Joint Distribution of Passage Times of an Ornstein-Uhlenbeck Diffusion and Real-Time Computational Methods for Financial Sensitivities

Yupeng Jiang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London

Department of Mathematics

University College London

October 13, 2019

I, Yupeng Jiang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis analyses two broad problems: the computation of financial sensitivities, which is a computationally expensive exercise, and the evaluation of barrier-crossing probabilities which cannot be approximated to reach a certain precision in certain circumstances. In the former case, we consider the computation of the parameter sensitivities of large portfolios and also valuation adjustments. The traditional approach to compute sensitivities is by the finite-difference approximation method, which requires an iterated implementation of the original valuation function. This leads to substantial computational costs, no matter whether the valuation was implemented via numerical partial differential equation methods or Monte Carlo simulations. However, we show that the adjoint algorithmic differentiation algorithm can be utilised to calculate these price sensitivities reliably and orders of magnitude faster compared to standard finite-difference approaches. In the latter case, we consider barrier-crossing problems of Ornstein-Uhlenbeck diffusions. Especially in the case where the barrier is difficult to reach, the problem turns into a rare event occurrence approximation problem. We prove that it cannot be estimated accurately and robustly with direct Monte Carlo methods because of the irremovable bias and Monte Carlo error. Instead, we adopt a partial differential equation method alongside the eigenfunction expansion, from which we are able to calculate the distribution and the survival functions for the maxima of a homogeneous Ornstein-Uhlenbeck process in a single interval. By the conditional independence property of Markov processes, the results can be further extended to inhomogeneous cases and multiple period barrier-crossing problems, both of which can be efficiently implemented by quadrature and Monte Carlo integration methods.

Impact Statement

This thesis analyses and solves two problems. The first problem pertains to the evaluation of sensitivities of financial products priced by numerically solving partial differential equations or regression-based Monte Carlo methods. The second problem concerns the multiple barrier-crossing probability of Ornstein-Uhlenbeck diffusions. We now summarise their impacts to audiences inside and outside of academia as follows.

In financial industry, finding the price of a financial derivative is only a start. More importantly, one needs to consider their position's exposure, whose risks should be managed by hedging based on price sensitivities to the risk factors of the underlying assets of the derivative. Although one may able to obtain a closed-form price function for special contingent claims under specific model assumptions, generally, one has to consider numerical methods for derivatives pricing, such as partial differential equation approaches or Monte Carlo methods. The traditional way to evaluate financial sensitivities is by the finite-difference approach, in which one needs to bump the value of a risk factor and re-compute the price again. This becomes time-consuming if the number of risk factors is large due to the nested implementation of the numerical pricing function. In this thesis, we apply the adjoint algorithmic differentiation method to compute the financial sensitivities. The adjoint algorithmic differentiation method reversely collects the linear combinations of the Jacobian matrices for the pricing function. Therefore, its theoretical time-consumption of evaluating all the sensitivities will not exceed four times the time-consumption of the pricing function. With the adjoint algorithmic differentiation method, one can evaluate the financial sensitivities efficiently in practice. This can

not only save the computational and technological costs for financial institutions, but can also improve the efficiency of derivatives hedging and risk management.

The other problem we consider in this thesis is the multiple barrier-crossing probabilities of Ornstein-Uhlenbeck diffusions. Ornstein-Uhlenbeck processes have been widely used in physics, finance and other fields. The main theoretical impact we provide is the semi-analytical formula for the joint distribution of passage times for a continuous Markov process in consecutive time intervals. It can not only be used in the Ornstein-Uhlenbeck diffusions setting, but can also be further applied in the context of general continuous Markov processes. One natural extension is to find the joint distributions of passage times for other known processes in consecutive intervals. Practically, one can use this result to model events with joint-passaging features. A typical example concerns the modelling of a heatwave. A heatwave is defined as period of at least several consecutive days during which the maximum temperature exceeds certain thresholds.

In sum, this thesis provides theoretically- and practically-useful mathematical theories. With these results, one can either boost the computational efficiency in risk management exercises in financial industry or study the multiple barrier-crossing probabilities for extensive group of stochastic processes.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my primary supervisor Dr Andrea Macrina for the continuous support and the excellent supervision of my PhD study and research. His knowledge and experience helped me throughout my research and write-up of this thesis. As an overseas student, I am also grateful for his care during the entire period of my study. I appreciate his guidance in my study, work and life.

Besides my supervisor, I would like to thank my secondary supervisor and all my research collaborators: Dr Marc Henrard, Dr Luca Capriotti and Prof. Gareth W. Peters, for their expertise which they shared with me, but also their insightful comments and encouragement. I enjoyed all the moments when I was working with them.

I would like to thank my two PhD examiners, Prof. Christoph Reisinger and Dr Daniel Schwarz, for sharing with me comments and suggestions so I could improve my thesis.

I am grateful to Prof. Helen Wilson, Prof. Robb McDonald, Prof. Valery Smyshlyaev, Prof. Dmitri Vassiliev, Prof. Carlo Marinelli, Dr Mark Roberts, Dr Camilo Garcia Trillos and Dr Hao Ni for always being available to offer help in both my study and research. Withstanding the above, I am also grateful to the department for offering me the Teaching Assistantship and Dean's Prize to cover my study expenses. I would like to thank the administrative staff of Department of Mathematics at UCL for all their support.

I appreciate all the help offered by the staff from the institutions and workshops I visited. I deeply appreciated the opportunities for me to attend the Zürich Spring

School on Lévy Processes at ETH, the Financial Mathematics Team Challenge at University of Cape Town, the 9th European Summer School in Financial Mathematics in St. Petersburg, the Osaka-UCL Workshop on Stochastics, Numerics and Risk at Osaka University, the 17th Winter school on Mathematical Finance in Lunteren, and finally the Fourth Young Researchers Meeting on BSDEs, Nonlinear Expectations and Mathematical Finance at Shanghai Jiao Tong University. In particular, I would like to thank Prof. David Taylor from the University of Cape Town, Prof. Jun Sekine from Osaka University, Prof. Matheus Grasselli from McMaster University and Prof. Stéphane Crépey from the University of Evry.

I also want to thank Morgan Stanley for offering me the internship on electronic foreign exchange quantitative strategy during my PhD. This gave me the chance to put what I have learnt into practice. My sincere thanks goes to Dr Xiaxi Huang and Dr Maria Alessandra Crisafi for their guidance and support during the whole period of my internship. I confirm that no data from Morgan Stanley has been used at any point of this thesis.

My sincere thanks also go to my friends Yanlong Fang, Chunxin Yuan, Tao Gao, Binbin Xue, Longjie Jia, Luting Li, Kangjianan Xie, Jiaqi Liang, Weiguan Wang, Zihua Liu, Deyu Ming, Shujian Liao, Francois Buet-Golfouse, Holly Brannelly, Piotr Kowalczyk, Mario Giuricich, Xuecan Cui, Qingyun Yang, Hongtao Xu, Yu Zheng, Yuan Miao and Bingdi Fan for their continuous support and encouragement. A PhD can be a lonely journey, but it is because of you all that my journey was worthwhile.

Last but not the least, I would like to thank my parents for spiritually supporting me all the time.

Contents

List of Figures

List of Tables

Abbreviations

AAD: adjoint algorithmic differentiation

AD: algorithmic differentiation

BK: Black-Karasinski

CDF: cumulative density function

CDS: credit default swap

CN: Crank-Nicolson

CVA: credit valuation adjustment

DNN: deep neural network

DVA: debit valuation adjustment

FD: finite-difference

FPT: first-passage-time

GPU: graphics processing unit

IFT: implicit function theorem

MC: Monte Carlo

ODE: ordinary differential equation

OTC: over-the-counter

OU: Ornstein-Uhlenbeck

PDE: partial differential equation

PDF: probability density function

RWR: right-way risk

WWR: wrong-way risk

XVA: x valuation adjustments

Chapter 1

Introduction

Increases in computational power have been witnessed since the start of the 21st century. However, computational problems have not become fewer, despite the improvement in computational technology. In fact, due to the advancements in computational power, problems that may have been computationally expensive or time-consuming to solve in the past have been brought back. The recent rise of machine learning is one of such examples: most of the machine learning algorithms were invented in the late 20th century, but it was not until recently that they drew everyone's attention. Moreover, alongside the development of computational power, the demand for quantitative analysis and the complexity of mathematical models also grew. For example, the financial world has changed after the financial crisis of 2007-2008. In the past, financial products were complicated under a relatively simple financial environment. However, the financial crisis of 2007-2008 brought more regulations to trades and transactions. Although these regulations reduced the percentage of complicated financial products in the market, they also made the pricing of simple financial products difficult. This has become especially true in the aftermath of the financial crisis because of the renewed emphasis on risk management and the introduction of a large number of valuation adjustments for financial derivatives prices, collectively known as x valuation adjustments (XVA), see for example [10] and [31]. In actuarial science, computational needs have also recently increased. In casualty insurance, ruin probabilities of some rare events, such as the occurrences of natural disasters or political hazards, are usually computed or

approximated. This requires extensive computational power due to the small occurrence likelihoods. Indeed, it seems reasonable to suggest that regardless of the rising computational power, the computational challenges we are facing have not decreased.

Nowadays, financial technology (FinTech) has become a popular word in industry. Artificial intelligence, Graphics Processing Units (GPU) computing, blockchain and many other tools have been used in financial applications. The applications of these new technologies can make the industry more efficient in many ways, however, they are not without risk. Actually, apart from the risks inside these emerging tools, their abuse may lead to computational inefficiencies in problem solving, compared with using techniques relying on traditional mathematical analysis. One such example is the computation of XVA. XVA aims at capturing counterparty risk and other funding as well as capital costs that significantly affect the profitability of trading operations. XVA is computationally expensive to obtain because it involves computing the expectation of a function together with all the conditional exposures of the transaction until expiry. This is a typical nested Monte Carlo (MC) problem. One can solve it naively with modern technologies by resorting to GPU computations. However, as shown in [60] and [71], the nested MC algorithm can be approximated efficiently with a regression-based MC scheme. The regression-based MC algorithm requires significantly less computational cost compared with the nested MC method, even when the latter is implemented with GPU clusters. Mathematics can not only bring efficiency in quantitative analysis, but can also overcome problems which can hardly be solved by FinTech methods alone. For instance, machine learning methods have been widely used in the insurance industry. For specific insurance contracts, machine learning "blackboxes" are trained with historical data samples, comprising hazard factors, in order to show the insurance company the risk profiles of their clients. However, for contracts protecting the insured from natural disasters, such as strong earthquakes or tsunamis, historical occurrences samples are so few that the supervised training of machine learning algorithms can be heavily biased. This may lead to inaccurate pricing of

insurance products or cause further problems such as insolvency.

Within the context specified above, this thesis focuses on applying advanced mathematical tools to solve complex problems in efficient and accurate ways. We summarise the main contributions of this thesis as follows:

1) We apply adjoint algorithmic differentiation (AAD) to the risk management of derivative securities in the situation where the dynamics of securities prices are given in terms of partial differential equations (PDEs). Through the use of simple examples, we show how AAD can be applied to both forward and backward PDEs in a straightforward manner. In particular, in the context of one-factor short-rate models for interest rates or default intensity processes, we show how one can compute price sensitivities reliably and orders of magnitude faster in AAD than with a standard finite-difference approach. This notable increased efficiency is obtained by combining (i) the adjoint of a forward PDE for calibrating the parameters of the model, (ii) the adjoint of a backward PDE for pricing the derivative security, and (iii) the implicit function theorem to avoid iterating the calibration procedure.

2) We show how AAD can be used to calculate price sensitivities in regression-based MC methods (also known as the least-square MC approach) reliably and orders of magnitude faster than with standard finite-difference approaches. In doing so, we introduce the AAD version of the traditional least-square algorithms of [60] and [71]. By discussing detailed examples which are of practical relevance, we demonstrate how accounting for the contributions associated with the algorithm's regression functions is crucial in order to obtain accurate estimates of the *Greeks*, especially in XVA applications.

3) We investigate the joint distribution and the multivariate survival functions for the maxima of an Ornstein-Uhlenbeck (OU) process in consecutive time-intervals. A PDE method with eigenfunction expansion is adopted, with which we firstly calculate the distribution and the survival functions for the maximum of a homogeneous OU-process in a single interval. By using a

deterministic time-change and a parameter translation, this result can be extended to an inhomogeneous OU-process. Secondly, we derive a general formula for the joint distribution and the survival functions for the maxima of a continuous Markov process over consecutive periods. With these results, one can obtain semi-analytical expressions for the joint distribution and the multivariate survival functions for the maxima of an OU-process, with piecewise constant parameter functions, over consecutive time periods. We show that the joint distribution and the survival functions can be evaluated numerically by an iterated quadrature scheme, which can be implemented efficiently via simple matrix multiplications. Moreover, we show that the computation can be further simplified to the product of single quadratures if the filtration is enlarged.

The content of this thesis is adapted from the following research papers: (i) Chapter 2 is based on Capriotti, Jiang and Macrina (2015), henceforth [21]; (ii) Chapter 3 is based on Capriotti, Jiang and Macrina (2016), henceforth [22]; and (iii) Chapter 4 is based on Jiang, Macrina and Peters (2019), henceforth [49]. In particular, Chapter 2 and 3 introduce the AAD algorithm, which can be used to evaluate financial sensitivities, such as the *Greeks*, efficiently and robustly. These chapters cover the applications of AAD to two major numerical methods: the numerical PDE approach and the MC method. Chapter 4 studies the multiple barrier-crossings of OU-diffusions in consecutive periods. A semi-analytical formula for the joint distribution and the multivariate survival functions for the maxima of an OU-process in consecutive time-intervals is achieved. The key message of this thesis is to show that with mathematical analysis, new robust and efficient algorithms for computationally expensive problems and the semi-closed-form expressions for theoretically difficult problems can be developed. We now consider the structure of this thesis in further detail.

Chapter 2 applies the AAD algorithm to the risk management of structural products, such as corporate bonds, bond options, credit default swaps (CDSs) and CDS options, when their prices may be characterized by PDEs. We show how AAD

can be applied to forward and backward PDEs in a straightforward manner. In the context of one-factor models for interest rates or default intensities, we show how price sensitivities are computed reliably and orders of magnitude faster than with a standard finite-difference approach. Specifically, in Section 2.2 we begin by recalling the standard formality for the valuation of securities prices by means of PDEs. We also recall the standard numerical approach for the numerical solution of backward PDEs. In Section 2.3, we review the equivalent forward PDE approach and its use for the efficient implementation of calibration algorithms. Section 2.4 presents the general principles of AAD and their application in numerical solution of the forward and backward PDEs. The results of numerical experiments are presented in Section 2.5 and conclusions are shown in Section 2.6.

In Chapter 3, we present the application of AAD to the regression-based MC approach, which is widely applied for the pricing and risk management of Bermudan-style options (see [24], [71], [60]), and in particular for XVA applications (see [25]). We first develop the AAD implementation of the well-known least-square part of the algorithm for the computation of conditional expectation; then we investigate numerically the impact on the *Greeks* arising from the sensitivities of the regression functions. In Section 3.2, the regression-based MC algorithms for both Bermudan-style options and XVA are presented. In Section 3.3, we discuss the AAD algorithms for the regression-based MC method. Here, we also present results on handling discontinuities within path-wise differentiations. We give two numerical examples: the *best of two stocks Bermudan-style call option* and its corresponding XVA in Section 3.4. We show how smoothening discontinuities associated with suboptimal exercise boundaries improves the accuracy of the *Greeks* of Bermudan-style options, and why the contribution to the sensitivities arising from the regression boundaries is essential for an accurate computation of XVA sensitivities. The efficiency and accuracy of AAD are also compared with the traditional finite difference approach.

Up to this point, the research has focussed on applications of the AAD approach to financial derivatives in a real-time setting. The goal of Chapter 4 is to de-

velop the mathematical theory to calculate the probability of OU-diffusions crossing multiple upper barriers in consecutive periods, especially when the barrier-crossing is a seldom event. Firstly, we study the first-passage-time (FPT) of a homogeneous OU-process with lower reflection barrier passing a constant upper barrier with a PDE method in Section 4.2. Moreover in Section 4.3, we transform the FPT of an inhomogeneous OU-process passing a time-dependent barrier to the FPT of a homogeneous OU-process with a different time-dependent barrier. The time-inhomogeneity from the process can be transferred to the time-dependent barrier only. Furthermore, an integral representation of the joint distribution and joint survival function for the maxima of a continuous Markov process in consecutive intervals is presented in Section 4.4. Finally, we show that the numerical integration can be computed efficiently and accurately by the quadrature scheme and the MC integration method in Section 4.5.

In concluding this introduction, we offer a few remarks to spur reflections on the relationship between mathematical analysis and cutting-edge modern technologies. The increasing attention drawn from the fast developing technologies makes many individuals think that these technologies will finally end the need for quantitative analysis in practical settings, especially in the financial industry. However, without a clear understanding, the abuse of these technologies may lead to untoward consequences. This is especially true in abusing machine learning when applied to the analysis of financial data. Compared with medical, graphical and natural data, financial data is strongly influenced by news, the behaviours of market participants and many other aspects. The patterns in financial data are hidden behind significant noise. Therefore, decisions based on machine learning algorithms trained with noisy and dirty data can be very risky.

Nevertheless, one of the most important messages we wish to transmit with this thesis is: *mathematics is an essential part in financial modelling, risk management and actuarial science, regardless of new upcoming technologies such as artificial intelligence or in general FinTech*. In fact, many aspects of these modern technologies are based on mathematics themselves. For example, deep learning has become

one of the most spoken-of topics after the success of *AlphaGo* from *DeepMind*, Google. Google has also introduced TensorFlow, a library based on GPU parallel computing to accelerate deep neural network (DNN) training by back-propagation. The core of TensorFlow in evaluating gradients is actually based on the same mathematical methodology as AAD, see Section 2.4. Through the use of mathematical theories, modern technologies can be further improved in terms of accuracy and efficiency. Conversely, modern technologies may affect ways of problem solving in applied mathematics. In another research project (see [23]) excluded from this thesis, we approximate solutions to non-linear PDEs by using a DNN based on the universal approximation theorem. Since the DNN estimator is infinitely differentiable with respect to all the input arguments, the problem of solving the PDE can be transformed to an optimisation problem, where the solution to the PDE is approximated by minimising the residuals. Such methods give rise to more options in solving mathematical problems.

Chapter 2

Real-time risk management: An AAD-PDE approach

2.1 Overview

The practice of risk management of derivative securities comes with a high computational burden and technology cost. Standard approaches for the calculation of risk require repeated portfolio valuation under hundreds of market scenarios. As a result, in order to complete risk calculations in practical time-spans, financial institutions employ vast computational resources bearing high infrastructure costs. Since the total cost of "through-the-life" risk management can determine whether it is profitable to execute a new trade, solving such a technological challenge is of paramount importance for a financial institution to remain competitive.

Against this backdrop, a computational technique known as AAD was recently introduced in [17], [19], [20] and [44]. It has been proven to be effective for speeding up the calculation of sensitivities, especially for MC applications, see [16] and [18] for example. This powerful technique allows for fast computation of first-order sensitivities without repeating the portfolio valuation several times, as happens in traditional finite-difference approaches.

Algorithmic differentiation (AD), see [42], is a scheme for the efficient calculation of derivatives of functions, which is implemented as a computer programme. What makes AD particularly attractive, when compared to standard finite-difference

methods for the calculation of derivatives, is its computational efficiency. AD exploits the information on the structure of the computer code in order to optimise the calculation. In particular, when derivatives of a small number of outputs with respect to a large number of inputs are required, the calculation can be optimised by applying the standard chain rule of calculus through the instructions of the programme in opposite order to their original evaluation. This gives rise to the adjoint algorithmic differentiation (AAD), which means the adverse mode of AD, see [37].

Most of the applications considered in the financial literature so far have focused on MC simulations, such as [16], [17], [18], [19] and [38]. In that context, AAD can be used to efficiently implement the so-called Pathwise Derivative Method for the calculation of sensitivities, see [12]. In this chapter, we extend this research area and demonstrate how AAD can be effective in applications to PDEs. In asset pricing, the PDE method is widely used to compute the price of low dimensional and non-path-dependent derivatives. This can be achieved either by solving a backward PDE (the PDE with terminal condition) for the derivative price directly, or by obtaining the Arrow-Debreu price from a forward PDE (the PDE with initial condition) first and then computing the price by numerical quadrature. In this chapter, we show how AAD can be utilised to speed up the calculation of the sensitivities in situations where the pricing of the derivative and the calibration of the underlying stochastic model rely on solving PDE. We further show how one can compute price sensitivities more reliably and in orders of magnitude faster than standard finite-difference approaches. This is achieved with a judicious combination of the adjoint version of the numerical schemes for forward and backward PDEs and by the implicit function theorem. We provide step-by-step instructions for the AAD-PDE scheme to facilitate its implementation in "real-time" financial risk management.

The two main contributions of this chapter are as follows: a) As far as we are aware, this is the first time that AAD algorithm has been paired with numerical PDE methods to evaluate sensitivities of financial derivatives. The financial industry may have been applying such techniques already. This chapter fills the gap between theory and practice. b) By the implicit function theorem, we show how sensitivities

from the calibration function with numerical PDE pricer can be computed magnitude faster than the traditional finite-difference estimator or the direct application of AAD to the calibration function. This further improves the efficiency of risk management in practice.

2.2 Option prices and backward PDEs

Option pricing problems can be often formulated in terms of a linear parabolic partial differential equation, of the second order, of the form

$$\frac{\partial V}{\partial t} + \mu(x,t;\theta)\frac{\partial V}{\partial x} + \frac{1}{2}\sigma^2(x,t;\theta)\frac{\partial^2 V}{\partial x^2} - \nu(x,t;\theta)V = 0, \tag{2.2.1}$$

where

$$V_t(\theta) = V(x_t,t;\theta) \equiv \mathbb{E}\left[\exp\left(-\int_t^T \nu(x_u,u;\theta)\mathrm{d}u\right)P(x_T;\theta)\,\middle|\,x_t\right] \tag{2.2.2}$$

is the value of a derivative contract at time t. Here and from now on, $\theta = (\theta_1,\ldots,\theta_{N_\theta})$ represents the vector of N_θ parameters on which the model depends. Unless further specification, the expectations in later context are taken under the risk neutral probability measure, under which the discounted asset prices are martingales. At the maturity date $T > t$, the value, $P(x_T;\theta)$, of the financial derivative depends on the realisation of the risk factor $(x_t)_{t\geq 0}$ that satisfies

$$\mathrm{d}x_t = \mu(x_t,t;\theta)\mathrm{d}t + \sigma(x_t,t;\theta)\mathrm{d}W_t, \tag{2.2.3}$$

where $\mu(x,t;\theta)$ and $\sigma(x,t;\theta)$ are the drift and the volatility functions, and $(W_t)_{t\geq 0}$ is a one-dimensional Brownian motion. By supplying appropriate spatial boundary conditions, see page 159 of [75], and the terminal condition $V(x,T;\theta) = P(x;\theta)$ at maturity T, Equation (2.2.1) can be solved backwards in time for the value of the derivative security at any time $t \leq T$, $V(x,t;\theta)$ for $x_t = x$.

Note that the Black-Scholes PDE for the price of European-style claims is of the form (2.2.1), where $\mu(x,t;\theta) = (r(t)-\delta(t))x$, $\sigma(x,t;\theta) = \sigma(t)x$ and

$\nu(x,t;\theta) = r(t)$. Here $r(t)$ and $\delta(t)$ denote the risk-free interest rate and dividend yield respectively.

One-factor short-rate models for applications to interest-rate derivatives pricing, such as the models by [9], [29] or by [47] can also be described in terms of a random driver that satisfies a diffusion of the form (2.2.3). For example, the Black-Karasinski (BK) model can be expressed by setting the stochastic instantaneous rate of interest to $r_t \equiv r(x_t) = \exp(x_t)$. Here, $(x_t)_{t\geq 0}$ satisfies

$$\mathrm{d}x_t = \kappa(t)\left(\mu(t) - x_t\right)\mathrm{d}t + \sigma(t)\mathrm{d}W_t, \tag{2.2.4}$$

where $\kappa(t)$ and $\mu(t)$ are the mean-reversion speed and level respectively. In the context of short-rate models, the value of a derivative security with expiry value $P(r_T;\theta)$ can be expressed as in (2.2.2) with $\nu(x_u,u;\theta) = r(x_u)$ and where the expectation is taken under the risk-neutral measure. In this case the components of the vector θ are typically the coefficients used to parameterise the mean-reversion speed and level as well as the volatility of the process.

Since the BK-model is a positive stochastic process, it can be applied to the modelling of stochastic default intensities, which is known as the Cox processes, see page 46 of [61]. In this context, we take $x_t = \ln(h_t)$ in Equation (2.2.4), the logarithm of the *hazard rate* $(h_t)_{t\geq 0}$, and it represents the default probability per unit of time (between times t and $t+dt$) for a reference entity under the risk-neutral measure, conditional on the survival up to time t. By modelling the default event of an obligor as the first arrival time, τ, of a Poisson process, the conditional (risk-neutral) probability of the obligor surviving up to time T is given by

$$Q(h_t,t,T) = \mathbb{E}\left[\exp\left(-\int_t^T h_u\,\mathrm{d}u\right)\Big|\, h_t, \tau > t\right]. \tag{2.2.5}$$

Any credit derivative for which its payoff at time T is a function of the hazard rate, h_T (such as defaultable bonds, CDS, bond options and CDS options) can be priced by using the backward PDE (2.2.1).

2.2.1 Numerical solutions to backward PDEs by finite-difference discretisation

The solution, $V_{t_0}(\theta) = V(x_{t_0}, t_0; \theta)$, of the PDE (2.2.1) can be found numerically by discretisation on the rectangular domain $(t,x) \in [t_0, T] \times [x_{\min}, x_{\max}]$ where $x_{\min}$ and $x_{\max}$ (such that $x_{\min} < x_{t_0} < x_{\max}$) are constants obtained by means of probabilistic considerations. In particular, by denoting (i) the points on the time axis by $t_m = t_0 + m\Delta t$ where $m = 0, \ldots, M$ and $\Delta t = (T - t_0)/M$, and (ii) the points on the spatial axis by $x_j = x_{\min} + j\Delta x$, where $j = 0, \ldots, N+1$ and $\Delta x = (x_{\max} - x_{\min})/(N+1)$, one can discretise the PDE (2.2.1) with finite-difference approximations for the first and second derivatives. A standard Euler discretisation scheme results in a matrix iteration of the form

$$L_B(t_m, \phi; \theta) V^m(\theta) = R_B(t_m, \phi; \theta) V^{m+1}(\theta) + \beta(t_{m+1}; \theta) \tag{2.2.6}$$

where $V^m(\theta) = (V(x_1, t_m; \theta), \ldots, V(x_N, t_m; \theta))^\top$ and $V(x_j, t_m; \theta)$ indicate the finite-difference approximation to the solution of the PDE (2.2.1)[1] and

$$L_B(t_m, \phi; \theta) = \mathbb{I} - \phi \Delta t D(\tilde{t}_m(\phi); \theta), \tag{2.2.7}$$

$$R_B(t_m, \phi; \theta) = \mathbb{I} + (1 - \phi) \Delta t D(\tilde{t}_m(\phi); \theta). \tag{2.2.8}$$

Here, $\tilde{t}_m(\phi) = (1 - \phi) t_m + \phi\, t_m$ and $\mathbb{I}$ is the $N \times N$ identity matrix. Both expressions (2.2.7) and (2.2.8) are defined in terms of the tri-diagonal matrix $D(t; \theta)$ specified by

$$[D(t; \theta)]_{j,j} = c_j(t; \theta), \tag{2.2.9}$$

$$[D(t; \theta)]_{j,j+1} = u_j(t; \theta), \tag{2.2.10}$$

$$[D(t; \theta)]_{j+1,j} = l_{j+1}(t; \theta), \tag{2.2.11}$$

[1] To keep the notation as light as possible, we denote the exact solution of the PDE (2.2.1) and its finite-difference approximation with the same symbol.

where $j = 1,\ldots,N$ in Equation (2.2.9) and $j = 1,\ldots,N-1$ in Equations (2.2.10) and (2.2.11). The coefficients

$$\begin{cases} c_j(t;\theta) = -\sigma(x_j,t;\theta)^2\Delta x^{-2} - \nu(x_j,t;\theta), \\ u_j(t;\theta) = \frac{1}{2}\mu(x_j,t;\theta)\Delta x^{-1} + \frac{1}{2}\sigma(x_j,t;\theta)^2\Delta x^{-2}, \\ l_j(t;\theta) = -\frac{1}{2}\mu(x_j,t;\theta)\Delta x^{-1} + \frac{1}{2}\sigma(x_j,t;\theta)^2\Delta x^{-2}, \end{cases} \tag{2.2.12}$$

are defined in terms of the functions $\mu(x,t;\theta)$, $\sigma(x,t;\theta)$ and $\nu(x_j,t;\theta)$ in the PDE (2.2.1). Moreover, $\beta(t_{m+1};(\theta))$ is an N-dimensional vector encoding suitable spatial boundary conditions which cannot be included in the matrix $D(t;\theta)$. The parameter ϕ is bounded between $\phi = 0$ (corresponding to the fully explicit scheme) and $\phi = 1$ (corresponding to the fully implicit scheme). Both schemes are characterised by an accuracy $O(\Delta x^2,\Delta t)$. The case $\phi = 1/2$ corresponds to the Cranck-Nicholson (CN) method, see [30], which is generally the method of choice in financial applications because it is characterised by an accuracy $O(\Delta x^2,\Delta t^2)$ and it is unconditionally stable in most cases. The Von Neumann stability analysis of different finite difference methods can be found in Appendix A. Here, we consider the stable discretization only. However, in some situations, e.g., for discontinuous payoff functions, combining the CN method with fully-implict iterations (as in the so-called "Rannacher stepping") has been shown to improve the accuracy of the numerical solution, see [68].

Given the value of the derivative at maturity $V_j^M(\theta) = P(x_j;\theta)$, Equation (2.2.6) can be solved recursively by utilising standard tri-diagonal solvers based on the LU-decomposition, for $m = M-1,\ldots,0$, in order to find the vector $V_j^0(\theta)$. From this, the value of the derivative $V_{t_0} = V(x_{t_0},t_0;\theta)$, corresponding to the state variable x_{t_0} observed at time t_0, can be computed by linear interpolation,

$$V_{t_0} = V_{j^\star}^0 + \frac{V_{j^\star+1}^0 - V_{j^\star}^0}{x_{j^\star+1} - x_{j^\star}}(x_{t_0} - x_{j^\star}), \tag{2.2.13}$$

with $j^\star$ such that $x_{j^\star} \le x_{t_0} < x_{j^\star+1}$. The associated algorithm is given as follows:

(S1) Initialise the value vector on the final time slice $V_j^M(\theta) = P(x_j;\theta)$ with $j =$

$0,\ldots,N$:

$$V^M = \texttt{PAYOFF}(\theta). \tag{2.2.14}$$

(S2) For $m = M-1,\ldots,0$ execute the following steps:

a) Compute the coefficient vectors $c^m(\theta) \equiv c(\tilde{t}_m(\phi);\theta)$, $u^m(\theta) \equiv u(\tilde{t}_m(\phi);\theta)$, and $l^m(\theta) \equiv l(\tilde{t}_m(\phi);\theta)$ in Equations (2.2.12):

$$(c^m, u^m, l^m) = \texttt{COMPUTECOEFFM}(\theta).$$

b) Compute the matrices $L_B^m(\theta) \equiv L_B(t_m,\phi;\theta)$ and $R_B^m(\theta) \equiv R_B(t_m,\phi;\theta)$ in Equations (2.2.7) and (2.2.8) from the coefficients vectors $c^m(\theta)$, $u^m(\theta)$, and $l^m(\theta)$:

$$(L_B^m, R_B^m) = \texttt{COMPUTELRB}(c^m, u^m, l^m). \tag{2.2.15}$$

c) Compute the boundary condition vector $\beta^{m+1}(\theta) \equiv \beta(t_m;\theta)$ by

$$\beta^{m+1} = \texttt{COMPUTEBC}(\theta). \tag{2.2.16}$$

d) Given V^{m+1}, solve Equation (2.2.6) for V^m by calling a suitable tri-diagonal solver such as

$$V^m = \texttt{TRIDIAGSOLVER}(L_B^m, R_B^m, \beta^{m+1}, V^{m+1}), \tag{2.2.17}$$

which we can represent mathematically as the following sequence of operations:

$$U^{m+1} = R_B^m V^{m+1}, \quad W^{m+1} = U^{m+1} + \beta^{m+1} \quad V^m = W^{m+1}/L_B^m, \tag{2.2.18}$$

where we adopte the notation "B/A" to represent finding the solution X of the linear system $AX = B$.

(S3) Compute $V_{t_0} = V(x_{t_0}, t_0;\theta)$ with a suitable interpolation scheme, for example,

the type shown in Equation (2.2.13), by calling a method of the kind

$$V_{t_0} = \texttt{COMPUTESPOTVALUE}(V^0).$$

Since the matrices (2.2.9), (2.2.10) and (2.2.11) are tri-diagonal, the cost of a single iteration of Equation (2.2.6) is $O(N)$. As a result, the overall computation complexity of the algorithm above is $O(NM)$.

2.2.2 Intermediate cashflows

Incorporating intermediate cash flows, which for instance may arise from coupon payments, in the finite-difference scheme is immediate and results in the following additional steps after a) – d) of (S2) in the previous section:

e) Initialise any additional payoff that might be necessary for the valuation of intermediate cash-flows, $C(t,x_t;\theta)$, when their value is not available in closed form, by $\mathscr{C}_{m+1}^{m+1} = \texttt{AUXILIARYPAYOFF}(\theta)$. Here $\mathscr{C}_k^m$ is the value vector at time t_m of a set of auxiliary securities with expiry t_k.

f) For $k = m+1,\ldots,M$, execute the tri-diagonal solver $\mathscr{C}_k^m = \texttt{TRIDIAGSOLVER}\left(L_B^m, R_B^m, \beta^{m+1}, \mathscr{C}_k^{m+1}\right)$.

g) Compute the intermediate cash-flow at time t_m and update the value vector

$$C^m = \texttt{COMPUTECASHFLOW}(\{\mathscr{C}_k^m\}_{k=m+1,\ldots,M}, \theta), \tag{2.2.19}$$

$$V^m \mathrel{+}= C^m, \tag{2.2.20}$$

where $C_j^m = C(t_m, x_j; \theta)$. We make use of the notation $+=$ for the standard "addition assignment" operator.

In the most common situations, when the backward PDE (2.2.1) is used to value an interest rate (respectively credit derivative), the auxiliary value vectors $\mathscr{C}_k^m$ in the steps above represent the value of the conditional discount factors,

$$\mathscr{C}_k^m = Z(r_{t_m}, t_m, t_k) \equiv \mathbb{E}\left[\exp\left(-\int_{t_m}^{t_k} r_u \,\mathrm{d}u\right) \,\middle|\, r_{t_m}\right],$$

with $r_t = r(x_t)$ (or respectively the value of the conditional survival probabilities $Q(h_{t_m}, t_m, t_k)$ in Equation (2.2.5)).

2.2.3 American-style options

The numerical algorithm described in the previous section can be extended to handle the pricing of securities with early exercise features, such as Bermudan-style and American-style options (see page 44 of [75]), provided that the exercise value can be expressed in terms of a deterministic function of the form $E(x_t, t; \theta)$. Indeed, on each exercise date, T_e, the Bellman principle (see [7]) can be expressed as a simple jump condition,

$$V(x, T_e; \theta) = \max\left(V(x, T_e^+; \theta), E(x, T_e^+; \theta)\right). \tag{2.2.21}$$

By indicating the set of early exercise dates (assumed for simplicity to be a subset of the discretisation dates t_m, $m = 0, \ldots, M$) by $\mathscr{T}_e$, early exercise can be incorporated into the finite-difference scheme of the previous section as the following additional steps after a) – d) of (S2):

e) Initialise any additional payoff that might be necessary for the valuation of the exercise function, $E(x_t, t; \theta)$, or the valuation of the intermediate cash-flow, $C(t, x_t; \theta)$, when their expressions are not available in closed-form from $\mathscr{C}_{m+1}^{m+1} = \texttt{AUXILIARYPAYOFF}(\theta)$. Here, as shown in Section 2.2.2, $\mathscr{C}_k^m$ is the value vector at time t_m of a set of auxiliary securities with expiry t_k.

f) As in Section 2.2.2.

g) As in Section 2.2.2.

h) If $t_m \in \mathscr{T}_e$, execute the instructions

$$E^m = \texttt{COMPUTEEXERCISEVALUE}(\{\mathscr{C}_k^m\}_{k=m+1,\ldots,M}, \theta), \tag{2.2.22}$$

$$H^m = V^m, \tag{2.2.23}$$

$$V^m = \texttt{EARLYEXCERCISE}(H^m, E^m), \tag{2.2.24}$$

where

(i) (2.2.22) computes the early exercise function $E(x_{t_m}, t_m; \theta)$, possibly using the auxiliary information $\mathscr{C}_k^m$, $k = m+1, \ldots, M$,

(ii) (2.2.23) assigns $V(x_{t_m}, t_m^+; \theta)$ to the so-called "hold value" H^m,

(iii) `EARLYEXCERCISE` in (2.2.24) applies the Bellman condition in Equation (2.2.21) to determine $V(x_{t_m}, t_m; \theta)$.

If the financial option may be exercised continuously in a given time interval, as is the case for American-style options, then the set $\mathscr{T}_e$ contains all the dates of the finite-difference grid in the time interval in which early option exercise is contractually allowed. This algorithm is generally accurate to the first-order convergence in the time step, even when the CN scheme is employed.

2.3 Arrow-Debreu prices and forward PDEs

An alternative approach to derivatives pricing is to solve the backward PDE (2.2.1) by the Arrow-Debreu price density, see e.g. page 42 of [51], which is also known as Green's function. In the present setting, the Arrow-Debreu price density reads:

$$\psi(y, T \,|\, x_t, t) = \mathbb{E}\left[\delta(y - x_T) \exp\left(-\int_t^T \nu(x_u, u; \theta) \mathrm{d}u\right) \,\middle|\, x_t\right], \tag{2.3.1}$$

where $\delta(\cdot)$ denotes the standard Dirac delta function. Similar to Section 2.2, the expectation is taken under the risk neutral measure. In the context of the interest rate derivatives and short-rate models introduced in Section 2.2, the price, $V_{t_0}(\theta)$, at time t_0 of a European-style option with maturity date T and payoff function, $P(r_T; \theta)$, is given by

$$V_{t_0}(\theta) = V(x_{t_0}, t_0; \theta) = \mathbb{E}\left[\exp\left(-\int_{t_0}^T r_u \mathrm{d}u\right) P(r_T; \theta) \,\middle|\, x_{t_0}\right],$$

where $r_t = r(x_t)$ is the instantaneous short rate. The option price can be computed by integrating the product of the payoff function and the Arrow-Debreu price density

over all the possible values the short rate may take at time T, which is

$$V_{t_0}(\theta) = \int_{\mathbb{R}} \psi(x, T \,|\, x_{t_0}, t_0) P(x;\theta) \mathrm{d}x. \tag{2.3.2}$$

The integration is performed over the range of the function $x = r^{-1}(r_T)$. In particular, the price, $Z(r_{t_0}, t_0, T)$, at time t_0 of a discount bond with maturity T, can be obtained as a special case by setting $P(x;\theta) = 1$, that is

$$Z(r_{t_0}, t_0, T) = \int_{\mathbb{R}} \psi(x, T \,|\, x_{t_0}, t_0) \mathrm{d}x, \tag{2.3.3}$$

where $r_{t_0} = r(x_{t_0})$. In the context of default intensity models, the conditional probability shown in Equation (2.2.5) can be expressed similarly to Equation (2.3.3). It is well known, see for example page 42 of [51], that the Arrow-Debreu price density (2.3.1) satisfies the following conjugate forward PDE:

$$\begin{aligned} \partial_t \psi(x,t \,|\, x_{t_0}, t_0) = &- \nu(x,t;\theta)\psi(x,t \,|\, x_{t_0}, t_0) - \partial_x \left(\mu(x,t;\theta)\psi(x,t \,|\, x_{t_0}, t_0)\right) \\ &+ \frac{1}{2}\partial_x^2 \left(\sigma^2(x,t;\theta)\psi(x,t \,|\, x_{t_0}, t_0)\right), \end{aligned} \tag{2.3.4}$$

where the initial condition is given by $\psi(x, t_0 \,|\, x_{t_0}, t_0) = \delta(x_{t_0} - x)$. The Arrow-Debreu price density, $\psi(x_t, t \,|\, x_{t_0}, t_0)$, can be determined for every $t > 0$ by solving Equation (2.3.4) forward in time.

