Efficient Pricing of High-Dimensional American-Style Derivatives: A Robust Regression Monte Carlo Method

INAUGURAL-DISSERTATION

zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Christian Jonen
aus Bocholt

im September 2011

Berichterstatter:	Prof. Dr. R. Seydel Prof. Dr. C. Tischendorf
Tag der mündlichen Prüfung:	11. November 2011

Abstract

Pricing high-dimensional American-style derivatives is still a challenging task, as the complexity of numerical methods for solving the underlying mathematical problem rapidly grows with the number of uncertain factors. We tackle the problem of developing efficient algorithms for valuing these complex financial products in two ways. In the first part of this thesis we extend the important class of regression-based Monte Carlo methods by our Robust Regression Monte Carlo (RRM) method. The key idea of our proposed approach is to fit the continuation value at every exercise date by robust regression rather than by ordinary least squares; we are able to get a more accurate approximation of the continuation value due to taking outliers in the cross-sectional data into account. In order to guarantee an efficient implementation of our RRM method, we suggest a new Newton-Raphson-based solver for robust regression with very good numerical properties. We use techniques of the statistical learning theory to prove the convergence of our RRM estimator. To test the numerical efficiency of our method, we price Bermudan options on up to thirty assets. It turns out that our RRM approach shows a remarkable convergence behavior; we get speed-up factors of up to over four compared with the state-of-the-art Least Squares Monte Carlo (LSM) method proposed by Longstaff and Schwartz (2001). In the second part of this thesis we focus our attention on variance reduction techniques. At first, we propose a change of drift technique to drive paths in regions which are more important for variance and discuss an efficient implementation of our approach. Regression-based Monte Carlo methods might be combined with the Andersen-Broadie (AB) method (2004) for calculating lower and upper bounds for the true option value; we extend our ideas to the AB approach and our technique leads to speed-up factors of up to over twenty. Secondly, we research the effect of using quasi-Monte Carlo techniques for producing lower and upper bounds by the AB approach combined with the LSM method and our RRM method. In our study, efficiency has high priority and we are able to accelerate the calculation of bounds by factors of up to twenty. Moreover, we suggest some simple but yet powerful acceleration techniques; we research the effect of replacing the double precision procedure for the exponential function and introduce a modified version of the AB approach. We conclude this thesis by combining the most promising approaches proposed in this thesis, and, compared with the state-of-the-art AB method combined with the LSM method, it turns out that our ultimate algorithm shows a remarkable performance; speed-up factors of up to over sixty are quite possible.

Key words: American options; Bermudan options; Monte Carlo simulation; multiple state variables; regression-based Monte Carlo methods; outliers; Newton-Raphson method; dual methods; variance reduction; quasi-Monte Carlo; importance sampling; stochastic approximation.

Zusammenfassung

Die Bewertung von höherdimensionalen amerikanischen Derivaten zählt nach wie vor zu den faszinierendsten und anspruchvollsten Thematiken auf dem Gebiet der numerischen Finanzmathematik. Die Schwierigkeit liegt insbesondere darin, dass mit wachsender Anzahl an unsicheren Faktoren auch die Komplexität der numerischen Methoden, die zur Lösung herangezogen werden, rapide ansteigt. Um dieses Problem zu bewältigen, haben wir uns zum Ziel gesetzt, effiziente Algorithmen zur Bewertung dieser komplexen Finanzprodukte zu entwickeln. Dabei werden zwei Herangehensweisen gewählt. Zum einen erweitern wir die bedeutende Klasse regressionsbasierter Monte-Carlo Methoden um unsere eigens konzipierte Robuste-Regression-Monte-Carlo (RRM) Methode. Kernidee dieses Ansatzes ist es, den Haltewert zu jedem Ausübungszeitpunkt mittels Robuster Regression anstelle der Methode der kleinsten Quadrate anzupassen. Dadurch erzielen wir eine präzisere Approximation des Haltewerts, da die Verwendung von Robuster Regression Ausreißer mit in Betracht zieht. Um eine effiziente Implementierung der RRM Methode gewährleisten zu können, empfehlen wir einen neuartigen Newton-Raphson-basierten Löser für die Robuste Regression, der über wünschenswerte numerische Eigenschaften verfügt. Zur Überprüfung der Konvergenz unseres RRM Schätzers kommen Techniken der Statistischen Lerntheorie zum Einsatz. Um die numerische Effizienz unserer Methode zu testen, bewerten wir Bermuda-Optionen auf bis zu 30 Assets. Es stellt sich heraus, dass unser RRM Ansatz ein bemerkenswertes Konvergenzverhalten zeigt. Der Vergleich unserer Methode mit der in der Praxis fest etablierten Least-Squares-Monte-Carlo (LSM) Methode von Longstaff und Schwartz (2001) liefert uns Speed-Up-Faktoren der Größe vier und höher. Des Weiteren beschäftigen wir uns in der vorliegenden Arbeit mit der Thematik der Varianzreduktion. Wir entwickeln eine Technik für den Driftwechsel, um Pfade in für die Varianz begünstigte Regionen umzulenken. Zusätzlich diskutieren wir die effiziente Implementierung dieser Technik. Zur Berechnung von oberen und unteren Schranken für den wahren Optionswert können regressionsbasierte Monte-Carlo Methoden mit der Andersen-Broadie (AB) Methode (2004) kombiniert werden. Durch die Erweiterung unseres Ansatzes auf eben diesen AB Ansatz lässt sich die Konvergenz um einen Faktor größer 20 beschleunigen. Zusätzlich analysieren wir die Auswirkungen von Quasi-Monte-Carlo Techniken bei der Berechnung oberer und unterer Schranken mittels der Kombination des AB und des LSM Ansatzes sowie unseres RRM Ansatzes. Der effizienten Implementierung messen wir wiederum oberste Priorität bei. Als Resultat erhalten wir eine bis zu 20-fache Beschleunigung der Berechnung der Schranken. Des Weiteren wenden wir einfache, allerdings äußerst effektive Techniken zur Konvergenzbeschleunigung an, insbesondere schlagen wir eine modifizierte Version des AB Ansatzes vor. Abschließend kombinieren wir eine Vielzahl der im Rahmen dieser Arbeit vorgeschlagenen Ansätze und vergleichen die Resultate mit denen der bekannten AB - in Kombination mit der LSM - Methode. Der finale Algorithmus zeigt eine außerordentliche Performance - Speed-Up-Faktoren der Größe 60 und höher sind durchaus erzielbar.

Contents

List of Acronyms **iii**

List of Figures **v**

List of Tables **viii**

1 Introduction **1**

2 Mathematical Framework **5**

2.1 Optimal Stopping Problem 5

2.2 Regression-Based Monte Carlo Methods 10

3 Robust Regression Monte Carlo Method **19**

3.1 Algorithm 19

3.2 Solving Robust Regression 22

3.3 Convergence 27

3.3.1 Error Decomposition 29

3.3.2 Error Estimates 32

3.4 Numerical Investigations 38

3.4.1 Numerical Experiments 39

3.4.2 Dual Methods 46

4 Efficiency Increase **54**

4.1 Variance Reduction via Importance Sampling 54

4.1.1 Variance Reduction by a Change of Drift 55
4.1.2 Optimization Methods 59
4.1.3 Dual Methods 61
4.1.4 Numerical Investigations 62
4.2 Quasi-Monte Carlo Methods 73
4.2.1 Sequences with Low Discrepancy 73
4.2.2 Randomization 78
4.2.3 Dimensionality Reduction 81
4.2.4 Numerical Investigations 84
4.3 Further Acceleration Techniques 92

5 Conclusions **97**

Bibliography **101**

List of Acronyms

AB	Andersen-Broadie
ATM	At-the-money
AV	Antithetic variables
BBC	Brownian bridge construction
BFGS	Broyden-Flechter-Goldfarb-Shanno
CI	Confidence interval
DPP	Dynamic programming principle
GPU	Graphics processing unit
IRLS	Iteratively re-weighted least squares
ISNew	Newton-Raphson solver for finding the optimal drift
ISQNew	Broyden-Fletcher-Goldfarb-Shanno solver for finding the optimal drift
ISRoMo	Truncated RM method for finding the optimal drift
ITM	In-the-money
LSM	Least Squares Monte Carlo
$\mathbf{LSM}_{\mathbf{AV}}$	Least Squares Monte Carlo method combined with AV
$\mathbf{LSM}_{\mathbf{Exp}}$	Least Squares Monte Carlo method combined with replacing the double precision routine $\exp(x)$
$\mathbf{LSM}_{\boldsymbol{\Delta}}$	Least Squares Monte Carlo method combined with modified inner simulation procedure
LSQM	Least Squares Quasi-Monte Carlo
$\mathbf{M}_{\mathbf{BBC}}$	Method M combined with BBC for the regression step and RWR for calculating bounds
$\mathbf{M}_{\mathbf{PCC}}$	Method M combined with PCC for the regression step and for calculating bounds
$\mathbf{M}_{\mathbf{PCRWC}}$	Method M combined with PCC for the regression step and for the ordinary simulations, but with RWR for the inner simulations
$\mathbf{M}_{\mathbf{RWR}}$	Method M combined with PCC for the regression step and RWR for calculating bounds

$\mathbf{M^{LH}}$	Method M combined with the Halton sequence leaped
$\mathbf{M^{S}}$	Method M combined with the Sobol sequence
Max	Maximum
Min	Minimum
OTM	Out-of-the-money
PCC	Principal component construction
QMC	Quasi-Monte Carlo
RM	Robbins-Monro
RMSE	Root mean square error
RRM	Robust Regression Monte Carlo
$\mathbf{RR_{New}}$	Proposed robust regression solver
$\widetilde{\mathbf{RR}}_{\mathbf{New}}$	Existing Newton-Raphson-based robust regression solver
RRQM	Robust Regression Quasi-Monte Carlo
$\mathbf{RRQM_{AllIn}}$	Robust Regression Quasi-Monte Carlo method combined with most efficient proposed approaches
RSM1	Random shift modulo 1
RWR	Random walk construction
SE	Standard error
$\mathbf{SUF}_1$	Speed-up factor for lower bound
$\mathbf{SUF}_2$	Speed-up factor for $\widehat{\Delta}_0$
SVD	Singular value decomposition
VR	Variance reduction

List of Figures

2.1 Left panel shows the payoff of a Max call option and right panel shows the payoff of an arithmetic average call option on two assets. 7

2.2 Early exercise region for an American Max put option on two assets. . . . 8

2.3 Paths of a two-dimensional geometric Brownian motion. 11

2.4 Pricing procedure based on the DPP of the value process. 12

2.5 Approximation of the continuation value via least squares at a fixed time date. 14

2.6 Functionality of the LSM method. 14

2.7 Evaluation procedure of an American option for a given early exercise strategy. 16

2.8 Path simulations for determining an upper bound for an American option by the AB approach. 17

3.1 Approximation of the continuation value by least squares for an American Max call option on two assets; red points denote outliers. 20

3.2 Loss functions for robust regression; γ_1 and γ_2 are transition points. 21

3.3 Convergence of $\mathrm{RR}_{\mathrm{New}}$ for the Jonen loss function. 41

3.4 Bias calculated by the LSM method ($\alpha = \beta = 1$) and the RRM method combined with the Huber ($\beta = 1$), Talwar ($\alpha = 1$), Jonen ($\alpha \neq 1$, $\beta \neq 1$) loss function for a Bermudan Max call option on two assets. 43

3.5 SE (standard error) ratios for several transition points calculated for an OTM, ATM and ITM Bermudan Max call option on two assets. 43

3.6 Bias and SE ratio on the left and on the right, respectively, calculated by the LSM method and the RRM method for a Bermudan arithmetic average call option on two assets. 44

3.7 RMSE calculated by the LSM method, the left half, and the RRM method, the right half, for an increasing number of paths and functions of basis (3.52). 45

3.8 LSM method vs. RRM method for max call options on two assets with $\sigma_d = 0.2$, on the left, and $\sigma_d = 0.4$, on the right. 46

3.9 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for a Bermudan arithmetic average option on five assets. 48

3.10 Δ_0 calculated by the LSM and RRM method with several basis functions and an increasing number of paths N_0 for an ATM Bermudan Max call option on five assets. 50

3.11 Δ_0 against number of paths calculated by the AB approach combined with the LSM and RRM method for an ATM Bermudan Max call option on five assets. 51

4.1 Change of drift in Brownian motion for American options. 55

4.2 Search paths of the truncated RM algorithm and Newton-Raphson method for finding the drift minimizing variance. 64

4.3 Standard error for several drift parameters. 65

4.4 Convergence of the truncated RM algorithm and Newton-Raphson method for finding the drift minimizing variance. 66

4.5 Early exercise regions for an American Max call option on two assets. . . . 70

4.6 First 2,000 Halton points in dimension 40 projected onto the first two coordinates (bases $p_1 = 2$ and $p_2 = 3$) and onto the last two coordinates (bases $p_{39} = 167$ and $p_{40} = 173$) in the left and right panel, respectively. . . 75

4.7 First 2,000 Sobol points in dimension 100 projected onto the first two coordinates (dimensions 1 and 2) and the last two coordinates (dimensions 99 and 100). 76

4.8 Construction of the 40-dimensional Halton sequence leaped with leap 179. Left panel shows the projection of the first 358,000 Halton points onto the last two dimensions and right panel shows the first 2,000 points of the Halton sequence leaped onto the last two dimensions. 77

4.9 First 2,000 points of the Halton sequence leaped in dimension 40. Left and right panels show the projection of the Halton sequence leaped with $\mathcal{L} = 233$ and $\mathcal{L} = 269$, respectively, onto the last two coordinates. 77

4.10 Path construction by BBC for four time steps. 82

4.11 Accumulated eigenvalues of a 5-dimensional Brownian motion for nine time steps. 85

4.12 Convergence of the lower bound for the LSM and RRQM methods combined with several quasi-Monte Carlo techniques. 89

4.13 CPU time partitions of the AB approach for an arithmetic average call option on five assets. Left and right pie charts show partitions with $N_3 = 1,200$ and $N_3 = 5,000$, respectively. 92

4.14 Convergence of estimator $\widehat{\Delta}_0$ for a static and an adaptive approach. 94

List of Tables

2.1 Several payoffs of multi-asset call and put options; S^d and $(x)^+$ denote the value of asset d, $d = 1, ..., D$, and $\max(0, x)$, respectively. 6

3.1 Several loss functions $\ell(\cdot)$ for robust regression. 21

3.2 A posteriori relative error of different robust regression solvers. 41

3.3 Lower and upper Bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan Max call options on five assets. . 47

3.4 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan arithmetic average call options on five assets. 49

3.5 Accuracy comparison of the LSM and RRM method with respect to several basis functions for an ATM Bermudan Max call option on five assets. . . . 51

3.6 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan Max call options on five assets and given levels of tightness. 52

3.7 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for a Bermudan Max call option on thirty assets and a given level of tightness. 53

4.1 Lower and upper bounds calculated by the AB approach combined with the LSM method and several importance sampling techniques for Bermudan arithmetic average call options on five assets. 68

4.2 Lower and upper bounds calculated by the AB approach combined with the LSM method and several importance sampling techniques for Bermudan Max call options. 72

4.3 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM methods and several quasi-Monte Carlo techniques for Bermudan Max call options. 87

4.4 Lower and upper bounds calculated by the AB approach combined with the LSM and RRM methods and several quasi-Monte Carlo techniques for Bermudan arithmetic average call options. 90

4.5 Lower and upper bounds calculated by several speed-up techniques for arithmetic average call options. 96

Acknowledgements

My special acknowledgement is due to my supervisor Prof. Dr. Rüdiger Seydel for his excellent assistance and for providing an outstanding research environment. I also express my gratitude to Prof. Dr. Caren Tischendorf for serving as a referee for my thesis. Thanks are also due to my colleagues Pascal Heider, Albrecht Budke and Alexander Schröter for valuable discussions and advice. Moreover, I am grateful to my brother Martin Jonen; a further acceleration technique at the end of Chapter 4 is motivated by his diploma thesis [77]. The focus of this thesis is on the numerical construction of efficient pricing algorithms for multi-asset American-style options, and, thus, I do not consider the interesting field of pricing other financial derivatives. Nevertheless, I wish to record my gratitude to Audrey Laude, LEO, Université d'Orléans, Orléans, France, for our successful joint work on real option pricing [87]. I would like to address special thanks to Anna Mischor for support in any kind. Several parts of this thesis were presented at numerous workshops and conferences, including presentations at the DMV Jahrestagung (September 2011), Cologne, Germany, the Amsterdam-Cologne Workshop on Computational Finance (March 2011), Amsterdam, The Netherlands, the SIAM Conference on Financial Mathematics and Engineering (November 2010), San Francisco, California, USA, the 6th World Congress of the Bachelier Finance Society (June 2010), Toronto, Canada, the 10th MathFinance Conference on Derivatives and Risk Management in Theory and Practice (March 2010), Frankfurt, Germany, the 2nd SMAI European Summer School on Financial Mathematics (August 2009), Paris, France, the 23rd European Conference on Operational Research (July 2009), Bonn, Germany, and several seminars. I would like to thank participants for helpful comments and fruitful discussions. Last but not least, I am grateful to SIAM for financial support to travel to the 2010 SIAM Conference on Financial Mathematics and Engineering in San Francisco, California, USA, and to Fintegral for sponsoring the Best Poster Award at the 10th MathFinance Conference on Derivatives and Risk Management in Theory and Practice (March 2010), Frankfurt, Germany. Owing to the high demand and the fast-moving research field, I have already submitted the approach proposed in Chapter 3 as a research article [75] to the SIAM Journal of Financial Mathematics.

Chapter 1

Introduction

A fast growing complexity of financial markets has made financial mathematics and engineering to be a vital field for researchers and practitioners. Financial instruments with an early exercise feature are in great demand, and, therefore, a couple of research has been spent on pricing American-style derivatives. Bringing more complex financial instruments to market presupposes the ability to value and hedge them. The underlying financial model drives the dimension of the mathematical problem; not also the number of assets determines the complexity but also multiple parameters of uncertainty, e.g., stochastic volatility and jump components motivated by empirical evidence.

Options on one or two factors of uncertainty with an early exercise feature can be priced with standard numerical methods; for instance, the Binomial tree [35] can be extended to more complex models, see [19], [20], [66], [116], [104] and [83]. There exists a wide range of well-working numerical pricing methods in lower dimensions, an overview is given in [117]. Penalty methods for pricing options in low-factor models have been suggested in [49], [52], [103], among others. Another alternative is to adapt numerical methods to parallel computing. Such a parallel method based on fast Fourier transform and sparse grids for valuing multi-asset options has been presented in [90]. If we focus our attention on multi-factor models, all these numerical methods seem to loose their strength with increasing dimension such that they become unfeasible. Therefore, in recent years Monte Carlo methods have become more and more attractive for practice due to their flexibility. In 1977 Boyle pioneered the valuation of European claims with Monte Carlo in [18].

Let us provide an insight into the rich history of Monte Carlo solvers for pricing financial products with an early-exercise feature; we refer to [58] for a wider overview. Boessarts and Tilley pioneered Monte Carlo methods for pricing American-style options in [16] and [124], respectively. Broadie and Glasserman proposed a random tree method in [24] to generate lower and upper biased estimators as well as resulting valid confidence intervals for the true option value. The computational complexity of their method grows exponential in the number of exercise dates, but not in the underlying dimension. Furthermore, they introduced the well-known Stochastic Mesh method with linear complexity in the number of exercise dates and quadratic complexity in the number of simulated paths of the underlying Markov process. Methods based on partitioning the state-space have been proposed in [9] and [8]. All these methods approximate the optimal stopping problem for the value of an American-style option by using the dynamic programming principle. Another ansatz is the parameterization of the early exercise rule leading to finite-dimensional optimization problems, see e.g. [54], [3], [55], [72]. The idea of regression-based Monte Carlo methods is the fitting of the continuation value to the cross-sectional information provided by the simulated paths of the underlying model. The approaches by Tilley [124] and Carrière [28] are cornerstones for this class of methods; refinements are proposed in [29], [125] and [126]. A breakthrough with respect to practical applications was achieved by Longstaff and Schwartz in [91]. The success of their Least Squares Monte Carlo (LSM) method can be attributed to the reformulation of the dynamic programming principle in terms of the optimal stopping time rather than in terms of the value process. The LSM method plays a leading role in this thesis, as it is often the method of choice for practitioners. Based on an approximation of the early exercise strategy, a vital extension of Monte Carlo approaches for American-style options are duality approaches, as they are able to produce lower and upper bounds as well as resulting confidence intervals, see [64] and [114]. The Andersen-Broadie (AB) approach [4] also belongs to the class of duality approaches, and, combined with the LSM method, it is implemented in many running option pricing systems in practice. Currently, two hybrid methods have been proposed: a finite-difference type method combined with Monte Carlo techniques and a stochastic grid method involving several techniques of different Monte Carlo approaches mentioned above have been introduced in [12] and [73], respectively.

The slow probabilistic convergence of Monte Carlo estimators motivates for spending more effort in variance reduction techniques. Antithetic variates, control variates and importance sampling are approaches for increasing efficiency, see [58] for a general treatment. Especially, on the basis of the LSM method, control variate approaches have been proposed in [47] and [23]. Another way to accelerate convergence is to work with quasi-random numbers rather than with pseudo-random numbers. Paskov and Traub [109] pioneered the use of sequences with low discrepancy by evaluating 360-dimensional integrals arising from a collateralized mortgage obligation with great success. A number of researchers investigated the effect of quasi-Monte Carlo techniques for pricing financial derivatives, see [78], [1], among others; we refer to Niederreiter [101] for a brief survey. The common believe is that theoretical error bounds on quasi-Monte Carlo integration are often too pessimistic and the use of quasi-random numbers leads to significant convergence improvements; the benefit can be explained by the concept of effective dimension introduced by Caflisch et al. [27]. We refer the interested reader to the technical review article by L'Ecuyer [88]. Chaudhary [31] studied the effect of running the LSM method with quasi-random numbers by pricing options on up to four assets; we could not confirm the success of using sequences with low discrepancy for higher dimensions in our study [76]. Needless to mention, the underlying pseudo-random number generator plays a crucial role in a comparative study.

This thesis aims at developing new algorithms for the efficient pricing of multi-asset options with an early exercise feature. In so doing, speed and accuracy have high priority and, therefore, we discuss efficient implementations of our proposed approaches. To achieve this objective, we proceed as follows:

In Chapter 2 we introduce this thesis by constructing a mathematical framework for pricing American-style derivatives. After considering the optimal stopping problem in Section 2.1, we give a brief overview of regression-based Monte Carlo methods in Section 2.2. The body of this thesis is organized as follows: In the first main part, Chapter 3, we extend the class of regression-based Monte Carlo methods, and, to do so, we present our Robust Regression Monte Carlo (RRM) method in Section 3.1. Our method is based on the dynamic programming principle in terms of the optimal stopping time, and, additionally, it can be interpreted as a modification of the state-of-the-art LSM method. The key idea of our approach is to take outliers into account during the approximation process of the continuation value. To guarantee an efficient implementation, we propose a new Newton-Raphson-based solver for robust regression in Section 3.2. Techniques of the statistical learning theory enable us to prove the convergence of our RRM estimator in Section 3.3. In order to test the performance of our proposed approach, in Section 3.4 we focus our attention on numerical experiments and price some Bermudan-style options; a comparative study with respect to the LSM method is given and we run our RRM method with the AB approach. In the second main part of this thesis, Chapter 4, we focus our attention on techniques for improving convergence. At first, we propose a variance reduction technique via importance sampling in Section 4.1. The key idea of our approach is to drive paths in regions which are more important for variance. We introduce our change of drift technique in Subsection 4.1.1 and discuss an efficient implementation in Subsection 4.1.2. Before we conclude this section by investigating the numerical performance of our proposed approach in Subsection 4.1.4, we show the extension to the vital class of dual methods in Subsection 4.1.3. In addition to our change of drift technique for accelerating convergence, we study the use of quasi-random numbers in Section 4.2. By doing so, we consider sequences with low discrepancy in Subsection 4.2.1. For practical purposes, we briefly discuss the randomization of deterministic point sets and extend the ideas to the AB approach. As sequences with low discrepancy lose their high quality in higher dimensions, dimensionality reduction techniques might be implemented to overcome this deficiency; these tools are the issue of Subsection 4.2.3. Finally, in Subsection 4.2.4 we investigate the effect of using randomized sequences with low discrepancy for the AB approach combined with the LSM method and our RRM method. In the last section of this chapter, Section 4.3, we propose some simple but yet powerful techniques for further efficiency increases. Moreover, we use this Section to combine all proposed approaches in this thesis and give an ultimate recommendation for pricing high-dimensional options with an early exercise feature. In Chapter 5 we sum up this thesis and discuss some further possible areas of research.

Computer hardware and software

All our codes are implemented in Java and we work with CPU time ratios to manage a comparative speed-accuracy study; the experiments are run on an Intel® Xeon® X5570 (2.93 GHz).

Chapter 2

Mathematical Framework

To start with, we focus our attention on the optimal stopping problem for pricing financial derivatives with an early exercise feature in Section 2.1. As mentioned in the introduction, Monte Carlo approaches using regression for approximating the continuation value are a cornerstone for this thesis, and, therefore, we review this class of methods in Section 2.2.

2.1 Optimal Stopping Problem

Before we study the pricing of American-style options with a finite maturity date T, we shall make some assumptions:

(A1) $(\Omega, \mathcal{F}, \widetilde{P})$ is a complete probability space, where the time horizon $[0, T]$ is finite and $\mathcal{F} = \{\mathcal{F}_t | 0 \leq t \leq T\}$ is the filtration with the σ-Algebra $\mathcal{F}_t$ at time date t.

(A2) There are no arbitrage opportunities in the markets and the markets are complete. This implies the existence of a unique martingale measure P, which is equivalent to $\widetilde{P}$.

The underlying of the options to be priced as well as the riskless investment opportunity in the asset market of our economy are specified by the following two assumptions:

(A3) B_t denotes the value at time t of 1 money unit invested in a riskless money market account at time date $t = 0$, i.e. B_t is described by

$$dB_t = r_t B_t dt, \ B_0 = 1,$$

where r_t is the risk-free interest rate at time t. Then, $D_{s,t}$ denotes the discount factor given by

$$D_{s,t} = B_s / B_t, \ s, t \in [0, T].$$

Option type	Call	Put
Maximum (Max)	$(\max(S^1,\dots,S^D)-K)^+$	$(K\text{-}\max(S^1,\dots,S^D))^+$
Minimum (Min)	$(\min(S^1,\dots,S^D)-K)^+$	$(K\text{-}\min(S^1,\dots,S^D))^+$
Geometric average	$((\prod_{d=1}^D S^d)^{1/D}-K)^+$	$(K-(\prod_{d=1}^D S^d)^{1/D})^+$
Arithmetic average	$(\frac{1}{D}\sum_{d=1}^D S^d-K)^+$	$(K-\frac{1}{D}\sum_{d=1}^D S^d)^+$

Table 2.1: Several payoffs of multi-asset call and put options; S^d and $(x)^+$ denote the value of asset d, $d = 1,...,D$, and $\max(0,x)$, respectively.

In the special case of a constant risk-free rate r, we have

$$B_t = e^{rt} \text{ and } D_{s,t} = e^{-r(t-s)}.$$

(A4) The risky underlying assets or state variables of the underlying model are supposed to follow a $\mathbb{R}^D$-valued Markov process $(S_t)_{0\leq t\leq T}$, where S_0 is assumed to be known.

For the expectation conditional on the information available until time date t notify that we have $\mathbb{E}[\cdot|\mathcal{F}_t] = \mathbb{E}[\cdot|S_t]$ due to (A4). Moreover, we reduce the valuation of American options to the valuation of Bermudan options, i.e. to options admitting a finite set of exercise opportunities:

(A5) The option can be exercised at $L+1$ time points, $0 = t_0 < t_1 < ... < t_L = T$, with a constant time step width of $\triangle t = T/L$.

This assumption should be seen as nonrestrictive, as the price of an American option can be calculated by increasing the number of time steps. In the following we use the short notation $\mathbb{E}_l[\cdot] = \mathbb{E}[\cdot|S_{t_l}]$. Based on these assumptions, in 1984 Bensoussan proved that the fair price of an American or Bermudan option at time t_0 is given by

$$\sup_{\tau\in\mathcal{T}_{0,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau], \tag{2.1}$$

where $\mathcal{T}_{0,L}$ is the set of all stopping times with values in $\{0,...,L\}$ and $(Z_l)_{0\leq l\leq L}$ is an adapted payoff process, see [11]. Some typical payoffs of multi-asset options with strike K are listed in Table 2.1, and Figure 2.1 shows the payoffs of a Max call option and an arithmetic average call option on two assets. A natural ansatz to solve the optimal stopping problem (2.1) is given by the dynamic programming principle (DPP) in terms of the value process $(V_l)_{0\leq l\leq L}$ itself:

$$V_L = Z_L \tag{2.2}$$

$$V_l = \max\{Z_l, \mathbb{E}_l[D_{l,l+1}V_{l+1}]\},\ l = L-1,...,0; \tag{2.3}$$

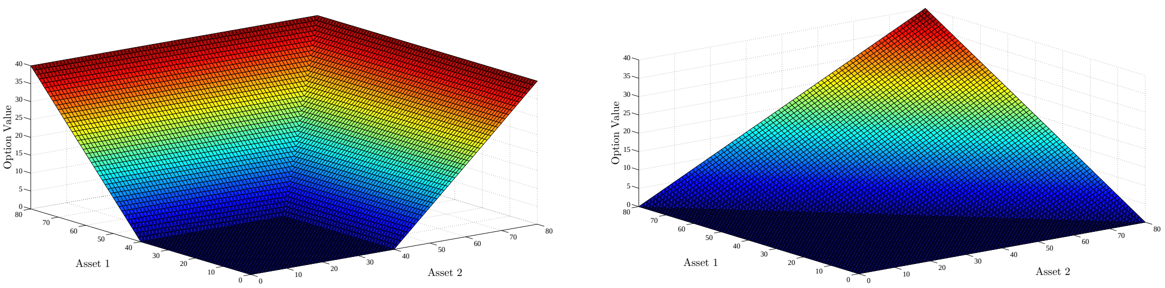

Figure 2.1: Left panel shows the payoff of a Max call option and right panel shows the payoff of an arithmetic average call option on two assets.

$(D_{0,l}V_l)_{0\leq l\leq L}$ is the smallest supermartingale dominating the sequence $(D_{0,l}Z_l)_{0\leq l\leq L}$ and is called the Snell envelope of $(D_{0,l}Z_l)_{0\leq l\leq L}$. Then, the continuation value of the underlying option at t_l is defined by

$$C_l := \mathbb{E}_l[D_{l,l+1}V_{l+1}].$$

Indeed, as the random variable

$$\tau_0^* = \inf\{l \geq 0 | V_l = Z_l\}$$

defines a stopping time and the sequence $(V_l^{\tau_0^*})_{0\leq l\leq L}$ is a martingale, we can show that

$$V_0 \;\; = \;\; \mathbb{E}_0[D_{0,\tau_0^*}Z_{\tau_0^*}] = \sup_{\tau\in\mathcal{T}_{0,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau],$$

see, e.g., [85]; $(V_l^{\tau_0^*})_{0\leq l\leq L}$ is the sequence stopped at time τ_0^* and is defined as $V_l^{\tau_0^*} := V_{\tau_0^*\wedge l}$ such that

$$V_l^{\tau_0^*} = \begin{cases} V_k, & k \leq l \\ V_l, & k > l \end{cases}$$

on the set $\{\tau_0^* = k\}$. Therefore, the concept of the Snell envelope (2.2)-(2.3) delivers a solution to the optimal stopping problem (2.1) and we call τ_0^* optimal. As a consequence, we have the more general relation

$$V_l \;\; = \;\; \mathbb{E}_l[D_{l,\tau_l^*}Z_{\tau_l^*}] = \sup_{\tau\in\mathcal{T}_{l,L}} \mathbb{E}_l[D_{l,\tau}Z_\tau]$$

with $\tau_l^* = \inf\{k \geq l | V_k = Z_k\}$, i.e. by stopping the sequence V_l adequately, we get a martingale. To get a better understanding of the valuation process of multi-asset options, Figure 2.2 shows the early exercise region of a Max put option on two assets; the intersection of the continuation value function and the payoff defines the early exercise region and the continuation region. Let us consider the continuation value process $(C_l)_{0\leq l\leq L}$

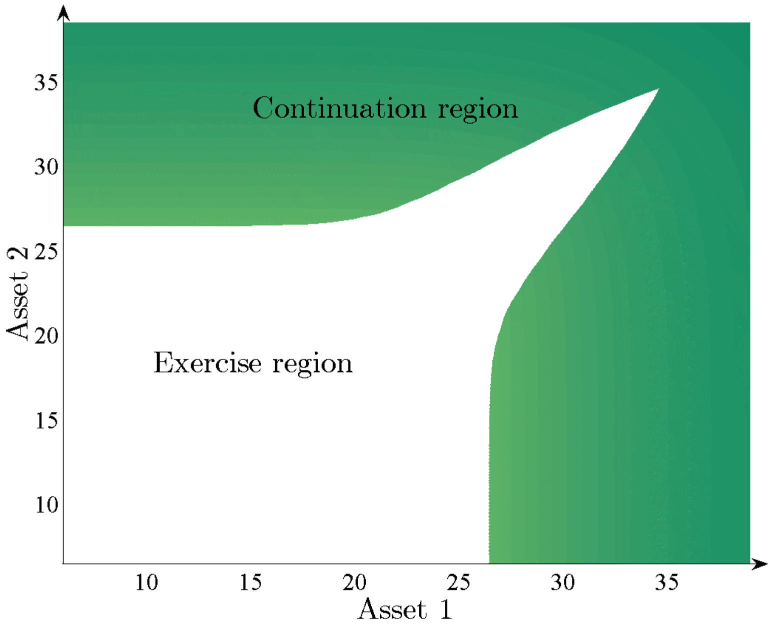

Figure 2.2: Early exercise region for an American Max put option on two assets.

vanishing at maturity date t_L. Evidentally, $(C_l)_{0\leq l\leq L}$ satisfies a DPP as well:

$$\begin{aligned} C_L &= 0 \\ C_l &= \mathbb{E}_l[D_{l,l+1}\max\{Z_{l+1}, C_{l+1}\}],\ l = L-1, ..., 0. \end{aligned}$$

Then, the fair option price at t_0 is given by C_0 or $V_0 = \max\{Z_0, C_0\}$.
Another but equivalent way is to formulate the DPP in terms of the optimal stopping time given by

$$\tau_L^* = L \tag{2.4}$$

$$\tau_l^* = \begin{cases} l & , Z_l \geq \mathbb{E}_l[D_{l,\tau_{l+1}^*} Z_{\tau_{l+1}^*}] \\ \tau_{l+1}^* & , \text{otherwise} \end{cases} \quad , l = L-1, ..., 0. \tag{2.5}$$

Apparently, the continuation value of the option at decision date t_l is defined by

$$C_l := \mathbb{E}_l[D_{l,\tau_{l+1}^*} Z_{\tau_{l+1}^*}], \tag{2.6}$$

and we obtain the fair option price at time t_0 by

$$V_0 = \mathbb{E}_0[D_{0,\tau_0^*} Z_{\tau_0^*}]. \tag{2.7}$$

For any stopping time τ with values in $\{1, ..., L\}$, the value of an American-style option is bounded below by the primal problem

$$L_0 := \mathbb{E}_0[D_{0,\tau} Z_\tau] \leq V_0, \tag{2.8}$$

due to (2.1). Now, from a practical point of view, it is desirable to derive an upper bound for our pricing problem (2.1). Based on the approach suggested by Davis and Karatzas [39] in 1994, Rogers derived in [114] a dual problem for pricing American-style options as follows:

Let $\mathcal{H}^1$ be the space of martingales $M = (M_l)_{0\leq l\leq L}$ for which $\sup_{0\leq l\leq L}|M_l| \in L_p$, where L_p denotes the space of all random variables with finite p-th moment, $p \geq 1$. For any martingale $M \in \mathcal{H}^1$, we have

$$\begin{aligned}\sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau] &= \sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau - M_\tau + M_\tau] \\ &= \sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau - M_\tau] + M_0 \qquad (2.9) \\ &\leq \mathbb{E}_0\left[\max_{l=1,\ldots,L}(D_{0,l}Z_l - M_l)\right] + M_0, \qquad (2.10)\end{aligned}$$

where the second equality (2.9) directly follows from the martingale property and the optional sampling theorem. As the upper bound (2.10) holds for any martingale M, the option value might be estimated by

$$\sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0[D_{0,\tau}Z_\tau] \leq \inf_{M\in\mathcal{H}^1}\left(\mathbb{E}_0\left[\max_{l=1,\ldots,L}(D_{0,l}Z_l - M_l)\right] + M_0\right). \qquad (2.11)$$

According to the primal problem (2.8), the right-hand side of (2.11) is called dual problem for pricing American-style derivatives. Considering (2.11) suggests to find a martingale $M \in \mathcal{H}^1$ so that we yield equality. For this purpose, let us consider the Doob-Meyer decomposition of the supermartingale $(D_{0,l}V_l)_{0\leq l\leq L}$, which allows for splitting the process $(D_{0,l}V_l)_{0\leq l\leq L}$ into two components, a martingale M^* and an increasing process $(A_l)_{0\leq l\leq L}$ with $M^*_0 = V_0$ and $A_0 = 0$, respectively. More precisely, the Doob-Meyer theorem justifies the unique decomposition

$$D_{0,l}V_l = M^*_l - A_l,$$

where the processes $(M^*_l)_{0\leq l\leq L}$ and $(A_l)_{0\leq l\leq L}$ are recursively given by

$$M^*_0 = V_0,\ M^*_l = M^*_{l-1} + D_{0,l}V_l - \mathbb{E}_{l-1}[D_{0,l}V_l],\ l = 1, ..., L, \qquad (2.12)$$

and

$$A_0 = 0,\ A_l = A_{l-1} + D_{0,l-1}V_{l-1} - \mathbb{E}_{l-1}[D_{0,l}V_l],\ l = 1, ..., L,$$

respectively, see [85]. Immediately, it follows the essential inequality

$$\mathbb{E}_0\left[\max_{l=1,\ldots,L}(D_{0,l}Z_l - D_{0,l}V_l - A_l)\right] + V_0 \leq V_0,$$

as $(D_{0,l}V_l)_{0\leq l\leq L}$ dominates $(D_{0,l}Z_l)_{0\leq l\leq L}$. Thus, to get a tight upper bound for the price of an American-style option, we should try to find a good approximation of M^* according

to (2.12).
Let us draw some conclusions from these theoretical results for the construction of algorithms for option pricing. Obviously, to value American-style derivatives we might build algorithms based on the DPP, and, to do so, a common way is to use the DPP (2.2)-(2.3); e.g., needless to mention, the Binomial tree [35] makes use of the DPP in terms of the value process itself. Anyway, in view of practical applications, it seems to be convenient to go a further step and to pursue the following strategy:

1. Find a stopping policy τ with values in $\{1, ..., L\}$ and a martingale M that are optimal in the sense that they are good approximations to the optimal stopping time τ_1^* and the optimal martingale M^*, respectively.

2. Calculate a lower bound L_0 and an upper bound U_0 for the fair value of an American-style option by

$$L_0 := \mathbb{E}_0[D_{0,\tau} Z_\tau] \leq \sup_{\tau \in \mathcal{T}_{1,L}} \mathbb{E}_0[D_{0,\tau} Z_\tau] \leq \mathbb{E}_0 \left[\max_{l=1,...,L} (D_{0,l} Z_l - M_l) \right] + M_0 =: U_0. \tag{2.13}$$

Following this approach constricts the fair price, and, therefore, we are not in the dark if we price high-complex financial instruments.

