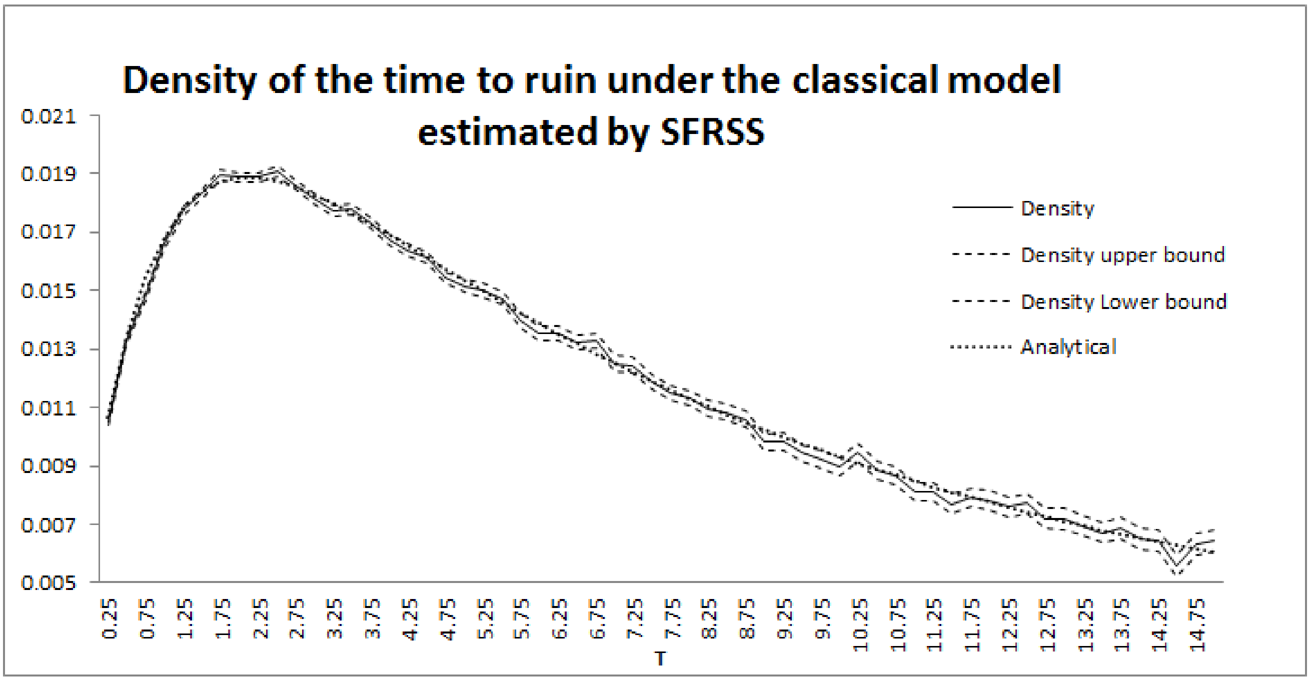

Figure 7.4: The classical risk model: the density of the time to ruin estimated by setting $u = 5$ and $\theta = 0.25$ with a 500,000 paths sample using SFRSS

The numerical results demonstrate

- that both the SFRSS and the SFRDS methods produce unbiased estimates of the density function,
- when t is small, it is preferable to use the SFRSS method since the corresponding finite-time ruin probability is small,
- when t is big, it is preferable to use the SFRDS method since the corresponding finite-time ruin probability is big.

The Sparre Andersen risk model with interest

Insurance companies typically hold large initial capitals for meeting prudential requirements. The density of the time to ruin sheds important insights into the risk profile over a long term horizon. In the following graph, the density of the time to ruin over the period $[0, 10]$ under the Sparre Andersen model with interest is plotted. For the plot, we assume $u = 30$, $\theta = 0.1$, and the other parameters are the same as in Section 7.2. The finite-time probability $\psi(30, 10)$ is 0.00078 by the stratified sampling algorithm, therefore, it is preferable to use the SFRSS method. The graph plotted demonstrates a low probability of ruin at the beginning, follows by a period of higher ruin probabilities. The ruin probability gradually decreases due to the premium collection with a loading factor at 10% as well as the interest accumulation at 10%. Based on the risk profile illustrated by the density function, insurance companies can design their pricing as well as risk management strategies to reflect such pattern. For instance, companies need to hold large capitals during the early years of the risk business, and some portion of the capital may be released later if a significant surplus has been accumulated.

In order to create a smooth graph, we use 5,000,000 paths for each simulation and 40 equally spaced-time points over a period of 10 years. The maximum obtained is 1.841E-04 when $t = 1$ with 2.7025E-06 standard error. The minimum obtained is 2.609E-05 when $t = 10$ with 2.835E-06 standard error. The time taken to produce the density estimated is 53 minutes and 6.75 seconds.

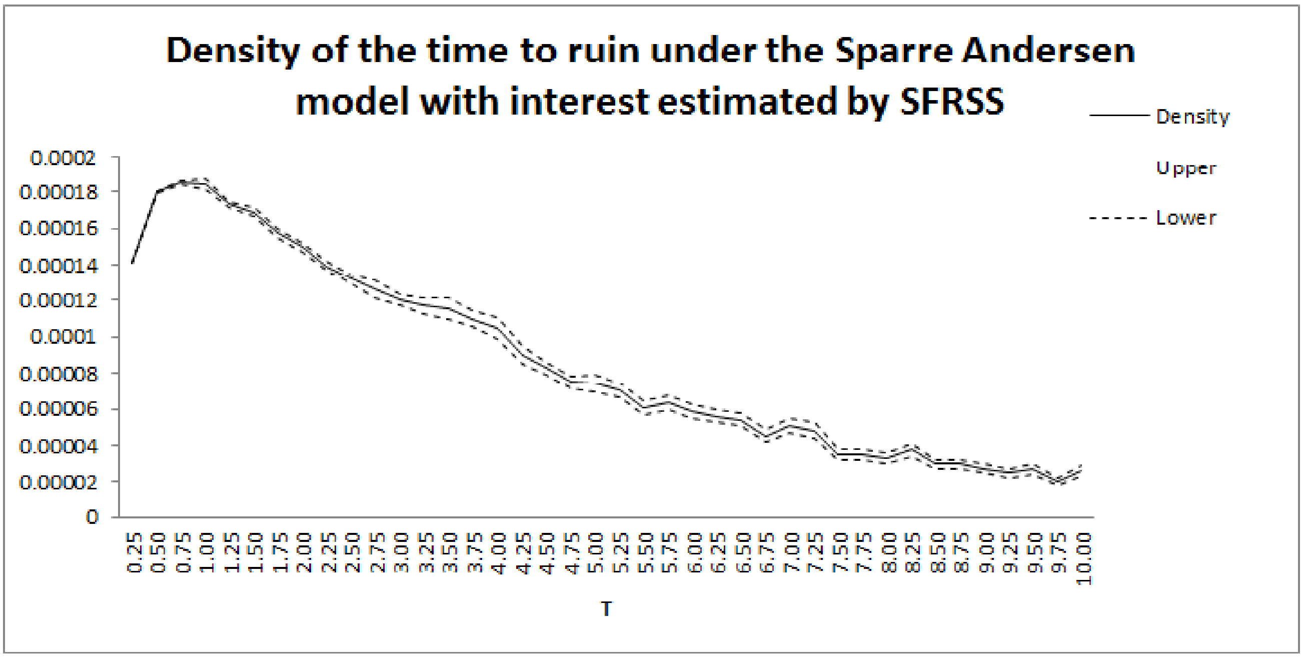

Figure 7.5: The Sparre Andersen with interest risk model with Erlang(2,2) inter-claim times and Pareto(3,2)claim sizes: the density of the time to ruin estimated by setting $u = 30$ and $\theta = 0.1$ with a 5,000,000 paths sample using SFRSS