2.3.1 Numerical solutions for forward PDEs by finite-difference discretisation

The conjugate forward PDE (2.3.4) can be discretised by the following steps, similar to the ones illustrated in Section 2.2.1 for the backward PDE. The main two differences are: 1) The finite difference method is to propagate starting from an initial condition, while the method in Section 2.2.1 is to backwards propagate from a terminal condition; 2) The boundary conditions for PDE (2.3.4) are modified as well, which take zero value at upper and lower boundaries. By approximating the Arrow-Debreu price density, $\psi(x, t_0 \,|\, x_{t_0}, t_0) = \delta(x_{t_0} - x)$, at time t_0 with its dis-

cretised counterpart, $\psi^0 \equiv \psi(x_j, t_0 \,|\, x_{t_0}, t_0) = \Delta x^{-1} \delta_{j,j^\star}$, where $\delta_{j,j^\star}$ is Kronecker's delta and $x_{j^\star}$ is the closest spatial grid point to x_{t_0}, and by setting for the spatial boundary conditions $\psi(x_{\min}, t_m \,|\, x_{t_0}, t_0) = \psi(x_{\max}, t_m \,|\, x_{t_0}, t_0) = 0$, one can compute the vector, $\psi^m(\theta) = (\psi(x_1, t_m \,|\, x_{t_0}, t_0), \ldots, \psi(x_N, t_m \,|\, x_{t_0}, t_0))^\top$, by iterating the matrix recursion

$$L_F(t_m, \phi; \theta)\psi^{m+1}(\theta) = R_F(t_m, \phi; \theta)\psi^m(\theta), \tag{2.3.5}$$

for $m = 0, \ldots, M-1$, where

$$L_F(t_m, \phi; \theta) = \mathbb{I} + (1-\phi)\Delta t D^T(\tilde{t}_m(\phi); \theta), \tag{2.3.6}$$

$$R_F(t_m, \phi; \theta) = \mathbb{I} - \phi \Delta t D^T(\tilde{t}_m(\phi); \theta). \tag{2.3.7}$$

The matrix $D(t; \theta)$ is determined by Equation (2.2.9) to (2.2.11). The algorithm to numerically solve the forward PDE (2.3.4) can therefore be described by:

(S1) Initialise the value vector on the initial time slice $\psi^0(\theta) = \Delta x^{-1} \delta_{j,j^\star}$ with $j = 0, \ldots, N$ by $\psi^0 = \texttt{DELTA}()$, which has, in our setup, no dependences on θ [2].

(S2) For $m = 0, \ldots, M-1$, execute

$$\psi^{m+1} = \texttt{PROPAGATEADPRICE}(\psi^m; \theta), \tag{2.3.8}$$

comprising of the following steps:

a) Compute the coeffiecients $c^m(\theta) \equiv c(\tilde{t}_m(\phi); \theta)$, $u^m(\theta) \equiv u(\tilde{t}_m(\phi); \theta)$, and $l^m(\theta) \equiv l(\tilde{t}_m(\phi); \theta)$ in Equations (2.2.12) by

$$(c^m, u^m, l^m) = \texttt{COMPUTECOEFFM}(\theta).$$

b) Compute the matrices, $L_F^m(\theta) \equiv L_F(t_m, \phi; \theta)$ and $R_F^m(\theta) \equiv R_F(t_m, \phi; \theta)$ in Equations (2.3.6) and (2.3.7) from the vectors of coefficients $c^m(\theta)$,

[2] The generalisation to the more general situation is straightforward.

$u^m(\theta)$ and $l^m(\theta)$ by

$$(L_F^m, R_F^m) = \texttt{COMPUTELRF}(c^m, u^m, l^m). \tag{2.3.9}$$

c) Given ψ^m, solve Equation (2.3.5) for ψ^{m+1} by calling a suitable tri-diagonal solver

$$\psi^{m+1} = \texttt{TRIDIAGSOLVER}(L_F^m, R_F^m, 0, \psi^m), \tag{2.3.10}$$

and executing $W^{m+1} = R_F^m \psi^m$ as well as $\psi^{m+1} = W^{m+1}/L_F^m$.

Here, we make use of the notation introduced at the end of Section 2.2.1. In order to compute the value $V_{t_0}(\theta)$ of a derivative asset, one needs to compute the integral (2.3.2) numerically, for example, by means of Gaussian quadrature.

(S3) Given the payoff vector $P = P(x;\theta)$, execute $V_{t_0} = \texttt{INTEGRATE}(P, \psi^M)$, where $P_j = P(x_j;\theta)$. It performs the numerical integration of Equation (2.3.2).

As for the backward PDE, the overall computational complexity of the algorithm above is of order $O(NM)$.

2.3.2 Forward PDEs and calibration

The valuation of a derivative security can be split in two distinct steps, a calibration and a pricing step. In the calibration step,

$$\theta = \texttt{CALIBRATION}(\mathscr{M}), \tag{2.3.11}$$

the parameters of the model $\theta = (\theta_1, \ldots, \theta_{N_\theta})$ are calibrated in order to reprice simple and liquidly-traded financial instruments. We denote the price of such instruments with the market parameter vector $\mathscr{M} = (\mathscr{M}_1, \ldots, \mathscr{M}_{N_\mathscr{M}})$. For instance, in the context of interest rate models (respectively credit models), the mean-reversion level $\mu(t)$ is calibrated to the prices of the instruments used to build a yield curve or, equivalently, a set of discount bond rates (respectively a set of prices of CDS

or also par spreads, see Chapter 5 of [61] for example). Similarly, the volatility function, or a combination of the volatility and the mean-reversion speed function, can be calibrated to swaption prices or implied volatilities. In the pricing step, the parameters θ are mapped to the values of the derivative security, or a portfolio of N_V securities:

$$V = \texttt{PRICING}(\theta), \tag{2.3.12}$$

so that the concatenation of the calibration of the calibration and the pricing step can be seen as a map of the form $\mathscr{M} \to \theta \to V$.

The calibration step, denoted by Equation (2.3.11), typically involves an iterative routine, for example, performing a numerical root search or least-square minimisation. Forward PDEs and combinations of forward and backward PDEs are usually used to efficiently implement the calibration step. In the following we will assume that the mean-reversion level $\mu(t)$, the mean-reversion speed $\kappa(t)$ and the volatility function $\sigma(t)$ in Equation (2.2.4) are all *càdlàg* piecewise constant on the time line $T_1 < \ldots < T_L$ (assumed uniform for simplicity in the following), which is a subset of the discretisation time axis t_m for $m = 0, \ldots, M$, with $T_1 > t_0$ and $T_L = t_M$. Also, we indicate by $\eta(i)$ the map such that $T_i = t_{\eta(i)}$, for $i = 1, \ldots, L$ and $\eta(0) = 0$. The model parameter θ can therefore be expressed in terms of the levels of such functions in each piecewise interval, namely $\theta = (\mu_1, \ldots, \mu_L, \kappa_1, \ldots, \kappa_L, \sigma_1, \ldots, \sigma_L)$, where $\mu_i = \mu(T_i)$, $\kappa_i = \kappa(T_i)$ and $\sigma_i = \sigma(T_i)$, for $i = 1, \ldots, L$.

In the context of credit derivatives, the first goal of the calibration is to match the survival probabilities in Equation (2.2.5), $Q(h_{t_0}, t_0, T_i)$ for $i = 1, \ldots, L$, with their market-implied counterparts, $Q^{\text{mkt}}(t_0, T_i)$, implied in turn via a standard bootstrap procedure from a set of CDS quotes observed in the market. Here $h_{t_0} = \exp(x_{t_0})$ is a free parameter of the model that for reasons of simplicity we assume fixed at some reasonable value.

The algorithm for the calibration of the mean-reversion level function can be described as follows. Initialise ψ^0 with (S1) of Section 2.3.1 and for $i = 1, \ldots, L$, proceed as follows.

(S1*) Choose μ_i.

(S2*) Perform the instructions in (2.3.8) for $m = \eta(i-1), \ldots, \eta(i) - 1$ and determine $\psi^{\eta(i)}$.

(S3*) Determine the numerical approximation of the survival probability in Equation (2.2.5) given by $Q(h_{t_0}, t_0, T_i) = \texttt{INTEGRATE}\left(\mathbf{1}, \psi^{\eta(i)}\right)$, where $\mathbf{1}$ is the N-dimensional unit vector.

(S4*) If the computed value $Q(h_{t_0}, t_0, T_i)$ equals what is quoted in the market, then stop, otherwise go back to (S1*) above.

The cost of the algorithm above is $O(MNN_{av})$ where N_{av} is the average number of root search iterations of (S1*)-(S3*) above, and it is $O(N)$ more efficient than what can be achieved with a backward PDE approach.

The calibration of the volatilities parameters to, for example, a collection of K CDS options with expiry dates T_i^e, $i = 1, \ldots K$ (assumed for simplicity to be a subset of $\{T_i\}_{i=1}^L$) and with underlying CDS maturity at time T_L, can be implemented efficiently with a combination of the forward and backward algorithms. Here we assume, for simplicity, a stylised payoff for a CDS option with expiry date T_i^e and underlying maturity T_L of the form

$$P^{\text{swpt}}\left(h_{T_i^e}, T_i^e, T_L\right) = \left(s\left(h_{T_i^e}\right) - c\right)^+ \mathscr{A}\left(h_{T_i^e}, T_i^e, T_L\right), \tag{2.3.13}$$

where c is the running coupon of the CDS at which the option can be exercised, and $\mathscr{A}(h_{T_i^e}, T_i^e, T_L)$ and $s(h_{T_i^e})$ are, respectively, the credit-risky annuity and the par-spread for a T_L maturity CDS contract starting at time T_i^e. The price at time T_i^e of a credit-risky annuity is given by $\mathscr{A}(h_{T_i^e}, T_i^e, T_L) = \sum_{t_m \in \mathscr{C}(T_i^e, T_L)} \Delta t_c \; Z(T_i^e, t_m) Q(h_{T_i^e}, T_i^e, t_m)$, where $\mathscr{C}(T_i^e, T_L)$ is the set of discretisation dates t_m corresponding to the coupon dates[3] for a CDS starting at time T_i^e and maturing at T_L. The interval Δt_c is the length of the coupon period, which is assumed uniform and commensurate with the spacing of the time grid $T_1, \ldots, T_L$ for

[3]We can assume for simplicity that the coupon dates are a subset of the discretization dates t_m, for $m = 1, \ldots, M$.

simplicity, and

$$Z(t,t_m) = \exp\left(-\int_t^{t_m} r_u \mathrm{d}u\right)$$

is the deterministic discount factor. The CDS par-spread is defined by

$$s(h_{T_i^e}) = \mathscr{L}(h_{T_i^e}, T_i^e, T_L)/\mathscr{A}(h_{T_i^e}, T_i^e, T_L)$$

where $\mathscr{L}(h_{T_i^e}, T_i^e, T_L)$ is the discounted expected loss specified by

$$\mathscr{L}(h_{T_i^e}, T_i^e, T_L) = (1-R)\int_{T_i^e}^{T_L} Z(T_i^e, u)\left(-\frac{\mathrm{d}Q(h_{T_i^e}, T_i^e, u)}{\mathrm{d}u}\right)\mathrm{d}u,$$

and R is the expected recovery rate, which is assumed independent of the default time. In a discretised setting, the expected loss is generally approximated by

$$\mathscr{L}(h_{T_i^e}, T_i^e, T_L) = (1-R)\sum_{t_m \in \mathscr{C}(T_i^e, T_L)} Z(T_i^e, t_m)\left(Q(h_{T_i^e}, T_i^e, t_m - \Delta t_c) - Q(h_{T_i^e}, T_i^e, t_m)\right),$$

so that the payoff in Equation (2.3.13) can be computed, provided that the conditional survival probabilities $Q(h_{T_i^e}, T_i^e, t_m)$ with $t_m \in \mathscr{C}(T_i^e, T_L)$ and $i = 1, \ldots, K$ can be calculated. The calculation of the price of a swaption with expiry T_i^e for $i = 1 \ldots, K$, and underlying CDS-maturity T_L,

$$V_{t_0}^{\mathrm{swpt}}(T_i^e, T_L) = Z(t_0, T_i^e)\,\mathbb{E}\left[P^{\mathrm{swpt}}\left(h_{T_i^e}, T_i^e, T_L\right) | x_{t_0}\right],$$

given a set of volatilities $\sigma_1, \ldots, \sigma_L$, can be implemented using the following method:

(S1′) By applying the forward induction from (S1*) to (S4*), calibrate μ_l, $l = 1, \ldots, L$. Save the Arrow-Debreu prices $\psi_j^{\eta(l)} = \psi(x_j, T_l \,|\, x_{t_0}, t_0)$ where $l = 1, \ldots, L$.

(S2′) Execute (S2) of the backward induction algorithm in Section 2.2.1 equipped with the steps $e)$ and $f)$ of Section 2.2.2. The auxiliary securities are chosen such that they provide a unit cash-flow at each coupon date of the CDS un-

derlying the options, namely $\mathscr{C}_m^m = \mathbf{1}$ for $t_m \in \mathscr{C}(T_i^e, T_L)$, and zero otherwise. As a result, $[\mathscr{C}_k^m]_j$ represents the conditional survival probability $Q(x_j, t_m, t_k)$.

(S3′) Given the survival probabilities $Q(x_j, T_i^e, t_m)$, for $t_m \in \mathscr{C}(T_i^e, T_L)$, computed in (S2′)

a) create the CDS option payoff in Equation (2.3.13), and

b) integrate the payoff as in Equation (2.3.2) against $\psi(x_j, T_i^e \,|\, x_{t_0}, t_0)$ by numerical quadrature. Then produce the value, $V_{t_0}^{\text{swpt}}(T_i^e, T_L)$, of the swaption at time t_0.

For processes characterised by a weak dependence of swaption prices on instantaneous volatility after the expiry, the calibration of the volatilities $\sigma_1, \ldots, \sigma_L$ can be performed with the following bootstrap procedure, see for example [5]. Starting from the first option expiry date T_1^e, one can vary all the knot points of the instantaneous volatilities at times before T_1^e while keeping the others constant until the value $V_{t_0}^{\text{swpt}}(T_1^e, T_L)$ of the swaption is matched. Similarly, for the subsequent dates T_i^e, one can simultaneously vary all the knot points of the instantaneous volatilities at the times between (and including) T_{i-1}^e and T_i^e until the value $V_{t_0}^{\text{swpt}}(T_i^e, T_L)$ of the swaption is matched. On the last expiry date T_K^e, one can vary all the knot points of the instantaneous volatilities at T_{K-1}^e and after until the value $V_{t_0}^{\text{swpt}}(T_K^e, T_L)$ of the swaption is matched. Since a swaption price has a weak dependence on all the volatilities past its expiry dates, the bootstrap procedure above needs to be repeated a few times until convergence is achieved. Conversely, when swaption prices have a strong dependence on volatility after expiry, one cannot apply the bootstrap procedure above and the recourse to a multidimensional solver is necessary.

2.4 AAD and PDEs

2.4.1 Adjoint algorithmic differentiation

The main idea underlying algorithmic differentiation, see for example [42], is that any computer implemented function – no matter how complicated – can be interpreted as a composition of basic arithmetic and intrinsic operations that are easy to

differentiate. In particular, when one requires the derivatives of a small number of outputs with respect to a large number of inputs, the calculation can be optimised by applying the chain rule through the instructions of the programme in opposite order with respect to their original evaluation. This gives rise to AAD. Page 75 to 80 in book [42] contains a detailed discussion of the computational cost of AAD. In this section, we will only recall the main results in order to clarify what financial computations and implementations can benefit from. This technique can be found in [19] with several simple examples illustrating the intuition behind these results.

We now consider a function,

$$Y = \texttt{FUNCTION}(X), \tag{2.4.1}$$

that maps a vector $X \in \mathbb{R}^n$ to a vector $Y \in \mathbb{R}^m$ through a sequence of steps $X \to \ldots \to U \to V \to \ldots \to Y$. The real vectors U and V represent intermediate variables in the calculation, and each step can be a distinct high-level function or even a specific instruction. The adjoint mode of algorithmic differentiation results from propagating the derivatives of the final output with respect to all the intermediate variables—the so called *adjoints*—until the derivatives with respect to the independent variables are formed. Using the standard AD notation, the adjoint of any intermediate variable V_k is defined by

$$\bar{V}_k = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial V_k},$$

where $\bar{Y}$ is a vector in $\mathbb{R}^m$. By applying the chain rule, we get for each variable U_i,

$$\bar{U}_i = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial U_i} = \sum_{j=1}^{m} \bar{Y}_j \sum_k \frac{\partial Y_j}{\partial V_k} \frac{\partial V_k}{\partial U_i},$$

which corresponds to the adjoint mode equation for the intermediate step represented by the function $V = V(U)$. We thus have a function of the form $\bar{U} = \bar{V}(U, \bar{V})$ where

$$\bar{U}_i = \sum_k \bar{V}_k \frac{\partial V_k}{\partial U_i}.$$

Working from the right to the left, $\bar{X} \leftarrow \ldots \leftarrow \bar{U} \leftarrow \bar{V} \leftarrow \ldots \leftarrow \bar{Y}$, we apply this rule to each step in the calculation until we obtain $\bar{X}$. In other words, we continue until one obtains the linear combination of the rows of the Jacobian of the function $X \to Y$, that is

$$\bar{X}_i = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial X_i} \qquad (i = 1, \ldots, n). \tag{2.4.2}$$

In the adjoint mode, the cost does not increase with the number of inputs, but it is linear in the number of rows of the Jacobian that need to be evaluated independently. If the full Jacobian is required, one needs to repeat the adjoint calculation m times, setting the vector $\bar{Y}$ equal to each of the elements of the canonical basis in $\mathbb{R}^m$.

One important theoretical result is that given a computer programme performing some high-level function (2.4.1), the execution time of its adjoint counterpart $\bar{X} = \texttt{FUNCTION\_b}(X, \bar{Y})$ (with suffix `_b` for "backward" or "bar") that computes the linear combination (2.4.2), is bounded by three to four times the cost of execution of the original one. That is,

$$\frac{\text{Cost}[\texttt{FUNCTION\_b}]}{\text{Cost}[\texttt{FUNCTION}]} \leq \omega_A \tag{2.4.3}$$

where $\omega_A \in [3,4]$, see page 80 of [42].

2.4.2 AAD and backward PDEs

The evaluation of the numerical solution of the PDE (2.2.1) by means of the algorithm described in Section 2.2.1 can be seen as a computer-implemented function mapping $\theta \to V_{t_0}(\theta)$. By following the principles of AAD, it is possible to design its adjoint counterpart $(\theta, \bar{V}_{t_0}) \to (V_{t_0}, \bar{\theta})$ which gives the sensitivities,

$$\bar{\theta}_k = \frac{\partial V(\theta)}{\partial \theta_k}, \tag{2.4.4}$$

for $k = 1, \ldots, N_\theta$. Here, we set $\bar{V}_{t_0} = 1$.

The adjoint of the solution of the backward PDE in Section 2.2.1 consists therefore of Steps (S1)-(S3) followed by their corresponding adjoint, executed in reverse

order:

($\bar{\text{S}}3$) Set $\bar{V}_{t_0} = 1$, and execute

$$\bar{V}^0 = \texttt{COMPUTESPOTVALUE\_b}(V^0, \bar{V}_{t_0})$$

to compute

$$\bar{V}_j^0 = \bar{V}_{t_0} \frac{\partial V_{t_0}}{\partial V_j^0}$$

for $j = 1, \ldots, N$, according to rule (2.2.13).

($\bar{\text{S}}2$) For $m = 0, \ldots, M-1$, in opposite order to (S2) from Section 2.2.1, execute

$\bar{d}$) Given $\bar{V}^m$, execute the adjoint of function (2.2.17), namely

$$(\bar{L}_B^m, \bar{R}_B^m, \bar{\beta}^{m+1}, \bar{V}^{m+1}) = \texttt{TRIDIAGSOLVER\_b}(L_B^m, R_B^m, \beta^{m+1}, V^{m+1}, \bar{V}^m),$$

which computes

$$[\bar{L}_B^m]_{j,l} = \sum_{r=1}^{N} \bar{V}_r^m \frac{\partial V_r^m}{\partial [L_B^m]_{j,l}}, \quad [\bar{R}_B^m]_{j,l} = \sum_{r=1}^{N} \bar{V}_r^m \frac{\partial V_r^m}{\partial [R_B^m]_{j,l}},$$
$$\bar{\beta}_j^{m+1} = \sum_{r=1}^{N} \bar{V}_r^m \frac{\partial V_r^m}{\partial \beta_j^{m+1}}, \qquad \bar{V}_j^{m+1} = \sum_{r=1}^{N} \bar{V}_r^m \frac{\partial V_r^m}{\partial V_j^{m+1}},$$

for $j = 1, \ldots, N$ and $l = 1, \ldots, N$.

$\bar{c}$) Compute the adjoint of (2.2.16), namely $\bar{\theta} = \texttt{COMPUTEBC\_b}(\theta, \bar{\beta}^{m+1})$, which gives

$$\bar{\theta}_k += \sum_{r=1}^{N} \bar{\beta}_r \frac{\partial \beta_r^m}{\partial \theta_k}.$$

The initialisation of the vector of sensitivities $\bar{\theta}$ is given by

$$\bar{\theta}_k = \sum_{r=1}^{N} \bar{\beta}_r \frac{\partial \beta_r^M}{\partial \theta_k}.$$

$\bar{b}$) Compute the adjoint of the function (2.2.15), that is

$$(\bar{c}^m, \bar{u}^m, \bar{l}^m) = \texttt{COMPUTELRB\_b}(c^m, u^m, l^m, \bar{L}_B^m, \bar{R}_B^m),$$

which produces the adjoint of the coefficient vectors

$$\bar{c}_j^m = [\bar{L}_B^m]_{j,j} \frac{\partial [L_B^m]_{j,j}}{\partial c_j^m} + [\bar{R}_B^m]_{j,j} \frac{\partial [R_B^m]_{j,j}}{\partial c_j^m}, \tag{2.4.5}$$

$$\bar{u}_j^m = [\bar{L}_B^m]_{j,j+1} \frac{\partial [L_B^m]_{j,j+1}}{\partial u_j^m} + [\bar{R}_B^m]_{j,j+1} \frac{\partial [R_B^m]_{j,j+1}}{\partial u_j^m}, \tag{2.4.6}$$

$$\bar{l}_{j+1}^m = [\bar{L}_B^m]_{j+1,j} \frac{\partial [L_B^m]_{j+1,j}}{\partial l_{j+1}^m} + [\bar{R}_B^m]_{j+1,j} \frac{\partial [R_B^m]_{j+1,j}}{\partial l_{j+1}^m}, \tag{2.4.7}$$

where $j = 1, \ldots, N$ in Equation (2.4.5) and $j = 1, \ldots, N-1$ in (2.4.6) and (2.4.7). Here we have used the fact that each component of the vectors c^m, u^m and l^m appears only in one element of the three main diagonals of the matrices L_B^m and R_B^m. By Equations (2.2.7) and (2.2.8) it is immediately verified that

$$\frac{\partial [L_B^m]_{j,j}}{\partial c_j^m} = \frac{\partial [L_B^m]_{j,j+1}}{\partial u_j^m} = \frac{\partial [L_B^m]_{j+1,j}}{\partial l_{j+1}^m} = -\phi\,\delta t,$$
$$\frac{\partial [R_B^m]_{j,j}}{\partial c_j^m} = \frac{\partial [R_B^m]_{j,j+1}}{\partial u_j^m} = \frac{\partial [R_B^m]_{j+1,j}}{\partial l_{j+1}^m} = (1-\phi)\,\delta t,$$

for $0 < \phi < 1$. In order to be fully explicit, $\phi = 0$ (respectively the fully explicit case, $\phi = 1$), L_B^m (respectively R_B^m) should be the identity matrix and $\bar{L}_B^m$ (respectively $\bar{R}_B^m$) is identically zero.

$\bar{a}$) Compute the adjoints of the coefficients (2.2.12),

$$\bar{\theta} \mathrel{+}= \texttt{COMPUTECOEFFM\_b}(\theta, \bar{c}^m, \bar{u}^m, \bar{l}^m). \tag{2.4.8}$$

This produces the following contribution to the adjoint of the vector $\bar{\theta}$,

$$\bar{\theta}_k \mathrel{+}= \sum_{j=1}^{N} \left[\bar{c}_j^m \frac{\partial c_j^m(\theta)}{\partial \theta_k} + \bar{u}_j^m \frac{\partial u_j^m(\theta)}{\partial \theta_k} + \bar{l}_j^m \frac{\partial l_j^m(\theta)}{\partial \theta_k} \right]$$

for $k = 1, \ldots, N_\theta$, with $u_N^m \equiv 0$ and $l_1^m \equiv 0$.

($\bar{\text{S}}1$) Compute the adjoint of the vector V^M in (2.2.14) by executing

$$\bar{\theta} \mathrel{+}= \texttt{PAYOFF\_b}(\theta, \bar{V}^M).$$

This gives the vector elements,

$$\bar{\theta}_k \mathrel{+}= \sum_{j=1}^{N} \bar{V}_j^M \frac{\partial P(x_j;\theta)}{\partial \theta_k},$$

for $k = 1, \ldots, N_\theta$, associated with the explicit dependence of the payoff on the model parameters θ (if any).

One can verify that the execution of the steps above produces the sensitivities shown in Equation (2.4.4) of the option value with respect to the parameters θ. According to the general result of AAD (see (2.4.3)), the cost to compute all the components of the adjoint vector $\bar{\theta}$ is a small multiplier of order four times the cost of computing (S1) to (S4), therefore resulting in an overall computation complexity of $O(NM)$.

We note that obtaining the adjoint `COMPUTESPOT_b` of the linear scheme in Equation (2.2.13) is straightforward. The procedure consists of setting $\bar{V}_j^0 = 0$ for $j \notin \{j^\star, j^\star + 1\}$, and allocating $\bar{V}_{j^\star}^0$ and $\bar{V}_{j^\star+1}^0$ with their coefficients in Equation (2.2.13), namely

$$\bar{V}_{j^\star}^0 = \bar{V}_{t_0} \left(1 - \frac{x_{t_0} - x_{j^\star}}{x_{j^\star+1} - x_{j^\star}}\right), \text{ and } \quad \bar{V}_{j^\star+1}^0 = \bar{V}_{t_0} \frac{x_{t_0} - x_{j^\star}}{x_{j^\star+1} - x_{j^\star}}.$$

The adjoint function `TRIADIAGSOLVER_b`, which gives the adjoint of (2.2.18), is produced by

$$\begin{aligned} &\bar{W}^{m+1} = [L_B^m]^{-T} \bar{V}^m, \quad \overline{[L_B^m]^{-1}} = \bar{V}^m \left[W^{m+1}\right]^T, \quad \bar{L}_B^m = -[L_B^m]^{-T} \overline{[L_B^m]^{-1}} [L^m]_B^{-T}, \\ &\bar{U}^{m+1} = \bar{W}^{m+1}, \quad \bar{\beta}^{m+1} = \bar{W}^{m+1}, \quad \bar{R}_B^m = \bar{U}^{m+1} \left[V^{m+1}\right]^T, \quad \bar{V}^{m+1} = [R_B^m]^T \bar{U}^{m+1}. \end{aligned} \tag{2.4.9}$$

Here we have used the fact that the adjoint of the linear operation $y = Bx$ is given by $\bar{x} = B^T \bar{y}$ and $\bar{B} = \bar{y} x^T$, and the identity $\bar{A} = -A^{-T} \overline{A^{-1}} A^{-T}$, which holds for

any invertible matrix A, see [37]. The computational cost of the instructions above is $O(N^2)$. In order to reduce the computational cost to $O(N)$, as in the original sequence (2.2.18), one needs to avoid the matrix inversion in the first instruction (2.4.9). This is obtained by utilising the solution of a linear system and then by combining the first three instructions of Equation (2.4.9) and the third of Equation (2.2.18). We thus have:

$$\bar{L}_B^m = -\left[L_B^m\right]^{-T} \bar{V}^m \left[W^{m+1}\right]^T \left[L^m\right]_B^{-T} = -\bar{W}^{m+1} \left[\left[L^m\right]_B^{-1} W^{m+1}\right]^T = -\bar{W}^{m+1} \left[V^m\right]^T.$$

Then, the resulting algorithm is given by

$$\begin{aligned} &\bar{W}^{m+1} = \bar{V}^m / [L_B^m]^T, \quad \bar{L}_B^m = -\bar{W}^{m+1} \left[V^m\right]^T, \quad \bar{U}^{m+1} = \bar{W}^{m+1}, \\ &\bar{\beta}^{m+1} = \bar{W}^{m+1}, \quad \bar{R}_B^m = \bar{U}^{m+1} \left[V^{m+1}\right]^T, \quad \bar{V}^{m+1} = [R_B^m]^T \bar{U}^{m+1}. \end{aligned} \tag{2.4.10}$$

We emphasise that only the elements on the three main diagonals of $\bar{L}_B^m$ and $\bar{R}_B^m$ contribute to the sensitivities, so that only $3N$ multiplications are required for their computation in the second and fourth instruction of Equation (2.4.10). The overall computational cost of the adjoint tri-diagonal solver is $O(N)$, exactly as for the forward counterpart shown in the sequence of equations (2.2.18), and as expected from the general result shown in Equation (2.4.3).

The execution of the adjoint instructions (2.4.10) requires the vector V^m. This is a manifestation of the general feature of the adjoint implementation which requires: (i) the execution of the original code; and (ii) the storage of the intermediate results as well as the final outputs before the execution of its adjoint counterpart.

In this case, `TRIADIAGSOLVER_b` needs to contain a forward sweep replicating the instructions (2.2.18) in order to compute V^m. Alternatively, if the values V_m were to be stored during the calculation in the forward sweep of (S1)-(S3), then one could use the stored values directly as inputs in `TRIADIAGSOLVER_b`. This scheme is more efficient as it avoids repeating the forward sweep. The first implementation comes with a reduced memory consumption as it does not store the vectors V^m for $m = 0, \ldots, M$ and is an example of the "checkpointing" technique,

see [19].

Finally, the adjoint of the function COMPUTECOEFFM_b in Equation (2.4.8) can be implemented by the adjoint of Equations (2.2.12), namely

$$\bar{\sigma}_j^m = -\bar{c}_j^m 2\sigma(x_j,\tilde{t}_m;\theta)\Delta x^{-2} + \bar{u}_j^m \sigma(x_j,\tilde{t}_m;\theta)\Delta x^{-2} + \bar{l}_j^m \sigma(x_j,\tilde{t}_m;\theta)\Delta x^{-2},$$
$$\bar{\nu}_j^m = -\bar{c}_j^m, \bar{\mu}_j^m = \bar{u}_j^m \tfrac{1}{2}\Delta x^{-1} - \bar{l}_j^m \tfrac{1}{2}\Delta x^{-1},$$

for $j = 1,\ldots,N$, and

$$\bar{\theta} \mathrel{+}= \bar{\sigma}(x_j,\tilde{t}_m;\theta,\bar{\sigma}_j^m), \bar{\theta} \mathrel{+}= \bar{\mu}(x_j,\tilde{t}_m;\theta,\bar{\mu}_j^m), \bar{\theta} \mathrel{+}= \bar{\nu}(x_j,\tilde{t}_m;\theta,\bar{\nu}_j^m), \qquad (2.4.11)$$

and adding the contributions to the sensitivities

$$\bar{\theta}_k \mathrel{+}= \bar{\sigma}_j^m \frac{\partial\sigma(x_j,\tilde{t}_m;\theta)}{\partial\theta_k}, \quad \bar{\theta}_k \mathrel{+}= \bar{\mu}_j^m \frac{\partial\mu(x_j,\tilde{t}_m;\theta)}{\partial\theta_k}, \quad \bar{\theta}_k \mathrel{+}= \bar{\nu}_j^m \frac{\partial\nu(x_j,\tilde{t}_m;\theta)}{\partial\theta_k},$$

for $k = 1,\ldots,N_\theta$. The implementation of the adjoint functions in Equation (2.4.11) depends on the particular model considered.

2.4.2.1 Intermediate cashflows and American-style options

The adjoint algorithm presented in the previous section can be extended to include early-exercise contracts such as those described in Sections 2.2.2 and 2.2.3, by employing the following modification before $\bar{d}$) of ($\bar{S2}$) above:

$\bar{h}$) For $t_m \in \mathscr{T}_e$, set $\{\bar{\mathscr{C}}_k^m\}_{k=m+1,\ldots,M} = 0$ and execute the following instructions:

$$(\bar{H}^m,\bar{E}^m) = \texttt{EARLYEXCERCISE\_b}(H^m,E^m,\bar{V}^m), \qquad \bar{V}^m = H^m,$$
$$(\{\bar{\mathscr{C}}_k^m\}_{k=m+1,\ldots,M},\bar{\theta}) \mathrel{+}=$$
$$\texttt{COMPUTEEXERCISEVALUE\_b}(\{\mathscr{C}_k^m\}_{k=m+1,\ldots,M},\theta,\bar{E}^m).$$

It is important to note that the application of the AAD rules in [19] require the adjoint $\bar{V}^m$ be overridden rather than incremented.

$\bar{g}$) Execute the adjoint of Equation (2.2.19), that is, $\bar{C}^m = \bar{V}^m$ and

$$(\{\bar{\mathscr{C}}_k^m\}_{k=m+1,\ldots,M}, \bar{\theta}) \mathrel{+}= \texttt{COMPUTECASHFLOW\_b}(\{\mathscr{C}_k^m\}_{k=m+1,\ldots,M}, \theta, \bar{C}^m).$$

$\bar{f}$) For $k = M, \ldots, m+1$, call the adjoint tri-diagonal solver,
$(\bar{L}_B^m, \bar{R}_B^m, \bar{\beta}^{m+1}, \bar{\mathscr{C}}_k^{m+1}) \mathrel{+}= \texttt{TRIDIAGSOLVER\_b}(L_B^m, R_B^m, \beta^{m+1}, \mathscr{C}_k^{m+1}, \bar{\mathscr{A}}_k^m)$.

$\bar{e}$) Then execute $\bar{\theta} \mathrel{+}= \texttt{AUXILIARYPAYOFF\_b}(\theta, \bar{\mathscr{C}}_{m+1}^{m+1})$, which gives the contribution to the sensitivities arising from the intermediate cashflows and the early-exercise optionality.

2.4.3 AAD and forward PDEs

Analogous to what is discussed in Section 2.4.2 for the backward PDE, the adjoint of the numerical solution of the forward PDE of Section 2.3.1 consists of (S1)-(S3) followed by their corresponding adjoint operations executed in reverse order. Or one can use a change of variable $\tau = T - t$ and follow the same procedures as in Section 2.4.2. Here, we present the AAD algorithm for (S1)-(S3) as follows:

($\bar{\text{S3}}$) Set $\bar{V}_{t_0} = 1$, execute $(\bar{P}, \bar{\psi}^M) = \texttt{INTEGRATE\_b}(P, \psi^M, \bar{V}_{t_0})$ and compute, according to the rule (2.2.13), the gradients

$$\bar{\psi}_j^M = \bar{V}_{t_0} \frac{\partial V_{t_0}}{\partial \psi_j^M}, \qquad\qquad \bar{P}_j = \bar{V}_{t_0} \frac{\partial V_{t_0}}{\partial P_j},$$

for $j = 1, \ldots, N$. For an example in which a Gaussian quadrature scheme is applied, we refer to [20]. The contribution to the sensitivities arising from the functional form of the payoff (if any) is then computed by

$$\bar{\theta}_k = \sum_{j=1}^{N} \bar{P}_j \frac{\partial P_j}{\partial \theta_j} \qquad (k = 1, \ldots, N_\theta).$$

($\bar{\text{S2}}$) For $m = M-1, \ldots, 0$ continue with the following steps:

$\bar{c}$) Given $\bar{\psi}^{m+1}$, execute the adjoint of the function in Equation (2.3.10), namely
$(\bar{L}_F^m, \bar{R}_F^m, \bar{\psi}^m) = \texttt{TRIDIAGSOLVER\_b}(L_F^m, R_F^m, 0, \psi^m, \bar{\psi}^{m+1})$.

$\bar{b}$) Compute the adjoint of the function in Equation (2.3.9), namely $(\bar{c}^m, \bar{u}^m, \bar{l}^m) = \texttt{COMPUTELRF\_b}(c^m, u^m, l^m, \bar{L}_F^m, \bar{R}_F^m)$.

$\bar{a}$) Compute the adjoints of the coefficients (2.2.12),

$$\bar{\theta} \mathrel{+}= \texttt{COMPUTECOEFFM\_b}(\theta, \bar{c}^m, \bar{u}^m, \bar{l}^m),$$

with the same adjoint functions as described in Section 2.4.2.

($\bar{\text{S1}}$) This step is void since the initialisation function, DELTA(), has no dependency on θ.

One can verify that the execution of the steps above produces the sensitivities of the option value with respect to the parameters θ, see Equation (2.4.4). As before, the general AAD result in (2.4.3) guarantees that the cost to compute all the components of the adjoint vector $\bar{\theta}$ is a maximum of four times the cost of computing (S1*)-(S4*) in Section 2.3.2, therefore resulting in an overall computational complexity of order $O(NM)$.

2.4.4 Calibration algorithm: AAD and the implicit function theorem

As demonstrated in Section 2.3.2, the valuation of a derivative security can be generally separated in two distinct steps, namely a calibration and a pricing step. While the calculation of the sensitivities with respect to the internal model parameters $\frac{\partial V}{\partial \theta}$, obtained by the adjoint of the pricing step (2.3.12), $\bar{\theta} = \texttt{PRICING\_b}(\theta, \bar{V})$, which computes

$$\bar{\theta}_k = \sum_{i=1}^{N_V} \bar{V}_i \frac{\partial V_i}{\partial \theta_k},$$

for $k = 1, \ldots, N_\theta$ is sometimes useful, what is required for the risk management of the portfolio of the derivative securities are the sensitivities $\frac{\partial V}{\partial \mathscr{M}}$, with respect to the liquid market prices, because they define the size of the hedges. These can be obtained, according to the general principles of AAD, by reversing the order of computations so the adjoint of the algorithm consists of the adjoint pricing step,

combined with the adjoint calibration step,

$$\bar{\mathscr{M}} = \texttt{CALIBRATION\_b}(\mathscr{M}, \bar{\theta}), \tag{2.4.12}$$

giving

$$\bar{\mathscr{M}}_m = \sum_{k=1}^{N_\theta} \bar{\theta}_k \frac{\partial \theta_k}{\partial \mathscr{M}_m},$$

for $m = 1, \ldots, N_{\mathscr{M}}$. The overall adjoint algorithm can be seen therefore as a map of the form $\bar{V} \to \bar{\theta} \to \bar{\mathscr{M}}$.

The adjoint calibration step, (2.3.11), can be implemented according to the general rules of AAD (see Section 2.4.1), paying attention to its iterative nature. However, following the work by [27] and [44], a much better performance can be obtained by exploiting the so-called *implicit function theorem* (IFT), as described below. Here we consider the case in which the calibration algorithm in Equation (2.4.12) consists of the numerical solution of a system of equations of the form

$$G_i(\mathscr{M}, \theta) = 0 \tag{2.4.13}$$

where $\mathscr{M} \in \mathscr{R}^{N_{\mathscr{M}}}$, $\theta \in \mathscr{R}^{N_\theta}$ and $i = 1, \ldots, N_\theta$. The function $G_i(\mathscr{M}, \theta)$ is often of the form

$$G_i(\mathscr{M}, \theta) = V_i^{\text{mkt}}(\mathscr{M}) - V_i(\theta) \tag{2.4.14}$$

where $V_i(\theta)$ is the price of the i-th calibration instrument as produced by the model to be calibrated, and $V_i^{\text{mkt}}(\mathscr{M})$ are the prices of the target instruments, possibly generated by a simpler model used as a quoting mechanism.

