Feasibility and Efficiency of Monte Carlo Based Calibration of Financial Market Models

Dissertation

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Dem Fachbereich IV der Universität Trier
vorgelegt von

Christoph Käbe

Trier, 28. Januar 2010

Zusammenfassung

Zur Bewertung von Finanzmarktprodukten, die nicht liquide gehandelt werden, benötigen Händler ein Preismodell, das zuvor an Marktdaten kalibriert wurde. Für die Entwicklung eines Optimierungsalgorithmus, der das auftretenden Kalibrierungsproblem löst, sind vor allem zwei Voraussetzungen zu nennen.

Zunächst ist seit der Erfindung des Modells von Black und Scholes im Jahre 1973 eine Entwicklung zu komplizierteren Modellen mit zum Beispiel stochastischer Volatilität oder lokaler Volatiliät zu beobachten. Diese Entwicklung erfordert eine möglichst flexible Methode zur Approximation der Modellpreise. Änderungen der Modellstruktur sollten durch möglichst wenige Änderungen der Implementierung realisierbar sein. In diesem Zusammenhang ist die Monte Carlo Simulation in Kombination mit einem Diskretisierungsverfahren zur approximativen Lösung der stochastischen Differentialgleichung eine gute Wahl. Bekanntermaßen ist jedoch die Konvergenzgeschwindigkeit der Monte Carlo Methode sehr langsam, was in direktem Widerspruch zur zweiten Voraussetzung an den Algorithmus steht. Damit das entwickelte Programm in der Praxis angewendet werden kann, muss die Preisbewertung zeitnah stattfinden können.

Aus diesem Grund ist eines der beiden Hauptziele dieser Arbeit die Beschleunigung des Kalibrierungsalgorithmus. Zunächst ist es wünschenswert Methoden der differenzierbaren Optimierung zur Lösung des Kalibrierungsproblems zu verwenden, da diese sich durch hohe Konvergenzgeschwindigkeiten auszeichnen, vor allem im Vergleich zu ableitungsfreien Verfahren. Es wird sich aber zeigen, dass die Zielfunktion im Allgemeinen nicht differenzierbar ist, so dass diese durch zweimal stetig differenzierbare Polynome angenähert werden muss. Des Weiteren wird sich herausstellen, dass die Berechnung des Gradienten der Zielfunktion, der für den Optimierungsalgorithmus erforderlich ist, effizient über eine adjungierte Gleichung berechnet werden kann. Vor allem im Vergleich zum häufig verwendeten Finiten Differenzen Verfahren führt die Adjungierte zu einer deutliche Effizienzsteigerung. Darüberhinaus werden verschiedene andere Methoden entwickelt, beschrieben und angewendet, wie zum Beispiel ein Multi Layer Verfahren. Die Idee dabei ist die Approximation der Zielfunktion zu Beginn der Optimierung mit sehr grober Genauigkeit, d.h. mit wenigen Monte Carlo Simulationen, wenigen Zeitschritten für die Diskretisierung der stochastischen Differentialgleichung und einem großen Glättungsparameter, und diese Genauigkeit während der Optimierung sukzessive zu eröhen. Vor allem in Fällen, in denen der gewählte Startwert nicht bereits nahe beim Optimum liegt, erweist sich diese Technik als sehr hilfreich. Des Weiteren führt das Speichern möglichst vieler Zufallszahlen für die Realisierung der Brownschen Zuwächse im Arbeitsspeicher sowie die Parallelisierung der Monte Carlo Simulation zu einer deutlichen Beschleunigung des Algorithmus. So kann z.B. eine Kalibrierung mit $100,000$ Simulationen,

einer Schrittweite von $\Delta t = 5 \times 10^{-3}$ und einem Glättungsparamter von $\epsilon = 3.1 \times 10^{-3}$ durch eine Kombination der vorgestellten Techniken von zunächst 1.5 Stunden auf 6 Minuten reduziert werden. Für den Fall, dass die zu bestimmenden Parameter stückweise zeitabhängig auf 10 Zeitintervallen gewählt werden, kann sogar eine Reduktion von 5.5 Stunden auf lediglich 10 Minuten erreicht werden.

Des Weiteren widmet sich diese Arbeit einer zweiten Fragestellung. Wie oben beschrieben, werden die Optionspreise mit Hilfe der Monte Carlo Simulation, der Diskretisierung der zugrundeliegenden stochastischen Differentialgleichung und der Ausglättung von Nichtdifferenzierbarkeitsstellen approximiert. Daraus resultieren folglich drei Fehlerquellen: der Monte Carlo, der Diskretisierungs- und der Glättungsfehler. Obwohl eine Lösung des approximierenden Problems intuitiv auch eine Lösung des eigentlichen Problems annähert, ist dies im Allgemeinen keineswegs der Fall. Ziel ist es also, zu zeigen, dass eine Folge von kritischen Punkten erster Ordnung, erzeugt durch Lösen des Optimierungsproblems mit zunehmender Anzahl von Monte Carlo Simulationen sowie Diskretisierungsschritten und einem abnehmenden Glättungsparameter gegen einen kritischen Punkt erster Ordnung des eigentlichen Problems konvergiert. Dies kann erwartungsgemäß nur unter bestimmten Voraussetzungen gezeigt werden, im konkreten Fall z.B. unter der Bedingung, dass die Koeffizienten der Differentialgleichung Lipschitz stetig sind.

Contents

Preface

First of all, I would like to thank my advisor Prof. Dr. Ekkehard W. Sachs for his outstanding support during my time as a PhD student at the University of Trier and numerous fruitful discussions but also for the opportunity to participate in international conferences. It was a very pleasant and interesting time as a member of his working group. Furthermore, I would like to thank Prof. Dr. Michael B. Giles for many helpful discussions and advices as well as for accepting the position as a referee of this thesis.

This work was supported by industry projects with the Financial Engineering Equities, Commodities and Funds division of the UniCredit Bank AG situated in Munich, Germany. I would like to thank Dr. Jan H. Maruhn, Alexander Giese and Frank Gerlich for the faithful cooperation. I am especially indebted to Dr. Maruhn, who supported me far beyond the scope of the projects.

I would also like to thank my fellow colleagues at the department of mathematics of the University of Trier, in particular: Claudia Schillings, Timo Hylla, Andre Lörx, Matthias Schu, Roland Stoffel, Stephan Schmidt, Christian Wagner, Nils Langenberg and Benjamin Rosenbaum for the extraordinary atmosphere and the moral support.

I am also very grateful to Miriam Machwitz for her enduring support over the last years. Last but not least, I would like to thank my parents allowing my studies with their encouragement and financial support.

Christoph Käbe
Trier, January 2010

Glossary

Abbreviations

AD	Automatic differentiation
(a.s.)	Almost surely
CPU	Central processing unit
EMS	Euler-Maruyama scheme
KKT	Karush-Kuhn-Tucker
LICQ	Linear independence constraint qualification
PDE	Partial differential equation
SAA	Sample average approximation
SDE	Stochastic differential equation
SQP	Sequential quadratic programming

Symbols and Variables

B	Number of time intervals for parameters
B_t	Riskless bond
δ	Dividend yield
$\mathcal{E}$	LSQ value based on the Monte Carlo function evaluations
$\mathcal{E}^*$	"true" LSQ value resulting the closed form parameters
η	Derivative of SDE solution, i.e. solution of sensitivity equation
I	Number of call options
K	Strike price of a European call option
κ	Mean reversion speed
$\mathcal{L}, \mathcal{L}_a, \mathcal{L}_{a,x}$	Lipschitz constants
λ	Adjoint variable
M	Number of Monte Carlo simulations
N	Number of discretization time steps
$N(\mu, \sigma)$	Normal distribution with mean μ and variance σ
$(\Omega, \mathcal{F}, P)$	Probability space
ϕ	Multiplier that absorbs the initial condition of the variance process

Q	Number of parameter types, i.e. $P = BQ$
r	Risk free rate
$\mathbb{R}_+$	Set of nonnegative real numbers
ρ	Correlation coefficient
σ	Volatility
S_0	Todays value of the underlying or spot
S_t	First component of Y_t
T	Maturity of a European call option
θ	Mean reversion level
W_t	Brownian motion at time point t
X	Feasible set
Y_t	Solution of SDE at time point t
$Y_{t,\epsilon}$	Solution of smoothed SDE
y_n	Solution of discretized but unsmoothed SDE at n-th discretization point
$y_{n,\epsilon}$	Solution of discretized and smoothed SDE at n-th discretization point
$\Rightarrow$	Convergence in distribution

Functions and Operations

$a \wedge b$	$\min(a, b)$
$a_\epsilon(\cdot)$	Smoothed SDE drift
$b_\epsilon(\cdot)$	Smoothed SDE diffusion
$C^i(x)$	Model based prices of i-th European call options
C^i_{obs}	Observed market prices of i-th European call options
$e^{\#}(\cdot)$	Penalty term
$E_P(\cdot)$	Expected value with respect to probability measure P
f	Objective function of the true problem
$f_{\mathrm{M},\Delta t}$	Objective function of the approximating problem with Monte Carlo and SDE discretization
$f_{\mathrm{M},\Delta t,\epsilon}$	Objective function of the smoothed approximating problem with Monte Carlo and SDE discretization
g	Objective function of the simplified problem
g_ϵ	Objective function of the smoothed simplified problem
$g_{\Delta t,\epsilon}$	Objective function of the smoothed simplified problem with SDE discretization
$g_{\mathrm{M},\Delta t,\epsilon}$	Objective function of the smoothed simplified problem with Monte Carlo and SDE discretization

$J_R(\cdot)$	Jacobian of residual vector
$L(\cdot)$	Lagrangian function
$P(\|)$	Conditional probability
$\pi_\epsilon(\cdot)$	Smoothed maximum function
$\psi(\cdot)$	Coefficient error function
$R(\cdot)$	Residual vector containing market and model price residuals
$\Theta_\gamma(\cdot)$	Merit function with parameter γ
$U(x)$, $U(x, \delta)$	Neighborhood of x, respectively with radius δ
x^+, $\pi(x)$	$\max(x, 0)$
$\chi(t)$	Mapping that maps time point t to the previous discretization point

Chapter 1

Introduction

1.1 Motivation and Literature Review

Financial derivatives have gained considerable importance in the last decade which is for instance reflected in the development at the EUREX, one of the world's largest derivatives exchanges. From 1998 to 2008 the number of traded contracts increased from approximately 250 million to 2,200 million (figure 1.1). But not only

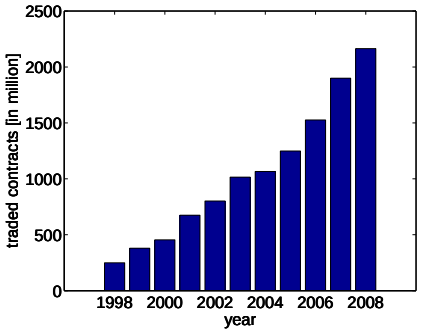

Figure 1.1: Total number of traded contracts at the EUREX from 1998 to 2008 in million.

the number of products, also the variety grew in a remarkable way. In addition to plain vanilla call and put options, exotic products like barrier options or cliquets, just to name but a few, are frequently traded. As traders in the banks need accurate pricing information, it is required that the pricing model has been adapted to the market, i.e. the model has to be calibrated to a set of liquidly traded instruments. In a simple Black-Scholes model (Black and Scholes [1973]) for example, the task would be to choose the volatility parameter, such that market prices are accurately predicted by the model. For plain-vanilla call or put options, this problem can be solved explicitly (see also Lemma 2.23). However, smile and skew patterns of

market prices have led to generalizations of the Black-Scholes model: stochastic volatility models, local volatility models, models based on jump diffusion or even combinations of those are nowadays used by practitioners. As a consequence, the calibration method becomes as crucial as the pricing model itself.