The periodic risk model with interest

Regulators are interested in the density of the time to ruin after the surplus level reaches zero. It provides information on the possibility of the surplus recovery. In particular, the seasonal pattern of insurance claims plays an important role in the performance of the insurance companies, it allows companies to recover after significant losses over the period with low claim arrival intensity. The periodic risk model captures such phenomena.

In the following graph, we plot the density of the time to ruin over the period $[0, 10]$ under the periodic risk model with interest by setting $u = 0$, $b = 0.9$ and $c = 0.25$, the other parameters are the same as in Section 7.2. The density is produced by the SFRDS method, since the probability of ruin in this case is large, even for $t = 0.25$. The graph plotted demonstrates that once the insurance portfolio survives through the initial period of high claim arrival intensity, it will recover its surplus level through the premium collection with a loading factor at 10$\%$ as well as the interest accumulation at 10 $\%$. The density of the time to ruin has a cyclical pattern due to the periodic risk structure, however the probability of ruin is generally low after the initial couple of cycles (4 years in this case).

In order to create a smooth graph, we use 50,000 paths for each simulation and 40 equally spaced-time points over a period of 10 years. The maximum obtained is 0.72433 when $t = 0.25$ with 0.00629 standard error. The minimum obtained is 5.660E-04 when $t = 10$ with 1.350E-04 standard error. The time taken to produce the density estimated is 17.483 seconds.

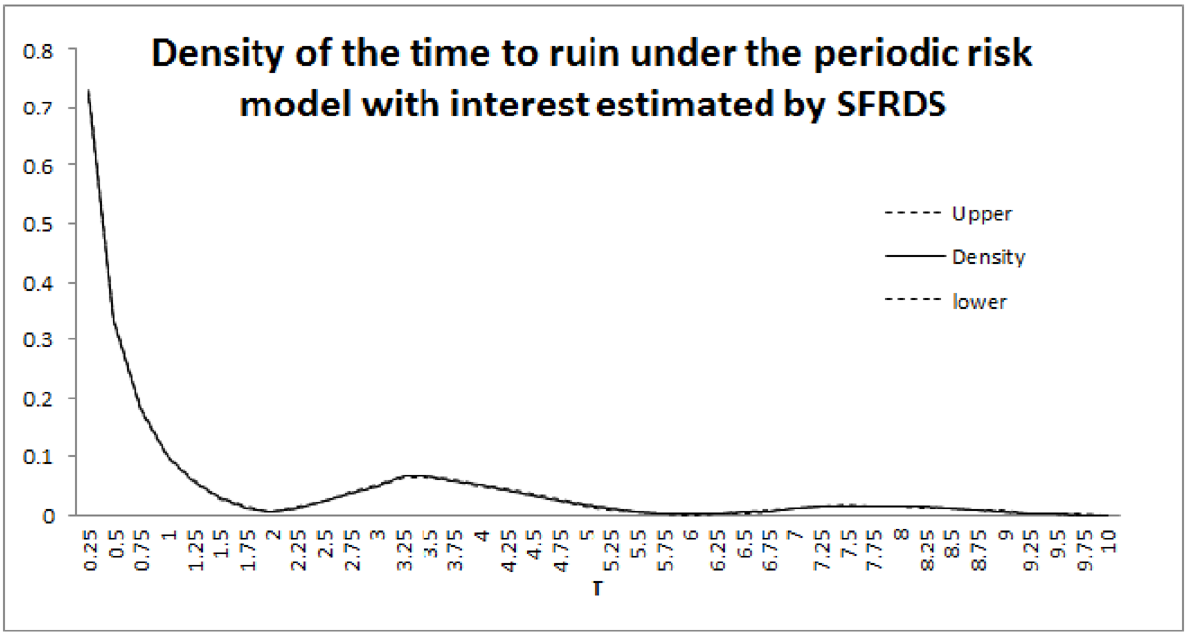

Figure 7.6: The periodic with interest risk model with $LN(-0.5, 1)$ claim sizes: the density of the time to ruin estimated by setting $u = 0$ and $\theta = 0.1$ with a 50,000 paths sample using SFRDS

7.4 Regulatory capital and its sensitivities

Regulatory capital requirements for insurers are the focus of the current development of a global framework for insurer solvency assessment. Risk theory is well known for its theoretical approach to the macro-analysis of underwriting portfolio risk that assists insurance companies to answer the question "how much capital to hold so that the finite-time ruin probability is smaller than $\alpha\%$?". To solve this question, we need to invert equation (7.4), i.e. solve

$$\psi^{-1}(\alpha, t) = u^*.$$

7.4.1 Numerically approximating u^* by the Newton-Raphson method

The Newton-Raphson method gives a numerical solution to this question. With an initial guess $\hat{u}_0$, the algorithm is performed iteratively as

$$\hat{u}_{i+1} = \hat{u}_i + \frac{\alpha - \psi(\hat{u}_i, t)}{\frac{\partial \psi}{\partial u}(\hat{u}_i, t)}, \tag{7.4.16}$$

until $|\psi(\hat{u}, t) - \alpha| < \epsilon$. Since the finite-time ruin probability ψ and its derivative with respect to u need to be computed repetitively, efficient algorithms for computing them are critical for the iterative process to converge. Generally, we do not have the luxury of analytical solutions for ψ and $\frac{\partial \psi}{\partial u}$; even for cases where they do exist, using them for such iterative calculation is not practically due to the computational cost. For the following numerical experiments, we use the stratified sampling algorithm for ψ and the SFRSS method for $\frac{\partial \psi}{\partial u}$, since the value of α is typically small in practice.

We perform experiments to search numerically for the regulatory capitals under the classical model, the Sparre Andersen model with interest and the periodic risk model with interest. The parameters are the same as in Section 7.2, for a finite-time horizon of $t = 10$ and $\alpha = 0.1\%$. The results are summarized in Tables 7.4, 7.5 and 7.6. The numerical experiments show that we can approximate the regulatory capital, u^*, using less than five iterations to obtain $|\psi(\hat{u}, t) - 0.001| < 10^{-5}$ for the three models considered.

7.4.2 Sensitivities of u^*

Our next objective is to perform sensitivity analysis of the regulatory capital. Since the value, u^*, is estimated based on the current assumptions made about the underlying risk process, insurance companies are also interested in how they should adjust the capital, u^*, in response to changes in the risk process. The derivatives of u^* with respect to the parameters of the underlying risk model provide a solution to this question.