As noted above, the adjoint calibration can be implemented in terms of the adjoint of the numerical scheme solving equation system (2.4.13). The associated computational cost is expected to be a few times of the cost of solving the equation system (2.4.13) (but approximately less than 4 times the cost, according to the general result of AAD). Better performance can be obtained by the IFT. Under mild regularity conditions, the IFT says that if there is a solution, $(\mathscr{M}_0, \theta_0)$, to the root

finding problem shown in equation system (2.4.13), such that $G_i(\mathscr{M}_0, \theta_0) = 0$, and the matrix of derivatives $[\frac{\partial G}{\partial \theta}]_{ij} = \frac{\partial G_i}{\partial \theta_j}(\mathscr{M}_0, \theta_0)$ is invertible, then one can define in the vicinity of $\mathscr{M}_0$ an implicit function $\theta = \theta(\mathscr{M})$ such that

$$G_i(\mathscr{M}, \theta(\mathscr{M})) = 0. \tag{2.4.15}$$

The derivatives $\frac{\partial \theta}{\partial \mathscr{M}}$ of such function can be expressed in terms of the derivatives of the objective function G. Indeed, by differentiating Equation (2.4.15) with respect to $\mathscr{M}$, one obtains

$$\frac{\partial G_i}{\partial \mathscr{M}_m} + \sum_{j=1}^{N_\theta} \frac{\partial G_i}{\partial \theta_j} \frac{\partial \theta_j}{\partial \mathscr{M}_m} = 0$$

for $m = 1, \ldots, N_{\mathscr{M}}$, or equivalently

$$\frac{\partial \theta_k}{\partial \mathscr{M}_m} = - \left[\left(\frac{\partial G}{\partial \theta} \right)^{-1} \frac{\partial G}{\partial \mathscr{M}} \right]_{k,m},$$

with $[\frac{\partial G}{\partial \mathscr{M}}]_{ij} = \frac{\partial G_i}{\partial \mathscr{M}_j}$. This relation allows the computation of the sensitivities of the function $\theta(\mathscr{M})$, locally defined in an implicit manner by Equation (2.4.13), in terms of the sensitivities of the function $G(\mathscr{M}, \theta)$. These can be computed by the corresponding adjoint function $(\bar{\mathscr{M}}, \bar{\theta}) = \bar{G}(\mathscr{M}, \theta, \bar{G})$ giving, according to the general rule (Section 2.4.1),

$$\bar{\mathscr{M}}_m = \sum_{i=1}^{N_\theta} \bar{G}_i \frac{\partial G_i}{\partial \mathscr{M}_m}, \qquad \bar{\theta}_k = \sum_{i=1}^{N_\theta} \bar{G}_i \frac{\partial G_i}{\partial \theta_k}.$$

This method is more efficient and stable than calculating the derivatives of the implicit functions $\mathscr{M} \to \theta(\mathscr{M})$ by directly differentiating the calibration step either by bumping or by applying AAD. This is because the $G(\mathscr{M}, \theta)$ in Equation (2.4.14) are explicit functions of the market and model parameters, which are easy to compute and differentiate. Moreover, by avoiding the numerical noise produced by the finite difference approximation to the calibration procedure, the accuracy of the sensitivities is improved when compared with the bumping scheme.

2.5 Numerical results

In this section, we present the numerical results arising from the pricing and the calibration of the BK model shown in Equation (2.2.4) for the stochastic instantaneous hazard rate, $h_t = \exp(x_t)$, that satisfies

$$\mathrm{d}\ln(h_t) = \kappa(t)\left(\mu(t) - \ln(h_t)\right)\mathrm{d}t + \sigma(t)\mathrm{d}W_t.$$

We fix the mean-reversion rate $\kappa = 0.01$ and assume that $\mu(t)$ and $\sigma(t)$ are left-continuous, piecewise constant functions. As shown in Section 2.3.2, we calibrate $\mu(t)$ and $\sigma(t)$ to a set of survival probabilities implied from liquid CDS prices and a set of (co-terminal) CDS option prices.

We compute the survival probabilities in terms of a piecewise constant hazard rate function, $\lambda^{\mathrm{mkt}}(t)$, with L knot points $(\lambda_1^{\mathrm{mkt}}, \ldots, \lambda_L^{\mathrm{mkt}})$ at times $(T_1, \ldots, T_L)$. These are determined, for convenience, on the same time grid with equally-spaced intervals $\Delta T = T_{i+1} - T_i = 0.5$, for $i = 1, \ldots, L-1$, as utilised for the mean-reversion level and volatility functions. The market survival probabilities, as seen at $t_0 = 0$, are then given by

$$Q^{\mathrm{mkt}}(t_0, T_i) = \exp\left[-\int_{t_0}^{T_i} \lambda^{\mathrm{mkt}}(u)\mathrm{d}u\right] = \prod_{j=1}^{i} \exp\left[-\lambda_j^{\mathrm{mkt}}\Delta T\right].$$

In these numerical examples, we choose the knot points of the hazard rate function to be the same and equal to $\lambda^{\mathrm{mkt}} = s/(1-R)$, where $s = 1\%$ is the the so-called par-spread and $R = 40\%$ is the recovery rate, as set by market practice. Similarly, the CDS option prices are derived using the standard Black formula from a set of market-implied volatilities σ_j^{mkt}, for $j = 1, \ldots, K$, corresponding to the set of option

maturities $(T_1^e, \ldots, T_K^e)$. We have:

$$V^{\text{mkt}}(t_0, T_j^e; T_L) = \Big(c\,\Phi(-d_1) - s\,\Phi(-d_2)\Big)\mathscr{A}^{\text{mkt}}(T_j^e, T_L),$$
$$\mathscr{A}^{\text{mkt}}(T_j^e, T_L) = \left(\sum_{m \in \mathscr{C}(T_j^e, T_L)} Z(t_0, t_m) Q^{\text{mkt}}(t_0, t_m) \Delta t_c\right),$$

where

$$d_1 = \frac{\ln s/c + (\sigma_j^{\text{mkt}})^2 (T_j^e - t_0)/2}{\sigma_j^{\text{mkt}} \sqrt{T_j^e - t_0}},$$
$$d_2 = d_1 - \sigma_j^{\text{mkt}} \sqrt{T_j^e - t_0},$$

and Φ is the standard normal distribution function, and c is the payment rate by the protection buyer for the CDS against which the option can be exercised. Here we set c equal to the par spread s, so as to represent at-the-money option quotes. We assume zero interest rates and set the implied volatilities to all have the same value, $\sigma^{\text{mkt}} = 60\%$. The market data is given in Tables 2.1 and 2.2 and the results of the calibration are shown in Figure 2.1.

Time	0.25	0.5	0.75	1	1.5	2
SP	0.996	0.992	0.988	0.983	0.975	0.967
Time	2.5	3	3.5	4	4.5	5
SP	0.959	0.951	0.943	0.935	0.928	0.920

Table 2.1: Survival probabilities utilised for the calibration of the parameters θ.

Expiry	0.25	0.5	1	1.5	2	2.5	3	4
Price (10^{-2})	0.570	0.719	0.894	0.947	0.926	0.853	0.739	0.417

Table 2.2: Prices of options written on a five-year CDS, which are utilised for the calibration of the parameters θ.

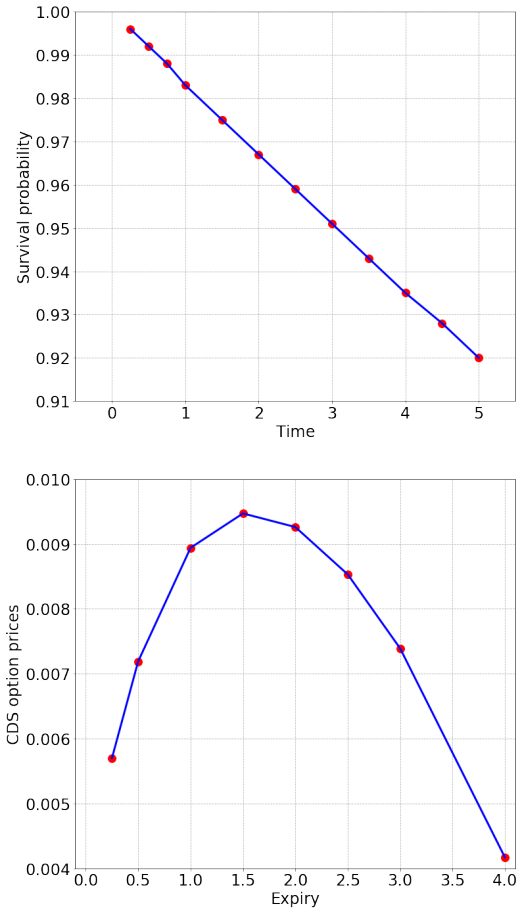

Figure 2.1: Calibration of the BK hazard rate model to a set of market implied survival probabilities and option prices.

2.5.1 AAD versus bumping in the computation of sensitivities

We consider the pricing of a defaultable discount bond with a five-year maturity and a unit redemption value. Its value at time $t_0 = 0$ is given by Equation (2.2.5), and it can be determined either by a backward or forward PDE by setting $V(T, h_T; \theta) = 1$.

To show the reliability of the AAD sensitivities calculation in the PDE framework, Table 2.3 display the sensitivities obtained by the AAD algorithms, described in Section 2.4.2 and 2.4.3, and by means of one-sided finite-difference approximations (bumping) with a perturbation (bump) size of 10^{-5}. As expected, the results obtained with both the AAD version of the backward and forward PDE are consistent with the ones obtained by bumping, with minor differences due to discretisation errors and the finite precision of the finite-difference approach.

	Bwd PDE (AAD)	Fwd PDE (AAD)	Bwd PDE (FD)	Fwd PDE (FD)
μ_1	-1.8e-4	-1.8e-4	-1.8e-4	-1.8e-4
μ_2	-1.7e-4	-1.6e-4	-1.7e-4	-1.6e-4
μ_3	-1.6e-4	-1.6e-4	-1.6e-4	-1.6e-4
μ_4	-1.5e-4	-1.5e-4	-1.5e-4	-1.5e-4
μ_5	-2.7e-4	-2.7e-4	-2.7e-4	-2.7e-4
μ_6	-2.3e-4	-2.3e-4	-2.3e-4	-2.3e-4
μ_7	-2.0e-4	-2.0e-4	-2.0e-4	-2.0e-4
μ_8	-1.6e-4	-1.6e-4	-1.6e-4	-1.6e-4
μ_9	-1.3e-4	-1.3e-4	-1.3e-4	-1.3e-4
μ_{10}	-9.1e-5	-9.1e-5	-9.2e-5	-9.1e-5
μ_{11}	-5.6e-5	-5.5e-5	-5.6e-5	-5.5e-5
μ_{12}	-1.8e-5	-1.8e-5	-1.9e-5	-1.8e-5

	Bwd PDE (AAD)	Fwd PDE (AAD)	Bwd PDE (FD)	Fwd PDE (FD)
σ_1	-7.8e-3	-7.8e-3	-7.5e-3	-7.8e-3
σ_2	-0.011	-0.01	-0.011	-0.01
σ_3	-0.016	-0.016	-0.017	-0.016
σ_4	-0.015	-0.015	-0.015	-0.015
σ_5	-0.013	-0.013	-0.013	-0.013
σ_6	-0.012	-0.011	-0.012	-0.011
σ_7	-9.9e-3	-9.9e-3	-0.01	-9.9e-3
σ_8	-0.019	-0.019	-0.02	-0.019

Table 2.3: Parameters sensitivities of a five-year defaultable discount bond computed by the AAD version of the forward (Fwd) and backward (Bwd) PDEs and by finite-difference (FD) approximations with a bump size of 10^{-5}.

Similarly, in Table 2.4 we compare the sensitivities results for a CDS option and a bond option using a combination of the forward and backward PDE approaches described in Section 2.3.2. Here we consider a two-year at-the-money swaption written on a five-year CDS, and a two-year European-style call option issued on the five-year defaultable bond with a strike of 0.75.

As in the previous example, these results confirm that the AAD approach provides accurate estimates of the sensitivities when benchmarked with the standard finite-difference approach.

	CDS option		Bond option	
	AAD	FD	AAD	FD
μ_1	-2.3e-5	-2.3e-5	-1.2e-4	-1.1e-4
μ_2	-2.3e-5	-2.3e-5	-1.1e-4	-1.1e-4
μ_3	-2.3e-5	-2.3e-5	-1.1e-4	-1.1e-4
μ_4	-2.3e-5	-2.3e-5	-1.1e-4	-1.1e-4
μ_5	-4.5e-5	-4.5e-5	-2.1e-4	-2.1e-4
μ_6	-4.5e-5	-4.4e-5	-2.1e-4	-2.1e-4
μ_7	-4.1e-5	-4.0e-5	-1.9e-4	-1.9e-4
μ_8	-3.4e-5	-3.3e-5	-1.6e-4	-1.6e-4
μ_9	-2.6e-5	-2.6e-5	-1.2e-4	-1.2e-4
μ_{10}	-1.9e-5	-1.9e-5	-8.8e-5	-8.8e-5
μ_{11}	-1.1e-5	-1.2e-5	-5.3e-5	-5.4e-5
μ_{12}	-3.7e-6	-3.9e-6	-1.7e-5	-1.8e-5

	CDS option		Bond option	
	AAD	FD	AAD	FD
σ_1	3.5e-4	3.5e-4	-4.3e-3	-4.1e-3
σ_2	5.0e-4	5.0e-4	-6.0e-3	-6.0e-3
σ_3	8.3e-4	8.3e-4	-0.01	-0.01
σ_4	8.3e-4	8.3e-4	-0.01	-0.01
σ_5	7.6e-4	8.2e-4	-0.01	-0.01
σ_6	-2.6e-3	-2.6e-3	-0.011	-0.011
σ_7	-2.3e-3	-2.3e-3	-9.7e-3	-9.7e-3
σ_8	-4.3e-3	-4.3e-3	-0.019	-0.019

Table 2.4: Parameters sensitivities of a CDS option and defaultable discount bond option computed by means of AAD and by FD approximations with a bump size of 10^{-5}.

The efficiency of AAD is shown in Figure 2.2. We plot the cost of computing the sensitivities of a defaultable discount bond with respect to the knot points of the mean-reversion level μ_i, $i = 1,\ldots,L$ and volatility σ_i, $i = 1,\ldots,K$, relative to the cost of performing a *single* valuation. As illustrated in Figure 2.2, for both, the AAD version of the backward and forward PDE scheme, the calculation of the sensitivities can be performed for about 3.3 times the cost of computing the value of the bond, which is well within the theoretical bound shown in (2.4.3). In contrast, the cost of bumping is, in general, $(1+N_\theta)$ times the cost of as single valuation, in other words, over 20 times the cost of computing the value of the bond in this case.

Similarly, the cost of computing the sensitivities of a bond and CDS option by AAD is also bounded, but the cost of the bumping scheme is proportional to the number of parameters.

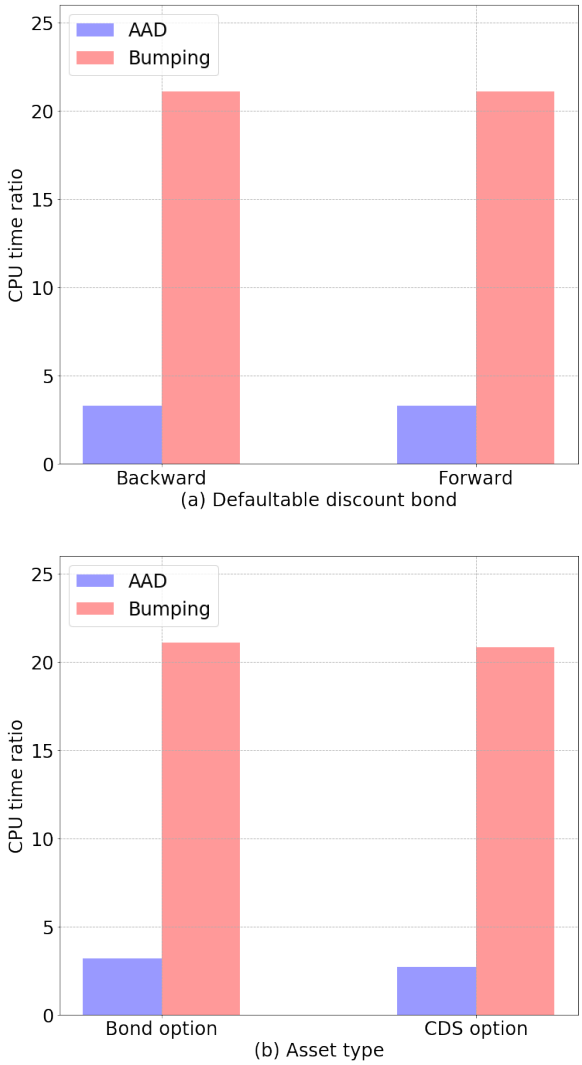

Figure 2.2: Cost of computing the sensitivities for a defaultable discount bond, CDS option and defaultable discount bond option, relative to the cost of a single valuation.

Furthermore, and as shown in Figure 2.3, the overall cost of running the AAD scheme to obtain all the sensitivities relative to the cost of computing the option value through a single valuation of the PDE scheme is independent of the number of sensitivities so that the computational gains, when compared to the bumping scheme, *increase* with the number of sensitivities[4].

[4]In these examples we have included the sensitivities with respect to the knot points of the mean-

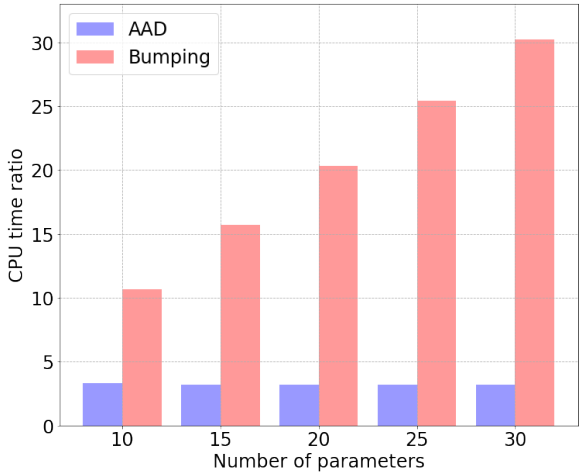

Figure 2.3: Cost of computing the sensitivities for a defaultable discount bond option, relative to the cost of a single valuation, as a function of the number of sensitivities.

2.5.2 Calibration and the implicit function theorem

As described in Section 2.4.4, the sensitivities with respect to the internal model parameter θ can be converted into more practically relevant sensitivities with respect to the market parameters, M. This can be achieved by combining sensitivities obtained with the AAD version of the forward and backward PDE executed during the calibration of the model parameters and also by invoking the IFT. As an illustration, we can again consider the two-year option, on a five-year defaultable bond, with strike price 0.6. By making use of the scheme described in Section 2.4.4, the sensitivities with respect to the model parameters $\bar{\theta} = \partial V/\partial\theta$ can be transformed into the sensitivities with respect to the market observables $\partial V/\partial\mathscr{M}$.

In this case, they are the sensitivities with respect to the implied hazard rates and the CDS options' implied volatilities, which are used for the calibration in Figure 2.1. Table 2.5 displays the bond option market sensitivities obtained by converting the model sensitivities in Table 2.4 by means of the AAD-IFT approach, and shows the satisfactory agreement with those obtained by the standard finite-difference approach.

---

reversion speed function.

Market observables	IFT	FD
λ_1^{mkt}	-0.088	-0.078
λ_2^{mkt}	-0.083	-0.089
λ_3^{mkt}	-0.087	-0.081
λ_4^{mkt}	-0.084	-0.086
λ_5^{mkt}	-0.172	-0.168
λ_6^{mkt}	-0.170	-0.174
λ_7^{mkt}	-0.454	-0.453
λ_8^{mkt}	-0.454	-0.452
λ_9^{mkt}	-0.455	-0.455
$\lambda_{10}^{\text{mkt}}$	-0.454	-0.453
$\lambda_{11}^{\text{mkt}}$	-0.455	-0.455
$\lambda_{12}^{\text{mkt}}$	-0.454	-0.454

Market observables	IFT	FD
σ_1^{mkt}	7.2e-07	7.9e-07
σ_2^{mkt}	2.5e-06	2.8e-06
σ_3^{mkt}	4.2e-06	-5.1e-06
σ_4^{mkt}	-6.4e-06	-5.7e-06
σ_5^{mkt}	0.001	0.001
σ_6^{mkt}	-2.5e-05	-2.1e-5
σ_7^{mkt}	-2.2e-05	-1.9e-05
σ_8^{mkt}	-1.5e-05	-1.2e-05

Table 2.5: Sensitivities of a defaultable discount bond option with respect to the market observables as obtained with AAD and by FD approximations with a bump size of 10^{-5}.

The notable computational gains that can be achieved with the AAD-IFT scheme are shown in Figure 2.4 (left, yellow column). We plot the ratio of time necessary to convert the model sensitivities into market sensitivities by both, the ADD-IFT approach and standard finite differences, relative to the cost of performing a *single* calibration and valuation.

For this application, the time necessary to compute the Jacobian $\frac{\partial \theta}{\partial \mathscr{M}}$ and model parameter sensitivities $\frac{\partial V}{\partial \theta}$ by the AAD-IFT approach is less than 1% of the amount of time necessary to perform a single calibration and valuation, thus resulting in a 3 times orders of magnitude speed-up with respect to standard bumping. This staggering difference in efficiency is due in part to the computationally intensive

calibration procedure of the BK model involving, as described in Section 2.3.2, a multidimensional root-search over the instantaneous volatilities.

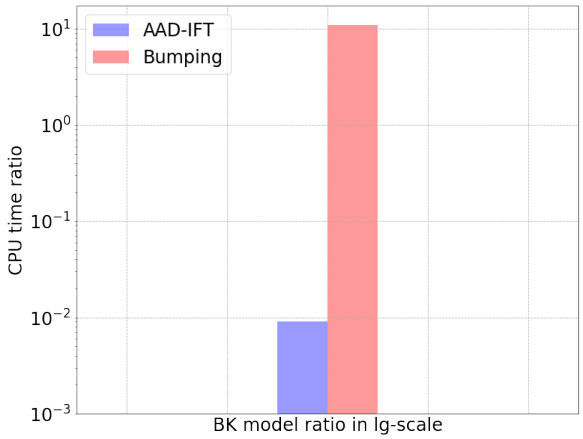

Figure 2.4: Cost of computing the market parameter sensitivities for a defaultable discount bond option relative to the cost of a single calibration and valuation for the BK model .

2.6 Conclusions

AAD can be applied to efficiently compute price sensitivities of generic financial securities as numerical solutions of PDEs. AAD is beneficial for computing the sensitivities with PDE-based calibration algorithms. By using a practically-relevant example, we show that by combining the adjoint versions of the algorithms for the numerical solution of backward and forward PDEs, along with the IFT, one can avoid repeating the calibration algorithm or the AAD-version of the calibration routine. This allows for the calculation of all price sensitivities at an additional computational cost that is a fraction of the cost of computing the portfolio P&L, thus typically resulting in procedures that are orders of magnitude faster than standard finite-difference approaches. We expect the insights presented in this chapter will be of significant importance for the efficient implementation of pricing and hedging approaches in practice.

Chapter 3

AAD and least-square MC: fast Bermudan-style options and XVA *Greeks*

3.1 Overview

The financial crisis of 2007-2008 has changed the financial world in many aspects. All over-the-counter (OTC) contracts are currently cleared in three different ways:

i) bilaterally non-collateralised trades;

ii) bilaterally collateralised trades; and

iii) centrally cleared trades.

After the financial crisis, market participants have been required by regulators to quantify the risks of default, funding, capital, liquidity and all other various potential risk factors during the traditional pricing procedure for all these OTC transactions. The impact caused by these factors are captured by the well-known valuation adjustments, collectively known as XVA, see [10] and [31].

Each component in the XVA is difficult to evaluate because it involves computing the expectation of a function together with all the future conditional exposures of the transaction until expiry. Even in the simplest case, namely unilateral credit valuation adjustment (CVA), analytical solutions are difficult to find. This will be

complicated even more if one consider the dependences between the counterparty default and the portfolio exposures, which are collectively known as wrong-way risk (WWR) or right-way risk (RWR), see [11], [48] and [56].

Generally, the valuation of XVA is implemented through the use of MC methods. However, XVA requires two layers of MC simulations: the first layer of MC simulation is necessary to obtain the exposures conditional on future time steps; and the second layer is necessary to compute the mean of a function of the conditional exposures. Together, this is known as the nested MC algorithm.

Although the evaluation of the XVA is difficult in terms of computational efficiency, the finalised Basel III directives emphasise the risk management of XVA, see [6]. As mentioned in Chapter 2, the efficient calculation of the risk factor sensitivities of financial derivatives, also known as the *Greeks*, is an essential component of modern risk management practices. Similar to the risk management of credit derivatives discussed in Chapter 2, the traditional approach for the calculation of the *Greeks* for the XVA is also the *bump and reval* or *bumping* technique. This comes at a significant computational cost, since it generally requires repeating the calculation of the P&L of a portfolio under hundreds of market scenarios in order to form finite-difference estimators. As a result, even after deploying vast amounts of computer power, in most cases these calculations cannot be completed in a practical amount of time.

Similar to the valuation of American options by MC methods in [24], [60] and [71], the nested MC algorithm for XVA valuation can also be replaced by the regression-based MC algorithm, see page 218 of [25]. The regression-based MC algorithm for XVA computations replaces the first layer of the nested MC algorithm. Its purpose is to compute the conditional exposures by using a regression step. The XVA computation will benefit from the regression-based MC algorithm due to it requiring fewer computational resources and it is efficient to implement. However, evaluating sensitivities for XVA are still computationally expensive if one resorts to the traditional finite-difference method.

Nevertheless, AAD has been proven to be effective for speeding up the calcu-

lation of risk factor sensitivities with numerical PDE methods as shown in Chapter 2. In this chapter, we show how the AAD algorithm can be used in the regression-based MC method, and we go on to demonstrate the efficiency as well as the accuracy that the AAD method can achieve in evaluating both the *Greeks* of American options and XVA.

We conclude by emphasizing the main contributions of this chapter. We present the AAD algorithm for the regression-based Monte Carlo method, which can be used for the path-dependent derivatives, such as Bermudan-style contracts, and XVA pricing. The application of AAD to Monte Carlo algorithm can be found in [38], who applied AAD to LIBOR market models. AAD and regression-based Monte Carlo methods have jointly been used in [32] to compute CVA sensitivities. However, only vanilla contracts are considered and it is without taking into account the sensitivities arising from the regression basis functions. In contrast, we introduce the smoothing method in Section 3.3.1 to collect the sensitivities from the indicator function, making collecting sensitivities emerging from the regression basis functions possible. This changes the situation that the sensitivities from indicator functions are usually neglected in practice. The smoothing method will be presented in Section 3.3.1 of this chapter, which is based on our publication [22]. Our methods are also cited in [36] as a way of collecting sensitivities arising from the regression functions. By the smoothing method and the AAD algorithm for regression-based Monte Carlo, we show with examples how this can improve the efficiency of Bermudan-style option *Greeks* and XVA sensitivities computation.

3.2 Valuation of Bermudan-style options and XVA by regression-based MC

In this section, we present the regression-based MC algorithm for the valuation of Bermudan-style options, see for example [24], [60] and [71], and XVA, see for example page 218 of [25].

3.2.1 Bermudan-style options

While European-style options can be exercised only at maturity, Bermudan-style options can be exercised on multiple discrete dates up to the trade's expiry.

We denote by $T_1, \ldots, T_M$ the exercise dates of the option and define $\mathscr{D}(t) = \{T_m \geq t\}$. We denote by $\eta(t)$ the smallest integer such that $T_{\eta(t)+1} > t$. An exercise policy is represented mathematically by a stopping time taking values in $\mathscr{D}(t)$. We denote by $\mathscr{T}(t)$ the set of stopping times taking values in $\mathscr{D}(t)$.

A rational investor will exercise the option that he holds in such a way as to maximise its economic value. As a result, the value of a Bermudan-style option is the supremum of the option's value over all possible exercise policies. With the notation introduced above, the value of a Bermudan-style option at time t can thus be expressed by

$$\frac{V(t)}{N(t)} = \sup_{\tau \in \mathscr{T}(t)} \mathbb{E}\left[\frac{E(\tau)}{N(\tau)}\right] , \tag{3.2.1}$$

where $E(t)$ is the exercise value of the option, and $N(t)$ is the chosen *numéraire*[1]. In this equation, $V(t)$ is to be interpreted as the early-exercise value of the option conditional on exercise not having taken place *strictly before* time t.

A useful concept is the *hold value* of the Bermudan-style option. We denote by $H(t)$ the value of the Bermudan-style option when the exercise dates are restricted to $\mathscr{D}(T_{\eta(t)+1})$, that is

$$\frac{H(t)}{N(t)} = \mathbb{E}_t\left[\frac{V\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)}\right] , \tag{3.2.2}$$

where we have assumed, for simplicity of exposition, no cashflow between t and $T_{\eta(t)+1}$. The option holder, following an optimal exercise policy, will exercise his option if the exercise value is larger than the hold value, i.e.

$$V\left(T_{\eta(t)}\right) = \max\left(E\left(T_{\eta(t)}\right), H\left(T_{\eta(t)}\right)\right) . \tag{3.2.3}$$

This, when combined with Equation (3.2.2), leads to the so-called *dynamic pro-*

[1] In the following, we will set $N(0) = 1$ for simplicity of notation.

gramming formulation:

$$\frac{H(t)}{N(t)} = \mathbb{E}_t \left[\max \left(\frac{E\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)}, \frac{H\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)} \right) \right] , \tag{3.2.4}$$

for $T_\eta \leq t < T_{\eta+1}$, and $\eta = 1,\ldots,M-1$. Starting from the terminal condition $H(T_M) \equiv 0$, Equation (3.2.4) defines a backward iteration in time for $H(t)$. By definition, this is also equal to $V(t)$ if t is not an exercise date, i.e. if $T_{\eta(t)} < t < T_{\eta(t)+1}$. Conversely, if t is an exercise date, that is $t = T_{\eta(t)}$, then $V(T_{\eta(t)}) = \max(E(T_{\eta(t)}), H(T_{\eta(t)}))$.

The dynamic programming formulation above implies that the stopping time, defining the optional exercise date as seen at time t, is given by

$$\tau^\star = \inf[T_m \geq t : E(T_m) \geq H(T_m)] . \tag{3.2.5}$$

The optimal exercise strategy defined by Equation (3.2.5) requires the computation of the hold value $H(t)$, $m = \eta(t)+1,\ldots,M-1$. In a setting in which the underlying risk factor process $\{X(t)\}_{0\leq t\leq T}$ is a generic k-dimensional Markov process, the hold value $H(t)$ is a function of the state vector at time t. That is,

$$H_t(x) := \mathbb{E}\left[\frac{N(X(t))}{N(X(T_{m+1}))} V\left(X\left(T_{m+1}\right)\right) \middle| X(t) = x \right] . \tag{3.2.6}$$

When the dimension of the Markov process, k, is small enough, the conditional expectation value in Equation (3.2.6) can be computed in a straightforward way by discretising the risk-factor process and performing standard backward induction on a tree or a grid, or by discretising an associated PDE. Here we refer to, for example, page 135 of [74]. However, the complexity of grid-based calculations is exponential in the dimension of the Markov process and numerical implementations become infeasible when $k \geq 4$. Here we mention that one may also use the neural network method to solve high dimensional PDEs numerically, see [23].

As we will review in Section 3.2.3, regression-based MC techniques provide an effective way of computing conditional expectation values of the form (3.2.6).

3.2.2 XVA

We consider next the computation of the CVA and the debt valuation adjustment (DVA) as the main measures of a dealer's counterparty credit risk, see for example [10] and [31]. For a given portfolio of trades with the same investor or institution, the CVA (respectively DVA) aims to capture the expected loss (respectively gain) associated with the counterparty (respectively dealer) defaulting in a situation in which the position, netted for any collateral posted, has a positive mark-to-market for the dealer (respectively counterparty).

This can be evaluated at time $T_0 = 0$ by

$$\text{XVA} = -\mathbb{E}\left[\mathbb{I}(\tau_c \leq T)\frac{L_c}{N(\tau_c)}\left(V(\tau_c)\right)^+ + \mathbb{I}(\tau_d \leq T)\frac{L_d}{N(\tau_d)}\left(V(\tau_d)\right)^-\right], \quad (3.2.7)$$

where τ_c (respectively τ_d) is the default time of the counterparty (*resp. the dealer*), $V(t)$ is the net present value of the portfolio or *netting set* at time t from the dealer's point of view, which is the so-called conditional future exposure, L_c (respectively L_d) is the loss given default of the counterparty (respectively the dealer), and $\mathbb{I}(\tau_c \leq T)$ (respectively $\mathbb{I}(\tau_d \leq T)$) is the indicator that the counterparty's (respectively dealer's) default happens before the longest deal maturity, T, in the portfolio. Here, for simplicity of notation, we consider the unilateral CVA and DVA, since the generalisation to the bilateral formulation, see for example page 15 of [31], is straightforward.

Equation (3.2.7) is typically computed on a discrete time grid of "horizon dates" $0 = T_0 < T_1 < \ldots < T_M = T$. For instance, we may have

$$\begin{aligned}\text{XVA} \approx -\sum_{m=1}^{M}\mathbb{E}\Big[&L_c(\text{SP}_c(T_{m-1}) - \text{SP}_c(T_m))\frac{\left(V(T_m)\right)^+}{N(T_m)} \\ &+L_d(\text{SP}_d(T_{m-1}) - \text{SP}_d(T_m))\frac{\left(V(T_m)\right)^-}{N(T_m)}\Big],\end{aligned} \quad (3.2.8)$$

where $\text{SP}_c(t)$ (respectively $\text{SP}_d(t)$) is the survival probability of the counterparty (respectively the dealer) up to time t, for example, conditional on a realisation of the default intensity or hazard rate process in a Cox framework, see [52]. Here

we assume that the default times, τ_c and τ_d, are conditionally independent of the portfolio values, V_{τ_c} and V_{τ_d}, respectively, for simplicity in later context. i.e. $\mathbb{I}(\tau_c \leq T)$ is independent of $(V(\tau_c))^+$ conditional on $\mathscr{F}_{\tau_c}$ and $\mathbb{I}(\tau_d \leq T)$ is independent of $(V(\tau_d))^+$ conditional on $\mathscr{F}_{\tau_d}$.

In general, the right hand side of Equation (3.2.8) depends on several correlated random market factors, including interest rates, recovery amounts and all the market factors the net conditional future exposure of the portfolio, $V(t)$, depends on. As such, its calculation typically requires a MC simulation. In the k-dimensional Markov setting introduced above, the conditional future exposure, $V(T_m)$, is a function $V(X(T_m))$ of the state vector at time T_m. However, only for vanilla securities and simple models for the evolution of the risk factors, such conditional future exposures can be expressed in closed form. In order to illustrate how the conditional future exposure can be computed by means of regression-based MC, we consider a specific example in which the underlying portfolio contains a basket of Bermudan-style options.

3.2.3 Conditional expectation values and Bermudan-style options by regression

The least square regression, see page 44 of [35] for example, with N observations of the predictor variables $x_1, x_2, \cdots x_d$ and response variable y is to fit the model

$$y = \beta_1 x_1 + \cdots + \beta_d x_d + \varepsilon,$$

where ε is the uncorrelated noise with zero mean and β_i for $i = 1, 2 \cdots, d$ are the regression coefficients. That is, we need to evaluate

$$\min_{\beta_1, \cdots, \beta_d} \quad \sum_{n=1}^{N} \left[y^{(d)} - \left(\beta_1 x_1^{(d)} + \cdots + \beta_d x_d^{(d)} \right) \right]^2$$

where we denote the n-th observation of the predictor variables by $x_1^{(n)}, x_2^{(n)}, \cdots x_d^{(n)}$ and the response variable by $y^{(n)}$. This can be written in matrix form as follows:

$$\min_{\beta_1,\cdots,\beta_d} \quad Y^T Y - 2\beta^T XY + \beta^T XX^T \beta,$$

where

$$Y = \left[y^{(1)}, y^{(2)}, \cdots, y^{(N)}\right]^T,$$
$$\beta = \left[\beta^{(0)}, \beta^{(1)}, \cdots, \beta^{(d)}\right]^T,$$
$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(N)} \\ \cdots & \cdots & \cdots & \cdots \\ x_d^{(1)} & x_d^{(2)} & \cdots & x_d^{(N)} \end{bmatrix}.$$

By the quadratic optimisation method, the optimal β is given by

$$\beta = (XX^T)^{-1} XY. \tag{3.2.9}$$

In the context of the valuation of Bermudan-style options, the hold value, shown in Equation (3.2.6), on an exercise date T_m is assumed to be of the form

$$\hat{H}_m(x) = \beta_m^T \psi(x), \tag{3.2.10}$$

where $\psi(x) = (\psi_1(x), \ldots, \psi_d(x))^T$ is a vector of d basis functions and $\beta_m = (\beta_{1m}, \ldots, \beta_{dm})^T$ is the vector of coefficients to be determined by regressing $V(X(T_{m+1}))\frac{N(X(T_m))}{N(X(T_{m+1}))}$ on $\psi(X(T_{m+1}))$. By Equation (3.2.9), we have

$$\beta_m = \Psi_m^{-1} \Omega_m \,, \tag{3.2.11}$$

where we define the $d \times d$ matrix,

$$\Psi_m = \psi(X(T_m))\psi^T(X(T_m)) \,,$$

and the $d \times 1$ vector

$$\Omega_m = \frac{N(X(T_m))V(X(T_{m+1}))}{N(X(T_{m+1}))}\psi(X(T_m)) \ .$$

These equations provide a straightforward way to compute the regression coefficients, β_m, by substituting Ψ_m and Ω_m with their sample average over N_{MC} MC replications. This can be achieved by the following steps:

(R1) Simulate N_{MC} independent MC paths, $X_m^{(n)}$, of $X(T_m)$ by the recursion

$$X_{m+1}^{(n)} = F\left(T_m, X_m^{(n)}; \theta\right)$$

for $m = 0, \ldots, M-1$, and $n = 1, \ldots, N_{\mathrm{MC}}$. Here, F is a function based on the chosen models for the risk factors, and θ is a vector of model parameters.

(R2) For $n = 1, \ldots, N_{\mathrm{MC}}$, compute the terminal payoff of the contract by setting

$$V_M^{(n)} = E_M^{(n)} \ ,$$

where $E_M := E(X_M^{(n)})$ is the final exercise value of the option.

(R3) Apply the following backward induction steps for $m = M-1, \ldots, 1$:

(a) Compute the MC sample average[2] of Ψ_m and Ω_m by

$$\Psi_m = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} (\psi_m^{(n)})^T \ , \tag{3.2.12}$$

$$\Omega_m = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} \frac{N_m^{(n)} V_{m+1}^{(n)}}{N_{m+1}^{(n)}} \ , \tag{3.2.13}$$

where $\psi_m^{(n)} := \psi\left(X_m^{(n)}\right)$ and $N_m^{(n)} := N(X_m^{(n)})$.

(b) Compute the regression coefficients β_m by matrix inversion and multi-

[2]Here and in the following, in order to keep the notation simple, we do not introduce different symbols for expectations and their respective sample averages.

plication, i.e.

$$\beta_m = \Psi_m^{-1}\Omega_m. \tag{3.2.14}$$

(c) For the estimate of the hold value, $H_m^{(n)} := H_m\left(X_m^{(n)}\right)$, set

$$H_m^{(n)} = \beta_m^T \psi_m^{(n)} ,$$

for $n = 1,\ldots,N_{\mathrm{MC}}$.

(d) For the estimate of the Bermudan-style option value at time T_m, set

$$V_m^{(n)} = \max\left(E_m^{(n)}, H_m^{(n)}\right) , \tag{3.2.15}$$

where $E_m^{(n)} := E(X_m^{(n)})$ is the exercise value at time T_m, for $n = 1,\ldots,N_{\mathrm{MC}}$.

(R4) Compute the MC estimate of the Bermudan-style option at time T_0 by

$$V_0 = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \frac{V_1^{(n)}}{N_1^{(n)}} . \tag{3.2.16}$$

A modification of this algorithm was proposed by [60] and it entails replacing Equation (3.2.15) in Step (R3) (d) with

$$V_m^{(n)} = \begin{cases} E_m^{(n)} & \text{if } E_m^{(n)} > H_m^{(n)} , \\ N_m^{(n)} V_{m+1}^{(n)} / N_{m+1}^{(n)} & \text{otherwise,} \end{cases} \tag{3.2.17}$$

which, in the examples considered, was shown to lead to more accurate results. In the following, however, for simplicity of exposition, we will consider the estimator in Equation (3.2.15).

Since our main goal of this chapter is to present the AAD algorithm for regression-based Monte Carlo, we refer to the following literature for the stability and the convergence of the least square Monte Carlo method. This method was

introduced in [71] where it is shown that the estimator, V_0, converges for $n \to \infty$ to the true value, $V(0)$, provided that the representation shown in Equation (3.2.10) holds exactly. Moreover, [28] studied the convergence of the algorithm with respect to the number of basis functions and the Monte Carlo paths. They proved the almost sure convergence of the least square Monte Carlo algorithm and obtained the rate of convergence with increasing number of Monte Carlo paths. The rate of convergence with respect to the number of basis functions in some specific settings is estimated in [40].