2.2 Regression-Based Monte Carlo Methods

The mathematical framework presented in the previous section is the starting point of a number of Monte Carlo methods for pricing financial derivatives with an early exercise feature. It seems to be obvious to use the DPP in terms of the value process itself; for instance, the well-known stochastic mesh method proposed by Broadie and Glasserman [25] is based on (2.2)-(2.3). Let us introduce the functionality of the vital class of regression-based Monte Carlo methods. The core of Monte Carlo methods for option pricing is the simulation of the underlying process as illustrated in Figure 2.3 for a two-dimensional geometric Brownian motion; in general, under the riskneutral probability measure the D-dimensional geometric Brownian motion is given by

$$dS_t^d = (r - \delta_d) S_t^d dt + \sigma_d S_t^d dW_t^d, \quad d = 1, ..., D, \tag{2.14}$$

where r is the annualized constant risk-free interest rate, δ_d the continuous dividend yield of asset d, σ_d the volatility of asset d, W_t^d a standard Brownian motion and W_t^d and W_t^e have correlation ρ_{de}. For a general treatment of the numerical integration of stochastic differential equations (SDEs), we refer the reader to [58] or [117], and the

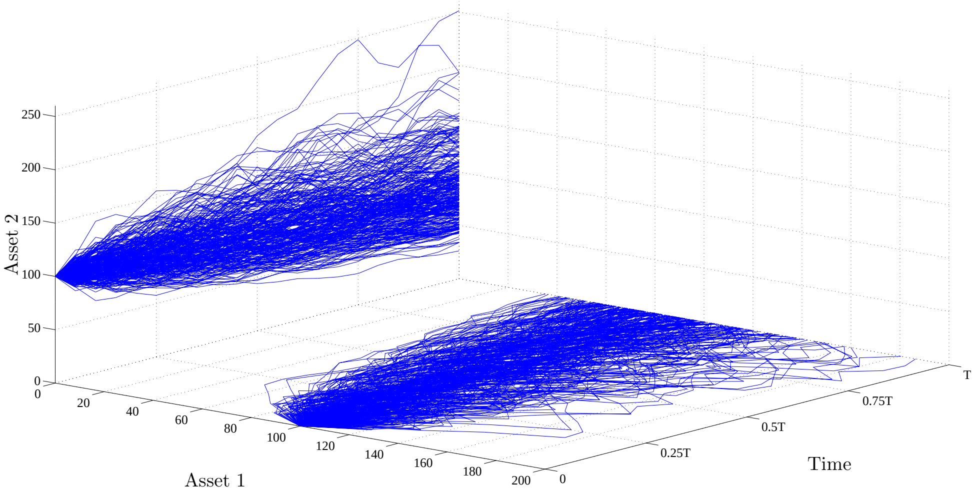

Figure 2.3: Paths of a two-dimensional geometric Brownian motion.

references therein. Without loss of generality, from now on, we assume a constant risk-free interest rate. On the basis of the DPP (2.2)-(2.3) and the set of N simulated paths $\{\mathbb{S}_{nl}\}_{n=1,...,N,l=0,...,L}$, the option value for each path n, $n = 1, ..., N$, at every exercise date t_l, $l = L, ..., 1$, is determined by

$$\begin{aligned} V_L^n &= Z_L^n \\ V_l^n &= \max\{Z_l^n, C_l(\mathbb{S}_{nl})\},\ l = L-1, ..., 1, \end{aligned}$$

as illustrated in Figure 2.4 for six paths; we assign an option value to each knot. In so doing, we clearly know the payoff of each path n, $n = 1, ..., N$, at every exercise date t_l with $1 \leq l \leq L-1$, but, obviously, we have no idea about the structure of the continuation value C_l. Thus, the key for constructing a well-working Monte Carlo method is to find a good approximation $\overline{C}_l$ of the continuation value C_l. To tackle this problem, the idea of regression-based Monte Carlo methods is to estimate a model function for the continuation value via regression. A practical model function for the continuation value is given by a simple linear combination of basis functions, i.e.

$$\overline{C}_l := \sum_{m=1}^{M} x_m \phi_m(S_l), \tag{2.15}$$

where the coefficients might be determined by solving the ordinary least squares (OLS) problem

$$\min_{x \in \mathbb{R}^M} \ \| C_l - \overline{C}_l \|_2^2; \tag{2.16}$$

for the sake of completeness, $\overline{C}_l$ is an element of a finite-dimensional linear subspace $\mathcal{H}_l$ of L_2, $\{\phi_m(\cdot)\}_{m=1}^M$ is a basis of $\mathcal{H}_l$ and the optimal coefficients $x_1, ..., x_M$ are solutions of

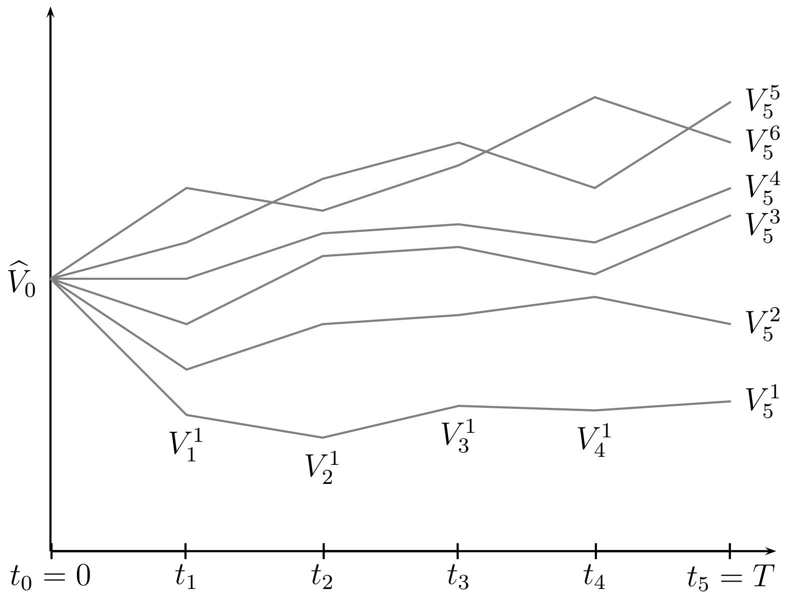

Figure 2.4: Pricing procedure based on the DPP of the value process.

the normal equations. Theoretically, a model function such as (2.15) is justified by the following further assumption:

(A5) The payoff Z_l at date t_l, $l = 0, ..., L$, is a square-integrable random variable. So, $\mathrm{E}[(Z_l)^2] < \infty$ for all Z_l, $l = 0, ..., L$, and $Z_l \in L_2(\Omega, \mathcal{F}_l, P_{S_l})$. P_{S_l} denotes the image probability measure induced by S_l on the state space $E \subseteq \mathbb{R}^D$, and L_2 is a Hilbert space.

Then, an important implication from this assumption is the following expression for the continuation value (2.6):

$$C_l \overset{(A5)}{=} \sum_{m=1}^{\infty} x_m \phi_m^*(S_l), \tag{2.17}$$

where $\{\phi_m^*(\cdot)\}_{m=1}^{\infty}$ is an orthonormal basis, and $x_m = \langle C_l, \phi_m^*(S_l)\rangle = \mathrm{E}[C_l \phi_m^*(S_l)]$ are the Fourier coefficients, see [115]. Note that the model function (2.15) results from truncating the infinite sum in (2.17). Numerically, we solve problem (2.16) by

$$\min_{\overline{x} \in \mathbb{R}^M} \frac{1}{N} \sum_{n=1}^{N} \left(C_l^n - h_l(\mathbb{S}_{nl})\right)^2, \tag{2.18}$$

where for $n = 1, ..., N$, we have $h_l(\mathbb{S}_{nl}) = \sum_{m=1}^{M} \overline{x}_m \phi_m(\mathbb{S}_{nl})$; the realizations C_l^n of the continuation value coincide with the cash flows of each path, i.e.

$$C_l^n = e^{-r\Delta t} V_{l+1}^n = e^{-r\Delta t} \max\{Z_{l+1}^n, h_{l+1}(\mathbb{S}_{n,l+1})\}.$$

Following this ansatz, an approximation of the fair option price today is given by

$$\widehat{V}_0 = \max\left\{ Z_0, e^{-r\Delta t}\frac{1}{N}\sum_{n=1}^{N} V_1^n \right\}. \tag{2.19}$$

Tilley pioneered the class of regression-based Monte Carlo methods in [124] by introducing the estimator

$$\widehat{V}_0 = \max\left\{ Z_0, \frac{1}{N}\sum_{n=1}^{N} e^{-r\tau_1^n\Delta t} Z_{\tau_1^n}^n \right\},$$

where τ_1 results from approximating the continuation value at every exercise date, i.e. $\tau_1^n = \inf\{k \geq 1 | Z_k^n \geq h_k(\mathbb{S}_{nk})\},\ n = 1, ..., N$. To approximate the continuation value, this approach works with a crude kernel smoothing technique; more precisely, at every exercise date realizations of the asset price are ordered in bundles to approximate the continuation value by a step function. In 1996 Carrière proposed in [28] to fit the continuation value by nonparametric regression. In addition to Tilley's estimator, Carrière introduced estimator (2.19). The approach suggested by Tsitsiklis and van Roy works with this second high-biased estimator combined with least squares, see [125] and [126]. Carrière priced options in [29] with the same estimators as in his previous paper and also with least squares. All these methods are either heavily extended to the vital multi-dimensional case or give poor approximations of the continuation value such that an accurate estimation of multi-asset options seems to be infeasible, see also [58]. A breakthrough with respect to pricing multi-dimensional options with an early exercise feature came with the proposal of Longstaff and Schwartz [91]. The key idea of their LSM method is to work with the DPP in terms of the optimal stopping time (2.4)-(2.5). By doing so, based on the set of simulated paths, we try to find the optimal stopping time for each path to approximate the fair option price by (2.7). So, numerically, the idea of the LSM algorithm is to realize the rule

$$\begin{aligned} \tau_L^n &= L \\ \tau_l^n &= \begin{cases} l & , \ Z_l(\mathbb{S}_{nl}) \geq h_l(\mathbb{S}_{nl}) \\ \tau_{l+1}^n & , \ \text{otherwise} \end{cases} \quad , l = L-1, ..., 1. \end{aligned}$$

The dependent variables for least squares are determined by $C_l^n = e^{-r(\tau_{l+1}^n - l)\Delta t} Z_{\tau_{l+1}^n},\ n = 1, ..., N,\ l = 1, ..., L$; to get a better approximation of the continuation value, Longstaff and Schwartz suggested to consider only paths in the money, see Figure 2.5. At time t_0 we estimate the continuation value by its empirical mean and get for the option value

$$\widehat{V}_0 = \max\left\{ Z_0, \frac{1}{N}\sum_{n=1}^{N} e^{-r\tau_1^n\Delta t} Z_{\tau_1^n}^n \right\}, \tag{2.20}$$

compare (2.6) and (2.7); the functionality of the LSM approach is illustrated in Figure 2.6. According to [58], the LSM estimator (2.20) is an interleaving option price estimator.

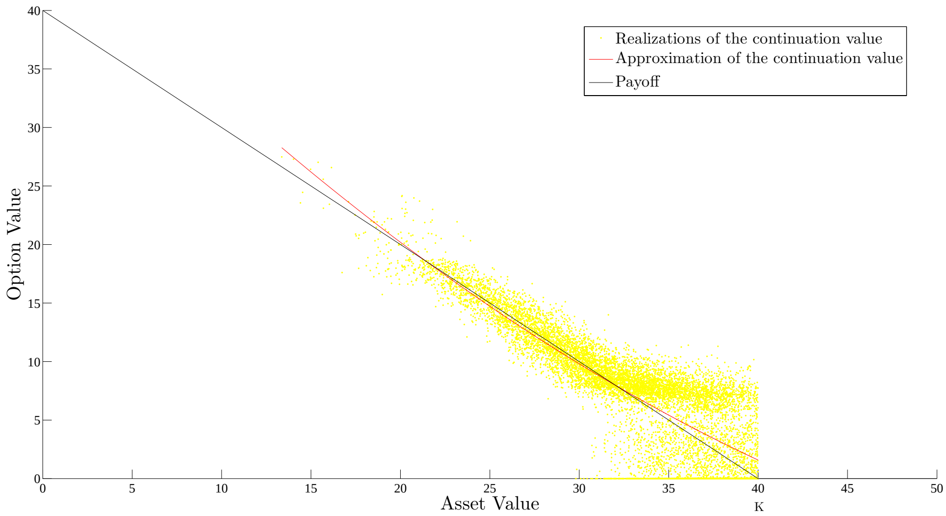

Figure 2.5: Approximation of the continuation value via least squares at a fixed time date.

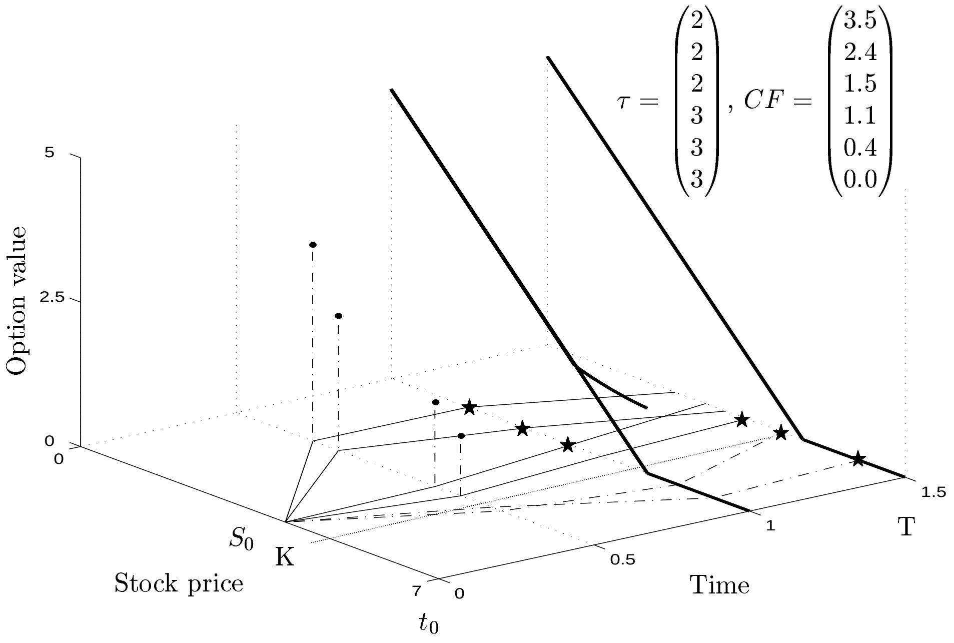
$\tau = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}, CF = \begin{pmatrix} 3.5 \\ 2.4 \\ 1.5 \\ 1.1 \\ 0.4 \\ 0.0 \end{pmatrix}$
Option value
5
2.5
0
0
S_0
K
Stock price
7
0
t_0
0.5
1
1.5
T
Time

Figure 2.6: Functionality of the LSM method.

Notes. τ and CF are vectors used for saving the current optimal stopping time and the corresponding payoff, respectively, for each path, see [76].

Notify that rough approximations of the continuation value have limited influence on the estimated option price, as just the intersection of the payoff and the continuation value function determine the optimal stopping time for each path. Even if we work with an approximation of lower quality, we meet the right decision for most of the paths; thus, the dependent variables for the regression problem and the estimated option price might be of high quality despite the dilemma of a poor continuation value function. This fact explains the striking success of the LSM method, whereby methods based on the DPP of the value process (2.2)-(2.3) are quite sensitive to perturbations in the continuation value function. Recently, in [76] we gave a guideline for an efficient implementation of the LSM method; especially, we could accelerate this method by a factor of about two or more by our vectorization procedure.

As previously mentioned, a vital extension of regression-based Monte Carlo methods are dual methods. Let us focus our attention on the AB algorithm. By doing so, consider any exercise strategy τ defined by an approximation h_l of the continuation value C_l for $l = 1, ..., L$ with $C_L = 0$ such as given by $\tau_l = \inf\{k \geq l | Z_l \geq h_l\}$, $l = 1, ..., L$; h_l might result from running the LSM method with N_0 paths. By drawing N_1 i.i.d. samples $e^{-r\tau_1^n \Delta t} Z^n_{\tau_1^n}$, $n = 1, ..., N_1$, an estimate of the lower bound L_0 in (2.13) is realized by

$$\widehat{L}_0 = \frac{1}{N_1} \sum_{n=1}^{N_1} e^{-r\tau_1^n \Delta t} Z^n_{\tau_1^n}. \tag{2.21}$$

Note that a valid $(1 - \alpha)$ confidence interval of L_0 is given by

$$\widehat{L}_0 \pm z_{1-\alpha/2} \frac{\widehat{\sigma}_L}{\sqrt{N_1}} \tag{2.22}$$

with the estimated standard error $\widehat{\sigma}_L$ of L_0; $z_{1-\alpha/2}$ denotes the $(1 - \alpha/2)$ quantile of a standard normal distribution. For a given early exercise strategy, Figure 2.7 illustrates the process of realizing estimator (2.21). Next, we consider the vital construction of a valid upper bound, and, to this end, we concentrate on the numerical realization of the right-hand side of (2.13) in the notion of the exercise policy τ; more precisely, the upper bound is given as

$$U_0 = \mathbb{E}_0 \left[\max_{l=1,...,L} (e^{-rl\Delta t} Z_l - M_l) \right] + M_0 =: \Delta_0 + M_0 \tag{2.23}$$

with

$$M_0 = \mathbb{E}_0[e^{-r\tau_1 \Delta t} Z_{\tau_1}],\ M_l = M_{l-1} + \mathbb{E}_l[e^{-r\tau_l \Delta t} Z_{\tau_l}] - \mathbb{E}_{l-1}[e^{-r\tau_l \Delta t} Z_{\tau_l}] =: M_{l-1} + \delta_l,\ l = 1, ..., L,$$

and

$$\mathbb{E}_l[e^{-r\tau_l \Delta t} Z_{\tau_l}] = \begin{cases} e^{-rl\Delta t} Z_l, & \text{if } Z_l \geq h_l \\ \mathbb{E}_l[e^{-r\tau_{l+1} \Delta t} Z_{\tau_{l+1}}], & \text{if } Z_l < h_l \end{cases}. \tag{2.24}$$

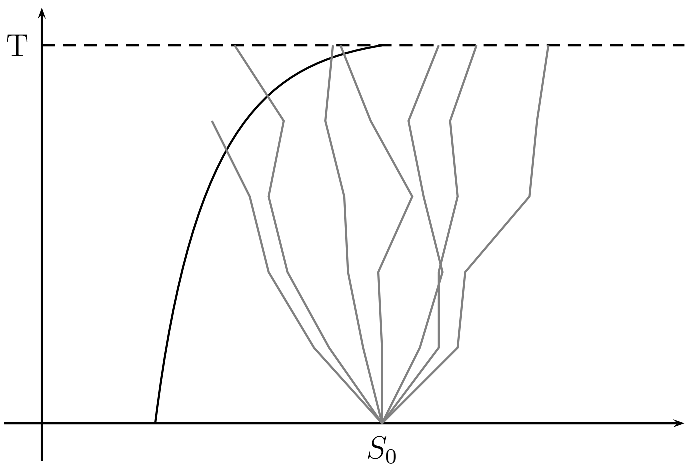

Figure 2.7: Evaluation procedure of an American option for a given early exercise strategy.

Now, to get an estimate $\widehat{U}_0$ of the upper bound U_0, we simulate another set of N_2 paths, $\{\mathbb{S}^2_{nl}\}_{n=1,...,N_2,l=1,...,L}$, to approximate Δ_0 in (2.23) by

$$\widehat{\Delta}_0 := \frac{1}{N_2}\sum_{n=1}^{N_2}\Delta_n, \tag{2.25}$$

where

$$\Delta_n = \max_{l=1,...,L}(e^{-rl\Delta t}Z^n_l - M^n_l),\ M^n_l = M^n_{l-1} + \delta^n_l;$$

to control the accuracy and to preserve the martingale property of M, Andersen and Broadie suggested to evaluate δ^n_l by

$$\delta^n_l = \begin{cases} e^{-rl\Delta t}Z^n_l, & \text{if } Z^n_l \geq h_l(\mathbb{S}^2_{nl}) \\ C^n_l, & \text{if } Z^n_l < h_l(\mathbb{S}^2_{nl}) \end{cases} - C^n_{l-1},$$

where

$$C^n_l = \frac{1}{N_3}\sum_{m=1}^{N_3} e^{-r\tau^m_{l+1}\Delta t}Z^m_{\tau^m_{l+1}} \tag{2.26}$$

are estimates of the continuation values in (2.24) resulting from drawing N_3 i.i.d. samples $e^{-r\tau^m_{l+1}\Delta t}Z^m_{\tau^m_{l+1}}$, $m = 1, ..., N_3$; $Z^m_{\tau^m_{l+1}}$ results from simulating the underlying Markov process starting at $\mathbb{S}^2_{nl}$ and stopping according to τ_{l+1}. As a final result, an estimate of the upper bound U_0 in (2.23) is given by

$$\widehat{U}_0 = \widehat{\Delta}_0 + \widehat{L}_0$$

such that

$$\widehat{U}_0 \pm z_{1-\alpha/2}\sqrt{\frac{\sigma^2_L}{N_1} + \frac{\sigma^2_\Delta}{N_2}} \tag{2.27}$$

is a valid $(1-\alpha)$ confidence interval of U_0 with the estimated standard error $\widehat{\sigma}_\Delta$ of Δ_0. In order to get a deeper look into the working procedure of the AB approach, the path construction for realizing estimator (2.25) is illustrated in Figure 2.8. As we can see, we start off by simulating N_2 paths of our underlying model; these paths are colored gray. Based on this set of paths, at each state of a path we run an inner simulation of N_3 paths to evaluate the continuation value (2.26); this procedure is demonstrated for two states and the red paths are these subsimulations stopped according to τ_l. Thus, following the AB approach, to produce valid upper bounds, we have to run subsimulations in a simulation, which might be seen critical at first glance. Anyway, as claimed by the authors, a small number of paths N_2 and N_3 are only necessary to get accurate upper bounds; we can underline this statement, see, e.g., our numerical experiments in [76]. By combining both valid confidence intervals, (2.22) and (2.27), we are in the situation to determine a valid $(1-\alpha)$ confidence interval for the true option value by

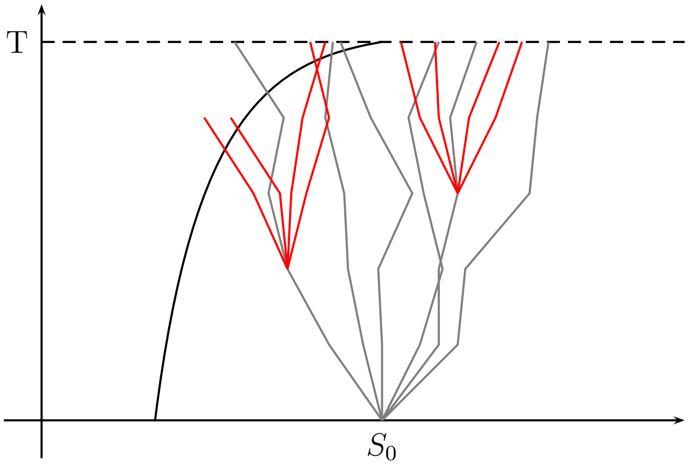

Figure 2.8: Path simulations for determining an upper bound for an American option by the AB approach.

$$\left[\widehat{L}_0 - z_{1-\alpha/2}\frac{\widehat{\sigma}_L}{\sqrt{N_1}}, \widehat{U}_0 + z_{1-\alpha/2}\sqrt{\frac{\widehat{\sigma}_L^2}{N_1}+\frac{\widehat{\sigma}_\Delta^2}{N_2}}\right];$$

this is just a simple result of taking the lower half width of (2.22) and the upper half width of (2.27). From a practical point of view, the AB approach is a valuable extension of the class of regression-based Monte Carlo methods due to the restriction of the fair option price. The higher the lower bounds and the tighter the lower and upper bounds, the more accurate is the used early exercise strategy. Thus, the tightness of the bounds is a good indicator of the quality of basis functions, which is especially of high interest for pricing new complex financial derivatives; we should spend more effort in searching well-working functions if the bounds are too large.

Independently to the proposal by Andersen and Broadie, Haugh and Kogan introduced in [64] another method to calculate lower and upper bounds as well as valid confidence intervals by using supermartingales rather than martingales. Their approach is based on the DPP of the value process (2.2)-(2.3) and also works with subsimulations. To determine an approximation of the continuation value, a neural network approach is used, which might be expensive in higher dimensions, see our comments in Chapter 5. In their numerical experiments, the authors priced multi-asset options with a huge number of paths and no CPU times are reported. We are not familiar with any comparative speed-accuracy study of both approaches, the AB method and the algorithm by Haugh and Kogan. Anyway, to the best of our knowledge, the AB approach combined with the LSM method is often the method of choice and implemented in many running option pricing systems of financial institutions for pricing complex financial derivatives with an early-exercise feature, see also [93]. This gives us rise to compare our proposed methods in this thesis with the AB approach combined with our efficient version of the LSM method rather than with the approach by Haugh and Kogan. Nevertheless, at the end of thesis we discuss some possible modifications of the Haugh-Kogan approach by our proposed methods.

Chapter 3

Robust Regression Monte Carlo Method

On the basis of the mathematical framework formulated in Chapter 2, we present our RRM method for pricing American-style options in Section 3.1. To give a complete guideline for an efficient implementation of our proposed method, we focus our attention on solving robust regression problems and suggest a new solver with very good numerical properties in Section 3.2. Techniques of the statistical learning theory help us to prove convergence of our proposed Monte Carlo estimator in Section 3.3. In Section 3.4 we investigate the numerical performance of our RRM method and aim at a comparative study with the state-of-the-art LSM method.

3.1 Algorithm

Following the idea of Longstaff and Schwartz [91], we concentrate on an approximation of the DPP in terms of the optimal stopping time (2.4)-(2.5) rather than of the option value process itself. In order to motivate our approach, let us have a closer look at one regression step. As we can see in Figure 3.1, there are some points, namely the red points, which are really far away from the light gray surface showing the continuation value function calculated by least squares for an American-style option on two assets. This observation directly leads to robust regression, as robust regression is able to take outliers into account. Coming back to the LSM algorithm considered in Chapter 2, we propose to replace least squares (2.18) by robust regression. For this purpose, to get an estimation

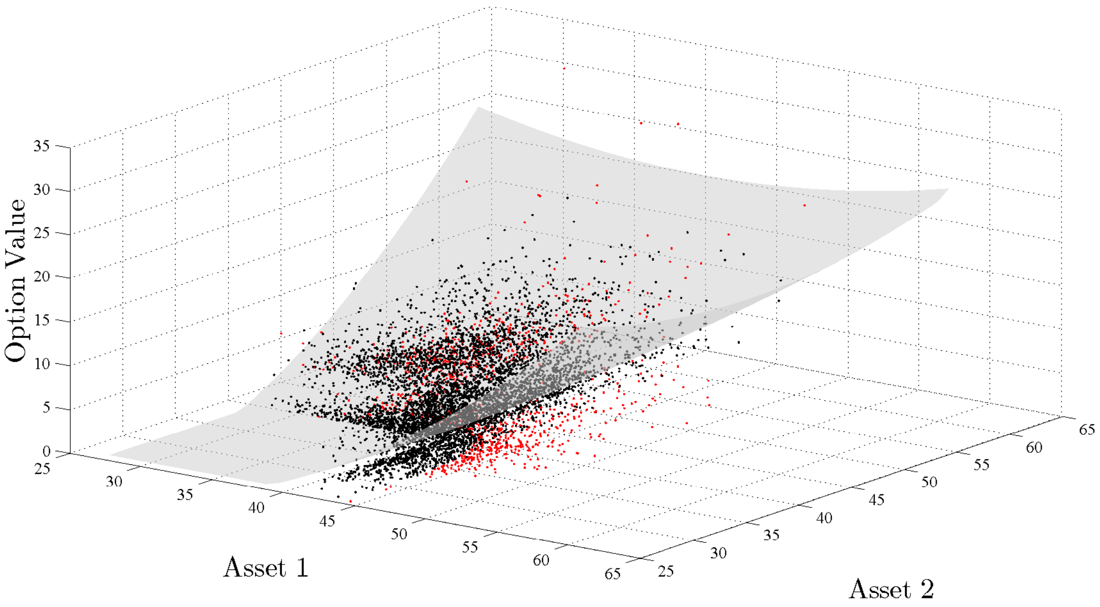

Figure 3.1: Approximation of the continuation value by least squares for an American Max call option on two assets; red points denote outliers.

of the coefficients in every time step, we suggest to solve

$$\min_{\overline{x}\in\mathbb{R}^M} f(\overline{x}) := \min_{\overline{x}\in\mathbb{R}^M} \frac{1}{N}\sum_{n=1}^{N} \ell\left(C_l^n - h_l(\mathbb{S}_{nl})\right) \tag{3.1}$$

with a suitably measurable loss function

$$\ell : \mathbb{R} \to [0, \infty)$$

specified a priori. In the following we denote the residuals by

$$r_n^l := C_l^n - h_l(\mathbb{S}_{nl}),\ n = 1, .., N,\ l = 1, ..., L-1.$$

Let us make some assumptions on this loss function:

(L1) $\ell(\cdot)$ is a piecewise twice continuously differentiable function with $\ell(r) = 0$ if $r = 0$.

(L2) $\ell'(r)/r \geq 0$ and $\ell''(r) \geq 0$ for any $r \in \mathbb{R}$.

(L3) $\ell(\cdot)$ is convex or piecewise convex.

Some selections of loss functions $\ell(\cdot)$ are listed in Table 3.1 and Figure 3.2 shows their graphs. We should remark that many loss functions are twice continuously differentiable

Table 3.1: Several loss functions $\ell(\cdot)$ for robust regression.

	$\ell(r)$		$\ell(r)$
OLS	r^2	Talwar	$\begin{cases} r^2 \ , & \lvert r\rvert \le \gamma_1 \\ \gamma_1^2 \ , & \lvert r\rvert > \gamma_1 \end{cases}$
Huber	$\begin{cases} r^2 & , \ \lvert r\rvert \le \gamma_1 \\ 2\gamma_1\lvert r\rvert - \gamma_1^2 & , \ \lvert r\rvert > \gamma_1 \end{cases}$	Jonen	$\begin{cases} r^2 & , \ \lvert r\rvert \le \gamma_1 \\ 2\gamma_1\lvert r\rvert - \gamma_1^2 & , \ \gamma_1 \le \lvert r\rvert < \gamma_2 \\ 2\gamma_1\gamma_2 - \gamma_1^2 & , \ \lvert r\rvert \ge \gamma_2 \end{cases}$

almost everywhere, and one-sided derivatives should be used at points where $\ell(\cdot)$ is not differentiable; see also our comments in Section 3.2. Huber proposed in [71] the Huber as well as the Talwar loss functions; many authors denote the Talwar loss function after [68]. Huber's function is a famous choice due to its nice statistical properties. Furthermore, we introduce a new loss function called Jonen, which is a mixture of both of them. For other common loss functions we refer to [70]. As we can see, the idea of robust regression is to give outliers fewer weight than the other points. By using robust regression, we attempt to eliminate outliers in order to improve the quality of an approximation of the continuation value function. Outliers occur by strongly fluctuated paths in the context of regression-based Monte Carlo methods based on the DPP of the optimal stopping time. By the way,

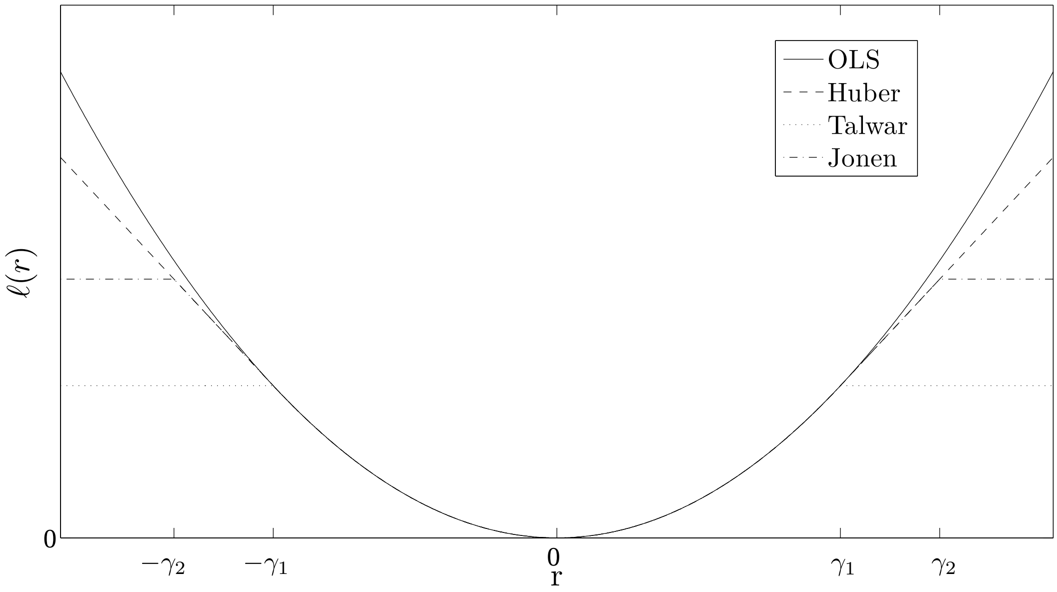

Figure 3.2: Loss functions for robust regression; γ_1 and γ_2 are transition points.

we expect fewer outliers by considering methods based on the DPP of the value process itself. Even though outliers are often bad data points, by locating points as outliers which contain valuable information, we might destroy the whole structure of the problem. So, we should investigate outliers carefully. However, in Section 3.4 we concentrate on eliminating these points from the data and address the determination of the parameters γ_1, γ_2.

Remark 3.1.1. *In a statistical framework practitioners are used to make an assumption about the distribution of the error to specify the loss function a priori. However, in our context we are not familiar with any distribution, and, that is why, we use loss functions to improve the quality of a continuation value function fit.*

A number of loss functions lead to optimization tasks which have to be solved iteratively, and we focus on that topic in the next section. Assumption (L3) guarantees the well-posedness of (3.1) in the sense that there is a unique (global) minimizer. Provided with these tools, we can formulate our RRM method, see Algorithm 1. In analogy to the LSM algorithm, we just take paths in-the-money into consideration for regression; J denotes the number of these paths. Notice that the LSM algorithm is a special case of our RRM algorithm by using the loss function

$$\ell\left(C_l^j - h_l(\mathbb{S}_{jl})\right) = \left(C_l^j - h_l(\mathbb{S}_{jl})\right)^2, \ j = 1, ..., J, \ l = 1, ..., L.$$

Thus, from an implementation point of view, we only have to replace the solver for regression if the LSM method is already implemented in a running option pricing system.

3.2 Solving Robust Regression

The routine *RobustRegression*(A, b, J) in Algorithm 1 needs special care to guarantee an efficient implementation. With the notation of the previous section the first order condition of the right-hand side of our minimization problem (3.1) becomes to the following set of M equations:

$$\sum_{j=1}^{J} \ell'(r_j^l) A_{jm} = 0, \ m = 1, ..., M \tag{3.2}$$

— for simplicity we drop in the following the overline index of x and the time index of r. In general, we obtain a nonlinear system, and, therefore, we should concentrate on iterative solvers for system (3.2). Even though a couple of approaches for solving robust regression problems exist, see e.g., [92], [33] and the mentioned approaches below, we propose a new Newton-Raphson-based solver with the purpose of an efficient implementation.

Algorithm 1 $RRM(N, M, T, L, \Theta)$.

```
Input: $N, M, T, L, \Theta$
Output: $\widehat{V}_0$
 1: $\triangle t = T/L$ // Time step width
 2: for $l \leftarrow 1$ to $L$ do
 3:     for $n \leftarrow 1$ to $N$ do
 4:         $\mathbb{S}_{nl} = (S_{(n-1)D+d,l})_{d=1,...,D} \leftarrow GeneratePaths(N, \Delta t, L, \Theta, )$
 5:     end for
 6: end for
 7: for $n \leftarrow 1$ to $N$ do
 8:     $CF_n = Z_L(\mathbb{S}_{nL})$   // Payoff for path $n$ at $t_L$
 9:     $\tau_n = L$  // Optimal exercise point for path $n$ at $t_L$
10: end for
11: for $l \leftarrow L-1$ to $1$ do
12:     $J = 0$
13:     for $n \leftarrow 1$ to $N$ do
14:         if ($n$-th path in-the-money) then
15:             $J = J + 1,\ \pi(J) = n$
16:             $A_{Jm} = \phi_m(\mathbb{S}_{nl}),\ m = 1, ..., M$  // Fill regressor matrix
17:             $b_J = e^{-r(\tau_n - l)\triangle t} CF_n$  // Fill vector with regressands
18:         end if
19:     end for
20:     $\widehat{x}_1, ..., \widehat{x}_M \leftarrow\ RobustRegression(A, b, J)$
21:     for $j \leftarrow 1, ..., J$ do
22:         $\widehat{C} = \sum_{m=1}^{M} \widehat{x}_m A_{jm}$
23:         if $(Z_l(\mathbb{S}_{\pi(j),l}) \geq \widehat{C})$ then
24:             $CF_{\pi(j)} = Z_l(\mathbb{S}_{\pi(j),l})$
25:             $\tau_{\pi(j)} = l$
26:         end if
27:     end for
28: end for
29: $\widehat{C}_0 = \frac{1}{N} \sum_{n=1}^{N} e^{-r\tau_n \triangle t} CF_n$  // Continuation value at $t_0$
30: $\widehat{V}_0$=max$\{Z_0, \widehat{C}_0\}$  // Approximated fair value at $t_0$
```

Notes. The routine $GeneratePaths(N, \Delta t, L, \Theta)$ in line 4 realizes the numerical integration of the underlying stochastic differential equation, where Θ denotes the set of parameters for the underlying model. The type of basis functions $\phi(\cdot)$ as well as the structure of the payoff Z_l must be specified a priori.

Considering minimization problem (3.1), we yield the Newton-Raphson iteration

$$x^{(k+1)} = x^{(k)} + s^{(k)},\ k = 0, 1, ..., \tag{3.3}$$

where the search direction $s^{(k)}$ results from solving

$$s^{(k)} = -(H^{(k)})^{-1} g^{(k)} \tag{3.4}$$

with the Hessian matrix

$$H^{(k)} = A^T D^{(k)} A \tag{3.5}$$

and the gradient

$$g^{(k)} = -A^T y^{(k)};$$

$y^{(k)}$ is a vector with entries $y_j^{(k)} = \ell'(r_j^{(k)})$ and $D^{(k)}$ is a diagonal matrix with elements $D_j^{(k)} = \ell''(r_j^{(k)}),\ j = 1, ..., J$. According to assumption (L1), if $\ell'(r)$ and $\ell''(r)$ are discontinuous at any point $r \in \mathbb{R}$, we work with the concept of the subgradient and the generalized Hessian matrix defined by [34], respectively, such that (3.3) is a step of the generalized Newton-Raphson method, see [112]. In order to guarantee that $s^{(k)}$ is a descent direction, the Hessian matrix (3.5) must be positive definite; indeed, it is symmetric and positive semi-definite for loss functions fulfilling (L2). A reasonable assumption in our framework is that $M \ll J$. Suppose that $U_J \Sigma_M V_M^T$ is the singular value decomposition (SVD) of $A \in \mathbb{R}^{J \times M}$, where $U_J = [u_1, ..., u_J] \in \mathbb{R}^{J \times J}$ and $V_M = [v_1, ..., v_M] \in \mathbb{R}^{M \times M}$ are orthogonal matrices with the i-th left singular vector u_i and the i-th right singular vector v_i, respectively; $\Sigma_M = \text{diag}(\sigma_1, ..., \sigma_M) \in \mathbb{R}^{J \times M}$ is a diagonal matrix with the singular values $\sigma_i,\ i = 1, ..., M$, of A, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_M \geq 0$, see [60] or [15]. Then, to avoid any numerical trouble caused by stability problems, we suggest to decompose the search direction (3.4) to

$$s^{(k)} = V\Sigma^{-1} \left(U^T D^{(k)} U\right)^{-1} U^T y^{(k)} =: V\Sigma^{-1} \left(\overline{H}^{(k)}\right)^{-1} U^T y^{(k)} \tag{3.6}$$

with $U = [u_1, ..., u_\rho] \in \mathbb{R}^{J \times \rho}$, $V = [v_1, ..., v_\rho] \in \mathbb{R}^{M \times \rho}$, and $\Sigma^{-1} = \text{diag}(1/\sigma_1, ..., 1/\sigma_\rho) \in \mathbb{R}^{\rho \times \rho}$, $\rho = \text{rank}(A)$. As U, V and Σ^{-1} are independent of the current residuals $r^{(k)}$, a cheap way to calculate the matrix $\overline{H}^{(k)}$ in step k is by adding block matrices to the matrix $\overline{H}^{(k-1)}$ of the previous step as follows:

$$\overline{H}^{(k)} = \overline{H}^{(k-1)} + \sum_{j=1}^{J} \Delta D_j^{(k)} U_j U_j^T 1_{\{|\Delta D_j^{(k)}| > \epsilon\}},\ k = 1, 2, 3, ... \tag{3.7}$$

with $\epsilon \geq 0$, $\Delta D_j^{(k)} = D_j^{(k)} - D_j^{(k-1)},\ j = 1, ..., J$; U_j^T and $1_{\{.\}}$ denote the j-th row of U and the indicator function, respectively. Notify that we have to run the SVD just one time at the beginning of every time step $t_l,\ l = L-1, ..., 1$. Similar update techniques have been used in [48],[131], [108]. Update technique (3.7) combined with search direction (3.6) lead to our solver for robust regression, see Algorithm 2.

Algorithm 2 $RobustRegression(A, b, J)$.

```
Input: $A, b, J$
Output: $x_1, ..., x_M$
$U, \Sigma^{-1}, V, \rho = rank(A) \leftarrow \ SVD(A)$
$x = V\Sigma^{-1}U^Tb$
$k = 0;\ \varepsilon \geq 0$
while (not converge) do
    $k = k + 1$
    $r = b - Ax$
    $y = \ell'(r)$
    $d = \ell''(r)$
    if (k==1) then
        $H = \sum_{j=1}^{J} d_jU_jU_j^T$
    else
        for $j \leftarrow 1$ to $J$ do
            if $(|d_j - d_j^{old}| > \varepsilon)$ then
                $H = H + (d_j - d_j^{old})U_jU_j^T$ // Update according to (3.7)
            end if
        end for
    end if
    $d^{old} = d$
    $g = U^Ty$
    $s \leftarrow CholeskySolver(H, g)$
    $x = x + V\Sigma^{-1}s$
end while
```

Notes. The loss function $\ell(\cdot)$ has to be specified a priori.

Remark 3.2.1. *As mentioned in [30], from a numerical point of view, setting derivatives at discontinuous points should not influence the search procedure for finding the minimum of a sum of almost everywhere differentiable functions as given in (3.1). Anyway, the local convergence as well as the resulting order of Newton-Raphson-based solvers for robust regression might be shown by using techniques of the non-smooth analysis, see [112].*

Making Newton-Raphson's method safe is vital for a successful implementation. As already mentioned, by construction the matrices $H^{(k)}$ in (3.5) and $\overline{H}^{(k)}$ in (3.6) are positive semi-definite as long as the elements of $D^{(k)}$ are non-negative. To assure the regularity of them and, thus, the positive definiteness, we should shift the eigenvalues into a positive range whenever necessary; $\overline{H}^{(k)}$ is better behaved than $H^{(k)}$. Following

[106], an ad hoc modified Cholesky decomposition may be implemented:

$$\eta := \begin{cases} 0 & \text{, if } \min_{i=1,\dots,\rho}(\overline{H}_{ii}^{(k)}) > 0 \\ -\min_{i=1,\dots,\rho}(\overline{H}_{ii}^{(k)}) + \xi & \text{, else} \end{cases} \quad , \ \zeta := 2, \ \xi := 10^{-3},$$

$$\textbf{do}\{\widetilde{H}^{(k)} := \overline{H}^{(k)} + \eta I_\rho; \text{Cholesky}(\widetilde{H}^{(k)}); \ \eta = \max(\zeta\eta, \xi) \ \} \ \textbf{while}(\text{Cholesky failed})$$

where I_ρ and $\overline{H}_{ii}^{(k)}$ denote the $(\rho \times \rho)$ identity matrix and the i-th diagonal element of $\overline{H}^{(k)}$. Notify that the routine *Cholesky*$(\cdot)$ is involved in *SolveCholesky*$(\cdot)$ in Algorithm 2. To make fast progress, ζ might be chosen larger such as $\zeta = 10$; the value of parameter ξ might be modified as well. A sufficient reduction of the objective function (3.1) in each step is desirable, but not necessarily guaranteed. In order to ensure global convergence, a damped Newton-Raphson version might be implemented

$$x^{(k+1)} = x^{(k)} + \lambda_k s^{(k)} \tag{3.8}$$

for some $\lambda_k > 0$. To decide whether the chosen λ_k is good enough, the Armijo condition can be verified

$$f(x^{(k)} + \lambda_k s^{(k)}) \leq f(x^{(k)}) + \kappa\lambda_k \nabla f(x^{(k)})^T s^{(k)}$$

with a sufficiently small number $\kappa > 0$, e.g., $\kappa = 10^{-4}$. After testing the full Newton-Raphson step, i.e. $\lambda_k = 1$, we select a new damped parameter $\lambda_k \in [0.1\lambda_k^{\text{old}}, 0.5\lambda_k^{\text{old}}]$ as suggested in the backtracking procedure in [41]; see [111] for implementation details. It makes sense to terminate the line search procedure close to the optimum as roundoff errors may cause numerical difficulties.