To compute $\frac{\partial u^*}{\partial \eta}$, we rely on the following relationship,

$$\alpha = \psi(u^*, t, \eta),$$

where $\psi(u^*, t, \eta)$ is the finite-time probability of ruin given the current distributional and structural parameters. Since $u^* = \psi^{-1}(\alpha, t, \eta)$, we have

$$\alpha = \psi(\psi^{-1}(\alpha, t, \eta), t, \eta).$$

Differentiate the above expression with respect to η, to obtain

$$0 = \frac{\partial \psi}{\partial \eta}(u^*, t) + \frac{\partial \psi}{\partial u}(u^*, t)\frac{\partial u^*}{\partial \eta},$$

so that,

$$\frac{\partial u^*}{\partial \eta} = -\frac{\partial \psi}{\partial \eta}(u^*, t) / \frac{\partial \psi}{\partial u}(u^*, t).$$

Since we have already obtained the regulatory capital by the Newton Raphson method and its sensitivities by the SFRSS method, the sensitivities of u^* are then easily computable. It allows the practitioners to focus more on selecting an appropriate valuation basis, than on computations of the sensitivities.

Numerical results are summarized in Tables 7.4, 7.5 and 7.6 for the three models. Standard errors are for 50,000 paths and these could be reduced by sampling more paths. Based on the numerical results, the most important factor influencing the regulatory capital is the distributional parameters of the claim size. For the Sparre Andersen model with interest and the periodic risk model with interest, another critical parameter is the deterministic interest rate. Insurance companies should constantly monitor their underlying insurance portfolios to obtain accurate estimates of the claim size distribution, especially for businesses with a heavy-tailed nature. In addition, adequate forecasts of the business cycle are crucial for insurance companies which rely heavily on their investment portfolio returns.

	u^*	$\frac{\partial u^*}{\partial \theta}$	$\frac{\partial u^*}{\partial t}$	$\frac{\partial u^*}{\partial \lambda}$	$\frac{\partial u^*}{\partial \mu}$
mean	18.1647	-8.1679	0.7280	7.2805	18.1648
S.E.	0.1878	0.5962	0.1550	1.5499	1.3445

Table 7.4: The classical risk model: the estimated regulatory capital for $\alpha = 0.1\%$ and its sensitivities with a 50,000 paths sample

	u^*	$\frac{\partial u^*}{\partial \theta}$	$\frac{\partial u^*}{\partial t}$	$\frac{\partial u^*}{\partial \beta}$	$\frac{\partial u^*}{\partial r}$	$\frac{\partial u^*}{\partial a}$	$\frac{\partial u^*}{\partial b}$
mean	27.5705	-3.7089	0.4329	6.8424	-93.5544	13.6638	-27.6281
S.E.	0.1179	0.2525	0.0787	0.5301	4.9944	0.6309	1.1929

Table 7.5: The Sparre Andersen risk model with interest: the estimated regulatory capital for $\alpha = 0.1\%$ and its sensitivities with a 50,000 paths sample

	u^*	$\frac{\partial u^*}{\partial \theta}$	$\frac{\partial u^*}{\partial t}$	$\frac{\partial u^*}{\partial a}$	$\frac{\partial u^*}{\partial b}$	$\frac{\partial u^*}{\partial c}$	$\frac{\partial u^*}{\partial r}$	$\frac{\partial u^*}{\partial \nu}$	$\frac{\partial u^*}{\partial \sigma}$
mean	21.3467	-3.5519	0.1087	6.9633	0.4363	-4.0693	-52.4456	21.3467	67.79286
S.E.	0.2548	0.21683	0.0539	2.1794	0.0355	0.7668	2.16133	0.8590	2.0081

Table 7.6: The period risk model with interest: the estimated regulatory capital for $\alpha = 0.1\%$ and its sensitivities with a 50,000 paths sample

7.5 Conclusion

We have introduced a stratified sampling method for computing finite-time ruin probabilities, which outperforms the direct simulation method when ruin is rare. We perform changes of measure to remove discontinuities of the pathwise estimators of ruin, so the pathwise method is applicable to provide unbiased estimates of first- and second-order sensitivities of finite-time ruin probabilities. The derivative with respect to the finite-time horizon, t, provides a way of density estimation for the time to ruin random variable, especially when analytical solutions are not feasible. We further use the Newton-Raphson method to estimate the regulatory capital, and as well as its sensitivities. Numerical experiments on the classical model, the Sparre Andersen model with interest and the periodic risk model with interest, demonstrate the validity and the efficiency of the methods suggested,

Appendix one: tables of numerical results

SFRSS vs SFRDS	u	θ	t	λ	μ
u	3.913vs4.114	26.319vs26.152	-1.99vs-1.35	-19.898vs-13.505	-26.304vs-29.425
θ	26.319vs26.152	217.779vs209.952	-21.937vs-17.709	-219.369vs-177.095	-263.195vs-261.516
t	-1.99vs-1.35	-21.937vs-17.709	0.039vs0.063	4.686vs5.702	19.898vs13.505
λ	-19.898vs-13.505	-219.369vs-177.095	4.686vs5.702	-138764.653vs-9327.25	198.978vs135.05
μ	-26.304vs-29.425	-263.195vs-261.516	19.898vs13.505	198.978vs135.05	134.79vs177.059
first orders	-12.825vs-11.719	-90.858vs-83.461	5.074vs5.074	50.741vs50.738	128.254vs117.187

Table 7.7: The classical model: the mean $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=10

SFRSS vs SFRDS	u	θ	t	λ	μ
u	0.349vs0.357	2.206vs2.244	0.411vs0.148	4.111vs1.48	3.411vs3.256
θ	2.206vs2.244	17.235vs17.836	3.319vs1.535	33.19vs15.35	22.056vs22.44
t	0.411vs0.148	3.319vs1.535	1.533vs0.388	16.078vs4.077	4.111vs1.48
λ	4.111vs1.48	33.19vs15.35	16.078vs4.077	63757.311vs2120.802	41.113vs14.804
μ	3.411vs3.256	22.056vs22.44	4.111vs1.48	41.113vs14.804	33.989vs29.768
first orders	0.491vs0.462	3.38vs3.099	1.196vs0.322	11.962vs3.217	4.915vs4.623

Table 7.8: The classical model: the standard errors $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=10

SFRSS vs SFRDS	u	θ	t	λ	μ
u	0.13vs0.114	1.018vs0.892	-0.058vs-0.065	-0.582vs-0.648	-2.409vs-2.072
θ	1.018vs0.892	8.299vs7.284	-0.527vs-0.683	-5.271vs-6.835	-20.358vs-17.831
t	-0.058vs-0.065	-0.527vs-0.683	-0.094vs-0.068	0.872vs0.792	1.163vs1.296
λ	-0.582vs-0.648	-5.271vs-6.835	0.872vs0.792	-62.847vs-628.937	11.63vs12.959
μ	-2.409vs-2.072	-20.358vs-17.831	1.163vs1.296	11.63vs12.959	44.365vs37.378
first orders	-0.191vs-0.203	-1.46vs-1.597	0.071vs0.116	0.706vs1.159	3.825vs4.064