3.2.4 Lower bound algorithm for Bermudan-style options

The hold value obtained by the regression as described in the previous section defines an exercise policy whereby on each exercise date, T_m, the option is exercised if

$$E(X(T_m)) > \beta_m^T \psi(X(T_m)) \ . \tag{3.2.18}$$

Such policy, being an approximation to the solution of the dynamic programming equation (3.2.4), will in general correspond to a suboptimal stopping time. As a result, when utilised in a second, independent, MC simulation, the exercise policy obtained by regression, will result in a lower-bound estimator for the Bermudan-style option value. The corresponding algorithm can be schematically described as follows.

For each MC replication indexed by $n = 1, \ldots, N_{\mathrm{MC}}$ perform steps (L1) to (L4) below:

(L1) Simulate the path $X_m^{(n)}$ of the risk factor vector $X(T_m)$ as in (R1).

(L2) For $m = 1, 2, ..., M-1$, compute the approximate hold value of the option at time T_m using the associated regression vector β_m, and regression functions ψ, by

$$H_m^{(n)} = \beta_m^T \psi_m^{(n)} \tag{3.2.19}$$

with the hold value at expiry T_M set to zero.

(L3) Compute the pathwise estimator for the discounted cash-flows of the option

$$P^{(n)} = \sum_{m=1}^{M} \left[\mathbb{I}^{(n)}(t_1, t_m) \mathbb{I}\left(E_m^{(n)} > H_m^{(n)}\right) \frac{E_m^{(n)}}{N_m^{(n)}} \right] , \tag{3.2.20}$$

where

$$\mathbb{I}^{(n)}(t_1, t_m) = \left(\prod_{i=1}^{m-1} \mathbf{1}\left(H_i^{(n)} > E_i^{(n)}\right) \right) \tag{3.2.21}$$

and the convention $\mathbb{I}^{(n)}(t_1, t_1) = 1$.

(L4) Compute the MC estimate of the Bermudan-style option at time $T_0 = 0$ by

$$V_0 = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} P^{(n)} . \tag{3.2.22}$$

3.2.5 Evaluating XVA by regression

As described in Section 3.2.2, the calculation of the XVA in Equation (3.2.8) requires the conditional future exposure $V(t)$ on a set of dates determined by a discretised time grid $T_1 \dots T_M$. The regression algorithm described in the previous section can be easily adapted to compute such quantity. Indeed, the conditional value of each of the options contained in the netting set can be obtained using the same least-square procedure. Once the regression algorithm is completed, we can use the regression functions to compute the hold value of each option in the portfolio on the discretised time grid by $H_m = \beta_m^T \psi_m$. If the discretised time T_m is not an exercise opportunity for the option under consideration, then this is its conditional future exposure. Conversely, the conditional future exposure is obtained by comparing the hold value to the exercise value as in Equations (3.2.15) and (3.2.17). These observations translate in the following algorithm.

For each MC replication indexed by $n = 1, \dots, N_{\mathrm{MC}}$, perform steps (X1) to (X3) below:

(X1) Simulate the path $X_m^{(n)}$ of the risk factor vector by the recursion:

$$X_{m+1}^{(n)} = F\left(T_m, X_m^{(n)}; \theta\right) ,$$

for $m = 0, \ldots, M-1$. Also, simulate the path of the counterparty's and the dealer's default intensity, $\lambda_m^{d,c} = \lambda^{d,c}(T_m)$, by the recursions

$$\lambda_{m+1}^{\mathrm{c},(n)} = G_c\left(T_m, \lambda_m^{\mathrm{c},(n)}; \theta\right),$$
$$\lambda_{m+1}^{\mathrm{d},(n)} = G_d\left(T_m, \lambda_m^{\mathrm{d},(n)}; \theta\right),$$

for $m = 0, \ldots, M-1$, where G_c (respectively G_d) is the function describing the dynamics of the counterparty's (respectively the dealer's) hazard rate.

(X2) Compute the discretised pathwise survival probabilities for the counterparty and the dealer by

$$\mathrm{SP}_m^{\mathrm{c},(n)} = \exp\left[-\sum_{j=0}^{m-1} \lambda_j^{\mathrm{c},(n)}\left(T_{j+1} - T_j\right)\right] , \tag{3.2.23}$$

$$\mathrm{SP}_m^{\mathrm{d},(n)} = \exp\left[-\sum_{j=0}^{m-1} \lambda_j^{\mathrm{d},(n)}\left(T_{j+1} - T_j\right)\right] , \tag{3.2.24}$$

for $m = 1, 2, ..., M$.

(X3) For $m = 1, 2, ..., M-1$, approximate the hold value of the p-th option in the portfolio at time T_m using the associated regression vector, $\beta_{p,m}$, and regression functions ψ_p by

$$H_{p,m}^{(n)} = \beta_{p,m}^T \psi_{p,m} , \tag{3.2.25}$$

for $p = 1, \ldots, P$. The hold value at the expiry date, T_M, is set to zero. The con-

ditional expected value of the portfolio is given by $V_m^{(n)} = \sum_{p=1}^{P} V_{p,m}^{(n)}$, where

$$V_{p,m}^{(n)} = \begin{cases} \max\left\{H_{p,m}^{(n)}, E_{p,m}^{(n)}\right\}, & \text{if } T_m \text{ is an exercise date for the } p\text{-th option} \\ H_{p,m}^{(n)}, & \text{otherwise,} \end{cases} \tag{3.2.26}$$

for $m = 1, 2, ..., M$, where $E_{p,m}^{(n)}$ is the exercise value of the p-th option at time T_m on the n-th path.

(X4) Compute the pathwise XVA by

$$\mathrm{XVA}^{(n)} = -\sum_{m=1}^{M} \left[L_c \left(\mathrm{SP}_{m-1}^{\mathrm{c},(n)} - \mathrm{SP}_{m}^{\mathrm{c},(n)}\right) \frac{\left(V_m^{(n)}\right)^+}{N_m^{(n)}} + L_d \left(\mathrm{SP}_{m-1}^{\mathrm{d},(n)} - \mathrm{SP}_{m}^{\mathrm{d},(n)}\right) \frac{\left(V_m^{(n)}\right)^-}{N_m^{(n)}} \right].$$

(X5) Finally, form the MC estimator,

$$\mathrm{XVA} = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \mathrm{XVA}^{(n)}. \tag{3.2.27}$$

3.3 The AAD algorithm for regression-based MC

As shown in Chapter 2, AAD can be utilised to compute the *Greeks* for derivatives which are priced using a numerical PDE approach. Here, we give the AAD algorithm for the regression-based MC method; one can use this algorithm to compute the *Greeks* for Bermudan-style options and XVA sensitivities can also be calculated numerically.

3.3.1 Function regularisations

In the context of MC methods, [19] shows that AAD allows for the calculation of the sensitivities by differentiating the relevant estimator on a path by path basis. Since a pathwise method is used, the MC estimators must satisfy specific regularity con-

ditions, see page 386 of [39]. For instance, all the functions appearing in each step leading to the computation of the payout estimator must be Lipschitz continuous.

A practical way of addressing non-Lipschitz continuous estimators is to smoothen out the singularities they contain. This can be achieved by observing that in most cases the singularities in the payout functions, although not necessarily implemented as such, can be expressed in terms of Heaviside functions. For instance, the payoff function of a digital option is,

$$P(X(T)) = \mathbb{I}(X(T) > K) \,,$$

while the payoff of a knock-out, path-dependent option with barrier monitored at times $T_1, \ldots, T_M$ is of the form

$$P(X(T_1), \ldots, X(T_M)) = \prod_{m=1}^{M} \mathbb{I}(X(T_m) > B_m),$$

where B_m is the the barrier level at time T_m.

The singularities in such payoff functions can be regularised by replacing the indicator function with one of its smoothened counterparts. A very common choice, for instance, is to approximate the step function with a "call spread" payoff functions,

$$\mathbb{I}(x > K) \approx \mathscr{H}_\delta^{\mathrm{cs}}(x-K) = \left(\min\left(\frac{x-(K-\delta)}{2\delta}, 1 \right) \right)^+, \tag{3.3.1}$$

where $\delta \ll K$. This is a standard choice for digital options, because it has a useful interpretation in terms of the hedging portfolio of a long and a short position in two calls with strike price $K-\delta$ and $K+\delta$ respectively. Alternatively, one can approximate the indicator function with a cumulative normal density function with zero mean and standard deviation δ, that is,

$$\mathbb{I}(x > K) \approx \mathscr{H}_\delta^{\mathrm{cn}}(x-K) = \int_{-\infty}^{x-K} \frac{\exp\left(-u^2/2\delta^2\right)}{\delta\sqrt{2\pi}} \mathrm{d}u. \tag{3.3.2}$$

Both regularisations give rise to functions that are Lipschitz continuous with respect

to x and can be differentiated in a straightforward manner. In particular, the adjoint regularised Heaviside functions are as follows:

$$\bar{\mathscr{H}}_\delta(x-K,\bar{x}) = \bar{x}\frac{\partial}{\partial x}\mathscr{H}_\delta(x-K)\ ,$$

where $\bar{x}$ is the adjoint of the input variable, as in Section 2.4.1. For the call spread regularisation shown in Equation (3.3.1), we then have

$$\bar{\mathscr{H}}_\delta^{\text{cs}}(x-K,\bar{x}) = \begin{cases} \bar{x}/2\delta & \text{if } K-\delta \le x \le K+\delta, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3.3}$$

and, for the cumulative normal regularisation shown in Equation (3.3.2),

$$\bar{\mathscr{H}}_\delta^{\text{cn}}(x-K,\bar{x}) = \bar{x}\,\phi_\delta(x-K), \tag{3.3.4}$$

where $\phi_\delta(x)$ is the Normal density function with standard deviation δ. When $\delta \to 0$, both Equations (3.3.1) and (3.3.2) give the correct derivative of the Heaviside function in the distributional sense, i.e. the Dirac delta function. However, while approaching this limit, the derivatives of such regularised Heaviside functions are zero or vanishingly small, apart from a very small portion of the sample space where instead they are very large. This leads to notably large variances in the MC sampling of the estimators expressed in terms of such functions, signaling the breakdown of the Lipschitz continuity condition. Hence, the choice of the smoothing parameter δ is necessarily a tradeoff between the bias, which vanishes as $\delta \to 0$, and the statistical errors of the MC sampling, which diverges in the same limit.

In general, the payoff estimator for Bermudan-style options, (3.2.20), is not differentiable with respect to the pathwise value of the approximate exercise boundary $H_m^{(n)}$, and it requires the regularisation described above. A common approximation among practitioners, see for example, [55], is to assume that the exercise boundary implied by the rule specified by Equation (3.2.18) is close to optimality so that the value of the contract is approximatively continuous across the exercise boundary; hence no regularisation is required. Under this assumption, no contribution to the

sensitivities is associated with the perturbations of the exercise boundary, and one can therefore keep the regression coefficients fixed while calculating the sensitivities. As discussed in Section 3.4, depending on the accuracy of the basis functions in representing the exercise boundary, this may or may not be an accurate approximation.

3.3.2 AAD for the lower bound algorithm for Bermudan-style options

The AAD implementation of the lower bound algorithm for Bermudan-style options described in Section 3.2.4, producing the MC estimators for the sensitivities of the estimator shown in Equation (3.2.22) with respect to a set of model parameters θ_k, for $k = 1,\ldots,N_\theta$, specified by

$$\bar{\theta}_k = \frac{\partial V(\theta)}{\partial \theta_k} ,$$

comprises of the adjoint of steps (L1) to (L4) executed backwards for each MC replication, for $n = 1,\ldots,N_{\mathrm{MC}}$. These are:

($\overline{\mathrm{L4}}$) Set the adjoint of the option value $\bar{V} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$ and set

$$\bar{P}^{(n)} = \bar{V}_0 \frac{1}{N_{\mathrm{MC}}} .$$

($\overline{\mathrm{L3}}$) Assuming the indicator functions in the estimator in Equation (3.2.20) have been regularised, as discussed in Section 3.3.1, compute

$$\begin{aligned}
\bar{H}_m^{(n)} &= \bar{\mathscr{H}_\delta}\left(H_m^{(n)} - E_m^{(n)}, \bar{P}^{(n)}\right)\left(J_m^{(n)} - Q_m^{(n)}\right), \\
\bar{E}_m^{(n)} &= \bar{P}^{(n)} R_m^{(n)} + \bar{\mathscr{H}_\delta}\left(E_m^{(n)} - H_m^{(n)}, \bar{P}^{(n)}\right)\left(J_m^{(n)} - Q_m^{(n)}\right), \\
\bar{N}_m^{(n)} &= -\bar{P}^{(n)} \frac{Q_m^{(n)} \mathscr{H}_\delta\left(E_m^{(n)} - H_m^{(n)}\right)}{N_m^{(n)}}
\end{aligned} \tag{3.3.5}$$

for $m = M, \ldots, 1$ and where

$$J_m^{(n)} = \sum_{k=m+1}^{M} \frac{E_k^{(n)}}{N_k^{(n)}} \mathbb{I}\left(E_k^{(n)} > H_k^{(n)}\right) \prod_{j=2, j\neq m}^{k-1} \mathbb{I}\left(H_j^{(n)} > E_j^{(n)}\right) ,$$
$$Q_m^{(n)} = \frac{E_m^{(n)}}{N_m^{(n)}} \mathbb{I}^{(n)}(t_1, t_m) ,$$
$$R_m^{(n)} = \frac{1}{N_m^{(n)}} \mathbb{I}\left(E_m^{(n)} > H_m^{(n)}\right) \mathbb{I}^{(n)}(t_1, t_m) .$$

Initialise the adjoints of the risk factors so that

$$\bar{X}_m^{(n)} = \bar{E}_m^{(n)} \frac{\partial E_m^{(n)}}{\partial X_m^{(n)}} + \bar{N}_m^{(n)} \frac{\partial N_m^{(n)}}{\partial X_m^{(n)}} . \tag{3.3.6}$$

($\overline{\text{L2}}$) For $m = M, \ldots, 1$, initialise the adjoint of the regression coefficients β_m to give

$$\bar{\beta}_m = \sum_{n=1}^{N_{\text{MC}}} \psi_m^{(n)} \bar{H}_m^{(n)} , \tag{3.3.7}$$

as well as the adjoints of the basis functions $\psi_m^{(n)}$,

$$\bar{\psi}_m^{(n)} = \beta_m \bar{H}_m^{(n)} , \tag{3.3.8}$$

and update the adjoints of the state vector

$$\bar{X}_m^{(n)} \mathrel{+}= (\bar{\psi}_m^{(n)})^T \frac{\partial \psi_m^{(n)}}{\partial X_m^{(n)}} , \tag{3.3.9}$$

where we use the standard notation "+=" for the addition assignment operator.

($\overline{\text{L1}}$) For $m = M, \ldots, 0$ compute the adjoint of the risk factor evolution such that

$$\bar{X}_m^{(n)} \mathrel{+}= \bar{X}_{m+1}^{(n)} \frac{\partial F}{\partial X_m^{(n)}} \left(T_m, X_m^{(n)}; \theta\right) , \quad \bar{\theta} \mathrel{+}= \bar{X}_{m+1}^{(n)} \frac{\partial F}{\partial \theta} \left(T_m, X_m^{(n)}; \theta\right) ,$$

where the gradients are computed by applying the rules of adjoint differentiation following the instructions that implement the function F. Finally, the adjoint, $\bar{X}_0$, is used to populate the component of $\bar{\theta}$ corresponding to the adjoint of the model parameter X_0.

3.3.3 AAD for XVA by regression

The AAD implementation of the algorithm for the calculation of XVA described in Section 3.2.5, producing the MC estimators for the sensitivities of the estimator shown in Equation (3.2.27) with respect to a set of model parameters θ_k, for $k = 1,\dots,N_\theta$, specified by

$$\bar{\theta}_k = \frac{\partial \mathrm{XVA}(\theta)}{\partial \theta_k} ,$$

comprises of the adjoint of steps (X1)-(X5) executed backwards for each MC replication $n = 1,\dots,N_{\mathrm{MC}}$:

($\overline{\mathrm{X5}}$) Set the adjoint of the XVA value, $\overline{\mathrm{XVA}} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$ and $\overline{\mathrm{XVA}}^{(n)} = \overline{\mathrm{XVA}} \frac{1}{N_{\mathrm{MC}}}$.

($\overline{\mathrm{X4}}$) For $m = M,\dots,1$ compute:

$$\begin{aligned}
\bar{V}_m^{(n)} = -\frac{\overline{\mathrm{XVA}}^{(n)}}{N_m^{(n)}} \Big[& L_c \left(\mathrm{SP}_{m-1}^{c,(n)} - \mathrm{SP}_m^{c,(n)} \right) \mathbb{I}(V_m^{(n)} > 0) \\
& + L_d \left(\mathrm{SP}_m^{d,(n)} - \mathrm{SP}_m^{d,(n)} \right) \mathbb{I}(V_m^{(n)} < 0) \Big], \\
\bar{N}_m^{(n)} = \frac{\overline{\mathrm{XVA}}^{(n)}}{\left(N_m^{(n)} \right)^2} \Big[& L_c \left(\mathrm{SP}_{m-1}^{c,(n)} - \mathrm{SP}_m^{c,(i)} \right) \left(V_m^{(n)} \right)^+ \\
& + L_d \left(\mathrm{SP}_{m-1}^{d,(n)} - \mathrm{SP}_m^{d,(n)} \right) \left(V_m^{(n)} \right)^- \Big],
\end{aligned}$$

and

$$\begin{aligned}
\overline{\mathrm{SP}}_m^{c,(n)} &= \overline{\mathrm{XVA}}^{(n)} \left[\frac{V_m^{(n)}}{N_m^{(n)}} (1 - \delta_{m,0}) - \frac{V_{m+1}^{(n)}}{N_{m+1}^{(n)}} \right] \mathbb{I}(V_m^{(n)} > 0) , \\
\overline{\mathrm{SP}}_m^{d,(n)} &= \overline{\mathrm{XVA}}^{(n)} \left[\frac{V_m^{(n)}}{N_m^{(n)}} (1 - \delta_{m,0}) - \frac{V_{m+1}^{(n)}}{N_{m+1}^{(n)}} \right] \mathbb{I}(V_m^{(n)} < 0) ,
\end{aligned}$$

where $\delta_{m,n}$ is the Kronecker delta symbol, and

$$V_{M+1}^{(n)} / N_{M+1}^{(n)} = 0.$$

($\overline{\text{X3}}$) For $m = M, \ldots, 1$, set

$$\bar{V}_{p,m}^{(n)} = \bar{V}_{m}^{(n)}$$

and compute the adjoint of Equation (3.2.26) by

$$\bar{H}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)} \mathbb{I}\left(H_{p,m}^{(n)} > E_{p,m}^{(n)}\right) , \quad \bar{E}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)} \mathbb{I}\left(H_{p,m}^{(n)} < E_{p,m}^{(n)}\right) ,$$

if T_m is an exercise date for the p-th option, and

$$\bar{H}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)} , \quad \bar{E}_{p,m}^{(n)} = 0 ,$$

otherwise. Initialise the adjoints of the risk factors,

$$\bar{X}_{m}^{(n)} = \bar{E}_{p,m}^{(n)} \frac{\partial E_{p,m}^{(n)}}{\partial X_{m}^{(n)}} + \bar{N}_{m}^{(n)} \frac{\partial N_{m}^{(n)}}{\partial X_{m}^{(n)}} .$$

Furthermore, initialise the adjoint of the regression coefficients $\beta_{p,m}$, the adjoints of the basis functions $\psi_{p,m}$, and update the adjoints of the state vector shown in Equations (3.3.7) to (3.3.9).

($\overline{\text{X2}}$) Update the adjoint of the simulated hazard rates by setting

$$\bar{\lambda}_{m}^{c,(n)} \mathrel{+}= -(T_{m+1} - T_m) \sum_{j=m+1}^{M} \overline{\text{SP}}_{m}^{c,(n)} \text{SP}_{m}^{c,(n)} ,$$

$$\bar{\lambda}_{m}^{d,(n)} \mathrel{+}= -(T_{m+1} - T_m) \sum_{j=m+1}^{M} \overline{\text{SP}}_{m}^{d,(n)} \text{SP}_{m}^{d,(n)} .$$

($\overline{\text{X1}}$) For $m = M, \ldots, 0$ compute the adjoint of the hazard rate evolution

$$\begin{aligned}
\bar{\lambda}_m^{c,(n)} &+= \bar{\lambda}_{m+1}^{c,(n)} \frac{\partial G_c}{\partial \lambda_m^{c,(n)}} \left(T_m, \lambda_m^{c,(n)}; \theta \right), \\
\bar{\lambda}_m^{d,(n)} &+= \bar{\lambda}_{m+1}^{d,(n)} \frac{\partial G_d}{\partial \lambda_m^{d,(n)}} \left(T_m, \lambda_m^{d,(n)}; \theta \right), \\
\bar{\theta} &+= \bar{\lambda}_{m+1}^{c,(n)} \frac{\partial G_c}{\partial \theta} \left(T_m, \lambda_m^{c,(n)}; \theta \right), \\
\bar{\theta} &+= \bar{\lambda}_{m+1}^{d,(n)} \frac{\partial G_d}{\partial \theta} \left(T_m, \lambda_m^{d,(n)}; \theta \right),
\end{aligned}$$

where the gradients can be computed through the AAD implementation of both the function G and the risk factor evolution as in step ($\overline{\text{L1}}$).

3.3.4 AAD for the regression algorithm

In the AAD implementations presented in Sections 3.3.2 and 3.3.3, the adjoint of the regression coefficients in Equation (3.3.7) do not contribute to the calculation of sensitivities. As mentioned in Section 3.3.1, and as will be illustrated with numerical examples in Section 3.4, this can be justified as a reasonable approximation in the case of Bermudan-style options when the exercise boundary approximated by regression is close to the optimal one. However, such arguments are generally approximations for Bermudan-style options, and cannot be invoked when regression is utilised for XVA. In this case, the contributions associated with the sensitivities of the regression coefficients must be taken into account in order to obtain accurate estimates of the model parameters' sensitivities.

In this section, we discuss how these contributions can be computed by the AAD implementation of the least-square MC algorithm described in Section 3.2.3.

($\overline{\text{R4}}$) Skip this step if the regression algorithm is utilised in conjunction with a second independent simulation for Bermudan-style options or in the context of XVA. Initialise the adjoint of the option value $\bar{V} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$, the adjoint of the regression coefficients $\bar{\beta}_m = 0$, for

$m = 1, \ldots, M-1$ and set

$$\bar{V}_1^{(n)} = \frac{\bar{V}}{N_{\mathrm{MC}}} \frac{1}{N_1^{(n)}} , \quad \bar{N}_1^{(n)} = -\frac{\bar{V}}{N_{\mathrm{MC}}} \frac{V_1^{(n)}}{\left(N_1^{(n)}\right)^2}$$

for $n = 1, \ldots, N_{\mathrm{MC}}$.

($\overline{\mathrm{R3}}$) For $m = 1$ to $M-1$, we have:

($\bar{\mathrm{d}}$) For $n = 1, \ldots, N_{\mathrm{MC}}$, compute

$$\bar{E}_m^{(n)} = \bar{V}_m^{(n)} \mathbb{I}\left(E_m^{(n)} > H_m^{(n)}\right), \quad \bar{H}_m^{(n)} = \bar{V}_m^{(n)} \mathbb{I}\left(E_m^{(n)} < H_m^{(n)}\right),$$

and initialise the adjoint of the risk factor path value $X_m^{(n)}$ by

$$\bar{X}_m^{(n)} = \bar{E}_m^{(n)} \frac{\partial E_m^{(n)}}{\partial X_m^{(n)}} .$$

($\bar{\mathrm{c}}$) Set

$$\bar{\beta}_m \mathrel{+}= \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} \bar{H}_m^{(n)},$$

and initialise the adjoint of the basis functions, ψ, as in Equation (3.3.8)[3].

($\bar{\mathrm{b}}$) Compute the adjoint of the two intermediate variables, Ω_m and Ψ_m, in Equation (3.2.14) using the results in [37] by

$$\bar{\Omega}_m = \Psi_m^{-T} \bar{\beta}_m , \quad \bar{\Psi}_m = -\bar{\Omega}_m \beta_m^T .$$

[3]Note that β_m will contain the sensitivities of the second simulation.

($\bar{\text{a}}$) For $n = 1, \ldots, N_{\text{MC}}$ compute the adjoint of Equation (3.2.13) by

$$\begin{aligned}
\bar{\psi}_m^{(n)} &+= \frac{1}{N_{\text{MC}}} \mathscr{N}_m^{(n)} V_{m+1}^{(n)} \bar{\Omega}_m \ , \\
\bar{V}_{m+1}^{(n)} &= \frac{1}{N_{\text{MC}}} \mathscr{N}_m^{(n)} (\psi_m^{(n)})^T \bar{\Omega}_m \ , \\
\bar{\mathscr{N}}_m^{(n)} &= \frac{1}{N_{\text{MC}}} V_{m+1}^{(n)} (\psi_m^{(n)})^T \bar{\Omega}_m \ ,
\end{aligned}$$

where $\mathscr{N}_m^{(n)} = N_m^{(n)} / N_{m+1}^{(n)}$, and compute the adjoint of Equation (3.2.12) by

$$\bar{\psi}_m^{(n)} += \frac{1}{2N_{\text{MC}}} \left(\bar{\Psi}_m^{(n)} + \left(\bar{\Psi}_m^{(n)} \right)^T \right) \psi_m^{(n)}.$$

Then, update the adjoint of the risk factor vectors by

$$\bar{X}_m^{(n)} += \frac{\bar{\mathscr{N}}_m^{(n)}}{N_{m+1}^{(n)}} \frac{\partial N_m^{(n)}}{\partial X_m^{(n)}} + \bar{\psi}^T \frac{\partial \bar{\psi}}{\partial X_m^{(n)}} \ , \qquad \bar{X}_{m+1}^{(n)} += - \frac{\bar{\mathscr{N}}_m^{(n)} \mathscr{N}_m^{(n)}}{N_{m+1}^{(n)}} \frac{\partial N_{m+1}^{(n)}}{\partial X_{T_{m+1}}^{(n)}} \ .$$

($\overline{\text{R2}}$) Compute the adjoint of the risk factor vector at expiry by

$$\bar{X}_M^{(n)} = \bar{V}_M^{(n)} \frac{\partial E_M^{(n)}}{\partial X_M^{(n)}},$$

for $n = 1, \ldots, N_{\text{MC}}$.

($\overline{\text{R1}}$) For $m = M, \ldots, 0$, compute the adjoint of the risk factor evolution as in step ($\overline{\text{L1}}$).

3.4 Numerical results

Here, we illustrate two numerical examples to demonstrate the efficiency and accuracy of the AAD algorithm for the regression-based MC method. We first compute the *Greeks* for the best of two assets Bermudan-style option via AAD and go on to make a comparison with the bump-and-revaluation method. Then as the second example, we consider the computation of its XVA sensitivities.

3.4.1 Best of two assets Bermudan-style option

As a first example, we consider the classical case of a Bermudan-style option on the maximum of two assets under a standard lognormal model for the asset price processes $\{X_1(t)\}$ and $\{X_2(t)\}$. The payoff function at exercise time t is specified by

$$(\max\{X_1(t), X_2(t)\} - K)^+, \tag{3.4.1}$$

where K is the strike price. A similar example is also studied in [13], [14] and page 462 of [39]. The almost exact estimates for this example using the regression-based Monte Carlo method equipped with twelve basis functions is given in [4].

We assume the option can be exercised every three months up to 3 years. As briefly mentioned before, we assume the underlying assets follow independent geometric Brownian motion processes with the same initial values $X(0)$, volatilities σ, and dividend rates d. In particular, we choose $X_1(0) = X_2(0) = 1$, $\sigma_1 = \sigma_2 = 0.2$ and $d_1 = d_2 = 0.1$ and let the risk-free rate $r = 0.05$. We compute the Bermudan-style option prices, the Deltas (i.e. the sensitivity with respect to the spot values, $X_i(0)$) and the Vegas (i.e. the sensitivity with respect to the volatilities σ_i) for the strike prices $K = 0.9$, 1.0 and 1.1 using both AAD and bumping. We also compare these with their "exact" value, which is obtained by a partial differential equation approach. The numerical results can be found in Table 3.1 and in Figure 3.1.

Throughout this section, the MC uncertainties are computed using the binning technique of Capriotti and Giles (2010). The binning method divides all the Monte Carlo paths into n bins with the same number of sample paths. Then, the AAD method is applied to each bin so that the sensitivity value for each bin can be obtained. Then one takes the average of the sensitivity values as the sensitivity estimator and the standard deviation over $\sqrt{n}$ as estimation error.

K = 0.9	Price	Delta	Vega
AAD	0.2012(2)	0.415(3)	0.456(2)
PDE	0.20107	0.41423	0.45740
K = 1.0			
AAD	0.1396(1)	0.337(2)	0.486(2)
PDE	0.13959	0.33588	0.48440
K = 1.1			
AAD	0.0941(2)	0.258(1)	0.463(2)
PDE	0.09431	0.25635	0.46253

Table 3.1: Prices, Deltas and Vegas for the Bermudan-style option in (3.4.1) with three different strike values. The smoothing parameter in the call spread regularisation (3.3.1) is $\delta = 0.005$. The number of MC paths is 400,000. The MC uncertainty (in parenthesis) is computed using the binning technique of [16] with 20 bins for each set of simulations.

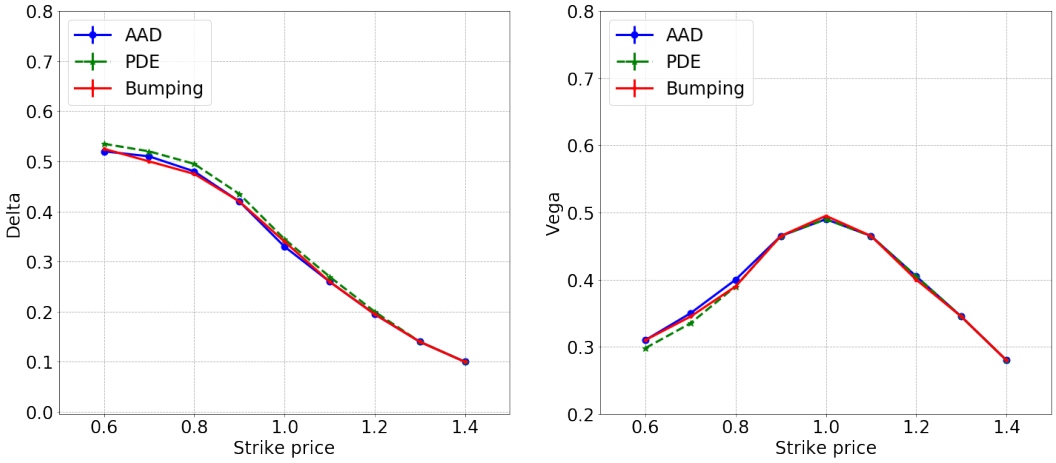

Figure 3.1: Deltas (left panel) and Vegas (right panel) for the Bermudan-style option in (3.4.1) as a function of strike. The smoothing parameter in the call spread regularisation (3.3.1) is $\delta = 0.005$. The number of MC paths is 400,000. The MC uncertainty (in parenthesis) is computed using the binning technique with 20 bins for each set of simulations.

In Table 3.1 and Figure 3.1, the smoothing parameter for the calculation of the *Greeks*, discussed in Section 3.3.1, $\delta = 0.005$, was chosen on the basis of it being a reasonable compromise between the variance and the bias of the estimator. This is illustrated in Figure 3.2, showing how for $\delta = 0.005$ the bias introduced by the finite δ is of the same order of magnitude as the statistical uncertainty for the chosen computational budget; this is negligible for any practical application.

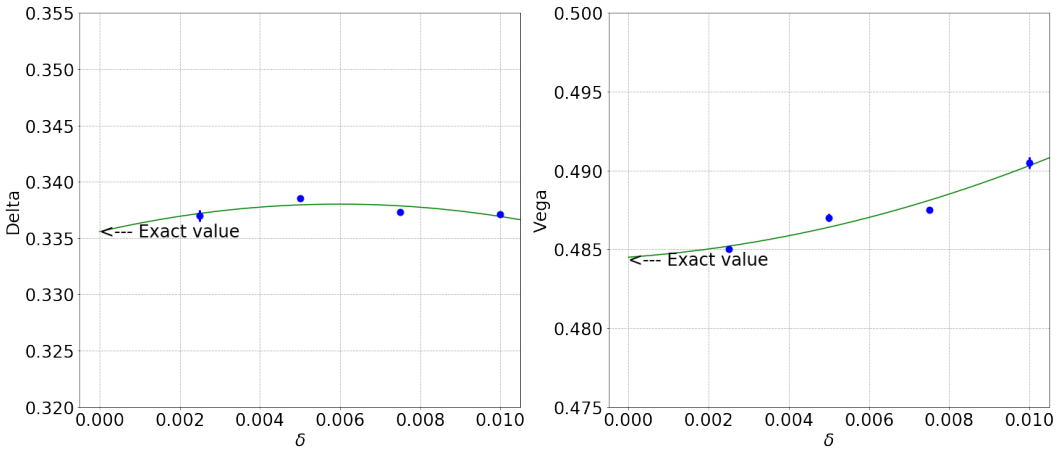

Figure 3.2: AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option with payoff specified in Equation (3.4.1) for $K = 1$ vs the smoothing parameter δ for the call spread regularisation (3.3.1). The number of simulated paths is 3,000,000 for $\delta = 0.01$ and is increased as δ is decreased in order to keep statistical uncertainties roughly constant. The values in the graphs are fitted based on a quadratic polynomial function (green lines).

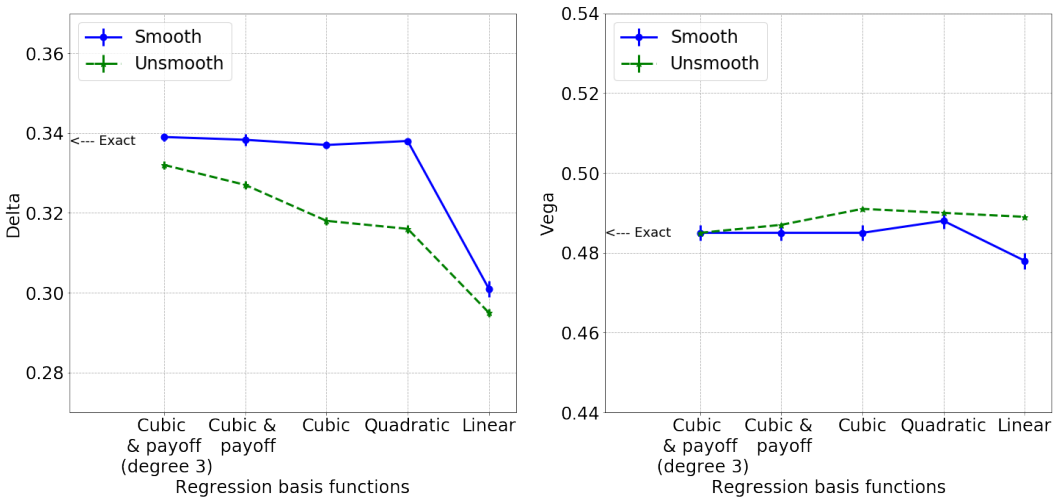

Figure 3.3: AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option with payoff specified in Equation (3.4.1) for $K = 1$ as obtained with the unsmoothed and the smoothened estimators with the call-spread approach (3.3.1) for five choices of the regression basis functions. The number of simulated paths is 1,000,000 with 20 bins and the smoothing component $\delta = 0.005$.

As discussed in Section 3.3.1, neglecting to smoothen out the exercise boundary, although common in the financial practice, introduces a bias in the computation of sensitivities because the exercise boundary is in general not optimal. This is illustrated in Figure 3.3, in which we compare the Delta, with and without smooth-

ing, for different choices of basis functions. Here, the smoothened estimator for the Delta turns out to be more accurate, especially when the exercise boundary obtained by regression is a less accurate approximation of the real one. However, it is difficult to establish *a priori* whether the unsmoothened estimator provides a smaller or a larger bias than the smoothened one. This is because the bias introduced by the lack of smoothing may be offset by the bias arising from the sub-optimality of the exercise boundary. This is illustrated in the right panel of Figure 3.3, showing that for Vega the smoothened and unsmoothened estimators have a comparable accuracy. In any case, a consideration to keep in mind is that smoothing the exercise boundary is generally required to obtain stable second-order risk values.

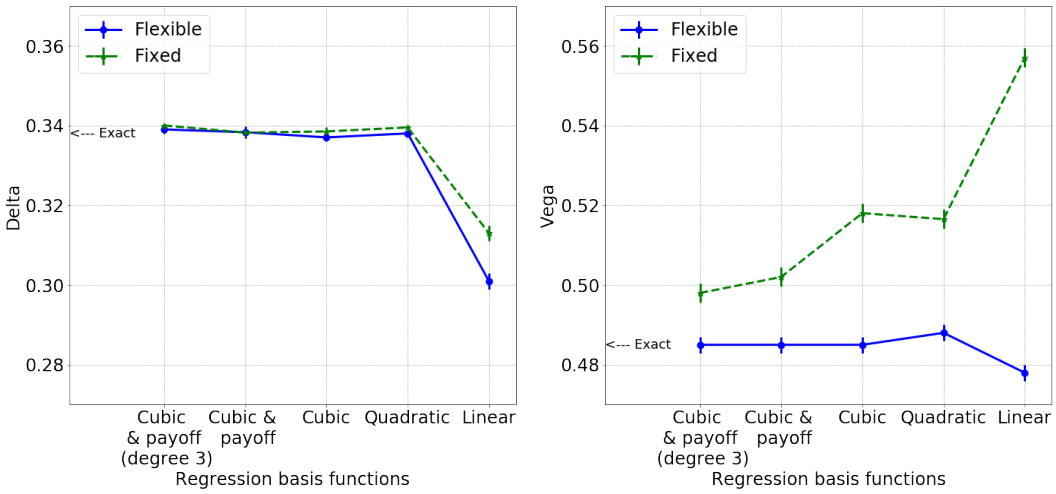

Figure 3.4: AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option with payoff specified in Equation (3.4.1) for $K = 1$ as obtained by keeping the exercise boundary fixed (Fixed) and accounting instead for its contributions to the sensitivities (Flexible). The number of simulated paths is 1,000,000 with 20 bins and the call-spread smoothing component $\delta = 0.005$.

The quasi-optimality of the exercise boundary is also generally invoked among industry practitioners as a justification for neglecting the contributions to the sensitivities arising from the exercise boundary. Clearly, the quality of this approximation is dependent on the accuracy of the regression functions in reproducing the actual exercise boundary. This is illustrated in Figure 3.4 where we plot Delta and Vega of the Bermudan-style option with payoff specified in Equation (3.4.1) for different choices of regression basis functions. Here, we compare the results obtained

by the AAD algorithm, as described in Section 3.3.4, in the case where a) the exercise boundary is kept fixed, and b) when accounting instead for its contributions to the sensitivities. Here, keeping regression boundaries fixed means that the steps to obtain regression coefficients will not contribute to sensitivity values. As expected, the difference between the two approaches vanishes as the regression functions become more and more accurate. As shown for the Vega, it can lead to a significant bias if a simple (for example, linear or quadratic) form of basis functions is adopted.

3.4.2 XVA sensitivities

As another example, we compute the sensitivities of the XVA shown in (3.2.7) for the same option with payoff specified in Equation (3.4.1). Here, for simplicity, we assume that the hazard rate and the volatility are piecewise constants. The XVA sensitivities with respect to some of the model parameters, namely the term structure of hazard rates of the counterparty and of the volatilities of the underlying, obtained with the AAD algorithm described in Section 3.2.5 are compared with the ones obtained by standard bumping. The comparative results are shown in Tables 3.2 and 3.3. As expected, the AAD sensitivities are in excellent agreement with those obtained by bumping, with discrepancies attributable to the bias of the finite-difference approach which is completely masked in this example by the MC uncertainties.

Similar to the case of Bermudan-style option *Greeks*, keeping the regression boundary fixed while computing the sensitivities may introduce a bias. However, for XVA, this issue is much more severe than in the case of the Bermudan-style option *Greeks* because no quasi-optimality argument can be invoked. As shown in Figure 3.5, the XVA sensitivities against the volatilities obtained by bumping and AAD, with flexible boundaries are almost identical. Instead, the results of AAD with fixed boundaries are notably different and, if utilised for risk management, this would lead to significant mis-hedging. On the other hand, and as is expected, the XVA sensitivities with respect to the hazard rates are not affected by the contribution arising from the regression functions. This is because the hazard rates do not enter the computation of the portfolio value conditional on default, and hence do not appear in the regression boundaries.