Following the discussion in [41], we use both convergence criteria

$$\max_{m=1,\dots,M} \frac{|x_m^{(k+1)} - x_m^{(k)}|}{\max\{|x_m^{(k+1)}|, 1\}} \leq TOL_{\Delta x}, \quad 0 < TOL_{\Delta x} \ll 1, \tag{3.9}$$

to test for convergence on Δx, and

$$\max_{m=1,\dots,M} \frac{|(\nabla f(x^{(k+1)}))_m \max\{|x_m^{(k+1)}|, 1\}|}{\max\{f(x^{(k+1)}), 1\}} \leq TOL_{\nabla f}, \quad 0 < TOL_{\nabla f} \ll 1, \tag{3.10}$$

to test for convergence on zero gradient; $(\nabla f(x^{(k+1)}))_m$ is the m-th entry of $\nabla f(x^{(k+1)})$. To highlight our proposed approach, denoted by RR_{New} in the following, we give a brief overview of alternative robust regression solvers in Remark 3.2.2; a similar overview is given in [5], but we only focus on solvers promising good numerical properties.

Remark 3.2.2. *An elegant way to find the descent direction $s^{(k)}$ in (3.3) is the solution of the least squares problem*

$$s^{(k)} = \underset{\widetilde{s} \in \mathbb{R}^M}{\operatorname{argmin}} \|\widetilde{A}\widetilde{s} - \widetilde{b}\|_2^2$$

with $\widetilde{A} = (\widehat{D}^{(k)})^{1/2}A$, $\widetilde{b} = (\widehat{D}^{(k)})^{-1/2}y^{(k)}$, and $\widehat{D}_j^{(k)} = \max\{D_j^{(k)}, \eta\}$, $j = 1, ..., J$, are modified elements of the diagonal matrix $D^{(k)}$ to avoid zero elements; η is a sufficiently small number, see [5]; we denote this approach by $\widetilde{\mathrm{RR}}_{\mathrm{New}}$ to refer to it later on. Replacing the elements of the diagonal matrix in (3.5) by the secant approximation

$$\ell''(r_j) \approx \frac{\ell'(r_j) - \ell'(0)}{r_j - 0} = \frac{\ell'(r_j)}{r_j} =: \widetilde{D}_j, \; j = 1, ..., J,$$

leads to the iteration step

$$x^{(k+1)} = (A^T \widetilde{D}^{(k)} A)^{-1} A^T \widetilde{D}^{(k)} r^{(k)}, \; k = 0, 1, 2, ... \tag{3.11}$$

As (3.11) is the solution of a weighted least squares problem, this approach is known as iteratively re-weighted least squares (IRLS). This might be seen as a comfortable way for implementing because a number of programming packages deliver stable solvers for (weighted) least squares problems. Many objective functions fulfill $\widetilde{D}_j \geq \max\{0, \ell''(r_j)\}$ for all $j = 1, ..., J$, and, generally, $\ell'(0) = 0$. Due to its simplicity and global convergence, IRLS is the common way to solve robust regression problems. In contrast to Newton-Raphson-based solvers, it is just linear convergent, see [13]. Following the approach proposed by O'Leary in [108], a residual instead of a coefficient iteration in combination with the QR decomposition leads to stability advantages. Finding the minimum residual r^* leads to the iteration

$$r^{(k+1)} = r^{(k)} - As^{(k)}, \; k = 0, 1, ...,$$

and, finally, the coefficients can be determined by solving

$$A\widehat{x} = b - \widehat{r},$$

where $\widehat{r}$ is the approximated optimal residual. Anyway, a residual iteration might become expensive if $M \ll J$.

3.3 Convergence

In order to proof convergence of our RRM estimator, we use techniques of the statistical learning theory. Similar ideas are used in [46] and [132] for a convergence proof of the LSM method. Before we start, let us complete our mathematical framework by some notations. So far, we have restricted ourselves on finite-dimensional linear approximation sets. According to [132], we want to drop this limiting assumption, and, thus, we consider arbitrary subsets $\mathcal{H}_l$ of L_2. To refer to the minimizer of optimization task (3.1), we introduce the function $\widehat{C}_{\mathcal{H}_l,l}$ to be any function $h_l \in \mathcal{H}_l$ satisfying

$$\frac{1}{N}\sum_{n=1}^{N}\ell\left(C_l^n - \widehat{C}_{\mathcal{H}_l,l}(\mathbb{S}_{nl})\right) \leq \inf_{h_l\in\mathcal{H}_l}\frac{1}{N}\sum_{n=1}^{N}\ell\left(C_l^n - h_l(\mathbb{S}_{nl})\right) + \widetilde{\varepsilon}.$$

with $\widetilde{\varepsilon} \geq 0$. From a numerical point of view, we have $\widetilde{\varepsilon} > 0$ if errors caused by stability problems, convergence criteria, or otherwise, influence the output produced by Algorithm 2. We may set $\widetilde{\varepsilon} = 0$ if a minimizer exists and may be computed exactly. Actually, we work with the DPP in terms of the approximated optimal stopping time,

$$\tau_L = L \tag{3.12}$$

$$\tau_l = \begin{cases} l & , Z_l \geq \widehat{C}_{\mathcal{H}_l,l} \\ \tau_{l+1} & , \text{otherwise} \end{cases} \quad , l = L-1,...,0, \tag{3.13}$$

and we define the continuation value at $t_l,\ l = L-1,...,0$, w.r.t. (3.12)-(3.13) by

$$C_l^* = \mathbb{E}_l[e^{-r(\tau_{l+1}-l)\triangle t}Z_{\tau_{l+1}}]$$

– C_l^* is often called regression function in a statistical learning framework. Moreover, for any $l = 0,...,L-1$, P_l denotes the image probability measure on $\Omega_l = E \times \mathbb{R}$ jointly induced by the variables $(S_l, e^{-r(\tau_{l+1}-l)\triangle t}Z_{\tau_{l+1}})$. The entire collection of variables $(S_k, e^{-r(\tau_{k+1}-l)\triangle t}Z_{\tau_{k+1}}),\ k = l,...,L-1$, jointly induces an image probability measure on $\Omega_l \times ... \times \Omega_{L-1} =: \mathbf{\Omega}_l$, and we give it the notation $\mathbf{P}_l$. Due to the simulation of N paths sampled independently according to $\mathbf{P}_l$, we consider the product measure $\mathbf{P}_l^1 \otimes ... \otimes \mathbf{P}_l^N$ on $\mathbf{\Omega}_l \times ... \times \mathbf{\Omega}_{L-1} =: \mathbf{\Omega}_l^N$. For notational purposes notify that

$$\boldsymbol{\omega} := \boldsymbol{\omega}^{lN} = ((\mathbb{S}_{1l}, C_l^1), ..., (\mathbb{S}_{Nl}, C_l^N)) \in \Omega^N = \Omega_l^N$$

for a fixed number $N \in \mathbb{N}$ and any $l = 0,...,L-1$, and

$$\boldsymbol{\omega}_{ln} := ((\mathbb{S}_{nl}, C_l^n), ..., (\mathbb{S}_{n,L-1}, C_{L-1}^n)) \in \mathbf{\Omega}_l$$

for any $l = 0,...,L-1,\ n = 1,...,N$. The ordered (L-l)-tuple $(\boldsymbol{\omega}^{lN}, ..., \boldsymbol{\omega}^{L-1,N})$ as well as the ordered N-tuple $(\boldsymbol{\omega}_{l1}, ..., \boldsymbol{\omega}_{lN})$ are generic sample elements from the probability space $\mathbf{\Omega}_l^N$. In the following we assume that F and H are positive integers with

$$F = \max\{1, \|Z_1\|_\infty, ..., \|Z_L\|_\infty\} < \infty$$

and

$$\sup\{\|h_l\|_\infty | h_l \in \mathcal{H}_l\} \leq H < \infty,$$

respectively. Equipped with these definitions, we are able to prove the convergence of our RRM estimator in Subsection 3.3.2 based on a error decomposition derived in Subsection 3.3.1.

3.3.1 Error Decomposition

To begin with, let us introduce some notations in the manner of the statistical learning theory, and we refer to [129] or [38] for a deeper look into this field. Let E be a compact domain or a manifold in Euclidean space and $Y = \mathbb{R}$. We denote by ρ the Borel probability measure on $\Omega = E \times Y$, whose regularity properties are assumed as needed. Then, in general, we define the error of a suitably measureable function $f : E \to \mathbb{R}$ by

$$\mathcal{E}(f) = \int_{\Omega} \ell(c - f(s)) d\rho,$$

where $\omega = (s, c)$ are the elements of Ω. To be consistent with our framework, for any function $h_l \in \mathcal{H}_l$ we write

$$\mathcal{E}_l(h_l) := \int \ell(C_l - h_l(S_l)) := \int_E \ell(C_l - h_l(S_l)) dP_{S_l}. \tag{3.14}$$

Moreover, a second error of any $h_l \in \mathcal{H}_l$ is given by

$$\mathcal{E}_l^*(h_l) := \int \ell(C_l^* - h_l(S_l)) := \int_E \ell(C_l^* - h_l(S_l)) dP_{S_l}. \tag{3.15}$$

The second equalities of (3.14) and (3.15) are for reasons of notational convenience, and we use these short notations whenever there is no confusion. According to quantity (3.15), we define the empirical errors of any $h_l \in \mathcal{H}_l$ (w.r.t. $\boldsymbol{\omega}$) by

$$\widehat{\mathcal{E}}_l^*(h_l) = \frac{1}{N} \sum_{n=1}^{N} \ell(C_l^n - h_l(\mathbb{S}_{nl})).$$

A further helpful definition is the projection of C_l^* on $\mathcal{H}_l$ given by

$$\overline{C}_{\mathcal{H}_l,l} := \arg \inf_{h_l \in \mathcal{H}_l} \mathcal{E}_l^*(h_l).$$

The existence of $\overline{C}_{\mathcal{H}_l,l}$ follows from the compactness of $\mathcal{H}_l$ and the continuity of the error. By the way, quantity $\inf_{h_l \in \mathcal{H}_l} \int \ell(C_l^* - h_l(S_l))$ is often called approximation error. Let t_l be any time date fixed for $l = 0, ..., L-1$. Then, we define a generalized version of the triangle inequality by

$$\begin{aligned} \int \ell(C_l - \widehat{C}_{\mathcal{H}_l,l}) &= \int \ell(C_l - C_l^* + C_l^* - \widehat{C}_{\mathcal{H}_l,l}) \\ &\leq O\left(\int \ell(C_l - C_l^*) + \int \ell\left(C_l^* - \widehat{C}_{\mathcal{H}_l,l}\right) \right), \end{aligned} \tag{3.16}$$

and, by choosing an appropriate constant $\gamma \geq 1$, we get the γ-triangle inequality

$$\mathcal{E}_l(\widehat{C}_l) \leq \gamma(\mathcal{E}_l(C_l^*) + \mathcal{E}_l^*(\widehat{C}_{\mathcal{H}_l,l})). \tag{3.17}$$

Remark 3.3.1. *The triangle constant γ in (3.17) depends on the loss function. For example, by Jensen's inequality and the convexity of $r \mapsto r^2$, one can show that $\gamma = 2$ for the squared loss function, see [36].*

Clearly, considering the last time step before maturity t_{L-1} leads to

$$\mathcal{E}^*_{L-1}(\widehat{C}_{\mathcal{H}_{L-1},L-1}) \;=\; \mathcal{E}_{L-1}(\widehat{C}_{\mathcal{H}_{L-1},L-1}). \tag{3.18}$$

However, we are interested in an error decomposition of $\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l})$ at any time date $t_l,\ l = L-1,...,1$. Remember that regression-based Monte Carlo methods are based on the DPP. Thus, on account of taking heed of propagation errors, we obtain the following vital inequalities for $l = L-1,...,0$:

Theorem 3.3.1. *Assume that the mathematical framework above holds. Then, at all $t_l,\ l = L-1,...,1$,*

$$\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l}) \;\le\; \sum_{k=l}^{L-1}\widetilde{\gamma}^{k-l}\gamma^{2(k-l)}(1_{\{k\neq l\}}+\gamma)\mathcal{E}^*_k(\widehat{C}_{\mathcal{H}_k,k}) \tag{3.19}$$

with $\widetilde{\gamma} > 0$. Moreover, at $t_0 = 0$ we have

$$(\widehat{V}_0 - V_0)^2 \;\le\; 2(\widehat{C}_0 - C^*_0)^2 + 2\sum_{k=1}^{L-1}\widetilde{\gamma}^k\gamma^{2k-1}(1+\gamma)\mathcal{E}^*_k(\widehat{C}_{\mathcal{H}_k,k}). \tag{3.20}$$

Proof. At first, at any time date $t_l,\ l = L-1,...,1$, consider

$$\begin{aligned}\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l}) \;&=\; \int \ell(C_l - \widehat{C}_{\mathcal{H}_l,l}) \\ &\le\; \gamma\left(\int \ell(C_l - C^*_l) + \int \ell(C^*_l - \widehat{C}_{\mathcal{H}_l,l})\right).\end{aligned} \tag{3.21}$$

In analogy to [132], we are able to show by reverse induction that

$$\mathcal{E}_l(C^*_l) \le \sum_{k=l+1}^{L-1}\widetilde{\gamma}^{k-l}\gamma^{2(k-l)-1}(1+\gamma)\mathcal{E}^*_k(\widehat{C}_{\mathcal{H}_k,k}),\ l = L-2,...,1, \tag{3.22}$$

holds for the first term on the right-hand side of (3.21). Obviously, inequality (3.22) will be clear at t_{L-2} with the same argumentation as follows, but with $\tau_L = \tau^*_L = L$. In general, inequality (3.22) is an immediate consequence of Jensen's inequality and γ-triangle inequality (3.17). To see this, we assume that inequality

$$\mathcal{E}_{l_0}(C^*_{l_0}) \;\le\; \sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-l_0}\gamma^{2(k-l_0)-1}(1+\gamma)\mathcal{E}^*_k(\widehat{C}_{\mathcal{H}_k,k}) \tag{3.23}$$

holds at time t_{l_0} for any l_0 fulfilling $L-1 > l_0 \geq 2$. Then, applying Jensen's inequality for convex functions leads to the reformulation

$$\begin{aligned}
\mathcal{E}_{l_0-1}(C^*_{l_0-1}) &= \int \ell(\mathbb{E}[e^{-r(\tau^*_{l_0}-(l_0-1))\triangle t} Z_{\tau^*_{l_0}}|S_{l_0-1}] - \mathbb{E}[e^{-r(\tau_{l_0}-(l_0-1))\triangle t} Z_{\tau_{l_0}}|S_{l_0-1}]) \\
&= \int \ell(\mathbb{E}[\mathbb{E}[e^{-r(\tau^*_{l_0}-(l_0-1))\triangle t} Z_{\tau^*_{l_0}}|S_{l_0}] - \mathbb{E}[e^{-r(\tau_{l_0}-(l_0-1))\triangle t} Z_{\tau_{l_0}}|S_{l_0}]|S_{l_0-1}]) dP_{S_{l_0-1}} \\
&\leq \widetilde{\gamma}_{l_0} \int \ell(\mathbb{E}[e^{-r(\tau^*_{l_0}-(l_0-1))\triangle t} Z_{\tau^*_{l_0}}|S_{l_0}] - \mathbb{E}[e^{-r(\tau_{l_0}-(l_0-1))\triangle t} Z_{\tau_{l_0}}|S_{l_0}]) dP_{S_{l_0}}. \quad (3.24)
\end{aligned}$$

Notify that we make use of assumption (L3). If our chosen loss function is just piecewise convex, we apply Jensen's inequality on subintervals; in this case the transition points define our decomposition. To be more flexible with respect to the chosen loss function at any time date t_{l_0-1}, we introduce a constant $\widetilde{\gamma}_{l_0} > 0$; $\widetilde{\gamma}_{l_0} = 1$ if we work with the same loss function as in the previous time step t_{l_0}. For simplicity, we define $\widetilde{\gamma} := \max_{k=l_0,\ldots,L-1} \widetilde{\gamma}_k$. Consider the following three events:

$$\begin{aligned}
A_1 &:= (\{\tau^*_{l_0} = l_0\} \cap \{\tau_{l_0} = \tau_{l_0+1}\}) = \{Z_{l_0} \geq \mathbb{E}[e^{-r(\tau^*_{l_0+1}-l_0)\triangle t} Z_{\tau^*_{l_0+1}}|S_{l_0}], Z_{l_0} < \widehat{C}_{\mathcal{H}_{l_0},l_0}\}, \\
A_2 &:= (\{\tau^*_{l_0} = \tau^*_{l_0+1}\} \cap \{\tau_{l_0} = l_0\}) = \{Z_{l_0} < \mathbb{E}[e^{-r(\tau^*_{l_0+1}-l_0)\triangle t} Z_{\tau^*_{l_0+1}}|S_{l_0}], Z_{l_0} \geq \widehat{C}_{\mathcal{H}_{l_0},l_0}\}, \\
A_3 &:= (\{\tau^*_{l_0} = \tau^*_{l_0+1}\} \cap \{\tau_{l_0} = \tau_{l_0+1}\}) = \{Z_{l_0} < \mathbb{E}[e^{-r(\tau^*_{l_0+1}-l_0)\triangle t} Z_{\tau^*_{l_0+1}}|S_{l_0}], Z_{l_0} < \widehat{C}_{\mathcal{H}_{l_0},l_0}\}.
\end{aligned}$$

Let us assume that the event A_1 occurs. Then,

$$\begin{aligned}
&\widetilde{\gamma} \int \ell(\mathbb{E}[e^{-r(\tau^*_{l_0}-(l_0-1))\triangle t} Z_{\tau^*_{l_0}}|S_{l_0}] - \mathbb{E}[e^{-r(\tau_{l_0}-(l_0-1))\triangle t} Z_{\tau_{l_0}}|S_{l_0}]) \\
&\leq \widetilde{\gamma}\gamma \left(\int \ell(C_{l_0} - \widehat{C}_{\mathcal{H}_{l_0},l_0}) + \int \ell(\widehat{C}_{\mathcal{H}_{l_0},l_0} - C^*_{l_0}) \right). \quad (3.25)
\end{aligned}$$

Obviously, the right-hand side of (3.25) is an upper bound for all three cases. All in all, by exploring that our chosen loss function is even, we obtain

$$\mathcal{E}_{l_0-1}(C^*_{l_0-1}) \leq \widetilde{\gamma}(\gamma^2 \mathcal{E}_{l_0}(C^*_{l_0}) + \gamma^2 \mathcal{E}^*_{l_0}(\widehat{C}_{\mathcal{H}_{l_0},l_0}) + \gamma \mathcal{E}^*_{l_0}(\widehat{C}_{\mathcal{H}_{l_0},l_0})). \quad (3.26)$$

If we work with an uneven loss function, we should work on with a modified even loss function $\widetilde{\ell}(\cdot)$ such that $\ell(\cdot) \leq \widetilde{\ell}(\cdot)$. Finally, by induction hypothesis (3.23) applied to the first term on the right-hand side of (3.26), we get (3.22), and (3.22) combined with (3.21) leads to (3.19). Following the argumentation above, we have

$$(C^*_0 - C_0)^2 \leq \widetilde{\gamma}_1(\gamma^2 \mathcal{E}_1(C^*_1) + \gamma^2 \mathcal{E}^*_1(\widehat{C}_{\mathcal{H}_1,1}) + \gamma \mathcal{E}^*_1(\widehat{C}_{\mathcal{H}_1,1})) \quad (3.27)$$

at t_0. Therefore, inequality (3.20) holds at $t_0 = 0$ due to

$$(\max(Z_0, \widehat{C}_0) - \max(Z_0, C_0))^2 \leq (\widehat{C}_0 - C_0)^2 \leq 2(\widehat{C}_0 - C^*_0)^2 + 2(C^*_0 - C_0)^2 \quad (3.28)$$

combined with (3.22) and (3.27).

□

3.3.2 Error Estimates

In the manner of the statistical learning theory, the defect function of a function f is defined by

$$\widehat{L}(f) := \widehat{\mathcal{E}}(f) - \mathcal{E}(f)$$

with the empirical error $\widehat{\mathcal{E}}(f)$ of f as above. Bounds on $\widehat{L}(f)$ seem to be helpful to bound the actual error from an observed quantity. A well-known uniform estimate on the defect is Pollard's inequality, which is often used to give bounds on the sample error, see [38]. Before we proceed, let us define the l_1-covering number.

Definition 3.3.1 ([43])**.** *Let A be a bounded subset of $\mathbb{R}^d$. Then, for every $\varepsilon > 0$, the l_1-covering number $\mathcal{N}(\varepsilon, A)$ is defined as the cardinality of the smallest finite set in $\mathbb{R}^d$ such that for every $z \in A$ there is a point $y \in \mathbb{R}^d$ in the finite set such that $1/d\|z - y\|_1 < \varepsilon$, where $\|x\|_1 := \sum_{i=1}^{d} |x^{(i)}|$ is the l_1-norm of the vector $x = (x^{(1)}, ..., x^{(d)}) \in \mathbb{R}^d$. So, $\mathcal{N}(\varepsilon, A)$ is the smallest number of l_1-balls of radius εd, whose union contains A.*

Let $\mathcal{G}$ be the class of real-valued functions defined on $\mathbb{R}^d$ such that for any $g \in \mathcal{G}$, $0 \leq g(x) \leq G$ for all $x \in \mathbb{R}^d$ and some $G < \infty$. If we define the set $\mathcal{G}(x) := \{(g(x_1), ..., g(x_N)); g \in \mathcal{G}\} \subset \mathbb{R}^N$ for N fixed points x_i, $i = 1, ..., N$, in $\mathbb{R}^d$, $x = (x_1, ..., x_N)$, $\mathcal{N}(\varepsilon, \mathcal{G}(x))$ denotes the l_1-covering number of $\mathcal{G}(x)$.

Theorem 3.3.2 (Pollard)**.** *Let $X_1, ..., X_N$ and X be i.i.d. random variables with $\boldsymbol{X} = (X_1, ..., X_N)$. For any $N \in \mathbb{N}$ and $\varepsilon > 0$,*

$$P\left(\sup_{g\in\mathcal{G}}\left|\frac{1}{N}\sum_{n=1}^{N} g(X_n) - \mathbb{E}[g(X)]\right| > \varepsilon\right) \leq \delta, \tag{3.29}$$

where the probability δ is defined by

$$\delta := 8\mathbb{E}\left[\mathcal{N}\left(\frac{\varepsilon}{8}, \mathcal{G}(\mathbf{X})\right)\right] \exp\left(-\frac{N\varepsilon^2}{128G^2}\right).$$

Proof. See [110] or [43]. □

Notify that $\mathcal{N}(\varepsilon, \mathcal{G}(\mathbf{X}))$ is a random variable whose expected value plays a decisive role in Pollard's theorem.

Remark 3.3.2. *Usually, Pollard's theorem does not give sharp bounds, and, that is why, we sometimes use sharper inequalities. Nevertheless, for our purposes it is sufficient to work with it.*

A further important definition in the statistical learning theory is the Vapnik-Chervonenkis (VC) dimension given in Definition 3.3.2 according to [46].

Definition 3.3.2 (VC Dimension). *Let $\mathcal{G}$ be a set of real-valued functions defined on some set $B \subseteq \mathbb{R}^d$. A set of points $\{x_1, ..., x_n\} \subset B$ is said to be shattered by $\mathcal{G}$ if there exists $r \in \mathbb{R}^n$ such that, for every $b \in \{0,1\}^n$, there is a function $g \in \mathcal{G}$ such that for each i, $g(x_i) > r_i$ if $b_i = 1$, and $g(x_i) \leq r_i$ if $b_i = 0$. Then, the VC dimension of $\mathcal{G}$ denoted by $vc(\mathcal{G})$ is defined as the cardinality of the largest set of points which can be shattered by $\mathcal{G}$.*

Let us include Theorem 3.3.2 in our framework. To do so, $\mathcal{G}$ denotes the set of functions g for which $g(\omega) = \ell(c - h_l(s))$, $h_l \in \mathcal{H}_l$, $\omega = (s, c)$, and $X = (S_l, e^{-r(\tau_{l+1}-l)\Delta t} Z_{\tau_{l+1}})$. Now, we are able to show a vital result for proofing convergence of our RRM estimator in Lemma 3.3.1.

Lemma 3.3.1. *Suppose that Pollard's inequality holds with the previous settings at all time dates t_l, $l = L-1, ..., 1$. Then, for any $\widetilde{\varepsilon} \geq 0$,*

$$\mathcal{P}_{\boldsymbol{\omega}^{lN} \in \Omega_l^N} \left(\mathcal{E}_l^*(\widehat{C}_{\mathcal{H}_l,l}) \leq 3\varepsilon + \mathcal{E}_l^*(\overline{C}_{\mathcal{H}_l,l}) \right) \geq 1 - \widetilde{\delta}_l, \quad l = L-1, ..., 1, \tag{3.30}$$

with $\widetilde{\varepsilon} \leq \varepsilon$, and

$$\widetilde{\delta}_l = 22(vc(\mathcal{H}_l) + 1) \left(\frac{348H(F+H)}{\varepsilon} \right)^{vc(\mathcal{H}_l)} \exp\left(-\frac{N\varepsilon^2}{128G_l^2} \right). \tag{3.31}$$

Proof. By Pollard's inequality (3.29), for any $h_l \in \mathcal{H}_l$ and $\varepsilon \geq 0$ we obtain that

$$\widehat{\mathcal{E}}_l^*(h_l) \leq \mathcal{E}_l^*(h_l) + \varepsilon$$

with probability at least

$$1 - 8\mathbb{E}\left[\mathcal{N}\left(\frac{\epsilon}{8}, \mathcal{G}_l(\boldsymbol{\omega}) \right) \right] \exp\left(-\frac{N\epsilon^2}{128G_l^2} \right) \tag{3.32}$$

over $\boldsymbol{\omega}^{lN} \in \Omega_l^N$. It is important to note that quantity G_l depends on the loss function itself, e.g. $G_l = (F+H)^2$ for the squared loss function, $G_l = \min\{(F+H)^2, 2\gamma_1(F+H) - \gamma_1^2\}$ for Huber's loss function, $G_l = \min\{(F+H)^2, \gamma_2^2\}$ for Talwar's function. We denote by $\widetilde{\mathcal{G}}_l$ the set of functions $\widetilde{g}$ for which $\widetilde{g}(\omega) = c - h_l(s)$ with $h_l \in \mathcal{H}_l$, $\omega = (s, c)$. Moreover, let $\widetilde{\mathcal{G}}_l^1$ be the set of functions g_1 for which $g_1(\omega) = h_l(s)$ and $\widetilde{\mathcal{G}}_l^2$ be the set of functions g_2 for which $g_2(\omega) = c$, $\omega = (s, c)$. Then, in analogy to [132], by definition of the l_1-covering number we obtain

$$\begin{aligned} \mathcal{N}\left(\frac{\epsilon}{8}, \mathcal{G}_l(\boldsymbol{\omega}) \right) &\leq \mathcal{N}\left(\frac{\epsilon}{16(F+H)}, \widetilde{\mathcal{G}}_l(\boldsymbol{\omega}) \right) \\ &\leq \mathcal{N}\left(\frac{\epsilon}{32(F+H)}, \widetilde{\mathcal{G}}_l^1(\boldsymbol{\omega}) \right) \mathcal{N}\left(\frac{\epsilon}{32(F+H)}, \widetilde{\mathcal{G}}_l^2(\boldsymbol{\omega}) \right). \end{aligned} \tag{3.33}$$

Applying Haussler's inequality to the first term of (3.33) leads to

$$\mathcal{N}\left(\frac{\epsilon}{32(F+H)}, \widetilde{\mathcal{G}}_l^1(\boldsymbol{\omega}) \right) \leq e(vc(\widetilde{\mathcal{G}}_l^1) + 1) \left(\frac{128eH(F+H)}{\varepsilon} \right)^{vc(\widetilde{\mathcal{G}}_l^1)}, \tag{3.34}$$

see [65] or [46] for Haussler's inequality. All in all, as the second term of (3.33) is equal to 1, quantity (3.32) can be estimated by

$$\begin{aligned}&8\mathbb{E}\left[\mathcal{N}\left(\frac{\epsilon}{8}, \mathcal{G}_l(\boldsymbol{\omega})\right)\right] \exp\left(-\frac{N\epsilon^2}{128G_l^2}\right)\\ &\leq 22(vc(\mathcal{H}_l)+1)\left(\frac{348H(F+H)}{\varepsilon}\right)^{vc(\mathcal{H}_l)} \exp\left(-\frac{N\varepsilon^2}{128G_l^2}\right) =: \widetilde{\delta}_l.\end{aligned}$$

Thus, with probability at least $1-\widetilde{\delta}_l$ we yield that

$$\mathcal{E}_l^*(\widehat{C}_{\mathcal{H}_l,l}) \leq \widehat{\mathcal{E}}_l^*(\widehat{C}_{\mathcal{H}_l,l}) + \varepsilon \tag{3.35}$$

and

$$\widehat{\mathcal{E}}_l^*(\overline{C}_{\mathcal{H}_l,l}) \leq \mathcal{E}_l^*(\overline{C}_{\mathcal{H}_l,l}) + \varepsilon \tag{3.36}$$

over $\boldsymbol{\omega}^{lN} \in \Omega_l^N$. As $\widehat{C}_{\mathcal{H}_l,l}$ minimizes $\widehat{\mathcal{E}}_l^*$ on $\mathcal{H}_l$, we have

$$\widehat{\mathcal{E}}_l^*(\widehat{C}_{\mathcal{H}_l,l}) \leq \widehat{\mathcal{E}}_l^*(\overline{C}_{\mathcal{H}_l,l}) + \widetilde{\varepsilon}. \tag{3.37}$$

Finally, bringing inequalities (3.35)-(3.37) together leads to the statement that

$$\mathcal{E}_l^*(\widehat{C}_{\mathcal{H}_l,l}) \leq \widehat{\mathcal{E}}_l^*(\widehat{C}_{\mathcal{H}_l,l}) + \varepsilon \leq \widehat{\mathcal{E}}_l^*(\overline{C}_{\mathcal{H}_l,l}) + \varepsilon + \widetilde{\varepsilon} \leq \mathcal{E}_l^*(\overline{C}_{\mathcal{H}_l,l}) + 2\varepsilon + \widetilde{\varepsilon} \tag{3.38}$$

with probability at least $1-\widetilde{\delta}_l$. Immediately, claim (3.30) follows from (3.38) with $\widetilde{\varepsilon} \leq \varepsilon$ over $\boldsymbol{\omega}^{lN} \in \Omega_l^N$. □

Now, we are in the situation to justify results regarding convergence of our RRM estimator:

Theorem 3.3.3 (Convergence of RRM estimator). *Suppose that all assumptions and definitions of the complete framework are maintained. Furthermore, assume that $vc(\mathcal{H}_l) \leq q < \infty$ for $l = L-1, ..., 1$. Then, for any $\varepsilon > 0$ and at each time date t_l, $l = L-1, ..., 1$,*

$$\mathcal{P}_{(\boldsymbol{\omega}_{l1},\ldots,\boldsymbol{\omega}_{lN})\in\boldsymbol{\Omega}_l^N}\left(\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l}) \leq \sum_{k=l}^{L-1}\widetilde{\gamma}^{k-l}\gamma^{2(k-l)}(1_{\{k\neq l\}}+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))\right) \geq 1-\delta_l \tag{3.39}$$

and

$$\mathcal{P}_{(\boldsymbol{\omega}_{l1},\ldots,\boldsymbol{\omega}_{lN})\in\boldsymbol{\Omega}_l^N}\left(\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l}) \leq \sum_{k=l}^{L-1}\widetilde{\gamma}^{k-l}(2\gamma^2+\gamma^3)^{k-l}(1+\gamma)(3\varepsilon+\gamma\Gamma^*(l))\right) \geq 1-\delta_l \tag{3.40}$$

with

$$\delta_l := \sum_{k=l}^{L-1} 22(q+1)\left(\frac{348H(F+H)}{\varepsilon}\right)^q \exp\left(-\frac{N\epsilon^2}{128G_k^2}\right)$$

and $\Gamma(l) := \max_{k=l,...,L-1} \mathcal{E}_k(\overline{C}_{\mathcal{H}_k,k})$, *where* $\Gamma(0) = \max_{k=1,...,L-1} \mathcal{E}_k(\overline{C}_{\mathcal{H}_k,k})$. *Especially, at* t_0 *we have*

$$\mathcal{P}_{(\boldsymbol{\omega}_{01},...,\boldsymbol{\omega}_{0N})\in\boldsymbol{\Omega}_0^N}\left((\widehat{V}_0 - V_0)^2 \leq 2\varepsilon^2 + 2\sum_{k=1}^{L-1}\widetilde{\gamma}^k\gamma^{2k-1}(1+\gamma)(3\varepsilon + \mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))\right) \geq 1 - \delta_0 \tag{3.41}$$

and

$$\mathcal{P}_{(\boldsymbol{\omega}_{01},...,\boldsymbol{\omega}_{0N})\in\boldsymbol{\Omega}_0^N}\left((\widehat{V}_0 - V_0)^2 \leq 2\varepsilon^2 + 2\sum_{k=1}^{L-1}\widetilde{\gamma}^k\gamma(2\gamma^2+\gamma^3)^{k-1}(1+\gamma)(3\varepsilon + \gamma\Gamma(0))\right) \geq 1 - \delta_0 \tag{3.42}$$

with

$$\delta_0 := \delta_1 + 8\exp\left(-\frac{N\varepsilon^4}{128F^2}\right)$$

Proof. Consider the intersection of $(L-l)$ events given by Lemma 3.1 at each time date t_l, $l = L-1,...,1$. Then, combining (3.30) and (3.19) leads to the statement that with probability at least $1-\delta_l$, $\delta_l := \sum_{k=l}^{L-1}\widetilde{\delta}_k$,

$$\mathcal{E}_l(\widehat{C}_{\mathcal{H}_l,l}) \leq \sum_{k=l}^{L-1}\widetilde{\gamma}^{k-l}\gamma^{2(k-l)}(1_{\{k\neq l\}} + \gamma)(3\varepsilon + \mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k})) \tag{3.43}$$

over all $(\boldsymbol{\omega}^{lN},...,\boldsymbol{\omega}^{L-1,N}) \in \boldsymbol{\Omega}_l^N$ at any time date t_l, $l = L-1,...,1$. For notational convenience we define $\mathcal{H}_0$ by the singleton set $\{\mathbb{E}[e^{-r\triangle t_{\tau_1}}Z_{\tau_1}|S_0]\}$. Using Pollard's theorem, for any $\varepsilon > 0$, we have

$$\mathcal{P}_{\boldsymbol{\omega}^{0N}\in\Omega_0^N}((\widehat{C}_0 - C_0^*)^2 \leq \varepsilon^2) \geq 1 - 8\exp\left(-\frac{N\varepsilon^4}{128F^2}\right). \tag{3.44}$$

Thus, at $t_0 = 0$ we obtain statement (3.41) by combining the intersection of (L-1) events given by Lemma 3.1 with the event given in (3.44) and taking (3.20) into account. In order to show statement (3.40) , we use (reverse) induction. To begin with, (3.40) is clear for $l_0 = L-1$. Let t_{l_0} be any time date with $0 < l_0 \leq L-2$, and assume that

$$\begin{aligned}&\sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-(l_0+1)}\gamma^{2(k-(l_0+1))}(1+\gamma)(3\varepsilon + \mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))\\ &\leq \sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-(l_0+1)}(2\gamma^2+\gamma^3)^{k-(l_0+1)}(1+\gamma)(3\varepsilon + \gamma\Gamma(l_0+1))\end{aligned} \tag{3.45}$$

holds at t_{l_0+1}. Note that the left-hand side of this inequality is the upper bound of (3.43).

At time date t_{l_0} we have

$$\sum_{k=l_0}^{L-1}\widetilde{\gamma}^{k-l_0}\gamma^{2(k-l_0)}(1+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k})) \tag{3.46}$$

$$= (1+\gamma)3\varepsilon+(1+\gamma)\mathcal{E}_{l_0}^*(\overline{C}_{\mathcal{H}_{l_0},l_0})+\sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-l_0}\gamma^{2(k-l_0)}(1+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k})).$$

By applying triangle inequality (3.17) and taking (3.22) and (3.30) into account, we yield for the second term on the right-hand side

$$\mathcal{E}_{l_0}^*(\overline{C}_{\mathcal{H}_{l_0},l_0}) \leq \gamma\left(\mathcal{E}_{l_0}(\overline{C}_{\mathcal{H}_{l_0},l_0})+\gamma\widetilde{\gamma}\sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-(l_0+1)}\gamma^{2(k-(l_0+1))}(1+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))\right). \tag{3.47}$$

All in all, (3.46) can be estimated by

$$\sum_{k=l_0}^{L-1}\widetilde{\gamma}^{k-l_0}\gamma^{2(k-l_0)}(1+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))$$

$$\leq (1+\gamma)(3\varepsilon+\gamma\mathcal{E}_{l_0}(\overline{C}_{\mathcal{H}_{l_0},l_0}))+\left(\sum_{k=l_0+1}^{L-1}\widetilde{\gamma}^{k-(l_0+1)}\gamma^{2(k-(l_0+1))}(1+\gamma)(3\varepsilon+\mathcal{E}_k^*(\overline{C}_{\mathcal{H}_k,k}))\right)\widetilde{\gamma}\gamma^2(2+\gamma),$$

and using the above definition of $\Gamma(l)$ as well as induction hypothesis (3.45) prove statement (3.40). Thus, at $t_0 = 0$ statement (3.42) directly follows from (3.41). □

Remark 3.3.3. *In accordance with the results in [132], our statements involve an exponential dependence on the number of time steps. Focusing on loss functions fulfilling $\ell(\cdot) \leq (\cdot)^2$ – as the loss functions in Table 3.1 – shows that our RRM estimator is superior to the LSM estimator; provided that the transition points are carefully chosen. Moreover, it makes sense to cut the chosen loss function above to get sharper convergence statements.*

A number of numerical experiments have shown that an approximation of the continuation value by polynomials performs well. To focus on this linear finite-dimensional approximation architecture, we denote by $\Pi_{\widetilde{M}}^D$ the space of all multivariate polynomials, i.e. polynomials in D variables $s_1, ..., s_D$, of total degree $\leq \widetilde{M}-1$ with real coefficients. Therefore, our model function for the continuation value at every time date $t_l,\ l = 1, ..., L-1$, is assumed to be a polynomial $\pi(s) = \sum_\alpha x_\alpha s^\alpha,\ s^\alpha = s_1^{\alpha_1}\cdots s_D^{\alpha_D},\ \alpha_i \in \mathbb{N}_0,\ i = 1, ..., D$. Let $\overline{\mathcal{Q}}_D(\theta)$ be a closed cube of side length $0 \leq 2\theta < \infty$ in $\mathbb{R}^D$ centered at the origin; the interior is denoted by $\mathcal{Q}_D(\theta)$. Then, $\Pi_{\widetilde{M}}^D(\overline{\mathcal{Q}}_D(\theta))$ is the space of all multivariate polynomials on $\overline{\mathcal{Q}}_D(\theta)$. Moreover, we denote by $\mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))$ the space of all continuous real-valued functions f on $\overline{\mathcal{Q}}_D(\theta)$ whose continuous classical derivatives $\partial^\alpha f$ exist for all multi-indices for which $|\alpha| \leq n$ and posses continuous extensions to $\overline{\mathcal{Q}}_D(\theta)$. Then, the norm on this space is given by

$$\|f\|_{\mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))} = \sum_{|\alpha|\leq n} \sup_{x\in\overline{\mathcal{Q}}_D(\theta)} |\partial^\alpha f(x)|.$$

In the same manner as the last two results in [132], we show by both results, Corollaries 3.1 and 3.2, that we are able to get bounds on the overall rate of convergence up to arbitrarily high probability provided that the continuation values satisfy some regularity assumptions. To get stronger statements, we suppose that our chosen loss function fulfills $\ell(\cdot) \leq (\cdot)^2$.

Corollary 3.3.1. *Assume that the framework above holds with state space $E = \overline{\mathcal{Q}}_D(\theta)$. Moreover, suppose that there exists $n \in \mathbb{N}$ such that C_l is the continuous extension to $\overline{\mathcal{Q}}_D(\theta)$ of some function $C_l \in \mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))$ for all $t_l,\ l = 1, ..., L-1$. Let $\widetilde{M}$ be a positive integer with $1 \leq n < \widetilde{M}$ and ζ be a constant with $\zeta \geq 1$, and define*

$$\mathcal{H}_l := \{\pi \in \Pi^D_{\widetilde{M}}(\overline{\mathcal{Q}}_D(\theta)); \|\pi\|_{\mathcal{C}^0(\overline{\mathcal{Q}}_D(\theta))} \leq 2\|C_l\|_{\mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))}\}.$$

Then, for any $\varepsilon > 0$, we have

$$\mathcal{P}_{(\boldsymbol{\omega}_{01},...,\boldsymbol{\omega}_{0N})\in\boldsymbol{\Omega}_0^N} \left((\widehat{V}_0 - V_0)^2 \leq 2\varepsilon^2 + 2\sum_{k=1}^{L-1} \widetilde{\gamma}^k \gamma (2\gamma^2 + \gamma^3)^{k-1}(1+\gamma)(3\varepsilon + \gamma\widetilde{\zeta}^2(\widetilde{M})^{-2n}) \right) \geq 1-\delta_0 \tag{3.48}$$

with $\delta_0 = 8\exp\left(-\frac{N\varepsilon^2}{128F^2}\right) + \sum_{k=1}^{L-1} 22(q+1)\left(\frac{696\widetilde{\zeta}(F+2\widetilde{\zeta})}{\varepsilon}\right)^q \exp\left(-\frac{N\varepsilon^2}{128G_k^2}\right)$ *and* $q = \frac{(\widetilde{M}-1+D)!}{(\widetilde{M}-1)!D!}$; $\widetilde{\zeta} \geq 1$ *is any constant fulfilling* $\zeta^2\|C_l\|^2_{\mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))} \leq \widetilde{\zeta}^2$ *for* $l = 1, ..., L-1$.

Proof. To begin with, let us remark that the VC-dimension of a finite-dimensional vector space of measurable real-valued functions coincides with its vector space dimension, see [46]. Thus, due to the dimension of the space of multivariate polynomials we obtain

$$vc(\mathcal{H}_l) \leq \begin{pmatrix} \widetilde{M} - 1 + D \\ D \end{pmatrix} = \dim(\Pi^D_{\widetilde{M}}) := q$$

at all $t_l,\ l = 1, ..., L-1$. Moreover, combining statement (3.42) with the relation

$$\inf_{\pi\in\mathcal{H}_l} \int \ell(C_l - \pi) \leq \int \ell(C_l - \pi_l) \leq \|C_l - \pi_l\|^2_\infty \leq \zeta^2(\widetilde{M})^{-2n}\|C_l\|^2_{\mathcal{C}^n(\overline{\mathcal{Q}}_D(\theta))} \tag{3.49}$$

for any $\pi_l \in \mathcal{H}_l$ and each $l = 1, ..., L-1$, leads to statement (3.48); the last inequality of (3.49) is a consequence of Jackson-type estimates, where the constant $\zeta \geq 1$ depends on $\mathcal{Q}_D(\theta)$, see [46]; quantity H is replaced by $2\widetilde{\zeta}$ in G_k as well. □

Let $Lip(\overline{\mathcal{Q}}_D(\theta))$ be the space of Lipschitz continuous functions f on the closed cube $\overline{\mathcal{Q}}_D(\theta)$ that satisfy

$$|f(x) - f(y)| \leq \zeta_L |x - y|, \ \forall x, y \in \overline{\mathcal{Q}}_D(\theta),$$

where $0 \leq \zeta_L < \infty$ is the Lipschitz constant.