Table 7.9: The classical model: the mean $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=20

SFRSS vs SFRDS	u	θ	t	λ	μ
u	0.032vs0.047	0.254vs0.353	0.033vs0.041	0.328vs0.406	0.588vs0.885
θ	0.254vs0.353	2.236vs2.831	0.275vs0.416	2.75vs4.159	5.073vs7.065
t	0.033vs0.041	0.275vs0.416	0.035vs0.114	0.319vs1.188	0.656vs0.812
λ	0.328vs0.406	2.75vs4.159	0.319vs1.188	42.908vs629.389	6.564vs8.119
μ	0.588vs0.885	5.073vs7.065	0.656vs0.812	6.564vs8.119	10.92vs16.7
first orders	0.06vs0.06	0.435vs0.467	0.046vs0.059	0.464vs0.589	1.196vs1.194

Table 7.10: The classical model: the standard errors $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=20

SFRSS vs SFRDS	u	θ	t	β	r	a	b
u	0.111vs0.114	0.431vs0.379	-0.079vs-0.065	-0.815vs-0.699	8.442vs7.81	-1.234vs-1.297	2.324vs2.386
θ	0.431vs0.379	3.228vs2.323	-0.469vs-0.062	-5.042vs-1.763	53.926vs41.43	-6.461vs-5.691	10.667vs9.384
t	-0.079vs-0.065	-0.469vs-0.062	1.312vs0.512	8.019vs2.44	-22.246vs-12.239	1.18vs0.75	-2.67vs-1.004
β	-0.815vs-0.699	-5.042vs-1.763	8.019vs2.44	43.101vs12.762	-120.136vs-127.828	12.23vs8.107	-25.79vs-26.365
r	8.442vs7.81	53.926vs41.43	-22.246vs-12.239	-120.136vs-127.828	1365.632vs1093.043	-126.633vs-117.15	248.791vs227.662
a	-1.234vs-1.297	-6.461vs-5.691	1.18vs0.75	12.23vs8.107	-126.633vs-117.15	12.009vs13.151	-34.866vs-35.79
b	2.324vs2.386	10.667vs9.384	-2.67vs-1.004	-25.79vs-26.365	248.791vs227.662	-34.866vs-35.79	82.175vs81.925
first orders	-0.867vs-0.84	-2.897vs-2.757	0.694vs0.411	7.705vs5.865	-84.746vs-76.04	12.998vs12.595	-26.993vs-25.942

Table 7.11: The Sparre Andersen model with interest: the mean $\times 10000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=30

SFRSS vs SFRDS	u	θ	t	β	r	a	b
u	0.019vs0.005	0.142vs0.038	0.076vs0.039	0.425vs0.193	1.968vs0.291	0.223vs0.065	0.407vs0.101
θ	0.142vs0.038	1.743vs0.688	0.448vs0.254	2.616vs1.261	16.362vs3.329	2.125vs0.574	3.331vs0.842
t	0.076vs0.039	0.448vs0.254	1.392vs1.332	8.353vs6.717	21.263vs2.268	1.133vs0.585	2.552vs0.944
β	0.425vs0.193	2.616vs1.261	8.353vs6.717	45.608vs33.291	113.819vs11.85	6.382vs2.902	14.164vs4.692
r	1.968vs0.291	16.362vs3.329	21.263vs2.268	113.819vs11.85	382.783vs31.282	29.524vs4.37	57.501vs7.091
a	0.223vs0.065	2.125vs0.574	1.133vs0.585	6.382vs2.902	29.524vs4.37	2.412vs0.89	6.109vs1.518
b	0.407vs0.101	3.331vs0.842	2.552vs0.944	14.164vs4.692	57.501vs7.091	6.109vs1.518	13.778vs2.549
first orders	0.138vs0.031	0.741vs0.384	0.664vs0.344	3.734vs1.69	17.435vs1.986	2.065vs0.459	4.221vs0.722

Table 7.12: The Sparre Andersen model with interest: the standard errors $\times 10000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=30

SFRSS vs SFRDS	u	θ	t	a	b	c	r	ν	σ
u	0.016vs0.036	0.039vs0.045	-0.014vs-0.001	-0.093vs-0.706	0.123vs0.556	0.135vs0.547	0.818vs2.056	-0.34vs-0.82	-1.083vs-2.026
θ	0.039vs0.045	0.196vs0.65	-0.049vs-0.007	-0.987vs-0.839	-0.062vs-0.464	1.148vs2.784	3.397vs5.344	-0.974vs-1.617	-2.749vs-3.587
t	-0.014vs-0.001	-0.049vs-0.007	0.008vs0.003	0.022vs0.013	-0.002vs-0.028	0.096vs0.115	-0.348vs-0.874	0.13vs0.128	0.335vs0.415
a	-0.093vs-0.706	-0.987vs-0.839	0.022vs0.013	-25.136vs-25.223	-23.01vs-21.567	26.237vs10.117	-6.097vs-6.213	2.335vs1.661	7.069vs6.528
b	0.123vs0.556	-0.062vs-0.464	-0.002vs-0.028	-23.01vs-21.567	-21.107vs-21.391	22.213vs13.298	7.23vs6.081	-3.065vs-3.906	-9.876vs-7.021
c	0.135vs0.547	1.148vs2.784	0.096vs0.115	26.237vs10.117	22.213vs13.298	-46.792vs-34.892	11.39vs14.173	-3.363vs-3.667	-10.182vs-6.42
r	0.818vs2.056	3.397vs5.344	-0.348vs-0.874	-6.097vs-6.213	7.23vs6.081	11.39vs14.173	84.873vs69.498	-20.45vs-31.393	-69.441vs-80.81
ν	-0.34vs-0.82	-0.974vs-1.617	0.13vs0.128	2.335vs1.661	-3.065vs-3.906	-3.363vs-3.667	-20.45vs-31.393	8.491vs20.506	27.077vs19.922
σ	-1.083vs-2.026	-2.749vs-3.587	0.335vs0.415	7.069vs6.528	-9.876vs-7.021	-10.182vs-6.42	-69.441vs-80.81	27.077vs19.922	88.752vs127.542
first orders	-0.065vs-0.086	-0.197vs-0.178	0.046vs0.042	0.587vs0.756	-0.099vs-0.081	-0.633vs-0.844	-4.038vs-4.769	1.625vs2.159	5.831vs5.956

Table 7.13: The periodic risk model with interest: the mean $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=25