	σ_0	σ_1	σ_2	σ_3
AAD	0.00298(1)	0.00297(2)	0.00294(1)	0.00283(1)
Bumping	0.00298(1)	0.00297(2)	0.00294(1)	0.00283(1)
	σ_4	σ_5	σ_6	σ_7
AAD	0.00271(1)	0.00255(1)	0.00236(1)	0.00218(1)
Bumping	0.00271(1)	0.00255(1)	0.00236(1)	0.00218(1)
	σ_8	σ_9	σ_{10}	σ_{11}
AAD	0.00201(1)	0.00183(1)	0.00165(1)	0.001450(8)
Bumping	0.00201(1)	0.00183(1)	0.00165(1)	0.001450(8)

Table 3.2: XVA sensitivities with respect to the piecewise constant volatility. For both AAD and bumping, results are computed with 1,000,000 MC paths and 25 bins.

	λ_0	λ_1	λ_2	λ_3
AAD	0.03337(1)	0.03336(2)	0.03329(2)	0.03306(3)
Bumping	0.03337(1)	0.03336(2)	0.03329(1)	0.03306(3)
	λ_4	λ_5	λ_6	λ_7
AAD	0.03269(3)	0.03218(3)	0.03151(3)	0.03074(4)
Bumping	0.03269(3)	0.03218(3)	0.03151(3)	0.03074(3)
	λ_8	λ_9	λ_{10}	λ_{11}
AAD	0.02989(4)	0.02893(4)	0.02784(4)	0.02669(4)
Bumping	0.02989(4)	0.02893(4)	0.02784(4)	0.02669(4)

Table 3.3: XVA sensitivities with respect to the piecewise hazard rates. For both AAD and bumping, results are computed with 1,000,000 MC paths and 25 bins.

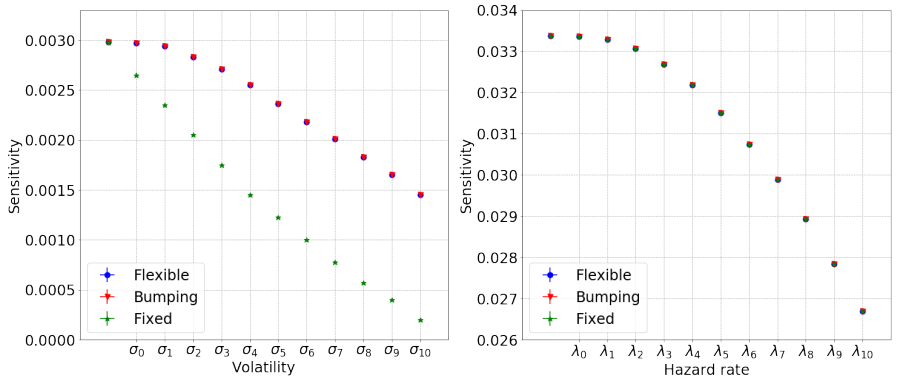

Figure 3.5: CVA sensitivities with respect to the piecewise constant volatility (left panel) and hazard rate (right panel), computed by AAD with flexible boundaries, fixed boundaries and bumping. The number of MC paths is 1,000,000 and the number of bins is 25.

Finally, the notable computational efficiency of the AAD approach is illustrated in Figure 3.6. Here we plot the cost of computing the XVA sensitivities,

with respect to the term structure of the counterparty hazard rate and the underlying volatility, relative to the cost of performing a single valuation. The calculation of the sensitivities by means of AAD can be performed by about three times the cost of computing the XVA value, which is well within the theoretical bound shown in (2.4.3). In contrast, the cost of bumping, for one-sided finite-difference estimators, is in general $(1+N)$ times the cost of a single valuation, where N is the number of model parameters, which is in this case over 20 times the cost of computing the value of the XVA.

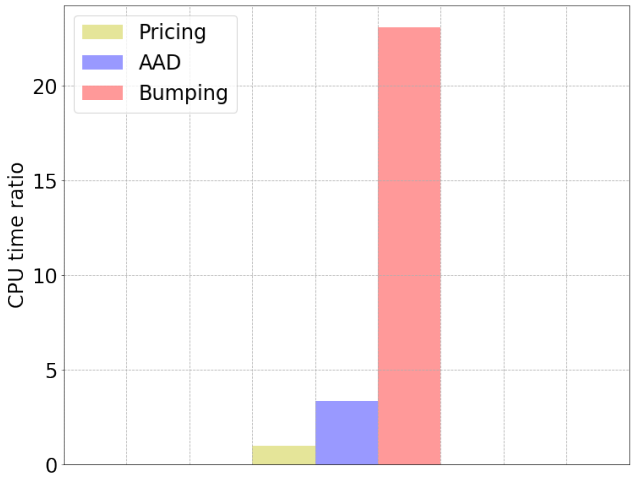

Figure 3.6: Ratio of the CPU time required for the calculation of the CVA and its sensitivities obtained by AAD and bumping. Here, CPU time required for the pricing function is set to be 1 unit.

3.5 Conclusions

We have shown how AAD can be used to efficiently implement the computation of sensitivities in regression-based MC methods. By discussing examples of practical relevance in detail, we have demonstrated how accounting for the sensitivities' contributions associated with the regression functions is crucial to obtain accurate estimates of the *Greeks* in XVA applications and also for Bermudan-style options especially when the exercise boundary is not particularly accurate. We also show how smoothing out the discontinuities associated with suboptimal exercise boundaries can lead, in some situations, to more accurate estimates of the sensitivities.

From a computational stand point, similarly to what was previously found in

other MC and PDE settings in [19], [21], the proposed method allows for the computation of the complete first-order risk at a cost which is at most four times the cost of calculating the value of the portfolio itself. This typically results in orders of magnitude of savings in computational time with respect to standard finite-difference approaches, thus making AAD an ever so indispensable technique in the toolbox of modern financial engineering and insurance mathematics.

Chapter 4

Multiple barrier-crossings of an Ornstein-Uhlenbeck diffusion in consecutive periods

4.1 Overview

The OU-process is a well-known diffusion process, widely used in physics, finance, biology and other fields. Due to its extensive use, the study of its FPT arises, naturally. The FPT density of a homogeneous OU-process for particular types of barrier functions can be found in closed-form. For example, if the barrier is equal to the OU-process long-term mean, its closed-form PDF has been found by [69], [41] and [76]. However, it is more involved to obtain the PDF of the FPT of the homogeneous OU-process to an arbitrary constant barrier. Although [54] claim that they managed to find the closed-form solution, [41] publish another paper in the same journal to point out the results in [54] were wrong due to the errors encountered when using the property of 3D Bessel bridges, see [66] and [67]. A Bessel bridge representation for the FPT-PDF of a homogeneous OU-process to an arbitrary constant barrier is provided in [41].

Since the closed-form expression of the moment generating function for the homogeneous OU-FPT has been well-studied, see for instance [69], [3] and [64], one may obtain an infinite series representation for the PDF of the FPT of a homo-

geneous OU-process crossing an arbitrary constant barrier by invoking the inverse Laplace transform, see [3]. The same infinite series representation is also obtained in [58] based on the spectral theory for options pricing, see [57]. However, to the best of our knowledge, the properties of the infinite series solution, especially the behaviour of the tail, have not yet been studied. The tail behaviour of the infinite series representation is practically important since it determines whether one can use the truncated series as a robust approximation.

The case of a time-inhomogeneous OU-process passing a time-dependent barrier tends to be more complicated. The FPT of a time-homogeneous Itô process where the barrier function must satisfy a first-order linear ordinary differential equation (ODE) is studied in [72]. Under such conditions, [72] provide a solution by numerical PDE methods. In [15], it shows that the FPT-PDF of a diffusion process passing a time-dependent boundary satisfies a Volterra integral equation of the second kind involving two arbitrary continuous functions. By using this method, the FPT-PDF for a homogeneous OU-process passing some special barrier specifications, for example where the barrier function is hyperbolic with respect to time, can be found analytically. This integral equation approach to time-inhomogeneous diffusion processes is generalised in [43]. The FPT-PDF of a time-inhomogeneous diffusion process passing a constant barrier can be obtained numerically by solving a PDE, see for example chapter 15 of [50] and [73]. The Fokker–Planck equation associated with an inhomogeneous OU-process passing a time-dependent barrier and introduce the method of images to derive the solution is studied in [59]. However, the generalisation to an unconstraint time-dependent barrier cannot be produced due to the strict conditions imposed by the method of images. [45] deals with the FPT of Itô processes whose local drift can be modelled in terms of a solution of the Burgers' equation. However, the OU-process does not belong to such a family of processes.

Since all continuous functions can be approximated to arbitrary precision, in the L_P-sense, by piecewise constant functions , it appears worthwhile to study the FPT of a homogeneous or inhomogeneous OU-process passing a piecewise con-

stant barrier function. In such cases, the probability that the FPT is within a certain threshold is equal to the probability that the maxima in all intervals, determined by the piecewise constant function, is less than the barrier level in the corresponding intervals. In this chapter, one of the main focuses is placed on the joint probability that the running maximum is above arbitrary fixed thresholds in pre-specified consecutive time intervals. Such a probabilistic problem arises, for example, in applications to environmental and climate risk, to which the insurance industry, but more importantly general global welfare, is exposed. Heat waves, or repeated and prolonged periods of droughts, can have substantial impact on economies, be these regional or (supra-)national. A heat wave is an event that often is defined by the temperature passing a pre-specified threshold on a number of consecutive days. This is an unequivocal case of where the joint probability of the running maximum of a stochastic process passing a fixed arbitrary barrier in consecutive intervals is necessary to address an important real-world challenge. However, to our knowledge, this mathematical problem has not yet been considered.

In this chapter, we study the multivariate survival function associated with an OU-process crossing arbitrary barriers in multiple time intervals. In Section 4.2, we adopt a PDE approach to deduce the infinite series representation of the survival function for the FPT of a homogeneous OU-process with lower reflection barrier passing a constant upper barrier. By considering the lower reflection barrier set at $-\infty$, we produce the same infinite series representation as in [58] and [3]. This can be viewed as a generalisation and an alternative derivation of the infinite series representation. The alternative derivation also gives a formula for the OU-process FPT density when a lower reflection barrier is introduced. This is novel. Moreover, we analyse the distributional properties of the deduced survival function, especially its tail behaviour and the truncation error. In Section 4.3, we provide a theorem that transforms the FPT of an inhomogeneous OU-process passing a time-dependent barrier to the FPT of a homogeneous OU-process with a different time-dependent barrier. This transfers the time-inhomogeneity from the process to the time-dependent barrier only, which simplifies the original problem. In Section

4.4, we deduce an integral representation of the joint distribution and joint survival function for the maxima of a continuous Markov process in consecutive intervals. With knowledge of the integral representation, the FPT density function and the numerical integration method, the joint distribution and joint survival function for the maxima of an OU-process with piecewise constant parameters in consecutive intervals can be efficiently obtained. We also show that under certain assumptions, the nested integration can be further simplified to become a product of single integrals, which leads to improved computational efficiency. Finally, we present the quadrature scheme and the MC integration method for the numerical integration in Section 4.5. Comparing with the direct MC approach, the results obtained by either the quadrature scheme or the MC integration method show improved accuracy and robustness. This is especially true in the *rare-event* cases, where the direct MC approach fails to reduce the approximation error efficiently.

4.2 Survival function for the FPT of a homogeneous OU-process passing a constant barrier

We begin by considering the FPT of a homogeneous OU-process crossing a constant barrier. The definitions for the homogeneous and standardised OU-processes are given in Definition 4.2.1. The FPT of a general continuous stochastic process is specified in Definition 4.2.2.

Definition 4.2.1 *An $\mathbb{R}$-valued stochastic process $(X_t)_{t\geq 0}$ is called a homogeneous OU-process if it satisfies the stochastic differential equation*

$$\mathrm{d}X_t = (\mu - \lambda X_t)\,\mathrm{d}t + \sigma \mathrm{d}W_t, \tag{4.2.1}$$

where $X_0 = x \in \mathbb{R}$, $\mu \in \mathbb{R}$, $\lambda > 0$, $\sigma > 0$ and $(W_t)_{t\geq 0}$ is a Brownian motion, which is defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$. When $\mu = 0$ and $\lambda = \sigma = 1$, we call the process $(\widetilde{X}_t)_{t\geq 0}$ that satisfies

$$\mathrm{d}\widetilde{X}_t = -\widetilde{X}_t \mathrm{d}t + \mathrm{d}W_t \tag{4.2.2}$$

a standardised OU-process.

Definition 4.2.2 *The FPT of a continuous process* $(X_t)_{t\geq 0}$ *to an upper constant barrier* $b > X_0 = x$ *is defined by* $\tau_{X,b} := \inf\{t \geq 0 : X_t \geq b\}$. *The survival function of* $\tau_{X,b}$, *denoted by* $\bar{F}_{\tau_{X,b}}(t;x)$, *is given by* $\bar{F}_{\tau_{X,b}}(t;x) = \mathbb{P}\left(\tau_{X,b} \geq t \mid X_0 = x\right)$.

As shown in [3], [64] and [58], if $(X_t)_{t\geq 0}$ is a homogeneous OU-process, the random variable $\tau_{X,b}$ is "properly" defined in the sense that $\mathbb{P}\left(\tau_{X,b} < \infty\right) = 1$.

Next we present a relation between the survival functions of the FPTs for two different homogeneous OU-processes. By invoking Lemma 4.2.1, if one knows the FPT distribution of a homogeneous OU-process to a given barrier, the FPT distribution of another homogeneous OU-process to a shifted barrier can also be obtained.

Lemma 4.2.1 *The random variable* $\tau_{X,b}$ *is equal to* $\tau_{\tilde{X},\tilde{b}}$ *in distribution for*

$$\tilde{t} = \lambda t, \quad \tilde{x} = \sqrt{\frac{\lambda}{\sigma^2}}\left(x - \frac{\mu}{\lambda}\right), \quad \tilde{b} = \sqrt{\frac{\lambda}{\sigma^2}}\left(b - \frac{\mu}{\lambda}\right)$$

That is, $\bar{F}_{\tau_{X,b}}(t,x) = \bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t},\tilde{x})$.

Proof. We have

$$\bar{F}_{\tau_{X,b}}(t,x) = \mathbb{P}\left(\sup_{s\in[0,t]} X_s < b \middle| X_0 = x\right).$$

Then by a change of time, it follows that

$$\mathbb{P}\left(\sup_{s\in[0,t]} X_s < b \middle| X_0 = x\right) = \mathbb{P}\left(\sup_{s\in[0,\lambda t]} X_{s/\lambda} < b \middle| X_0 = x\right).$$

Then,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{s\in[0,\lambda t]} X_{s/\lambda} < b \middle| X_0 = x\right) \\
&= \mathbb{P}\left(\sup_{s\in[0,\lambda t]} \sqrt{\frac{\lambda}{\sigma^2}} X_{s/\lambda} < \sqrt{\frac{\lambda}{\sigma^2}} b \middle| \sqrt{\frac{\lambda}{\sigma^2}} X_0 = \sqrt{\frac{\lambda}{\sigma^2}} x\right) \\
&= \mathbb{P}\left(\sup_{s\in[0,\lambda t]} \left(\sqrt{\frac{\lambda}{\sigma^2}} X_{s/\lambda} - \frac{\mu}{\sigma\sqrt{\lambda}}\right) < \tilde{b} \middle| \sqrt{\frac{\lambda}{\sigma^2}} X_0 - \frac{\mu}{\sigma\sqrt{\lambda}} = \tilde{x}\right)
\end{aligned}$$

The dynamics of the process $(\sqrt{\lambda/\sigma^2} X_{s/\lambda} - \mu/(\sigma\sqrt{\lambda}))_{s\geq 0}$ are given by

$$\mathrm{d}\left(\sqrt{\frac{\lambda}{\sigma^2}} X_{s/\lambda} - \frac{\mu}{\sigma\sqrt{\lambda}}\right) = \sqrt{\frac{\lambda}{\sigma^2}} \mathrm{d}X_{t/\lambda} = -\left(\sqrt{\frac{\lambda}{\sigma^2}} X_{t/\lambda} - \frac{\mu}{\sigma\sqrt{\lambda}}\right) \mathrm{d}t + \mathrm{d}W_t.$$

This means that, in law, the process $(\sqrt{\lambda/\sigma^2} X_{s/\lambda} - \mu/(\sigma\sqrt{\lambda}))_{s\geq 0}$ is a standardised OU-process. Therefore,

$$\mathbb{P}\left(\sup_{s\in[0,t]} X_s < b \middle| X_0 = x\right) = \mathbb{P}\left(\sup_{s\in[0,\tilde{t}]} \widetilde{X}_s < \tilde{b} \middle| \widetilde{X}_0 = \tilde{x}\right),$$

that is, $\bar{F}_{\tau_{X,b}}(t,x) = \bar{F}_{\tau_{\widetilde{X},\tilde{b}}}(\tilde{t},\tilde{x})$. □

Lemma 4.2.1 provides a relationship between the survival functions – that is between the distributions of the FPTs – of the homogeneous, respectively, the standardised OU-process. In order to calculate the FPT survival function for a homogeneous OU-process, one can first calculate the FPT survival function for a standardised OU-process. Therefore, and from now on in this section, we consider the case of a standardised OU-process.

4.2.1 The FPT survival function of the standardised OU-process with a constant barrier

Similar to the valuation of a binary option with a constant up-and-out barrier for a fixed expiration, see [78] for example, the FPT survival function of the standard

OU-process with a constant upper barrier can be characterised by the following PDE problem. On $C^{1,2}\left([0,\infty),(-\infty,\tilde{b}]\right)$, the function $\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t},\tilde{x})$ satisfies the forward PDE

$$\frac{\partial \bar{F}_{\tau_{\tilde{X},\tilde{b}}}}{\partial \tilde{t}} = \mathscr{A}\bar{F}_{\tau_{\tilde{X},\tilde{b}}}, \tag{4.2.3}$$

subject to the initial and boundary conditions

$$\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(0,\tilde{x}) = 1, \tag{4.2.4}$$

$$\bar{F}_{\tau_{\tilde{X},\tilde{b}}}\left(\tilde{t},\tilde{b}\right) = 0. \tag{4.2.5}$$

Here $\mathscr{A}$ is the infinitesimal operator of a standardised OU-process $(\widetilde{X}_{\tilde{t}})_{\tilde{t}\geq 0}$ given by

$$\mathscr{A} = -\tilde{x}\frac{\partial}{\partial \tilde{x}} + \frac{1}{2}\frac{\partial^2}{\partial \tilde{x}^2}.$$

In order to solve this PDE, we add in the lower boundary condition,

$$\frac{\partial \bar{F}_{\tau_{\tilde{X},\tilde{b}}}}{\partial \tilde{x}}(\tilde{t},\tilde{a}) = 0, \tag{4.2.6}$$

which is the condition for a reflecting lower boundary at location $\tilde{a} < \tilde{b}$. Here, we follow the definition from Section 3 of [57] for the reflection boundary: the boundary point $\tilde{a}$ is included in the state space and $(\mathscr{A}f)(\tilde{a}) = \lim_{x\downarrow\tilde{a}}(\mathscr{A}f)(x)$ for a function $f \in \mathbb{C}^2([\tilde{a},\tilde{b}])$. The boundary condition (4.2.6) can be found in [73], for example.

Through this chapter, we use several special functions and their properties. We list them in Appendix B. The following proposition provides a generalisation to the infinite series representation for the FPT of a homogeneous OU-process in [3] and [58]. It recovers their results when $\tilde{a} \to -\infty$, which will be shown in Theorem 4.2.1.

Proposition 4.2.1 *The analytic solution to the PDE (4.2.3), subject to the initial condition and boundary conditions specified in Equations (4.2.4), (4.2.5) and*

(4.2.6) is given by

$$\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t},\tilde{x}) = \sum_{k=1}^{\infty} c_k \mathrm{e}^{-\alpha_k \cdot \tilde{t}} H(\alpha_k,\tilde{x};\tilde{a}),$$

for $k \in \mathbb{N}$, where

$$H(\alpha_k,\tilde{x};\tilde{a}) = \frac{2^{\alpha}\sqrt{\pi}}{\Gamma\left(\frac{1-\alpha}{2}\right)}\left({}_1F_1\left(-\frac{\alpha_k}{2};\frac{1}{2};\tilde{x}^2\right) + y(\alpha_k,\tilde{a})\tilde{x}\ {}_1F_1\left(\frac{1-\alpha_k}{2};\frac{3}{2};\tilde{x}^2\right)\right),$$

$$y(\alpha_k,\tilde{a}) = \frac{2\alpha_k\tilde{a}\ {}_1F_1\left(\frac{2-\alpha_k}{2};\frac{3}{2};\tilde{a}^2\right)}{{}_1F_1\left(\frac{1-\alpha_k}{2};\frac{3}{2};\tilde{a}^2\right) + \frac{2}{3}(1-\alpha_k)\tilde{a}^2\ {}_1F_1\left(\frac{3-\alpha_k}{2};\frac{5}{2};\tilde{a}^2\right)}.$$

Here, ${}_1F_1$ is the confluent hypergeometric function of the first kind, see Appendix B, and α_k are the ordered solutions to the equation,

$${}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{b}^2\right) + y(\alpha,\tilde{a})\tilde{b}\ {}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{b}^2\right) = 0, \tag{4.2.7}$$

with respect to α. Furthermore, the coefficient c_k is given by

$$c_k = -1/[\alpha_k\, \partial_{\alpha_k} H(\alpha_k,\tilde{x};\tilde{a})].$$

Proof. By the method of eigenfunction expansion, $\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t},\tilde{x})$ admits the following representation

$$\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t},\tilde{x}) = \sum_{k=1}^{\infty} c_k \mathrm{e}^{-\alpha_k \tilde{t}} \phi_k(\tilde{x}),$$

where c_k are the constant coefficients, and α_k as well as $\phi_k(\tilde{x})$ are the eigenvalues and eigenfunctions that satisfy the general eigenfunction equation

$$\mathscr{A}\phi_k(\tilde{x}) = -\alpha_k\phi_k(\tilde{x}), \tag{4.2.8}$$

subject to

$$\phi_k'(\tilde{a}) = \phi_k(\tilde{b}) = 0.$$

The pair $(\phi(\cdot),\alpha)$ satisfies

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}\tilde{x}^2}-2\tilde{x}\frac{\mathrm{d}\phi}{\mathrm{d}\tilde{x}}+2\alpha\phi=0 \tag{4.2.9}$$

subject to

$$\phi'(\tilde{a})=\phi(\tilde{b})=0. \tag{4.2.10}$$

As shown in page 761 of [77], the ODE (4.2.9) is known as the Hermite differential equation, whose general solution is given by

$$\phi(\tilde{x})=A\ {}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{x}^2\right)+B\tilde{x}\ {}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{x}^2\right) \tag{4.2.11}$$

where A and B are independent of $\tilde{x}$. After substituting Equation (4.2.11) into condition (4.2.10), we obtain the system

$$\begin{cases} A\ {}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{b}^2\right)+B\tilde{b}\ {}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{b}^2\right)=0, \\ \\ B\left[\ {}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)+\frac{2}{3}(1-\alpha)\tilde{a}^2\ {}_1F_1\left(\frac{3-\alpha}{2};\frac{5}{2};\tilde{a}^2\right)\right]=2A\alpha\tilde{a}\ {}_1F_1\left(\frac{2-\alpha}{2};\frac{3}{2};\tilde{a}^2\right). \end{cases}$$

Therefore, the eigenvalues α_k must be the zeros of the equation

$${}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{b}^2\right)+y(\alpha,\tilde{a})\tilde{b}\ {}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{b}^2\right)=0$$

with respect to α. So we let

$$\begin{aligned}\phi_k(\tilde{x}) =& H(\alpha_k,\tilde{x};\tilde{a}) \\ =& \frac{2^\alpha\sqrt{\pi}}{\Gamma\left(\frac{1-\alpha}{2}\right)}\left(\ {}_1F_1\left(-\frac{\alpha_k}{2};\frac{1}{2};\tilde{x}^2\right)+y(\alpha_k,\tilde{a})\tilde{x}\ {}_1F_1\left(\frac{1-\alpha_k}{2};\frac{3}{2};\tilde{x}^2\right)\right),\end{aligned}$$

which will be convenient for later use. By Proposition 2 in [57], the coefficient of

each term can be calculated and are given by

$$c_k = -1/[\alpha_k \, \partial_{\alpha_k} H(\alpha_k, \tilde{x}; \tilde{a})].$$

□

Proposition 4.2.1 gives the survival function of the FPT for a homogeneous OU-process passing a given upper barrier subject to a lower reflection boundary. We now re-derive the formulae after removing the lower reflection boundary by taking a limit in the following theorem. This is a different derivation of the infinite series representation in [3] and [58] based on relaxing specific conditions. Here, the definition of Hermite function $\mathscr{H}_\alpha(x)$ is given in page 509 of [1].

Theorem 4.2.1 *The analytic solution to the PDE (4.2.3) subject to the initial and boundary conditions specified in Equations (4.2.4) and (4.2.5) is given by*

$$\bar{F}_{\tau_{\tilde{X},\tilde{b}}}(\tilde{t}, \tilde{x}) = \sum_{k=1}^{\infty} c_k \mathrm{e}^{-\alpha_k \tilde{t}} \mathscr{H}_{\alpha_k}(-\tilde{x})$$

for $k \in \mathbb{N}$, where $\mathscr{H}_\alpha(\cdot)$ is the Hermite function with parameter α, α_k are the solutions to the equation $\mathscr{H}_\alpha\left(-\tilde{b}\right) = 0$ with respect to α, and $c_k = -1/[\alpha_k \cdot \partial_{\alpha_k} \mathscr{H}_{\alpha_k}(-\tilde{b})]$.

Proof. By page 506 of [1], we have the asymptotic relationship ${}_1F_1(x; y; z) \sim \Gamma(y)\mathrm{e}^z z^{x-y}/\Gamma(x)$ for $z \to \infty$, that is,

$$\lim_{z\to\infty} \frac{{}_1F_1(x; y; z)\,\Gamma(x)}{\Gamma(y)\mathrm{e}^z z^{x-y}} = 1.$$

Therefore,

$$\begin{aligned}\lim_{\tilde{a}\to-\infty} y(\alpha,\tilde{a}) &= \lim_{\tilde{a}\to-\infty} \frac{2\alpha\tilde{a}\ {}_1F_1\left(\frac{2-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)}{{}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)+\frac{2}{3}(1-\alpha)\tilde{a}^2\ {}_1F_1\left(\frac{3-\alpha}{2};\frac{5}{2};\tilde{a}^2\right)} \\ &= \frac{2\alpha}{\lim\limits_{\tilde{a}\to-\infty}\frac{{}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)}{{}_1F_1\left(\frac{2-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)\tilde{a}}+\frac{2}{3}(1-\alpha)\lim\limits_{\tilde{a}\to-\infty}\frac{{}_1F_1\left(\frac{3-\alpha}{2};\frac{5}{2};\tilde{a}^2\right)\tilde{a}}{{}_1F_1\left(\frac{2-\alpha}{2};\frac{3}{2};\tilde{a}^2\right)}} \\ &= \frac{2\Gamma(\frac{1-\alpha}{2})}{\Gamma(-\frac{\alpha}{2})},\end{aligned}$$

in particular, when $\alpha=\alpha_k$. For $\tilde{a}\to-\infty$ and $\alpha=\alpha_k$, the eigenvalues α_k are required to satisfy

$$2^{\alpha}\sqrt{\pi}\left(\frac{{}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{b}^2\right)}{\Gamma\left(\frac{1-\alpha}{2}\right)}+2\tilde{b}\frac{{}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{b}^2\right)}{\Gamma\left(-\frac{\alpha}{2}\right)}\right)=0,$$

which turn out to be the zeros of the Hemite function $\mathscr{H}_{\alpha}(-\tilde{b})$ with respect to α, i.e.

$$\mathscr{H}_{\alpha}(-\tilde{b}):=2^{\alpha}\sqrt{\pi}\left[\frac{{}_1F_1\left(-\frac{\alpha}{2};\frac{1}{2};\tilde{b}^2\right)}{\Gamma\left(\frac{1-\alpha}{2}\right)}+2\tilde{b}\frac{{}_1F_1\left(\frac{1-\alpha}{2};\frac{3}{2};\tilde{b}^2\right)}{\Gamma\left(-\frac{\alpha}{2}\right)}\right]=0.$$

Thus, the eigenfunctions are represented by $\phi_k(\tilde{x})=\mathscr{H}_{\alpha_k}(-\tilde{x})$. The coefficients can then be obtained by Proposition 4.2.1. □

Remark 4.2.1 *The Hermite function $\mathscr{H}_{\alpha}(x)$ is equal to the limit,*

$$\lim_{\tilde{a}\to-\infty} H(\alpha,x;\tilde{a}),$$

where $H(\alpha,x;\tilde{a})$ is specified as in Proposition 4.2.1.

Example 4.2.1 Here we consider the PDF for the FPT of a standardised OU-process hitting the upper barrier $\tilde{b}$ with different lower reflection barriers $\tilde{a}$. Figure 4.1 shows that the distances of the ordered eigenvalues tend to increase, regardless of the value which $\tilde{a}$ takes. We can observe from Figure 4.2 that when $\tilde{a}$ becomes smaller, the PDF with lower reflection barrier approaches the PDF without lower reflection barrier.

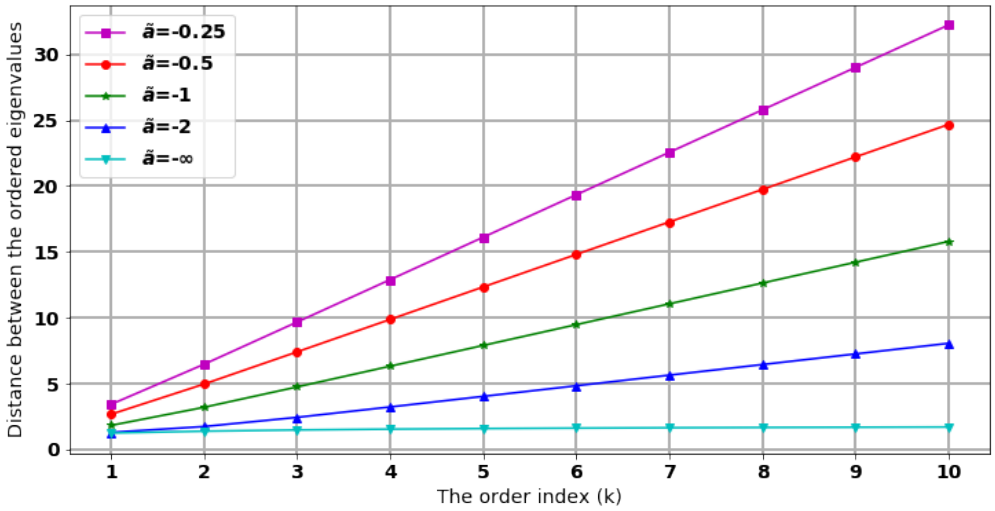

Figure 4.1: Distance between the first eleven ordered eigenvalues for the PDE (4.2.3), with upper barrier $\tilde{b} = 1.5$ and different lower reflection barriers $\tilde{a}$.

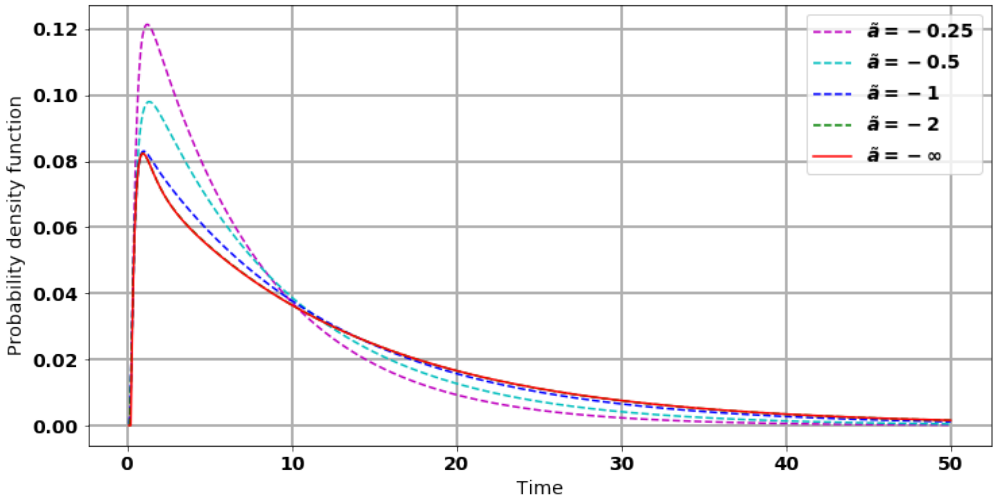

Figure 4.2: The probability density function for the first-passage-time of a standardised OU-process crossing the upper barrier $\tilde{b} = 1.5$, with lower reflection barrier $\tilde{a}$.

Corollary 4.2.1 *The analytical form of the FPT survival function of the OU-process shown in Equation (4.2.1) is given by*

$$\bar{F}_{\tau_{x,b}}(t,x) = \sum_{k=1}^{\infty} c_k \mathrm{e}^{-\lambda \alpha_k t} \mathscr{H}_{\alpha_k}\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(x - \frac{\mu}{\lambda}\right)\right), \tag{4.2.12}$$

where $\mathscr{H}_{\alpha_k}(\cdot)$ is the Hermite function, and the α_k's are the ordered solutions to the

equation

$$\mathscr{H}_{\alpha_k}\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(b-\frac{\mu}{\lambda}\right)\right)=0.$$

Furthermore, the coefficient c_k is given by

$$c_k=-\frac{1}{\alpha_k\cdot\partial_{\alpha_k}\mathscr{H}_{\alpha_k}\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(b-\frac{\mu}{\lambda}\right)\right)}.$$

Proof. By Lemma 4.2.1, we have

$$\tilde{t}=\lambda t,\quad \tilde{x}=\sqrt{\frac{\lambda}{\sigma^2}}\left(x-\frac{\mu}{\lambda}\right),\quad \tilde{b}=\sqrt{\frac{\lambda}{\sigma^2}}\left(b-\frac{\mu}{\lambda}\right).$$

Substituting these parameters into Theorem 4.2.1, we obtain

$$\bar{F}_{\tau_{X,b}}(t,x)=\sum_{k=1}^{\infty}c_k\mathrm{e}^{-\lambda\alpha_k t}\mathscr{H}_{\alpha_k}\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(x-\frac{\mu}{\lambda}\right)\right),\tag{4.2.13}$$

where α_k and c_k are given as above. □

In the following theorem, we demonstrate the absolute convergence of the infinite series (4.2.12) and the bound of the truncation error by using Corollary 4.2.1.

Theorem 4.2.2 *The infinite series in Equation (4.2.12) is absolutely convergent. The truncated series $\sum_{k=1}^{K}c_k\mathrm{e}^{-\lambda\alpha_k t}\mathscr{H}_{\alpha_k}\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(x-\frac{\mu}{\lambda}\right)\right)$ has truncation error of*

$$O\left(\mathrm{e}^{-2K\lambda t}\right)$$

as $K\to\infty$. Moreover, the absolute value of the truncation error is bounded by

$$\varepsilon(\alpha_K)=\frac{\exp\left(\frac{x'^2-b'^2}{2}\right)}{\sqrt{2}|b'|}\left[\frac{\exp\left(-\lambda t\alpha_K\right)}{\alpha_K}+(1-\lambda t)\Gamma(0,\lambda t\alpha_K)\right],$$

where $x'=\sqrt{\frac{\lambda}{\sigma^2}}\left(x-\frac{\mu}{\lambda}\right)$, $b'=\sqrt{\frac{\lambda}{\sigma^2}}\left(b-\frac{\mu}{\lambda}\right)$, and $\Gamma(a,x)$ is the upper incomplete Gamma function with parameter a.

Proof. By adopting the results in Section 4.14 on page 66 of [53], as $\alpha_k \to \infty$, which means $k \to \infty$, we have that

$$\begin{aligned}\mathscr{H}_{\alpha_k}\left(-x'\right) =& 2^{\alpha_k+1/2}\mathrm{e}^{x'^2-\alpha_k/2-1/4}\left(\frac{\alpha_k}{2}+\frac{1}{4}\right)^{\alpha_k/2} \\ &\times\cos\left(2x'\sqrt{\frac{\alpha_k}{2}+\frac{1}{4}}-\frac{\alpha_k\pi}{2}\right)\left[1+O\left(\frac{1}{\sqrt{\alpha_k/2+1/4}}\right)\right].\end{aligned} \tag{4.2.14}$$

Hence, for a large enough $k \in \mathbb{N}$, we have

$$\begin{aligned}c_k =& -\left[1+O\left(\frac{1}{\sqrt{\alpha_k/2+1/4}}\right)\right]\cdot\left[\alpha_k 2^{\alpha_k+\frac{1}{2}}\exp\left(\frac{b'^2}{2}-\frac{\alpha_k}{2}-\frac{1}{4}\right)\left(\frac{\alpha_k}{2}+\frac{1}{4}\right)^{\alpha_k/2}\right. \\ &\left.\times\sin\left(2b'\sqrt{\frac{\alpha_k}{2}+\frac{1}{4}}-\frac{\alpha_k\pi}{2}\right)\left(\frac{\pi}{2}+\frac{b'\alpha_k}{\sqrt{\alpha_k/2+1/4}}\right)\right]^{-1}.\end{aligned}$$

We can also obtain the asymptotic behaviour of α_k,

$$\alpha_k = 2k+1+\frac{4b'^2}{\pi^2}+\frac{2b'}{\pi}\sqrt{4k+3+\frac{4b'^2}{\pi^2}}, \tag{4.2.15}$$

when $k \to \infty$. Therefore, for large enough $K \in \mathbb{N}$, the exact truncation error of Equation (4.2.12) is

$$\begin{aligned}&\sum_{k=K}^{\infty} c_k \mathscr{H}_{\alpha_k}\left(-x'\right)\mathrm{e}^{-\lambda\alpha_k t} \\ &=\sum_{k=K}^{\infty} -\exp\left(\frac{x'^2-b'^2}{2}\right)\frac{\cos\left(2x'\sqrt{\alpha_k/2+1/4}-\frac{\alpha_k\pi}{2}\right)\mathrm{e}^{-\lambda\alpha_k t}}{\alpha_k\sin\left(2b'\sqrt{\alpha_k/2+1/4}-\frac{\alpha_k\pi}{2}\right)\left(\frac{\pi}{2}+\frac{b'\alpha_k}{\sqrt{\alpha_k/2+1/4}}\right)}.\end{aligned}$$

Since α_k are the zeros such that $\mathscr{H}_{\alpha_k}\left(-b'\right)=0$, by (4.2.14) we obtain

$$\cos\left(2b'\sqrt{\frac{\alpha_k}{2}+\frac{1}{4}}-\frac{\alpha_k\pi}{2}\right)=0.$$

Therefore,

$$\left|\sin\left(2b'\sqrt{\frac{\alpha_k}{2}+\frac{1}{4}}-\frac{\alpha_k\pi}{2}\right)\right|=1.$$

Therefore, we have the asymptotic inequality

$$\begin{aligned}
\sum_{k=K}^{\infty}\left|c_k\mathscr{H}_{\alpha_k}\left(-x'\right)\mathrm{e}^{-\lambda\alpha_k t}\right| &\leq \sum_{k=K}^{\infty}\frac{\exp\left(\frac{x'^2-b'^2}{2}-\lambda\alpha_k t\right)}{\alpha_k\left|\frac{\pi}{2}+\frac{b'\alpha_k}{\sqrt{\alpha_k/2+1/4}}\right|}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)C_1\sum_{k=K}^{\infty}\frac{\exp\left(-\lambda\alpha_k t\right)}{\alpha_k^{3/2}}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)C_1\exp\left(-\lambda\alpha_K t\right)\sum_{k=K}^{\infty}\frac{1}{\alpha_k^{3/2}}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)C_1C_2\exp\left(-\lambda\alpha_K t\right)\sum_{k=K}^{\infty}\frac{1}{k^{3/2}}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)C_1C_2C_3\exp\left(-\lambda\alpha_K t\right)\\
&= O\left(\exp\left(-\lambda\alpha_K t\right)\right)\\
&= O\left(\mathrm{e}^{-2K\lambda t}\right).
\end{aligned}$$

Meanwhile, since $\alpha_K>0$, we have that

$$\begin{aligned}
\sum_{k=K}^{\infty}\left|c_k\mathscr{H}_{\alpha_k}\left(-x'\right)\mathrm{e}^{-\lambda\alpha_k t}\right| &\leq \sum_{k=K}^{\infty}\frac{\exp\left(\frac{x'^2-b'^2}{2}-\lambda\alpha_k t\right)}{\alpha_k\left|\frac{\pi}{2}+\frac{b'\alpha_k}{\sqrt{\alpha_k/2+1/4}}\right|}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)\sum_{k=K}^{\infty}\frac{\exp\left(-\lambda\alpha_k t\right)\sqrt{\alpha_k+1}}{\sqrt{2}|b'|\alpha_k^2}\\
&\leq \exp\left(\frac{x'^2-b'^2}{2}\right)\int_{\alpha_K}^{\infty}\frac{\exp\left(-\lambda\alpha t\right)(\alpha+1)}{\sqrt{2}|b'|\alpha^2}\mathrm{d}\alpha\\
&= \frac{\exp\left(\frac{x'^2-b'^2}{2}\right)}{\sqrt{2}|b'|}\left[\frac{\exp\left(-\lambda t\alpha_K\right)}{\alpha_K}+(1-\lambda t)\Gamma(0,\lambda t\alpha_K)\right].
\end{aligned}$$

□

Theorem 4.2.2 gives an upper bound for the truncation error. One can determine the terms to be kept in order to achieve such a specific accuracy level with an explicit function. An example of the error and its upper bound can be found in Figure 4.3.