Considering that the fair price of a call option depends on uncertain future changes of the underlying, e.g. the stock, it might be tempting to use stochastic optimization methods to solve the arising calibration problem. Hamida and Cont [2005] for instance set up an evolutionary algorithm. Practitioners seem to prefer these methods as well as derivative free algorithms. Mikhailov and Nögel [2004] compared the *generalized reduced gradient method* implemented in Microsoft Excel with *simulated annealing* for the calibration of Heston's stochastic volatility model (Heston [1993]). However, these methods are well known to converge very slow.

Alternatively one can make use of the large variety of deterministic optimization algorithms in combination with suitable numerical methods for the approximation of the pricing models, as closed form solutions are only rarely available. In this context Sachs and Schu [2007], Sachs and Strauss [2008], Coleman et al. [1999] or Kindermann et al. [2008] discretize the associated pricing partial (integro) differential equation and embed the approximation within the calibration framework. For the few cases where the distribution of the underlying pricing model is known, *fast fourier transformation* indeed is fast (see for instance Kilin [2007]). Gerlich et al. [2006] achieve a very good performance for the calibration of Heston's model, where they use the closed-form solution in a feasibility perturbed sequential quadratic program algorithm. These highly specialized methods on the one hand lead to quickly converging algorithms. On the other hand it is often hard or even impossible to adapt the resulting codes to changes of the model dynamics—in particular if the number of stochastic drivers of the model increases.

Moreover, if flexibility and ease of implementation come into play, a calibration based on Monte Carlo in combination with a discretization of the corresponding stochastic differential equation (SDE) may be the method of choice, since it can be programmed rather quickly and allows to switch easily the model dynamics, even if the dimension of the problem increases. The drawback, however, is the well-known slow convergence of the Monte Carlo method. In particular, if the number of model parameters is large, the calibration may take several hours until convergence of the method is achieved. This holds true especially if the gradient of the objective function is computed via finite differences. Hence it is desirable to search for more efficient ways to compute the gradients.

In this context, the *Likelihood Ratio Method* is based on differentiating the probability density defined by the model for the underlying stock dynamics. An introduction to this method is for instance given in Broadie and Glasserman [1996]. Unfortunately, this method requires the probability density of the model dynamics

which is only known for a few financial market models. The *Pathwise Method* (see for instance Glasserman [2003] pp. 386 ff. or Broadie and Glasserman [1996]) is based on the closed-form solution of the model defining the dynamics of the underlying. In absence of such a solution formula, the pathwise method leads to the sensitivity equation, which suffers from the same computational effort as the finite difference approach. A totally different but evolving approach is *automatic differentiation* (AD), introduced for instance in Griewank and Corlis [1991]. However, tests show that the reverse mode of AD leads to highly specialized codes, which is hard to adapt to changing model dynamics. Giles and Glasserman [2006] demonstrate how an adjoint method can be used to significantly reduce the computation time for option sensitivities in a Libor market model. Furthermore, Giles [2007] shows that the development of adjoint codes can be assisted by making use of AD, but that the automatically derived backward equations are less efficient than their hand-coded counterparts. Motivated by the potential performance gains it seems to be advantageous to apply adjoint methods within a Monte Carlo calibration framework. However, in most applications the option payoff or the coefficients of the stochastic differential equations describing the financial market model are unfortunately not differentiable everywhere such that gradient methods as well as the adjoint calculus are not immediately applicable to Monte Carlo approximations of the calibration problem. This is the case for e.g. the constant elasticity of variance-model (Cox [1996]), the Heston model (Heston [1993]), the Hull-White stochastic volatility model (Hull and White [1987]), the 3/2-model (Lewis [2000]) or the SABR-model of Hagan *et al.* (Hagan et al. [2002]).

Thus, after applying Monte Carlo and discretizing the underlying stochastic differential equation, an application of differentiable optimization methods and the adjoint calculus requires to smooth out potential non-differentiabilities. This in summary leads to an approximation of the original calibration problem based on three error sources, which raises the question, if a solution of this problem is an approximation of a solution of the true problem. Rubinstein and Shapiro [1993] prove convergence in the sense of a first order critical point for an approximation based on only Monte Carlo. Shapiro [2000] proves convergence under the assumption that the optimization problem produces a global minimum. The case of an optimization problem that produces a complete set of solutions has been examined by Robinson [1996]. Bastin et al. [2006] consider additional second order optimality conditions and stochastic constraints. However, the literature does not provide any convergence theory if the approximation is based on multiple error sources.

1.2 Summary of the Thesis

This thesis introduces a calibration problem for financial market models based on a Monte Carlo approximation of the option payoff and a discretization of the underlying stochastic differential equation. As motivated above, it is desirable to benefit from fast deterministic optimization methods to solve this problem. To be able to achieve this goal, possible non-differentiabilities are smoothed out with an appropriately chosen twice continuously differentiable polynomial. On the basis of this so derived calibration problem, this work is essentially concerned about two issues.

First, the question occurs, if a computed solution of the approximating problem, derived by applying Monte Carlo, discretizing the SDE and preserving differentiability is an approximation of a solution of the true problem. Unfortunately, this does not hold in general but is linked to certain assumptions. It will turn out, that a uniform convergence of the approximated objective function and its gradient to the true objective and gradient can be shown under typical assumptions, for instance the Lipschitz continuity of the SDE coefficients. This uniform convergence then allows to show convergence of the solutions in the sense of a first order critical point. Furthermore, an order of this convergence in relation to the number of simulations, the step size for the SDE discretization and the parameter controlling the smooth approximation of non-differentiabilites will be shown. Additionally the uniqueness of a solution of the stochastic differential equation will be analyzed in detail.

Secondly, the Monte Carlo method provides only a very slow convergence, namely $\mathcal{O}(1/\sqrt{M})$ where M is the number of simulations. The numerical results in this thesis will show, that the Monte Carlo based calibration indeed is feasible if one is concerned about the calculated solution, but the required calculation time is too long for practical applications. Thus, techniques to speed up the calibration are strongly desired. As already mentioned above, the gradient of the objective is a starting point to improve efficiency. Due to its simplicity, finite differences is a frequently chosen method to calculate the required derivatives. However, finite differences is well known to be very slow and furthermore, it will turn out, that there may also occur severe instabilities during optimization which may lead to the break down of the algorithm before convergence has been reached. In this manner a sensitivity equation is certainly an improvement but suffers unfortunately from the same computational effort as the finite difference method. Thus, an adjoint based gradient calculation will be the method of choice as it combines the exactness of the derivative with a reduced computational effort. Furthermore, several other techniques will be introduced throughout this thesis, that enhance the efficiency of the calibration algorithm. A multi-layer method will be very effective in the case, that the chosen initial value is not already close to the solution. Variance reduction techniques are helpful to increase accuracy of the Monte Carlo estimator

and thus allow for fewer simulations. Storing instead of regenerating the random numbers required for the Brownian increments in the SDE will be efficient, as deterministic optimization methods anyway require to employ the identical random sequence in each function evaluation. Finally, Monte Carlo is very well suited for a parallelization, which will be done on several central processing units (CPUs).

These techniques to increase efficiency of a Monte Carlo based calibration algorithm, were developed in two papers, Käbe, Maruhn, and Sachs [2009] and Giese, Käbe, Maruhn, and Sachs [2007]. In the first, also the question of convergence has been briefly treated.

1.3 Outline

This thesis is structured as follows. Chapter 2 introduces some basic theory which will be frequently referred to throughout this thesis. The first parts contains important results from the area of probability theory, stochastic processes and stochastic differential equations. On the basis of this, some fundamental concepts of mathematical finance will be explained in the second part. Finally, important results of numerical analysis, in particular optimization, will be explained.

In the third chapter the calibration problem will be defined, beginning with a continuous version. Subsequently the underlying stochastic differential equation will be discretized with an Euler-Maruyama scheme in combination with a Monte Carlo approximation of the expected value. To be able to benefit from fast converging deterministic optimization methods, the differentiability of the objective function has to be ensured as a third step. The particularly chosen algorithm will be introduced in the last part.

Thus, the derivation of the approximating calibration problem described above results in three sources of errors, namely the Monte Carlo error, the time discretization error and the smoothing error. Consequently, the fourth chapter analyzes the convergence behavior of a solution of the approximating problem towards a solution of the true optimization problem. The first part deals with the existence and uniqueness of solutions of the stochastic differential equation under various assumptions. The second part contains a prove of first order optimality under conditions, that preliminarily allow to show a uniform convergence of the objective functions as well as the corresponding gradients.

The fifth chapter then considers the calculation of the objective's gradient. Initially, the finite difference method, which is a simple but expensive way to approximate the gradient, will be explained. As a first improvement the second part introduces the sensitivity equation. As this method suffers from the same complexity as the finite difference approximation, the third part will subsequently show how the calculation can be sped up with an adjoint method before the fourth part approves

this numerically. To round the topic out, alternative approaches like automatic differentiation are briefly explained and discussed in the last part.

Chapter 6 introduces a number of computational methods and techniques to reduce the overall calibration time. The first section deals with methods of variance reduction, that reduce the Monte Carlo estimator's variance which consequently allows for a smaller number of simulations. Secondly, a multi layer method will be introduced, where the idea is to increase the accuracy of the objective function evaluations during optimization. The third section then explains the idea of storing the random numbers instead of regenerating them every time they are needed which is finally followed by parallelizing the algorithm.

The seventh chapter presents numerical results to determine the efficiency and the theoretical coherence of the Monte Carlo calibration method developed in this thesis.

As the adjoint method presented in chapter 5 is not immediately applicable if one leaves the model class of diffusion processes and allows for the possibility of jumps, chapter 8 shows that transforming the model allows a significant adjoint-based calibration speedup.

Finally, chapter 9 summarizes this thesis with an an outlook on potential future work.

Chapter 2

Theoretical Background

This chapter introduces the basic theory which will be frequently referred to throughout this thesis. Section 2.1 starts with an introduction to probability theory, stochastic processes and stochastic differential equations. On the basis of this, some fundamental concepts of mathematical finance will be explained in section 2.2. The last part finally contains important results of numerical analysis, in particular optimization.