SFRSS vs SFRDS	u	θ	t	a	b	c	r	ν	σ
u	0.003vs0.015	0.014vs0.072	0.006vs0.001	0.063vs0.47	0.06vs0.61	0.095vs0.413	0.186vs1.133	0.067vs0.353	0.183vs0.869
θ	0.014vs0.072	0.067vs0.389	0.025vs0.006	0.401vs2.644	0.404vs3.427	0.597vs2.247	1.06vs6.813	0.34vs1.804	0.939vs4.352
t	0.006vs0.001	0.025vs0.006	0.173vs0.003	0.361vs0.012	0.398vs0.003	1.933vs0.054	0.695vs0.12	0.146vs0.022	0.521vs0.067
a	0.063vs0.47	0.401vs2.644	0.361vs0.012	10.135vs9.795	8.872vs9.499	11.813vs13.015	3.967vs48.91	1.573vs11.741	4.562vs31.791
b	0.06vs0.61	0.404vs3.427	0.398vs0.003	8.872vs9.499	8.173vs12.708	11.15vs11.429	4.234vs66.002	1.498vs15.251	4.57vs41.41
c	0.095vs0.413	0.597vs2.247	1.933vs0.054	11.813vs13.015	11.15vs11.429	26.246vs16.415	8.417vs44.125	2.373vs10.32	7.494vs28.225
r	0.186vs1.133	1.06vs6.813	0.695vs0.12	3.967vs48.91	4.234vs66.002	8.417vs44.125	20.676vs120.206	4.659vs28.33	14.533vs73.6
ν	0.067vs0.353	0.34vs1.804	0.146vs0.022	1.573vs11.741	1.498vs15.251	2.373vs10.32	4.659vs28.33	1.678vs8.83	4.584vs21.733
σ	0.183vs0.869	0.939vs4.352	0.521vs0.067	4.562vs31.791	4.57vs41.41	7.494vs28.225	14.533vs73.6	4.584vs21.733	12.982vs55.241
first orders	0.007vs0.027	0.033vs0.109	0.019vs0.001	0.208vs0.775	0.225vs0.993	0.304vs0.683	0.59vs1.956	0.167vs0.681	0.525vs1.814

Table 7.14: The periodic risk model with interest: the standard errors $\times 1000$ of finite-time ruin probabilities derivatives by the SFRSS method and the SFRDS with 50,000 paths samples when u=25

Chapter 8

Summary and Conclusion

The efficient valuation and risk-management of complex problems written on multiple stochastic processes have always been challenging tasks for financial institutions as well as other large organizations. The principal difficulty lies in the non-existence of a simple analytical solution. Fast numerical techniques such as lattice methods are, in general, not applicable due to the problem of high dimensionality. Most of the time, Monte-Carlo simulation methods are required to approximate the multi-dimensional integral in order to provide feasible solutions to these questions. The issue we addressed in this monograph is to the compution of sensitivities of these Monte-Carlo estimates, which enables the analysts to unravel the whole picture of the dynamic and interactive context.

Monte-Carlo sensitivities are difficult to compute when the performance measure function is discontinuous or angular, as well as when the underlying distribution is either singular or with intractable inverse cumulative density function. The irregularities of the performance measure function limit the application of the pathwise method, while the irregularities in the underlying distribution make the application of the likelihood ratio method implausible. For cases where the likelihood ratio method is applicable, it often yields derivative estimates with very large standard errors. To address this issue, we first considered the irregularities of the performance measure function and presented the HOPP methods for computing second-order Greeks of financial products in Chapter 3. We introduced a measure change, which removes the pathwise discontinuities of both the discounted payoff function and its first-order derivatives. The measure change is optimal in terms of minimizing the variance of the likelihood ratio part. The simulated discounted payoff function under the new scheme is $\hat{C}^2$, and the algorithmic Hessian method is used to calculate the pathwise Hessian of the discounted payoff under the new scheme. This method allows multiple discontinuities per step as well as as multiple parameters of interest. It is also important to point out that the likelihood ratio method is often applicable to financial products as the density functions are usually smooth, however, it is not applicable for the reduced-factor LMM for pricing exotic interest-rate derivatives since the underlying distribution is singular. The HOPP methods, on the other hand, are applicable to reduced-factor models. In the numerical examples, the HOPP methods are compared to the PWLR and the likelihood ratio methods under the full-factor LMM. Our numerical results suggested that the new method is significantly better than the pathwise-likelihood ratio and the pure likelihood ratio method in terms of reducing the standard error of the Hessian estimators.

In Chapter 4, we introduced a new method for computing the Hessians of exotic derivatives with early-exercise features, such as Bermudan swaptions and cancellable swaps, the HOMC method. The key to our approach is to modify the HOPP(1) method that a measure change is only performed at each exercise point if the path is near the boundary. The sequence of measure changes ensures that the first-order derivatives of the pathwise estimate of the price are Lipschitz continuous. The pathwise estimate of the price under the new scheme is $\hat{C}^2$, and the pathwise method is then applied to calculate the pathwise estimate of the Hessian. Our numerical results suggest that the HOMC algorithm outperforms the PWLR method. The exact and efficient Hessian of the price computed by HOMC would be used in practice for hedging Bermudan swaptions and cancellable swaps.

While these methods are shown to be effective for computing first- and second-order Greeks of financial products, they reply on the fact that the critical points of the payoff function passing the point of discontinuities can be obtained explicitly. We then considered the cases where this is not explicit in Chapter 5. In particular, we used the Newton-Raphson's method to numerically approximate these points. For financial product with angular payoffs, the one-step version is used; for financial products with discontinuous payoffs, the two-step version is used. We proved that the first- and second-order derivative estimates with approximated critical value functions are unbiased. The numerical examples are presented for both exotic derivative products and insurance portfolios. The results demonstrate that the computational cost of the approximation is tolerable, and the derivative estimates outperform the pathwise-likelihood ratio and the pure likelihood ratio method.

In Chapter 6, we then examined the cases where the underlying distribution does not exhibit an explicit inverse cumulative density function. We have introduced an efficient algorithm for computing unbiased estimates of first- and second-order derivatives for performance measures simulated by rejection techniques. The method is to perform a measure change at each acceptance test on the decision uniform random variable which ensures that the points of discontinuities do not move with the parameters of interest up to the third order; the change of variable function is also chosen to be optimal in terms of minimizing the variance of the likelihood ratio terms. We applied OSRS to computing sensitivities of options prices with L$\acute{e}$vy-driven underlings and the average time spent by a customer in an $M_t|M|1$ queue to demonstrate the superiority of OSRS compared to the traditional methods of estimating derivatives. We saw that is very effective in both cases.

In Chapter 7, we have also addressed some interesting problems in Risk theory for which traditional Laplace-based approaches are not feasible. In particular, we first introduced a stratified sampling method for computing finite-time ruin probabilities, for which outperforms the direct simulation method when ruin is rare. We perform changes of measure to remove discontinuities of the pathwise estimators of ruin, so the pathwise method is applicable to provide unbiased estimates of first- and second-order sensitivities of finite-time ruin probabilities. The derivative with respect to the finite-time horizon, t, provides a way of density estimation for the time to ruin random variable, especially when analytical solutions are not feasible. We further use the Newton-Raphson method to estimate the regulatory capital, and as well as its sensitivities. Numerical experiments on the classical model, the Sparre Andersen model with interest and the periodic risk model with interest, demonstrate the validity and the efficiency of the methods suggested.