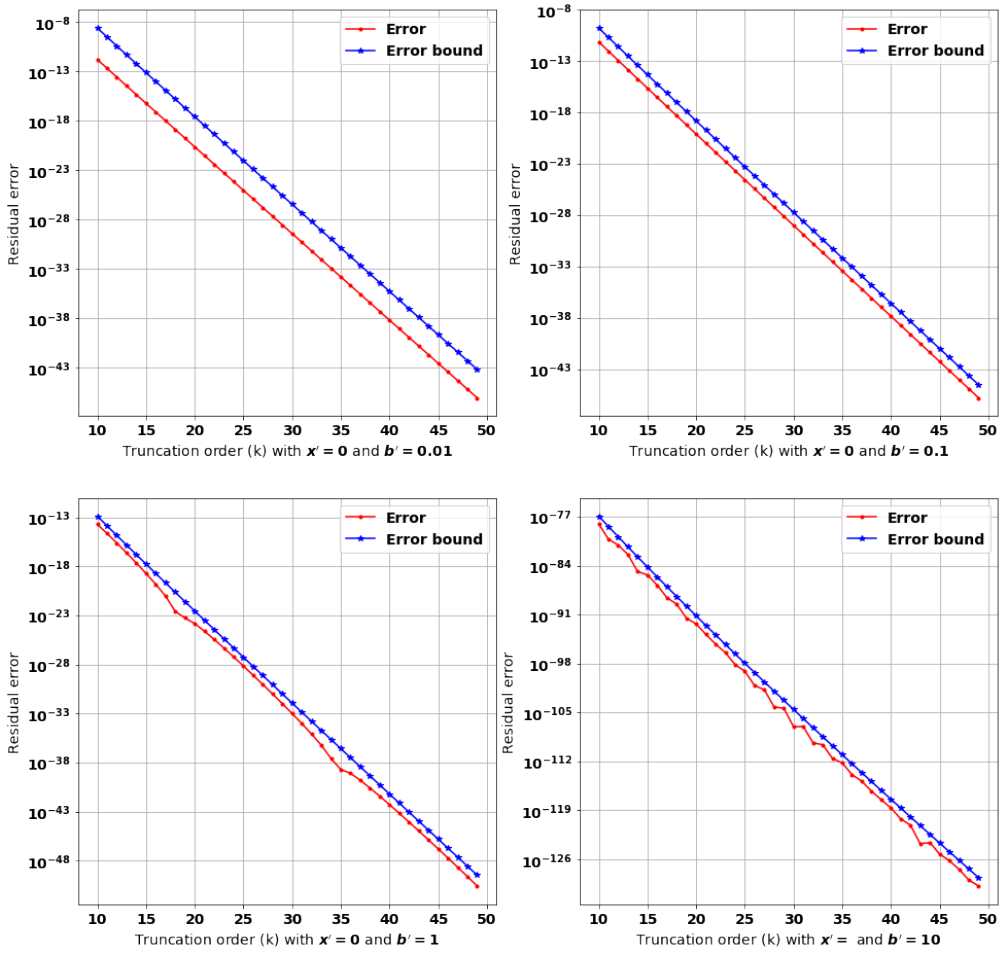

Figure 4.3: The log-scale plot between the truncation terms and the error for OU-process with $\mu = 0$, $\sigma = 1$ and $\lambda = 0.1$ for initial values 0 and different upper barriers 0.01, 0.1, 1 and 10.

To use the infinite series approximation for a probability density function, one needs to analyse how many terms we should truncate in order to reach a certain precision level for the approximated distribution. Since one can transform the homogeneous OU-process barrier-crossing problem to the standardised OU-process barrier-crossing problem, see Lemma 4.2.1, we now study the truncation precision of the standardised OU-process. Here, we plot the number of truncations required

for various initial values and barrier levels in Figure 4.4, where the α-zeros are taken in $[0, 70]$. We observe that when the barrier level is far away from the initial value, the number of truncations required becomes smaller. Figure 4.4 can be treated as a benchmark to determine how many terms one should truncate for a given quantile precision requirement.

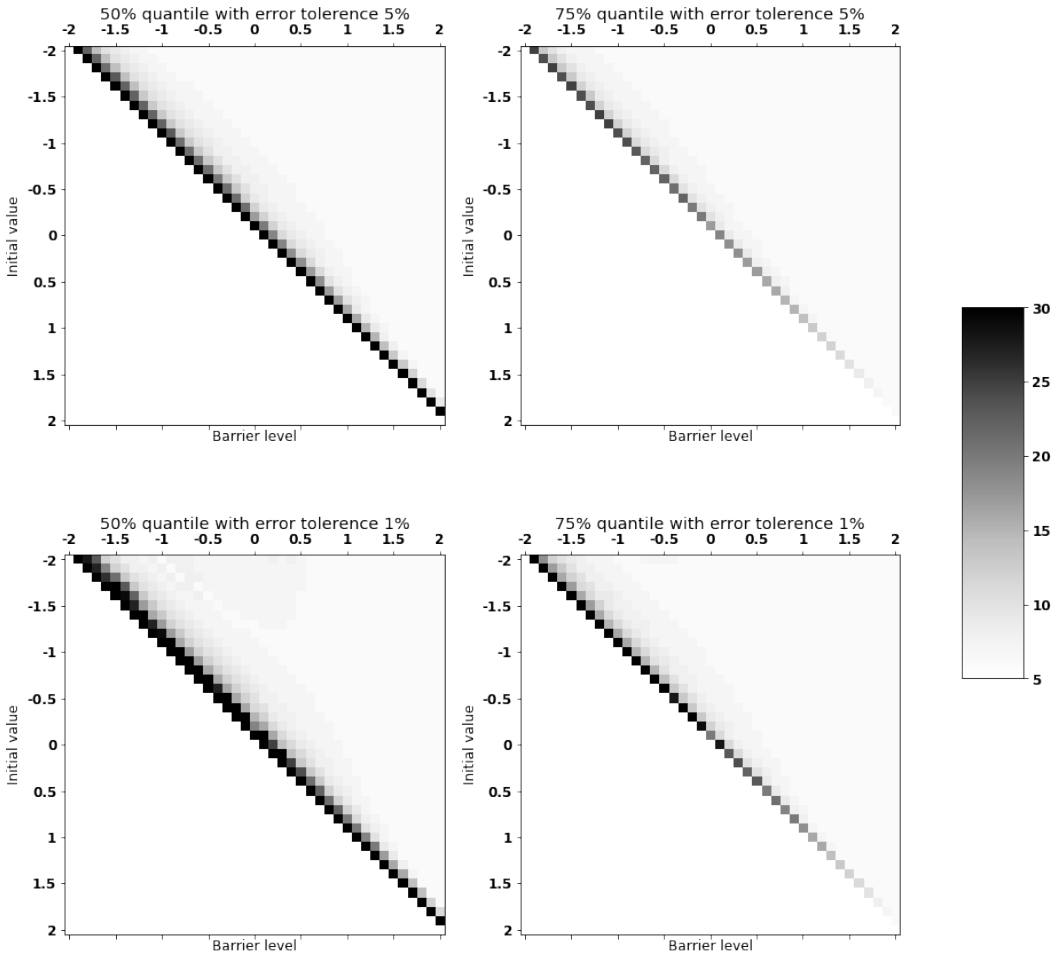

Figure 4.4: Relationship between the process initial values, barrier levels and the number of truncations.

Remark 4.2.2 *The α-zeros can be obtained by the bisection method. However, these α-zeros do not need to be obtained with high precision. Through the numerical test, we notice that if the α-zeros are accurate up to 10^{-4}, the approximation can be stable and reliable.*

Remark 4.2.3 *For a barrier level b which is larger than 5, numerically solving the*

higher orders of α-zeros (for $\alpha \geq 70$) becomes unstable. This is due to the value of $\mathscr{H}_\alpha(-b)$ becoming too large to be stored in computer memory, leading to the overflow of the mantissa under double precision issue.

4.2.2 Tail behaviour of the FPT distribution for a homogeneous OU-process passing a constant barrier

With the given infinite series representation in Equation (4.2.12), we can analyse

the property of the FPT distribution for a homogeneous process passing a constant barrier. With the method given on page 114 of [65], its tail behaviour can be characterised by the "hazard rate function".

Lemma 4.2.2 *The distribution of the FPT of OU-process shown in Equation (4.2.1) to a constant barrier b is light-tailed, that is, the exponential moments exist up to the $\lambda\alpha_1$ order, where α_1 is given in Corollary 4.2.1. i.e.*

$$\mathbb{E}\left[e^{\theta\tau_{X,b}}\right] < \infty, \qquad \forall\theta < \lambda\alpha_1.$$

Proof. We consider the hazard rate function given in page 114 of [65]. The hazard rate function $r(t)$ for the FPT of OU-process in Equation (4.2.1) is specified by

$$r(t) := \frac{-\partial_t \bar{F}_X(x,t;b)}{\bar{F}_X(x,t;b)} = \frac{\sum_{k=1}^{\infty} B_k e^{-\lambda\alpha_k t}}{\sum_{k=1}^{\infty} C_k e^{-\lambda\alpha_k t}},$$

where $B_k = \lambda\alpha_k c_k \mathscr{H}_{\alpha_k}(-\sqrt{\frac{\lambda}{\sigma^2}}(x-\frac{\mu}{\lambda}))$ and $C_k = c_k \mathscr{H}_{\alpha_k}(-\sqrt{\frac{\lambda}{\sigma^2}}(x-\frac{\mu}{\lambda}))$. As shown in Remark 3.6 of [65], if $\lim_{t\to\infty} r(t) > 0$ exists, then the distribution is light-tailed and the exponential moments exist up to $\liminf_{t\to\infty} r(t)$.

In our case, since $\lambda > 0$ is given by the definition of an OU-process and $\{\alpha_k\}_{k\in\mathbb{N}}$ are ordered positive solutions to the equation

$$\mathscr{H}_\alpha\left(-\sqrt{\frac{\lambda}{\sigma^2}}\left(b-\frac{\mu}{\lambda}\right)\right) = 0$$

with respect to α, we have

$$\lim_{t\to\infty} r(t) = \lim_{t\to\infty} \frac{\sum_{k=1}^{\infty} B_k e^{-\lambda\alpha_k t}}{\sum_{k=1}^{\infty} C_k e^{-\lambda\alpha_k t}} = \lim_{t\to\infty} \frac{B_1 + \sum_{k=2}^{\infty} B_k e^{-\lambda(\alpha_k-\alpha_1)t}}{C_1 + \sum_{k=2}^{\infty} C_k e^{-\lambda(\alpha_k-\alpha_1)t}} = \lambda\alpha_1.$$

□

4.3 FPT transformation between an inhomogeneous and homogeneous OU-process hitting time-dependant barriers

In Section 4.2 we derive the eigenvalue expansion formulae to compute the survival function of the FPT for a homogeneous OU-process hitting a constant barrier. However, the homogeneous condition is usually too strong to model real events. For example, if we want to model the process with periodic structure by a homogeneous OU-process, the periodic structure in the process cannot be modelled. On the other hand, the barriers in reality will also vary with time. It is therefore natural and practically important to study the FPT for an inhomogeneous OU-process hitting a time-dependent barrier. The inhomogeneous OU-process is defined in Definition 4.3.2. We also present the definition of the FPT that a stochastic process hits a time-dependent barrier in Definition 4.3.1.

Definition 4.3.1 *Let $(Z_t)_{t\geq 0}$ be a continuous Markov process. The first-passage-time of $(Z_t)_{t\geq 0}$ with initial value $Z_0 = z$ to an upper time-dependent barrier $b(t)$, where $b(0) > z$, is defined by*

$$\mathscr{T}_{Z,b(t)} := \inf\{t \geq 0 : Z_t \geq b(t)\}. \tag{4.3.1}$$

The survival function of $\mathscr{T}_{Z,b(t)}$ is denoted by $\bar{F}_{\mathscr{T}_{Z,b(t)}}(t;z)$ and is given by

$$\bar{F}_{\mathscr{T}_{Z,b(t)}}(t;z) = \mathbb{P}\left(\mathscr{T}_{Z,b(t)} > t\right). \tag{4.3.2}$$

In particular, we focus on inhomogeneous OU-processes, which are defined as follows.

Definition 4.3.2 *Let $(W_t)_{t\geq 0}$ denote a Brownian motion on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$. A solution, $(Y_t)_{t\geq 0}$, to the stochastic differential equation*

$$\mathrm{d}Y_t = (\mu(t) - \lambda(t)Y_t)\,\mathrm{d}t + \sigma(t)\mathrm{d}W_t, \tag{4.3.3}$$

where $Y_0 = y \in \mathbb{R}$, is called an inhomogeneous Ornstein-Uhlenbeck process if the coefficients satisfy the existence and uniqueness conditions. i.e. For $\mu(t) : \mathbb{R}^+ \to \mathbb{R}$, $\lambda(t) : \mathbb{R}^+ \to \mathbb{R}^+$ and $\sigma(t) : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying (a) that $\exists C \in \mathbb{R}$ such that $|\mu(t) - \lambda(t)y'| + |\sigma(t)| \leq C(1 + |y'|)$ for $\forall y' \in \mathbb{R}$, and (b) that $\lambda(t)$ is bounded, $\forall t \geq 0$, the solution $(Y_t)_{t\geq 0}$ exists and is unique.

By Theorem 5.3.2 in [62], the properties (a) and (b) in Definition 4.3.2 ensure that the SDE (4.3.3) has a unique t-continuous solution. A sufficient condition for t-continuity is for $\mu(t)$, $\lambda(t)$ and $\sigma(t)$ to be bounded.

Next we show that the FPT distribution of an inhomogeneous OU-process crossing a time-dependent barrier is equivalent to the FPT distribution of a standardised OU-process crossing another time-dependent barrier.

Definition 4.3.3 *The mean-reverting scaling function $\alpha(t) : \mathbb{R}^+ \to \mathbb{R}^+$, the shift function $\beta(t) : \mathbb{R}^+ \to \mathbb{R}$, and the time-compensation function $\gamma(t) : \mathbb{R}^+ \to \mathbb{R}^+$ are to satisfy.*

a) $\alpha(t)$, $\beta(t)$ and $\gamma(t) \in C^1(\mathbb{R}^+)$ for $t > 0$;

b) $\gamma(t)$ is non-decreasing for $t > 0$;

c) $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ satisfy the ODE system

$$\begin{cases} \sigma(\lambda(t))\alpha(t)\sqrt{\gamma'(t)} & = 1 \\ \lambda(\gamma(t))\gamma'(t) - \frac{\alpha'(t)}{\alpha(t)} & = 1 \\ \alpha(t)\mu(\gamma(t))\gamma'(t) + \frac{\alpha'(t)\beta(t)}{\alpha(t)} - \beta'(t) - \beta(t)\lambda(\gamma(t))\gamma'(t) & = 0 \end{cases} \tag{4.3.4}$$

subject to the initial condition

$$\alpha(0) = \alpha_0 \in \mathbb{R}^+, \qquad \beta(0) = \beta_0 \in \mathbb{R}, \qquad \gamma(0) = 0 \tag{4.3.5}$$

where the constants α_0 and β_0 are pre-determined.

The time-dependent parameters $\mu(t)$, $\lambda(t)$ and $\sigma(t)$ are specified in Definition 4.3.2.

Lemma 4.3.1 (A sufficient condition for the uniqueness and existence of $\alpha(t)$, $\beta(t)$ and $\gamma(t)$)
For $\mu(t)$, $\lambda(t)$ and $\sigma(t)$ given in Definition 4.3.2, if $\lambda(t) \in C^1(\mathbb{R}^+)$ and $\sigma(t) \in C^2(\mathbb{R}^+)$, then the ODE system (4.3.4) has a unique local solution with initial conditions $\alpha(0) = \alpha_0 \in \mathbb{R}^+$, $\beta(0) = \beta_0 \in \mathbb{R}$ and $\gamma(0) = 0$.

Proof. The first two equations in system (4.3.4) can be rearranged such that

$$\alpha(t) = \frac{1}{\sigma(t)\sqrt{\gamma'(t)}}, \qquad \lambda(\gamma(t))\gamma'(t) = 1 + \frac{\alpha'(t)}{\alpha(t)}.$$

Substituting the first equation into the second, we obtain

$$\gamma'(t) = \frac{1}{\lambda(\gamma(t))}\left(1 - \frac{\sigma'(t)}{\sigma(t)} - \frac{1}{2\gamma(t)}\right).$$

Next we consider

$$f(t,\gamma) := \frac{1}{\lambda(\gamma)}\left(1 - \frac{\sigma'(t)}{\sigma(t)} - \frac{1}{2\gamma}\right).$$

Since $\lambda(t) \in C^1(\mathbb{R}^+)$ and $\sigma(t) \in C^2(\mathbb{R}^+)$, $f(t,\gamma)$ is continuous in t and $\partial f/\partial\gamma$ is continuous in γ. By the Picard-Lindelöf theorem, see page 38 of [46] for example, the unique local solution to $\gamma(t)$ is guaranteed, which implies the solution to $\alpha(t)$ also exists and is unique. Since

$$\alpha(t)\mu(\gamma(t))\gamma'(t) + \frac{\alpha'(t)\beta(t)}{\alpha(t)} - \beta'(t) - \beta(t)\lambda(\gamma(t))\gamma'(t) = 0$$

is a first-order linear ODE with respect to $\beta(t)$, whose solution is guaranteed to be

unique, the ODE system (4.3.4) admits a unique solution. □

Remark 4.3.1 *The necessary and sufficient conditions for the uniqueness and existence of $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ are non-trivial.*

For $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ specified as in Definition 4.3.3, an inhomogeneous OU-process is transformed into a standardised one as follows.

Proposition 4.3.1 *Consider the inhomogeneous OU-process $(Y_t)_{t\geq 0}$ given in Definition 4.3.2. Assume $\alpha(t), \beta(t)$ and $\gamma(t)$, given in Definition 4.3.3, satisfy the sufficient conditions in Lemma 4.3.1. Then the transformed process $(\alpha(t)Y_{\gamma(t)} - \beta(t))_{t\geq 0}$ is a standardised OU-process $(\widetilde{X}_t)_{t\geq 0}$ with initial value $\widetilde{X}_0 = \alpha_0 y - \beta_0$ almost surely.*

Proof. The solution to the SDE (4.3.3) is given by

$$\begin{aligned} Y_t =& \mathrm{e}^{-\int_0^t \lambda(u)\mathrm{d}u} \\ &\times \left[x + \int_0^t \mu(s) \exp\left(\int_0^s \lambda(u)\mathrm{d}u\right)\mathrm{d}s + \int_0^t \sigma(s)\exp\left(\int_0^s \lambda(u)\mathrm{d}u\right)\mathrm{d}W_s\right] \end{aligned}$$

Since $\gamma(\cdot)$ satisfies the sufficient condition in Lemma 4.3.1, $\gamma(\cdot)$ exists. Therefore,

$$\begin{aligned} Y_{\gamma(t)} =& \mathrm{e}^{-\int_0^{\gamma(t)} \lambda(u)\mathrm{d}u}\left[x + \int_0^{\gamma(t)} \mu(s)\exp\left(\int_0^s \lambda(u)\mathrm{d}u\right)\mathrm{d}s \right. \\ &\left. + \int_0^{\gamma(t)} \sigma(s)\exp\left(\int_0^s \lambda(u)\mathrm{d}u\right)\mathrm{d}W_s\right], \end{aligned}$$

and

$$\begin{aligned} Y_{\gamma(t)} =& \exp\left(-\int_0^t \lambda(\gamma(u))\gamma'(u)\mathrm{d}u\right) \\ &\times \left[x + \int_0^t \mu(\gamma(s))\gamma'(s)\exp\left(\int_0^s \lambda(\gamma(u))\gamma'(u)\mathrm{d}u\right)\mathrm{d}s \right. \\ &\left. + \int_0^t \sigma(\gamma(s))\sqrt{\gamma'(s)}\exp\left(\int_0^s \lambda(\gamma(u))\gamma'(u)\mathrm{d}u\right)\mathrm{d}W_s\right]. \end{aligned}$$

It follows that

$$\mathrm{d}Y_{\gamma(t)} = \left[\mu(\gamma(t))\gamma'(t) - \lambda(\gamma(t))\gamma'(t)Y_{\gamma(t)}\right]\mathrm{d}t + \sigma(\gamma(t))\sqrt{\gamma'(t)}\mathrm{d}W_t,$$

and hence

$$\begin{aligned}\mathrm{d}\widetilde{X}_t = &\left[\alpha'(t)\frac{\widetilde{X}_t + \beta(t)}{\alpha(t)} - \beta'(t) + \alpha(t)\mu(\gamma(t))\gamma'(t)\right.\\ &\left.-\alpha(t)\lambda(\gamma(t))\gamma'(t)\frac{\widetilde{X}_t + \beta(t)}{\alpha(t)}\right]\mathrm{d}t + \alpha(t)\sigma(t)\sqrt{\gamma'(t)}\mathrm{d}W_t\\ = &-\widetilde{X}_t\mathrm{d}t + \mathrm{d}W_t.\end{aligned}$$

In the last step Definition 4.3.3 is used. We thus have that $(\widetilde{X}_t)_{t\geq 0}$ is a standardised OU-process. □

Now we are in the position to present the main theorem that links the FPT distribution functions of the inhomogeneous and the standardised OU-processes.

Theorem 4.3.1 *Let $(Y_t)_{t\geq 0}$ be the inhomogeneous OU-process in Definition 4.3.2, and assume that the ODE (4.3.4) has a unique solution. Then,*

$$\bar{F}_{\mathscr{T}_{Y,b(t)}}(t;y) = \bar{F}_{\mathscr{T}_{\widetilde{X},g(t)}}\left(\gamma^{-1}(t);\tilde{x}\right) \tag{4.3.6}$$

where $(\widetilde{X}_t)_{t\geq 0}$ is a standardised OU-process with initial value $\widetilde{X}_0 = \tilde{x} = \alpha_0 y - \beta_0$, and

$$g(t) = \alpha(t)b(\gamma(t)) - \beta(t).$$

An equivalent statement is that $\mathscr{T}_{Y,b(t)}$ and $\gamma(\mathscr{T}_{\widetilde{X},g(t)})$ are equal in distribution.

Proof. First, we show $\mathscr{T}_{Y,b(t)}$ and $\gamma\left(\mathscr{T}_{\widetilde{X},\alpha(t)b(\gamma(t))-\beta(t)}\right)$ are equal in distribution. We have that

$$\begin{aligned}\mathscr{T}_{Y,b(t)} &= \inf\{t>0 : Y_t \geq b(t)\} = \inf\left\{\gamma(t)>0 : Y_{\gamma(t)} \geq b(\gamma(t))\right\}\\ &= \inf\left\{\gamma(t)>0 : \alpha(t)Y_{\gamma(t)} - \beta(t) \geq \alpha(t)b(\gamma(t)) - \beta(t)\right\}.\end{aligned}$$

Since $\gamma(\cdot)$ is monotone, non-decreasing and positive, we deduce

$$\mathscr{T}_{Y,b(t)} = \gamma\left(\inf\left\{t>0 : \alpha(t)Y_{\gamma(t)} - \beta(t) \geq \alpha(t)b(\gamma(t)) - \beta(t)\right\}\right).$$

By Proposition 4.3.1, we know that the process $(\alpha(t)Y_{\gamma(t)} - \beta(t))_{t\geq 0}$ has the law of a standardised OU-process. Therefore,

$$\begin{aligned}\mathscr{T}_{Y,b(t)} =& \gamma\left(\inf\left\{t>0 : \alpha(t)Y_{\gamma(t)} - \beta(t) \geq \alpha(t)b(\gamma(t)) - \beta(t)\right\}\right)\\ =& \gamma\left(\mathscr{T}_{\widetilde{X},\alpha(t)b(\gamma(t))-\beta(t)}\right).\end{aligned}$$

Then, it follows that

$$\begin{aligned}\bar{F}_{\mathscr{T}_{Y,b(t)}}(t;x) =& \mathbb{P}\left(\mathscr{T}_{Y,b(t)} > t \,|\, Y_0 = x\right) = \mathbb{P}\left(\gamma\left(\mathscr{T}_{\widetilde{X},\alpha(t)b(\gamma(t))-\beta(t)}\right) > t \,|\, \widetilde{X}_0 = \widetilde{x}\right)\\ =& \mathbb{P}\left(\mathscr{T}_{\widetilde{X},\alpha(t)b(\gamma(t))-\beta(t)} > \gamma^{-1}(t) \,|\, \widetilde{X}_0 = \widetilde{x}\right) = \bar{F}_{\mathscr{T}_{\widetilde{X},g(t)}}\left(\gamma^{-1}(t);\widetilde{x}\right).\end{aligned}$$

□

Example 4.3.1 (Seasonal trend) One example of above is to apply a seasonality function to the mean-reverting level function $\mu(t)$. In this example, we show how we can use Theorem 4.3.1 to transform the problem for an inhomogeneous OU-process hitting a constant barrier to the problem for a standardised OU-process hitting a periodic barrier.

We consider the inhomogeneous OU-process $(Y_t)_{t\geq 0}$, parametrised by $\mu(t) = A\sin(\theta t + \varphi)$, $\lambda(t) = \lambda$ and $\sigma(t) = \sigma$, with initial value $Y_0 = x$, where A, θ, $\varphi \in \mathbb{R}$ and $\lambda, \sigma > 0$. The constant barrier is denoted by b. The mean-reverting scaling function $\alpha(t)$ and time-compensation function $\gamma(t)$ are given by $\alpha(t) = \sqrt{\lambda}/\sigma$ and $\gamma(t) = t/\lambda$. Then $\beta(t)$ satisfies

$$\beta'(t) + \beta(t) = \mu\left(\frac{t}{\lambda}\right)\frac{1}{\sigma\sqrt{\lambda}}. \tag{4.3.7}$$

The associated ODE (4.3.7), in this particular case, has unique solution

$$\beta(t) = Be^{-t} + \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\frac{\theta}{\lambda}t + \varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right), \tag{4.3.8}$$

where B is a constant so to match the initial condition. For convenience, we let $B = 0$ by imposing the initial condition

$$\beta(0) = \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right),$$

which, by Theorem 4.3.1, means that the standardised OU-process $(\widetilde{X}_t)_{t\geq 0}$ starts from

$$\widetilde{X}_0 = \frac{\sqrt{\lambda}}{\sigma}X_0 - \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right).$$

Then we have a particular solution for $\beta(t)$:

$$\beta(t) = \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\frac{\theta}{\lambda}t + \varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right).$$

By Theorem 4.3.1, we only need to study the probability of a standardised OU-process with initial value,

$$\widetilde{X}_0 = \widetilde{x} = \frac{x\sqrt{\lambda}}{\sigma} - \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right)$$

crossing a periodic barrier

$$g(t) = \frac{b\sqrt{\lambda}}{\sigma} - \frac{A\sqrt{\lambda}}{\sigma\sqrt{\lambda^2+\theta^2}} \sin\left(\frac{\theta}{\lambda}t + \varphi - \arctan\left(\frac{\theta}{\lambda}\right)\right).$$

4.4 Multiple crossings of an inhomogeneous OU-process

For an inhomogeneous OU-process with parameter functions in $C^1(\mathbb{R}^+)$, one can transform its barrier-crossing problem to one associated with the standardised OU-process and a time-dependent barrier function. The time-dependent barrier can be

approximated by a piecewise constant function. This is due to the fact that any continuous function can be approximated with a piecewise constant function to arbitrary accuracy given a sufficiently large number of partitions. The problem thus reduces to a multiple-crossing problem for a standardised OU-process.

Alternatively, one may directly use the piecewise constant approximation for the parameter functions of the inhomogeneous OU-process. This alternative method leads to a multiple-crossing problem for a locally-homogeneous OU-process.

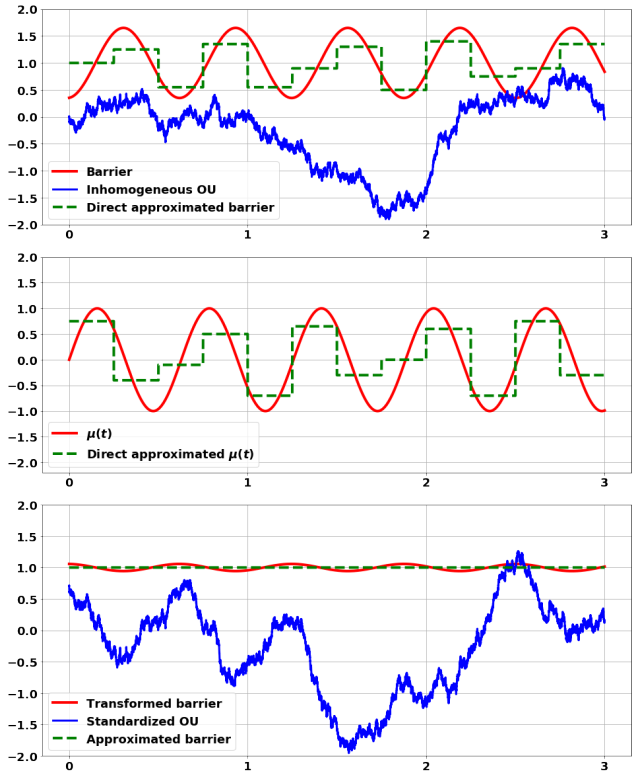

Figure 4.5: An example where the *transformation method* is advantageous and the piecewise-constant approximation is applied. The inhomogeneous OU-process crossing the time-dependent barrier $b(t) = 1 + 0.65\sin(10t + \arctan(10))$ is shown in the upper panel. The parameter functions of the OU-process are $\mu(t) = \sin(10t)$ and $\lambda = \sigma = 1$ as shown in the middle panel. The original problem can be transformed to a standardised OU-process with smoother time-dependent barrier, see the lower panel.

The first method transforms the inhomogeneous OU-process to a global standardised OU-process by solving the ODE system given in Definition 4.3.3. It uses piecewise constant functions to approximate the time-varying barriers. This scheme requires further conditions to be satisfied, such as continuity for parameter func-

tions, for the transformation to be well-defined. In addition, solving the ODE system can be difficult. The second method does not rely on such a transformation, however, it results in a locally homogeneous OU-process with piecewise constant barriers, where the time steps for the barriers and OU-parameters may not necessarily match.

In applications, the method one should select is decided on a case-by-case basis. In Figure 4.5, we simulate a time-inhomogeneous OU-process with a time-dependent barrier. The application of the first method can offset most of the time-dependences from the parameters and the barrier, however, the direct approximation method will lead to higher approximation error. In order to reach the same level of accuracy, one may have to approximate using more time segments, which complicates the problem.

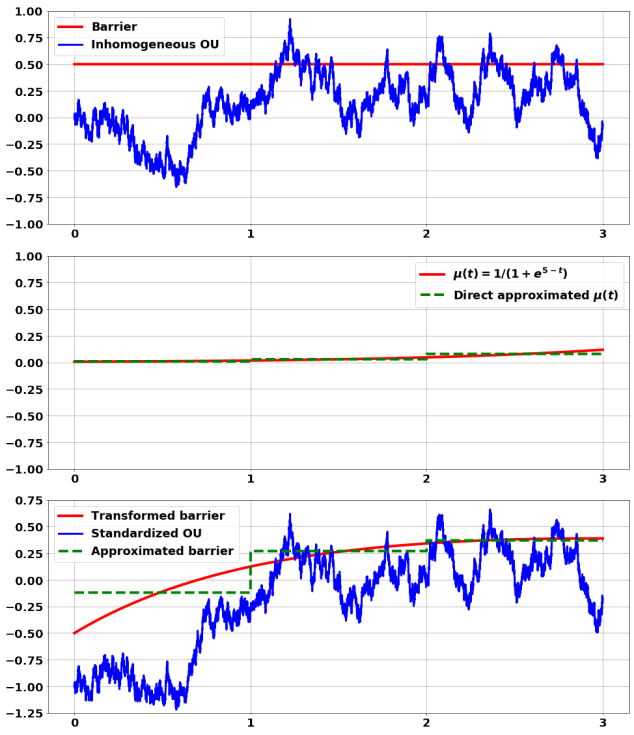

Figure 4.6: An example where the piecewise-constant approximation is inefficient and the *transformation method* is disadvantageous. The inhomogeneous OU-process crossing the constant barrier $b(t) = 0.5$ is shown in the upper panel. The parameter functions of the OU-process are $\mu(t) = 1/(1+e^{5-t})$ and $\lambda = \sigma = 1$ as shown in the middle panel. The original problem is transformed to a standardised OU-process with a steeper time-dependent barrier, see the lower panel.

However, this does not mean that the first method is always better than the direct approximation approach. For example in Figure 4.6, with the same number of discretisations, the transformation method leads to a higher approximation error.

In general, if any of the three OU parameter functions are not in $C^1(\mathbb{R}^+)$, one should use the direct approximation method. Otherwise, when applying the transformation method, the transformed barrier function $g(t) = \alpha(t)b(\gamma(t)) - \beta(t)$ can be written as

$$g(t) = \mathrm{e}^{-t}\left[\alpha_0 \mathrm{e}^{\int_0^{\gamma(t)}\lambda(s)\mathrm{d}s} b(\gamma(t)) - \beta_0 - \alpha_0\beta_0 \int_0^{\gamma(t)} \mathrm{e}^{\int_0^s \lambda(s)\mathrm{d}s}\mu(s)\mathrm{d}s\right],$$

where $\gamma(t)$ is obtained from the equation

$$\alpha_0 \mathrm{e}^{\int_0^{\gamma(t)}\lambda(s)\mathrm{d}s - t}\sigma(\gamma(t))\sqrt{\gamma'(t)} = 1.$$

It is difficult to come up with a general principle on which method makes less estimation error by approximations with constants. However, this can be achieved under certain conditions. For example, we assume the inhomogeneous OU-process in Definition 4.3.2 has constant coefficients $\sigma(t) = \sigma$, $\lambda(t) = 1$, $\mu(t) \in C^2([t_1, t_2])$, and the barrier function is a constant $b(t) = b$. Under this assumption, we have the transformed barrier function $g(t)$ defined in domain $[t_1, t_2]$. Therefore, if $\forall t \in [t_1, t_2]$, $g(t)$ and $\mu(t)$ are monotone, and

$$|g(t_2) - g(t_1)| < |\mu(t_2) - \mu(t_1)|,$$

then the transformation method leads to less approximation error. That is, using a constant to approximate $g(t)$ in domain $[t_1, t_2]$ is more accurate than using a constant to approximate $\mu(t)$. This is because the range of monotone transformed function $g(t)$ is larger than than the range of monotone $\mu(t)$.

Next we investigate the case where $\mu(t)$, $\lambda(t)$ and $\sigma(t)$, the parameter functions of the inhomogeneous OU-process $(Y_t)_{t\geq 0}$, and the barrier function $b(t)$ are càdlàg piecewise constant functions.

Let the parameter functions b specified by

$$\mu(t) = \sum_{i=1}^{N^{(\mu)}} \mu_i \mathbb{I}\left(t \in [t_{i-1}^{(\mu)}, t_i^{(\mu)})\right), \quad \lambda(t) = \sum_{i=1}^{N^{(\lambda)}} \lambda_i \mathbb{I}\left(t \in [t_{i-1}^{(\lambda)}, t_i^{(\lambda)})\right)$$
$$\sigma(t) = \sum_{i=1}^{N^{(\sigma)}} \sigma_i \mathbb{I}\left(t \in [t_{i-1}^{(\sigma)} t_i^{(\sigma)})\right), \quad b(t) = \sum_{i=1}^{N} b_i \mathbb{I}(t \in [t_{i-1}, t_i))$$

$\forall i = 1, 2, ..., N$, where $\lambda_i, \sigma_i \in \mathbb{R}^+$, $\mu_i, b_i \in \mathbb{R}$ and $b_0 > Y_0$. Here, we consider a finite-time horizon where $t_{N^{(\mu)}}^{(\mu)} = t_{N^{(\lambda)}}^{(\lambda)} = t_{N^{(\sigma)}}^{(\sigma)} = t_N$, and $\mathbb{I}(\cdot)$ denotes the indicator function. We study the following probabilities:

$$\mathbb{P}\left(M_{t_0,t_1}^Y < b_1, M_{t_1,t_2}^Y < b_2, \ldots, M_{t_{N-1},t_N}^Y < b_N\right), \tag{4.4.1}$$
$$\mathbb{P}\left(M_{t_0,t_1}^Y \geq b_1, M_{t_1,t_2}^Y \geq b_2, \ldots, M_{t_{N-1},t_N}^Y \geq b_N\right), \tag{4.4.2}$$

where $M_{t_{i-1},t_i}^Y = \sup_{t \in (t_{i-1},t_i]} Y_t$. In particular, probability (4.4.1) is equal to the probability that the FPT of this inhomogeneous OU-process is larger than t_N; probability (4.4.2) is the probability that the inhomogeneous OU-process crosses the barrier in each interval, i.e. the multiple crossing probability. We may consider the following discretisation schemes:

1) Matching time-discretisation for $\mu(t)$, $\lambda(t)$, $\sigma(t)$ and $b(t)$, i.e. $t_i^{(\mu)} = t_i^{(\lambda)} = t_i^{(\sigma)} = t_i$ for all $i = 0, 1, \ldots, N$.

2) Matching time-discretisation for $\mu(t)$, $\lambda(t)$ and $\sigma(t)$ only. i.e. $t_i^{(\mu)} = t_i^{(\lambda)} = t_i^{(\sigma)}$ for all $i = 0, 1, \ldots, N^{(\mu)}$.

3) Non-matching time-discretisations for any of $\mu(t)$, $\lambda(t)$, $\sigma(t)$ and $b(t)$.

One can show that the probabilities (4.4.1) and (4.4.2) in the last two cases can be further reduced to the first case by utilising Theorem 4.4.1. Therefore, in what follows, we will focus on the first case unless specified otherwise, and we shall show the reduction methods from the case 2) and 3) to case 1) in Section 4.4.3.

4.4.1 The joint distribution and multivariate survival function for multiple maxima of a continuous Markov process in consecutive intervals

We begin with a general theorem for continuous Markov process. We recall the definition of a Markov process, see for instance Section 5.2.3 in [8]. In this section, we consider the probability space $(\Omega, \mathscr{F}, \mathbb{P})$ that is equipped with a filtration $\mathscr{F}_t$.

Definition 4.4.1 *Let $(\mathscr{F}_t)$ be the natural filtration generated by $(Z_s)_{s\in[0,t]}$. A Markov process $(Z_t)_{t\geq 0}$ satisfies $\mathbb{P}(A \mid \mathscr{F}_t) = \mathbb{P}(A \mid Z_t)$, for all $A \in \sigma(Z_u : u \geq t)$.*

Lemma 4.4.1 *If $(Z_t)_{t\geq 0}$ is a Markov process, then we have $P(A \cap B|Z_t) = P(A|Z_t)P(B|Z_t)$ for all $A \in \sigma(Z_u : u \geq t), B \in \sigma(Z_u : u \leq t)$.*

Proof. This is straightforward and shown in most standard textbooks on probability, see for example Section 5.2.3 in [8]. □

We now go on to present Theorem 4.4.1 and prove it by the conditional independence property. We consider the time steps $0 = t_0 < t_1 < \cdots < t_N = T$, and denote the barrier level in the interval $[t_{i-1}, t_i)$ by b_i. In Theorem 4.4.1, we calculate the joint distribution function, or the survival function, of the maxima of a continuous Markov process in each interval. Here, we denote $M_{t_{i-1},t_i} := \sup_{t\in(t_{i-1},t_i]} Z_t$.