Corollary 3.3.2. *Assume that the framework above holds with state space* $E = \overline{\mathcal{Q}}_D(\theta)$. *Moreover, suppose that* $C_l \in Lip(\overline{\mathcal{Q}}_D(\theta))$. *Let* $\widetilde{M}$ *be a positive integer such that*

$$\left(\frac{88\zeta_L \theta D}{\widetilde{M} - 1 + D}\right)^2 \leq \|C_l\|^2_{\mathcal{C}^0(\overline{\mathcal{Q}}_D(\theta))}$$

and

$$\mathcal{H}_l := \{\pi \in \Pi^D_{\widetilde{M}}(\overline{\mathcal{Q}}_D(\theta)); \|\pi\|_{\mathcal{C}^0(\overline{\mathcal{Q}}_D(\theta))} \leq 2\|C_l\|_{\mathcal{C}^0(\overline{\mathcal{Q}}_D(\theta))}\}$$

for $l = 1, ..., L-1$. *Then, for any* $\varepsilon > 0$, *we have*

$$\mathcal{P}_{(\boldsymbol{\omega}_{01},...,\boldsymbol{\omega}_{0N}) \in \boldsymbol{\Omega}_0^N} \left((\widehat{V}_0 - V_0)^2 \leq 2\varepsilon^2 + 2\sum_{k=1}^{L-1} \widetilde{\gamma}^k \gamma (2\gamma^2 + \gamma^3)^{k-1}(1+\gamma) \left(3\varepsilon + \gamma \left(\frac{88\zeta_L \theta D}{\widetilde{M} - 1 + D} \right)^2 \right) \right)$$
$$\geq 1 - \delta_0 \qquad (3.50)$$

with $\delta_0 = 8 \exp\left(-\frac{N\varepsilon^2}{128F^2}\right) + \sum_{k=1}^{L-1} 22(q+1) \left(\frac{696\widetilde{\zeta}(F+2\widetilde{\zeta})}{\varepsilon}\right)^q \exp\left(-\frac{N\varepsilon^2}{128G_k^2}\right)$ *and* $q = \frac{(\widetilde{M}-1+D)!}{(\widetilde{M}-1)!D!}$; $\widetilde{\zeta} \geq 1$ *is a constant fulfilling* $\|C_l\|^2_{\mathcal{C}^0(\overline{\mathcal{Q}}_D(\theta))} \leq \widetilde{\zeta}^2$, $l = 1, ..., L-1$.

Proof. Statement (3.50) is an immediate consequence of Feinerman-Newman's Lemma, see [50] or [132]. □

3.4 Numerical Investigations

In this section we concentrate on the numerical performance of our RRM method. In Subsection 3.4.1 we examine numerical properties of our RRM method and aim at a comparative study with the LSM method. Based on both methods, we focus our attention on dual methods in Subsection 3.4.2. Before we proceed, let us make mention of some technical details. In all our experiments we assume that the vector-valued price process **S** of D assets follows a multi-dimensional geometric Brownian motion, compare (2.14). To reduce variance, we use antithetic variables for generating paths. In general, the coder should take special care about generation of uniform pseudo-random numbers. Our experiments with different random number generators, namely linear congruential methods and lagged fibonacci generators, have shown that another choice than the Mersenne Twister

MT19937 [95] can lead to inaccurate values. We use the following linear congruential method for generating a seed vector,

$$x_0 = (as_0 + b)\text{mod}M,\ x_i = (ax_{i-1} + b)\text{mod}M,$$

$i = 1, ..., 623,$ s_0 seed, $a = 214013,$ $b = 2531011,$ $M = 4294967296$. We convert uniformly distributed random numbers with the Ziggurat algorithm rather than with the Polar method because of speed, see [94]. Clearly, the practical efficiency of a Monte Carlo estimator does not only depend on the theoretical properties, but also on the quality of the underlying random number generator. A vital calibration of regression-based Monte Carlo methods for pricing complex financial products with an early exercise feature is the choice of basis functions. There are a variety of proposals for choosing a basis, e.g., see [91], [97], [4], [58]. We specify our chosen functions in a suitable position. At this point we would like to remark that a well-considered choice of basis functions can avoid the curse of dimensionality; taking multivariate polynomials in D input variables of total degree $\widetilde{M}$ into account leads to a power law growth of the number of coefficients proportional to $D^{\widetilde{M}}$.

3.4.1 Numerical Experiments

Before we investigate the performance of our RRM method, we briefly discuss the outlier detection procedure used in our experiments. Robust regression is often applied in a statistical context, and, thus, we suppose reasonable distributions of the error. Unfortunately, we are not familiar with any distribution, and, that is why, we should work with empirical distributions as pointed out by [53]. In so doing, working with empirical α-quantiles suggests itself, and a possible outlier detection procedure is given as follows:

$$\begin{gathered} r_j^{help} = |r_j|,\ j = 1, ..., J,\ r = (r_1, ..., r_J) \\ r^{help} = sort(r^{help}) \qquad (3.51) \\ \gamma_1 = r^{help}_{\lfloor \alpha J \rfloor},\ 0 \ll \alpha < 1, \end{gathered}$$

and, if a loss function with a further transition point is chosen,

$$\gamma_2 = r^{help}_{\lfloor \beta J \rfloor},\ \alpha < \beta < 1.$$

Notify that we presume $(1-\alpha)100$ $((1-\beta)100)$ per cent of the data points to be (extreme) outliers a priori. For pricing put options it makes more sense to run the above procedure without taking absolute values of the residuals due to asymmetric data; put options are naturally bounded above. Needless to mention, other approaches for detecting outliers are cogitable. In our experiments we determine transition points in each regression procedure by the initial estimation of the coefficients at the beginning of the iteration. As we focus on Bermudan options, we take the least squares solution as initial values for our Newton-Raphson-based solver; taking the approximation of the previous time step seems to be more efficient if we price options with a couple of exercise opportunities.

Remark 3.4.1. *Following [111], due to complexity $\mathcal{O}(J \log_2 J)$ Quicksort should be selected for the procedure sort$(\cdot)$ in (3.51) if $J > 50$. It is stable and often superior to Heap-Sort by a factor of 1.5-2 with respect to speed; nevertheless, Heap-Sort is the more elegant algorithm. For $J < 50$ we prefer the stable method by Shell having worst case complexity $\mathcal{O}(J^{3/2})$.*

To get a first impression of our RRM algorithm, we price Bermudan Max call options on two uncorrelated assets with

$$T = 3,\ \Delta t = 1/3,\ K = 100,\ r = 0.05,\ \sigma_d = 0.2,\ \delta_d = 0.1,\ S_0^d = 90, 100, 110\ \forall d.$$

To measure error quantities, we run each algorithm fifty times with different seeds in the random number generator; the benchmark values are 8.0724, 13.9018 and 21.3441 for the out-of-the-money (OTM), at-the-money (ATM) and in-the-money (ITM) option, respectively, and result from the three-dimensional Binomial tree proposed in [116] with Richardson extrapolation based on 4,500 and 9,000 time steps. Before we move on, let us compare several solvers for robust regression. For this comparative study we price the ATM max call option; we run the algorithms with just one seed and work with the first seven functions of the basis

$$\{1, \{X_d\}_{d=1}^{D}, \{X_d^2\}_{d=1}^{D}, \{X_d X_e\}_{e \neq d}^{D}, \{X_d^3\}_{d=1}^{D}, X_1^2 X_2, X_1 X_2^2\}, \tag{3.52}$$

where $X_1, X_2, ...$ denote the highest, second highest, and so on asset value, and $N = 100,000$ paths. Table 3.2 shows the convergence behavior of IRLS, $\widetilde{\mathrm{RR}}_{\mathrm{New}}$ and $\mathrm{RR}_{\mathrm{New}}$ combined with Huber's loss function measured by the a posteriori relative error

$$\|x^{(k)} - x^*\|_2 / \|x^*\|_2$$

with the approximated optimum x^*. Both methods based on the Newton-Raphson iteration perform quadratic convergence; as $\widetilde{\mathrm{RR}}_{\mathrm{New}}$ works with a perturbed system by construction, our robust regression solver shows better performance. As $\widetilde{\mathrm{RR}}_{\mathrm{New}}$ solves a weighted least squares problem in each iteration step, using this approach leads to a speed-up factor of about 5 compared with the IRLS approach. We obtain speed-up factors of about 18 or more by using our proposed solver $\mathrm{RR}_{\mathrm{New}}$ rather than using IRLS. Needless to say, it pays

Table 3.2: A posteriori relative error of different robust regression solvers.

	IRLS		$\widetilde{\text{RR}}_{\text{New}}$		RR_{New}	
Step	t_{L-1}	t_1	t_{L-1}	t_1	t_{L-1}	t_1
0	6.5851	1.0617	6.5851	1.0617	6.5851	1.0617
1	2.7867	$1.2561{\cdot}10^{-1}$	$2.5500{\cdot}10^{-1}$	$1.6567{\cdot}10^{-2}$	$2.5502{\cdot}10^{-1}$	$1.6567{\cdot}10^{-2}$
2	1.31	$4.8139{\cdot}10^{-2}$	$3.0171{\cdot}10^{-5}$	$2.4134{\cdot}10^{-6}$	$2.4928{\cdot}10^{-5}$	$2.4133{\cdot}10^{-6}$
3	$6.6768{\cdot}10^{-1}$	$1.1464{\cdot}10^{-2}$	$1.1940{\cdot}10^{-14}$	-	-	-
⋮	⋮	⋮	-	-	-	-
15	$1.2429{\cdot}10^{-4}$	$1.1180{\cdot}10^{-10}$	-	-	-	-
⋮	⋮	-	-	-	-	-
27	$1.0836{\cdot}10^{-8}$	-	-	-	-	-
CPU Time Ratio	-	-	5.93	4.09	28.13	18.04

Notes. Algorithm specific parameters are as follows: $\alpha = 0.9$, $TOL_{\Delta x} = TOL_{\nabla f} = 10^{-10}$.

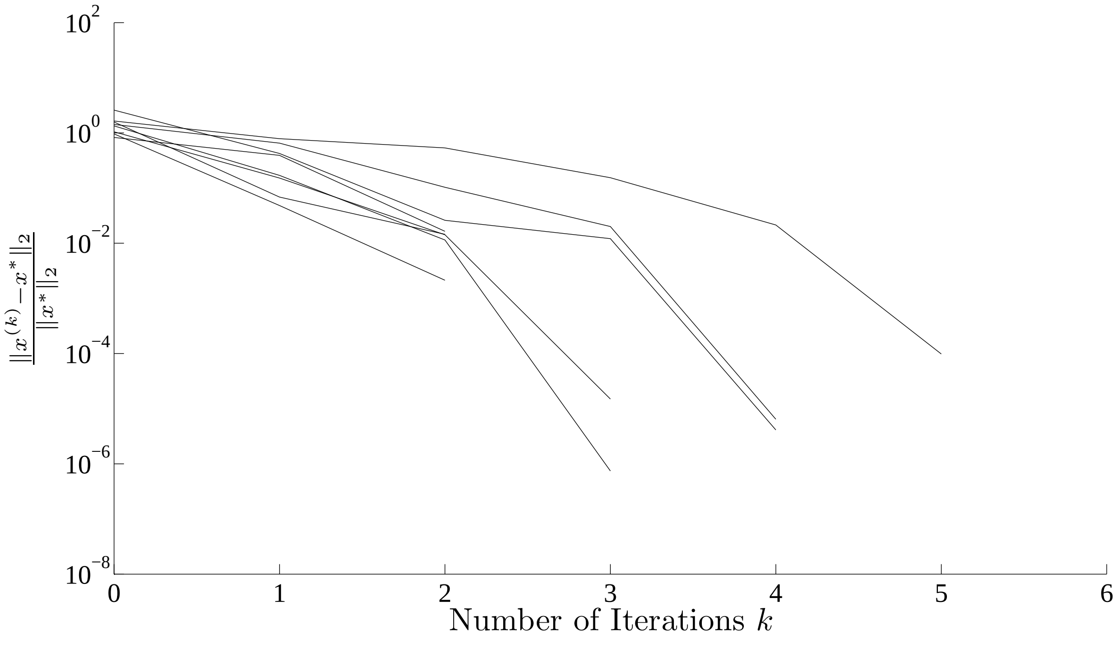

Figure 3.3: Convergence of RR_{New} for the Jonen loss function.

Notes. Algorithm specific parameters are as follows: $\alpha = 0.9$, $\beta = 0.99$, $TOL_{\Delta x} = TOL_{\nabla f} = 10^{-10}$.

to work with the update technique (3.7). Figure 3.3 shows the convergence behavior for our solver combined with our proposed loss function, i.e. the Jonen loss function. Even though we work with subgradients and generalized Hessian matrices, we observe second order convergence in the neighborhood of the optimum; see Remark 3.2.1. The speed-up factors of Table 3.2 are representative for this test as well. Needless to mention, the lower we choose the transition points the more iterations we need. As the results of these numerical experiments reflect our experience, we use our proposed solver RR_{New} in all our experiments combined with

$$TOL_{\Delta x} = 10^{-15} \text{ and } TOL_{\nabla f} = 10^{-12}.$$

Figure 3.4 shows the bias for the ATM max call option determined by the LSM method, i.e. $\alpha = \beta = 1$, and our RRM method in combination with the three robust loss functions listed in Table 3.1; again, we run the algorithms with the first seven functions of basis (3.52) and $N = 100,000$ paths. At first glance we see that we get a striking bias reduction by using our RRM method rather than the LSM method. Our proposed loss function called Jonen shows a better performance than Huber's and Talwar's loss function; nevertheless, our RRM method combined with the Huber and Talwar loss function shows a remarkable accuracy improvement. As we consider a bias reduction for any combination of transition points with $\alpha \in [0.85, 1.00]$ and $\beta \in [0.99, 1.00]$, we conclude that our RRM method shows robustness against the choice of these parameter settings. We see a similar convergence behavior for the ATM, ITM and OTM option; optimal transition point combinations lead to a bias reduction by a factor of up to 219, 1,452 and 3.66 for the OTM, ATM and ITM option, respectively. By our chosen algorithm settings, we obtain an unbiased estimator for the OTM and ATM option. The bias variance tradeoff is a well-known problem in estimation theory – a reduction in bias yields in an increasing variance, and vice versa. Anyway, the surfaces of Figure 3.5 show that our method does not suffer by this effect. On the contrary, we often see a slight reduction in variance by using robust regression; this fact seems to be natural as outliers increase variance.

To underline these results we price a Bermudan arithmetic average call option on two uncorrelated assets with

$$T = 3,\ \Delta t = 1/3,\ K = 100,\ r = 0.05,\ \sigma_d = 0.4,\ \delta_d = 0.1,\ S_0^d = 90, 100, 110\ \forall d.$$

Again, we calculate benchmark values by the Rubinstein tree with Richardson extrapolation based on 4,500 and 9,000 time steps; the benchmark values are 8.9553, 13.1573 and 18.3282 for the OTM, ATM and ITM option, respectively. We run the algorithms with the basis

$$\{1, \{s_d\}_{d=1}^{D}, \{s_d^2\}_{d=1}^{D}, 1/D \sum_{d=1}^{D} s_d\} \tag{3.53}$$

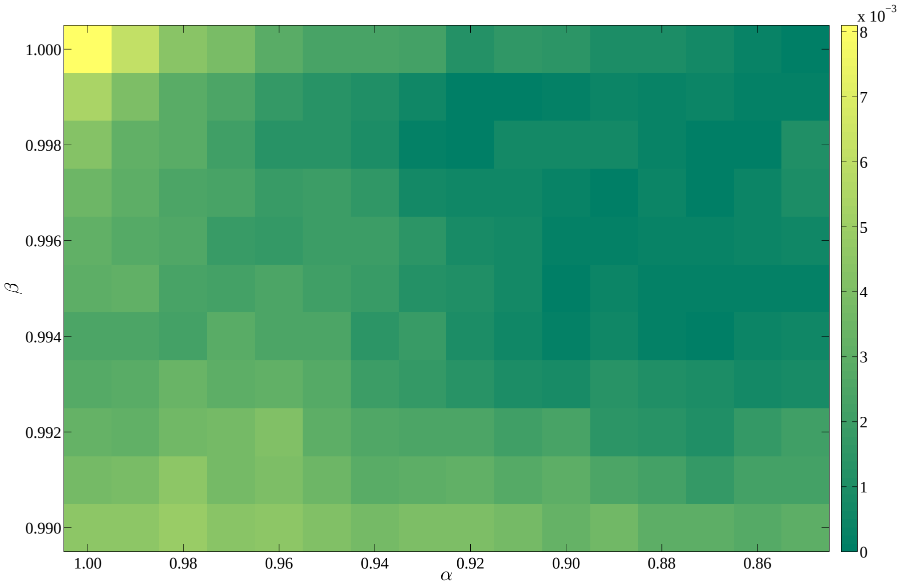

Figure 3.4: Bias calculated by the LSM method ($\alpha = \beta = 1$) and the RRM method combined with the Huber ($\beta = 1$), Talwar ($\alpha = 1$), Jonen ($\alpha \neq 1$, $\beta \neq 1$) loss function for a Bermudan Max call option on two assets.

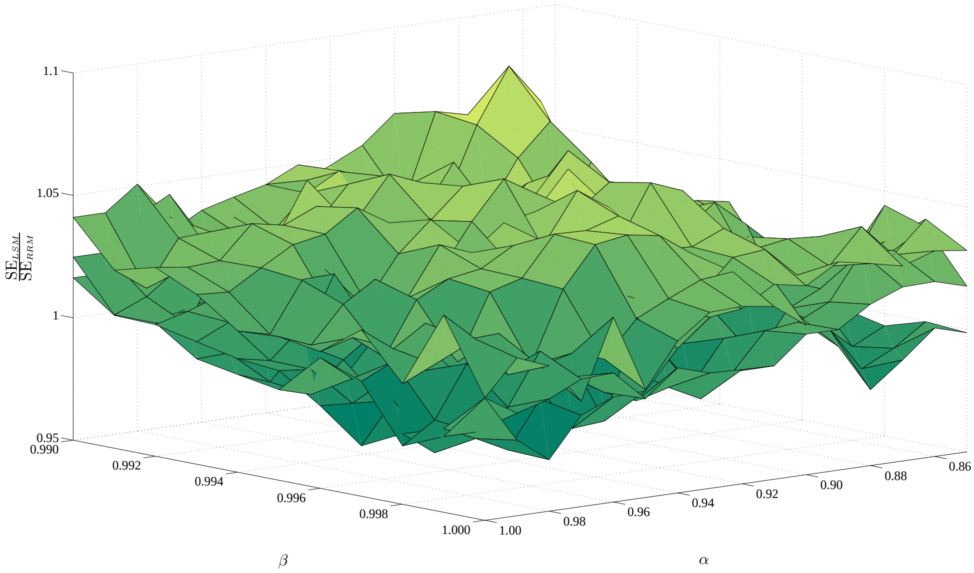

Figure 3.5: SE (standard error) ratios for several transition points calculated for an OTM, ATM and ITM Bermudan Max call option on two assets.

and N=100,000 paths; s_d is the value of asset d. Again, Figure 3.6 shows a significant

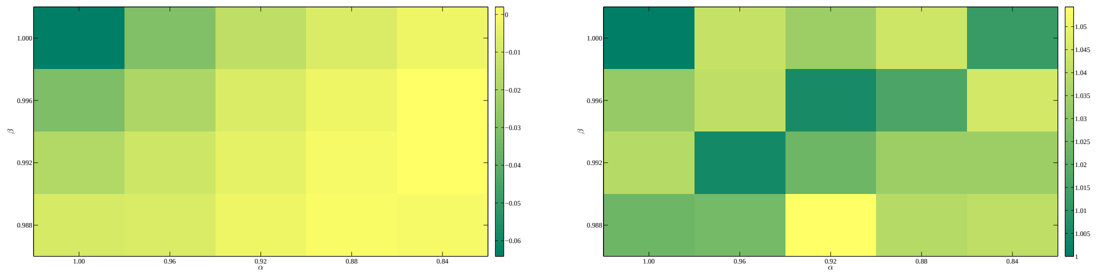

Figure 3.6: Bias and SE ratio on the left and on the right, respectively, calculated by the LSM method and the RRM method for a Bermudan arithmetic average call option on two assets.

reduction in bias with decreasing variance simultaneously for the OTM option by using robust regression. These numerical results are representative, as we observe the same convergence behavior for the ATM and ITM option.

A further bias variance tradeoff caused by regression estimations should be taken into consideration: increasing the number of paths and basis functions leads to a lower and higher variance, but to a higher and lower bias, respectively. Thus, regression-based Monte Carlo methods are seriously effected by the same dilemma, and [59] gives an optimal relation between the number of paths and the number of basis functions in the (geometric) Brownian motion case. However, to ensure a fair comparison between the LSM method and our RRM method, we measure the performance of both estimators $\overline{V}$ for a quantity V by the root mean square error (RMSE) given by

$$\mathrm{RMSE}(\overline{V}) := \sqrt{\mathrm{E}[(\overline{V} - V)^2]} = \sqrt{\mathrm{Bias}(\overline{V})^2 + \mathrm{Variance}(\overline{V})};$$

it quantifies the balance of variance and bias. Even though we are able to reduce bias by a significant factor without increasing variance for a fixed number of paths and basis functions, a vital investigation of efficiency of our proposed Monte Carlo estimator is a comparative study with respect to accuracy and computational efficiency, in terms of CPU time and memory requirements. Let J_l be the number of paths for which we should exercise the option at time date t_l, i.e. $\tau_j^* = l$ in Algorithm 1. It is well known that the early exercise regions of max call options become smaller with decreasing time steps, see [42]. The closer we are at t_0 the lower should become the quantity J_l as the variance of the simulated paths increases in time. Taking this observation into account, to save CPU time a natural adaptive control of α and β might be favorable such as realized by

$$\alpha_l = \max\left(\alpha_{help}, \alpha_{\min}\right), \quad l = L-1, ..., 1,$$

where α_{help} might be given by the equation of a linear function through the points

$(x_1, \alpha_{\max})$ and (x_2, α_2), $0 < x_1 < x_2 < 1,\ 0 \ll \alpha_{\min} \leq \alpha_2 < \alpha_{\max} \leq 1,$

$$\alpha_{help} = \alpha_{max} + \frac{\alpha_2 - \alpha_{max}}{x_2 - x_1}(J_{l+1} - x_1), \tag{3.54}$$

and

$$\beta = \alpha_{\max} - \theta(\alpha_{\max} - \alpha_l),\ 0 < \theta \ll 1,$$

if we choose a loss function with a further transition point. Numerical tests have indicated that this strategy is very robust to the free parameters; in the following we work with

$$(x_1, \alpha_{\max}) = (0.15, 1.0),\ (x_2, \alpha_2) = (0.4, 0.9),\ \alpha_{\min} = 0.85,\ \theta = 0.1.$$

Additionally, we work with the Jonen loss function if $\alpha_l < 0.9$ and with Huber's loss function if $0.9 \leq \alpha_l < 0.99$; otherwise, we switch to least squares. We take J_{l+1} as an approximation for J_l in (3.54), where J_L is the number of paths in-the-money at t_L. In order to calculate speed-up factors, we increase the number of paths and number of basis functions simultaneously, and Figure 3.7 indicates that we get the same accuracy by using our RRM method with seven functions and the LSM method with ten functions for an ATM Bermudan Max call option on two assets. Thus, as illustrated in Figure 3.8

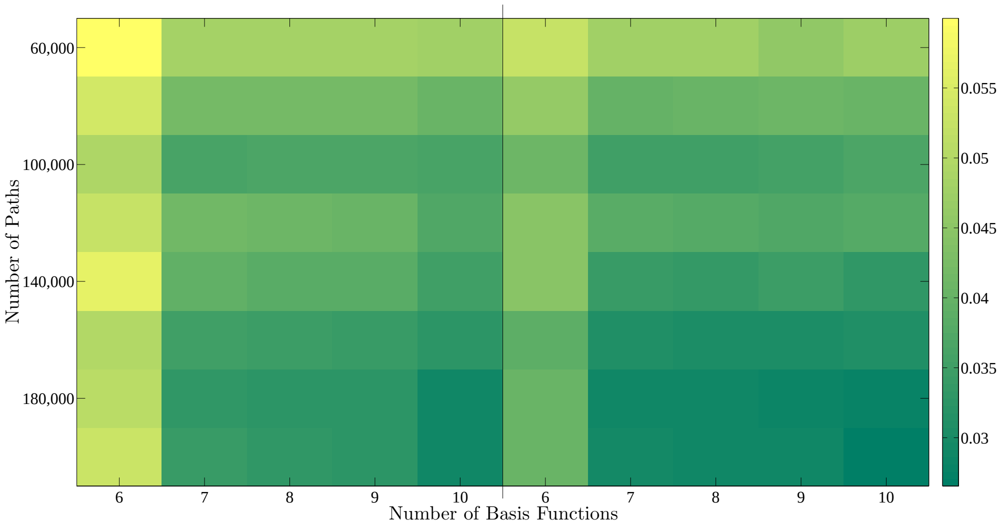

Figure 3.7: RMSE calculated by the LSM method, the left half, and the RRM method, the right half, for an increasing number of paths and functions of basis (3.52).

we get speed-up factors between 1.3 and 2.15 for max call options on two assets with $\sigma_d = 0.2,\ d = 1, 2$, and between 1.6 and 2.6 for this option type with $\sigma_d = 0.4,\ d = 1, 2$. The lower half of the right sketch of Figure 3.8 shows the validity of our comparative study, as using just nine basis functions for the LSM method leads to higher RMSEs

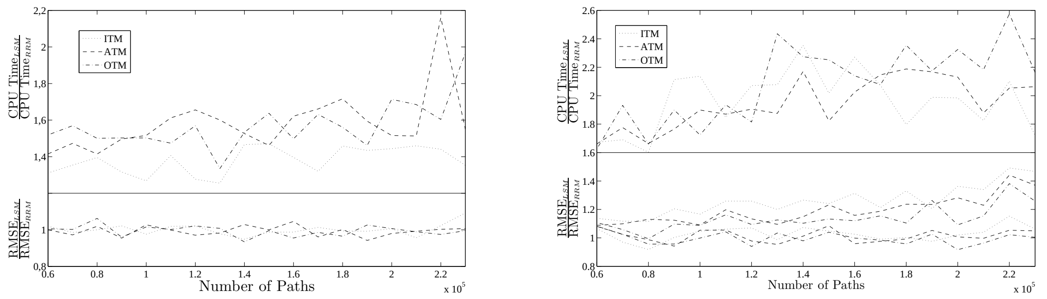

Figure 3.8: LSM method vs. RRM method for max call options on two assets with $\sigma_d = 0.2$, on the left, and $\sigma_d = 0.4$, on the right.

such that a comparison between both methods is not possible. After analyzing numerical results of this subsection, we come to the conclusion that we can significantly improve convergence by using robust regression rather than least squares. We discuss the vital investigation of our RRM method in higher dimensions in the following subsection.

3.4.2 Dual Methods

Measuring errors for options on more than two assets is often hard, as calculating benchmark values for these options is time expensive or often nearly infeasible due to hardware constraints. Remember that pricing multi-asset options with an early exercise feature by the LSM method, and, hence, by our RRM method, create an interleaving estimator, see [58]. Anyway, without having knowledge of benchmark values we cannot make a statement about the performance of both methods in higher dimensions. Thus, we focus our attention on the AB approach [4] introduced in Chapter 2, and as a measure of accuracy we take the difference between the upper and lower bound denoted by Δ_0; as previously mentioned, the tighter the lower and upper bound and the higher the lower bound the more accurate is the approximated early exercise rule. We use the notation of Chapter 2, i.e. N_0 is the number of paths used for determining the early exercise strategy, N_1 is the number of paths used for estimating the lower bound, N_2 is the number of paths used for estimating the upper bound, and N_3 is the number of simulations in the simulation for estimating the upper bound. To get a first impression of the performance of our RRM method in higher dimensions, we price Bermudan Max call options on five uncorrelated and correlated assets by the AB approach combined with both methods, the LSM and our RRM method, for a fixed number of paths and basis functions. As pointed out by [23], it makes sense to distinguish between options with reasonable symmetric and asymmetric parameters. Taking this investigation into account, we run the algorithms with the basis

$$\{1, \{X_d\}_{d=1}^{D}, \{X_d^2\}_{d=1}^{D}, \{X_d^3\}_{d=1}^{2}, X_1X_2, X_1X_3, X_2X_3, X_1^2X_2, X_1X_2^2\} \tag{3.55}$$

Table 3.3: Lower and upper Bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan Max call options on five assets.

	S_0	Algorithm	Lower Bound	Upper Bound	95% CI	CPU Time Ratio	Δ Ratio
a)	90	LSM	16.622 (0.016)	16.640 (0.017)	[16.590,16.672]	-	–
		RRM	16.623 (0.016)	16.630 (0.016)	[16.591,16.662]	0.99	2.56
	100	LSM	26.106 (0.019)	26.133 (0.020)	[26.067,26.172]	-	–
		RRM	26.132 (0.019)	26.141 (0.019)	[26.094,26.179]	1.00	2.86
	110	LSM	36.716 (0.022)	36.751 (0.022)	[36.673,36.795]	-	–
		RRM	36.741 (0.022)	36.764 (0.022)	[36.698,36.808]	0.99	1.53
b)	90	LSM	27.517 (0.033)	27.698 (0.039)	[27.452,27.775]	-	–
		RRM	27.577 (0.033)	27.641 (0.034)	[27.513,27.707]	1.00	2.84
	100	LSM	37.825 (0.038)	38.062 (0.047)	[37.750,38.154]	-	–
		RRM	37.915 (0.037)	38.028 (0.039)	[37.842,38.106]	0.99	2.09
	110	LSM	49.254 (0.043)	49.507 (0.047)	[49.170,49.600]	-	–
		RRM	49.380 (0.041)	49.542 (0.044)	[49.299,49.628]	1.20	1.56
c)	90	LSM	38.933 (0.051)	39.184 (0.079)	[38.833,39.340]	-	–
		RRM	39.040 (0.050)	39.155 (0.052)	[38.943,39.256]	1.00	2.20
	100	LSM	50.161 (0.057)	50.513 (0.071)	[50.048,50.652]	-	–
		RRM	50.263 (0.056)	50.494 (0.070)	[50.153,50.631]	1.01	1.53
	110	LSM	62.291 (0.064)	62.676 (0.080)	[62.166,62.832]	-	–
		RRM	62.400 (0.061)	62.639 (0.068)	[62.276,62.772]	1.00	1.59

Notes. Common option parameters are $T = 3,\ L = 9,\ K = 100,\ r = 0.05,\ \delta_d = 0.1,\ d = 1, ..., 5,$ and, especially, option parameters for a) are $\sigma_d = 0.2\ d = 1, ..., 5,\ \rho_{de} = 0\ \forall d \neq e$, for b) are $\sigma_d = 0.08 + (d-1)0.08,\ d = 1, ..., 5,\ \rho_{de} = 0\ \forall d \neq e$, and for c) are given by the covariance matrix

$$\Sigma = \begin{pmatrix} 0.2^2 & 0.2 \cdot 0.35 \cdot (-0.1) & 0.2 \cdot 0.08 \cdot (-0.2) & 0.2 \cdot 0.5 \cdot 0.05 & 0.2 \cdot 0.4 \cdot 0.0 \\ 0.2 \cdot 0.35 \cdot (-0.1) & 0.35^2 & 0.35 \cdot 0.08 \cdot 0.4 & 0.35 \cdot 0.5 \cdot 0.1 & 0.35 \cdot 0.4 \cdot 0.25 \\ 0.2 \cdot 0.08 \cdot (-0.2) & 0.35 \cdot 0.08 \cdot 0.4 & 0.08^2 & 0.08 \cdot 0.5 \cdot 0.2 & 0.08 \cdot 0.4 \cdot 0.25 \\ 0.2 \cdot 0.5 \cdot 0.05 & 0.35 \cdot 0.5 \cdot 0.1 & 0.08 \cdot 0.5 \cdot 0.2 & 0.5^2 & 0.5 \cdot 0.4 \cdot 0.15 \\ 0.2 \cdot 0.4 \cdot 0.0 & 0.35 \cdot 0.4 \cdot 0.25 & 0.08 \cdot 0.4 \cdot 0.25 & 0.5 \cdot 0.4 \cdot 0.15 & 0.4^2 \end{pmatrix}.$$

Algorithm specific parameters are as follows: $N_0 = 130,000,\ N_1 = 1,000,000,\ N_2 = 1,000,$ $N_3 = 5,000$. The CPU time and Δ_0 ratios are defined by (CPU Time$_{\text{LSM}}$/CPU Time$_{\text{RRM}}$) and ($\Delta_{\text{LSM}}/\Delta_{\text{RRM}}$), respectively.

and

$$\{1, \{s_d\}_{d=1}^{D}, \{s_d^2\}_{d=1}^{D}, \{X_d^j\}_{d,j=1}^{2}, X_1X_2, \{X_d^3\}_{d=1}^{2}, \} \tag{3.56}$$

in case of symmetric and asymmetric parameters, respectively. For options on many assets choosing a basis with non-distinguishable functions might be favorable. In doing so, Table 3.3 reports results by using the functions (3.55) for case a) and (3.56) for cases b),c). The lower bounds produced by our method are higher for all options. Obviously, we obtain a better approximation of the optimal stopping rule strengthened by the fact that the gap between the upper and lower bound is tighter by a factor between 1.54 and 2.86. Especially for the upper bound, we consider a slight reduction in variance by using robust regression. Both methods calculate a 95 % confidence interval in about the same CPU time; more precisely, the CPU time ratios are between 0.99 and 1.20. Even though a least squares solver is superior to our RRM method with respect to CPU time, it might happen that our method is slightly faster than the LSM method as the second phase of dual methods depends on the approximated early exercise strategy. To underline these results, Figure 3.9

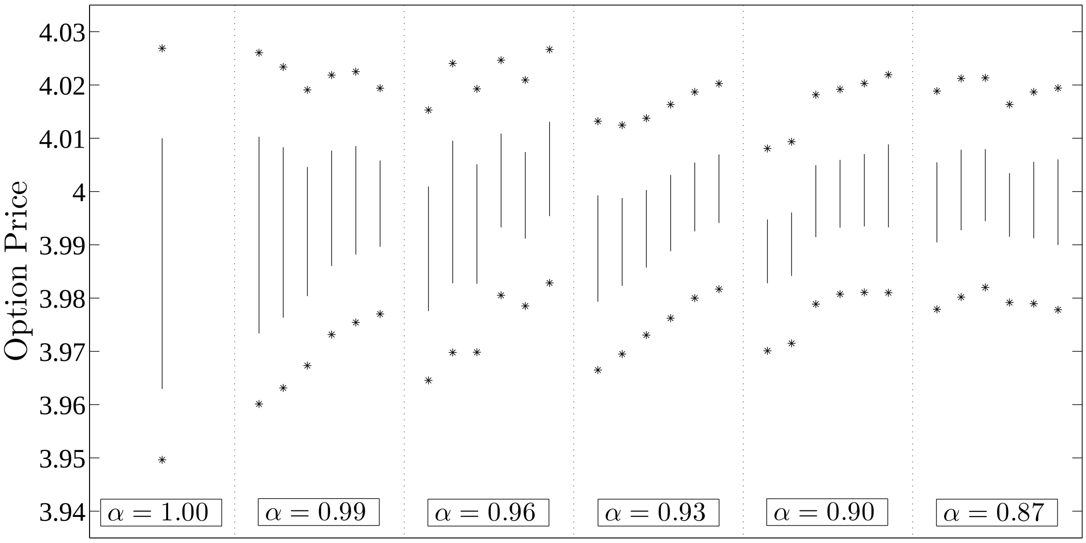

Figure 3.9: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for a Bermudan arithmetic average option on five assets.

Notes. The solid line is the connection between the lower and upper bound; the stars denote the lower and upper bound of the calculated 95% confidence interval. The first column shows the result of the LSM method, i.e. $\alpha = \beta = 1.00$; the other columns show results calculated by the RRM method for a given α combined with the following decreasing values for β: 1.000, 0.999, 0.996, 0.993, 0.990, 0.987. Option parameters and algorithm settings are concorde with the parameters used in Table 3.3 b).

shows lower and upper bounds with resulting 95% confidence intervals estimated by the LSM method and our RRM method combined with several transition points for an ATM Bermudan arithmetic average call option on five assets; we run the algorithms with the first eleven functions of basis (3.53). Once again, we see a remarkable robustness against the choice of transition points. Thus, we are able to improve convergence by using robust regression with any combination of $\alpha \in [0.87, 0.99]$ and $\beta \in [0.987, 1.000]$; depending on these transition points the difference between the upper and lower bound is tighter by a factor between 1.28 and 3.95, and the lower bounds are higher than the value calculated by the LSM algorithm. We observe similar convergence behaviors for an ITM and OTM option with the same parameter settings, and Table 3.4 shows results for the LSM method and our RRM method combined with $\alpha = 0.87,\ \beta = 0.993$. By using our RRM method we obtain a reduction of Δ_0 by factors between 2.91 and 3.46 in the same CPU time. Again, we see that we get higher lower bounds and a slight reduction in variance by using robust regression. For all practical purposes our first experiments are meaningful, since practitioners are often obliged to price options in a fixed time budget.

Table 3.4: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan arithmetic average call options on five assets.

S_0	Algorithm	Lower Bound	Upper Bound	95% CI	CPU Time Ratio	Δ Ratio
90	LSM	1.540 (0.005)	1.560 (0.006)	[1.530,1.573]	-	–
	RRM	1.547 (0.004)	1.553 (0.005)	[1.538,1.562]	1.03	3.46
100	LSM	3.961 (0.007)	4.002 (0.008)	[3.948,4.027]	-	–
	RRM	3.987 (0.006)	3.999 (0.007)	[3.974,4.012]	0.99	3.34
110	LSM	9.301 (0.008)	9.359 (0.010)	[9.285,9.378]	-	–
	RRM	9.340 (0.007)	9.360 (0.008)	[9.326,9.376]	0.98	2.91

Notes. See Table 3.3 b) for option parameters. Algorithm settings for the LSM and RRM method are as follows: $N_0 = 300,000,\ N_1 = 1,000,000,\ N_2 = 1,000,\ N_3 = 5,000$.

To make a meaningful comparative speed-accuracy study between the LSM and our RRM method, we fix the number of paths for determining the lower and upper bounds, i.e. we fix the quantities N_1, N_2, N_3. Since we are able to interpret the tightness of the lower and upper bound as a measure of accuracy, we increase the number of paths for calculating the coefficients of the model function for the continuation value, i.e. we increase N_0, as well as the number of basis functions. Before we discuss a performance comparison, let us address a vital point from a practical point of view. For this purpose, we add to basis

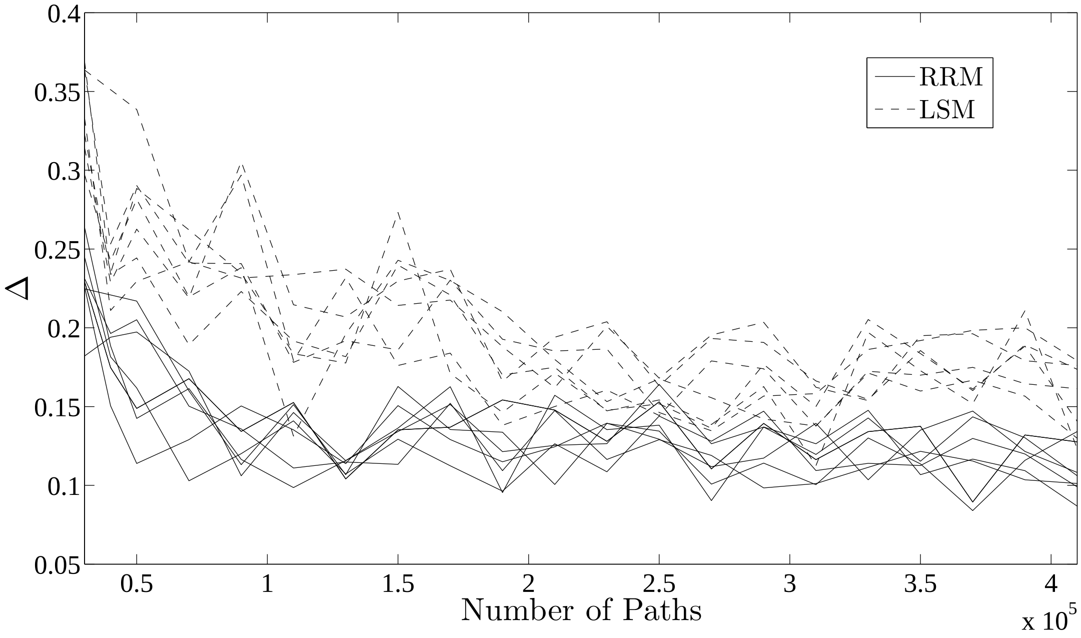

Figure 3.10: Δ_0 calculated by the LSM and RRM method with several basis functions and an increasing number of paths N_0 for an ATM Bermudan Max call option on five assets.

Notes. See Table 3.3 b) for option parameters. Algorithm settings for the LSM and RRM method are as follows: $N_1 = 1,000,000$, $N_2 = 1,000$, $N_3 = 5,000$.

(3.56) the following functions:

$$\begin{aligned} &\{X_1^2X_2, X_1X_2^2, X_3, X_1X_3, X_2X_3, \{X_3^j\}_{j=2}^3, X_1^2X_3, \\ &X_1X_3^2, X_2^2X_3, X_2X_3^2, \{X_4^j\}_{j=1}^3, X_1X_4, X_2X_4, X_3X_4\}. \end{aligned} \tag{3.57}$$

Figure 3.10 shows Δ_0 calculated by both methods with the first $M = 16, 18, 20, 23, 25, 29, 35$ basis functions for an increasing number of paths N_0. At first glance our RRM method shows a better convergence behavior than the LSM method. Even if we use more than double as much basis functions with the LSM method, we never reach the accuracy that we obtain with our RRM method. Evidentally, our RRM method seems to be less sensitive with respect to the choice of basis functions strengthened by the results of Table 3.5, in which we determine bounds by

(B1) basis (3.56) combined with $\{X_3, X_3^2, X_1X_3, X_2X_3, X_1^4, X_1^5, X_1X_2^2, X_1^2X_2, X_4, X_4^2\}$,

(B2) basis (3.56) combined with the first eleven functions of (3.57),

(B3) basis $\{1, \{s_d\}_{d=1}^D, \{s_d^2\}_{d=1}^D\}$ combined with monomials up to a total degree of two in the ordered statistics.