We would like to emphasize the breadth of applicability of the combination of these new algorithms: they can be applied to situations in which the performance measure function is highly discontinuous, the underlying state variables have intractable distribution functions and none of the traditional methods are feasible. With mild underlying assumptions, it is clearly that these methods will have wide range of applications. Some future applications include:

- The HOMC method for computing sensitivities of the Bermudan-type products given an exercise strategy can be easily modified to compute sensitivities of Convertible bonds that both the issuer and the holder have early exercise rights.

- The sensitivities of portfolio values with respect to excess-of-loss reinsurance and dividend policy parameters can be determined. From the each sample path, one can then derive a consistent estimator of the optimal and interactive reinsurance-dividend solution.

- A natural extension of the model in Chapter 7 is to consider the ruin probabilities in discrete set-up, i.e. a fixed-time horizon is divided into equal time periods. By assuming a copula structure for the insurer's realized net profit and the discounting factor within each period, we can apply the SFRSS-alike algorithm to solve various practical problems of the insurance business.

In addition, while the numerical examples in this monograph showed that the methods developed are fast, we have not explored other methods of acceleration such as the use of parallel processing and further methodologies for variance reduction such as randomized quasi-Monte Carlo. All numerical examples are computed using single-threaded Monte Carlo C^{++} programs. The use of multi-core CPUs and graphics cards to speed up computer programs has received considerable attention. Aldrich et al (2011) demonstrated the effectiveness of GPUs for solving dynamic equilibrium problems in economics using iterative methods; Joshi (2014) demonstrated that over one hundred times speed up can be achieved in a realistic case for the pricing of cancellable swaps using the displaced diffusion LIBOR market model using a multi-core graphics card. The algorithms here are naturally parallel in nature and so multiple processing cores could be similarly employed for massive speed ups.

In summary, we have developed a generic algorithm for computing sensitivities of stochastic dynamical systems. While the method efficiently computes first- and second-order derivatives with respect to all parameter inputs, the computational cost is small. Applications were presented for the computation of sensitivities in the fields of financial mathematics, operations research and ruin theory. We believe that the new derivative estimation method developed in this monograph may also provide feasible solutions to more practical problems in the future.

Bibliography

[1] Ahrens, J.H. and Dieter, U. (1974). *Computer methods for sampling from gamma, beta, Poisson and Binomial distribution.* Computing, Vol. 12, 223 - 246.

[2] AMP. (2009). *North Investment Guarantee: Product Disclosure Statement* AMP, Australia, Retrieved 9th April 2014,<https:www.amp.com.au>.

[3] Andersen, E. S. (1957). *On the collective theory of risk in case of contagion between claims.* Bulletin of the Institute of Mathematics and its Applications, Vol. 12, 275–279.

[4] Andricopoulos, A. D. (2002). *Option pricing using quadrature and other numerical methods. Ph.D. Dissertation.* University of Manchester, Manchester, UK.

[5] Asmussen, S and Binswanger, K. (1997). *Simulation of ruin probabilities for subexponential claims.* Astin Bulletin, Vol. 27, No. 2, 297–318.

[6] Asmussen, S and Glynn, P. W. (2007). *Stochastic simulation: Algorithms and analysis.* Springer Science & Business Media.

[7] Asmussen, S and Kroese, P. D. (2006). *Improved algorithms for rare event simulation with heavy tails.* Advances in Applied Probability, Vol. 38, No.2, 545–558.

[8] Asmussen, S and Rubinstein, R. Y. (1999). *Sensitivity Analysis of Insurance Risk Model via Simulation.* Management Science, Vol. 45, Issue 8, 1125-1141.

[9] Asmussen, S and Rolski, T. (1994). *Risk theory in a periodic environment: the Cramer-Lundberg approximation and Lundberg's inequality.*Mathematics of Operations Research, Vol. 19, No. 2, 410–433.

[10] Barndorff-Nielsen, O. E. (1998). *Processes of normal inverse gaussian type.* Finance and Stochastics, Vol. 2, 41-68.

[11] Benhamou E. (2003). *Optimal Malliavin Weighting Function for the Computation of the Greeks.* Mathematical Finance, 13(1), 37-53.

[12] Black, F., and Scholes, M. (1973). *The pricing of options and corporate liabilities.* Journal of Political Economy, 637-654.

[13] Blanchet, J and Glynn, P. (2008). *Efficient rare-event simulation for the maximum of heavy-tailed random walks.* The Annals of Applied Probability, Vol. 18, No. 4,1351–1378.

[14] Blanchet, J and Lam, H. (2012). *State-dependent importance sampling for rare-event simulation: An overview and recent advances.* Surveys in Operations Research and Management Science, Vol. 17, No. 1, 38–59.

[15] Boyle, P. (1977). *Options: a Monte Carlo approach.* Journal of Financial Economics, Vol. 4, No. 4, 323–338.

[16] Brace, A. (2002). *Engineering BGM.* Chapman and Hall.

[17] Brace, A., Gatarek, D., and Musiela, M. (1997). *The market model of interest-rate dynamics.* Mathematical Finance, 7(2), 127-155.

[18] Broadie, M., and Glasserman, P. (1996). *Estimating security price derivatives using simulation.* Management science, 42(2), 269-285.

[19] Carriere, J, F. (1996). *Valuation of the early-exercise price for options using simulations and non-parametric regression.* Insurance: Mathematics and Economics, Vol.19, No.1, 19-30.

[20] Chan, J. H and Joshi, M. S. (2012). *Minimal partial proxy simulation schemes for generic and robust Monte Carlo Greeks.* The Journal of Computational Finance, Vol. 15, No. 2, 77-109.

[21] Chan, J. H and Joshi, M. S. (2013). *Fast Monte Carlo Greeks for Financial Products with Discontinuous Payoffs.*Mathematical Finance, Vol. 23, No.3, 459-495.

[22] Chan, J. H., and Joshi, M. S. (2015). *Optimal Limit Methods for Computing Sensitivities of Discontinuous Integrals Including Triggerable Derivative Securities.* IIE transactions, Vol 47, No. 9, 978–997.

[23] Cheng, R.C.H., and Feast, G.M. (1980) *Gamma variate generators with increased shape parameter range.* Communications of the ACM, Vol. 23, 389-395.

[24] Christianson, B. (1992). *Automatic Hessians by reverse accumulation.* Journal of Numerical Analysis, Vol.12(2), 135-150.

[25] Cocozza, R and Di Lorenzo, E. (2006). *Solvency of life insurance companies: methodological issues.* J. Actuarial Practice, Vol. 13, 81-101.

[26] Dean, T and Dupius, P. (2009). *Splitting for rare event simulation: A large deviation approach to design and analysis.* Stochastic processes and their applications, Vol. 119, No. 2, 562–587.

[27] Dean, T and Dupius, P. (2011). *The design and analysis of a generalized RESTART/DPR algorithm for rare event simulation.* Annals of Operations Research, Vol. 189, No. 1, 63–102.

[28] Denson, N., and Joshi, M. S. (2009). *Flaming Logs.* Wilmott Journal, 1(5), 259-262.