Theorem 4.4.1 *Let $b_1, \ldots, b_N \in \mathscr{D} := Dom(Z_t)$. The joint distribution and survival functions of the maxima of a continuous Markov process $(Z_t)_{t\geq 0}$ in consecutive intervals are given, respectively, by*

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, \cdots, M_{t_{N-1},t_N} < b_N \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \psi_1(t_0, t_1, z_0, z_1, b_1) \cdots \int_{\mathscr{D}} \psi_{N-1}(t_{N-2}, t_{N-1}, z_{N-2}, z_{N-1}, b_{N-1}) \\
&\qquad \times q(t_{N-1}, t_N, Z_{N-1})\, \mathrm{d}z_{N-1} \cdots \mathrm{d}z_1,
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} \geq b_1, \cdots, M_{t_{N-1},t_N} \geq b_N \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \kappa_1\left(t_0, t_1, z_0, z_1, b_1\right) \cdots \int_{\mathscr{D}} \kappa_{N-1}\left(t_{N-2}, t_{N-1}, z_{N-2}, z_{N-1}, b_{N-1}\right) \\
&\qquad \times \bar{q}\left(t_{N-1}, t_N, Z_{N-1}\right) \mathrm{d}z_{N-1} \cdots \mathrm{d}z_1,
\end{aligned}$$

where

$$\begin{aligned}
\psi_i(t_{i-1}, t_i, z_{i-1}, z_i, b_i) &= \mathbb{P}\left(M_{t_{i-1},t_i} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) p(t_{i-1}, t_i, z_{i-1}, z_i), \\
\kappa_i(t_{i-1}, t_i, z_{i-1}, z_i, b_i) &= \mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) p(t_{i-1}, t_i, z_{i-1}, z_i), \\
q\left(t_{N-1}, t_N, Z_{N-1}\right) &= \mathbb{P}\left(M_{t_{N-1},t_N} < b_N \,|\, Z_{t_{N-1}} = z_{N-1}\right), \\
\bar{q}\left(t_{N-1}, t_N, Z_{N-1}\right) &= 1 - q\left(t_{N-1}, t_N, Z_{N-1}\right).
\end{aligned}$$

Here $p(t_{i-1}, t_i, z_{i-1}, z_i)$, where $i = 1, \ldots, N-1$, is the transition density function of the process $(Z_t)_{t \geq 0}$ from state z_{i-1} at time t_{i-1} to state z_i at time t_i.

Proof. Here, we show the proof for the joint distribution function. The proof for the joint survival function is similar, since we also utilise the Markov conditional independence property. We proceed with a proof by induction.

(1) Case $N = 2$: We know that

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2 \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2 \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \mathbb{P}_{z_0}\left(Z_{t_1} \in \mathrm{d}z_1 \,|\, Z_0 = z_0\right).
\end{aligned}$$

Since

$$\{M_{t_0,t_1} < b_1\} \in \sigma(Z_s : s \leq t_1)$$

and

$$\{M_{t_1,t_2} < b_2\} \in \sigma(Z_s : s \geq t_1),$$

we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2 \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \mathbb{P}\left(M_{t_0,t_1} < b_1 \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \mathbb{P}\left(M_{t_1,t_2} < b_2 \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \\
&\qquad \times \mathbb{P}\left(Z_{t_1} \in \mathrm{d}z_1 \,|\, Z_0 = z_0\right),
\end{aligned}$$

by Lemma 4.4.1. Then by the Markov property, we obtain

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2 \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \psi_1\left(t_0, t_1, z_0, z_1, b_1\right) q\left(t_1, t_2, Z_1\right) \mathrm{d}z_1,
\end{aligned}$$

which is the case $N = 2$ in Theorem 4.4.1.

(2) Consider Theorem 4.4.1 for $N = K$ such that

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2, \cdots, M_{t_{K-1},t_K} < b_K \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \psi_1\left(t_0, t_1, z_0, z_1, b_1\right) \cdots \int_{\mathscr{D}} \psi_{K-1}\left(t_{K-2}, t_{K-1}, z_{K-2}, z_{K-1}, b_{K-1}\right) \\
&\qquad \times q\left(t_{K-1}, t_K, Z_{K-1}\right) \mathrm{d}z_{K-1} \cdots \mathrm{d}z_1.
\end{aligned}$$

Now consider the case when $N = K + 1$. We have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2, \cdots, M_{t_{K-1},t_K} < b_K, M_{t_K,t_{K+1}} < b_{K+1} \,|\, Z_0 = z_0\right) \\
&= \int_{\mathscr{D}} \mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2, \cdots, M_{t_K,t_{K+1}} < b_{K+1} \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \\
&\qquad \times \mathbb{P}_{z_0}\left(Z_{t_1} \in \mathrm{d}z_1 \,|\, Z_0 = z_0\right).
\end{aligned}$$

Then, by Lemma 4.4.1, since

$$\left\{M_{t_0,t_1} < b_1\right\} \in \sigma(Z_s : s \leq t_1)$$

and

$$\left\{M_{t_1,t_2} < b_2, \cdots, M_{t_K,t_{K+1}} < b_{K+1}\right\} \in \sigma(Z_s : s \geq t_1),$$

we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} < b_1, M_{t_1,t_2} < b_2, \cdots, M_{t_K,t_{K+1}} < b_{K+1} \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \\
&= \mathbb{P}\left(M_{t_0,t_1} < b_1 \,|\, Z_{t_1} = z_1, Z_0 = z_0\right) \\
&\qquad \times \mathbb{P}\left(M_{t_1,t_2} < b_2, \cdots, M_{t_K,t_{K+1}} < b_{K+1} \,|\, Z_{t_1} = z_1, Z_0 = z_0\right).
\end{aligned}$$

For $N = K$, by the Markov property, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_1,t_2} < b_2, M_{t_2,t_3} < b_3, \cdots, M_{t_K,t_{K+1}} < b_{K+1} \,|\, Z_{t_1} = z_1\right) \\
&= \int_{\mathscr{D}} \psi_1\left(t_1, t_2, z_1, z_2, b_2\right) \cdots \int_{\mathscr{D}} \psi_K\left(t_{K-1}, t_K, z_{K-1}, z_K, b_K\right) \\
&\qquad \times q\left(t_K, t_{K+1}, Z_K\right) \mathrm{d}z_K \cdots \mathrm{d}z_2.
\end{aligned}$$

By iterated substitutions, the proof is complete for the case $N = K + 1$.

□

Now we decompose the joint distribution and survival functions of the maxima of a continuous Markov process in consecutive intervals into three components:

a) The distribution or survival function of the maximum of the continuous Markov process in a given interval conditional on its starting value and terminal value;

b) The transition density function in a given interval;

c) The distribution function of the maximum of the continuous Markov process in a given interval conditional on its starting value only.

The third item is equivalent to the FPT distribution for a Markov process to cross a constant barrier in a given interval. Essentially, the first item involves the calculation of the maximum of a continuous Markov bridge, which requires use of Proposition 4.4.1.

Proposition 4.4.1 *Let $(Z_t)_{t\geq 0}$ be a continuous Markov process where $Z_0 = z$, and $\tau_{Z,b} := \inf\{t \geq 0 : Z_t \geq b\}$. Then,*

$$\mathbb{P}\left(M_{0,T} \geq b \,|\, Z_T = z', Z_0 = z\right) = \begin{cases} \int_0^T \frac{p(t,T,b,z')}{p(0,T,z,z')} f_{\tau_{Z,b}}(t;z)\mathrm{d}t, & \text{if } z, z' < b, \\ \\ 1, & \text{otherwise,} \end{cases}$$

where $p(t,T,b,z')$ denotes the transition density function of $(Z_t)_{t\geq 0}$ from state b at time t to state z' at time T, and $f_{\tau_{Z,b}}(t;z)$ is the probability density function of first-passage-time $\tau_{Z,b}$ with $Z_0 = z$.

Proof. We first consider the case that $z, z' < b$. Since the two events $\left\{M_{0,T} \geq b\right\}$ and $\left\{\tau_{Z,b} \leq T\right\}$ are equivalent,

$$\begin{aligned} \mathbb{P}\left(M_{0,T} \geq b \,|\, Z_T = z', Z_0 = z\right) =& \mathbb{P}\left(\tau_{Z,b} \leq T \,|\, Z_T = z', Z_0 = z\right) \\ =& \int_0^T f_{\tau_{Z,b}}\left(t \,|\, Z_T = z', Z_0 = z\right) \mathrm{d}t, \end{aligned}$$

where $f_{\tau_{Z,b}}\left(t \,|\, Z_T = z', Z_0 = z\right)$ denotes the conditional density function of first-passage-time $\tau_{Z,b}$. By Bayes' theorem, we have

$$\mathbb{P}\left(M_{0,T} \geq b \,|\, Z_T = z', Z_0 = z\right) = \int_0^T p\left(0,T,z,z' \,|\, \tau_{Z,b} = t\right) \frac{f_{\tau_{Z,b}}(t;z)}{p(0,T,z,z')} \mathrm{d}t.$$

Here, $p\left(0,T,z,z' \,|\, \tau_{Z,b} = t\right)$ denotes the conditional transition density of $(Z_t)_{t\geq 0}$ from state b at time t to state z' at time T. Since (Z_t) is a continuous process, we have $\left\{\tau_{Z,b} \geq t\right\} \cap \{Z_t \geq b\} = \left\{\tau_{Z,b} \geq t\right\}$. Hence,

$$p\left(0,T,z,z' \,|\, \tau_{Z,b} = t\right) = p\left(0,T,z,z' \,|\, \tau_{Z,b} = t, Z_t = b\right).$$

Because $\left\{\tau_{Z,b} \geq t\right\} \in \sigma\left(Z_s : 0 \leq s \leq t\right)$ and $\left\{Z_T \geq z'\right\} \in \sigma\left(Z_s : t < s \leq T\right)$, by Lemma 4.4.1, we have

$$p\left(0,T,z,z' \,|\, \tau_{Z,b} = t, Z_t = b\right) = p\left(0,T,z,z' \,|\, Z_t = b\right) = p\left(t,T,b,z'\right).$$

Therefore,

$$\mathbb{P}\left(M_{0,T} \geq b \,|\, Z_T = z', Z_0 = z\right) = \int_0^T \frac{p(t,T,b,z')}{p(0,T,z,z')} f_{\tau_{Z,b}}(t;z)\mathrm{d}t.$$

In the case either $z \geq b$ or $z' \geq b$, because of the continuous property of process $(Z_t)_{t\geq 0}$, the probability turns to be 1 obviously. □

With Proposition 4.4.1 and Theorem 4.4.1, we are able to at least approximate the joint distribution and survival functions of the maxima of a continuous Markov process in consecutive intervals, provided that we know its transition density function and its FPT density for a constant barrier.

4.4.2 Simplified computation of survival functions via restrictions

Now we present a theorem which simplifies the calculation of the survival function in Theorem 4.4.1 based on the assumption that $\{Z_{t_0} < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\}$. We can prove that, if at the end of each interval the terminal value $Z_{t_{i-1}}$ of the process is lower than the barrier level b_i in the subsequent time interval, the nested integral simplifies to a product of single integrals.

Theorem 4.4.2 *Given that $\{Z_{t_0} < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\}$, the joint survival function of the maxima of a continuous Markov process $(Z_t)_{t\geq 0} \in \mathbb{R}$ in consecutive left-open and right-closed time intervals is given by*

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} \geq b_1, \cdots, M_{t_{N-1},t_N} \geq b_N \,|\, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right) \\
&= \mathbb{P}\left(M_{t_0,t_1} \geq b_1 \,|\, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right) \\
&\quad \prod_{i=2}^{N} \left[\int_{-\infty}^{b_i} \frac{\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = x_i, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right)}{\mathbb{P}\left(M_{t_{i-2},t_{i-1}} \geq b_{i-1} \,|\, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right)} \right. \\
&\quad \times \mathbb{P}\left(M_{t_{i-2},t_{i-1}} \geq b_{i-1} \,|\, Z_{t_{i-1}} = x_i, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right) \\
&\quad \left. \times \mathbb{P}\left(Z_{t_{i-1}} \in \mathrm{d}x_i \,|\, Z_{t_0} = z_0 < b_1, Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right) \right] \qquad (4.4.3)
\end{aligned}$$

Proof. Let $\tau^{(i)} = \inf\{t \geq t_{i-1} : Z_t = b_i\}$, $\forall\, i = 1, 2, \cdots, N$. The event $\left\{M_{t_{i-1},t_i} \geq b_i\right\}$

is equivalent to $\left\{\tau^{(i)} \le t_i\right\}$, and furthermore set

$$C = \left\{Z_{t_1} < b_2, \cdots, Z_{t_{N-1}} < b_N\right\}.$$

Since $\left\{\tau^{(i-1)} \le t_{i-1}\right\} \subseteq \left\{Z_{\tau^{(i-1)}} = b_{i-1}\right\}$, $\forall\, i = 1,2,\cdots,N$, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, M_{t_0,t_1} \ge b_1, \cdots, M_{t_{i-2},t_{i-1}} \ge b_{i-1}\right) \\
&= \mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, M_{t_0,t_1} \ge b_1, \cdots, M_{t_{i-3},t_{i-2}} \ge b_{i-2}, \tau^{(i-1)} \le t_{i-1}\right) \\
&= \mathbb{P}\Big(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, M_{t_0,t_1} \ge b_1, \cdots, M_{t_{i-3},t_{i-2}} \ge b_{i-2}, \tau^{(i-1)} \le t_{i-1}, \\
&\qquad\qquad Z_{\tau^{(i-1)}} = b_{i-1}\Big)
\end{aligned}$$

Furthermore, since

$$\left\{M_{t_0,t_1} \ge b_1, \cdots, M_{t_{i-3},t_{i-2}} \ge b_{i-2}\right\} \subset \mathscr{F}_{\tau^{(i-1)}}$$

and

$$\left\{M_{t_{i-1},t_i} \ge b_i\right\} \subset \mathscr{F}_{t_i} \backslash \mathscr{F}_{\tau^{(i-1)}},$$

by Lemma 4.4.1, we obtain

$$\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, M_{t_0,t_1} \ge b_1, \cdots, M_{t_{i-2},t_{i-1}} \ge b_{i-1}\right) \\
&= \mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, \tau^{(i-1)} \le t_{i-1}, Z_{\tau^{(i-1)}} = b_{i-1}\right).
\end{aligned}$$

Since

$$\left\{\tau^{(i-1)} \le t_{i-1}\right\} \subseteq \left\{Z_{\tau^{(i-1)}} = b_{i-1}\right\},$$

the above formula equals to

$$\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, \tau^{(i-1)} \le t_{i-1}\right) \\
&= \mathbb{P}\left(M_{t_{i-1},t_i} \ge b_i \,|\, Z_0 = z_0, C, M_{t_{i-2},t_{i-1}} \ge b_{i-1}\right).
\end{aligned}$$

It means that conditional on event C, the discrete process $(L_i)_{i\in\mathbb{N}}$ defined by $L_i =$

$\mathbb{I}_{M_{t_{i-1},t_i}\geq b_i}$ is a discrete Markov process. Hence, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1}\geq b_1,\cdots,M_{t_{N-1},t_N}\geq b_N\,|\,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)\\
&=\mathbb{P}\left(M_{t_{N-1},t_N}\geq b_N\,|\,Z_0=z_0,C,M_{t_0,t_1}\geq b_1,\cdots,M_{t_{N-2},t_{N-1}}\geq b_{N-1}\right)\\
&\quad\times\mathbb{P}\left(M_{t_{N-2},t_{N-1}}\geq b_{N-1}\,|\,Z_0=z_0,C,M_{t_0,t_1}\geq b_1,\cdots,M_{t_{N-3},t_{N-2}}\geq b_{N-2}\right)\\
&\quad\times\cdots\times\mathbb{P}\left(M_{t_0,t_1}\geq b_1\,|\,Z_0=z_0,C\right)\\
&=\mathbb{P}\left(M_{t_0,t_1}\geq b_1\,|\,Z_0=z_0,C\right)\prod_{i=2}^{N}\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_0=z_0,C,M_{t_{i-2},t_{i-1}}\geq b_{i-1}\right).
\end{aligned}$$

Based on the Markov property, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_0=z_0,C,M_{t_{i-2},t_{i-1}}\geq b_{i-1}\right)\\
&=\int_{-\infty}^{b_i}\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_{t_{i-1}}=x,C\right)\mathbb{P}\left(Z_{t_{i-1}}\in\mathrm{d}x\,|\,Z_0=z_0,M_{t_{i-2},t_{i-1}}\geq b_{i-1},C\right)
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1}\geq b_1,\cdots,M_{t_{N-1},t_N}\geq b_N\,|\,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)\\
&=\mathbb{P}\left(M_{t_0,t_1}\geq b_1\,|\,Z_0=z_0,C\right)\\
&\times\prod_{i=2}^{N}\left[\int_{-\infty}^{b_i}\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_{t_{i-1}}=x,C\right)\mathbb{P}\left(Z_{t_{i-1}}\in\mathrm{d}x\,|\,Z_0=z_0,M_{t_{i-2},t_{i-1}}\geq b_{i-1},C\right)\right]\\
&=\mathbb{P}\left(M_{t_0,t_1}\geq b_1\,|\,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)\\
&\quad\prod_{i=2}^{N}\left[\int_{-\infty}^{b_i}\frac{\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_{t_{i-1}}=x_i,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)}{\mathbb{P}\left(M_{t_{i-2},t_{i-1}}\geq b_{i-1}\,|\,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)}\right.\\
&\quad\times\mathbb{P}\left(M_{t_{i-2},t_{i-1}}\geq b_{i-1}\,|\,Z_{t_{i-1}}=x_i,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)\\
&\quad\left.\times\mathbb{P}\left(Z_{t_{i-1}}\in\mathrm{d}x_i\,|\,Z_{t_0}=z_0<b_1,Z_{t_1}<b_2,\cdots,Z_{t_{N-1}}<b_N\right)\right]
\end{aligned}$$

□

We can simplify the previous nested integral in Theorem 4.4.1 to the product of multiple single integrals under restrictions.

4.4.3 Non-matching time-discretisation

As discussed before, we may have non-matching time-discretisation schemes for the piecewise constant functions $\mu(t)$, $\lambda(t)$, $\sigma(t)$ and $b(t)$. In such a situation, the process is still continuous and Markov. By Theorem 4.4.1, if we can obtain

$$\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) \text{ and } \mathbb{P}\left(M_{t_{N-1},t_N} \geq b_N \,|\, Z_{t_{N-1}} = z_{N-1}\right)$$

$\forall i = 1,\ldots,N-1$, the joint distribution and survival function for the maxima of the inhomogeneous OU-process in consecutive intervals can still be calculated. We have the following two sub-cases for non-matching time-discretisations in the interval $[t_{i-1},t_i]$.

Case 1: Matching time-discretisation for $\mu(t)$, $\lambda(t)$ and $\sigma(t)$, but non-matching for $b(t)$. An example of this case is shown in Figure 4.7. Here, the time-discretisations for $\mu(t)$, $\lambda(t)$ and $\sigma(t)$ are the same.

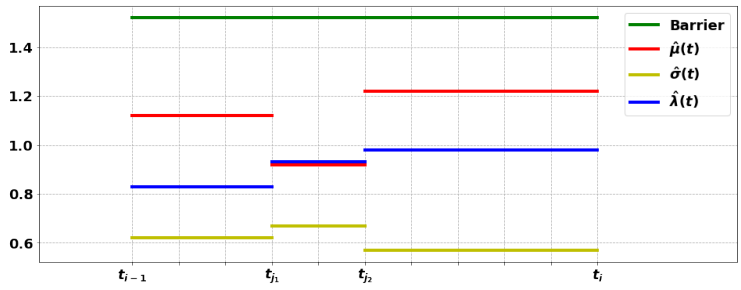

Figure 4.7: Matching time-discretisation for $\mu(t)$, $\lambda(t)$ and $\sigma(t)$, but different for $b(t)$.

In this case, $\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}\right)$ can still be calculated by Theorem 4.4.1. For example in Figure 4.7, we have

$$\begin{aligned}&\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}\right)\\ &= 1 - \mathbb{P}\left(M_{t_{i-1},t_{j_1}} < b_i, M_{t_{j_1},t_{j_2}} < b_i, M_{t_{j_2},t_i} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}\right),\end{aligned}$$

which can be solved by a nested integration formula, see Theorem 4.4.1, with the local homogeneous property for each sub-interval.

In terms of $\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right)$, we can also obtain follows by Theorem 4.4.1.

$$
\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) \\
&= 1 - \mathbb{P}\left(M_{t_{i-1},t_{j_1}} < b_i, M_{t_{j_1},t_{j_2}} < b_i, M_{t_{j_2},t_i} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) \\
&= 1 - \int_{\mathbb{R}} \mathbb{P}\left(Z_{t_{j_1}} \in \mathrm{d}x \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_i} = z_i\right) \\
&\times \mathbb{P}\left(M_{t_{i-1},t_{j_1}} < b_i, M_{t_{j_1},t_{j_2}} < b_i, M_{t_{j_2},t_i} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_{j_1}} = x, Z_{t_i} = z_i\right) \\
&= 1 - \int_{\mathbb{R}} \mathbb{P}\left(M_{t_{i-1},t_{j_1}} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_{j_1}} = x\right) \\
&\qquad \times \mathbb{P}\left(M_{t_{j_1},t_{j_2}} < b_i, M_{t_{j_2},t_i} < b_i \,|\, Z_{t_{j_1}} = x, Z_{t_i} = z_i\right) \\
&\qquad \times \frac{p(t_{j_1}, t_i, x, z_i)\, p(t_{i-1}, t_{j_1}, z_{i-1}, x)}{p(t_{i-1}, t_i, z_{i-1}, z_i)} \mathrm{d}x \\
&= 1 - \int_{\mathbb{R}} \mathbb{P}\left(M_{t_{i-1},t_{j_1}} < b_i \,|\, Z_{t_{i-1}} = z_{i-1}, Z_{t_{j_1}} = x\right) \\
&\qquad \int_{\mathbb{R}} \mathbb{P}\left(M_{t_{j_1},t_{j_2}} < b_i \,|\, Z_{t_{j_1}} = x, Z_{t_{j_2}} = y\right) \\
&\qquad\qquad \times \mathbb{P}\left(M_{t_{j_2},t_i} < b_i \,|\, Z_{t_{j_2}} = y, Z_{t_i} = z_i\right) \\
&\qquad\qquad \times \frac{p(t_{j_2}, t_i, y, z_i)\, p(t_{j_1}, t_{j_2}, x, y)\, p(t_{i-1}, t_{j_1}, z_{i-1}, x)}{p(t_{i-1}, t_i, z_{i-1}, z_i)} \mathrm{d}y\mathrm{d}x.
\end{aligned}
$$

Theorem 4.4.2 simplifies the nested integral in Theorem 4.4.1 to a product of single integrals, provided that some additional restrictions are satisfied. For non-matching time-discretisation, Theorem 4.4.2 can still be applied. However, the terms

$$\mathbb{P}\left(M_{t_{i-2},t_{i-1}} \geq b_{i-1} \,|\, Z_{t_{i-1}} = x_i, Z_0 = z_0\right)$$

and

$$\mathbb{P}\left(M_{t_{i-1},t_i} \geq b_i \,|\, Z_{t_{i-1}} = x_i\right)$$

can only be evaluated by the nested integral in Theorem 4.4.1.

- **Case 2: Non-matching time-discretisations for any of the functions $\mu(t)$, $\lambda(t)$, $\sigma(t)$ and $b(t)$.**

An example of this case is shown in Figure 4.8. This case can be reduced back to Case 1 by taking the union of all the time-discretisations steps as the an overall discretisations scheme. For example in Figure 4.8, we can consider it as a special case of Case 1 for the time steps $t_{i-1}, t_{j_1}, t_{j_2}, t_{j_3}, t_{j_4}, t_{j_5}$ and t_i.

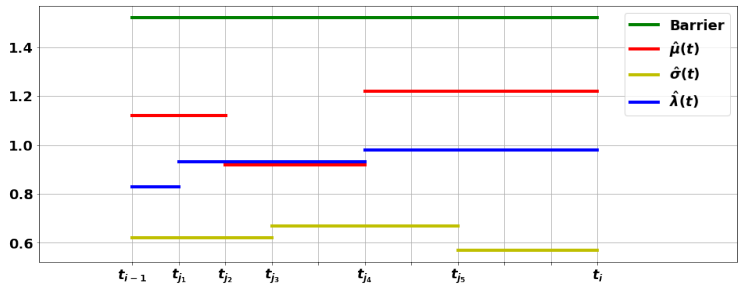

Figure 4.8: Non-matching time-discretisations for any of $\mu(t), \lambda(t), \sigma(t)$ and $b(t)$.

Remark 4.4.1 *When the discretisation is non-matching, we have two layers of nested integration:*

a) The nested integration due to non-matching time-discretisation;

b) The nested integration arising from the application of Theorem 4.4.1.

By Theorem 4.4.2, one can simplify the nested integral in b) to a product of single integrations under some restrictions. This can reduce the computational complexity. Although the nested integral in a) cannot be further reduced, in practice, if the variations of the piecewise constants within a single segment are much smaller than the variations of the piecewise constants among all the segments, one can use matching time-discretisation as an efficient approximation.

4.5 Computational methods and numerical results

For now, we have obtained decomposition formulae for both, the joint distribution and the survival function for the maxima of a continuous Markov process in consecutive intervals. For convenience, in this section we take the survival function for a standardised OU-process in consecutive intervals as an example to illustrate the computational methods. For simplicity, in this section, we consider the case that the lengths of all the time intervals are constant Δt, i.e. $t_i = i\Delta t$ for $i = 0, 1, 2, \ldots, N$.

Corollary 4.5.1 *Let* $(Z_t)_{t\geq 0}$ *be a standardised OU-process and* $M_{t_{i-1},t_i} = \sup_{t\in[t_{i-1},t_i)} Z_t$, *where* $t_i = i\Delta t$, *for* $i = 0,1,2,...,N$. *Then*

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} \geq b_1, \cdots, M_{t_{N-1},t_N} \geq b_N \,|\, Z_{t_0} = z_0\right) \\
&= \int_{\mathbb{R}} \kappa(z_0, z_1, b_1) \cdots \int_{\mathbb{R}} \kappa(z_{N-3}, z_{N-2}, b_{N-2}) \int_{\mathbb{R}} \kappa(z_{N-2}, z_{N-1}, b_{N-1}) \\
&\qquad \times \bar{q}(z_{N-1}, b_N)\, \mathrm{d}z_{N-1}\mathrm{d}z_{N-2}\cdots\mathrm{d}z_1, \qquad (4.5.1)
\end{aligned}$$

where

$$\begin{aligned}
\bar{q}(z_{N-1}, b_N) &= \left[1 - \sum_{k=1}^{\infty} c_k^{(N)} \mathrm{e}^{-\alpha_k^{(N)} t}\, \mathscr{H}_{\alpha_k^{(N)}}(-z_{N-1})\right] \mathbb{I}(z_{N-1} < b_N) \\
&\quad + \mathbb{I}(z_{N-1} \geq b_N), \\
\kappa(z_{i-1}, z_i, b_i) &= \sum_{k=1}^{\infty} c_k^{(i)} \alpha_k^{(i)}\, \mathscr{H}_{\alpha_k^{(i)}}(-z_{N-1}) \\
&\quad \times \int_{\mathrm{e}^{-\Delta t}}^{1} \frac{x^{-\alpha_k^{(i)}-1} \mathbb{I}(z_{i-1} < b_i)\, \mathbb{I}(z_i < b_i)}{\sqrt{\pi(1-x^2)}} \\
&\qquad \times \exp\left\{-\frac{(z_i - b_i x)^2}{1-x^2} - \alpha_k^{(i)} \Delta t\right\} \mathrm{d}x \\
&\quad + p(0, \Delta t, z_{i-1}, z_i)\left(1 - \mathbb{I}(z_{i-1} < b_i)\, \mathbb{I}(z_i < b_i)\right).
\end{aligned}$$

Here, $\mathscr{H}_\alpha(\cdot)$ *is the Hermite function with parameter* α, $\left\{\alpha_k^{(i)}\right\}$ *are the solutions to the equation* $\mathscr{H}_\alpha(-b_i) = 0$ *with* $c_k^{(i)} = -1/(\alpha_k^{(i)} \partial_{\alpha_k^{(i)}} \mathscr{H}_{\alpha_k^{(i)}}(-b_i))$.

Proof. By Theorem 4.4.1, we have

$$\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1} \geq b_1, \cdots, M_{t_{N-1},t_N} \geq b_N \,|\, Z_{t_0} = z_0\right) \\
&= \int_{\mathbb{R}} \mathbb{P}(M_{t_0,t_1} \geq b_1 \,|\, Z_{t_0} = z_0, Z_{t_1} = z_1)\, p(t_0, t_1, z_0, z_1) \cdots \\
&\quad \times \int_{\mathbb{R}} \mathbb{P}\left(M_{t_{N-2},t_{N-1}} \geq b_{N-1} \,|\, Z_{t_{N-2}} = z_{N-2}, Z_{t_{N-1}} = z_{N-1}\right) \\
&\quad \times p(t_{N-2}, t_{N-1}, z_{N-2}, z_{N-1}) \\
&\quad \times \left(1 - \mathbb{P}\left(M_{t_{N-1},t_N} < b_N \,|\, Z_{t_{N-1}} = z_{N-1}\right)\right) \mathrm{d}z_{N-1}\cdots\mathrm{d}z_1,
\end{aligned}$$

where $p(t_{i-1},t_i,z_{i-1},z_i)$ is the transition density function of the process $(Z_t)_{t\geq 0}$. By Theorem 4.2.1 we have

$$
\begin{aligned}
&1-\mathbb{P}\left(M_{t_{N-1},t_N}<b_N\,|\,Z_{t_{N-1}}=z_{N-1}\right)\\
&=\left[1-\sum_{k=1}^{\infty}c_k^{(N)}\mathrm{e}^{-\alpha_k^{(N)}\Delta t}\mathscr{H}_{\alpha_k^{(N)}}(-z_{N-1})\right]\mathbb{I}(z_{N-1}<b_N)+\mathbb{I}(z_{N-1}\geq b_N).
\end{aligned}
$$

By the homogeneous property of the standardised OU-process and Proposition 4.4.1, we have

$$
\begin{aligned}
&\mathbb{P}\left(M_{t_{i-1},t_i}\geq b_i\,|\,Z_{t_{i-1}}=z_{i-1},Z_{t_i}=z_i\right)p(t_{i-1},t_i,z_{i-1},z_i)\\
&=\mathbb{P}\left(M_{0,\Delta t}\geq b_i\,|\,Z_0=z_{i-1},Z_{\Delta t}=z_i\right)p(0,\Delta t,z_{i-1},z_i)\\
&=\int_0^{\Delta t}p(t,\Delta t,b_i,z_i)\mathbb{P}\left(\tau_{Z,b_i}\in\mathrm{d}t\,|\,Z_0=z_{i-1}\right)\mathbb{I}(z_{i-1}<b_i)\,\mathbb{I}(z_i<b_i)\\
&\quad+p(0,\Delta t,z_{i-1},z_i)\left(1-\mathbb{I}(z_{i-1}<b_i)\,\mathbb{I}(z_i<b_i)\right)\\
&=\sum_{k=1}^{\infty}c_k^{(i)}\alpha_k^{(i)}\mathscr{H}_{\alpha_k^{(i)}}(-z_{N-1})\int_{\mathrm{e}^{-\Delta t}}^{1}\frac{x^{-\alpha_k^{(i)}-1}\mathbb{I}(z_{i-1}<b_i)\,\mathbb{I}(z_i<b_i)}{\sqrt{\pi(1-x^2)}}\\
&\qquad\times\exp\left\{-\frac{(z_i-b_ix)^2}{1-x^2}-\alpha_k^{(i)}\Delta t\right\}\mathrm{d}x\\
&\quad+p(0,\Delta t,z_{i-1},z_i)\left[1-\mathbb{I}(z_{i-1}<b_i)\,\mathbb{I}(z_i<b_i)\right].
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathbb{P}\left(M_{t_0,t_1}\geq b_1,\cdots,M_{t_{N-1},t_N}\geq b_N\,|\,Z_{t_0}=z_0\right)\\
&=\int_{\mathbb{R}}\kappa(z_0,z_1,b_1)\cdots\int_{\mathbb{R}}\kappa(z_{N-3},z_{N-2},b_{N-2})\int_{\mathbb{R}}\kappa(z_{N-2},z_{N-1},b_{N-1})\\
&\qquad\times\bar{q}(z_{N-1},b_N)\,\mathrm{d}z_{N-1}\mathrm{d}z_{N-2}\cdots\mathrm{d}z_1,
\end{aligned}
$$

□

The iterated integral can be approximated efficiently by quadrature schemes or MC integration methods. We describe the two methods in what follows.

4.5.1 Quadrature schemes

We first present a quadrature scheme to evaluate

$$\begin{aligned} I &:= \mathbb{P}\left(M_{t_0,t_1} \geq b_1, \cdots, M_{t_{N-1},t_N} \geq b_N \,|\, Z_{t_0} = z_0\right) \\ &= \int_{\mathbb{R}} \kappa(z_0, z_1, b_1) \cdots \int_{\mathbb{R}} \kappa(z_{N-3}, z_{N-2}, b_{N-2}) \int_{\mathbb{R}} \kappa(z_{N-2}, z_{N-1}, b_{N-1}) \\ &\qquad \times \bar{q}(z_{N-1}, b_N)\, \mathrm{d}z_{N-1} \cdots \mathrm{d}z_1, \end{aligned} \tag{4.5.2}$$

Since the OU-process is defined on $\mathbb{R}$, we choose a sufficiently large number $Z_{\max}$ and sufficiently small number $Z_{\min}$. We partition the domain $[Z_{\min}, Z_{\max}]$ into L pieces of equal length δz, where the grid points are denoted $Z_{\min} = z^{(1)} < z^{(2)} < \cdots < z^{(L)} = Z_{\max}$. We can then approximate the integration with the following proposition. The nested integral in Equation (4.5.1) can be approximated by the product of matrices

$$I \approx \prod_{i=1}^{N-1} K_i \bar{Q} (\delta z)^{N-1},$$

where for $i = 1$

$$K_1 \;=\; \left[\kappa\left(z_0, z_1^{(1)}\right), \kappa\left(z_0, z_1^{(2)}\right), \cdots, \kappa\left(z_0, z_1^{(L)}\right)\right],$$

for $i = 2, 3, \cdots, N-1$,

$$\bar{Q} \;=\; \left[\bar{q}\left(z_{N-1}^{(1)}, b_N\right), \bar{q}\left(z_{N-1}^{(2)}, b_N\right), \cdots, \bar{q}\left(z_{N-1}^{(L)}, b_N\right)\right]^{\top},$$

and

$$K_i = \begin{bmatrix} \kappa\left(z_{i-1}^{(1)}, z_i^{(1)}, b_i\right) & \kappa\left(z_{i-1}^{(1)}, z_i^{(2)}, b_i\right) & \cdots & \kappa\left(z_{i-1}^{(1)}, z_i^{(L)}, b_i\right) \\ \kappa\left(z_{i-1}^{(2)}, z_i^{(1)}, b_i\right) & \kappa\left(z_{i-1}^{(2)}, z_i^{(2)}, b_i\right) & \cdots & \kappa\left(z_{i-1}^{(2)}, z_i^{(L)}, b_i\right) \\ \vdots & \vdots & & \vdots \\ \kappa\left(z_{i-1}^{(L)}, z_i^{(1)}, b_i\right) & \kappa\left(z_{i-1}^{(L)}, z_i^{(2)}, b_i\right) & \cdots & \kappa\left(z_{i-1}^{(L)}, z_i^{(L)}, b_i\right) \end{bmatrix}.$$

4.5.2 MC integration method

The integral (4.5.2) can also be evaluated efficiently by an importance sampling approximation. Assume Z_1, Z_2, ..., Z_{N-1} are independent and identical random variables with density function $p : \mathbb{R} \to \mathbb{R}^+$, which gives

$$\varphi\left(z, z', u\right) := \kappa\left(z, z', u\right) / p(z')$$

such that

$$\int_{\mathbb{R}} \varphi\left(z, z', u\right) \mathrm{d}z' < \infty.$$

Let $Z_{i-1}^{(k_i)}$ be the k_i-th random number in the sample generated from the random variable Z_{i-1}, and let L_i be the sample size of the random variable Z_{i-1}. Then the nested integral (4.5.1) in Proposition 4.5.1 can be approximated by the product of matrices

$$I \approx (\prod_{i=1}^{N-1} \Omega_i / L_i) \bar{\mathscr{Q}},$$

where for $i = 1$,

$$\Omega_1 = \left[\varphi\left(z_0, Z_1^{(1)}\right), \varphi\left(z_0, Z_1^{(2)}\right), \cdots, \varphi\left(z_0, Z_1^{(K_1)}\right)\right],$$

for $i = 2, 3, \cdots, N-1$,

$$\bar{\mathscr{Q}} = \left[\bar{q}\left(Z_{N-1}^{(1)}, b_N\right), \bar{q}\left(Z_{N-1}^{(2)}, b_N\right), \cdots, \bar{q}\left(Z_{N-1}^{(K_{N-1})}, b_N\right)\right]^\top.$$

and

$$\Omega_i = \begin{bmatrix} \varphi\left(Z_{i-1}^{(1)}, Z_i^{(1)}, b_i\right) & \varphi\left(Z_{i-1}^{(1)}, Z_i^{(2)}, b_i\right) & \cdots & \varphi\left(Z_{i-1}^{(1)}, Z_i^{(K_i)}, b_i\right) \\ \varphi\left(Z_{i-1}^{(2)}, Z_i^{(1)}, b_i\right) & \varphi\left(Z_{i-1}^{(2)}, Z_i^{(2)}, b_i\right) & \cdots & \varphi\left(Z_{i-1}^{(2)}, Z_i^{(K_i)}, b_i\right) \\ \cdots & \cdots & \cdots & \cdots \\ \varphi\left(Z_{i-1}^{(K_{i-1})}, Z_i^{(1)}, b_i\right) & \varphi\left(Z_{i-1}^{(K_{i-1})}, Z_i^{(2)}, b_i\right) & \cdots & \varphi\left(Z_{i-1}^{(K_{i-1})}, Z_i^{(K_i)}, b_i\right) \end{bmatrix},$$

4.5.3 Numerical tests

4.5.3.1 Benchmark: direct MC

The two methods can be compared with the direct MC approach, which can be shown in Algorithm 1.

Algorithm 1 Direct MC Algorithm

while path $n \leq N$ **do**
    while time step $m \leq M$ **do**
        1, simulate realisation of standard normal random variale $\phi_m^{(n)}$
        2, evaluate $x_{m+1}^{(n)} = x_m^{(n)} \mathrm{e}^{-\lambda\delta} + \mu(1-\mathrm{e}^{-\lambda\delta}) + \sigma\sqrt{\frac{1-\mathrm{e}^{-2\lambda\delta}}{2\lambda}}\phi_m^{(n)}$

    if $\max(x_1^{(n)}, \cdots, x_{M/2}^{(n)}) \geq b_1$ **and** $\max(x_{M/2+1}^{(n)}, \cdots, x_M^{(n)}) \geq b_2$ **then**
        $\mathbb{I}^{(n)} = 1$
    else
        $\mathbb{I}^{(n)} = 0$
$Final\_Prob = \mathrm{Mean}(Ind)$
$Final\_Err = \mathrm{StD}(Ind)/\sqrt{N}$

Here, we denote

NB:	Number of sets.
N:	Number of paths for a given simulation set.
M:	Number of time steps per path.
b_1, b_2:	barrier level at interval one and two, consecutively.
n:	n-th path.
m:	m-th time step.
$\phi_m^{(n)}$:	realisation from standard normal random variable for path n at time step m
$\delta = T/M$:	time step length.
$x_m^{(n)}$:	the realised OU-process value of path n at time step m.
$\mathbb{I}^{(n)}$:	the indicator.
$Prob_{(nb)}$:	the probability of joint crossing in two consecutive intervals for set nb.

Alternatively, we can implement the direct MC method with a small memory version Algorithm 2.