Notify that basis (B2) and (B3) are natural choices due to Taylor expansion. For this experiment, we choose a reasonable large number of paths for the regression step, namely $N_0 = 330,000$. As we can see, the lower bounds produced by the LSM method are

Table 3.5: Accuracy comparison of the LSM and RRM method with respect to several basis functions for an ATM Bermudan Max call option on five assets.

Basis	Algorithm	Lower Bound	Upper Bound	95% CI	Δ
(B1)	LSM	37.921 (0.038)	38.067 (0.041)	[37.847,38.147]	0.146
	RRM	37.947 (0.037)	38.064 (0.039)	[37.875,38.141]	0.117
(B2)	LSM	37.872 (0.038)	38.045 (0.043)	[37.797,38.128]	0.172
	RRM	37.956 (0.037)	38.069 (0.040)	[37.883,38.147]	0.114
(B3)	LSM	37.855 (0.039)	38.062 (0.042)	[37.779,38.145]	0.207
	RRM	37.937 (0.037)	38.040 (0.039)	[37.864,38.117]	0.103

Notes. See Table 3.3 b) for option parameters. Algorithm settings for the LSM and RRM method are as follows: $N_0 = 330,000$, $N_1 = 1,000,000$, $N_2 = 1,000$, $N_3 = 5,000$.

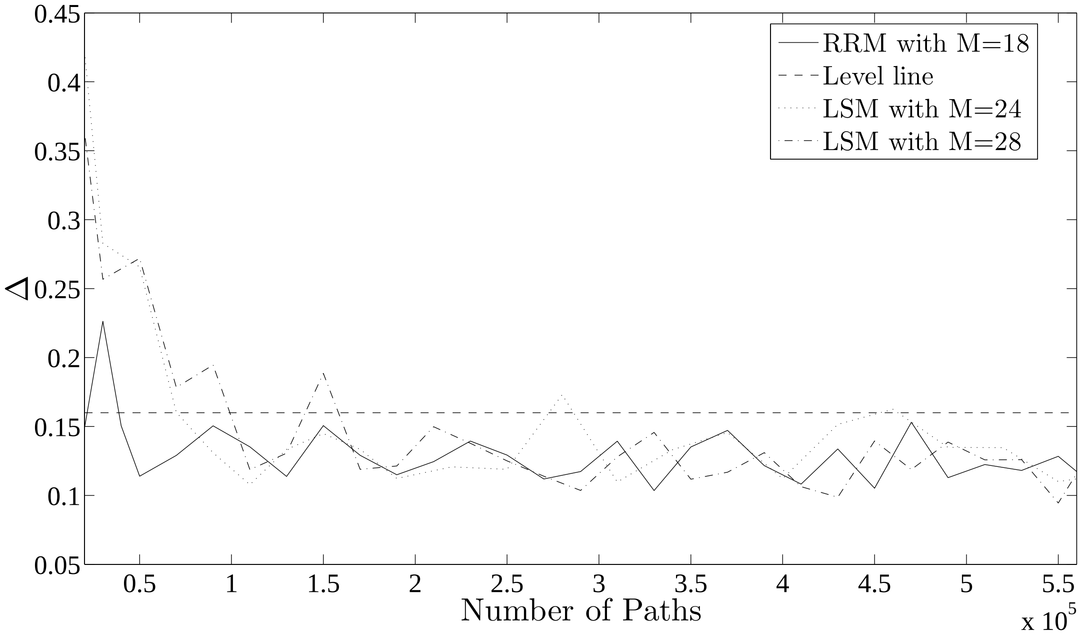

Figure 3.11: Δ_0 against number of paths calculated by the AB approach combined with the LSM and RRM method for an ATM Bermudan Max call option on five assets.

Notes. See notes of Figure 3.10 for algorithm settings.

quite unstable with respect to several basis functions, and Δ_0 strongly varies for several basis functions, although our chosen basis are of about the same size. To sum up, we are strongly limited in the choice of basis functions, if we are interested in a comparative study between both methods. Therefore, in the following we use functions working well for the LSM method. More precisely, for symmetric option parameters we add to basis (3.55) functions in the order

$$\{X_1^4, X_1^5, X_3^3, X_1X_3^2, X_1^2X_3, X_2X_3^2, X_2^2X_3, X_4^3, X_1X_4, X_2X_4, X_3X_4\},$$

and for asymmetric option parameters we work with basis (B1). Let us keep in mind that our study is a worst case study. Figure 3.11 illustrates our test for getting a comparative study with respect to CPU time and is as follows: we fix a given level of tightness, e.g., 0.16 as in Figure 3.11, and we examine the number of paths and the number of basis functions needed to guarantee that we remain below this bound; for instance, to guarantee that we remain below the limit of 0.16 we need eighteen basis functions and at least 30,000 paths for our RRM method as well as twenty-eight basis functions and at least 160,000 paths for the LSM method; by using only twenty-four basis functions we cannot consider a long term undercut for the LSM method. Measuring the first time

Table 3.6: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for Bermudan Max call options on five assets and given levels of tightness.

	Δ_0 Level	S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time Ratio	Settings N_0|M
a)	0.013	90	LSM	16.638 (0.016)	16.647 (0.017)	[16.606,16.679]	-	370,000|29
			RRM	16.636 (0.016)	16.647 (0.016)	[16.605,16.679]	2.24	110,000|18
	0.017	100	LSM	26.132 (0.020)	26.145 (0.020)	[26.094,26.184]	-	210,000|29
			RRM	26.132 (0.019)	26.141 (0.019)	[26.094,26.180]	2.49	130,000|18
	0.03	110	LSM	36.748 (0.022)	36.769 (0.022)	[36.705,36.812]	-	210,000|29
			RRM	36.747 (0.022)	36.776 (0.022)	[36.704,36.821]	2.05	190,000|18
b)	0.12	90	LSM	27.573 (0.033)	27.678 (0.035)	[27.507,27.748]	-	170,000|28
			RRM	27.569 (0.033)	27.650 (0.034)	[27.504,27.716]	1.85	90,000|18
	0.16	100	LSM	37.895 (0.038)	38.014 (0.042)	[37.820,38.095]	-	160,000|28
			RRM	37.857 (0.038)	38.008 (0.046)	[37.786,38.087]	1.94	30,000|18
	0.2	110	LSM	49.336 (0.042)	49.509 (0.045)	[49.254,49.597]	-	90,000|28
			RRM	49.369 (0.041)	49.538 (0.047)	[49.289,49.631]	4.75	70,000|18

Notes. The option parameters for a) and b) are consistent with the parameters used in Table 3.3 a) and b), respectively. See notes of Figure 3.10 for algorithm settings.

we go below our given level of accuracy such that we can guarantee that we stay below this level gives us speed-up factors reported in Table 3.6. Notify that we pass our test with significant fewer number of paths and number of basis functions by using our RRM method rather than the LSM method. Thus, pricing five-dimensional options with an early exercise feature by our RRM method leads to speed-up factors between 1.85 and 4.75. To conclude our numerical experiments, lower and upper bounds as well as resulting 95% confidence intervals are reported in Table 3.7 for an ATM Bermudan Max call option on thirty uncorrelated assets. Numerical experiments have indicated that we need at least fourty-four basis functions and at least 1,000,000 paths to ensure that we remain below the bound of 1.0 with the LSM method; the basis consists of the functions

Table 3.7: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM method for a Bermudan Max call option on thirty assets and a given level of tightness.

Δ_0 Level	Algorithm	Lower Bound	Upper Bound	95% CI	CPU Time Ratio	Settings N$_0$|M
1.0	LSM	298.499 (0.155)	299.405 (0.242)	[298.194,299.879]	-	1,000,000|44
	RRM	298.701 (0.148)	299.475 (0.180)	[298.411,299.829]	3.30	320,000|11

Notes. Option parameters are as follows: $T = 3$, $L = 12$, $K = 100$, $r = 0.05$, $S_0^d = 100$, $\sigma_d = 0.08 + 0.015d$, $\delta_d = 0.1$, $d = 1, ..., 30$. Algorithm settings for the LSM and RRM method are as follows: $N_1 = 2,000,000$, $N_2 = 1,000$, $N_3 = 5,000$.

$$\{1, \{X_d\}_{d=1}^3, \{X_d^2\}_{d=1}^2, \{X_d^3\}_{d=1}^2, X_1X_2, X_1X_3, X_2X_3, \{X_3^j\}_{j=2}^3, X_1^4, \{s_d\}_{d=1}^{30}\}. \tag{3.58}$$

As it turned out that we could not reach our given level of tightness by using just non-distinguishable basis functions, we worked with distinguishable functions as well. The results for our RRM method are very remarkable, as we just need the first eleven functions of basis (3.58) and at least 320,000 paths for passing our test such that we get a speed-up factor of 3.30; we observe a variance reduction factor of 1.81 for the upper bound.

Concluding this section, let us make some remarks on computational efficiency in terms of memory requirements. To ensure convergence of regression-based Monte Carlo, we have to increase the number of paths and number of basis functions simultaneously. Our numerical experiments have shown that we can improve convergence significantly by using our RRM method rather than the LSM method. Thus, contrary to the LSM method, the implementation of our RRM method has much less memory requirements, especially for higher-dimensional options.

Chapter 4

Efficiency Increase

The probabilistic convergence rate $\mathcal{O}(1/\sqrt{N})$ might be the proof of Monte Carlo estimation's inadequacy. To overcome this shortcoming, a number of variance reduction techniques have been proposed, see, e.g., [58], for an overview. From our point of view, it is essentiell to speed up convergence by several techniques, and, thus, we pursue the goal of increasing efficiency in two ways. To start with, we propose a variance reduction technique via importance sampling in Section 4.1. We introduce our change of drift technique and discuss an efficient implementation. Random numbers with low discrepancy are also widely-used catalyzers for improving convergence of Monte Carlo estimators. Thus, in Section 4.2 we discuss the application of quasi-Monte Carlo techniques for pricing financial derivatives with an early exercise feature and study the extension to dual methods. To conclude this chapter, we discuss some further acceleration techniques in Section 4.3.

4.1 Variance Reduction via Importance Sampling

The key idea in reducing variance via importance sampling is to change the probability measure. By doing so, a common way is to change the drift in Brownian motion of the underlying model to drive paths in regions which are more important for variance. In Subsection 4.1.1 we propose a change of drift technique to improve the convergence behavior of Monte-Carlo estimators for financial derivatives with an early exercise feature. In order to determine the drift minimizing variance, we discuss algorithms for solving the underlying stochastic optimization problem in Subsection 4.1.2. To conclude the theoretical part, we extend our proposed approach to the vital class of dual methods for pricing American derivatives in Subsection 4.1.3. The last part of this section, Subsection 4.1.4, is devoted to investigating the numerical performance of our proposed approach.

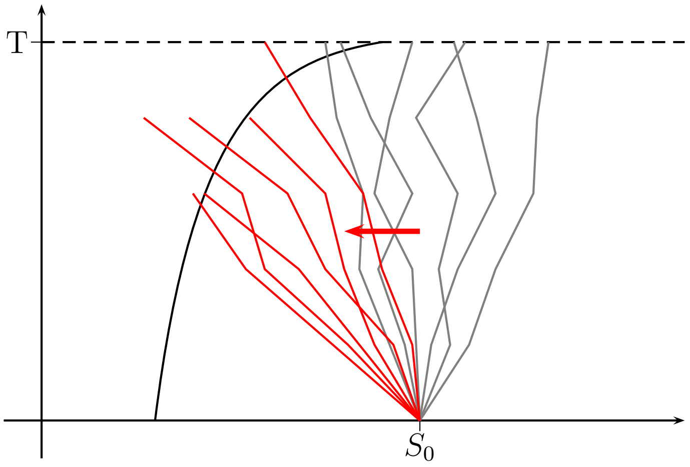

Figure 4.1: Change of drift in Brownian motion for American options.

4.1.1 Variance Reduction by a Change of Drift

To start with, let us motivate a change of drift in Brownian motion by Figure 4.1 showing the valuation procedure for American options by Monte Carlo methods. As we can see, the idea of a change of drift is to enforce the early exercise decision such that zero-paths vanish; indeed, zero-paths, i.e. $\tau_1^n = L$ and $Z_L^n = 0$ in (4.5), lie out-of-the-money at maturity $t_L = T$ and are drivers for an increasing variance. Having this sketch in mind, let us discuss a realization of this concept in the following. Throughout this section, we denote by $\{\mathcal{F}_t^W | 0 \leq t \leq T\}$ the filtration generated by a standard D-dimensional P-Brownian motion $(W_t)_{0 \leq t \leq T}$, which is augmented to involve all subsets of sets having P-probability 0. A basic tool for changing the probability measure in our context is given by Girsanov's theorem:

Theorem 4.1.1 (Girsanov). *For each fixed $T \in [0, \infty)$, let θ be a $\mathbb{R}^D$-valued process adapted to $\{\mathcal{F}_t^W\}$ satisfying*

$$\int_0^t \|\theta_s\|_2^2 ds < \infty \ a.s., \ \ 0 \leq t \leq T. \tag{4.1}$$

Define the processes $(\zeta_t)_{0 \leq t \leq T}$ and $(\widetilde{W}_t)_{0 \leq t \leq T}$ by

$$\zeta_t := \exp\left\{ \int_0^t \theta_s^T dW_s - 0.5 \int_0^t \|\theta_s\|_2^2 ds \right\} \tag{4.2}$$

and

$$\widetilde{W}_t := W_t - \int_0^t \theta_s ds,$$

respectively. Under the assumption that $\mathbb{E}_P[\zeta_T] = 1$, $(\zeta_t)_{0\leq t\leq T}$ *is a martingale and the measure* Q *on* $(\Omega, \mathcal{F}_T)$ *defined by* $dQ = \zeta_T dP$ *is equivalent to* P*. Moreover, the process* $(\widetilde{W}_t)_{0\leq t\leq T}$ *is a standard Brownian motion with respect to* $\{\mathcal{F}_t^W\}$ *under* Q*.*

Proof. See [57]. □

By the way, a more general version of a change of two equivalent probability measures is defined by the Radon-Nikodym theorem; therefore, process (4.2) is also called Radon-Nikodym derivative or likelihood probability, see [105] for the original work in French. Under the assumption that (4.1) is fulfilled, a sufficient condition for the requirement that $\mathbb{E}_P[\zeta_T] = 1$ in Girsanov's theorem is given by the Novikov condition

$$\mathbb{E}^P\left[\exp\left\{\frac{1}{2}\int_0^T \|\theta_s\|_2^2 ds\right\}\right] < \infty,$$

see [107]; we refer to [80] or [100] for a deeper look into this theory. Assume that our underlying asset model is driven by an Itô process under the risk-neutral probability measure P,

$$dS_t = a(S_t, t)dt + b(S_t, t)dW_t, \tag{4.3}$$

where W_t is a standard D-dimensional Brownian motion under P; in order to ensure existence and uniqueness of the solution, we assume that $a(\cdot,\cdot) \in \mathbb{R}^m$ and $b(\cdot,\cdot) \in \mathbb{R}^{m\times D}$ fulfill the common boundedness and regularity conditions; for convenience, again, we set $t_0 = 0$ and the value of S_0 at t_0 denoted by s_0 is assumed to be known, i.e. $s_0 \in \mathbb{R}^m$. Based on the vital framework above, we can justify a change of drift in Brownian motion, and we get that

$$\widetilde{W}_t := W_t - \int_0^t \theta_s ds$$

is a Brownian motion under Q such that

$$dS_t = (a(S_t, t) + b(S_t, t)\theta_t)dt + b(S_t, t)d\widetilde{W}_t \tag{4.4}$$

is the Itô process (4.3) under Q. Let us come back to the valuation of American-style derivatives on assets driven by (4.3) such that our starting point is the optimal stopping problem (2.1). By exploring Girsanov's theorem, for any stopping time $\tau \in \mathcal{T}_{1,L}$ and any time-invariant θ, we obtain that

$$\mathbb{E}_0^P[e^{-r\tau\Delta t}Z_\tau] = \mathbb{E}_0^Q[e^{-r\tau\Delta t}Z_\tau e^{-\theta^T\widetilde{W}_\tau - 0.5\|\theta\|_2^2\tau\Delta t}],$$

and, consequently, according to (2.1) the fair price can be rewritten as

$$\sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0^P[e^{-r\tau\Delta t}Z_\tau] = \sup_{\tau\in\mathcal{T}_{1,L}} \mathbb{E}_0^Q[e^{-r\tau\Delta t}Z_\tau e^{-\theta^T\widetilde{W}_\tau - 0.5\|\theta\|_2^2\tau\Delta t}].$$

Let us make for a moment the assumption that the optimal stopping rule τ_1^* is known, i.e. we suppose the knowledge of the continuation value process $(C_l)_{1\leq l\leq L}$. Then, an unbiased estimate of the continuation value at time date t_0 is given by

$$\frac{1}{N}\sum_{n=1}^{N} e^{-r\Delta t(\tau_1^*)^n}\widetilde{Z}^n_{(\tau_1^*)^n} e^{-\theta^T\widetilde{W}^n_{(\tau_1^*)^n}-0.5\|\theta\|_2^2(\tau_1^*)^n\Delta t} =: \sum_{n=1}^{N} X_n \tag{4.5}$$

with the N i.i.d. samples X_n; $(\tau_l^*)^n$ and $\widetilde{Z}_l^n$ are the optimal stopping time and the payoff of path n, $n = 1, ..., N$, at time t_l, $l = 1, ..., L$, respectively, whereby $\widetilde{Z}_l^n$ results from simulating the process (4.4); $\widetilde{W}_l^n$ is the value of the simulated Brownian motion under Q for path n, $n = 1, ..., N$, at time t_l, $l = 1, ..., L$. Again, notify that estimator (4.5) combined with any stopping strategy τ with values in $\{1, ..., L\}$ is low-biased. As the variance of estimator (4.5) is ruled by the samples X_n, $n = 1, ..., N$, we concentrate on

$$Var_0^Q\left(e^{-r\Delta t\tau_1^*}Z_{\tau_1^*}e^{-\theta^T\widetilde{W}_{\tau_1^*}-0.5\|\theta\|_2^2\tau_1^*\Delta t}\right). \tag{4.6}$$

One can easily verify that (4.6) can be written as

$$\mathbb{E}_0^Q\left[\left(e^{-r\tau_1^*\Delta t}Z_{\tau_1^*}e^{-\theta^T\widetilde{W}_{\tau_1^*}-0.5\|\theta\|_2^2\tau_1^*\Delta t}\right)^2\right] - C_0^2 \tag{4.7}$$

such that our target quantity is the first term of (4.7) simplified to

$$\mathbb{E}_0^Q\left[\left(e^{-r\tau_1^*\Delta t}Z_{\tau_1^*}e^{-\theta^T\widetilde{W}_{\tau_1^*}-0.5\|\theta\|_2^2\tau_1^*\Delta t}\right)^2\right] = \mathbb{E}_0^P\left[\left(e^{-r\tau_1^*\Delta t}Z_{\tau_1^*}\right)^2 e^{-\theta^T W_{\tau_1^*}+0.5\|\theta\|_2^2\tau_1^*\Delta t}\right] =: \mathcal{V}(\theta). \tag{4.8}$$

In the context of variance reduction by a change of drift in Brownian motion for European-style options, a similar derivation to an objective function can be found in [122]. From a statistical point of view, the optimal estimator for an unbiased estimator is the one with the smallest variance. Thus, for the purpose of determining the drift minimizing variance, we should solve the optimization problem

$$\min_{\theta\in\mathbb{R}^D}\mathcal{V}(\theta). \tag{4.9}$$

Note that the second term of the right-hand side of (4.7) and the payoff of $\mathcal{V}(\theta)$ are independent of the parameters θ_d, $d = 1, ..., D$; this fact makes our optimization problem very attractive. The following theorem indicates that our minimization problem is well-posed, which means that a minimum exists and is unique.

Theorem 4.1.2. *Suppose that the mathematical framework above holds with the reasonable assumption that $P(e^{-r\tau_1^*}Z_{\tau_1^*} > 0) \neq 0$. Then, $\mathcal{V}(\theta)$ is strictly convex on $\mathbb{R}^D$ such that the minimization problem (4.9) has a unique solution $\theta^* \in \mathbb{R}^D$.*

Proof. In analogy to [7], we verify that $\mathcal{V}(\theta)$ is twice continuously differentiable with gradient

$$\nabla\mathcal{V}(\theta) = \mathbb{E}_0^P\left[(\theta\tau_1^* - W_{\tau_1})(e^{-r\tau_1^*\Delta t}Z_{\tau_1^*})^2 e^{-\theta^T W_{\tau_1^*}+0.5\|\theta\|_2^2\tau_1^*\Delta t}\right]$$

and Hessian matrix

$$\nabla^2\mathcal{V}(\theta) = \mathbb{E}_0^P\left[(I_D\tau_1^* + (\theta\tau_1^* - W_{\tau_1})(\theta\tau_1^* - W_{\tau_1})^T)(e^{-r\tau_1^*\Delta t}Z_{\tau_1^*})^2 e^{-\theta^T W_{\tau_1^*}+0.5\|\theta\|_2^2\tau_1^*\Delta t}\right],$$

where I_D is the $(D \times D)$ identity matrix. Since $\nabla^2\mathcal{V}(\theta)$ is positive definite and $\lim_{\|\theta\|_2\to\infty}\mathcal{V}(\theta) = \infty$, $\mathcal{V}(\theta)$ is a strictly convex function on $\mathbb{R}^D$ with a unique minimum. □

As in practical applications the optimal stopping rule is not known, we might use regression-based Monte Carlo method for the numerical realization of this step; for instance, the LSM method or our RRM method can be used to approximate the continuation value at every exercise date. Our variance reduction approach involving a change of drift in Brownian motion for pricing options with an early exercise feature based on an approximated exercise policy τ_1 is given in Algorithm 3.

Before we discuss the vital step 1 in Algorithm 3 of finding the optimal solution of (4.9) in the next subsection, let us briefly address another approach on a change of drift in Brownian motion for American options. Moreni [96] presumed that the LSM estimator with an unknown exercise strategy is ruled by the quantity $\mathcal{V}(\theta)$; in particular, the willful neglect that the discounted cash flows of the option price estimator are dependent was done. For practical applications, as a compromise, Moreni proposed to work with the optimal drift of the estimator for the European option with the same parameter settings as the American option to be priced; finding the optimal drift for the American option has remained unsolved. Compared to this approach, we suppose a priori that an optimal stopping strategy is known to derive an estimator as well as an optimization problem for finding the drift minimizing variance.

Algorithm 3 $ABIS1(N_1, T, L, \Theta)$.

```
Input: $\tau_1, N_1, T, L, \Theta, \Gamma$
Output: $\widehat{L}_0, \widehat{\sigma}_L$
1: $\theta \leftarrow SolveSOP(\tau_1, T, L, \Theta, \Gamma)$
2: Draw $N_1$ i.i.d. samples $X_n = e^{-r\Delta t\tau_1^n}\widetilde{Z}_{\tau_1^n}^n e^{-\theta^T\widetilde{W}_{\tau_1^n}^n - 0.5\|\theta\|_2^2\tau_1^n\Delta t}$
3: $\widehat{L}_0 = \frac{1}{N_1}\sum_{n=1}^{N_1} X_n$
4: $\widehat{\sigma}_L^2 = \frac{1}{N_1-1}\sum_{n=1}^{N_1}(X_n - \widehat{L}_0)^2$
```

Notes. The routine $SolveSOP(\tau_1, T, L, \Theta, \Gamma)$ in line 2 realizes the approximation of the optimal drift; Γ contains parameters for setting up this solver, see Subsection 4.1.2. $\widehat{\sigma}_L$ might be used for calculating an confidence interval of L_0.

4.1.2 Optimization Methods

Stochastic approximation for solving stochastic optimization problems is still going strong, and a widely-used approach is the Robbins-Monro (RM) algorithm proposed in [113]. To begin with, we direct our attention to this stochastic search algorithm given by

$$\theta^{(k+1)} = \theta^{(k)} - \gamma_k Y_k, \ k = 0, 1, 2, ..., \tag{4.10}$$

where Y_k is a noisy estimate of $\nabla\mathcal{V}(\theta^{(k)})$ and γ_k is an appropriate sequence. The convergence is ensured by the following theorem:

Theorem 4.1.3. *Assume that the following hypotheses hold:*

(H1) $\gamma_k > 0, \ \gamma_k \to 0$ *as* $k \to \infty, \ \sum_{k=0}^{\infty} \gamma_k = \infty, \ \sum_{k=0}^{\infty} \gamma_k^2 < \infty.$

(H2) $\exists \theta^* \in \mathbb{R}^D, \nabla\mathcal{V}(\theta^*) = 0, \ \forall \theta \in \mathbb{R}^D : \ \theta \neq \theta^*, \ (\theta - \theta^*) \cdot \nabla\mathcal{V}(\theta) > 0.$

(H3) $\exists c > 0, \forall k \geq 0 : \mathbb{E}[\|Y_k\|^2 | \mathcal{F}_k] < c(1 + \|\theta^{(k)} - \theta^*\|^2)$ *a.s..*

Then, the sequence $(\theta^{(k)})_{k\geq 0}$ *defined by (4.10) converges almost surely (a.s.) to* θ^**. This statement is even true, if (H3) is replaced by*

(H4) $\exists c > 0, \sum_{k=0}^{\infty} \gamma_k^2 \mathbb{E}[\|Y_k\|^2 | \mathcal{F}_k] \leq c < \infty.$

Proof. See [113], [45] or [84] for probabilistic convergence statements under (H1)-(H3); for a proof with respect to the assumptions (H1), (H2) and (H4) see [17]. □

Unfortunately, neither (H3) nor (H4) are trivial assumptions in our framework such that we cannot verify them; numerical tests have confirmed our guess that these assumptions do not hold at all. To this end, Arouna [7] could gain a variance reduction approach via a change of drift in Brownian motion for European-style options by using a truncated version of the RM method to find the optimal drift. The idea of a truncation is to make sure that the new determined drift $\theta^{(k+1)}$ does not jump out of a given compact set U at each iteration step. In the case of such an event, the set U is expanded and the entries of the new drift $\theta^{(k+1)}$ might be set back to constant values. To this end, to avoid large steps, the projection algorithm proposed by [32], is given by the procedure

$$\theta^{(k+1)} = \begin{cases} \theta^{(k)} - \gamma_k Y_k, & \text{if } \|\theta^{(k)} - \gamma_k Y_k\|_2 \leq U_{c_k} \\ \overline{\theta}^{(k)}, & \text{otherwise} \end{cases}, \ k = 0, 1, 2, ..., \tag{4.11}$$

with

$$\overline{\theta}^{(k)} = \begin{cases} \overline{\theta}_1, & \text{if } c_k \text{ is even} \\ \overline{\theta}_2, & \text{if } c_k \text{ is odd} \end{cases} \tag{4.12}$$

and

$$c_k = \sum_{i=0}^{k-1} 1_{\|\theta^{(i)}-\gamma_i Y_i\|_2 > U_{c_i}}. \tag{4.13}$$

With Theorem 4.1.2 and the additional assumption that $\mathbb{E}[|e^{-r\tau_1^*\Delta t} Z_{\tau_1^*}|^{4p}] < \infty,\ p > 1$, in analogy to [7], we are able to prove that a sequence U_c can be selected such that $\theta^{(k)}$ converges a.s. to the unique solution of the equation $\nabla\mathcal{V}(\theta) = 0,\ \theta \in \mathbb{R}^D$. In accordance with Arouna's choice, we run the truncated RM algorithm with the sequences $\gamma_k = 1/(1+k)$ and $U_p = \sqrt{0.1\log(p) + U_0},\ p = 1, 2, ...,\ U_0 = 10$.

Let us discuss an alternative way of solving our stochastic optimization task (4.9). By generating N_4 i.i.d. samples, we are able to replace our original stochastic optimization task

$$\min_{\theta\in\mathbb{R}^D} \mathbb{E}_0^P \left[\left(e^{-r\tau_1^*\Delta t} Z_{\tau_1^*}\right)^2 e^{-\theta^T W_{\tau_1^*} + 0.5\|\theta\|_2^2 \tau_1^* \Delta t} \right] \tag{4.14}$$

by its deterministic counterpart

$$\min_{\theta\in\mathbb{R}^D} \frac{1}{N_4} \sum_{n=1}^{N_4} \left(e^{-r(\tau_1^*)^n \Delta t} Z^n_{(\tau_1^*)^n}\right)^2 e^{-\theta^T W^n_{(\tau_1^*)^n} + 0.5\|\theta\|_2^2 (\tau_1^*)^n \Delta t} =: \min_{\theta\in\mathbb{R}^D} \nu(\theta), \tag{4.15}$$

and ordinary solvers such as the Newton-Raphson method might be used to determine the optimal drift. Notify that the entries of the gradient $\nabla\nu(\theta) \in \mathbb{R}^D$ and the Hessian matrix $\nabla^2\nu(\theta) \in \mathbb{R}^{D\times D}$ of $\nu(\theta)$ are given by

$$\begin{aligned}
(\nabla\nu(\theta))_d &:= \frac{\partial\nu(\theta)}{\partial\theta_d} \\
&= \frac{1}{N_4} \sum_{n=1}^{N_4} (\theta_d(\tau_1^*)^n \Delta t - (W^n_{(\tau_1^*)^n})_d) \left(e^{-r(\tau_1^*)^n \Delta t} Z^n_{(\tau_1^*)^n}\right)^2 e^{-\theta^T W^n_{(\tau_1^*)^n} + 0.5\|\theta\|_2^2 (\tau_1^*)^n \Delta t},
\end{aligned} \tag{4.16}$$

$d = 1, ..., D$, and

$$\begin{aligned}
(\nabla^2\nu(\theta))_{de} &:= \frac{\partial^2\nu(\theta)}{\partial\theta_d\partial\theta_e} \\
&= \frac{1}{N_4} \sum_{n=1}^{N_4} ((\theta_e(\tau_1^*)^n \Delta t - (W^n_{(\tau_1^*)^n})_e)(\theta_d(\tau_1^*)^n \Delta t - (W^n_{(\tau_1^*)^n})_d) + (\tau_1^*)^n \Delta t 1_{\{d=e\}}) \cdot \\
&\qquad \left(e^{-r(\tau_1^*)^n \Delta t} Z^n_{(\tau_1^*)^n}\right)^2 e^{-\theta^T W^n_{(\tau_1^*)^n} + 0.5\|\theta\|_2^2 (\tau_1^*)^n \Delta t},
\end{aligned} \tag{4.17}$$

$d, e = 1, ..., D$, respectively, where $1_{\{d=e\}}$ denotes the indicator function. Quasi-Newton-Raphson methods for solving deteministic unconstrained optimization tasks build up an approximation of the inverse Hessian matrix such that cheap update techniques in each iteration step can be used. These methods promise to accelerate the search procedure for unconstrained optimization tasks. Thus, in our numerical experiments in Subsection 4.1.4 we test the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approach to avoid the expensive evaluation of (4.17); see [51] for the BFGS algorithm. It is well known that the Newton-Raphson method is locally quadratically convergent, see [41]. Following the discussion in [41], we use both convergence criteria

$$\max_{d=1,\dots,D} \frac{|\theta_d^{(k+1)} - \theta_d^{(k)}|}{\max\{|\theta_d^{(k+1)}|, 1\}} \leq TOL_{\Delta\theta}, \quad 0 < TOL_{\Delta\theta} \ll 1, \tag{4.18}$$

to test for convergence on $\Delta\theta$, and

$$\max_{d=1,\dots,D} \frac{|(\nabla\nu(\theta^{(k+1)}))_d \max\{|\theta_d^{(k+1)}|, 1\}|}{\max\{\nu(\theta^{(k+1)}), 1\}} \leq TOL_{\nabla\nu}, \quad 0 < TOL_{\nabla\nu} \ll 1, \tag{4.19}$$

to test for convergence on zero gradient.

Remark 4.1.1. *Even though, to the best of our knowledge, solving stochastic optimization tasks via reformulating as deterministic optimization tasks is not widely-used, theoretically, it is guaranteed that an optimal solution $\widehat{\theta}_N$ of (4.15) provides an approximation of the exact optimal solution of problem (4.9), see, e.g., [40] for an overview of convergence proofs; especially, see [81] or [118] for Central Limit Theorem-type results.*

4.1.3 Dual Methods

As already pointed out in the second chapter, a vital extension of Monte Carlo methods for pricing American-style derivatives are dual methods. In order to embed our change of drift technique in the framework of dual methods, we need the abstract Bayes formula:

Proposition 4.1.1 (Abstract Bayes Formula)**.** *Let P and Q be two equivalent probability measures and X be a random variable integrable with respect to P. Then, for any $t \in [0, T]$,*

$$\mathbb{E}_t^P[X] = \frac{\mathbb{E}_t^Q\left[X\frac{dP}{dQ}\right]}{\mathbb{E}_t^Q\left[\frac{dP}{dQ}\right]}.$$

Proof. See [100]. □

Remember that the dual problem (2.11) coincides with the optimal stopping problem (2.1) by choosing the martingale according to the Doob-Meyer factorization. By exploring Girsanov's Theorem and applying Proposition 4.1.1, we are able to reformulate the dual problem (2.11) with (2.12) as follows:

$$\begin{aligned} &\sup_{\tau^* \in \mathcal{T}_{1,L}} \mathbb{E}_0^P[e^{-r\tau^*\Delta t} Z_{\tau^*}] \\ &= \mathbb{E}_0^Q\left[\max_{l=1,\dots,L}(e^{-rl\Delta t} Z_l e^{-\theta^T \widetilde{W}_l - 0.5\|\theta\|_2^2 l\Delta t} - \widetilde{M}_l)\right] + \widetilde{M}_0 \end{aligned} \tag{4.20}$$

with

$$\widetilde{M_0} = V_0,\ \widetilde{M_l} = \widetilde{M_{l-1}} + V_l e^{-\theta^T \widetilde{W}_l - 0.5\|\theta\|_2^2 l\Delta t} - \mathbb{E}_{l-1}^Q[V_l e^{-\theta^T \widetilde{W}_l - 0.5\|\theta\|_2^2 l\Delta t}]$$

for $l = 1, ..., L$. To gain from this theory, again, we focus our attention on the dual method proposed by Andersen and Broadie [4], and, by doing so, we consider an exercise strategy defined by an approximation h_l of the continuation value C_l for $l = 1, ..., L$ with $C_L = 0$ such as given by $\tau_l = \inf\{k \geq l | Z_l \geq h_l\},\ l = 1, ..., L$. Then, we have

$$\widetilde{M_0} = \mathbb{E}_0^Q[e^{-r\tau_1 \Delta t} Z_{\tau_1} e^{-\theta^T \widetilde{W}_{\tau_1} - 0.5\|\theta\|_2^2 \tau_1 \Delta t}]$$

and, for $l = 1, ..., l$,

$$\widetilde{M_l} = \widetilde{M_{l-1}} + \mathbb{E}_l^Q[e^{-r\tau_l \Delta t} Z_{\tau_l} e^{-\theta^T \widetilde{W}_{\tau_l} - 0.5\|\theta\|_2^2 \tau_l \Delta t}] - \mathbb{E}_{l-1}^Q[e^{-r\tau_l \Delta t} Z_{\tau_l} e^{-\theta^T \widetilde{W}_{\tau_l} - 0.5\|\theta\|_2^2 \tau_l \Delta t}]$$

with

$$\begin{aligned} &\mathbb{E}_l^Q[e^{-r\tau_l \Delta t} Z_{\tau_l} e^{-\theta^T \widetilde{W}_{\tau_l} - 0.5\|\theta\|_2^2 \tau_l \Delta t}] \\ &= \begin{cases} e^{-rl\Delta t} Z_l e^{-\theta^T \widetilde{W}_l - 0.5\|\theta\|_2^2 l\Delta t}, & \text{if } Z_l \geq h_l \\ \mathbb{E}_l^Q[e^{-r\tau_{l+1}^* \Delta t} Z_{\tau_{l+1}} e^{-\theta^T \widetilde{W}_{\tau_{l+1}} - 0.5\|\theta\|_2^2 \tau_{l+1} \Delta t}], & \text{if } Z_l < h_l \end{cases}. \end{aligned}$$

So, on the basis of Algorithm 3, we propose to implement this modification of the AB approach to calculate an upper bound and a confidence interval for the option to be priced, see Algorithm 4.

Remark 4.1.2. *We should like to stress out that the estimation of upper bounds and confidence intervals as presented in the context of the AB method is just representative for modifying dual problems by our variance reduction approach; e.g., our change of drift technique might be applied to the dual method of Haugh and Kogan [64].*

4.1.4 Numerical Investigations

To start with, we should like to motivate our numerical experiments by Figure 4.4 showing the search paths of the truncated RM algorithm and the Newton-Raphson method for finding the optimal drift; the option to be priced is an ATM arithmetic average call option on two assets. We start both search procedures at the origin, i.e. with zero-drift. The red line shows the strongly fluctuating search path of the truncated RM algorithm. There are two cluster regions: Points are clustered close to the origin, as we set $\overline{\theta}_1 = (0.0045, 0.0045)^T$ and $\overline{\theta}_2 = (0.004, 0.004)^T$ in (4.12); i.e. to ensure convergence we set $\theta^{(k+1)}$ back to these points if too large steps are done. A further accumulation region is around the optimal drift, compare also Figure 4.3 showing the standard error σ_L in Algorithm 4 for several values of the drift vector. Different from that chaotic convergence behavior, the Newton-Raphson method finds the minimum of the deterministic optimization task (4.15) in six

Algorithm 4 $ABIS2(N_4, N_5, T, L, \Theta, \theta, L_0, \widehat{\sigma}_L, \tau_l, z_{1-\alpha/2})$.

Input: $N_2, N_3, T, L, \Theta, \theta, L_0, \widehat{\sigma}_L, \tau_l, z_{1-\alpha/2}$

Output: $U_0, CI_{1-\alpha}$

```
 1: $\Delta t = T/L$ // Time step width
 2: for $l \leftarrow 1$ to $L$ do
 3:     for $n \leftarrow 1$ to $N_2$ do
 4:         $\mathbb{S}_{nl} = (S_{(n-1)D+d,l})_{d=1,...,D} \leftarrow GeneratePaths(N_2, \Delta t, L, \Theta, \theta)$
 5:     end for
 6: end for
 7: for $n \leftarrow 1$ to $N_2$ do
 8:     $\widehat{C}_0 = 0$
 9:     for $l \leftarrow 1$ to $L$ do
10:         $\widehat{C}_l = \frac{1}{N_3} \sum_{m=1}^{N_3} e^{-r\Delta t \tau^m_{l+1}} \widetilde{Z}^m_{\tau^m_{l+1}} e^{-\theta^T \widetilde{W}^m_{\tau^m_{l+1}} - 0.5\|\theta\|_2^2 \tau^m_{l+1} \Delta t}$
11:         if $(\widetilde{Z}^n_l \geq h_l(\mathbb{S}_{nl}))$ then
12:             $V = \widetilde{Z}^n_l e^{-\theta^T \widetilde{W}^n_l - 0.5\|\theta\|_2^2 l \Delta t}$
13:         else
14:             $V = \widehat{C}_l$
15:         end if
16:         $\Delta_l = V - \widehat{C}_{l-1}$
17:         $M_l = \sum_{k=1}^{l} \Delta_k$
18:     end for
19:     $\pi^n = \max_{l=1,...,L} \left( \widetilde{Z}^n_l e^{-\theta^T \widetilde{W}^n_l - 0.5\|\theta\|_2^2 l \Delta t} - M_l \right)$
20: end for
21: $\Delta_0 = \frac{1}{N_2} \sum_{n=1}^{N_2} \pi^n$.
22: $U_0 = L_0 + \Delta_0$
23: $\widehat{\sigma}^2_\Delta = \frac{1}{N_2 - 1} \sum_{n=1}^{N_2} (\pi^n - \Delta_0)^2$
24: $CI_{1-\alpha} = \left[ L_0 - z_{1-\alpha/2} \frac{\widehat{\sigma}_L}{\sqrt{N_1}}, U_0 + z_{1-\alpha/2} \sqrt{\frac{\widehat{\sigma}^2_L}{N_1} + \frac{\widehat{\sigma}^2_\Delta}{N_2}} \right]$
```

Notes. The routine $GeneratePaths(N_2, \Delta t, L, \Theta, \theta)$ in line 4 realizes the simulation of paths according to (4.4), where Θ denotes the set of parameters for the underlying model and θ is the calculated optimal drift. $\widetilde{Z}^m_{\tau^m_{l+1}}$, $m = 1, ..., N_3$, are samples resulting from simulating paths according to (4.4) beginning at state $\mathbb{S}_{nl}$. Notify that the knowledge of an approximated optimal stopping strategy τ_l, $l = 1, ..., L$, implies the knowledge of an approximation h_l of the continuation value C_l.

iteration steps and we observe quadratic convergence. Both methods convergence to the optimal drift such that the variance reduction factors are 10.27 and 14.15 for calculating the drift by the truncated RM algorithm and the Newton-Raphson alogrithm, respectively. Notify that the drift calculated by the Newton-Raphson method seems to be of higher quality, even though we set up the Newton-Raphson method with only $N_4 = 5,000$ samples and the truncated RM method with $TOL_{\Delta\theta} = 10^{-10}$, i.e. 50, 600 samples are required to yield convergence; we calculate speed-up factors in the following examples. Before we study the numerical performance of our variance reduction technique, let

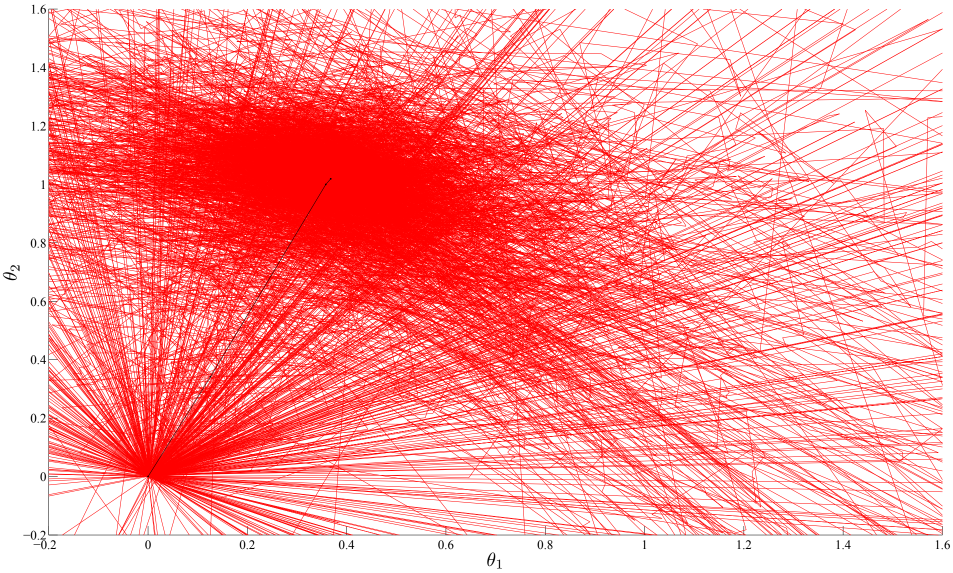

Figure 4.2: Search paths of the truncated RM algorithm and Newton-Raphson method for finding the drift minimizing variance.

us consider the numerical minimization of (4.15) with respect to efficiency reasons. By defining

$$z_l^n := e^{-rl\Delta t} Z_l^n,\ l = 1, ..., L,\ x_n := (z_{\tau_1^n}^n)^2 e^{-\theta^T W_{\tau_1^n}^n + 0.5\|\theta\|_2^2 \tau_1^n \Delta t},\ n = 1, ..., N_4,$$

and

$$y_n^d := (\theta_d \tau_1^n - (W_{\tau_1^n}^n)_d) x_n,\ d = 1, ..., D,\ n = 1, ..., N_4,$$

we are able to rewrite the entries of the gradient (4.16) and the Hessian matrix (4.17) as

$$(\nabla \nu(\theta))_d = \frac{1}{N_4} \sum_{n=1}^{N_4} (\theta_d \tau_1^n \Delta t - (W_{\tau_1^n}^n)_d) x_n,\ d = 1, ..., D,$$

and

$$(\nabla^2 \nu(\theta))_{de} = \frac{1}{N_4} \sum_{n=1}^{N_4} (\theta_e \tau_1^n \Delta t - (W_{\tau_1^n}^n)_e) y_n^d + \tau_1^n \Delta t 1_{\{d=e\}} x_n,\ d, e = 1, ..., D,$$

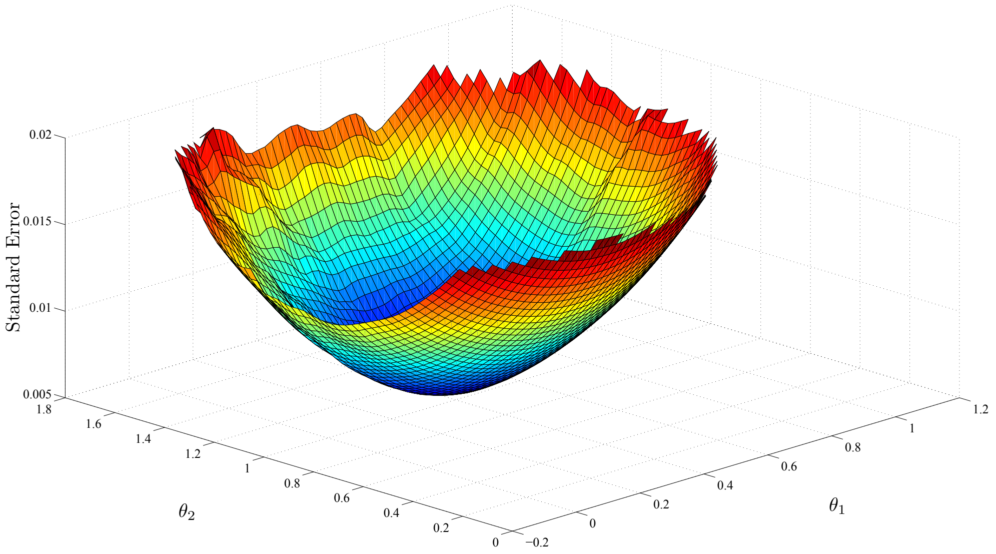

Figure 4.3: Standard error for several drift parameters.