[29] Devroye, L. (1986). *Non-Uniform Random Variate Generation.* New York: Springer-Verlag.

[30] De Vylder, F. E. (1999). *Numerical finite time ruin probabilities by the Picard-Lefévre Formula.* Scandinavian Actuarial Journal, Vol. 2, 97-105.

[31] Dickson, D. C. M. (2007).*Some finite time ruin problems.* Annals of Actuarial Science, Vol.2, No.2, 217-232.

[32] Dickson, D. C. M. and Willmot, E. (2005). *The density of the time to ruin in the classical Poisson risk model.* Astin Bulletin, Vol.35, Issue. 1, 45-60.

[33] Dufresne. F and Gerber, H. U. (1989). *Three methods to calculate the probability of ruin.* ASTIN Bulletin, Vol. 9, No.1, 71-90.

[34] Fishman, G.S. (1996) *Monte Carlo. Concepts, Algorithms, and Applications.* New York: Springer

[35] Fries, C. (2007). *Mathematical Finance: Theory, Modeling, Implementation.* Wiley.

[36] Fournie, F. E., Lasry, J. M., Lebuchoux, J., Lions, P. L., and Touzi, N. (1999).*Applications of Malliavin calculus to Monte Carlo methods in finance.* Finance and Stochastics, 3(4), 391-412.

[37] Fu, M. (2008) *What you should know about simulation and derivatives.* Naval Research Logistics, 55(8), 723-736.

[38] Giles, M. (2008).*Vibrato Monte Carlo sensitivities.* In Monte Carlo and Quasi-Monte Carlo Methods 2008. Springer, New York.

[39] Giles, M., and Glasserman, P. (2006). *Smoking adjoints: fast Monte Carlo Greeks.* RISK, 19(1), 88-92.

[40] Glasserman, P. (1991). *Structural conditions for perturbation analysis derivative estimation: Finite-time performance indices.* Operations Research, Vol. 39(5), 724-738.

[41] Glasserman, P. (1992), *Smoothing complements and randomized score functions,* Annals of Operations Research, 39, 41–67

[42] Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering.* Springer, New York.

[43] Glasserman, P., Heidelberger, P and Shahabuddin, P. (1999). *Asymptotically Optimal Importance Sampling and Stratification for Pricing Path-Dependent Options.* Mathematical Finance, Vol. 9, No. 2, 117–152.

[44] Glasserman, P., Heidelberger, P., Shahabuddin, P. and Zajic, T. (1999). *Multilevel splitting for estimating rare event probabilities.* Operations Research, Vol. 47, No. 4, 585–600.

[45] Glasserman, P., and Liu, Z.(2011).*Estimating Greeks in simulating Lévy-driven models.* The Journal of Computational Finance, Vol.14(2), 3-56.

[46] Griewank, A., and Walther, B. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM

[47] Hardy, M. R. (1993). *Stochastic simulation in life office solvency assessment.* Journal of the Institute of Actuaries, Vol. 120, No. 1, 131-151.

[48] Heath, D., Jarrow, R and Morton, A. (1992). *Bond pricing and the term structure of interest rates: a new methodology for contingent claims valuation.* Econometrica, Vol. 60, 77–105.

[49] Heidergott, B., Pflug, G., and Farenhorst-Yuan, T. (2010). *Gradient estimation for discrete-event systems by measure-valued differentiation.* ACM Transactions on Modeling and Computer Simulation (TOMACS), 20(1), 5.

[50] Henrard, M. (2014). *Interest Rate Modelling in the Multi-Curve Framework: Foundations, Evolution and Implementation.* Palgrave Macmillan.

[51] Hobbs, C., Krishnaraj, B., Liu, Y., and Musselman, J. (2009) *Calculation of variable annuity market sensitivities using a pathwise methodology.* Life & Pensions, September 2009.

[52] Hong, L. J., and Liu, G. (2010). *Pathwise estimation of probability sensitivities through terminating or steady-state simulations.* Operations Research, 58(2), 357-370.

[53] Hong, L. J., and Liu, G. (2011). *Kernel estimation of the Greeks for options with discontinuous payoffs,* Operations Research, 59(1), 96–108.

[54] Jamshidian, F. (1997). *LIBOR and swap market models and measures.* Finance and Stochastics, Vol.1, 293–330.

[55] Joshi, M. S.(2003). *Rapid computation of drifts in a reduced factor LIBOR market model.* Wilmott Magazine, 5, 84-85.

[56] Joshi, M.S. (2011). *More mathematical finance.* Pilot Whale Press.

[57] Joshi, M.S. (2014). *Kooderive: Multi-core graphics cards, the libor market model, least-squares monte carlo and the pricing of cancellable swaps.* Least-Squares Monte Carlo and the Pricing of Can- cellable Swaps (January 30, 2014).

[58] Joshi, M. S. and Kainth, D. S. (2003). *Rapid and accurate development of prices and Greeks for nth to default credit swaps in the Li model.* Quantitative Finance, Vol. 3, Issue. 6, 458-469.

[59] Joshi, M. S., and Pitt, D. (2010). *Fast Sensitivity Computations for Monte Carlo Valuation of Pension Funds.* Astin Bulletin, 40(2), 655 - 667.

[60] Joshi, M. S., and Yang, C. (2011). *Algorithmic Hessians and the fast computation of cross-gamma risk.* IIE Transactions, 43(12),878-892.

[61] Joshi, M. S. and Zhu, D. (2014a). *Optimal Partial Proxy Method for Computing Gammas of Financial Products with Discontinuous and Angular Payoff.* Available at SSRN 2431580.

[62] Joshi, M. S. and Zhu, D. (2014b). *An Exact Method for Sensitivity Analysis of Systems Simulated by Rejection Techniques.* Available at SSRN 2488376.

[63] Joshi, M. S. and Zhu, D. (2016a). *The efficient computation and sensitivity analysis of finite-time ruin probabilities and the estimation of risk-based regulatory capital.* To appear in ASTIN Bulletin.

[64] Joshi, M. S. and Zhu, D. (2016b). *An Exact Method for Sensitivity Analysis of Systems Simulated by Rejection Techniques.* To appear in European Journal of Operational Research.

[65] Jouandeau, N and Cazenave, T. *Monte-Carlo Tree Reductions for Stochastic Games.* Technologies and Applications of Artificial Intelligence,228–238.

[66] Khinchin, A. Y. (1967). *The mathematical theory of a stationary queue.* DTIC Document.

[67] Kinderman, A. J., and Monahan, J. F. (1977). *Computer Generation of Random Variates Using Ratio of Uniform Deviates.* ACM Transaction in Mathematical Software, Vol. 3, 257-260.

[68] Konstantinides, D., Tang, Q. H and Tsitsiashvili, G. (2002). *Estimates for the ruin probability in the classical risk model with constant interest force in the presence of heavy tails.* Insurance: Mathematics and Economics, Vol. 31, No. 3, 447-460.

[69] Korn, R and Liang, Q. (2013). *Robust and accurate monte carlo simulation of (cross-) gammas for bermudan swaptions in the libor market model.* Journal of Computational Finance, Vol.17, No.3, 87-110.