2.1 Fundamentals of Stochastic Processes

Throughout this thesis the existence of a probability space $(\Omega, \mathcal{F}, P)$ will be assumed, where $\mathcal{F}$ is the sigma algebra over the set $\Omega \neq \emptyset$ and P an adequate probability measure. In this manner, an event $E \in \mathcal{F}$ is said to happen *almost surely* (a.s.) if it happens with probability one, thus if $P(E) = 1$.

Let $\mathcal{T}$ be a set with $\mathcal{T} \subset \mathbb{R}_+$. A family $(\mathcal{F}_t)_{t\in\mathcal{T}}$ of sigma algebras is called *filtration* if $\mathcal{F}_s \subset \mathcal{F}_t, \forall s \leq t$. Heuristically one can say that the filtration $\mathcal{F}_t$ contains all information available up to time t. A mapping $\tau : \Omega \to \mathcal{T} \cup \{\infty\}$ is called *stopping time* if $\{\omega \in \Omega : \tau(\omega) \leq t\} \subset \mathcal{F}_t$ for a given filtration $(\mathcal{F}_t)_{t\in\mathcal{T}}$. Consider a measurable space (Ξ, Σ). A family $(X_t)_{t\in\mathcal{T}}$ of random variables with $X_t : (\Omega, \mathcal{F}) \to (\Xi, \Sigma)$ is called *stochastic process*. For practical applications, (Ξ, Σ) is often chosen as $(\mathbb{R}^m, \mathcal{B}^m)$, where $\mathcal{B}^m$ is the *Borel sigma algebra*. Furthermore, for a fixed $\omega \in \Omega$, $X_t(\omega) : \Omega \to \mathbb{R}^m$ describes a *path* of the stochastic process. An important example for stochastic processes is the *Brownian motion*:

Definition 2.1. *Let $(\Omega, \mathcal{F}, P)$ be a probability space with filtration $(\mathcal{F}_t)_{t\in\mathcal{T}}$. A stochastic process $(W_t)_{t\in\mathcal{T}}$ is called* Brownian motion *if*

(i) $W_0 = 0$ *(a.s.)* .

(ii) The increments $W_t - W_s$ are independent from $\mathcal{F}_s$, $\forall s, t \in \mathcal{T} \quad 0 \leq s < t$.

(iii) $W_t - W_s$ *are independent and normally distributed with mean* 0 *and variance* $t - s$*, i.e.* $W_t - W_s \sim N(0, t-s), \quad \forall s, t \in \mathcal{T} \quad 0 \leq s < t.$

Furthermore, a stochastic process $(W_t)_{t\in\mathcal{T}}$ *with* $W_t = (W_t^1, ..., W_t^m)$ *is a multidimensional Brownian motion, if* $(W_t^1)_{t\in\mathcal{T}}, ..., (W_t^m)_{t\in\mathcal{T}}$ *are independent Brownian motions. Brownian motions are alternatively called Wiener processes.*

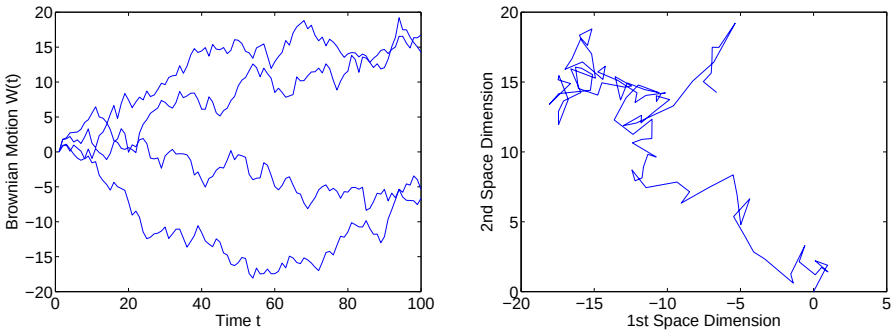

Figure 2.1: Some Brownian paths (left side) and two dimensional motion of a particle in a container filled with gas (right side).

Four different paths of a Brownian motion are illustrated in figure 2.1 on the left side. This special stochastic process was named after the Scottish botanist Robert Brown who analyzed the motion of pollen through a microscope in 1827. This has been modelled with a two dimensional Brownian motion on the right side of figure 2.1.

Financial market models are often and in particular in this thesis formulated on the basis of Ito stochastic differential equations. For their introduction, the Ito stochastic integral is a crucial issue. As a first step, this integral can be defined for processes which are *simple predictable*.

Definition 2.2. *A process* $(H_t)_{t\in\mathbb{R}_+}$ *is called* simple predictable *if there exists a bounded and* $\mathcal{F}_{t_n}$ *measurable process* $(Z_n)_{n=1,...,N}$ *such that*

$$H_t(\omega) = \sum_{n=0}^{\infty} Z_n(\omega) I_{(t_n, t_{n+1}]}(t) \text{ with } 0 = t_0 \leq ... \leq t_n \to \infty$$

and

$$I_{(t_n,t_{n+1}]}(t) = \begin{cases} 1 & ; \quad t \in (t_n, t_{n+1}] \\ 0 & ; \quad \text{else.} \end{cases}$$

The name arises from the observation, that the value of the process over the whole interval $(t_n, t_{n+1}]$ is already known in t_n. Consequently, $(H_t)_{t\in\mathbb{R}_+}$ is often also called

step processes or elementary processes. In this simple case, it is natural to set

$$\int_0^t H_s(\omega)dW_s(\omega) = \sum_{n=0}^{\infty} Z_n(\omega)(W_{t_{n+1}\wedge t}(\omega) - W_{t_n\wedge t}(\omega))$$

where $t_n \wedge t := \min(t_n, t)$. To extend this integral to a broader class of processes or functions consider the following definition.

Definition 2.3. *A function $f : [0, T] \times \Omega \to \mathbb{R}$ belongs to the set $\mathcal{L}_T^2$ if*

(i) f is jointly $\mathcal{L} \times \mathcal{F}$ measurable.

(ii) $\int_0^T E(f(t,\cdot)^2)dt < \infty$.

(iii) $E(f(t,\cdot)^2) < \infty$ for each $0 \leq t \leq T$.

(iv) $f(t,\cdot)$ is $\mathcal{F}_t$ measurable for each $0 \leq t \leq T$.

where $\mathcal{L}$ is the sigma algebra of Lebesgue measurable subsets of $\mathbb{R}$.

As one now can show that the set of simple predictable functions is dense in $\mathcal{L}_T^2$ equipped with the norm $\|f(T,\cdot)\| := \sqrt{\int_0^T E(f(t,\cdot)^2)dt}$ (Kloeden and Platen [1999], Lemma 3.2.1), every $\mathcal{L}_T^2$ function can be approximated by a simple predictable function to any accuracy. Thus for an arbitrary function $f \in \mathcal{L}_T^2$ there exists a sequence of simple predictable functions f_n which approximate f in the above defined norm, i.e.

$$\int_0^t E((f_n(t,\cdot) - f(t,\cdot))^2)dt \to 0 \quad (n \to \infty).$$

The Ito stochastic integral with respect to a Brownian motion can now be defined in the following way.

Definition 2.4. (Ito Stochastic Integral) *Let $(W_t)_{t\in[0,T]}$ be a Brownian motion and $f \in \mathcal{L}_T^2$. The Ito stochastic integral is defined as*

$$\int_0^t f(t,\cdot)dW_t := \lim_{n\to\infty} \int_0^t f_n(t,\cdot)dW_t$$

with f_n a sequence of simple predictable functions satisfying

$$\int_0^t E((f_n(t,\cdot) - f(t,\cdot))^2)dt \to 0 \quad (n \to \infty).$$

This integral has the following important properties:

Lemma 2.5. *For a function $f \in \mathcal{L}_T^2$ it holds true that:*

(i) $E\left(\left(\int_0^t f(s,\cdot)dW_s\right)^2\right) = \int_0^t E\left(f(s,\cdot)^2\right)ds$ *(Ito isometry),*

(ii) $X_t(\omega) := \int_0^t f(s,\omega)dW_s$ *has almost surely continuous paths.*

Proof. (i) Kloeden and Platen [1999] p. 84

(ii) Arnold [1973] pp. 96 f.

□

A combination of a white noise containing this so defined Ito stochastic integral and an ordinary differential equation is a stochastic differential equation, defined in the following.

Definition 2.6. *Let $W_t = (W_t^1, ..., W_t^L)^T$ be a L-dimensional vector of Brownian motions and $a : [0,T] \times \mathbb{R}^L \to \mathbb{R}^L$ as well as $b : [0,T] \times \mathbb{R}^L \to \mathbb{R}^L \times \mathbb{R}^L$. The following equation is called* stochastic differential equation*:*

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t,\ X_0 = c,\ 0 \leq t \leq T < \infty$$

or componentwise

$$dX_t^I = a^I(t, X_t)dt + \sum_{j=1}^{L} b^{I,j}(t, X_t)dW_t^j, \quad I = 1, ..., L.$$

This SDE can also be written in integral form:

$$X_t = X_0 + \int_0^t a(s, X_s)ds + \int_0^t b(s, X_s)dW_s,$$

where the first integral is the standard Riemann integral and the second the stochastic Ito integral defined above. X_t is called an Ito process*, $a(t, X_t)dt$ drift and $b(t, X_t)dW_t$ diffusion.*

Note that the literature alternatively provides the Stratonovich integral. Without loss of generality and due to the fact that finance theory uses Ito's calculus almost exclusively, it will always be referred to Ito's stochastic differential equation throughout this thesis.

A direct consequence of Lemma 2.5 (ii) is the path continuity of an Ito process:

Lemma 2.7. *Let $\sqrt{|a|}$, $b \in \mathcal{L}_T^2$ and X_t an Ito process, i.e.*

$$X_t(\omega) = X_0(\omega) + \int_0^t a(s,\omega)ds + \int_0^t b(s,\omega)dW_s(\omega).$$

X_t has almost surely continuous paths.

Proof. Define $A(t,\omega) := \int_0^t a(s,\omega)ds$ and $B(t,\omega) := \int_0^t b(s,\omega)dW_s(\omega)$. By definition $A(\cdot,\omega)$ is continuous and Lemma 2.5 (ii) provides the path continuity of $B(\cdot,\omega)$ almost surely. Consequently X_t has almost surely continuous paths. □

Skorokhod [1965] proved the existence of a SDE solution under relatively mild conditions, namely the continuity of the coefficients and an additional linear growth constraint.

Theorem 2.8. *Consider*

$$dX_t = a(t,X_t)dt + b(t,X_t)dW_t,\ X_0 = c,\ 0 \le t \le T \tag{2.1}$$

and suppose that the following conditions hold:

(i) The mappings $a(t,\cdot)$ and $b(t,\cdot)$ are continuous for $t \in [0,T]$.