Notes Figures 4.4 and 4.3. Option parameters are $T = 3$, $L = 9$, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$, $\rho = 0$, $\delta_{1,2} = 0.1$, $r = 0.05$. Algorithm specific parameters are as follows: $N_0 = 200,000$, $N_1 = 500,000$; algorithms are run with monomials up to a total degree of two.

respectively. Hence, in every iteration step we save $N_3(D - 1 + D^2)$ evaluations of the exponential function dominating the calculation of both quantities, (4.16) and (4.17). Even though we expect that this simplification makes the Newton-Raphson method very efficient, a comparative study with a quasi Newton-Raphson approach such as the BFGS method should be done. Provided that all parameters of the truncated RM algorithm are well-adjusted, we are able to control the accuracy with just one parameter, namely the one for the stopping criteria $TOL_{\Delta\theta}$. In general, we control the accuracy for Newton-Raphson-based solvers by both convergence criteria, (4.18) and (4.19); a further free parameter is the number of paths N_4 used to replace the stochastic optimization task (4.9) by the deterministic optimization task (4.15). Figure 4.4 illustrates the convergence behavior for finding the optimal drift by both methods, the truncated RM algorithm and the Newton-Raphson method, for an increasing number of paths N_4; the underlying option is an OTM Bermudan Max call option, see Table 4.2 for parameter settings. Due to their working procedure both methods cannot be compared one by one in the sense that stochastic approximation methods belong to the class of online algorithms and algorithms for solving the deterministic optimization problem (4.15) belong to the class of offline algorithms, but we observe some vital facts. The red line shows the convergence of the truncated RM algorithm for a decreasing value of $TOL_{\Delta\theta}$. The convergence of the Newton-

Raphson method resulting from approximating the stochastic optimization problem (4.9) is illustrated by the blue line; here and in the following we use $TOL_{\Delta\theta} = 10^{-10}$ and $TOL_{\nabla f} = 10^{-10}$. At first glance, we observe a remarkable convergence behavior of the Newton-Raphson method. To find the drift minimizing variance, the Newton-Raphson method need significant fewer paths than the truncated RM algorithm; more precisely, the truncated RM method must be set up with $TOL_{\Delta\theta} = 10^{-15}$, i.e. 10,322,007 paths, and the Newton-Raphson method must be set up with only $N_4 = 3,500$ paths. Measuring the CPU time of both methods required to find the optimal drift gives us a speed-up factor of 552 to the credit of the Newton-Raphson solver. To underline this impressive

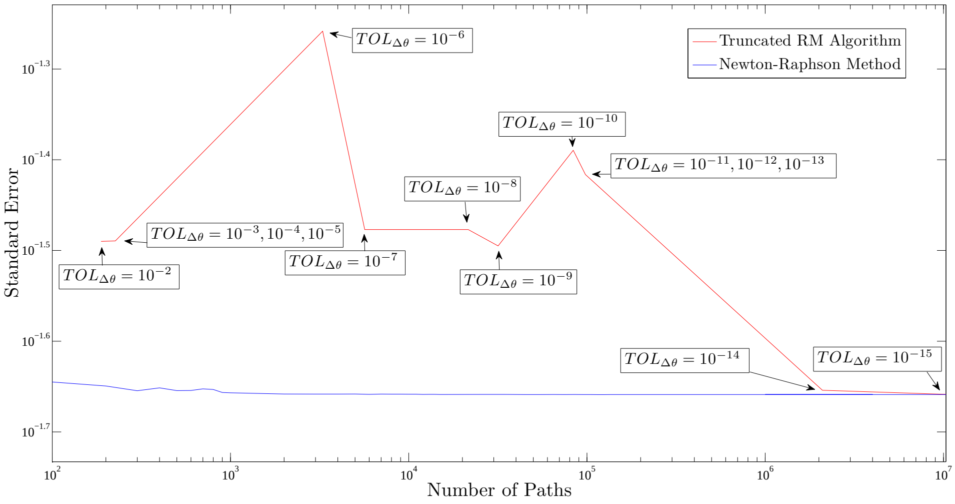

Figure 4.4: Convergence of the truncated RM algorithm and Newton-Raphson method for finding the drift minimizing variance.

speed-up factor, our general test for a comparative speed-accuracy study is as follows: To begin with, we determine the level of accuracy, i.e. the value of $TOL_{\Delta\theta}$, required by the truncated RM algorithm such that we do not recognize any more changes in the resulting variance. Then, for the Newton-Raphson-based solvers we determine the minimum number of paths needed to guarantee that we obtain the same minimal variance. Measuring the CPU times of both methods for this search procedure gives us speed-up factors denoted by CPU Time Ratio II in Tables 4.1 and 4.2. Moreover, let us introduce some quantities with respect to the CPU time to determine speed-up factors for a comparative study with and without our proposed variance reduction technique:

r_1 CPU time required to calculate the naive lower bound $\widehat{L}_0$

r_2 CPU time required to calculate $\widehat{\Delta}_0$ for creating the naive upper bound

$\widetilde{r}_1$ CPU time required to calculate the lower bound $\widehat{L}_0$ via importance sampling

$\widetilde{r}_2$ CPU time required to calculate $\widehat{\Delta}_0$ via importance sampling for creating the upper bound

$\widetilde{r}_3$ CPU time required to approximate the optimal drift

As we produce valid confidence intervals by using the AB method and both estimators, $\widehat{L}_0$ and $\widehat{\Delta}_0$, are independent of each other, we determine the number of replications required to achieve a confidence interval half width of ε_L for the lower estimator by

$$n_1 = \frac{z^2_{1-\alpha/2}\widehat{\sigma}^2_L}{\varepsilon^2_L}$$

and a confidence interval half width of ε_Δ for $\widehat{\Delta}_0$ by

$$n_2 = \frac{z^2_{1-\alpha/2}\widehat{\sigma}^2_\Delta}{\varepsilon^2_\Delta};$$

we denote the run lengths for the lower bounds by n_1 and $\widetilde{n}_1$ for the naive estimator and the estimator with importance sampling, respectively, and the run lengths for $\widehat{\Delta}_0$ by n_2 and $\widetilde{n}_2$ for the naive estimator and the estimator with importance sampling, respectively. Then, under the assumption that an early exercise region is given, we estimate speed-up factors by

$$\mathrm{SUF}_1 := \frac{r_1 n_1/N_2}{\widetilde{r}_1\widetilde{n}_1/N_2 + \widetilde{r}_3}$$

and

$$\mathrm{SUF}_2 := \frac{r_2 n_2/N_3}{\widetilde{r}_2\widetilde{n}_2/N_3 + \widetilde{r}_3}$$

for the lower bound and $\widehat{\Delta}_0$, respectively. In all our numerical experiments, we set $\varepsilon_L = \varepsilon_\Delta = 0.001$. In the remainder of this thesis, we denote by ISRoMo the truncated RM method for solving the stochastic optimization task (4.9); ISNew and ISQNew are the Newton-Raphson solver and the BFGS solver for finding the optimal drift via (4.15).

Table 4.1: Lower and upper bounds calculated by the AB approach combined with the LSM method and several importance sampling techniques for Bermudan arithmetic average call options on five assets.

	S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time Ratio I	CPU Time Ratio II	VR Ratio $L_0 \mid U_0$	$\mathrm{SUF}_1 \mid \mathrm{SUF}_2$
a)	90	LSM	0.428 $(3.87 \cdot 10^{-3})$	0.439 $(4.81 \cdot 10^{-3})$	[0.420,0.449]	–	–	–	–
		LSM + ISRoMo	0.422 $(8.59 \cdot 10^{-4})$	0.429 $(1.24 \cdot 10^{-3})$	[0.420,0.431]	0.93	–	20.32|15.14	18.66| 9.63
		LSM + ISNew	0.423 $(8.41 \cdot 10^{-4})$	0.430 $(1.16 \cdot 10^{-3})$	[0.421,0.432]	0.97	9.38	21.19|17.26	22.12|12.37
		LSM + ISQNew	0.423 $(8.41 \cdot 10^{-4})$	0.430 $(1.16 \cdot 10^{-3})$	[0.421,0.432]	0.97	9.83	21.19|17.26	22.14|12.37
	100	LSM	2.345 $(8.38 \cdot 10^{-3})$	2.370 $(9.00 \cdot 10^{-3})$	[2.329,2.388]	–	–	–	–
		LSM + ISRoMo	2.349 $(2.39 \cdot 10^{-3})$	2.377 $(4.70 \cdot 10^{-3})$	[2.344,2.387]	1.11	–	12.31|3.66	11.63| 0.79
		LSM + ISNew	2.348 $(2.37 \cdot 10^{-3})$	2.371 $(4.30 \cdot 10^{-3})$	[2.343,2.380]	1.16	51.88	13.05|4.38	13.05| 1.02
		LSM + ISQNew	2.348 $(2.37 \cdot 10^{-3})$	2.371 $(4.30 \cdot 10^{-3})$	[2.343,2.380]	1.16	55.23	13.05|4.38	13.06| 1.02
	110	LSM	8.443 $(1.17 \cdot 10^{-2})$	8.461 $(1.21 \cdot 10^{-2})$	[8.420,8.485]	–	–	–	–
		LSM + ISRoMo	8.471 $(4.03 \cdot 10^{-3})$	8.490 $(5.66 \cdot 10^{-3})$	[8.463,8.501]	1.14	–	8.36|4.57	8.89| 0.82
		LSM + ISNew	8.467 $(3.55 \cdot 10^{-3})$	8.480 $(4.98 \cdot 10^{-3})$	[8.460,8.490]	1.15	36.71	10.80|5.91	12.57| 1.02
		LSM + ISQNew	8.467 $(3.55 \cdot 10^{-3})$	8.480 $(4.98 \cdot 10^{-3})$	[8.460,8.490]	1.15	36.71	10.80|5.91	12.57| 1.02
b)	90	LSM	3.965 $(2.20 \cdot 10^{-2})$	4.006 $(2.32 \cdot 10^{-2})$	[3.921,4.052]	–	–	–	–
		LSM + ISRoMo	3.954 $(7.87 \cdot 10^{-3})$	4.006 $(1.09 \cdot 10^{-2})$	[3.938,4.028]	0.67	–	7.81|4.52	6.72|3.38
		LSM + ISNew	3.954 $(7.86 \cdot 10^{-3})$	4.013 $(1.04 \cdot 10^{-2})$	[3.938,4.033]	0.97	689	7.83|4.99	7.36|4.82
		LSM + ISQNew	3.954 $(7.86 \cdot 10^{-3})$	4.013 $(1.04 \cdot 10^{-2})$	[3.938,4.033]	0.97	689	7.83|4.99	7.36|4.82
	100	LSM	7.105 $(2.71 \cdot 10^{-2})$	7.210 $(3.29 \cdot 10^{-2})$	[7.052,7.274]	–	–	–	–
		LSM + ISRoMo	7.113 $(1.22 \cdot 10^{-2})$	7.196 $(1.54 \cdot 10^{-2})$	[7.089,7.226]	0.45	–	4.95|4.54	4.94|3.93
		LSM + ISNew	7.118 $(1.08 \cdot 10^{-2})$	7.202 $(1.35 \cdot 10^{-2})$	[7.097,7.228]	1.03	7378	6.32|5.95	6.51|5.43
		LSM + ISQNew	7.118 $(1.08 \cdot 10^{-2})$	7.202 $(1.35 \cdot 10^{-2})$	[7.097,7.228]	1.03	7378	6.32|5.95	6.51|5.43
	110	LSM	11.968 $(3.05 \cdot 10^{-2})$	12.115 $(3.34 \cdot 10^{-2})$	[11.908,12.180]	–	–	–	–
		LSM + ISRoMo	12.006 $(1.41 \cdot 10^{-2})$	12.128 $(1.95 \cdot 10^{-2})$	[11.978,12.166]	1.03	–	4.65|2.92	5.45|1.25
		LSM + ISNew	11.981 $(1.29 \cdot 10^{-2})$	12.115 $(1.76 \cdot 10^{-2})$	[11.956,12.149]	1.13	820	5.59|3.59	6.43|1.48
		LSM + ISQNew	11.981 $(1.29 \cdot 10^{-2})$	12.115 $(1.76 \cdot 10^{-2})$	[11.956,12.149]	1.13	820	5.59|3.59	6.43|1.48

Notes Table 4.1. See Table 3.3 a) and Table 3.3 c) for option parameters for a) and b), respectively. Algorithm specific parameters are as follows: $N_0 = 50,000$, $N_1 = 200,000$, $N_2 = 1,000$, $N_3 = 1,500$. To achieve the drift minimizing variance, the truncated RM method must be set up as follows: a) $TOL_{\Delta\theta} = 10^{-11}$ for $S_0 = 90$, $TOL_{\Delta\theta} = 10^{-12}$ for $S_0 = 100$, $TOL_{\Delta\theta} = 10^{-11}$ for $S_0 = 110$; b) $TOL_{\Delta\theta} = 10^{-13}$ for $S_0 = 90$, $TOL_{\Delta\theta} = 10^{-14}$ for $S_0 = 100$, $TOL_{\Delta\theta} = 10^{-13}$ for $S_0 = 110$. To get the optimal drift by solving the deterministic optimization task (4.15), the optimization solvers must be set up as follows: a) $N_4 = 1,500$ for $S_0 = 90$, $N_4 = 1,000$ for $S_0 = 100$, $N_4 = 700$ for $S_0 = 110$; b) $N_4 = 500$ for $S_0 = 90$, $N_4 = 100$ for $S_0 = 100$, $N_4 = 100$ for $S_0 = 110$. CPU Time Ratio I coincides with CPU Time Ratio in Section 3.4; VR Ratio is the variance reduction ratio.

Table 4.1 reports lower and upper bounds calculated by the AB method combined with and without several importance sampling techniques for Bermudan arithmetic average call options on five assets; the early exercise strategy is calculated by the LSM method. At first glance, we see that we observe a significant variance reduction by changing the drift in Brownian motion; the VR ratios for the lower bounds are between 4.65 and 21.19 and for the upper bounds between 3.66 and 17.26. As expected, due to the construction, the effect of a change of drift for OTM options is stronger than for ATM and ITM options. Both approaches, the stochastic approximation approach realized by a truncated version of the RM method and the solution of the deterministic optimization problem based on drawing i.i.d. samples a priori, perform well for symmetric option parameters; the CPU time ratios for the whole procedure are of about the same size. Nevertheless, we observe that using the Newton-Raphson or quasi Newton-Raphson solver significantly speeds up the search procedure of the optimal drift; more precisely, the CPU time ratios II are between 9.38 and 36.71, whereby the quasi Newton-Raphson approach is slightly faster. Anyway, the CPU time ratios I for asymmetric option parameters show that searching the drift by the truncated RM algorithm is quite expensive; the CPU time ratios I are 0.67 and 0.45 for the OTM and ATM options, respectively, and the CPU time ratios II are between 689 and 7,378. All in all, we observe remarkable speed-up factors for the lower bounds for both methods; the speed-up factors for the lower bounds are between 4.94 and 22.14, whereby the Newton-Raphson-based approaches clearly dominate the truncated RM-based approach with respect to efficiency. Anyway, the speed-up factors for the upper bounds are between 1.02 and 12.37 for the Newton-Raphson-based method and between 0.79 and 9.63 for the truncated RM-based method. Thus, it pays to use our change of drift technique provided that we work with deterministic solvers; the use of the truncated RM algorithm even leads to a break-down in the sense that a variance reduction becomes quite inefficient for the ATM and ITM options in Table 4.1 a). Note that the high

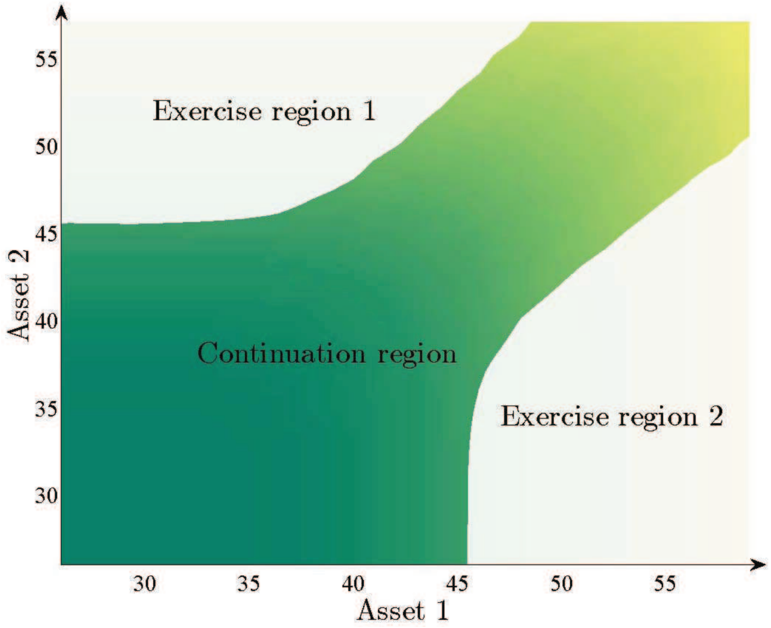

Figure 4.5: Early exercise regions for an American Max call option on two assets.

variance reduction factors for the upper bounds result from the high variance reduction of the lower bounds. However, our speed-accuracy analysis takes this fact into account. By the way, for all option parameter settings the drifts calculated by solving the deterministic optimization task seem to be of higher quality than the calculated drifts by the truncated RM algorithm. To get a better impression about the effect of changing the drift, let us consider the pricing of Max call options. Table 4.2 shows that we are able to reduce variance by using our importance sampling technique for Bermudan Max call options on five assets as well. We see that the deterministic solvers outperform the truncated RM algorithm; running our change of drift technique with stochastic approximation slows down the procedure of calculating lower and upper bounds. Nevertheless, for both methods we get speed-up factors between 2.15 and 2.40 for the lower bounds and between 0.80 and 2.16 for the upper bounds. We observe that the factors are lower than, e.g., for basket call options, and we cannot recognize an acceleration for the upper bound of the ATM option. Considering Figure 4.5 brings light in the dark. As we can see, there are two disjunct early exercise regions such that driving paths into these regions by constant drift parameters seems to be difficult. Therefore, our idea is to work with flexible drift parameters, i.e. state-dependent drifts. A possible realization for Max call options might be given by choosing a drift vector depending on the underlying payoff such as realized by

$$\theta_d = \begin{cases} \overline{\theta}_d, & \text{if } S^d = \max\{S^1, ..., S^D\} \\ \underline{\theta}_d, & \text{otherwise} \end{cases}, \quad d = 1, ..., D. \tag{4.21}$$

Table 4.2 b) reports the results of using (4.21), and we see that it makes sense to work with a flexible drift parameter for this option type. The more effort in using a flexible drift is negligible such that we are able to speed up convergence by factors between 3.61 and 3.90 for the lower bounds and we observe a further improvement for the upper bounds. In our experience, these test results are representative for finding the optimal drift, and, thus, we highly recommend to search the drift minimizing variance by replacing the stochastic optimization problem by its determinic counterpart for practical applications; this is the more robust way and clearly outperforms stochastic approximation in our framework. Both deterministic solvers, the Newton-Raphson method and the BFGS method, show a good performance and the user's favorite algorithm should be implemented. Although we have seen that we get a remarkable convergence improvement by our change of drift technique attributed by an efficient solution of the underlying optimization problem, let us conclude this section with a final remark:

Remark 4.1.3. *The dimension of optimization task (4.9) is linear in the dimension of the underlying SDE, and pricing financial derivatives with many uncertain factors leads to high-dimensional optimization problems. To reduce the dimension of the underlying optimization task, we might think about working with drifts which are independent of the factors; for instance, for Max call options we might set* $\underline{\theta}_d = \underline{\theta}$ *and* $\overline{\theta}_d = \overline{\theta}$ *for all assets* d*,* $d = 1, ..., D$*, in (4.21) such that we just work with two drift parameters.*

Table 4.2: Lower and upper bounds calculated by the AB approach combined with the LSM method and several importance sampling techniques for Bermudan Max call options.

	S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time	CPU Time	VR Ratio	SUF$_1$|SUF$_2$
						Ratio I	Ratio II	$L_0 \vert U_0$	
a)	90	LSM	27.522 ($3.34 \cdot 10^{-2}$)	27.687 ($3.67 \cdot 10^{-2}$)	[27.456,27.759]	–	–	–	–
		LSM + ISRoMo	27.532 ($2.19 \cdot 10^{-2}$)	27.655 ($2.50 \cdot 10^{-2}$)	[27.489,27.703]	0.65	–	2.15|2.18	2.15|1.49
		LSM + ISNew	27.525 ($2.19 \cdot 10^{-2}$)	27.676 ($2.51 \cdot 10^{-2}$)	[27.482,27.725]	0.98	552	2.16|2.10	2.16|1.48
		LSM + ISQNew	27.525 ($2.19 \cdot 10^{-2}$)	27.676 ($2.51 \cdot 10^{-2}$)	[27.482,27.725]	0.98	597	2.16|2.10	2.16|1.48
	100	LSM	37.816 ($3.82 \cdot 10^{-2}$)	37.839 ($3.86 \cdot 10^{-2}$)	[37.741,37.914]	–	–	–	–
		LSM + ISRoMo	37.817 ($2.50 \cdot 10^{-2}$)	38.035 ($3.16 \cdot 10^{-2}$)	[37.768,38.097]	0.86	–	2.33|1.76	2.24|0.80
		LSM + ISNew	37.824 ($2.49 \cdot 10^{-2}$)	38.052 ($3.10 \cdot 10^{-2}$)	[37.775,38.113]	1.02	290	2.34|1.82	2.27|0.87
		LSM + ISQNew	37.824 ($2.49 \cdot 10^{-2}$)	38.052 ($3.10 \cdot 10^{-2}$)	[37.775,38.113]	1.02	307	2.34|1.82	2.27|0.87
	110	LSM	49.256 ($4.24 \cdot 10^{-2}$)	49.567 ($5.30 \cdot 10^{-2}$)	[49.172,49.671]	–	–	–	–
		LSM + ISRoMo	49.277 ($2.76 \cdot 10^{-2}$)	49.574 ($3.51 \cdot 10^{-2}$)	[49.223,49.643]	0.74	–	2.36|2.29	2.39|2.16
		LSM + ISNew	49.277 ($2.76 \cdot 10^{-2}$)	49.558 ($3.52 \cdot 10^{-2}$)	[49.223,49.627]	1.01	665	2.36|2.28	2.40|2.16
		LSM + ISQNew	49.277 ($2.76 \cdot 10^{-2}$)	49.558 ($3.52 \cdot 10^{-2}$)	[49.223,49.627]	1.01	690	2.36|2.28	2.40|2.16
b)	90	LSM +ISNew2	27.528 ($1.73 \cdot 10^{-2}$)	27.660 ($2.08 \cdot 10^{-2}$)	[27.494,27.701]	0.97	–	3.72|3.11	3.61|1.76
	100	LSM +ISNew2	37.821 ($1.99 \cdot 10^{-2}$)	38.018 ($2.55 \cdot 10^{-2}$)	[37.782,38.068]	1.01	–	3.67|2.70	3.76|1.22
	110	LSM +ISNew2	49.262 ($2.21 \cdot 10^{-2}$)	49.536 ($3.01 \cdot 10^{-2}$)	[49.219,49.595]	1.03	–	3.68|3.09	3.90|2.45

Notes. See Table 3.3 b) for option parameters and algorithm settings. To achieve the drift minimizing variance, the truncated RM method must be set up with $\varepsilon = 10^{-15}$ for all options. To get the optimal drift by solving the deterministic optimization task (4.15), the optimization solvers must be set up as follows: $N_4 = 3,500$ for $S_0 = 90$, $N_4 = 2,500$ for $S_0 = 100$, $N_4 = 1,800$ for $S_0 = 110$.

4.2 Quasi-Monte Carlo Methods

Quasi-random numbers are a widely-spread tool for increasing the efficiency of estimators. To start with, we introduce these sequences with low discrepancy in Subsection 4.2.1. As our ultimate goal is to price options in higher dimensions with quasi-Monte Carlo (QMC) techniques, we consider the randomization of quasi-random numbers as a way of measuring errors in Subsection 4.2.2. It is well known that the quality of sequences with low discrepancy becomes poor for higher dimensions, and, therefore, in Subsection 4.2.3 we consider dimensionality reduction techniques to increase the performance. Finally, we investigate the effect of using QMC tools for pricing high-dimensional options with an early exercise feature in Subsection 4.2.4.

4.2.1 Sequences with Low Discrepancy

The ultimate goal of QMC integration is to evaluate the integral of a function f

$$\mu := \int_{[0,1]^D} f(u)du \tag{4.22}$$

by the estimate

$$\overline{\mu} := \frac{1}{N}\sum_{n=1}^{N} f(Q_n)$$

with the deterministic point set $Q = \{Q_1, ..., Q_N\},\ Q_n \in [0,1]^D,\ n = 1, ..., N$ – rather than using a point set $P = \{U_1, ..., U_N\}$ with uniformly distributed random variables $U_1, ..., U_N$ such as in ordinary Monte Carlo integration. In so doing, the requirement on the point set Q is high, as we want to beat the probabilistic convergence rate of MC integration $\mathcal{O}(1/\sqrt{N})$. It is desirable to work with points $Q_1, ..., Q_N$ that are evenly distributed over $[0,1]^D$. The discrepancy of point sets is a measure of uniformity and is defined as follows:

Definition 4.2.1 ((Star) Discrepancy). *Let $\mathcal{R}$ be the set of all rectangles in $[0,1)^D$ of the form $\prod_{d=1}^{D}[y_d, z_d),\ 0 \leq y_d < z_d \leq 1$, and $\sharp\{x_i | x_i \in R\}$ be the number of points x_i contained in $R,\ R \subseteq \mathcal{R}$. Then, the discrepancy of a point set Q is defined as*

$$D_N(Q, \mathcal{R}) := \sup_{R \subseteq \mathcal{R}} \left| \frac{\sharp\{x_i | x_i \in R\}}{N} - vol(R) \right|,$$

where $vol(R)$ is the volume of R. Furthermore, let $\mathcal{R}^*$ be the set of all rectangles in $[0,1)^D$ of the form $\prod_{d=1}^{D}[0, y_d),\ 0 < y_d \leq 1$. Then, according to D_N, the star discrepancy D_N^* is given by

$$D_N^* := D_N(Q, \mathcal{R}^*). \tag{4.24}$$

The star discrepancy is a vital quantity for error bounds of QMC integration; for instance, provided that the function f satisfies some regularity assumptions, the Koksma-Hlawka inequality is given as follows:

Theorem 4.2.1 (Koksma-Hlawka Inequality). *Suppose that the integrand f in (4.22) has bounded variation $V(f)$ on $[0,1)^D$ in the sense of Hardy and Krause. Then, the inequality*

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(Q_n) - \int_{[0,1)^D} f(u) du \right| \leq V(f) D_N^* \tag{4.25}$$

holds for any point set Q.

Proof. See [69] for the original work in German; a proof can also be found in [102]. □

For an overview of error bounds we refer the reader to [102]. Results like (4.25) motivate for finding sequences with low discrepancy, i.e. sequences characterized by

$$D_N^* = \mathcal{O}\left(\frac{(\log N)^D}{N} \right).$$

Even though the practibility of results like the Koksma-Hlawka inequality is problematic, a number of studies report a great efficiency increase by working with quasi-random numbers, see the cited papers in this section. There are a variety of sequences fulfilling criteria (4.24), e.g., the Halton sequence, the Faure sequence, the Niederreiter sequence or the Sobol sequence, to mention just a few of them. The construction of all these sequences are often closely related, and the concept of (t, m, s)-nets plays a key role, see [102]. Let us focus our attention on the Halton and Sobol sequences. No doubt, the construction of the Halton sequence is quite simple compared with the construction of the Sobol sequence. Anyway, the common believe is that the Sobol sequence performs well, see the references later on, and, therefore, we should include it in our comparative study.

Definition 4.2.2 (Radical Inverse Function). *Consider the b-base expansion of any integer n with $b \geq 2$,*

$$n = \sum_{k=0}^{j} \alpha_k b^k, \;\; \alpha_k \in \{0, ..., b-1\}.$$

Then, the radical inverse function of n is given by

$$\varphi_b(n) := \sum_{k=0}^{j} \alpha_k b^{-k-1}.$$

Thus, a stepwise refinement of the mesh is achieved with an increasing number of digits j in n. In 1960 Halton [61] proposed the following sequence with low discrepancy:

Definition 4.2.3 (Halton Sequence)**.** *Let* $p_1, ..., p_D$ *be relatively prime integers greater than 1. Then, the Halton sequence is defined as*

$$Q_n = (\varphi_{p_1}(n), ..., \varphi_{p_D}(n)), \;\; n = 1, 2, ...$$

Actually, the Halton sequence is a slight modification of the point set introduced by Hammersley [62].

Definition 4.2.4 (Hammersley's Point Set)**.** *For any* $N \in \mathbb{N}$*, the points* $Q_n, \;\; n = 1, ..., N$*, of the Hammersley set* $Q = \{Q_1, ..., Q_N\}$ *are given by*

$$Q_n = (n/N, \varphi_{p_1}(n), ..., \varphi_{p_D}(n)), \;\; n = 1, ..., N.$$

To construct the Hammersley point set, we have to set up the number of points N a priori; this fact might be seen as a drawback for practical applications such that Halton's sequence is often preferred. However, using prime integers as bases ensures that the hypercube is completely filled. Originally, Hammersley's point set is an extension of the van der Corput sequence $(n/N, \varphi_2(n)), \;\; n = 1, 2, ...,$ proposed in [128]. Figure 4.6 illustrates the first 2,000 points of the Halton sequence in dimension 40. We see the typical behavior of this sequence in lower and higher dimensions; more precisely, the points in dimensions 1 and 2 produced by the bases 2 and 3, respectively, cover the space more evenly than the points in dimensions 39 and 40 produced by bases 167 and 173, respectively. The correlation of coordinates in higher dimensions is a well-known phenomenon and is caused by the van der Corput sequence: In general, if we examine coordinates produced by large primes p_{d_1} and p_{d_2} with $d_1 < d_2$, the unit square is filled in cycles of length p_{d_2}, and points are clustered into lines as indicated in the right panel of Figure 4.6. In financial

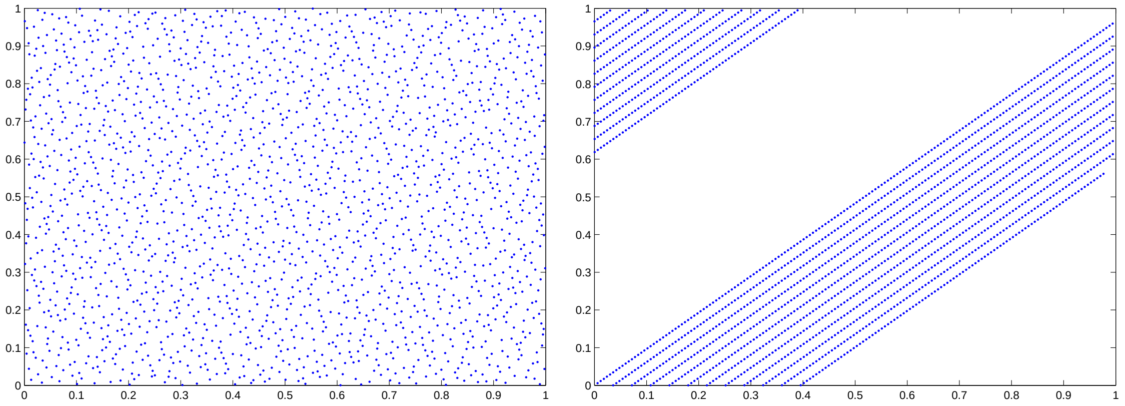

Figure 4.6: First 2,000 Halton points in dimension 40 projected onto the first two coordinates (bases $p_1 = 2$ and $p_2 = 3$) and onto the last two coordinates (bases $p_{39} = 167$ and $p_{40} = 173$) in the left and right panel, respectively.

applications, the dimensions of the underlying problems are often high, and, therefore, other ways of generating deterministic point sets should be considered. For instance, the illustrative proof that the Sobol sequence [120] promises a better behavior in higher dimensions is given in Figure 4.7; we observe more evenly distributed points and even in higher dimensions we recognize just slight patterns. Based on the Gray code, Antonov and Saleev [6] proposed an efficient implementation of the Sobol sequence, see also [22] for details; we run our codes with initial direction numbers of [74]. To overcome the deficiency of the Halton sequence in higher dimensions, Kocis and Whiten [82] proposed the Halton sequence leaped, which can be described as follows: The green points in the left panel of Figure 4.8 are the first 358,000 points of the Halton sequence in dimension 40 projected onto the last two coordinates (bases 167 and 173). The idea is now to construct a sequence consisting of only every $\mathcal{L}$-th point of the original Halton sequence; for instance, every 179-th point is colored black in the left panel and these points build the elements of the Halton sequence leaped. By doing so, we try to enforce more uniformity compared to the original sequence. This simple but obviously effective modification of the Halton sequence is numerically described as follows:

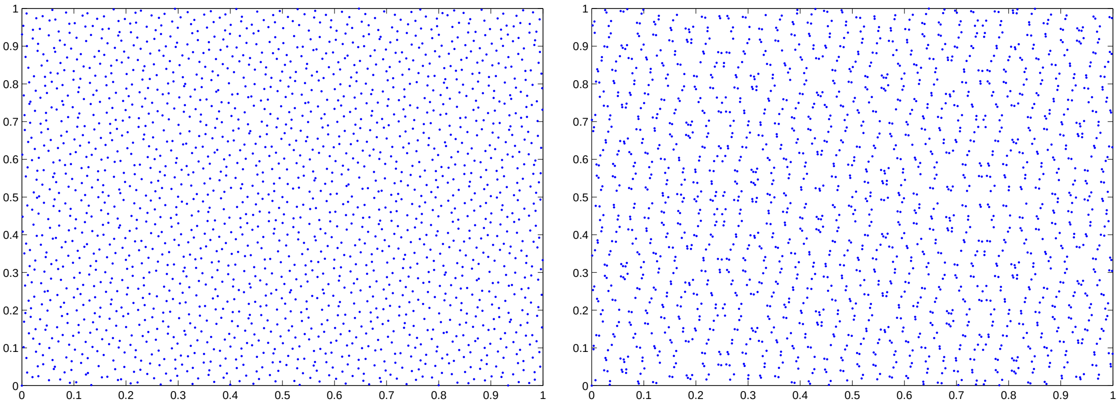

Figure 4.7: First 2,000 Sobol points in dimension 100 projected onto the first two coordinates (dimensions 1 and 2) and the last two coordinates (dimensions 99 and 100).

Definition 4.2.5 (Halton Sequence Leaped). *In accordance with the original Halton sequence, Definition 4.2.3, the Halton sequence leaped is defined as*

$$Q_n = (\varphi_{p_1}(n\mathcal{L}), ..., \varphi_{p_{\tilde{D}}}(n\mathcal{L})),\ n = 1, 2, ...$$

with the leap $\mathcal{L}$, where $\mathcal{L}$ is a positive integer relatively prime to the bases $p_1, ..., p_D$.

Kocis and Whiten find that the Halton sequence leaped can be interpreted as a generalized Halton sequence [21], [67], because leaping has the same effect as permuting the digits of the Halton sequence. Despite the hope of producing quasi-random numbers with higher quality than the original sequence by this simple modification, the leap $\mathcal{L}$ should be chosen carefully. Figure 4.9 warns us to select any leap value; poor values of $\mathcal{L}$ lead to the same dilemma as with the original sequence. Therefore, by minimizing the integration error of some test functions for a range of dimensions D and number of sample points N, Kocis and Whiten suggested good values for $\mathcal{L}$; e.g., 31, 61, 149, 409 and 1949 are recommended for the ranges $D \in [1, 400]$ and $N \in [10, 10^5]$. The common believe is that points of the

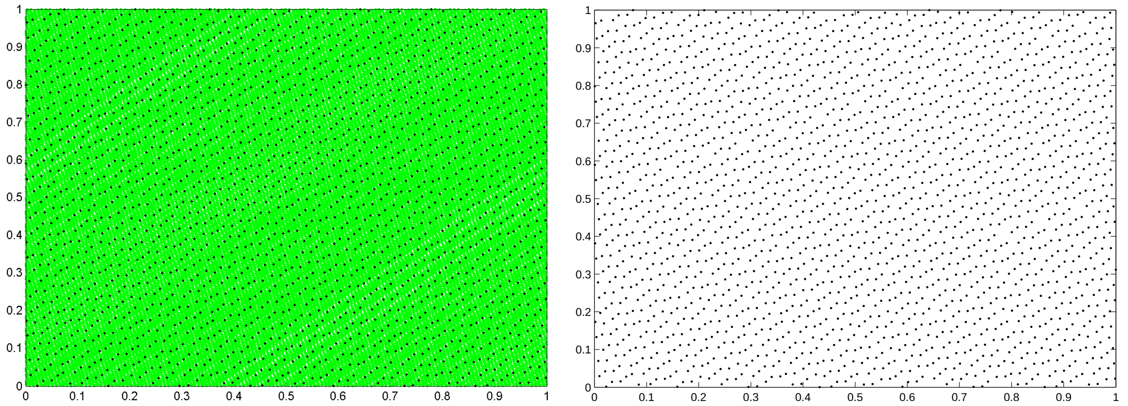

Figure 4.8: Construction of the 40-dimensional Halton sequence leaped with leap 179. Left panel shows the projection of the first 358,000 Halton points onto the last two dimensions and right panel shows the first 2,000 points of the Halton sequence leaped onto the last two dimensions.

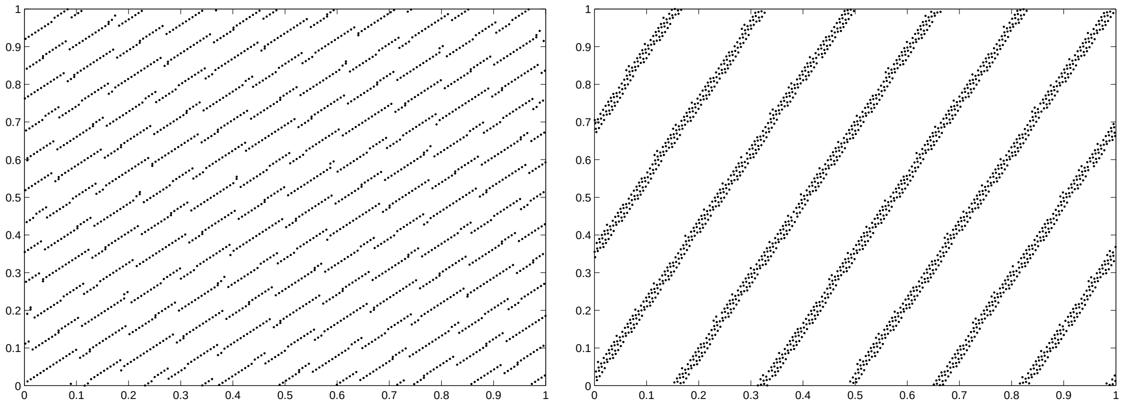

Figure 4.9: First 2,000 points of the Halton sequence leaped in dimension 40. Left and right panels show the projection of the Halton sequence leaped with $\mathcal{L} = 233$ and $\mathcal{L} = 269$, respectively, onto the last two coordinates.