Algorithm 2 Direct MC Algorithm (low memory requirement)

while set $nb \leq NB$ **do**
  while path $n \leq N$ **do**
    while time step $m \leq M$ **do**
      1, simulate realisation of standard normal random variale $\phi_m^{(n)}$
      2, evaluate $x_{m+1}^{(n)} = x_m^{(n)} \mathrm{e}^{-\lambda\delta} + \mu(1-\mathrm{e}^{-\lambda\delta}) + \sigma\sqrt{\frac{1-\mathrm{e}^{-2\lambda\delta}}{2\lambda}}\phi_m^{(n)}$

    if $\max(x_1^{(n)}, \cdots, x_{M/2}^{(n)}) \geq b_1$ **and** $\max(x_{M/2+1}^{(n)}, \cdots, x_M^{(n)}) \geq b_2$ **then**
      $\mathbb{I}^{(n)} = 1$
    else
      $\mathbb{I}^{(n)} = 0$
  $Prob_{(nb)} = \mathrm{Mean}(Ind)$
$Final\_Prob = \mathrm{Mean}(Prob)$
$Final\_Err = \mathrm{StD}(Prob)/\sqrt{NB}$

In this part, we consider the probability $\mathbb{P}\left(\sup_{t\in(t_0,t_1]} X_t \geq b_1, \sup_{t\in(t_1,t_2]} X_t \geq b_2\right)$ as an example. We compute it by the following algorithm:

a) We discretise the time interval $[t_0, t_2]$ into M pieces:

$$t_0 = t^{(0)} < t^{(1)} < \cdots < t^{(\frac{M}{2})} < t_1 = t^{(\frac{M}{2}+1)} < \cdots < t^{(M)} = t_2.$$

b) We estimate the maximum in each interval by

$$\sup_{t\in(t_0,t_1]} X_t \approx \max\left\{X_{t^{(0)}}, X_{t^{(1)}}, \cdots, X_{t^{(\frac{M}{2})}}\right\},$$
$$\sup_{t\in(t_1,t_2]} X_t \approx \max\left\{X_{t^{(\frac{M}{2}+1)}}, X_{t^{(\frac{M}{2}+2)}}, \cdots, X_{t^{(M)}}\right\}.$$

c) We approximate $\mathbb{P}\left(\sup_{t\in(t_0,t_1]} X_t \geq b_1, \sup_{t\in(t_1,t_2]} X_t \geq b_2\right)$ with

$$\mathbb{P}\left(\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M}{2})}}\right\} \geq b_1, \right.$$
$$\left. \max\left\{X_{t^{(\frac{M}{2}+1)}}, X_{t^{(\frac{M}{2}+2)}}, \cdots, X_{t^{(M)}}\right\} \geq b_2\right).$$

Lemma 4.5.1 *The direct MC algorithm (a) - (c) underestimates the real probability due to the time discretisation, that is*

$$\begin{aligned}&\mathbb{P}\left(\sup_{t\in(t_0,t_1]} X_t \geq b_1,\ \sup_{t\in(t_1,t_2]} X_t \geq b_2\right)\\&\geq \mathbb{P}\left(\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M}{2})}}\right\} \geq b_1, \max\left\{X_{t^{(\frac{M}{2}+1)}}, X_{t^{(\frac{M}{2}+2)}}, \cdots, X_{t^{(M)}}\right\} \geq b_2\right).\end{aligned}$$

Proof. Since the probability of a sub-event is smaller than that of the event itself, i.e.

$$\begin{aligned}&\left\{\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M}{2})}}\right\} \geq b_1, \max\left\{X_{t^{(\frac{M}{2}+1)}}, X_{t^{(\frac{M}{2}+2)}}, \cdots, X_{t^{(M)}}\right\} \geq b_2\right\}\\&\subseteq \left\{\sup_{t\in(t_0,t_1]} X_t \geq b_1,\ \sup_{t\in(t_1,t_2]} X_t \geq b_2\right\},\end{aligned}$$

we have

$$\begin{aligned}&\mathbb{P}\left(\sup_{t\in(t_0,t_1]} X_t \geq b_1,\ \sup_{t\in(t_1,t_2]} X_t \geq b_2\right)\\&\geq \mathbb{P}\left(\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M}{2})}}\right\} \geq b_1, \max\left\{X_{t^{(\frac{M}{2}+1)}}, X_{t^{(\frac{M}{2}+2)}}, \cdots, X_{t^{(M)}}\right\} \geq b_2\right).\end{aligned}$$

□

Although a higher number of time steps in the discretization can reduce the underestimation bias in Lemma 4.5.1 for the direct Monte Carlo estimator, Lemma 4.5.2 shows that it may lead to higher Monte Carlo error.

Lemma 4.5.2 *For a fixed number of paths, the errors of the direct MC algorithm (a) - (c), based on a discretisation with M_1 and M_2 time steps ($M_1 < M_2$), satisfy the relations*

$$\begin{aligned}&Err_{MC(M_1)} \geq Err_{MC(M_2)} \quad \textit{if and only if} \quad prob_{MC(M_1)} + prob_{MC(M_2)} \geq 1,\\&Err_{MC(M_1)} \leq Err_{MC(M_2)} \quad \textit{if and only if} \quad prob_{MC(M_1)} + prob_{MC(M_2)} \leq 1.\end{aligned}$$

Proof. For convenience, we denote

$$\left\{\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M_1}{2})}}\right\} \geq b_1, \max\left\{X_{t^{(\frac{M_1}{2}+1)}}, X_{t^{(\frac{M_2}{2}+2)}}, \cdots, X_{t^{(M_1)}}\right\} \geq b_2\right\}$$

by A and

$$\left\{\max\left\{X_{t^{(1)}}, X_{t^{(2)}}, \cdots, X_{t^{(\frac{M_2}{2})}}\right\} \geq b_1, \max\left\{X_{t^{(\frac{M_2}{2}+1)}}, X_{t^{(\frac{M_2}{2}+2)}}, \cdots, X_{t^{(M_2)}}\right\} \geq b_2\right\}$$

by B. Then the MC algorithm (a) - (c) computes $\text{prob}_{\text{MC}(M_1)} = \mathbb{P}(A) = \mathbb{E}[\mathbb{I}_A]$ for M_1 time steps, and $\text{prob}_{\text{MC}(M_2)} = \mathbb{P}(B) = \mathbb{E}[\mathbb{I}_B]$ for M_2 time steps. If we implement the MC algorithm (a) - (c) to compute $\mathbb{E}[\mathbb{I}_A]$ and $\mathbb{E}[\mathbb{I}_B]$, the ratio between the resulting errors is equal to the ratio between the standard deviation of $\mathbb{I}_A$ and $\mathbb{I}_B$. That is:

$$\frac{Err(\mathbb{I}_A)}{Err(\mathbb{I}_B)} = \frac{StD(\mathbb{I}_A)}{StD(\mathbb{I}_B)} = \sqrt{\frac{Var(\mathbb{I}_A)}{Var(\mathbb{I}_B)}} = \sqrt{\frac{\mathbb{E}\left[(\mathbb{I}_A)^2\right] - \mathbb{E}[\mathbb{I}_A]^2}{\mathbb{E}\left[(\mathbb{I}_B)^2\right] - \mathbb{E}[\mathbb{I}_B]^2}}.$$

We observe that $\mathbb{I}_A$ and $\mathbb{I}_B$, we have $(\mathbb{I}_A)^2 = \mathbb{I}_A$ and $(\mathbb{I}_B)^2 = \mathbb{I}_B$. Therefore,

$$\frac{Err(\mathbb{I}_A)}{Err(\mathbb{I}_B)} = \sqrt{\frac{\mathbb{E}\left[(\mathbb{I}_A)^2\right] - \mathbb{E}[\mathbb{I}_A]^2}{\mathbb{E}\left[(\mathbb{I}_B)^2\right] - \mathbb{E}[\mathbb{I}_B]^2}} = \sqrt{\frac{\mathbb{E}[\mathbb{I}_A] - \mathbb{E}[\mathbb{I}_A]^2}{\mathbb{E}[\mathbb{I}_B] - \mathbb{E}[\mathbb{I}_B]^2}} = \sqrt{\frac{\mathbb{P}(A)\mathbb{P}(\bar{A})}{\mathbb{P}(B)\mathbb{P}(\bar{B})}}.$$

This shows that

$$Err_{\text{MC}(M_1)} = \sqrt{\frac{\text{prob}_{\text{MC}(M_1)}}{\text{prob}_{\text{MC}(M_2)}} \frac{1 - \text{prob}_{\text{MC}(M_1)}}{1 - \text{prob}_{\text{MC}(M_2)}}}\, Err_{\text{MC}(M_2)}.$$

Since $M_1 < M_2$, we have $\text{prob}_{\text{MC}(M_1)} \leq \text{prob}_{\text{MC}(M_2)}$. We denote $\text{prob}_{\text{MC}(M_2)}$ by $p \in \mathbb{R}^+$, then

$$\text{prob}_{\text{MC}(M_1)} = p - a$$

for some $a \in [0, p]$. Therefore,

$$\begin{aligned} Err_{\mathrm{MC}(M_1)} &= \sqrt{\frac{p-a}{p} \cdot \frac{1-(p-a)}{1-p}}\, Err_{\mathrm{MC}(M_2)} \\ &= \sqrt{1 + \frac{a}{p(1-p)} \left(\mathrm{prob}_{\mathrm{MC}(M_1)} + \mathrm{prob}_{\mathrm{MC}(M_2)} - 1 \right)}\, Err_{\mathrm{MC}(M_2)}. \end{aligned}$$

Since $a \geq 0$ and $p \in [0, 1]$,

$$Err_{\mathrm{MC}(M_1)} \geq Err_{\mathrm{MC}(M_2)} \quad \text{if and only if} \quad \mathrm{prob}_{\mathrm{MC}(M_1)} + \mathrm{prob}_{\mathrm{MC}(M_2)} \geq 1,$$

$$Err_{\mathrm{MC}(M_1)} \leq Err_{\mathrm{MC}(M_2)} \quad \text{if and only if} \quad \mathrm{prob}_{\mathrm{MC}(M_1)} + \mathrm{prob}_{\mathrm{MC}(M_2)} \leq 1.$$

□

Therefore, as shown in Lemma 4.5.1 and 4.5.2, direct Monte Carlo is not ideal for passage-time approximations for two reasons: 1) it is a biased estimator which underestimate the true probability; 2) adding the number of time steps in discretization can cause the increase of Monte Carlo error.

4.5.3.2 Results

Since the direct MC method needs to be implemented by a time discretisation, this method underestimates the real passage-time probability, see Lemma 4.5.1. We observe that when the number of time steps increase, the results obtained by direct MC method become closer to the results obtained by quadrature scheme and the MC integration. However, direct MC results become increasingly noisy when the joint passage event becomes rarer, while the quadrature scheme and MC integration methods remain stable. It shows the quadrature and MC integration methods can improve the accuracy if the joint passage is a rare event. Another interesting result is that although the direct MC result will be more accurate for larger number of time steps, its MC error will be bigger as well, provided that the event occurrence is a rare event, see Lemma 4.5.2. It means the direct MC method is not suitable for the passage-time approximation. In the comparison of the three methods, i.e. the direct MC, quadrature and MC integration, we test the following two cases: 1)

We fix the number of consecutive intervals and change the level of barriers in each interval; 2) We fix the level of barrier and increase the number of intervals. In both cases, the probability we want to approximate becomes small when the barriers rise or the number of intervals increase. The direct MC estimator will become noisy when the joint event becomes rare, due to Lemma 4.5.1 and 4.5.2. In this subsection, we want to show that with the quadrature scheme and MC integration estimators, the approximation can still be accurate and robust and the efficiency can also be smaller compared with the direct MC methods. Both the quadrature scheme and the MC integration scheme contain two types of error source: 1) The truncation error from the approximation of the FPT density infinite series; 2) The deterministic or stochastic error from the numerical integration. As we have shown in Section 4.2, the truncation error in 1) can be reduced to a small level by introducing few truncation terms. The numerical error 2) depends on its discretisation size in the quadrature scheme while depends on the number of MC paths in the MC integration scheme. This type of error can be reduced by introducing more time-discretisations or more MC samples.

b_1	b_2	MC (500)	MC (1000)	MC (2000)	Quad.	MC int.
1	1	1.417×10^{-1}	1.448×10^{-1}	1.469×10^{-1}	1.517×10^{-1}	1.515×10^{-1}
		(2×10^{-4})	(2×10^{-4})	(2×10^{-4})	(3×10^{-4})	(6×10^{-4})
1	2	1.27×10^{-2}	1.311×10^{-2}	1.352×10^{-2}	1.426×10^{-2}	1.440×10^{-2}
		(1×10^{-4})	(8×10^{-5})	(8×10^{-5})	(2×10^{-5})	(7×10^{-5})
2	1	5.08×10^{-3}	5.38×10^{-3}	5.54×10^{-3}	5.837×10^{-3}	5.843×10^{-3}
		(7×10^{-5})	(5×10^{-5})	(6×10^{-5})	(2×10^{-5})	(3×10^{-5})
2	2	2.35×10^{-3}	2.50×10^{-3}	2.62×10^{-3}	2.72×10^{-3}	2.74×10^{-3}
		(4×10^{-5})	(4×10^{-5})	(4×10^{-5})	(2×10^{-5})	(3×10^{-5})
2	3	4.1×10^{-5}	5.0×10^{-5}	5.4×10^{-5}	5.08×10^{-5}	5.10×10^{-5}
		(4×10^{-6})	(5×10^{-6})	(4×10^{-6})	(2×10^{-7})	(3×10^{-7})
3	2	1.1×10^{-5}	1.1×10^{-5}	1.2×10^{-5}	1.455×10^{-5}	1.458×10^{-5}
		(3×10^{-6})	(3×10^{-6})	(4×10^{-6})	(1×10^{-7})	(9×10^{-8})
3	3	7×10^{-6}	6×10^{-6}	6×10^{-6}	5.47×10^{-6}	5.42×10^{-6}
		(2×10^{-6})	(2×10^{-6})	(2×10^{-6})	(9×10^{-8})	(5×10^{-8})

Table 4.1: The probability of the maxima for a standardised OU-process in $[0,1)$ and $[1,2]$ are above b_1 and b_2. The number below in the bracket is the absolute MC or quadrature error. The three sets of direct MC results are implemented with $2,000,000$ sample paths. The number of time steps for the three sets of direct MC results are 500, $1,000$ and $2,000$, respectively. The quadracture scheme is implemented between the state domain $[-5,5]$ with state increment 0.005. The MC integration method is implemented with $100,000$ sample paths.

The first numerical example is to compute the probability of the maxima for a standardised OU-process in $[0,1)$ and $[1,2]$ above b_1 and b_2, whose values vary in the left of Table 4.1. In this example, we choose the number of paths and discretisation numbers to fix the error order of magnitude as 10^{-4} for the case $b_1 = 1$ and $b_2 = 1$. It gives the following results:

We can also observe from Table 4.1 that in the direct MC cases, the error cannot be improved by introducing more time-discretisation due to Lemma 4.5.2. We also compare the time consumption to obtain the results in Table 4.1. In Figure 4.9, we can observe that the quadrature and MC integration methods are more efficient than the direct MC methods. In fact, the quadrature scheme can be implemented in real-time.

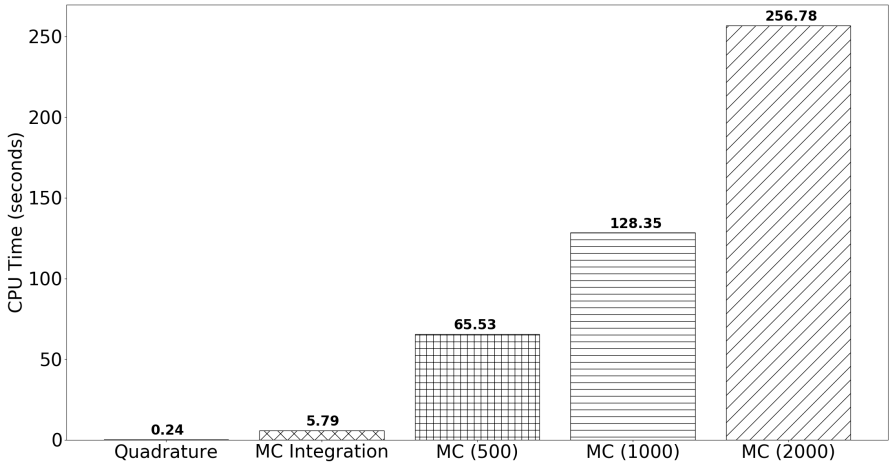

Figure 4.9: With given absolute error at $2 \sim 4 \times 10^{-5}$ for all schemes, the CPU time consumption for the case that $b_1 = b_2 = 2$.

Our second example is to fix the barrier and increase the number of intervals, which will lead to smaller joint passage probability. We can observe in Table 4.2 that the numerical results obtained by the direct MC methods tends to be less accuracy when the number of intervals increases. However, the quadrature and MC integration methods remain reliable compared with the noisy direct MC results.

N	MC (500)	MC (1000)	MC (2000)	Quadrature	MC integration
2	2.35×10^{-3}	2.50×10^{-3}	2.62×10^{-3}	2.72×10^{-3}	2.74×10^{-3}
	(5×10^{-5})	(4×10^{-5})	(4×10^{-5})	(2×10^{-5})	(3×10^{-5})
3	3.1×10^{-4}	3.3×10^{-4}	3.3×10^{-4}	3.23×10^{-4}	3.29×10^{-4}
	(1×10^{-5})	(1×10^{-5})	(1×10^{-5})	(1×10^{-6})	(5×10^{-6})
4	5.5×10^{-5}	5.7×10^{-5}	5.6×10^{-5}	5.23×10^{-5}	5.30×10^{-5}
	(4×10^{-6})	(7×10^{-6})	(8×10^{-6})	(3×10^{-7})	(8×10^{-7})
5	8×10^{-6}	7×10^{-6}	8×10^{-6}	8.06×10^{-6}	8.07×10^{-6}
	(2×10^{-6})	(2×10^{-6})	(2×10^{-6})	(6×10^{-8})	(9×10^{-8})

Table 4.2: The probability of the maxima of a standardised OU-process in N consecutive intervals are all above $b = 2$. The number below is the absolute MC or quadrature error. The three sets of direct MC results are implemented with $2,000,000$ sample paths. The number of time steps for the three sets of direct MC results are 500, $1,000$ and $2,000$, respectively. The quadracture scheme is implemented between the state domain $[-5,5]$ with state increment 0.005. The MC integration method is implemented with $100,000$ sample paths.

We can observe from Figure 4.10 that the time consumption by direct MC method increases linearly with respect to the number of intervals. On the other hand, there is a small jump in the time consumption for the quadrature scheme and the MC integration method. This is because when the number of intervals is two, we do not need matrix K_i and Ω_i in Section 4.5.1 and 4.5.2. Once matrix K_i and Ω_i are obtained, they will be saved for future computation. It means that the quadrature scheme and the MC integration method remain similar after three intervals. The probability for large number of intervals can be evaluated much more efficiently by the quadrature and MC integration methods.

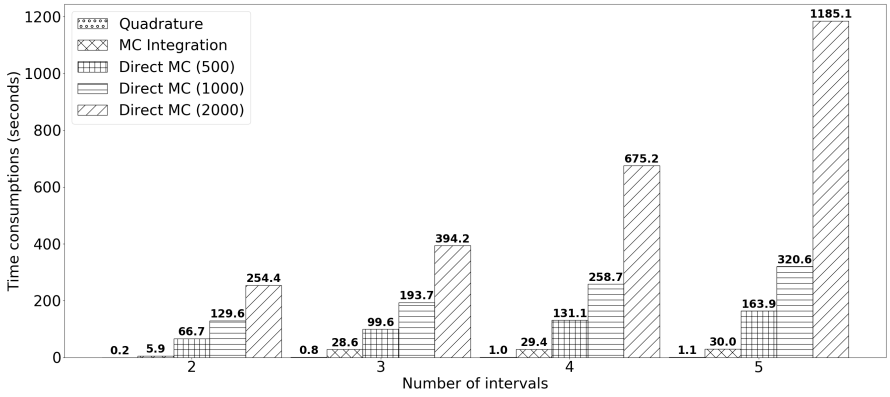

Figure 4.10: Time consumption of different schemes for a different number of intervals in Table 4.2.

4.6 Conclusions

In this chapter, we considered the multiple barrier-crossing problem of an OU-process in consecutive periods. To analyse this problem, we first presented the formulae for the FPT survival function, which coincided with the formulae given in [3] and [58]. Furthermore, we also showed that the FPT distribution is not heavy-tailed. Afterwards, we showed the transformation formulae of the FPTs between an inhomogeneous OU-process to a time-dependent barrier and a standardised OU-process to another time-dependent barrier. We investigated the running maxima of an inhomogeneous OU-process with piecewise constant parameters in consecutive intervals in detail. We presented a nested integration formulae for the joint distribution and survival functions for the maxima in consecutive intervals of an inhomogeneous OU-process with piecewise constant parameters. By matrix multiplications, the nested integral can be solved efficiently by quadrature or MC integration method. We also showed that if we put further constraints on the problem, the event that the maxima in each interval are above the barrier can be simplified to the product of single integrals.

Chapter 5

Overall conclusions

This thesis treated two broad classes of problems. The first problem was the known existence of repeated and costly computations in the traditional approach to sensitivity computations. The second problem arose due to the difficulty that exists in approximating barrier-crossing probabilities by using the traditional approach.

In order to overcome the first problem, we proposed the use of the AAD method. The AAD method computes partial derivatives of a certain function by considering linear combinations of its Jacobian matrices. By doing so, one can obtain sensitivities for all the input variables but at a bounded computational cost. In modern computational finance, it is noted that the use of AAD can accelerate computational speed in risk management exercises and can furthermore save computational costs. This becomes increasingly important in the context of XVA computations, which themselves have been known to be complex enough to carry out. We noted, however, that as a pathwise method AAD cannot be directly applied to non-Lipschitz continuous functions. We therefore developed two smoothing methods in order to overcome this limitation. With these methods, the sensitivities for functions with non-Lipschitz continuous components can be computed accurately and robustly.

We also showed that with the help of mathematical analysis, the second problem can be overcome in an elegant way which has a low approximation error. This method was found to be more superior when compared to direct Monte Carlo methods. By using the partial differential equation characterisation as well as the

Markov property, we obtained a semi-analytical formula for the multiple barrier-crossing probability of an Ornstein-Uhlenbeck diffusion. Moreover, by invoking either quadrature or importance-sampling methods, we noted that one can compute the semi-analytical formula numerically in an efficient manner. These approaches also outperform the traditional Monte Carlo approach in terms of stability and efficiency.

In sum, we emphasise that against the backdrops of the rapid growth in computational power and the increase in volume of cutting-edge technologies available, mathematical and statistical analyses will always take priority in the context of financial problem solving. Moreover, new technologies are not to be viewed as competitors but rather as complements to traditional mathematical and statistical models. With the aid of advanced technologies, models and methods that used to be difficult to implement can now be reconsidered in financial problem-solving contexts, as we demonstrated in this thesis.

Appendix A

Stability analysis for finite difference methods

We consider the following diffusion equation:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial t^2}, \quad a > 0. \tag{A.0.1}$$

The ϕ-method shown in Chapter 2 can be written as

$$\frac{u_j^{m+1} - u_j^m}{\Delta t} = a\left[\phi\frac{u_{j+1}^{m+1} - 2u_j^{m+1} + u_{j-1}^{m+1}}{\Delta x^2} + (1-\phi)\frac{u_{j+1}^m - 2u_j^m + u_{j-1}^m}{\Delta x^2}\right]. \tag{A.0.2}$$

Here, $\phi = 0$ corresponds to the explicit method; $\phi = 1$ corresponds to the implicit method; and $\phi = 0.5$ corresponds to the Crank-Nicolson method. Based on Section V of [26], we analyse the Von Neumann stability of the finite difference method for PDEs. We assume a numerical scheme admits a solution of the form

$$u_j^m = a^{(n)}(\omega)\mathrm{e}^{i\cdot j\omega\Delta x}, \tag{A.0.3}$$

where ω is the wave number and $i = \sqrt{-1}$. The Von Neumann stability condition for an initial value problem is given by

$$\left|\frac{a^{(n+1)}(\omega)}{a^{(n)}(\omega)}\right| \leq 1 \tag{A.0.4}$$

and the Von Neumann stability condition for terminal value problem is given by

$$\left| \frac{a^{(n)}(\omega)}{a^{(n+1)}(\omega)} \right| \leq 1, \tag{A.0.5}$$

for $0 \leq \omega \Delta x \leq \pi$. It can be shown that the explicit scheme is stable if and only if

$$\rho := a\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2},$$

which is known as conditionally stable. The implicit and Crank–Nicolson schemes are stable for any values of ρ, known as unconditionally stable, see page 95 of [70]. However, it is the growth factor that governs the propagation of the finite difference solution from one time step to the next. As shown in [63], the growth factor for (i) the explicit method is

$$g(\varphi) = 1 - 4\rho \sin^2 \frac{\varphi}{2},$$

for (ii) the implicit method it is

$$g(\varphi) = \frac{1}{1 + 4\rho \sin^2 \frac{\varphi}{2}},$$

and for (iii) the Crank-Nicolson method it is

$$\frac{1 - 2\rho \sin^2 \frac{\varphi}{2}}{1 + 2\rho \sin^2 \frac{\varphi}{2}},$$

for $-\pi \leq \varphi \leq \pi$, which is the parameter in the frequency domain. The φ closes to 0 corresponds to slowly varying components and the φ close to π corresponds to highly oscillatory components of the solution. The latter is the case when there are discontinuities in the initial conditions, such as the forward PDE in Chapter 2 with the Dirac Delta function initial condition. We have stability when $|g(\varphi)| < 1$. We can see that for the explicit and implicit methods, we have the same results for their stability. However, we notice that when $|\varphi| \to \pi$, we have $g(\varphi) \to -1$ for large ρ. This means that the oscillatory components are propagated as weakly damped oscillations in time. Here, we refer to [63] for detailed methods to reduce

the oscillations in the Crank-Nicolson scheme. One sufficient condition is to control ρ such that

$$\frac{1}{4N+2} \leq \rho \leq N + \frac{1}{2}$$

for $N \in \mathscr{N}$. With such ρ, we have

$$|g(\varphi)| \leq \frac{N}{N+1}$$

even for $|\varphi| = \pi$. The larger N we use, the more oscillations the solution is exposed to.

Appendix B

Special functions and their corresponding properties used

B.1 Confluent hypergeometric function

The confluent hypergeometric function is given by

$$ {}_1F_1(a;b;x) = \sum_{n=0}^{\infty} \frac{a^{(n)} x^n}{b^{(n)} n!} \tag{B.1.1} $$

where

$$ \begin{aligned} a^{(0)} &= 1, \\ a^{(n)} &= a(a+1)(a+2)\cdots(a+n-1)\,. \end{aligned} $$

We have the asymptotic relationship

$$ {}_1F_1\,(a;b;x) \sim \Gamma(b)\mathrm{e}^x x^{a-b}/\Gamma(a) $$

for $x \to \infty$.

B.2 Hermite function

The Hermite function is given by

$$\mathscr{H}_\alpha(x) := 2^\alpha \sqrt{\pi} \left[\frac{{}_1F_1\left(-\frac{\alpha}{2}; \frac{1}{2}; x^2\right)}{\Gamma\left(\frac{1-\alpha}{2}\right)} + 2x \frac{{}_1F_1\left(\frac{1-\alpha}{2}; \frac{3}{2}; x^2\right)}{\Gamma\left(-\frac{\alpha}{2}\right)} \right],$$

where $\Gamma(\cdot)$ is the Gamma function

$$\Gamma(x) = \int_0^\infty y^{x-1} \mathrm{e}^{-y}\, dy$$

and ${}_1F_1(x;y;x)$ is the confluent hypergeometric function in Equation B.1.1. When $\alpha \to \infty$, we have

$$\begin{aligned} \mathscr{H}_\alpha(x) =& 2^{\alpha+1/2} \mathrm{e}^{x^2-\alpha/2-1/4} \left(\frac{\alpha}{2} + \frac{1}{4}\right)^{\alpha/2} \\ & \times \cos\left(2x\sqrt{\frac{\alpha}{2} + \frac{1}{4}} + \frac{\alpha\pi}{2}\right) \left[1 + O\left(\frac{1}{\sqrt{\alpha/2+1/4}}\right)\right]. \end{aligned}$$

For large k, the α-zeros of $\mathscr{H}_\alpha(b) = 0$ for a given b is

$$\alpha_k = 2k + 1 + \frac{4b^2}{\pi^2} - \frac{2b}{\pi}\sqrt{4k + 3 + \frac{4b^2}{\pi^2}}.$$

Bibliography

[1] Abramowitz, M. and Stegun, I.A., 1965. Handbook of mathematical functions with formulas, graphs, and mathematical table. US Department of Commerce. National Bureau of Standards Applied Mathematics, Series 55.

[2] Alaton, P., Djehiche, B. and Stillberger, D., 2002. On modelling and pricing weather derivatives. Applied Mathematical Finance, 9(1), pp.1-20.

[3] Alili, L., Patie, P. and Pedersen, J.L., 2005. Representations of the first hitting time density of an Ornstein-Uhlenbeck process. Stochastic Models, 21(4), pp.967-980.

[4] Andersen, L. and Broadie, M., 2004. Primal-dual simulation algorithm for pricing multidimensional American options. Management Science, 50(9), pp.1222-1234.

[5] Anderson, L. and Piterbarg, V., 2010. Interest Rate Modeling (Volume 1): Foundations and Vanilla Models. Atlantic Financial Press, London.

[6] Basel Committee on Banking Supervision, 2017. Basel III: Finalising post-crisis reforms.URL: `https://www.bis.org/bcbs/publ/d424.pdf`

[7] Bellman, R., 1952. On the theory of dynamic programming. Proceedings of the National Academy of Sciences, 38(8), pp.716-719.

[8] Bingham, N.H. and Kiesel, R., 2013. Risk-neutral Valuation: Pricing and Hedging of Financial Derivatives. Springer, Berlin.

[9] Black, F. and Karasinski, P., 1991. Bond and option pricing when short rates are lognormal. Financial Analysts Journal, 47(4), pp.52-59.

[10] Brigo, D., Morini, M. and Pallavicini, A., 2013. Counterparty Credit Risk, Collateral and Funding: with Pricing Cases for All Asset Classes. John Wiley & Sons, New York.

[11] Brigo, D., Capponi, A., Pallavicini, A. and Papatheodorou, V., 2013. Pricing counterparty risk including collateralization, netting rules, re-hypothecation and wrong-way risk. International Journal of Theoretical and Applied Finance, 16(02), pp.1350007.

[12] Broadie, M. and Glasserman, P., 1996. Estimating security price derivatives using simulation. Management Science, 42(2), pp.269-285.

[13] Broadie, M. and Glasserman, P., 1997. Pricing American-style securities using simulation. Journal of Economic Dynamics and Control, 21(8-9), pp.1323-1352.

[14] Broadie, M. and Glasserman, P., 2004. A stochastic mesh method for pricing high-dimensional American options. Journal of Computational Finance, 7, pp.35-72.

[15] Buonocore, A., Nobile, A. G. and Ricciardi, L. M., 1987. A new integral equation for the evaluation of first-passage-time probability densities. Advances in Applied Probability, 19(4), pp.784-800.

[16] Capriotti, L. and Giles, M., 2010. Fast correlation Greeks by adjoint algorithmic differentiation. Risk, 23, pp.79-85.

[17] Capriotti, L., 2010. Fast Greeks by algorithmic differentiation. Journal of Computational Finance, 3, pp.3-35.

[18] Capriotti, L., Lee, S. J. and Peacock, M., 2011. Real time counterparty credit risk management in Monte Carlo. Risk, 24, pp.86-90.

[19] Capriotti, L. and Giles, M.B., 2011. Algorithmic differentiation: adjoint Greeks made easy. Risk, 25, pp.92-98.

[20] Capriotti, L. and Lee, S. J., 2013. Adjoint credit risk management. Risk, 27:92-98.

[21] Capriotti, L., Jiang, Y. and Macrina, A., 2015. Real-time risk management: An AAD-PDE approach. International Journal of Financial Engineering, 2(4), pp.1550039.

[22] Capriotti, L., Jiang, Y. and Macrina, A., 2017. AAD and least-square Monte Carlo: Fast Bermudan-style options and XVA Greeks. Algorithmic Finance, 6(1-2), pp.35-49.

[23] Capriotti, L. and Jiang, Y., 2019. Solving partial differential equations by artificial neural network. UCL working paper.

[24] Carriere, J. F., 1996. Valuation of the early-exercise price for options using simulations and nonparametric regression. Insurance: Mathematics and Economics, 19(1), pp.19-30.

[25] Cesari, G., Aquilina, J., Charpillon, N., Filipovic, Z., Lee, G. and Manda, I., 2009. Modelling, Pricing, and Hedging Counterparty Credit Exposure: A Technical Guide. Springer, Berlin.

[26] Charney, J.G., Fjörtoft, R. and Neumann, J.V., 1950. Numerical integration of the barotropic vorticity equation. Tellus, 2(4), pp.237-254.

[27] Christianson, B., 1994. Reverse accumulation and attractive fixed points. Optimization Methods and Software, 3(4), pp.311-326.

[28] Clément, E., Lamberton, D. and Protter, P., 2002. An analysis of a least squares regression method for American option pricing. Finance and Stochastics, 6(4), pp.449-471.

[29] Cox, J. C., Ingersoll Jr, J. E. and Ross, S. A., 2005. A theory of the term structure of interest rates. Theory of Valuation, pp.129-164.

[30] Crank, J. and Nicolson, P., 1996. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Advances in Computational Mathematics, 6(1), pp.207-226.

[31] Crépey, S., Bielecki, T. R. and Brigo, D., 2014. Counterparty Risk and Funding: A Tale of Two Puzzles. Chapman & Hall/CRC Financial Mathematics Series, Boca Raton.

[32] D-fine Gmbh, 2015. Computing Sensitivities of CVA Using Adjoint Algorithmic Differentiation. M. Sc. thesis, University of Oxford. Available from `https://www.nag.co.uk/market/articles/computing-sensitivities-of-cva-using-aad.pdf`.

[33] Dornier, F. and Queruel, M., 2000. Caution to the wind. Energy & Power Risk Management, 13(8), pp.30-32.

[34] Frich, P., Alexander, L. V., Della-Marta, P. M., Gleason, B., Haylock, M., Tank, A. K. and Peterson, T., 2002. Observed coherent changes in climatic extremes during the second half of the twentieth century. Climate Research, 19(3), pp.193-212.

[35] Friedman, J., Hastie, T. and Tibshirani, R., 2001. The Elements of Statistical Learning (Vol. 1, No. 10). Springer, New York.

[36] Fries, C.P., 2017. Automatic Backward Differentiation for American Monte-Carlo Algorithms-ADD for Conditional Expectations and Indicator Functions. SSRN, available from `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3000822`.

[37] Giles, M. B., 2008. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. Advances in Automatic Differentiation (pp. 35-44). Springer, Berlin.

[38] Giles, M. and Glasserman, P., 2006. Smoking adjoints: Fast monte carlo greeks. Risk, 19(1), pp.88-92.

[39] Glasserman, P., 2013. Monte Carlo Methods in Financial Engineering (Vol. 53). Springer, New York.

[40] Glasserman, P. and Yu, B., 2004. Number of paths versus number of basis functions in American option pricing. The Annals of Applied Probability, 14(4), pp.2090-2119.

[41] Göing-Jaeschke, A. and Yor, M., 2003. A clarification note about hitting times densities for Ornstein-Uhlenbeck processes. Finance and Stochastics, 7(3), pp.413-415.

[42] Griewank, A. & Walther, A., 2008. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Society for Industrial and Applied Mathematics, Philadelphia.

[43] Gutiérrez, R., Ricciardi, L. M., Román, P. and Torres, F., 1997. First-passage-time densities for time-non-homogeneous diffusion processes. Journal of Applied Probability, 34(3), pp.623-631.

[44] Henrard, M., 2011. Adjoint algorithmic differentiation: Calibration and implicit function theorem. OpenGamma Quantitative Research, 1. URL: `https://developers.opengamma.com/quantitative-research/Adjoint-Algorithmic-Differentiation-OpenGamma.pdf`

[45] Hernandez-del-Valle, G., 2012. On the first time that an Ito process hits a barrier. Available at arXiv:1209.2411.

[46] Hirsch, M.W., Devaney, R.L. and Smale, S., 1974. Differential equations, dynamical systems, and linear algebra (Vol. 60). Academic press.

[47] Hull, J., 1996. Using Hull-White interest rate trees. Journal of Derivatives, 3(3), pp.26-36.

[48] Hull, J. and White, A., 2012. CVA and wrong-way risk. Financial Analysts Journal, 68(5), pp.58-69.

[49] Jiang, Y., Peters, G. W., and Macrina, A, 2018. Multiple barrier-crossings of an Ornstein-Uhlenbeck diffusion in consecutive periods. Available at arxiv:1902.05282.

[50] Karlin, S. and Taylor, H.E., 1981. A Second Course in Stochastic Processes. Academic Press, London.

[51] Karatzas, I. and Shreve, S. E., 1998. Methods of Mathematical Finance (Vol. 39, pp. xvi+-407). New York: Springer.

[52] Lando, D., 1998. On Cox processes and credit risky securities. Review of Derivatives research, 2(2-3), pp.99-120.

[53] Lebedev, N. N. and Silverman, R. A., 1972. Special Functions and Their Applications. Prentice-Hall, Englewood Cliffs.

[54] Leblanc, B., Renault, O. and Scaillet, O., 2000. A correction note on the first passage time of an Ornstein-Uhlenbeck process to a boundary. Finance and Stochastics, 4(1), pp.109-111.

[55] Leclerc, M., Liang, Q. and Schneider, I., 2009. Fast Monte Carlo Bermudan greeks. Risk, 22(7), pp.84.

[56] Lee, S. and Capriotti, L., 2015. Wrong way risk done right. Risk (September), pp. 74-79.

[57] Linetsky, V., 2004. Lookback options and diffusion hitting times: A spectral expansion approach. Finance and Stochastics, 8(3), pp.373-398.

[58] Linetsky, V., 2004. Computing hitting time densities for CIR and OU diffusions: Applications to mean-reverting models. Journal of Computational Finance, 7, pp.1-22.

[59] Lo, C. F. and Hui, C. H., 2006. Computing the first passage time density of a time-dependent Ornstein-Uhlenbeck process to a moving boundary. Applied Mathematics Letters, 19(12), pp.1399-1405.

[60] Longstaff, F. A. and Schwartz, E. S., 2001. Valuing American options by simulation: a simple least-squares approach. The Review of Financial Studies, 14(1), pp.113-147.

[61] O'Kane, D., 2011. Modelling Single-name and Multi-name Credit Derivatives (Vol. 573). John Wiley & Sons, New York.

[62] Øksendal, B., 2003. Stochastic Differential Equations (pp. 65-84). Springer, Berlin.

[63] Østerby, O., 2003. Five ways of reducing the Crank–Nicolson oscillations. BIT Numerical Mathematics, 43(4), pp.811-822.

[64] Patie, P., 2004. On some first passage time problems motivated by financial applications. Ph.D. thesis, Universität Zürich.

[65] Peters, G. W. and Shevchenko, P. V., 2015. Advances in Heavy Tailed Risk Modeling: A Handbook of Operational Risk. John Wiley & Sons, New York.

[66] Pitman, J. and Yor, M., 1981. Bessel processes and infinitely divisible laws. Stochastic Integrals (pp. 285-370). Springer, Berlin.

[67] Pitman, J. and Yor, M., 1982. A decomposition of Bessel bridges. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, 59(4), pp.425-457.

[68] Pooley, D. M., Vetzal, K. R. and Forsyth, P. A., 2003. Convergence remedies for non-smooth payoffs in option pricing. Journal of Computational Finance, 6(4), pp.25-40.

[69] Ricciardi, L. M. and Sato, S., 1988. First-passage-time density and moments of the Ornstein-Uhlenbeck process. Journal of Applied Probability, 25(1), pp.43-57.

[70] Strikwerda, J.C., 2004. Finite difference schemes and partial differential equations (Vol. 88). SIAM, Philadelphia.

[71] Tsitsiklis, J. N. and Van Roy, B., 2001. Regression methods for pricing complex American-style options. IEEE Transactions on Neural Networks, 12(4), pp.694-703.

[72] Tuckwell, H. C. and Wan, F. Y., 1984. First-passage time of Markov processes to moving barriers. Journal of Applied Probability, 21(4), pp.695-709.

[73] Wenocur, M. L., 1987. Diffusion first passage times: approximations and related differential equations. Stochastic Processes and their Applications, 27, pp.159-177.

[74] Wilmott, P., Howison, S. and Dewynne, J., 1995. The Mathematics of Financial Derivatives: A Student Introduction. Cambridge University Press, Cambridge.

[75] Wilmott, P., 2007. Paul Wilmott Introduces Quantitative Finance. John Wiley & Sons, New York.

[76] Yi, C., 2010. On the first passage time distribution of an Ornstein-Uhlenbeck process. Quantitative Finance, 10(9), pp.957-960.

[77] Zaitsev, V. F. and Polyanin, A. D., 2002. Handbook of Exact Solutions for Ordinary Differential Equations. Chapman & Hall/CRC, Boca Raton.

[78] Zvan, R., Vetzal, K.R. and Forsyth, P.A., 2000. PDE methods for pricing barrier options. Journal of Economic Dynamics and Control, 24(11-12), pp.1563-1590.