(ii) There exists a constant $\mathcal{G} > 0$ such that $\forall t \in [0,T]$ and $y \in \mathbb{R}^L$

$$\|a(t,y)\| + \|b(t,y)\| \le \mathcal{G}(1 + \|y\|),$$

where $\|\cdot\|$ is a vector or respectively matrix norm, for instance the Euclidian norm $\|y\| := \sum_{i=1}^n y_i^2$ for $y \in \mathbb{R}^n$ or $\|Y\| := \sum_{i,j=1}^{n,m} Y_{ij}^2$ for $Y \in \mathbb{R}^{n\times m}$.

Then (2.1) *has almost surely a bounded solution.*

Proof. Skorokhod [1965] pp. 59 f. □

As the uniqueness of such a solution is a more complex issue, section 4.1 will address this in detail.

In the context of stochastic processes and stochastic differential equations, the Ito formula is certainly one of the most important results.

Theorem 2.9. (Ito Formula) *Consider a stochastic process $(X_t)_{t\in\mathcal{T}}$ following the SDE*

$$dX_t = a(t,X_t)dt + b(t,X_t)dW_t$$

with $\sqrt{|a|}$, $b \in \mathcal{L}_T^2$ *and a mapping* $f : \mathcal{T} \times \mathbb{R}^L \to \mathbb{R}^m$ *with continuous partial derivatives* $\frac{\partial f(s,X_s)}{\partial s}$, $\frac{\partial f(s,X_s)}{\partial X_s}$ *and* $\frac{\partial^2 f(s,X_s)}{\partial X_s^2}$. *Then,* f *follows the integral equation*

$$\begin{aligned} f(t,X_t) &= f(0,X_0) + \int_0^t \Big(\tfrac{\partial f(s,X_s)}{\partial s} + a(s,X_s)\tfrac{\partial f(s,X_s)}{\partial X_s} \\ &\quad + \tfrac{1}{2} b(s,X_s)^2 \tfrac{\partial^2 f(s,X_s)}{\partial X_s^2} \Big) ds + \int_0^t b(s,X_s)\tfrac{\partial f(s,X_s)}{\partial X_s} dW_s \end{aligned}$$

or in differential form

$$\begin{aligned} df(t,X_t) &= \Big(\tfrac{\partial f(t,X_t)}{\partial t} + a(t,X_t)\tfrac{\partial f(t,X_t)}{\partial X_t} + \tfrac{1}{2} b(t,X_t)^2 \tfrac{\partial^2 f(t,X_t)}{\partial X_t^2} \Big) dt \\ &\quad + b(t,X_t)\tfrac{\partial f(t,X_t)}{\partial X_t} dW_t. \end{aligned}$$

Proof. See for instance Kloeden and Platen [1999] pp. 92 ff. □

The following two inequalities are additional important results, which will become helpful for the convergence analysis in chapter 4.

Lemma 2.10. (Gronwall Inequality) *Let* $\alpha, \beta : \mathcal{T} \to \mathbb{R}$ *integrable with*

$$0 \leq \alpha(t) \leq \beta(t) + L \int_0^t \alpha(s) ds$$

for $t \in \mathcal{T}$ *and* $L > 0$. *Then*

$$\alpha(t) \leq \beta(t) + L \int_0^t e^{L(t-s)} \beta(s) ds.$$

Proof. e.g. Kloeden and Platen [1999] pp. 129 ff. □

Theorem 2.11. (Jensen's Inequality) *Let* X *an integrable random variable taking values in* $I \subset \mathbb{R}$. *For every convex function* f *and every concave function* g *defined on* I *it is essential that*

$$f(E(X)) \leq E(f(X))$$

and

$$g(E(X)) \geq E(g(X)).$$

Proof. Bauer [2002] p. 23. □

In the area of probability theory, there exists a large variety of different kinds of convergence concepts. For the analysis in chapter 4, first and foremost the following two definitions will be used.

Definition 2.12. (Convergence Almost Surely) *A sequence of random variables $(X_n)_n$ is said to converge almost surely to a random variable X if*

$$P\left(\left\{\omega \in \Omega : \lim_{n\to\infty} |X_n(\omega) - X(\omega)| = 0\right\}\right) = 1.$$

It is written

$$X_n \underset{n\to\infty}{\longrightarrow} X \quad (a.s.).$$

This convergence is also called convergence with propability one.

Definition 2.13. (Convergence in Distribution) *A sequence of random variables $(X_n)_n$ with distribution functions F_n converge* in distribution *to a random variable X with distribution F if*

$$\lim_{n\to\infty} F_n(x) = F(x), \quad \forall x \in \mathbb{R}.$$

This convergence is denoted by $X_n \Rightarrow X$.

The following theorem addresses the permutability of limit and integral.

Theorem 2.14. (Lebesgue's Dominated Convergence Theorem) *Let $f_n, f : \Omega \to \mathbb{R}^m \cup \{\infty\}$ measurable and $f_n \longrightarrow_{n\to\infty} f \quad (a.s.)$. If there exists an additional integrable function g defined on Ω with $|f_n| \leq g$, $\forall n \in \mathbb{N}$ one obtains*

$$\lim_{n\to\infty} \int_\Omega f_n dP = \int_\Omega f dP.$$

Proof. Bauer [1992] p. 96. □

In this thesis, the expected value occurring in the call price formula (see Definition 2.20) will be approximated with Monte Carlo simulation. The idea of this method is the estimation of a random variables' expected value by calculating the mean of a large number of realizations. This is motivated by the law of large numbers:

Theorem 2.15. (Law of Large Numbers)
Let $(X_n)_n$ be a sequence of independent and identically distributed random variables and suppose that $E(X_1)$ exists. Then it holds

$$\lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} X_m = E(X_1) \quad (a.s.).$$

Proof. Bauer [2002] pp. 86 ff. □

The determination of the quality of such an estimator is frequently realized by bias and root mean square error.

Definition 2.16. (Bias / Root Mean Square Error) *Let $\hat{X}$ be an estimator of $E(X)$. The accuracy of this estimator can be calculated via the* bias

$$E\left(\hat{X} - X\right)$$

and the root mean square error

$$\sqrt{E\left(\left(\hat{X} - X\right)^2\right)}.$$

In particular the Monte Carlo estimator is unbiased as

$$E\left(\frac{1}{M}\sum_{m=1}^{M} X_m\right) = \frac{1}{M}\sum_{m=1}^{M} E\left(X_m\right) = \frac{1}{M}\sum_{m=1}^{M} E\left(X_1\right) = E(X_1). \tag{2.2}$$

A detailed overview on the mentioned topics can be found in Bauer [2002], Bauer [1992], Feller [1970a], Feller [1970b] or Karatzas and Shreve [1991] for probability theory, stochastic calculus and Brownian motions, Kloeden and Platen [1999] or Arnold [1973] for stochastic differential equations and Glasserman [2003] for Monte Carlo simulation.

2.2 Financial Markets

On the basis of section 2.1 some fundamental concepts of financial markets will be explained in the following. Initially, several typical assumptions are stated.

Remark 2.17. *The following properties of financial markets are assumed to hold in the latter of this section:*

(i) The market is liquid, i.e. arbitrary amounts of assets are always available.

(ii) Market participants can sell assets they do not hold. This is called short selling.

(iii) It is possible to buy fractional quantities of assets.

(iv) There are no transaction costs, no dividend yields and no arbitrage, i.e. riskless returns.

Some assumptions may be contrary to intuition like the absence of transaction costs but they are required to model the real world. Under these assumptions the considered financial market model can be introduced.

Definition 2.18. *Let $(\Omega, \mathcal{F}, P)$ be a probability space, $(W_t)_{t\in[0,T]}$ a L-dimensional Brownian motion and $(\mathcal{F}_t)_{t\in[0,T]}$ the augmented filtration generated by $(W_t)_{t\in[0,T]}$*

under an equivalent martingale measure Q. The financial market model is generated by the stochastic processes $(B_t)_{t\in[0,T]}$ and $(S_t)_{t\in[0,T]} = (S_t^1, ..., S_t^L)_{t\in[0,T]}^T$ defined as the solution of the $L+1$ dimensional system of stochastic differential equations:

$$dS_t = rS_t dt + \sigma^l S_t dW_t \quad S_0 \in (0,\infty) \quad 0 \le t \le T \tag{2.3}$$

$$dB_t = rB_t dt \quad B_0 \in (0,\infty) \quad 0 \le t \le T. \tag{2.4}$$

r is the risk free rate, i.e. the premium of a risk free bond $(B_t)_{t\in[0,T]}$ and σ^l the volatility of the l-th stock $(S_t^l)_{t\in[0,T]}$.

Thus, in this model each stock follows a Black-Scholes SDE (Black and Scholes [1973]). In the latter of this work, this will be expanded to a more general model which also contains stochastic or local volatility models. In addition to the stocks and bonds, the considered market provides the possibility to buy or sell European options.

Definition 2.19. (European Call/Put Option) *A European call (put) option is the right to buy (sell) an underlying, e.g. stock, at a given future time T, called maturity, for a given price K, denoted as strike.*

To understand how market participants benefit from call or put options, consider two companies closing a contract for a product delivery after N years. If for instance the delivering company accounts in Euro and the receiving company pays in US dollar, the first should be aware of changing exchange rates. However, as a put option provides the right to sell dollar for a fixed amount of Euros after N years, such an option can be used to hedge against exchange rate fluctuations. Though buying this right is not for free, this price is known whereas the future exchange rate fluctuations are uncertain.

The question then arises, what is the fair price, denoted as $C(S_t, t)$ of such an option. The fundamental theorem of asset pricing (e.g. Karatzas and Shreve [1998]) states, that an arbitrage-free price of a European call/put option with maturity T and strike K is given by the discounted expected future payoff of the option:

Definition 2.20. (Price of a European Call/Put Option)
The price of a European call/put option under the risk neutral measure is defined as

$$\text{Call:} \quad C(S_0,0) = e^{-rT} E_Q(\max(S_T - K, 0))$$
$$\text{Put:} \quad P(S_0,0) = e^{-rT} E_Q(\max(K - S_T, 0))$$

where T is the maturity and K the strike of option.

In the selected situation of Definition 2.18 a solution formula for $C(S_0, 0)$ exists. This will be derived in the following. Note that $P(S_0, 0)$ can be treated analogously.

Consider a portfolio Π_t containing $b(t)$ bonds as well as $s(t)$ stocks at time t and one sold option $C(S_t, t)$. The value of this portfolio in t can thus be calculated via

$$\Pi_t = b(t)B_t + s(t)S_t - C(S_t, t). \tag{2.5}$$

This portfolio is assumed to be *self financing*, which implies that

$$d\Pi_t = b(t)dB_t + s(t)dS_t - dC(S_t, t). \tag{2.6}$$

Hence, portfolio shifts from stocks to bonds and vice versa can exclusively be financed from the existing portfolio. Additionally, this portfolio is riskless, i.e.

$$d\Pi_t = r\Pi_t dt \tag{2.7}$$

with the same premium as the bond. If the portfolio rate would differ from the rate of the bond, this would allow *arbitrage*. The following Lemma now describes the price of the option as the solution of a partial differential equation (PDE).