[70] Korn, R and Liang, Q. (2014). *Erratum: Robust and accurate monte carlo simulation of (cross-) gammas for bermudan swaptions in the libor market model.* Journal of Computational Finance, Vol.18, No.3, 129-133. .

[71] Kreyszig,E. (1988). *Advanced Engineering Mathematics* (6th ed.). John Wiley & Sons.

[72] L'Ecuyer, P. (1990). *A unified view of the IPA, SF, and LR gradient estimation techniques.* Management Science, 36(11), 1364-1383.

[73] Lee, T. (2013). *Pricing Interest RateDerivatives and Computing Pathwise Greeks in the Extended LIBOR Market Model.* Available at SSRN 2366231.

[74] Lewis, P. A. W. (1972). *Recent Results in Statistical Analysis of Univeriate Point Processes.* Statistical Point Processes. New York, John Wiley & sons, 1-54.

[75] Lewis, P. A., and Shedler, G. S. (1979). *Simulation of nonhomogeneous Poisson processes by thinning.* Naval Research Logistics Quarterly, 26(3), 403-413.

[76] Longstaff, F.A.and Schwartz, E.S. (2001).*Valuing american options by simulation: A simple least-squares approach.* Review of Financial studies, Vol.14, No.1, 113-147.

[77] Loisel, S and Privault ,N. (2009). *Sensitivities analysis and density estimation for finite-time ruin probabilities.* Journal of Computational and Applied Mathematics, Vol. 230, 107-120.

[78] Cizek, P., Härdle, W. K. and Weron, R. (2011). *Stochastic tools for Finance and Insurance.* Springer.

[79] Madan, D. B., and Seneta, E. (1990). *The Variance-Gamma (V.G.) model for share market returns.* Journal of Business, Vol. 63, 511-524.

[80] Madan, D., Carr, P., and Chang, E. (1998). *The variance gamma process and option pricing.* European Finance Review, Vol. 2, 79-105

[81] Malliavin, P. (1991), *Stochastic analysis.* Springer-Verlag.

[82] Mercurio,F. (2010). *Modern libor market models: using different curves for projecting rates and for discounting.* International Journal of Theoretical and Applied Finance, Vol.13, No.1,113-137.

[83] Michael, J.R., Schucany, W. R., and Haas, R. W. (1976).*Generating Random Variates Using Transformations with Multiple Roots.* The American Statistician, Vol. 30(2), 88-90

[84] Morales, M. (2004). *On a surplus process under a periodic environment: a simulation approach.* North American Actuarial Journal, Vol. 8, No. 4, 76–89.

[85] Neumann, J. V. (1951). *Various techniques used in connection with random digits. Monte Carlo methods,* Nat. Bureau Standards, Vol.12, 36-38.

[86] Pflug, G. (1996). *Optimisation of Stochastic Models.* Kluwer Academic, Boston, USA.

[87] Picard, P and Lefévre, C. (1997). *The probability of ruin in finite time with discrete claim size distribtuion.* Scandinavian Actuarial Journal, Vol. 1, 58-69.

[88] Pietersz, R., Pelsser, A and Regenmortel, M.V. (2004). *Fast drift-approximated pricing in the BGM model.* Journal of Computational Finance, Vol. 8, 93-124.

[89] Piterbarg, V. V. (2004). *TARNs: models,valuation, risk sensitivities.* Wilmott Magazine, 14, 62-71.

[90] Piterbarg, V. V. (2004). *Computing deltas of callable libor exotics in forward libor models. Journal of Computational Finance,* Vol.3:107-144.

[91] Pollaczek, F. (1930). *Über eine Aufgabe der Wahrscheinlichkeitstheorie. I.* Mathematische Zeitschrift, Vol.32, No. 1, 64–100.

[92] Ponssard, J.P and Sorin, S. (1980). *The LP formulation of finite zero-sum games with incomplete information.* International Journal of Game Theory, Vol. 9, No. 2, 99–105.

[93] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge university press.

[94] Privault, N. and Wei, X. (2004).*A Malliavin Calculus approach to Sensitivity Analysis in Insurance.* Insurance: Mathematics and Economics, Vol. 35, 679-690.

[95] Robbins, H. and Monro, S. (1951). *A stochastic approximation method.* The annals of mathematical statistics, 400-407.

[96] Rubinstein, R. (1992). *Sensitivity analysis of discrete event systems by the score function method.* Annals of Operations Research, Vol. 39, 229-250.

[97] Rogers, L. C. G. (2007). *Pathwise stochastic optimal control.* SIAM Journal on Control and Optimization, Vol. 46, No. 3, 1116–1132.

[98] Rolski,T., Schmidil, H., Schmidt, V and Teugels.J. (1999).*Stochastic Processes for Insurance and Finance.* John Wiley & Sons. West Sussex, England.

[99] Ross, R. (2006). *Simulation.* Elsevier, Burlington, MA.

[100] Rubino, G and Tuffin, B. (2009). *Rare event simulation using Monte-Carlo methods.* John Wiley & Sons Ltd, New York.

[101] Sato, K. (1999).*Lévy Processes and Infinitely Divisible Distributions.* Cambridge University Press.

[102] Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivatives.* Wiley.

[103] Suhasini, A. V. S., Rao, K. S., and Reddy, P. R. S.(2013) *On parallel and series non homogeneous bulk arrival queueing model.* OPSEARCH, Vol. 50(4),521-548.

[104] Sundt, B. and Teugels, L. (2995). *Ruin estimates under interest force.* Journal of Applied Mathematics and Stastistics, Vol.16, 7-22.

[105] Tanizaki, H. (2008). *A Simple Gamma Random Number Generator for Arbitrary Shape Parameters.* Economics Bulletin, Vol. 3(7), 1-10.

[106] Teugels, J. L and Vynckier. P. (1996). *The structure distribution in a mixed Poisson process.* Journal of Applied Mathematics and Stastistics, Vol. 9,489-496.

[107] Vasicek, O. (1977). *An equilibrium characterization of the term structure.* Journal of Financial Economics,Vol. 5, 177–188.

[108] Vazquez-Abad, F. J. (2000). *RPA pathwise derivative estimation of ruin probabilities.* Insurance: Mathematics and Economics, Vol. 26, No. 2, 269-288.

[109] Villen-Altamirano, M and Villen-Atamirano, J. (1991). *RESTART: a method for accelerating rare event simulations.* Queueing, Performance and Control in ATM, J. W. Cohen and C. D. Pack, Eds. Amsterdam, The Netherlands: Elsevier Sci, 7176.

[110] Volk-Makarewicz, W. (2014), Advances in Derivative Estimation: Ranked Data, Quantiles, and Options, Thesis Vrije Universiteit Amsterdam.

[111] Wang, Y., Fu, M. C., and Marcus, S. I.(2012). *A new stochastic derivative estimator for discontinuous payoff functions with application to financial derivatives.* Operations Research, 60(2), 447-460.

[112] Wilmott, P., Dewynne, J. and Howison, S., (1993). *Option pricing: mathematical models and computation.* Oxford Financial Press.