Sobol sequence are of high quality, and, thus, they are used as the ultimate quasi-random numbers for empirical studies, see [31], [89], [44], [58], among others. Nevertheless, as we are not familiar with any study investigating the performance of the Halton sequence leaped in computational finance, we will include this slight modification of the Halton squence in our experiments in Subsection 4.2.4. At this point we want to stress out that comparative studies with respect to Monte Carlo and quasi-Monte Carlo methods should be taken with a pinch of salt, as the underlying pseudo-random number generator plays a key role in a comparative study; for instance, the well-working Mersenne Twister generator [95] was proposed in 1998 and a number of studies were published around this year, see [101] for a chronology of some studies. We might tackle the problem of correlations between higher dimensions in two ways. On the one hand, we might try to construct sequences of high quality in higher dimensions as well. This is the topic of recent research and is out of the scope of this thesis; we refer the interested reader to [101] and the references therein. Anyway, by investigating the performance of the Halton sequence leaped, empirically, we will do a first step in this direction. Alternatively, we might try to evade the dilemma by reducing the effective dimensionality of the problem itself. This is exactly the topic of Subsection 4.2.3. Before we introduce such techniques, we proceed with randomized quasi-random numbers as a possibility to measure errors. Let us conclude this subsection with a final technical remark:

Remark 4.2.1. *Numerically, inversion methods realize the relation* $Z_n = F^{-1}(U_n)$ *to transform a sequence of independent random variables* U_n, $n = 1, 2, \ldots$, *uniformly distributed on* $[0, 1]$ *into independent standard normally distributed random variables* Z_n, $n = 1, 2, \ldots$; F^{-1} *denotes the inverse of the standard normal distribution. In order to preserve the structure of the point set of quasi-random numbers, we should use these methods for generating normally distributed quasi-random numbers rather than rejection-acceptance methods; rejecting points out of our given point set destroys the structure and, hence, all the trouble for constructing points with good properties was for nothing. Following this way entails the numerical approximation of* F^{-1}, *and an efficient algorithm with an absolute error of* 3×10^{-9} *for up to seven standard deviations was proposed by Moro in [98]; especially, to achieve higher accuracy than the normal inversion algorithm by Beasley and Springer [10], Moro's method approximates the tails by truncated Chebyshev series.*

4.2.2 Randomization

As previously mentioned, even if the function f in (4.22) satisfies the required regularity conditions, it is often hard to calculate bounds on the deterministic error. Provided that we are in such a situation, the calculated bounds are often too conservative, see [88]. For this purpose, randomizing quasi-random numbers enables us to get error estimates, which are more practical than using bounds like the Koksma-Hlawka inequality (4.25). This process of making sequences random combine the good features of both techniques, the high accuracy of quasi-random numbers and the ability to estimate errors of Monte Carlo integration. By doing so, we take the possible loss of precision to get a more practical way of measuring errors. In the following, we denote by $Q = \{Q_1, ..., Q_N\}$ a set of N points $Q_n,\ n = 1, ..., N$, in $[0,1]^D$, where the elements of Q are quasi-random points. To get randomized quasi-random numbers, the following two widely-used techniques might be applied:

Definition 4.2.6 (Random Shift Modulo 1). *Let U be uniformly distributed in $[0,1)^D$. Based on the set Q, a new set of N points $\widetilde{Q} = \{\widetilde{Q}_1, ..., \widetilde{Q}_N\}$ with $\widetilde{Q}_n \in [0,1)^D$ is given by*

$$\widetilde{Q}_{nd} := (Q_{nd} + U_d)\text{mod}1,\ d = 1, ..., D,\ n = 1, ..., N.$$

Cranley and Patterson [37] pioneered this random shift modulo 1 (RSM1) technique for randomizing points produced by lattice rules. Tuffin [127] suggested to use their approach in combination with low-discrepancy sequences. Another but similar method is the random digital shift in base b described by:

Definition 4.2.7 (Random Digital Shift in Base b). *Let U be uniformly distributed in $[0,1)^D$. By writing each U_d and Q_{nd} in its base b expansion, i.e.*

$$U_d = \sum_{k=0}^{j} \alpha_k b^{-k-1},\ d = 1, ..., D,$$

and

$$Q_{nd} = \sum_{k=0}^{j} \widetilde{\alpha}_{nk} b^{-k-1},\ d = 1, ..., D,\ n = 1, ..., N,$$

respectively, a new set of N points $\widetilde{Q} = \{\widetilde{Q}_1, ..., \widetilde{Q}_N\}$ with $\widetilde{Q}_n \in [0,1)^D$ is defined by

$$\widetilde{Q}_{nd} = \sum_{k=0}^{j} ((\widetilde{\alpha}_{nk} + \alpha_k)\text{mod}b) b^{-k-1},\ d = 1, ..., D,\ n = 1, ..., N.$$

Notify that the random digital shift in base b might be easily realized by a bitwise exclusive-or operation for $b = 2$. We refer the interested reader to [88] or [58] for an overview of other techniques; [88] also covers theoretical results regarding randomized deterministic point sets.

For any positive integer p, let $\widetilde{Q} = \{\widetilde{Q}_1, ..., \widetilde{Q}_p\}$ be a set of p randomized quasi-random points. Then, we are able to get an estimate of (4.22) by

$$\widehat{Q}_k = \frac{1}{p} \sum_{i=1}^{p} f(\widetilde{Q}_i).$$

The trick is now to repeat this procedure, say q times, such that $\widehat{Q}_k,\ k = 1, ..., q$, are i.i.d. – this is realized by generating a new pseudo-random number U for each trial – and to estimate (4.22) by

$$\widehat{Q} = \frac{1}{q}\sum_{k=1}^{q}\widehat{Q}_k.$$

Following this procedure, a valid $(1-\alpha)$ confidence interval might be determined by

$$\left[\widehat{Q} - t_{q-1,1-\alpha/2}\frac{\widehat{\sigma}_q}{\sqrt{q}}, \widehat{Q} + t_{q-1,1-\alpha/2}\frac{\widehat{\sigma}_q}{\sqrt{q}}\right],$$

where $t_{q-1,1-\alpha/2}$ denotes the $(1-\alpha/2)$ quantile of the Student's t distribution with $q-1$ degrees of freedom (or simply the t_{q-1} distribution) and $\widehat{\sigma}_q$ is the sample error given by

$$\widehat{\sigma}_q = \sqrt{\frac{1}{q-1}\sum_{k=1}^{q}(\widehat{Q}_k - \widehat{Q})^2}.$$

For the sake of completeness, the t_q distribution has the probability density function

$$f(x) = \frac{\left(\frac{q-1}{2}\right)!}{\sqrt{q\pi}\left(\frac{q-2}{2}\right)!}\left(1+\frac{x^2}{q}\right)^{-\frac{q+1}{2}}$$

and converges to the standard normal distribution as $q \to \infty$. Hence, for sufficiently large values of q, it is reasonable to replace $t_{q-1,1-\alpha/2}$ by $z_{1-\alpha/2}$; compare also (2.27). Let us briefly address the adequate choice of the quantities p and q. In practical applications, q might be chosen small, e.g. $q < 25$, such that the total number $N = pq$ is dominated by the value of p. By this choice, much of the accuracy resulting from using random numbers with low discrepancy is preserved, and, therefore, we expect to obtain a good estimate of (4.22). Anyway, if a comparative study between an ordinary Monte Carlo approach and a (randomized) quasi-Monte Carlo approach has top priority, we should increase the value of the number of trials q, say $q \geq 30$, to draw more meaningful conclusions. The reason for this choice is that we get a more accurate estimation of the sample deviation by using a richer set of test trials, see [130] for a statistical treatment of these topics. We address that point in more detail in our numerical experiments when we determine variance reduction factors. This section aims at producing lower and upper bounds as well as resulting confidence intervals under the quasi-Monte Carlo framework. Focusing on the AB approach, we are able to produce valid confidence intervals by

$$\left[\widehat{L}_0^{q_1} - t_{q_1-1,1-\alpha/2}\frac{\widehat{\sigma}_L}{\sqrt{q_1}}, \widehat{L}_0^{q_1} + \widehat{\Delta}_0^{q_2} + t_{q-1,1-\alpha/2}\sqrt{\frac{\widehat{\sigma}_L^2}{q_1} + \frac{\widehat{\sigma}_\Delta^2}{q_2}}\right] \tag{4.26}$$

with $q = \min\{q_1, q_2\}$; $\widehat{L}_0^{q_1}$ and $\widehat{\Delta}_0^{q_2}$ are the estimates (2.21) and (2.25) of L_0 and Δ_0 based on q_1 and q_2 trials, respectively. To ensure the validity of these intervals, we should choose $q_1, q_2 \geq 30$. Needless to mention, to generate normally distributed random numbers, we should not differ from using inverse methods, compare Remark 4.2.1. To permit a comparison with the results by [31], we studied in [76] the effect of using quasi-random numbers by permuting coordinates. Anyway, we see in randomization a good alternative for practical applications; therefore, we neglect the approach of permuting dimensions in this thesis.

4.2.3 Dimensionality Reduction

First practical applications in computational finance could improve convergence by using quasi-random numbers rather than pseudo-random numbers, see [109], [78], [1], among others. To explain this phenomenon, Caflisch et al. [27] introduced the concept of the effective dimension of a problem. The common believe is that many problems in mathematical finance have a low effective dimension, i.e. the number of important dimensions of the quasi-random sequence is much lower than the true dimension; we refer the interested reader to [88] for a detailed treatment. To reach a low effective dimension, we try to reformulate our underlying problem with the goal that larger fractions of the variance are explained by the first few dimensions of the sequence with low discrepancy. Let us concretize this point by considering two tools promising an effective dimensionality reduction in computational finance. The common way to simulate a standard Brownian motion is to use the random walk recursion (RWR)

$$\begin{pmatrix} W_1 \\ \vdots \\ W_L \end{pmatrix} = \begin{pmatrix} a_{11} \\ \vdots \\ a_{L1} \end{pmatrix} Z_1 + \ldots + \begin{pmatrix} a_{1L} \\ \vdots \\ a_{LL} \end{pmatrix} Z_L$$

where $Z_l \sim \mathcal{N}(0,1),\ l = 1, ..., L$, and $a_l = (a_{1l}, ..., a_{Ll})^T$ are the columns of the matrix

$$A = \begin{pmatrix} \sqrt{dt} & 0 & 0 & \ldots & 0 \\ \sqrt{dt} & \sqrt{dt} & 0 & \ldots & 0 \\ \vdots & & & & 0 \\ \sqrt{dt} & \sqrt{dt} & \sqrt{dt} & \ldots & \sqrt{dt} \end{pmatrix} \tag{4.27}$$

such that $AZ \sim N(0, \Sigma_1)$ with $AA^T = \Sigma_1$. As mentioned in Subsection 4.2.1, the first dimensions of sequences with low discrepancy have better properties than the higher dimensions. Therefore, the idea is to generate much of the shape of each path by the first dimensions being of high quality. In the following we describe two techniques, the Brownian bridge construction (BBC) and the principal component construction (PCC), for generating paths such that most of the variability is explained by the first dimensions of a quasi-random sequence.

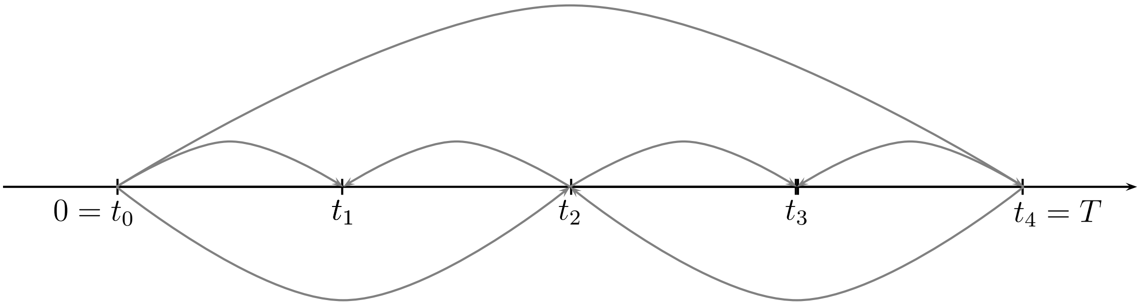

Figure 4.10: Path construction by BBC for four time steps.

Brownian Bridge Construction Let t_k, t_{k+1}, t_l be any time dates in $[0, t_L]$ such that $t_k = t_l - \Delta t_1$ and $t_{k+1} = t_l + \Delta t_2$ with $\Delta t_1, \Delta t_2 > 0$. Given that $W_k = w_k$ and $W_{k+1} = w_{k+1}$ are known, we are able to generate paths of a standard Brownian motion via

$$W_l = \frac{\Delta t_2 w_k + \Delta t_1 w_{k+1}}{\Delta t_1 + \Delta t_2} + \sqrt{\frac{\Delta t_1 \Delta t_2}{\Delta t_1 + \Delta t_2}} Z, \; Z \sim \mathcal{N}(0,1). \tag{4.28}$$

In so doing, we are free in simulating paths of the Brownian motion in any time order. This is a valid construction and is reasoned as follows: Let $t_0, ..., t_K$ be any time dates with $t_0 = 0$. Then, for any time date t_l between t_k and t_{k+1}, the distribution of W_l conditional on $(W_0 = 0, ..., W_K = w_K)$ is known is given by

$$\begin{aligned}(W_l | W_0 = 0, ..., W_K = w_K) &= (W_l | W_k = w_k, W_{k+1} = w_{k+1}) \\ &\sim \mathcal{N}\left(\frac{\Delta t_2 w_k + \Delta t_1 w_{k+1}}{\Delta t_1 + \Delta t_2}, \frac{\Delta t_1 \Delta t_2}{\Delta t_1 + \Delta t_2}\right)\end{aligned}$$

see [80] or [119]. In order to get a better view of this technique, Figure 4.10 illustrates the construction of BBC for $L = 4$. Notify that (4.28) can be easily extended to simulate multi-dimensional processes for which the variability is explained by Brownian motions. The gain of using BBC rather than RWR is that the variance is concentrated into large time steps. Thus, the first dimensions of a quasi-random sequence control much of the structure of the generated path, and, needless to mention, this has the effect of getting a lower effective dimensionality. Caflisch and Moskowitz [99] pioneered the use of BBC in combination with quasi-Monte Carlo approaches, and a number of reseachers successfully applied BBC for pricing financial derviatives with quasi-Monte Carlo estimators, see [1], [31], [44], to mention just a few.

Principal Component Construction The symmetric time covariance matrix Σ_1 might be decomposed by the spectral decomposition such that

$$\Sigma_1 = V \Lambda V^T,$$

where V is an orthonormal matrix and Λ is a diagonal matrix; the columns of V are the normalized eigenvectors $v_1, ..., v_L$ of Σ_1 and the elements of Λ are the real eigenvalues $\lambda_1, ..., \lambda_L$ of Σ_1. Then, to generate paths of the Brownian motion, we might set up $A = V\Lambda^{1/2}$ rather than using (4.27). Let us assume that the columns of A are ordered with respect to the eigenvalues λ_l, i.e. $\lambda_1 \geq ... \geq \lambda_L$. In general, the principal component analysis tells us that the optimal lower dimensional approximation to W for which the variance of the projection is maximized is given by the k eigenvectors $v_1, ..., v_k$ of Σ_1 corresponding to the k largest eigenvalues $\lambda_1, ..., \lambda_k$, see [14]; more precisely, according to [58], the mean square approximation error

$$\mathbb{E}\left[\|W - \sum_{l=1}^{k} a_l Z_l\|_2^2 \right]$$

is minimized by setting $a_l = \sqrt{\lambda_l} v_l$ and $Z_l = v_l^T W / \sqrt{\lambda_l}$ for any $k = 1, ..., L$, and the explained variance is given by the fraction

$$\frac{\lambda_1 + ... + \lambda_k}{\lambda_1 + ... + \lambda_L}.$$

In case of constant time step width, the eigenvalues and eigenvectors of Σ_1 are given by

$$\lambda_l = 0.25 \Delta t \sin^{-2}\left(\frac{2l-1}{2L+t} 0.5\pi\right), \; l = 1, ..., L,$$

and

$$v_{lk} = \frac{2}{\sqrt{2L+1}} \sin\left(\frac{2l-1}{2L+1} k\pi\right), \; k, l = 1, ..., L,$$

respectively, see [2]. Let us fix for a moment the time step t_l. Realizations of a multi-dimensional Brownian motion with covariance matrix Σ_2 at t_l might be generated by

$$\begin{pmatrix} W_l^1 \\ \vdots \\ W_l^D \end{pmatrix} = \begin{pmatrix} b_{11} \\ \vdots \\ b_{D1} \end{pmatrix} Z_1 + \ldots + \begin{pmatrix} b_{1D} \\ \vdots \\ b_{DD} \end{pmatrix} Z_D$$

with $Z_d \sim \mathcal{N}(0,1)$, $d = 1, ..., D$, and an appropriate chosen $(D \times D)$ matrix B. Numerically, we are used to work with the Cholesky decomposition to obtain B such that $BB^T = \Sigma_2$ with $W \sim \mathcal{N}(0, \Sigma_2)$. Using this framework for simulating multi-dimensional Brownian motions might be seen critical, as, in general, Cholesky factorizations for positive semi-definite matrices are not unique. With the same argumentation as above, denoting by μ_d the D eigenvalues with corresponding normalized eigenvectors w_d of Σ_2, the matrix B can also be defined by the D columns $b_d = \sqrt{\mu_d} w_d$, $d = 1, ..., D$, due to the factorization $\Sigma_2 = WMW^T$ with $W = [w_1, ..., w_D]$ and $M = \text{diag}(\mu_1, ..., \mu_D)$. Let us now consider the vital case of simulating paths of multidimensional Brownian motions with the help of the principal component analysis. Combining the steps above justifies the construction

$$\begin{pmatrix} W_1^1 \\ \vdots \\ W_1^D \\ \vdots \\ W_L^1 \\ \vdots \\ W_L^D \end{pmatrix} = \sum_{l=1}^{L}\sum_{d=1}^{D} \sqrt{\lambda_l}\sqrt{\mu_d} \begin{pmatrix} v_{1l}\begin{pmatrix} w_{1d} \\ \vdots \\ w_{Dd} \end{pmatrix} \\ \vdots \\ v_{Ll}\begin{pmatrix} w_{1d} \\ \vdots \\ w_{Dd} \end{pmatrix} \end{pmatrix} Z_{ld} =: (A \otimes B)Z \qquad (4.29)$$

such that $CZ := (A \otimes B)Z \sim N(0, \Sigma_3)$ with $CC^T = \Sigma_3$ and $Z \sim \mathcal{N}(0, I)$, where I is the $(\widetilde{D} \times \widetilde{D})$ identity matrix, $\widetilde{D} = LD$. Indeed, for $i = 1, ..., \widetilde{D}$ the eigenvalues and corresponding eigenvectors of Σ_3 are given by $\eta_i = \mu_l \lambda_d$ and $c_i = v_l w_d$ $l = 1, ..., L, d = 1, ..., D$, respectively, see [86]. Notify that the evaluation of (4.29) requires $\mathcal{O}(D^2L^2)$ operations. Under the assumption that $\eta_1 \geq ... \geq \eta_{\widetilde{D}}$, the recipe for a successful implementation is to simulate paths via (4.29) such that Z_k coincides with the k-th dimension of the sequence with low discrepancy. In so doing, most of the variability of the paths is covered by quasi-random numbers of high quality. In many practical applications, working with only the first k principal components leads to a significant dimensionality reduction provided that k is sufficiently smaller than the original dimension. Figure 4.11 shows the accumulated ordered eigenvalues of the (45×45) covariance matrix resulting from the parameters of Table 3.3 c). As we can see, the first eigenvalues explain much of the variability, but a value of about 99% is even reached at $k = 28$; faster drop offs of the eigenvalues are observed for scalar Brownian motions, see [58]. We refer to pertinent numerical analysis literature for calculating eigenpairs, see, e.g., [60]. PCC combined with sequences with low discrepancy was introduced in [1] for option pricing; [44] priced plain vanilla American options by the LSM method combined with PCC and quasi-random numbers.

4.2.4 Numerical Investigations

Let us study the effect of using randomized quasi-random numbers for calculating lower and upper bounds. To do so, we proceed as follows: For approximating the early exercise strategy, i.e. for the regression procedure, we use quasi-random numbers with dimensionality reduction techniques. To draw i.i.d. samples for calculating lower and upper bounds as well as valid confidence intervals, we use randomized quasi-random numbers. Considering

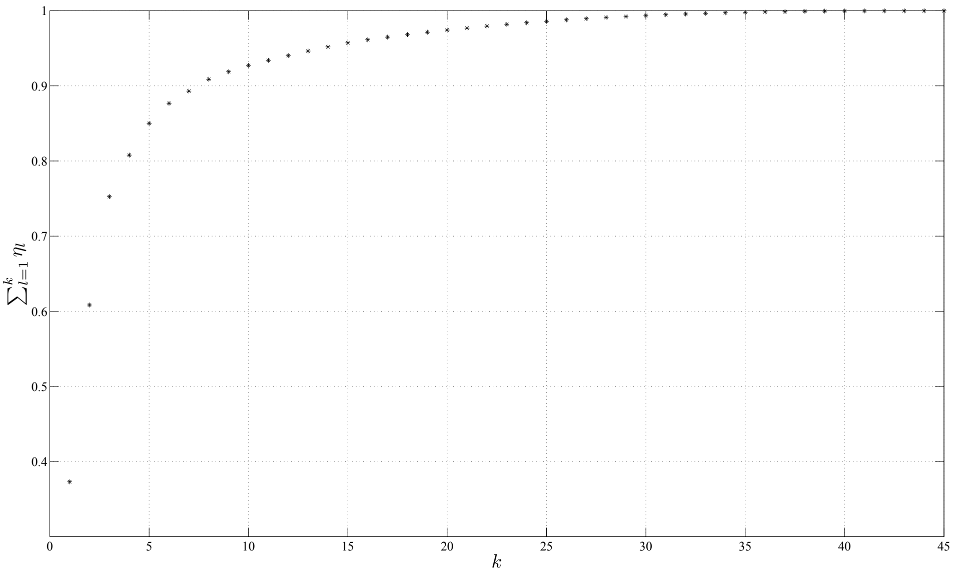

Figure 4.11: Accumulated eigenvalues of a 5-dimensional Brownian motion for nine time steps.

Figure 4.10 in the previous subsection suggests to work with RWR rather than with BBC, whereas we apply the PCC approach without restriction of any kind; the key idea of BBC is to construct large time steps at first, but, as we have no idea about the optimal exercise date of each path, it seems to be senseless to use BBC. As discussed in Subsection 4.2.2, the choice of the values p and q is somewhat tricky and depends on the user's intuitions. Following our discussion, our choice is as follows: As we are more interested in getting a more accurate estimator (2.26), we set $q = 7$ for the inner simulation procedure. Moreover, each subsimulation starts with the first dimension of the randomized sequence with low discrepancy; notify that randomization allows us to simulate in this way, and, hence, even the choice $q = 1$ would be justified. For a comparative study with the LSM method and the evaluation procedure, we should increase the value of q; for instance, in our experiments we set $q = 70$ to get estimated variance reduction factors with a standard error of approximately 25 per cent or more. This choice is sufficient to draw any conclusions from our numerical tests and to estimate quite accurate bounds. In all our experiments we set $q = q_1 = q_2$ in (4.26). Table 4.3 reports lower and upper bounds calculated by the Least Squares Quasi-Monte Carlo (LSQM) method and our Robust Regression Quasi-Monte Carlo (RRQM) method combined with several quasi-random number techniques; the index BBC indicates that we run the algorithms in combination with BBC for the regression step and RWR for calculating bounds; the index PCC indicates that we run the algorithms with PCC for the regression step and for calculating bounds; the index

S and LH denote the use of the Sobol sequence and the Halton sequence leaped, respectively. Moreover, RRQM$^{LH}_{PCRWC}$ denotes our RRQM method combined with PCC for the regression step and for the ordinary simulation procedure, but with RWR for the inner simulations; we combine both techniques due to the higher complexity of PCC.

Table 4.3: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM methods and several quasi-Monte Carlo techniques for Bermudan Max call options.

S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time	Δ Ratio	VR Ratio	SUF$_1$|SUF$_2$
					Ratio		$L_0 \mid U_0$	
90	LSM	27.531 $(3.29 \cdot 10^{-2})$	27.715 $(3.68 \cdot 10^{-2})$	[27.466,27.787]	–	–	–	–
	LSQM$^S_{BBC}$	27.575 $(1.72 \cdot 10^{-2})$	27.690 $(1.81 \cdot 10^{-2})$	[27.541,27.726]	0.92	1.59	3.65|4.14	3.24|8.03
	RRQM$^S_{BBC}$	27.593 $(1.82 \cdot 10^{-2})$	27.672 $(1.87 \cdot 10^{-2})$	[27.557,27.709]	0.92	2.34	3.26|3.88	2.94|13.88
	LSQM$^S_{PCC}$	27.569 $(1.36 \cdot 10^{-2})$	27.691 $(1.50 \cdot 10^{-2})$	[27.542,27.721]	0.85	1.51	5.81|6.06	4.99|6.16
	RRQM$^S_{PCC}$	27.603 $(1.21 \cdot 10^{-2})$	27.683 $(1.30 \cdot 10^{-2})$	[27.579,27.709]	0.88	2.29	7.34|8.03	6.34|10.18
	RRQM$^{LH}_{PCC}$	27.560 $(1.14 \cdot 10^{-2})$	27.631 $(1.24 \cdot 10^{-2})$	[27.537,27.656]	0.89	2.57	8.25|8.82	7.33|10.44
	RRQM$^{LH}_{PCRWC}$	27.560 $(1.14 \cdot 10^{-2})$	27.642 $(1.20 \cdot 10^{-2})$	[27.537,27.666]	1.03	2.23	8.25|9.37	7.33|20.49
100	LSM	37.840 $(3.75 \cdot 10^{-2})$	38.090 $(3.98 \cdot 10^{-2})$	[37.766,38.168]	–	–	–	–
	LSQM$^S_{BBC}$	37.856 $(2.01 \cdot 10^{-2})$	38.023 $(2.15 \cdot 10^{-2})$	[37.816,38.065]	0.98	1.50	3.46|3.43	3.69|3.32
	RRQM$^S_{BBC}$	37.898 $(2.02 \cdot 10^{-2})$	38.015 $(2.13 \cdot 10^{-2})$	[37.858,38.056]	0.98	2.13	3.44|3.50	3.74|4.19
	LSQM$^S_{PCC}$	37.850 $(1.53 \cdot 10^{-2})$	38.034 $(1.69 \cdot 10^{-2})$	[37.820,38.068]	0.92	1.36	6.00|5.54	6.39|3.28
	RRQM$^S_{PCC}$	37.896 $(1.68 \cdot 10^{-2})$	38.015 $(1.78 \cdot 10^{-2})$	[37.863,38.051]	0.93	2.11	4.98|5.03	5.34|5.28
	RRQM$^{LH}_{PCC}$	37.895 $(1.33 \cdot 10^{-2})$	38.010 $(1.45 \cdot 10^{-2})$	[37.868,38.039]	0.91	2.17	7.94|7.54	8.64|5.06
	RRQM$^{LH}_{PCRWC}$	37.895 $(1.33 \cdot 10^{-2})$	38.016 $(1.46 \cdot 10^{-2})$	[37.868,38.045]	1.07	2.08	7.94|7.41	8.64|5.52
110	LSM	49.320 $(4.17 \cdot 10^{-2})$	49.625 $(4.33 \cdot 10^{-2})$	[49.239,49.710]	–	–	–	–
	LSQM$^S_{BBC}$	49.300 $(2.30 \cdot 10^{-2})$	49.533 $(2.49 \cdot 10^{-2})$	[49.254,49.583]	0.94	1.30	3.28| 3.02	3.29|1.49
	RRQM$^S_{BBC}$	49.356 $(2.18 \cdot 10^{-2})$	49.509 $(2.30 \cdot 10^{-2})$	[49.313,49.554]	0.94	2.00	3.66| 3.55	3.67|2.60
	LSQM$^S_{PCC}$	49.300 $(1.81 \cdot 10^{-2})$	49.548 $(2.01 \cdot 10^{-2})$	[49.264,49.588]	0.89	1.23	5.36| 4.62	5.05|1.58
	RRQM$^S_{PCC}$	49.370 $(1.67 \cdot 10^{-2})$	49.524 $(1.84 \cdot 10^{-2})$	[49.336,49.561]	0.91	1.97	6.22| 5.55	5.89|2.23
	RRQM$^{LH}_{PCC}$	49.376 $(1.46 \cdot 10^{-2})$	49.532 $(1.65 \cdot 10^{-2})$	[49.347,49.561]	0.89	1.96	8.24| 6.88	7.79|2.13
	RRQM$^{LH}_{PCRWC}$	49.376 $(1.46 \cdot 10^{-2})$	49.543 $(1.67 \cdot 10^{-2})$	[49.347,49.577]	1.03	1.82	8.24| 6.75	7.79|2.24

Notes. See Table 3.3 b) for option parameters and basis functions. Algorithm specific parameters are as follows: $N_0 = 480,000$, $N_1 = 1,050,000$, $N_2 = 4,900$ and $N_3 = 4,900$; i.e. $N_1 = 70 \cdot 15,000$, $N_2 = 70 \cdot 70$ and $N_3 = 7 \cdot 480$ for the LSQM and RRQM methods.

Let us comment our test results. We see that the bounds produced by our RRQM method are of higher quality; our lower bounds are higher and we get tighter bounds. Obviously, our calculated early exercise strategy is more accurate than the policy resulting from the LSM method and the LSQM method. Nevertheless, we observe a remarkable convergence improvement by using sequences with low discrepancy rather than using pseudo-random numbers for both methods, the LSQM method and our RRQM method. Using PCC for reducing the effective dimensionality of the problem is more efficient than using BBC for the regression step and RWR for the evaluation procedure. The use of PCC leads to estimated variance reduction factors between 4.62 and 8.82; on the contrary, the variance reduction factors for the alternative construction, BBC, are between 3.02 and 4.14. Although PCC is more expensive than RWR and BBC, the speed-up factors coincides with the variance reduction factors for the lower bounds; using our RRQM method combined with PCC leads to speed-up factors of about 6 or more. However, it also pays to work with the LSQM method rather than with the LSM method. We see another behavior for the speed-up factors of the upper bounds; using randomized quasi-random numbers accelerates the convergence by a factor of about 6 or more for the OTM option, of about 3 or more for the ATM option and of about 1.5 or more for the ITM option. Obviously, ITM options are more robust against the quality of random numbers; we think that this behavior can be reduced to the fast exercise decision of these options. Moreover, we observe that using RWR for the inner simulations has no significant effect on the variance of the upper bound, but it is much more cheaper than using PCC for the nested estimation procedure such that we observe a further efficiency increase. Our numerical tests show a remarkable phenomenon: The Halton sequence leaped with $\mathcal{L} = 409$ outperforms the well-working Sobol sequence. Even though the bounds are of about the same size, this simple modification of the Halton sequence leads to variance reduction factors which are significant higher than the factors produced by the Sobol sequence; more precisely, almost all factors are larger than 7.5. By the way, this statement also holds for using the LSQM or our RRQM method with or without a dimensionality reduction technique, for brevity we omit reporting these results. As pointed out in Subsection 4.2.1, the value of the leap is ultimately responsible for the success of this sequence; our chosen leap value $\mathcal{L} = 409$ is recommended by Kocis and Whiten [82], but we have good working experience with other leap values. By the way, in our experience, both sequences, the Sobol sequence and the Halton sequence leaped, are superior to the Halton sequence and, therefore, we have just considered these both sequences. As an additional information, it turned out that it makes sense to work with dimensionality reduction techniques for the regression step, and, thus, we run all our codes in combination with these tools for approximating the continuation value in a first step. In order to underline these results, let us price an option on five correlated assets with more exercise opportunities, i.e. we work with sequences of higher dimension. In the following, we set $q = 30$ for calculating bounds and $q = 5$ for inner simulations. Figure 4.12 shows a typical convergence behavior of the standard

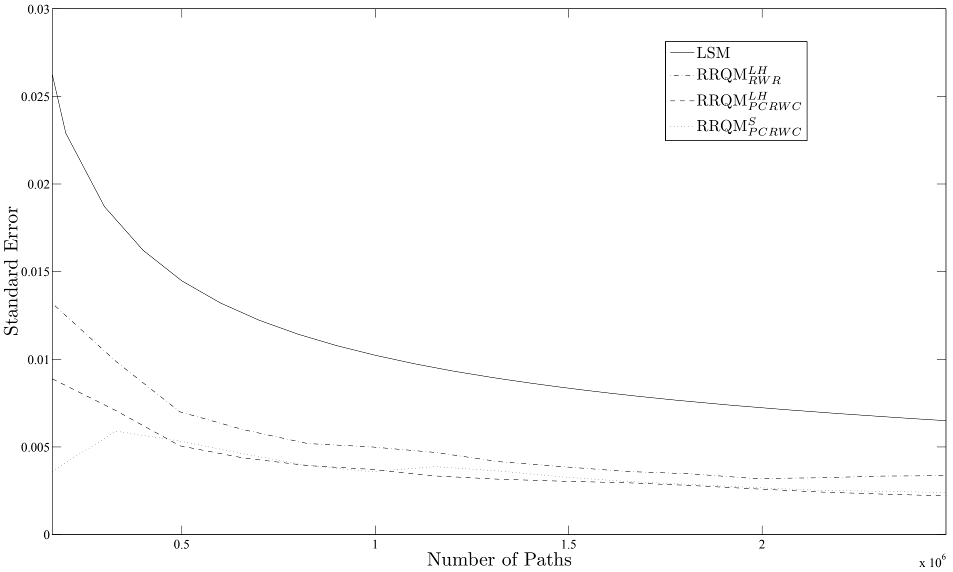

Figure 4.12: Convergence of the lower bound for the LSM and RRQM methods combined with several quasi-Monte Carlo techniques.

Notes. See Table 3.3 c) for option parameters, but with $L = 16$. The early exercise strategy is determined by $N_0 = 150,000$ paths and the first eleven functions of basis (3.53). Standard errors calculated by the RRQM method result from an increasing number of trials q such that $N_1 = q \cdot 33,000$.

error for the lower bound calculated by the LSM method and our RRQM method combined with PCC for the regression step; we run our RRQM method with RWR and the promising hybrid technique PCRWC for the evaluation process. We see that our RRQM method clearly outperforms the LSM method for an increasing number of paths N_1, even though we work with 80-dimensional sequences of low discrepancy. As expected, it is more efficient to work with PCRWC rather than with RWR. Moreover, we see a remarkable convergence behavior of the Halton sequence leaped ($\mathcal{L} = 1951$).

Table 4.4: Lower and upper bounds calculated by the AB approach combined with the LSM and RRM methods and several quasi-Monte Carlo techniques for Bermudan arithmetic average call options.

S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time	Δ Ratio	VR Ratio	$\text{SUF}_1\|\text{SUF}_2$
			Ratio			$L_0\|U_0$	$L_0\|U_0$	
90	LSM	$3.991\ (1.00 \cdot 10^{-2})$	$4.076\ (1.39 \cdot 10^{-2})$	[3.971,4.103]	–	–	–	
	RRQM^{LH}_{RWR}	$4.009\ (5.00 \cdot 10^{-3})$	$4.047\ (6.38 \cdot 10^{-3})$	[4.000,4.059]	0.87	2.30	4.02|4.73	3.89|5.21
	RRQM^{S}_{RWR}	$4.019\ (5.67 \cdot 10^{-3})$	$4.058\ (7.07 \cdot 10^{-3})$	[4.008,4.072]	0.87	2.18	3.20|3.85	3.02|4.65
	LSQM^{LH}_{PCRWC}	$3.974\ (4.42 \cdot 10^{-3})$	$4.050\ (7.91 \cdot 10^{-3})$	[3.980,4.081]	0.87	1.12	5.13|3.08	2.34|1.96
	RRQM^{LH}_{PCRWC}	$4.003\ (3.72 \cdot 10^{-3})$	$4.041\ (5.08 \cdot 10^{-3})$	[4.005,4.061]	0.94	2.24	7.25|7.45	3.54|7.66
	RRQM^{S}_{PCRWC}	$4.017\ (3.59 \cdot 10^{-3})$	$4.063\ (5.43 \cdot 10^{-3})$	[4.010,4.074]	0.93	1.89	7.78|6.53	3.64|5.48
100	LSM	$7.155\ (1.23 \cdot 10^{-2})$	$7.310\ (1.66 \cdot 10^{-2})$	[7.131,7.342]	–	–	–	
	RRQM^{LH}_{RWR}	$7.195\ (6.10 \cdot 10^{-3})$	$7.282\ (9.52 \cdot 10^{-3})$	[7.183,7.301]	0.87	1.76	4.09|3.04	3.28|1.95
	RRQM^{S}_{RWR}	$7.190\ (4.29 \cdot 10^{-3})$	$7.277\ (8.13 \cdot 10^{-3})$	[7.181,7.293]	0.87	1.78	8.27|4.17	6.75|2.26
	LSQM^{LH}_{PCRWC}	$7.147\ (5.56 \cdot 10^{-3})$	$7.294\ (1.12 \cdot 10^{-2})$	[7.137,7.302]	0.93	1.06	4.91|2.18	2.22|1.23
	RRQM^{LH}_{PCRWC}	$7.181\ (4.95 \cdot 10^{-3})$	$7.263\ (7.73 \cdot 10^{-3})$	[7.174,7.285]	0.96	1.91	6.19|4.60	2.76|3.43
	RRQM^{S}_{PCRWC}	$7.185\ (5.17 \cdot 10^{-3})$	$7.272\ (8.00 \cdot 10^{-3})$	[7.175,7.289]	0.94	1.78	5.68|4.30	2.45|3.19
110	LSM	$12.087\ (1.34 \cdot 10^{-2})$	$12.319\ (1.93 \cdot 10^{-2})$	[12.061,12.357]	–	–	–	
	RRQM^{LH}_{RWR}	$12.127\ (7.28 \cdot 10^{-3})$	$12.289\ (1.17 \cdot 10^{-2})$	[12.113,12.312]	0.92	1.45	3.41| 2.71	2.96|2.02
	RRQM^{S}_{RWR}	$12.141\ (6.90 \cdot 10^{-3})$	$12.280\ (1.04 \cdot 10^{-2})$	[12.127,12.300]	0.89	1.67	3.79| 3.43	3.34|2.72
	LSQM^{LH}_{PCRWC}	$12.083\ (5.37 \cdot 10^{-3})$	$12.281\ (1.32 \cdot 10^{-2})$	[12.071,12.304]	0.92	1.17	6.26| 2.14	2.77|1.22
	RRQM^{LH}_{PCRWC}	$12.134\ (5.50 \cdot 10^{-3})$	$12.276\ (9.82 \cdot 10^{-3})$	[12.125,12.286]	1.01	1.65	5.97| 3.86	2.91|2.96
	RRQM^{S}_{PCRWC}	$12.135\ (5.02 \cdot 10^{-3})$	$12.267\ (8.17 \cdot 10^{-3})$	[12.125,12.284]	0.99	1.76	7.16| 5.58	3.11|4.55

Notes. See Figure 4.12 for option parameters. Algorithm specific parameters are as follows: $N_0 = 150,000$, $N_1 = 990,000$, $N_2 = 1,200$ and $N_3 = 5,000$; i.e. the LSQM and RRQM methods are run with $N_1 = 30 \cdot 33,000$, $N_2 = 30 \cdot 40$, $N_3 = 5 \cdot 1,000$; especially for the RRQM method, $\alpha = 0.887$, $\beta = 0.993$; codes are run with the first eleven functions of basis (3.53).

This observation is strengthened by the results of Table 4.4 showing lower and upper bounds calculated by our RRQM method with PCC for the regression step as well as RWR and PCRWC for the evaluation step; the index RWR indicates the use of RWR in the second phase. We prefer the use of PCRWC, as it is more efficient than using PCC alone for the evaluation procedure. Once again, we observe a remarkable reduction in variance by using randomized quasi-random numbers; the estimated variance reduction factors are between 4.91 and 6.26 for the LSQM method combined with PCRWC and between 5.68 and 7.78 for our RRQM method combined with PCRWC for the lower bounds; for the upper bounds the estimated variance reduction factors are of about the same size for the OTM option, but significantly lower for the ATM and ITM options. Using randomized quasi-random numbers combined with RWR for calculating lower and upper bounds seems to make sense as well, but the factors are somewhat lower than using PCRWC. Anyway, the higher complexity of PCC leads to lower speed-up factors for the lower bounds than in the previous example such that using PCRWC is no longer more efficient than RWR for a smaller time step width. Anyway, the estimated speed-up factors of using PCRWC for the upper bounds are slightly superior to the factors achieved by using RWR. This is the result of using RWR for the inner simulations such that the total CPU time is not greatly affected by PCC; generally, the CPU time for the upper bound is dominated by the inner simulation procedures.

Let us draw some conclusions from our experiments. In general, it pays to work with randomized quasi-random numbers for estimating lower and upper bounds. We should implement the regression step in combination with PCC rather than with BBC. Provided that the number of time steps is sufficiently small, our hybrid approach PCRWC is more efficient than using RWR or PCC alone. Once again, we have seen a remarkable convergence behavior of our RRM method. Therefore, we highly recommend to implement our RRQM method combined with PCC for approximating an early exercise policy and with PCRWC for the evaluation process. To deal with a larger number of time steps, for European-style options Giles [56] proposed to implement PCC in combination with the sine transform; more precisely, by rewriting $(A \otimes B)Z$ in (4.29) as $A\widetilde{Z}B^T$ with the $(L \times D)$-matrix $\widetilde{Z}$ filled with the elements of Z, we might evaluate $A\widetilde{Z}$ in $\mathcal{O}(DL\log_2(L))$ operations and $\mathcal{O}(LD^2)$ further operations are required for the multiplication with B^T. We might implement this approach to get a further acceleration of our proposed method. Anyway, in this thesis we tackle the problem in another way: Combining our change of drift technique proposed in Section 4.1 with RWR leads to a natural dimensionality reduction, as we drive paths into early exercise regions. In so doing, most of the paths are generated by the first dimensions of the randomized point set, and, thus, we expect to price options with high accuracy. Let us conclude this section with a final remark on the sequences with low discrepancy itself. It turned out that using the random digital shift in

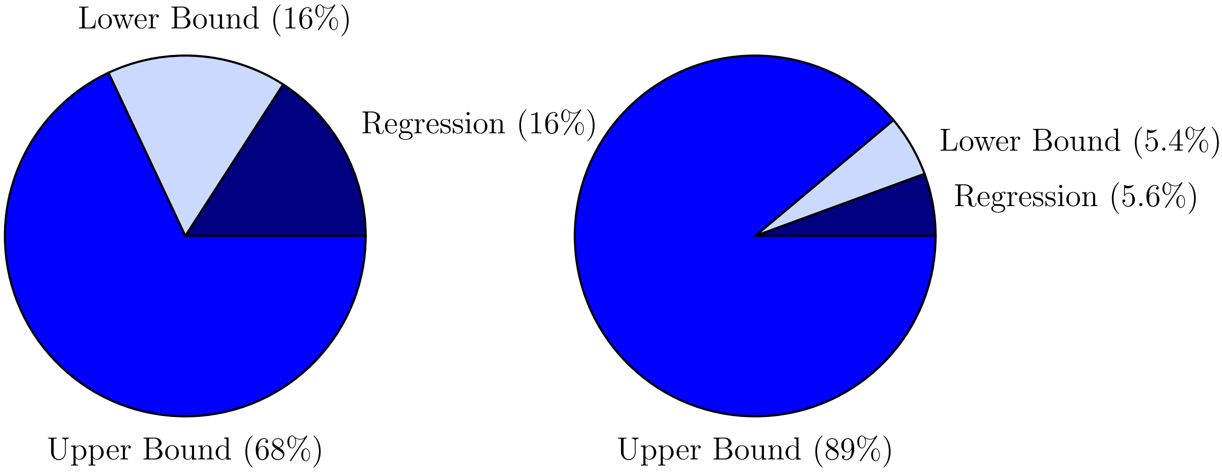

Figure 4.13: CPU time partitions of the AB approach for an arithmetic average call option on five assets. Left and right pie charts show partitions with $N_3 = 1,200$ and $N_3 = 5,000$, respectively.

base 2 rather than RSM1 leads to the same results, and, therefore, we have implemented our codes with RSM1. We are not familiar with any studies regarding option pricing by the Halton sequence leaped. However, our experience is that this slide modification of the Halton sequence does a great job. The construction of the Sobol sequence is more complex, which might be seen as an advantage for the Halton sequence leaped. Our chosen leap values perform well, and, thus, due to its simplicity we prefer the Halton sequence leaped rather than the Sobol sequence.