Lemma 2.21. *Consider the portfolio* Π_t *from* (2.5). *The option price* $C(S_t, t)$ *follows the partial differential equation*

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0. \tag{2.8}$$

Proof. Definition 2.18 describes S_t as an Ito process following the Black-Scholes SDE. Ito's Lemma (Theorem 2.9) now provides that

$$dC = \left(\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2}\right) dt + \sigma S \frac{\partial C}{\partial S} dW. \tag{2.9}$$

Inserting (2.3), (2.4) and (2.9) in (2.6) leads to

$$\begin{aligned} d\Pi_t &= \left(brB + srS - \tfrac{\partial C}{\partial t} - rS\tfrac{\partial C}{\partial S} - \tfrac{1}{2}\sigma^2 S^2 \tfrac{\partial^2 C}{\partial S^2}\right) dt \\ &\quad + \left(s\sigma S - \sigma S \tfrac{\partial C}{\partial S}\right) dW. \end{aligned} \tag{2.10}$$

Due to (2.7) any randomness has to be eliminated in (2.10). This can be achieved by choosing $s = \frac{\partial C}{\partial S}$. Note that this implies to choose the number of stocks according to the sensitivity of the option price with respect to the stock price, which is called *delta hedging*. On the other hand it follows from (2.7) and (2.5) that

$$d\Pi_t = r\Pi_t dt = r\left(bB + \frac{\partial C}{\partial S}S - C\right) dt. \tag{2.11}$$

Identifying (2.11) with (2.10) provides

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0.$$

which means the proof of the statement. □

Due to the equivalence of the Black-Scholes SDE (2.3) and the above PDE, (2.8) is denoted as Black-Scholes PDE. If one introduces boundary conditions, the Black-Scholes formula which provides a solution for (2.8) can be proven subsequently. These boundary and final conditions are

$$C(0,t) = 0, \quad C(S,t) \underset{S\to\infty}{\longrightarrow} S, \quad C(S,T) = \max(S-K).$$

It is clearly true that the call price at maturity T has to be equal to the payoff $max(S-K,0)$. If today's stock price is zero, nobody would be willing to buy the option for any positive price and for a fixed strike the option price should converge to S_0 if the latter converges to infinity.

Theorem 2.22. *The Black-Scholes PDE* (2.8) *with boundary condition* $C(0,t) = 0$, $C(S,t) \underset{S\to\infty}{\longrightarrow} S$ *and final condition* $C(S,T) = \max(S-K)$ *has the solution*

$$C(S,t) = S\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \quad S > 0, \quad 0 \le t \le T \tag{2.12}$$

with $\Phi(x)$ *the distribution function of the standard normal distribution, i.e.*

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} dy$$

and d_1 *and* d_2 *defined as*

$$d_1 = \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}.$$

Proof. Black and Scholes [1973] □

Hence, the Black-Scholes price of a European call option, depends on the current value of the underlying S_0, also called the spot, option maturity T, strike K, interest rate r and volatility σ. For trading purposes, this implies that the option price changes with changing spot. To avoid this, consider the following Lemma, which describes a bijective mapping between a given Black Scholes call price $\overline{C}$ and volatility σ assuming arbitrage bounds which will be explained subsequently. First, the cost of a call option should never be more than today's value of the stock. Otherwise, one could sell an option for $\overline{C}$ and buy the stock for S_0. If the value of

the stock has exceeded the strike at maturity, the stock is sold to the option holder for K. Otherwise, the option will not be exercised. In any case, one gains at least $\overline{C} - S_0$ for selling the option and buying the stock. Furthermore the option price should not be less than $\max(S - Ke^{-r(T-t)}, 0)$. To be more precise, assuming the opposite the option would provide a return of $\max(e^{r(T-t)}S - K, 0)$ at maturity. Thus, today's value is this discounted future value, i.e. $\max(S - Ke^{-r(T-t)}, 0)$. If this is larger than the price, buying the option is worth it in any situation. Now consider the lemma addressing the relation between price and volatility.

Lemma 2.23. *Under the assumption that* $\max(S_0 - Ke^{-r(T-t)}, 0) \leq \overline{C} \leq S_0$ *for a known Black Scholes price* $\overline{C}$ *the mapping*

$$\sigma \to C(\sigma) - \overline{C}$$

has a unique root.

Proof. Consider the case that $\sigma = 0$. It holds by definition of d_1 and d_2 that $d_1 = d_2 = \infty$ and thus $\Phi(d_1) = \Phi(d_2) = 1$. For $\sigma = \infty$ it follows analogously that $d_1 = \infty$ and $d_2 = -\infty$ and consequently $\Phi(d_1) = 1$ as well as $\Phi(d_2) = 0$. Inserting this in the solution formula (2.12) provides

$$C^{\mathrm{BS}}(\sigma) - \overline{C}^{\mathrm{BS}} = \begin{cases} S - Ke^{-r(T-t)} - \overline{C}^{\mathrm{BS}} & \leq \quad 0 \quad ; \quad \sigma = 0 \\ S - \overline{C}^{\mathrm{BS}} & \geq \quad 0 \quad ; \quad \sigma = \infty. \end{cases}$$

Thus $\sigma \to C^{\mathrm{BS}}(\sigma) - \overline{C}^{\mathrm{BS}}$ has at least one root. For the uniqueness, it remains to show that $C^{\mathrm{BS}}(\sigma)$ is monotone. Consider therefore that

$$\Phi'(d_1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{d_1^2}{2}}, \quad \Phi'(d_2) = \frac{1}{\sqrt{2\pi}} e^{-\frac{d_2^2}{2}}. \tag{2.13}$$

It can be seen, that

$$\frac{\Phi'(d_1)}{\Phi'(d_2)} = e^{-\frac{1}{2}\left(d_1^2 - d_2^2\right)} = e^{-\frac{1}{2}(d_1 - d_2)(d_1 + d_2)}.$$

As $d_1 - d_2 = \sigma\sqrt{T-t}$ and $d_1 + d_2 = \frac{2\ln(S/K) + 2r(T-t)}{\sigma\sqrt{T-t}}$ it holds that

$$\frac{\Phi'(d_1)}{\Phi'(d_2)} = K\frac{e^{-r(T-t)}}{S}$$

and thus

$$S\Phi'(d_1) = Ke^{-r(T-t)}\Phi'(d_2). \tag{2.14}$$

Thus

$$\frac{\partial C^{\mathrm{BS}}(\sigma)}{\partial \sigma} = S\Phi'(d_1)\frac{\partial d_1}{\partial \sigma} - Ke^{-r(T-t)}\Phi'(d_2)\frac{\partial d_2}{\partial \sigma}.$$

Inserting (2.14) and

$$\begin{aligned}\frac{\partial d_1}{\partial \sigma} &= \frac{\sigma^2(T-t)\sqrt{T-t} - \left(\ln(S/K) + r(T-t) + \frac{1}{2}\sigma^2(T-t)\right)\sqrt{T-t}}{\sigma^2(T-t)} \\ \frac{\partial d_2}{\partial \sigma} &= \frac{-\sigma^2(T-t)\sqrt{T-t} - \left(\ln(S/K) + r(T-t) - \frac{1}{2}\sigma^2(T-t)\right)\sqrt{T-t}}{\sigma^2(T-t)}\end{aligned}$$

provides

$$\frac{\partial C^{\mathrm{BS}}(\sigma)}{\partial \sigma} = S\Phi'(d_1)\left(\frac{\partial d_1}{\partial \sigma} - \frac{\partial d_2}{\partial \sigma}\right) = S\sqrt{T-t}\Phi'(d_1)$$

which is strictly positive due to (2.13) which completes the proof. □

Thus for every call price, there exists a unique volatility parameter σ, called *implied volatility*. In a *sticky-strike* scenario, the implied volatility does not change for a fixed strike and changing spot. Similarly in a *sticky-moneyness* situation, a constant difference of spot and strike, the so called moneyness, leads to a constant implied volatility. As these situations can be observed in many markets, the implied volatility provides more stability than the Black Scholes option price. Supported by Lemma 2.23, practitioners therefore trade implied volatilities. An example of a whole set of implied volatilities will be given in section 7.1.

A more detailed introduction to the topic of financial markets is given in Karatzas and Shreve [1998].

2.3 Numerical Optimization

Generally speaking, optimization describes the minimization or maximization of a function, for example the least squares difference of model and market prices for European call options, which is the scope of this thesis. Subsequently, important basics of numerical optimization will be explained.

Consider initially the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^P} f(x)$$

where $f : \mathbb{R}^P \to \mathbb{R}$. The following definitions summarize the basic concepts of local and global minima:

Definition 2.24. *Let $f : X \to \mathbb{R}$ with $X \subset \mathbb{R}^P$. A point $x^* \in X$ is denoted as*

(i) *global minimizer if $f(x^*) \leq f(x)$, $\forall x \in X$.*

(ii) *local minimizer if there exists a neighborhood $U(x^*)$ of x^* such that $f(x^*) \leq f(x)$, $\forall x \in X \cap U(x^*)$.*

(iii) *strict global minimizer* *if* $f(x^*) < f(x)$, $\forall x \in X, x \neq x^*$.

(iv) *strict local minimizer* *if there exists a neighborhood* $U(x^*)$ *of* x^* *such that* $f(x^*) < f(x)$, $\forall x \in X \cap U(x^*), x \neq x^*$.

The Taylor series expansion is important for the study of local minimizers.

Theorem 2.25. (Taylor Series Expansion) *Let* $I \subset \mathbb{R}$ *be a subset,* $f : I \to \mathbb{R}$ *k-times continuously differentiable and* $a \in I$. *Then, it holds for all* $x \in I$*:*

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k + \mathcal{O}\left((x-a)^{n+1}\right).$$

Proof. Forster [1999] p. 226 f. □

If the function f is smooth, there are efficient ways, to identify local minimizers. This is motivated by the following two Theorems.

Theorem 2.26. *If* x^* *is a local minimizer of f and f continuously differentiable then* $\nabla f(x^*) = 0$.

Proof. Nocedal and Wright [1999], p. 15. □

A point satisfying Theorem 2.26 is called *critical point first order* or *stationary point*. A sufficient condition for a local minimizer is provided by the following definition.

Theorem 2.27. *Suppose that f is twice continuously differentiable,* $\nabla f(x^*) = 0$ *and* $\nabla^2 f(x^*)$ *positive definite.Then* x^* *is a strict local minimizer.*

Proof. Nocedal and Wright [1999] p. 16. □

In contrast to these conditions for unrestricted problems, the first order necessary optimality condition in the restricted case

$$\min_{x \in X} f(x) \tag{2.15}$$

where $X \subset \mathbb{R}^P$ nonempty, convex and closed, is as follows:

Theorem 2.28. *Let f continuously differentiable and X nonempty, convex and closed. If* x^* *is a local minimizer of* (2.15) *it follows that*

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad \forall x \in X.$$

This condition is called variational inequality.

Proof. Assume that $x^* \in X$ is a local minimizer and

$$\nabla f(x^*)^T(x - x^*) < 0$$

for a $x \in X$. Consider a second point $\tilde{x} := x^* + c(x - x^*)$ for $c \in [0, 1]$. As X is convex, it holds that $\tilde{x} \in X$. An application of the Taylor series expansion (Theorem 2.25) now provides

$$f(\tilde{x}) = f(x^*) + c\nabla f(x^*)^T(x - x^*) + \mathcal{O}(c).$$

$\nabla f(x^*)^T(x - x^*)$ is negative and thus

$$f(\tilde{x}) < f(x^*)$$

for a sufficiently small $c > 0$ which is contrary to the local minimizer assumption. □

For practical applications often the more general formulation of an optimization problem subject to equality and inequality constraints on the variables is considered:

$$\begin{aligned} & \min_{x \in \mathbb{R}^P} f(x) \\ \text{s.t.} \quad & c_i(x) = 0,\ i = 1, ..., m \\ & c_i(x) \leq 0,\ i = m+1, ..., m+n \end{aligned} \tag{2.16}$$

with $f : \mathbb{R}^P \to \mathbb{R}$, $c : \mathbb{R}^P \to \mathbb{R}^{m+n}$. To take the constraints into account the Lagrangian function is a linear combination of these involving additional Lagrange multipliers:

Definition 2.29. (Lagrangian Function) *Consider f and c from* (2.16). $L : \mathbb{R}^P \times \mathbb{R}^{m+n} \to \mathbb{R}$ *with*

$$L(x, \lambda) := f(x) + \sum_{i=1}^{m+n} \lambda_i c_i(x)$$

is called Lagrangian function and $\lambda \in \mathbb{R}^{m+n}$ *denoted as Lagrange multiplier (vector).*

To be able to define optimality conditions, the following set of active inequality constraint indices and the subsequent linear independence constraint qualification are fundamental.

Definition 2.30. (Active Set) *The active set* $\mathcal{A}(x)$ *at any feasible x consists of the equality constraints and those indices of the inequality constraint for which* $c_i(x) = 0$, *i.e.*

$$\mathcal{A}(x) := \left\{ i = 1, ..., m \right\} \cup \left\{ i \in \{m+1, ..., m+n\} \,\middle|\, c_i(x) = 0 \right\}.$$

Those components of $c(x)$ for which $c_i(x) = 0$ are called active, the others consequently inactive.

Definition 2.31. (Linear Independence Constraint Qualification) *It is said that the linear independence constraint qualification (LICQ) holds for a point x and the active set $\mathcal{A}(x)$ if the set of active constraint gradients $\{\nabla c_i(x), i \in \mathcal{A}\}$ is linearly independent.*

Now the necessary first order optimality conditions for (2.16) can be introduced. They will be crucial for the derivation of the optimization method to solve the considered calibration problem in section 3.4.

Theorem 2.32. (Karush-Kuhn-Tucker) *Assume that x^* is a local minimizer of* (2.16)*, that the function f and c are continuously differentiable and that the LICQ holds at x^*. Then there exists a Lagrange multiplier vector $\lambda^* \in \mathbb{R}^{m+n}$ such that the following conditions are satisfied*

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= 0 \\ c_i(x^*) &= 0 \quad i = 1, ..., m \\ c_i(x^*) &\leq 0 \quad i = m+1, ..., m+n \\ \lambda_i^* c_i(x^*) &= 0 \quad i = m+1, ..., m+n \\ \lambda^* &\geq 0 \quad i = 1, ..., m. \end{aligned}$$

These conditions are called Karush-Kuhn-Tucker (KKT) conditions.

Proof. Nocedal and Wright [1999] pp. 323 ff. □

The books of Nocedal and Wright [1999], Geiger and Kanzow [2002] or Bonnans et al. [2003] provide a detailed overview on the topic of optimization.

Chapter 3

An Optimization Problem for the Calibration of Financial Market Models

In this chapter the calibration problem for the calibration of standard European call options will be defined, beginning with a continuous version. Subsequently a discretized version will be introduced in the second part, obtained by applying Monte Carlo simulation and a discretization scheme to approximate the SDE solution. As a third step, the differentiability of the objective function will be ensured. In particular, this will be achieved by smoothing non-differentiabilities with a twice continuously differentiable polynomial. The chosen optimization method will be introduced in the last part.

3.1 Calibration Problem

The focus lies in the calibration of the parameter vector $x = (x_1, ..., x_P)^T \in \mathbb{R}^P$ of an equity-type stock price model, which will be defined later in this chapter, to a given set of European call options (Definition 2.19) with pricing formula (Definition 2.20)

$$C(x) := e^{-rT} E_Q(\max(S_T(x) - K, 0)).$$

The dynamics of the underlying stock are described by a L+1-dimensional system of stochastic differential equations

$$\begin{aligned} dY_t(x) &= a(x, Y_t(x))dt + b(x, Y_t(x))dW_t\,,\ Y_0 \in (0, \infty)\,,\ 0 \le t \le T \quad (3.1) \\ dB_t &= rB_t dt, \quad B_0 \in (0, \infty). \end{aligned}$$

Here $W_t := (W_t^1, ..., W_t^L)^T$ is a L-dimensional vector of Brownian motions, B_t a riskless bond and thus $r > 0$ the corresponding risk-free rate. The mappings $a : \mathbb{R}^P \times \mathbb{R}^L \to \mathbb{R}^L$ and $b : \mathbb{R}^P \times \mathbb{R}^L \to \mathbb{R}^L \times \mathbb{R}^L$ satisfy conditions, such that a solution of (3.1) exists. A more detailed analysis on this topic follows in section 4.1. The dimensions of a and b differ, because the possible correlation between the components of W_t is incorporated in b. Thus, $a(x, Y_t(x))dt$ denotes the componentwise integral $a^l(x, Y_t(x))dt$, $l = 1, ..., L$ and $b(x, Y_t(x))dW_t$ is to understood in the sense of $\sum_{j=1}^{L} b^{j,l}(x, Y_t(x))dW_t^j$ like already introduced in Definition 2.6. As the first component of the solution Y_t of (3.1) describes the dynamics of the underlying stock, it is denoted as S_t to keep the usual notation in the finance literature:

$$Y_t = [S_t, Y_t^2, ..., Y_t^L]^T \text{ , } 0 \leq t \leq T.$$

The general structure of (3.1) covers many interesting models in the finance sector, for instance the well known Heston stochastic volatility model (Heston [1993])

$$\begin{aligned} dS_t &= (r - \delta)S_t dt + \sqrt{v_t} S_t dW_t^1 \text{ , } S_0 \in (0, \infty) \text{ , } 0 \leq t \leq T \\ dv_t &= \kappa(\theta - v_t)dt + \sigma\sqrt{v_t}(\rho dW_t^1 + \sqrt{1 - \rho^2} dW_t^2) \text{ , } v_0 \in (0, \infty) \end{aligned} \tag{3.2}$$

with $L = 2$, $Y_t^1 = S_t$ denoting the stock-price at time t, δ is the dividend yield and $Y_t^2 = v_t$ is the variance, following a mean-reversion process with mean-reversion speed κ, mean-reversion level θ, volatility σ and correlation coefficient ρ. In this process, the variance v_t tends to a long term variance level θ with speed κ. This model is strongly related to the interest rate model of Cox, Ingersoll and Ross (Cox et al. [1985]), where the same mean reverting square root process has been used. Other models covered by (3.1) are for instance the Black-Scholes model with constant volatility (Black and Scholes [1973]), the stochastic volatility models of Stein and Stein [1991] or Hull and White [1987], the stochastic interest rate model of Vasicek [1977] or the local volatility model (e.g. Dupire [1994]).

Apart from the concrete model choice, it is usually necessary to employ a set of feasible vectors $X \subset \mathbb{R}^P$, which for instance may contain lower and upper bounds for every single parameter:

$$lb_p \leq x_p \leq ub_p \text{ , } p = 1, ..., P.$$

It might occur, that additional constraints need to be employed. Feller [1951] proved for instance that a process following the Cox, Ingersoll and Ross model stays positive, if the *Feller constraint* $2\kappa\theta \geq \sigma^2$ — alternatively denoted as Novikov condition — is employed. An example for such a process is Heston's variance process. This may help avoiding problems with the stock price process crossing over to the imaginary domain (see Section 3.3 for a more detailed discussion). Note

that the Feller constraint as well as the box constraints lead to a convex and compact set. Hence, the first assumption for the calibration problem is stated here:

$(A.1)$ $\quad X \neq \emptyset$ is a convex and compact subset of $\mathbb{R}^P$.

This assumption will be helpful for the convergence analysis in chapter 4 and is also not restrictive due to the comments above.

Let C^i_{obs} denote the observed market price and $C^i(x)$ the model price of an option with maturity T_i and strike K_i for a set of options $i = 1, ..., I$. Note, that T_i and K_i are not necessarily different. If one now defines the objective function as a least squares function, the calibration problem can be formulated as follows:

$$\begin{aligned} & \min_{x \in X} f(x) := \sum_{i=1}^{I} \left(C^i(x) - C^i_{\text{obs}} \right)^2 \\ \text{where} \quad & C^i(x) = e^{-rT_i} E_Q \left(\max(S_{T_i}(x) - K_i, 0) \right), \; i = 1, ..., I \\ \text{s.t.} \quad & dY_t(x) = a(x, Y_t(x))dt + b(x, Y_t(x))dW_t, \;\; Y_0 > 0 \\ & 0 \leq t \leq T, \quad T := \max_{i=1,...,I} T_i. \end{aligned} \tag{P}$$

The next section deals with a first approximation of this problem.

3.2 Discretization of the Problem

For the solution of the problem (P) the calculation of the expectation functional $E_Q(\cdot)$ is a key point. For some models, fitting the notation of (3.1), (semi-)closed form solutions are available, for example for Heston's modell (3.2) (see Heston [1993]). However, in most cases there exists no such explicit solution formula, so that numerical methods come into play. According to the discussion in chapter 1, a Monte Carlo simulation is considered here for the approximation of the expected value functional. Following the law of large numbers (Theorem 2.15), one obtains

$$E_Q(\max(S_T(x) - K, 0)) \approx \frac{1}{M} \sum_{m=1}^{M} (\max(s^m_T(x) - K, 0)), \tag{3.4}$$

for M sufficiently large, where s^m_T denotes the m-th random sample or realization of the solution of (3.1) for $m = 1, ..., M$.