4.3 Further Acceleration Techniques

The purpose of this section is to explore the effect of implementing some simple techniques for further acceleration. To start with, Figure 4.13 gives us insight into typical CPU time partitions of the AB approach combined with the LSM method for calculating an ATM arithmetic average call option on five assets; see Table 4.3 for option and algorithm parameters. We clearly see that the CPU time for calculating upper bounds dominates the total CPU time; the more effort we spend in the inner simulation procedure, the more this process dominates the total CPU time. Notice that the worst-case computational complexity for calculating $\widehat{\Delta}_0$, compare (2.25), is $\mathcal{O}(N_2 N_3 L^2 D)$. The user might decrease the number of inner simulations, but, in general, it is often not clear which size gives sufficiently accurate approximations of the quantities (2.26). It is obvious that there is a tradeoff between speed and accuracy. To deal with this deficiency, we suggest to work with online algorithms such as the RM method (4.10); in Section 4.1 we have seen that these algorithms take as many paths as they need for convergence. To be more precisely, let us consider the estimates for the continuation values (2.26), which we denote by $\overline{c}_{N_3}$

with

$$\overline{c}_{N_3} = \frac{1}{N_3}\sum_{m=1}^{N_3} c_m,\ c_m = e^{-r\tau_l^m \Delta t} Z^m_{\tau_l^m}$$

for notational convenience. According to [121], a stepwise calculation of $\overline{c}_{N_3}$ is recursively defined by

$$\begin{aligned}\overline{c}_{N_3+1} &= \frac{1}{N_3+1}\sum_{m=1}^{N_3+1} c_m \\ &= \frac{N_3}{N_3+1}\overline{c}_{N_3} + \frac{1}{N_3+1}c_{N_3+1} \\ &= \overline{c}_{N_3} - \frac{1}{N_3+1}(\overline{c}_{N_3} - c_{N_3+1}),\ N_3 = 0, 1, 2, ..., \end{aligned} \tag{4.30}$$

with $\overline{c}_0 = 0$; notice that this recursion can be interpreted as a search step of the RM algorithm (4.10) by setting

$$\gamma_{N_3} = \frac{1}{N_3+1},\ Y_{N_3} = \overline{c}_{N_3} - c_{N_3+1}.$$

Following this reformulation, we are now in the situation to estimate the continuation values (2.26) by an online procedure in the sense that i.i.d. samples c_{N_3+1} are drawn until convergence. In our view, rather than using (4.18) a more robust convergence criteria is to compare only the J-th approximations of $\overline{c}_{N_3}$ with each other, i.e. for $N_3 = J, 2J, ...$, we test for convergence on Δc via

$$\frac{|\overline{c}_{N_3} - \overline{c}_{N_3-J}|}{\max\{|\overline{c}_{N_3}|, 1\}} < TOL_{\Delta c},\ 0 < TOL_{\Delta c} \ll 1.$$

Moreover, we cancel the search procedure in case of non-convergence after a reasonable number of iterations $\overline{N}_3$. Figure 4.14 illustrates the calculated $\widehat{\Delta}_0$ for an increasing number of paths. The solid line shows the convergence for the static approach, i.e. for a fixed number of inner simulations N_3. The dash-dotted and dashed lines reflect the convergence of our proposed RM approach with tolerance values $TOL_{\Delta c} = 10^{-6}$ and $TOL_{\Delta c} = 10^{-8}$, respectively; the maximum number of paths $\overline{N}_3$ coincides with the increasing number of inner simulations N_3 for the static approach; see Table 4.3 for algorithm and option parameters with $S_0 = 90$, $N_1 = 1,000,000$, $N_2 = 1,000$, $J = 20$. We see that we achieve the same level of accuracy as the original static approach by running our approach with $TOL_{\Delta c} = 10^{-8}$; demanding for a lower value such as $TOL_{\Delta c} = 10^{-6}$ leads to almost the same convergence. In our experience, a value in the range of both values is sufficient for getting stable results. Thus, we are able to guarantee the same level of accuracy with lower effort; see Table 4.3 for speed-up factors. Before we consider last test results, let us discuss two further techniques that might lead to efficiency increases. It is well known that evaluating the exponential function $\exp(x)$ for any $x \in \mathbb{R}$ is a time-consuming

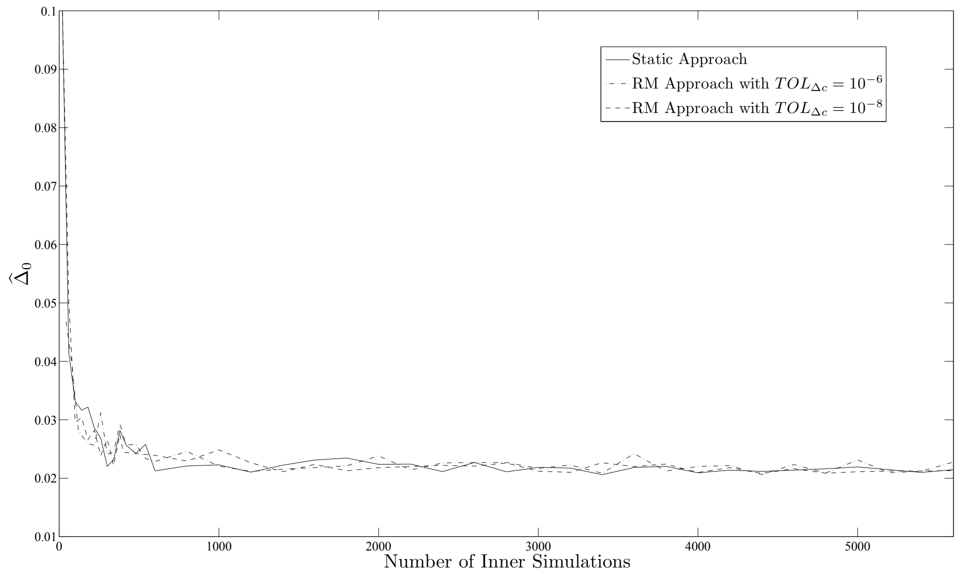

Figure 4.14: Convergence of estimator $\widehat{\Delta}_0$ for a static and an adaptive approach.

process. Thus, we should test the effect of replacing the pre-implemented double precision routine for realizing this operation by evaluation algorithms with lower accuracy; for instance, according to Hart [63] and Jonen [77], we might approximate $\exp(x)$ by rational polynomials for which the coefficients determine the accuracy that we fix a priori. In this thesis we focus our attention on the approach by Jonen and test the influence of replacing $\exp(x)$ by the rational polynomial

$$\exp(x) \approx 2^a \left(\frac{2b(a_0 + b^2 a_1)}{(b_0 + b^2 b_1) - x(a_0 + b^2 a_1)} + 1 \right) \tag{4.31}$$

with $x = a \ln 2 + b \geq 0,\ b \in [0, \ln(2)),\ a \in \mathbb{N}$; the coefficients resulting from the rational Chebyshev approximation are $a_0 = 4.9999994 \cdot 10^{-1},\ a_1 = 8.2832497 \cdot 10^{-3},\ b_0 = 1.0,\ b_1 = 9.9897819 \cdot 10^{-2}$. Notify that we have $\exp(x) = 1/\exp(|x|)$ in case of $x < 0$. Our guess is that the relative accuracy of 10^{-7} of this approximation is quite sufficient for getting stable results. Let us address a further vital point in Monte Carlo applications. Antithetic variables (AVs) might reduce variance and we are used to work with this technique. Anyway, in our experience using AVs does not often have the desired effect we expect from a variance reduction technique for calculating lower and upper bounds by the AB approach; this is caused by the construction of the AB approach. Nevertheless, generating a normally distributed random number Z and taking -Z as a further random number reduces the CPU time of the random number generation by a factor of 2. Therefore, from this point of view, we should work with AVs, even if the effect of this technique is often not clear. Table 4.3 reports lower and upper bounds by using the LSM method with and without AVs denoted by LSM and LSM$_{AV}$, respectively, for arithmetic average call options on five assets. At first glance, it seems to be senseless to reduce variance via AVs; the variance reduction factors are between 0.85 and 1.00 for the estimated lower and upper bounds. However, the further gain is in the lower CPU time such that around 10 per cent of the CPU time might be saved by using AVs; we get speed-up factors of 1.15 for the lower bound of the OTM option, but around 1.0 for the ATM and ITM options. Anyway, the speed-up factors for the upper bounds are between 0.59 and 0.91, as $\widehat{\sigma}_\Delta$ is higher by using AVs. At this point we should remark that in our experience using AVs does not lead to a slow-down for calculating upper bounds in any case, i.e. we often consider a slight speed-up for upper bounds as well, but there is no clear trend. We also run this experiment by the LSM method in combination with replacing the routine $\exp(x)$ in Java by (4.31), denoted by LSM$_{Exp}$, and our modified inner simulation procedure (4.30), denoted by LSM$_\Delta$. Working with the approximation (4.31) is worth mentioning, as we are able to accelerate the calculation of lower and upper bounds by 11 up to 17 per cent without any loss in accuracy. This impressive speed-up factors are caused by the fact that the operations of evaluating $\exp(x)$ have a great influence on the complexity of the path simulations. Rather than running the algorithms with the static approach for the inner simulation process, we should use our proposed dynamic approach (4.30); our modification leads to speed-up factors of about 2 or more without remarkable changes in the results. Last but not least, we combine all proposed approaches of this thesis to calculate lower and upper bounds; i.e. we price this type of option by our RRM method combined with the Halton sequence leaped ($\mathcal{L} = 409$) and PCC for approximating an early exercise strategy; for calculating lower and upper bounds we use our proposed change of drift technique ISQNew combined with RWR for reducing the effective dimensionality; additionally, we replace $\exp(x)$ by (4.31) and work with (4.30) for the inner simulation procedure; we denote our approach by RRQM$_{AllIn}$. First of all, we are able to get much higher lower bounds and the differences between the lower and upper bounds are tighter by factors between 2.76 and 3.50. It is obvious that our approximated early-exercise strategy is of high quality. We observe a remarkable reduction in variance for the estimated bounds; the factors for the lower bounds are between 14.98 and 20.55; the factors for the upper bounds are between 4.70 and 16.43. As for all three initial values, the CPU time is much lower than the required time by the LSM method, we are able to speed-up the evaluation procedure by factors between 15.69 and 25.45 for the lower bound and between 8.19 and 62.12 for the upper bound. These impressive speed-up factors originates from combining the most promising techniques proposed in this thesis. Further numerical tests have shown that these results are representative for more exercise opportunities as well.

Table 4.5: Lower and upper bounds calculated by several speed-up techniques for arithmetic average call options.

S_0	Method	Lower Bound	Upper Bound	95% CI	CPU Time	Δ Ratio	VR Ratio	SUF$_1$|SUF$_2$
					Ratio		L_0|U_0	
90	LSM	1.537 ($4.87 \cdot 10^{-3}$)	1.550 ($5.42 \cdot 10^{-3}$)	[1.528,1.560]	–	–	–	–
	LSM$_{AV}$	1.540 ($4.88 \cdot 10^{-3}$)	1.558 ($5.84 \cdot 10^{-3}$)	[1.530,1.570]	1.07	0.67	0.99 |0.86	1.15 |0.59
	LSM$_{Exp}$	1.537 ($4.87 \cdot 10^{-3}$)	1.550 ($5.42 \cdot 10^{-3}$)	[1.528,1.560]	1.16	–	–	–
	LSM$_\Delta$	1.537 ($4.87 \cdot 10^{-3}$)	1.549 ($5.35 \cdot 10^{-3}$)	[1.528,1.559]	2.50	–	–	–
	RRQM$^{LH}_{AllIn}$	1.545 ($1.15 \cdot 10^{-3}$)	1.550 ($1.34 \cdot 10^{-3}$)	[1.542,1.552]	3.50	2.38	17.95|16.43	16.05|62.12
100	LSM	3.963 ($6.80 \cdot 10^{-3}$)	4.006 ($8.19 \cdot 10^{-3}$)	[3.950,4.022]	–	–	–	–
	LSM$_{AV}$	3.961 ($6.82 \cdot 10^{-3}$)	4.010 ($8.87 \cdot 10^{-3}$)	[3.948,4.028]	1.13	0.88	0.99| 0.85	1.02 |0.76
	LSM$_{Exp}$	3.963 ($6.80 \cdot 10^{-3}$)	4.006 ($8.19 \cdot 10^{-3}$)	[3.950,4.022]	1.17	–	–	–
	LSM$_\Delta$	3.963 ($6.80 \cdot 10^{-3}$)	4.009 ($8.41 \cdot 10^{-3}$)	[3.950,4.026]	2.00	–	–	–
	RRQM$^{LH}_{AllIn}$	3.992 ($1.51 \cdot 10^{-3}$)	4.005 ($2.24 \cdot 10^{-3}$)	[3.989,4.009]	3.38	3.23	20.55|13.44	25.45|40.00
110	LSM	9.311 ($7.84 \cdot 10^{-3}$)	9.364 ($9.25 \cdot 10^{-3}$)	[9.296,9.382]	–	–	–	–
	LSM$_{AV}$	9.301 ($7.86 \cdot 10^{-3}$)	9.353 ($9.51 \cdot 10^{-3}$)	[9.285,9.372]	1.08	1.01	1.00 |0.95	1.01 |0.91
	LSM$_{Exp}$	9.311 ($7.84 \cdot 10^{-3}$)	9.364 ($9.25 \cdot 10^{-3}$)	[9.296,9.382]	1.11	–	–	–
	LSM$_\Delta$	9.311 ($7.84 \cdot 10^{-3}$)	9.364 ($9.27 \cdot 10^{-3}$)	[9.296,9.382]	1.89	–	–	–
	RRQM$^{LH}_{AllIn}$	9.342 ($1.89 \cdot 10^{-3}$)	9.362 ($4.27 \cdot 10^{-3}$)	[9.338,9.371]	2.76	2.67	14.98|4.70	15.69|8.19

Notes. See Table 3.3 b) for option parameters and basis functions. Algorithm settings are as follows: $N_0 = 300,000$, $N_1 = 990,000$, $N_2 = 1,200$, $N_3 = 5,000$; especially for RRQM$^{LH}_{AllIn}$, $N_4 = 3,000$, $q = 30$, i.e. $N_1 = 30 \cdot 33,000$, $N_2 = 30 \cdot 40$, $J = 10, \alpha = 0.887$, $\beta = 0.993$; especially for LSM$_\Delta$, $J = 20$; especially for LSM$_\Delta$ and RRQM$^{LH}_{AllIn}$, $\overline{N}_3 = 5,000$, $TOL_{\Delta c} = 10^{-6}$.

Chapter 5

Conclusions

In recent years high-dimensional American-style derivatives have gained in importance for the financial market. From a practical point of view, regression-based Monte Carlo methods in combination with dual methods have established themselves for pricing these complex financial instruments. In this thesis we have proposed efficient pricing algorithms and have compared them with state-of-the-art approaches. We have tackled the problem of developing efficient algorithms in two ways: In the first main part of this thesis we have extended the class of regression-based Monte Carlo methods by introducing our RRM method. In order to get an impression about the strength of our proposed approach, we have priced options on up to thirty assets with an early-exercise feature. In the second main part we have focused our attention on variance reduction techniques as well as some simple but powerful tools for increasing efficiency. We have investigated the performance of our proposed techniques by valuing several options with an early exercise feature.
Let us sum up the key results of our proposed methods. Based on the fact that outliers destroy the quality of an approximation of the continuation value, we have suggested to take bad data points into account during the regression procedure. We have seen that we get a remarkable bias reduction without increasing variance – that is important from a practical point of view – by using robust regression rather than ordinary least squares. Compared with the state-of-the-art LSM estimator, we get a nearly unbiased estimator, and in combination with the AB approach we often obtain tighter bounds by a factor of more than two. In our speed-accuracy test we have seen that we get speed-up factors of up to over four; as a consequence, we are able to improve the approximated early exercise strategy significantly by using our approach. A further gain is that we have lower memory requirements due to faster convergence, which seems to be very desirable. In our experience and considering applications in the literature, especially for high-dimensional problems, it is often not clear to choose an adequate basis for regression-based Monte

Carlo methods, and numerical tests must be done to specify well-working basis functions; efficient basis functions do not only depend on the underlying payoff, but also on the underlying model and input parameters of the model. This procedure might be time-consuming, and, that is why, we have a huge advantage by pricing options with our RRM method rather than with the LSM method. We have shown that our method is much less sensitive to the choice of basis functions. In practice, our RRM method might be implemented as an interleaving estimator for pricing complex financial derivatives with an early exercise feature. Anyway, based on an approximation of the continuation value by our RRM method, we might combine the resulting early exercise strategy with dual methods; this application might be seen more favorable, as bounds for the true option value are more meaningful, especially for high-dimensional products. We have tackled the problem of slow convergence for producing lower and upper bounds in three ways: To start with, we have introduced our change of drift technique for driving paths in regions which are more important for variance. It turned out that it is much more efficient to solve the underlying stochastic optimization problem for finding the drift minimizing variance by the deterministic counterpart; due to huge speed-up factors of up to 7,378 and more robustness, we highly recommend this way of implementation. Remarkable speed-up factors of up to over 20 for the lower bounds are the proof of the success of our proposed approach; provided that we use the efficient way of finding the optimal drift, we have observed that it makes sense to reduce variance by a change of drift. Especially for OTM and ATM options, this approach is a worthwhile acceleration technique. Secondly, we have studied the effect of using quasi-random numbers for the AB approach. We have considered several quasi-Monte Carlo techniques and applied them in numerical experiments. It turned out that it pays to work with quasi-random numbers combined with the dimensionality reduction technique PCC for the regression and evaluation steps; we achieved speed-up factors of up to over 20. However, due to its high complexity using PCC for a smaller time step width might be seen critical. In our numerical investigations we have tested the Halton sequence leaped with great success. This slight modification of the Halton sequence is competitive to the well-working Sobol sequence. To the best of our knowledge, our study of using quasi-Monte Carlo techniques with respect to the AB approach is the first one. Moreover, we are not familiar with any study in computational finance investigating the performance of the Halton sequence leaped. Last but not least, we have proposed some simple but yet powerful acceleration techniques. Our proposed RM approach for the inner simulation procedure shows a remarkable speed-up for calculating upper bounds, and replacing the double-precision evaluation procedure of the exponential function by a cheaper approximation leads to a further convergence improvement. Moreover, we have critically discussed the use of antithetic variables as a variance reduction technique. Combining all our proposed approaches in this thesis leads to our ultimate pricing algorithm. We highly recommend to work with our RRM method combined with quasi-random numbers and PCC for approximating an early exercise strategy. Note that PCC does not greatly affect the CPU time of the regression step as the complexity is determined by the regression solver itself. Our change of drift technique combined with randomized deterministic point sets is a natural powerful dimensionality reduction technique, also for smaller time step widths, and our further acceleration techniques suggested in Section 4.3 perform well such that our RRQM$_{AllIn}$ solver might be help practitioners to price high-dimensional American-style financial products much more efficient.

We would like to conclude this thesis with a brief overview of further possible areas of research. Our chosen loss functions for specifying the robust regression problem show very good performance, but other loss functions might be tested as well. We have focused our attention on an outlier detection procedure defined by the empirical distribution of the error, and it has turned out that our method is very robust against the selection of empirical quantiles. However, it seems to be convenient to research for other outlier detection procedures, e.g. working with natural bounds for options. As our convergence proof allows for considering nonlinear approximation architectures such as neural networks, wavelet thresholding, etc., a logical next step might be to apply the idea of robust regression to these interesting approaches. Anyway, first numerical tests have shown that nonlinear large-residual approximation architectures should be investigated carefully; see also [15]. In our experience, they might significantly slow down the pricing procedure such that linear regression is much more efficient. Needless to mention, initial values play a key role for an efficient implementation, and, thus, besides recent developments of solvers for nonlinear regression, see [133], we should spend more effort on that issue. As mentioned in Chapter 2, the approach by Haugh and Kogan [64] works with modified samples for approximating the continuation value such that we might consider distribution-dependent outlier detection procedures if we apply the idea of robust regression to their method. Moreover, our RM approach for the inner simulation procedure might be used for accelerating the determination of samples. A number of methods work with the minimization of the squared loss function to approximate the continuation value; some of them are cited in the introduction. Roughly speaking, all these methods might be tested under our robust regression framework. Recently, [26] has proposed an approach for estimating greeks by the LSM method; we are sure that this approach combined with our RRM method leads to more accurate results. As much effort has been spent on increasing the efficiency by variance reduction techniques, see e.g. [47], [23] and [79], our method might be implemented in combination with these approaches to accelerate convergence. Needless to mention, we are concerned in developing further variance reduction techniques for our RRQM$_{AllIn}$ method. An active research field are quasi-Monte Carlo techniques, constructing sequences of higher quality and new dimensionality reduction techniques might help to improve convergence. Recently, Tian et al. [123] have compared our vectorization approach of the LSM algorithm [76] with their graphics processing unit (GPU)-based approach and have drawn the conclusion that it is more efficient to program on GPUs. However, it is obvious that this is just a technical outsourcing of the problem and should not keep us from developing faster and more accurate algorithms. To conclude, our suggested ideas might open up new fields of study regarding the efficient pricing of high-dimensional financial derivatives with an early exercise feature.

Bibliography

[1] P. Acworth, M. Broadie and P. Glasserman, *A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing*, in Monte Carlo and Quasi-Monte Carlo Methods, H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, eds., Springer, New York, USA, 1998, pp. 1–18.

[2] F. Åkesson and J. Lehoczky, *Discrete eigenfunction expansion of multi-dimensional Brownian motion and the Ornstein-Uhlenbeck process*, Working Paper, Department of Statistics, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA, 1993.

[3] L. Andersen, *A simple approach to the pricing of Bermudan swaptions in the multi-factor Libor market model*, Journal of Computational Finance, 3 (2000), pp. 5–32.

[4] L. Andersen and M. Broadie, *A primal-dual simulation algorithm for pricing multi-dimensional American options*, Management Science, 50 (2004), pp. 1222–1234.

[5] J. Antoch and H. Ekblom, *Recursive robust regression computational aspects and comparison*, Computational Statistics and Data Analysis, 19 (1995), pp. 115–128.

[6] I. Antonov and V. Saleev, *An economic method of computing l p_t sequences*, USSR Computational Mathematics and Mathematical Physics, 19 (1979), pp. 252–256.

[7] B. Arouna, *Robbins-Monro algorithms and variance reduction in finance*, Journal of Computational Finance, 7 (2003), pp. 35–61.

[8] V. Bally and G. Pagès, *A quantization algorithm for solving multi-dimensional discrete-time optimal stopping problems*, Bernoulli, 9 (2003), pp. 1003–1049.

[9] J. Barraquand and D. Martineau, *Numerical valuation of high-dimensional multivariate American securities*, Journal of Financial and Quantitative Analysis, 30 (1995), pp. 383–405.

[10] J. Beasley and S. Springer, *The percentage points of the normal distribution*, Applied Statistics, 26 (1977), pp. 118–121.

[11] A. Bensoussan, *On the theory of option pricing*, Acta Applicandae Mathematicae, 2 (1984), pp. 139–158.

[12] S. Berridge and J. Schumacher, *Pricing high-dimensional American options using local consistency conditions*, in Numerical Methods for Finance, J. Appleby, D. Edelman and J. Miller, eds., Chapman & Hall/CRC, Boca Raton, Florida, USA, 2008, pp. 13–52.

[13] J. Birch, *Some convergence properties of iterated reweighted least squares in the location model*, Communications in Statistics - Simulation and Computation, 9 (1980), pp. 359–369.

[14] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, USA, 2006.

[15] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, Pennsylvania, USA, 1996.

[16] P. Bossaerts, *Simulation estimators of optimal early exercise*, Working Paper, Carnegie Mellon University, Pittsburg, Pennsylvania, USA, 1989.

[17] N. Bouleau, D. Lamberton and B. Lapeyre, *A numerical method suited to the probabilistic design of structure*, Journal of Theoretical and Applied Mechanics, 5 (1986), pp. 781–801.

[18] P. Boyle, *Options: A Monte Carlo approach*, Journal of Financial Economics, 4 (1977), pp. 323–338.

[19] ——, *A lattice framework for option pricing with two state variables*, Journal of Financial and Quantitative Analysis, 23 (1988), pp. 1–12.

[20] P. Boyle, J. Evnine and S. Gibbs, *Numerical evaluation of multivariate contingent claims*, Review of Financial Studies, 2 (1989), pp. 241–250.

[21] Braaten and Weller, *An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration*, Journal of Computational Physics, 33 (1979), pp. 249–258.

[22] P. Bratley and B. Fox, *Algorithm 659: implementing sobol's quasirandom sequence generator*, ACM Transactions on Modeling and Computer Simulation, 14 (1988), pp. 88–100.

[23] M. Broadie and M. Cao, *Improved lower and upper bound algorithms for pricing American options by simulation*, Quantitative Finance, 8 (2008), pp. 845–861.

[24] M. Broadie and P. Glasserman, *Pricing American-style securities using simulation*, Journal of Economic Dynamics and Control, 21 (1997), pp. 1323–1352.

[25] ——, *A stochastic mesh method for pricing high-dimensional American options*, Journal Of Computational Finance, 7 (2004), pp. 35–72.

[26] R. Caflisch and Y. Wang, *Pricing and hedging American-style options: a simple simulation-based approach*, Journal of Computational Finance, 13 (2010), pp. 95–125.

[27] R. Caflisch, W. Morokoff and A. Owen, *Valuation of mortgage backed securities using Browninan bridges to reduce effective dimension*, Journal of Computational Finance, 1 (1997), pp. 27–46.

[28] J. Carrière, *Valuation of the early-exercise price for options using simulations and nonparametric regression*, Insurance: Mathematics and Economics, 19 (1996), pp. 19–30.

[29] ——, *Linear regression and standardized quasi-Monte Carlo for approximating Bermudan options*, Working Paper, University of Alberta, Alberta, Canada, 2001.

[30] X.-W. Chang and Y. Guo, *Huber's M-estimation in relative GPS positioning: computational aspects*, Journal of Geodesy, 79 (2005), pp. 351–362.

[31] S. Chaudhary, *American options and the LSM algorithm: quasi-random sequences and Brownian bridges*, Journal of Computational Finance, 8 (2005), pp. 101–115.

[32] H. Chen, A. Gao and L. Guo, *Convergence and robustness of the Robbins-Monro algorithm truncated at randomly varying bounds*, Stochastic Processes and their Applications, 27 (1988), pp. 217–231.

[33] D. Clark and M. Osborne, *Finite algorithms for Huber's M-estimator*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 72–85.

[34] F. Clarke, *Optimization and Nonsmooth Analysis*, J. Wiley & Sons, New York, USA, 1983.

[35] J. Cox, S. Ross and M. Rubinstein, *Option pricing: a simplified approach*, Journal of Financial Economics, 7 (1979), pp. 229–263.

[36] K. Crammer, M. Kearns and J. Wortman, *Learning from multiple sources*, Journal of Machine Learning Research, 9 (2008), pp. 1757–1774.

[37] R. Cranley and T. Patterson, *Randomization of number theoretic methods for multiple integration*, SIAM Journal on Numerical Analysis, 13 (1976), pp. 904–914.

[38] F. Cucker and S. Smale, *On the mathematical foundations of learning*, Bulletin of the American Mathematical Society, 39 (2001), pp. 1–49.

[39] M. Davis and I. Karatzas, *A deterministic approach to optimal stopping*, in Probability, Statistics and Optimization: A Tribute to Peter Whittle, F. Kelly, ed., J. Wiley & Sons, New York, USA, and Chichester, England, 1994, pp. 455–466.

[40] T. H. de mello and A. Shapiro, *On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs*, SIAM Journal of Optimization, 11 (2000), pp. 70–86.

[41] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983.

[42] J. Detemple, *American-Style Derivatives: Valuation and Computation*, Chapman & Hall/CRC, Boca Raton, Florida, USA, 2006.

[43] P. Devroye, L. Györfi and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, New York, USA, 1996.

[44] M. Dion and P. L'Ecuyer, *American option pricing with randomized quasi-Monte Carlo simulations*, in Proceedings of the Winter Simulation Conference, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan and E. Yücesan, eds., IEEE Press, New York, USA, 2010, pp. 2705–2720.

[45] M. Duflo, *Random Iterative Models*, Springer, Berlin, Germany, 1997.

[46] D. Egloff, *Monte Carlo algorithms for optimal stopping and statistical learning*, Annals of Applied Probability, 15 (2005), pp. 1396–1432.

[47] S. Ehrlichman and S. Henderson, *Adaptive control variates for pricing multi-dimensional American options*, Journal of Computational Finance, 11 (2007), pp. 65–91.

[48] H. Ekblom, *A new algorithm for the Huber estimator in linear models*, BIT Numerical Mathematics, 28 (1988), pp. 123–132.

[49] G. Fasshauer, A. Khaliq and D. Voss, *Using meshfree approximation for multi-asset American options*, Journal of Chinese Institute of Engineers, 27 (2004), pp. 563–571.

[50] R. Feinerman and D. Newman, *Polynomial Approximation*, MD: Williams and Wilkins, Baltimore, Maryland, USA, 1973.

[51] R. Fletcher, *Practical Methods of Optimization*, J. Wiley & Sons, Hoboken, New Jersey, USA, 2nd ed., 2008.

[52] P. Forsyth and K. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2095–2122.

[53] J. Friedman, *Greedy function approximation: a gradient boosting machine*, The Annals of Statistics, 29 (2001), pp. 1189–1232.

[54] M. Fu and J.-Q. Hu, *Sensitivity analysis for Monte Carlo simulation of option pricing*, Probability in the Engineering and Informational Sciences, 9 (1995), pp. 417–446.

[55] D. Garcia, *Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule*, Journal of Economic Dynamics and Control, 27 (2003), pp. 1855–1879.

[56] M. Giles, F. Kuo, I. Sloan and B. Waterhouse, *Quasi-Monte Carlo for finance applications*, ANZIAM Journal, 50 (2008), pp. 308–323.

[57] I. Girsanov, *On transforming a certain class of stochastic processes by absolutely continuous substitution of measures*, Theory of Probability and its Applications, 5 (1960), pp. 285–301.

[58] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, New York, USA, 2004.

[59] P. Glasserman and B. Yu, *Number of paths versus number of basis functions in American option pricing*, Annals of Applied Probability, 14 (2004), pp. 2090–2119.

[60] G. Golub and C. van Loan, *Matrix Computations*, vol. 3rd, Johns Hopkins University Press, Baltimore, Maryland, USA, 1996.

[61] J. Halton, *On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals*, Numerische Mathematik, 2 (1960), pp. 84–90.

[62] J. Hammersley, *Monte Carlo methods for solving multivariable problems*, Annals of the New York Academy of Sciences, 86 (1960), pp. 844–874.

[63] J. Hart, *Computer Approximations*, J. Wiley & Sons, New York, USA, 1968.

[64] M. Haugh and L. Kogan, *Approximating pricing and exercising of high-dimensional American options: a duality approach*, Operations Research, 52 (2004), pp. 258–270.

[65] D. Haussler, *Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension*, Journal of Combinatorial Theory A, 69 (1995), pp. 217–232.

[66] H. He, *Convergence from discrete- to continous-time contingent claim prices*, Review of Financial Studies, 3 (1990), pp. 523–546.

[67] P. Hellekalek, *Regularities in the distribution of special sequences*, Journal of Number Theory, 18 (1984), pp. 41–55.

[68] M. Hinich and P. Talwar, *A simple method for robust regression*, Journal of the American Statistical Association, 70 (1975), pp. 113–119.

[69] E. Hlawka, *Funktionen von beschränkter Variation in der Theorie der Gleichverteilung*, Annali di Matematica Pura ed Applicata, 54 (1961), pp. 325–333.

[70] P. Holland and R. Welsch, *Robust regression using iteratively reweighted least-squares*, Communications in Statistics - Theory and Methods, 6 (1977), pp. 813–827.

[71] P. Huber, *Robust estimation of location parameter*, The Annals of Mathematical Statistics, 35 (1964), pp. 73–101.

[72] A. Ibáñez and F. Zapatero, *Monte Carlo valuation of American options through computation of the optimal exercise frontier*, Journal of Financial and Quantitative Analysis, 39 (2004), pp. 253–275.

[73] S. Jain and C. Oosterlee, *Pricing higher-dimensional American options using the stochastic grid method.* Available at SSRN: http://ssrn.com/abstract=1723712, 2010.

[74] S. Joe and F. Kuo, *Remark on algorithm 659: implementing Sobol's quasirandom sequence generator*, ACM Transactions on Mathematical Software, 29 (2003), pp. 49–57.

[75] C. Jonen, *Valuing high-dimensional American-style derivatives: A Robust Regression Monte Carlo method*, to be submitted.

[76] ——, *An efficient implementation of a Least Squares Monte Carlo method for valuing American-style options*, International Journal of Computer Mathematics, 86 (2009), pp. 1024–1039.

[77] M. Jonen, *Die Black-Scholes Formel: Die Konstruktion von Minimal-Algorithmen bei vorgegebener Genauigkeit*, Diploma Thesis, University of Cologne, Cologne, Germany, 2010.

[78] C. Joy, P. Boyle and K. Tan, *Quasi-Monte Carlo methods in numerical finance*, Management Science, 42 (1996), pp. 926–938.

[79] S. Juneja and H. Kalra, *Variance reduction techniques for pricing American options using function approximations*, Journal of Computational Finance, 12 (2009), pp. 79–102.

[80] I. Karatzas and S. Shreve, *Brownian Motion and Stochastic Calculus*, Springer, New York, USA, 1998.

[81] A. King and R. Rockafellar, *Asymptotic theory for solutions in statistical estimation and stochastic programming*, Mathematics of Operations Research, 18 (1993), pp. 148–162.

[82] L. Kocis and W. Whiten, *Computational investigations of low-discrepancy sequences*, ACM Transactions on Mathematical Software, 23 (1997), pp. 266–294.

[83] R. Korn and S. Müller, *The decoupling approach to binomial pricing of multi-asset options*, Journal of Computational Finance, 12 (2009), pp. 1–30.

[84] H. Kushner and G. Yin, *Stochastic approximation algorithms and applications*, Springer, New York, USA, 1997.

[85] D. Lamberton and B. Lapeyre, *Introduction to Stochastic Calculus Applied to Finance*, vol. 2nd, Chapman & Hall/CRC, London, England, 2000.

[86] A. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM, Philadelphia, Pennsylvania, USA, 2005.

[87] A. Laude and C. Jonen, *Biomass and CCS: The influence of learning effect*, to be submitted.

[88] P. L'Ecuyer, *Quasi-Monte Carlo methods with applications in finance*, Finance and Stochastics, 13 (2009), pp. 307–349.

[89] P. L'Ecuyer, C. Lècot and B. Tuffin, *A randomized quasi-Monte Carlo simulation method for Markov chains*, Operations Research, 56 (2008), pp. 958–975.

[90] C. Leentvaar and C. Oosterlee, *Multi-asset option pricing using a parallel Fourier-based technique*, Journal of Computational Finance, 12 (2008), pp. 1–26.

[91] F. Longstaff and E. Schwartz, *Valuing American options by simulation: a simple least-squares approach*, Review of Financial Studies, 14 (2001), pp. 113–147.

[92] K. Madsen and H. Nielsen, *Finite algorithms for robust linear regression*, BIT Numerical Mathematics, 30 (1990), pp. 682–699.

[93] M. Mair and J. Maruhn, *On the Primal-Dual Algorithm for Callable Bermudan Options*, Technical Report, to be submitted, 2011.

[94] G. Marsaglia and W. Tsang, *The ziggurat method for generating random variables*, Journal of Statistical Software, 5 (2000), pp. 1–7.

[95] M. Matsumoto and T. Nishimura, *Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator*, ACM Transactions on Modeling and Computer Simulation, 8 (1998), pp. 3–30.

[96] N. Moreni, *A variance reduction technique for American option pricing*, Physica A: Statistical Mechanics and its Applications, 338 (2004), pp. 292–295.

[97] M. Moreno and J. Navas, *On the robustness of Least-Squares Monte Carlo (LSM) for pricing American derivatives*, Review of Derivatives Research, 6 (2003), pp. 107–128.

[98] B. Moro, *The full Monte*, Risk, 8 (1995), pp. 57–58.

[99] B. Moskowitz and R. Caflisch, *Smoothness and dimension reduction in quasi-Monte Carlo methods*, Mathematical and Computer Modelling, 23 (1996), pp. 37–54.

[100] M. Musiela and M. Rutkowski, *Martingale Methods in Financial Modelling*, vol. 2nd, Springer, Berlin, Germany, 2007.

[101] H. Niederreiter, *Low-Discrepancy Simulation*, to appear in: Handbook of Computational Finance, J. Duan, W. Härdle and J. Gentle, eds., Springer, Berlin, Germany.

[102] ——, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, Pennsylvania, USA, 1992.

[103] B. Nielsen, O. Skavhaug and A. Tveito, *Penalty methods for the numerical solution of American multi-asset option problems*, Journal of Computational and Applied Mathematics, 222 (2008), pp. 3–16.

[104] H. Nieuwenhuis and M. Vellekoop, *A tree-based method to price American options in the Heston model*, Journal of Computational Finance, 13 (2009), pp. 1–21.

[105] O. Nikodym, *Sur une généralisation des intégrales de M. J. Radon*, Fundamenta Mathematicae, 15 (1930), pp. 131–179.

[106] J. Nocedal and S. Wright, *Numerical Optimization*, vol. 2nd, Springer, New York, USA, 2006.

[107] A. Novikov, *On an identity for stochastic integrals*, Theory of Probability and its Applications, 17 (1973), pp. 717–720.

[108] D. O'Leary, *Robust regression computation using iteratively reweighted least squares*, SIAM Journal of Matrix Analysis and Applications, 11 (1990), pp. 466–480.

[109] S. Paskov and J. Traub, *Faster valuation of financial derivatives*, Journal of Portfolio Management, 22 (1995), pp. 113–120.

[110] D. Pollard, *Convergence of Stochastic Processes*, Springer, New York, USA, 1984.

[111] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, England, 3rd ed., 2007.

[112] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.

[113] H. Robbins and S. Monro, *A stochastic approximation method*, Annals of Mathematical Statistics, 22 (1951), pp. 400–407.

[114] C. Rogers, *Monte Carlo valuation of American options*, Mathematical Finance, 12 (2002), pp. 271–286.

[115] H. Royden, *Real Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 3rd ed., 1988.

[116] M. Rubinstein, *Return to oz*, Risk, 7 (1994), pp. 67–71.

[117] R. Seydel, *Tools for Computational Finance*, vol. 4th, Springer, Berlin, Germany, 2009.

[118] A. Shapiro, *Asymptotic behavior of optimal solutions in stochastic programming*, Mathematics of Operations Research, 18 (1993), pp. 829–845.

[119] S. Shreve, *Stochastic Calculus for Finance II: Continuous-Time Models*, Springer, New York, USA, 2004.

[120] I. Sobol, *On the distribution of points in a cube and the approximate evaluation of integrals*, USSR Computational Mathematics and Mathematical Physics, 7 (1967), pp. 86–112.

[121] J. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, Hoboken, New Jersey, USA, 2003.

[122] Y. Su and M. Fu, *Optimal importance sampling in securities pricing*, Journal of Computational Finance, 5 (2002), pp. 27–50.

[123] Y. Tian, Z. Zhu and F. Klebaner, *Pricing barrier and American options under the SABR model on the graphics processing unit*, Concurrency and Computation: Practice and Experience, doi: 10.1002/cpe.1771 (2011).

[124] J. Tilley, *Valuing American options in a path simulation model*, Transactions of the Society of Actuaries, 45 (1993), pp. 83–104.

[125] J. Tsitsiklis and B. van Roy, *Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives*, IEEE Transactions on Automatic Control, 44 (1999), pp. 1840–1851.

[126] ——, *Regression methods for pricing complex American-style options*, IEEE Transactions on Neural Networks, 12 (2001), pp. 694–703.

[127] B. Tuffin, *On the use of low-discrepancy sequences in Monte Carlo methods*, Monte Carlo Methods and Applications, 2 (1996), pp. 295–320.

[128] J. van der Corput, *Verteilungsfunktionen*, Akademie van Wetenschappen, 38 (1935), pp. 813–821.

[129] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, USA, 1998.

[130] R. Walpole and R. Myers, *Probability and Statistics for Engineers and Scientists*, Macmillan Publishing Company, New York, USA, 1989.

[131] R. Wolke, *Iteratively reweighted least squares: a comparison of several single step algorithms for linear models*, BIT Numerical Mathematics, 32 (1992), pp. 506–524.

[132] D. Zanger, *Convergence of a least-squares Monte Carlo algorithm for bounded approximationg sets*, Applied Mathematical Finance, 16 (2009), pp. 123–150.

[133] W. Zhou and X. Chen, *Global convergence of a new hybrid Gauss-Newton structured BFGS method for nonlinear least squares problems*, SIAM Journal on Optimization, 20 (2010), pp. 2422–2441.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Herrn Prof. Dr. Rüdiger Seydel betreut worden.

Nachfolgend genannte Teilpublikationen liegen vor:

C. Jonen, An efficient implementation of a Least Squares Monte Carlo method for valuing American-style options, International Journal of Computer Mathematics, 86, 2009, pp. 1024-1039.

C. Jonen, Valuing High-Dimensional American-Style Derivatives: A Robust Regression Monte Carlo Method, to be submitted.

Ich versichere, dass ich alle Angaben wahrheitsgemäß nach bestem Wissen und Gewissen gemacht habe und verpflichte mich, jedmögliche, die obigen Angaben betreffenden Veränderungen, dem Dekanat unverzüglich mitzuteilen.

....
Ort, Datum Unterschrift