The remaining question then is, how to calculate these realizations. Obviously, if one knows the joint distribution defined by the modell (3.1), one can sample directly from this distribution. This is the fact in the Black-Scholes modell for instance, where the stock price process follows a *geometric Brownian motion* (see also Example 5.12). Unfortunately, in most cases, this is not possible, such that alternative methods become desirable. Broadie and Kaya [2006], generate samples

recursively form parts of the system of SDEs and thus receive a realization of the exact distribution. The advantage certainly is the relatively high convergence order. Broadie and Kaya achieve an order of $\mathcal{O}(s^{-\frac{1}{2}})$ compared to $\mathcal{O}(s^{-\frac{1}{3}})$ for an Euler discretization in combination with Monte Carlo simulation (Duffie and Glynn [1995]) where s is the computational budget. Due to the complexity and the lack of computational speed this method cannot be recommended for practical implementations (Andersen [2007]).

Alternatively the SDE solution can be approximated with discretization schemes. The simplest time discrete approximation scheme is the *Euler-Maruyama scheme* (EMS). For a given time discretization

$$0 = \tau_0 < ... < \tau_N = T,$$

step size $\Delta t_n := (\tau_{n+1} - \tau_n)$ and $\Delta W_n := (W_{n+1} - W_n)$ for $n = 0, ..., N-1$ the increments of the vector of Brownian motions, the solution of the iterative Euler-Maruyama scheme

$$y^m_{n+1}(x) = y^m_n(x) + a(x, y^m_n(x))\Delta t_n + b(x, y^m_n(x))\Delta W_n\,,\ m = 1, ..., M \qquad (3.5)$$

is an approximation of the exact solution Y_T (e.g. Kloeden and Platen [1999]). The simplest choice for the step size would be an equidistant $h > 0$, such that $\Delta t_n = h\,,\ n = 0, ..., N-1$. However, in practice is is often required to fit different points in time $T_i\,,\ i = 1, ..., I$ with

$$0 = \tau_0 < ... < \tau_{N_1} = T_1 < \tau_{N_1+1} < ... < \tau_{N_2} = T_2 < ... < \tau_{N_I} = T_I = T.$$

Thus at least a different step size for every interval $[T_i, ..., T_{i+1}]$, which means choosing $\Delta t_n = h_i > 0\,,\ i = 1, ..., I$, might become necessary. In this context, let $\Delta t := \max_{n=0,...,N-1}(\Delta t_n)$.

On the one hand, a big advantage of the EMS is its implementability. Changing the model requires only few adaptions of the implementation. This suits perfectly the discussion of Monte Carlo simulation above. On the other hand, there exist other schemes with a higher rate of convergence, like the explicit or the implicit Milstein scheme. Without loss of generality, this work focuses on the EMS, as all steps are transferable to many other discretization schemes.

Reconsider that $\Delta W_n := (W_{n+1} - W_n)$ denote the increments of the vector of Brownian motions. These increments are normally distributed with mean zero and variance Δt_n (see Definition 2.1). In practice, the generation of sequences of random numbers on the computer, that follow a given distribution, leads to pseudo random numbers, as most generators naturally deliver deterministic instead of really random sequences. Section 6.3 will give a more detailed analysis on this topic.

Summarizing, i.e. applying (3.4) and (3.5) to problem (P), leads to

$$\begin{aligned} & \min_{x\in X} f_{M,\Delta t} := \sum_{i=1}^{I} \left(C^i_{M,\Delta t}(x) - C^i_{\text{obs}}\right)^2 \\ \text{where} \quad & C^i_{M,\Delta t}(x) := e^{-rT_i}\tfrac{1}{M}\sum_{m=1}^{M}\left(\max(s^m_{N_i}(x) - K_i, 0)\right),\ i = 1, ...I \\ \text{s.t.} \quad & y^m_{n+1}(x) = y^m_n(x) + a(x, y^m_n(x))\Delta t_n + b(x, y^m_n(x))\Delta W^m_n \\ & y^m_0 = Y_0,\ n = 0, ..., N-1,\ N := \max_{i=1,...,I} N_i,\ m = 1, ...M. \end{aligned} \tag{$P_{M,\Delta t}$}$$

This problem is denoted with two lower indices to underline the dependency of the number of Monte Carlo simulations M and the maximal discretization step size Δt. An immediate application of smooth optimization methods to this problem might lead to two main difficulties: positivity of the SDE solution and differentiability of the objective function. The next section deals with these two problems.

3.3 Preserving Positivity and Differentiability

A closer look to ($P_{M,\Delta t}$) reveals that fast converging gradient based methods are not immediately applicable.

Firstly, if one considers a square-root process, like in the Heston model (3.2), one has to take care of the process crossing over to the imaginary domain, like the following Lemma shows for the example of a mean reverting process:

Lemma 3.1. *Consider the mean reverting β-process*

$$dv_t = \kappa(\theta - v_t)dt + \sigma v_t^{\beta} dW_t,\ v_0 > 0,\ 0 \le t \le T \tag{3.6}$$

where $\frac{1}{2} \le \beta \le 1$ and $\kappa, \theta, \sigma > 0$. Then

(i) the solution $(v_t)_t$ of (3.6) takes with probability 1 an infinite time to reach zero if either $\frac{1}{2} < \beta \le 1$ or $\beta = \frac{1}{2}$ and $2\kappa\theta \ge \sigma^2$.

(ii) the solution $(v_t)_t$ of (3.6) reaches zero with probability 1 in finite time if $\beta = \frac{1}{2}$ and $2\kappa\theta < \sigma^2$.

Proof. Mao et al. [2006], pp. 5 ff. □

This result can be helpful when solving models including such a mean reverting β-process, as a negative value for v_t would imply problems with taking the square root. Thus for $\beta = \frac{1}{2}$ the Feller condition $2\kappa\theta \ge \sigma^2$ can be applied albeit for the cost of restricting the set of parameter values. Unfortunately, the Feller condition does not help if it is applied to the process from Lemma 3.1 discretized with the Euler-Maruyama scheme:

Lemma 3.2. *Consider the Euler-Maruyama discretized mean reverting β-process with $\beta = \frac{1}{2}$*

$$v_{n+1} = v_n + \kappa(\theta - v_n)\Delta t + \sigma\sqrt{v_n}\Delta W_n, v_0 > 0, \ n = 0, ..., N-1.$$

If $v_n > 0$ the conditional probability that $v_{n+1} < 0$ is strictly positive for any chosen discretization step size Δt.

Proof. e.g. Lord et al. [2006]. □

As a consequence of this, practitioners truncate the process in zero: $y_n = 0$, if $y_n < 0$ or reflect it: $y_n = -y_n$, if $y_n < 0$, see e.g. Gatheral [2004]. From a mathematical point of view, there may be better positivity preserving methods. Lord et al. [2006] introduce a technique called *full truncation*, which they apply to Heston's stochastic volatility model. This means replacing selected values of v_n by the truncated counterparts $\max(0, v_n)$, denoted as v_n^+:

$$\begin{aligned} S_{n+1} &= S_n + (r-\delta)S_n\Delta t_n + \sqrt{v_n^+}S_t\Delta W_n^1 \\ v_{n+1} &= v_n + \kappa(\theta - v_n^+)\Delta t_n + \sigma\sqrt{v_n^+}(\rho\Delta W_n^1 + \sqrt{1-\rho^2}\Delta W_n^2). \end{aligned}$$

They compared this scheme with several others, for instance the *partial truncation* of Deelstra and Delbaen [1998] and received a significantly lower bias and even more a root mean square error (Definition 2.16) of the same size as an exact scheme. The computation time compared to other positivity preserving schemes is the same.

Secondly, a closer look reveals, that $C^j_{M,\Delta t}(x)$ is not differentiable due to the maximum function. There are several ways to deal with this problem. Firstly, one might apply methods of non-differentiable optimization, e.g. methods based on the subgradient (Geiger and Kanzow [2002]), or stochastic search algorithms, which do not require any gradient information. However, as these methods are well known to converge very slowly, it is desirable to use smooth optimization algorithms. Thus the non-differentiability is smoothed out with an adequate polynomial π_ϵ:

$$\pi_\epsilon(x) := \begin{cases} 0 & , \ x \leq -\epsilon \\ -\frac{1}{16\epsilon^3}x^4 + \frac{3}{8\epsilon}x^2 + \frac{1}{2}x + \frac{3\epsilon}{16} & , \ -\epsilon < x < \epsilon \\ x & , \ x \geq \epsilon. \end{cases} \tag{3.7}$$

A comparison of coefficients shows, that for a given smoothing parameter $\epsilon > 0$, (3.7) is the polynomial with the smallest degree, which is a twice continuously differentiable approximation of the maximum function. The drawback of this approach is the approximation error. The following lemma quantifies this error.

Lemma 3.3. *For π_ϵ from (3.7) it holds true that*

$$\| \max(x, 0) - \pi_\epsilon(x)\|_\infty = \frac{3}{16}\epsilon.$$

Proof. By definition of π_ϵ one has

$$| \max(x, 0) - \pi_\epsilon(x)| = 0, \quad \forall x \in (-\infty, -\epsilon] \cup [\epsilon, \infty).$$

A simple extreme value analysis shows that $| \max(x, 0) - \pi_\epsilon(x)|$ attains its maximum at $x = 0$ (see also figure 3.1). Thus

$$\sup_{x\in\mathbb{R}} | \max(x, 0) - \pi_\epsilon(x)| = | - \pi_\epsilon(0)| = \frac{3}{16}\epsilon.$$

□

In addition, the mappings a and b in the SDE may not be differentiable, e.g. due to positivity preserving schemes like *full truncation*, which have been introduced above. Thus these mappings are smoothed with a polynomial similar to (3.7). Figure 3.1 shows the effect of smoothing the maximum as well as the absolute value function (if reflection instead of truncation is applied) — each with an adequate polynomial.

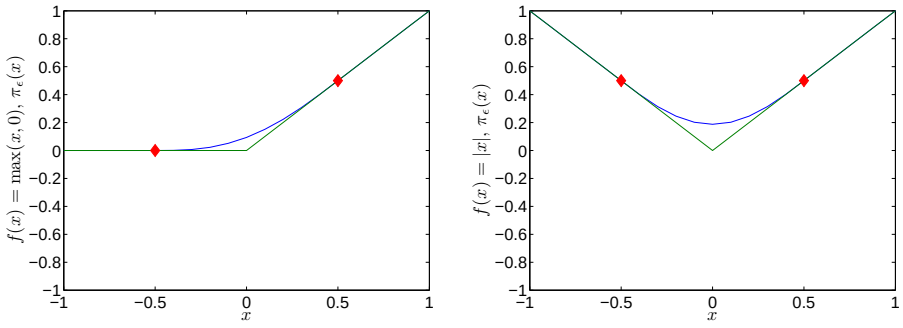

Figure 3.1: Smoothing property of polynomial $\pi_\epsilon(x)$ (blue line) from (3.7) to maximum function (green line) and a similar polynomial to absolute value function for $\epsilon = 0.5$ (red diamonds) and $-1 \leq x \leq 1$.

For the convergence analysis in the latter of this thesis, not only the error term $\| \max(x, 0) - \pi_\epsilon(x)\|_\infty$ but the resulting errors of the smoothed and unsmoothed coefficient functions, namely $\|a_\epsilon(x, y) - a(x, y)\|_\infty$ and $\|b_\epsilon(x, y) - b(x, y)\|_\infty$ will be important. To be able to derive preferably arbitrary convergence results, the

following assumption is stated

$$(A.2) \qquad \begin{array}{l} \|a_\epsilon(x,y)-a(x,y)\|_\infty^2+\|b_\epsilon(x,y)-b(x,y)\|_\infty^2 \leq \psi(\epsilon) \\ \text{with } \psi:\mathbb{R}_+\to\mathbb{R} \text{ and } \lim\limits_{\epsilon\to 0}\psi(\epsilon)=0. \end{array}$$

Several financial market models, for instance the Stein-Stein model (Stein and Stein [1991]), which will be used for the numerical convergence results in chapter 7, have coefficient functions with a linear structure such that the error $\psi(\epsilon)$ can be drilled down to the smoothing error from Lemma 3.3 and is thus of order $\mathcal{O}(\epsilon^2)$. However, the above introduced Heston model (3.2) would provide a coefficient error order of $\mathcal{O}(\epsilon)$ due to the introduced square root function. This result is based on the Hölder continuity of the square root function (see also Remark 4.5).

To facilitate notation no difference will be made between the resulting three polynomials and smoothing parameters ϵ in the following. Replacing $\max(\cdot)$, $a(\cdot)$ and $b(\cdot)$ by their smoothed counterparts $\pi_\epsilon(\cdot)$, $a_\epsilon(\cdot)$ and $b_\epsilon(\cdot)$ the optimization problem can now be written as follows:

$$(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon}) \qquad \begin{array}{ll} & \min\limits_{x\in X} f_{M,\Delta t,\epsilon}(x) := \sum\limits_{i=1}^{I}\left(C^i_{M,\Delta t,\epsilon}(x)-C^i_{\text{obs}}\right)^2 \\ \text{where} & C^i_{M,\Delta t,\epsilon}(x) := e^{-rT_i}\frac{1}{M}\sum\limits_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon}(x)-K_i)\right),\ i=1,...,I \\ \text{s.t.} & y^m_{n+1,\epsilon}(x)=y^m_{n,\epsilon}(x)+a_\epsilon(x,y^m_{n,\epsilon}(x))\Delta t_n+b_\epsilon(x,y^m_{n,\epsilon}(x))\Delta W^m_n \\ & y^m_0=Y_0,\ n=0,...,N-1,\ N:=\max\limits_{i=1,...,I}N_i,\ m=1,...M. \end{array}$$

Note that the SDE in the problem above is the discretized counterpart of

$$dY_{t,\epsilon}(x)=a_\epsilon(x,Y_{t,\epsilon}(x))dt+b_\epsilon(x,Y_{t,\epsilon}(x))dW_t. \tag{3.9}$$

If a and b are already twice continuously differentiable, so that smoothing is not necessary, it will still be referred to a_ϵ and b_ϵ for the sake of readability. In this case, the smoothing parameter is zero. Consequently the second assumption is stated as follows:

$$(A.3) \qquad \begin{array}{l} \pi_\epsilon:\mathbb{R}\to\mathbb{R},\ C^i_{\mathrm{M},\Delta t,\epsilon}:X\to\mathbb{R},\ i=1,...,I,\ a_\epsilon:X\times\mathbb{R}^L\to\mathbb{R}^L \text{ and} \\ b_\epsilon:X\times\mathbb{R}^L\to\mathbb{R}^L\times\mathbb{R}^L \text{ are twice continuously differentiable on} \\ \mathbb{R},\ X,\ X\times\mathbb{R}^L, \text{ respectively.} \end{array}$$

This smoothing may have a welcome side effect regarding the existence and uniqueness of solutions of the SDE. On the other hand, it affects the convergence behavior of the objective function, because there is a third error besides the Monte-Carlo and the discretization error, namely the smoothing error. Both observations

are being examined in chapter 4, whereas the next section finally deals with the numerical solution of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$.

3.4 Sample Average Approximation

The literature provides several ways to deal with the minimization of a function like $f_{M,\Delta t,\epsilon}(x)$ from $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$, e.g. *stochastic approximation* or *sample average approximation* (SAA). For reasons already explained in section 1.1 and section 3.3 it is desirable to make use of fast deterministic optimization methods. In this manner the idea of SAA, which is sometimes also called sample-path optimization (e.g. Robinson [1996]), is to fix the random vector during optimization, such that $f_{M,\Delta t,\epsilon}(x)$ becomes a deterministic function. In addition, since the non-differentiabilities of problem $(\mathrm{P}_{\mathrm{M},\Delta t})$ have been smoothed out, the use of fast converging gradient based optimization algorithms is possible. Reconsider that if the smoothing step would not have been taken, one would have to apply methods of non-differentiable optimization, which are for instance based on subgradients (e.g. Geiger and Kanzow [2002], chapter 6). Note again, that these methods converge very slowly.

Before solving $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ one should recall that it is a nonlinear least squares problem with a special structure, which will be explained in the following. Defining

$$R(x) = [R_i(x)]_{i=1}^{I} := \left[C^i_{M,\Delta t,\epsilon}(x) - C^i_{\mathrm{obs}}\right]_{i=1}^{I} \tag{3.11}$$

the objective function of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ can be written as the squared 2-norm of this residual vector $R(x) \in \mathbb{R}^I$, that is $f_{M,\Delta t,\epsilon}(x) = \|R(x)\|_2^2$. An efficient way to calculate the gradient and even the Hessian is shown in the following Lemma.

Lemma 3.4. *Let Assumption* (A.3) *hold and consider* $f_{M,\Delta t,\epsilon}(x) = \|R(x)\|_2^2$ *with* R *defined in* (3.11) *and let* $J_R : \mathbb{R}^P \to \mathbb{R}^{I\times P}$ *with* $J_R(x) := [\frac{\partial}{\partial x_p} R_i(x)]_{i,p=1}^{I,P}$ *denote the Jacobian of* R*. Then the gradient is defined as*

$$\nabla f_{M,\Delta t,\epsilon}(x) = 2J_R(x)^T R(x)$$

and the Hessian can be approximated through

$$\nabla^2 f_{M,\Delta t,\epsilon}(x) \approx 2J_R(x)^T J_R(x).$$

Proof. The first equation holds by definition. As the functions $C^i_{M,\Delta t,\epsilon}(x)$ are twice continuously differentiable (see assumption (A.3)) the exact formula for the Hessian is

$$\nabla^2 f_{M,\Delta t,\epsilon}(x) = 2J_R(x)^T J_R(x) + 2\sum_{i=1}^{I} R_i(x)\nabla^2 R_i(x).$$

If the residuals $R_i(x)$ are small, that is the model fits the market data well, the so called Gauss-Newton approximation

$$\nabla^2 f_{M,\Delta t,\epsilon}(x) \approx 2J_R(x)^T J_R(x), \tag{3.12}$$

which has been derived by leaving out the second term, can be expected to be of good quality. □

In this case one is able to obtain good approximations of the Hessian by only making use of first order derivative information. Chapter 5 deals with the analysis of several ways how to compute this Jacobian.

Based on the computed first and second order derivatives, one can now apply nonlinear optimization algorithms to the solution of the subproblems ($P_{M,\Delta t,\epsilon}$). Gerlich et al. [2006] show that feasibility perturbed sequential quadratic programming methods in combination with a Gauss-Newton approximation (3.12) of the Hessian perform very well for typical calibration problems in finance. However, infeasible sequential quadratic programming codes or interior point methods might also yield a good choice (see for example Forsgren et al. [2002] or Boggs [1995]).

In this work, a *line-search sequential quadratic programming* method has been chosen to solve ($P_{M,\Delta t,\epsilon}$). As the name indicates, it is a combination of a local convergent sequential quadratic programming (SQP) method in combination with a line-search technique to globalize the convergence behavior. Before considering the line-search approach, the SQP method will be briefly introduced. A detailed description is for instance given by Nocedal and Wright [1999] (chapter 18) or Geiger and Kanzow [2002] (chapter 5). Consider the inequality-constrained optimization problem

$$\min_{x\in\mathbb{R}^P} F(x) \quad \text{s.t.} \quad c(x) = 0,\ d(x) \leq 0 \tag{3.13}$$

with $F : \mathbb{R}^P \to \mathbb{R}$, $c : \mathbb{R}^P \to \mathbb{R}^m$ and $d : \mathbb{R}^P \to \mathbb{R}^n$, which has already been introduced in section 2.3. The KKT conditions (see Theorem 2.32) for this problem are

$$\begin{aligned} \nabla_x L(x,\lambda,\mu) &= 0 \\ c(x) &= 0 \\ d(x) &\leq 0 \\ \mu^T d(x) &= 0 \\ \mu &\geq 0, \end{aligned}$$

where $L(x,\lambda,\mu) : \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is the Lagrangian function of F, namely

$$L(x,\lambda,\mu) := F(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \sum_{i=1}^{n} \mu_i d_i(x)$$

with Lagrange multipliers $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^n$. Note that the notation here differs from the one used in section 2.3 in denoting the inequality constraints with d and the corresponding Lagrange multiplier with μ. Consequently, a solution x of (3.13) fulfills in particular the conditions

$$\begin{aligned} \nabla_x F(x) + J_c(x)^T \lambda + J_d^*(x)^T \mu^* &= 0 \\ c(x) &= 0 \\ d^*(x) &= 0. \end{aligned} \tag{3.14}$$

J_c is the Jacobi matrix of the equality constraints c, d^* denotes the active inequality constraints and J_d^* and μ^* the corresponding Jacobi matrix and Lagrange multiplier respectively. Applying Newton's method to solve (3.14), leads to the iteration

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ \mu_{k+1}^* \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ \mu_K^* \end{pmatrix} + \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta \mu_k^* \end{pmatrix}$$

where $(\Delta x_k, \Delta \lambda_k, \Delta \mu_k^*)^T$ is the solution of the linear system of equations

$$\begin{pmatrix} H_k & J_c(x_k)^T & J_d^*(x_k)^T \\ J_c(x_k)^T & 0 & 0 \\ J_d^*(x_k)^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta \mu_k^* \end{pmatrix} = - \begin{pmatrix} \nabla_x F(x) + J_c(x)^T \lambda + J_d^*(x)^T \mu^* \\ c(x) \\ d^*(x) \end{pmatrix}.$$

H_k is an approximation of the Hessian of the Lagrangian function $\nabla^2_{xx} L(x, \lambda, \mu)$. Subtracting $J_c(x)^T \lambda + J_d^*(x)^T \mu^*$ on both sides shows that this equation is equivalent to

$$\begin{pmatrix} H_k & J_c(x_k)^T & J_d^*(x_k)^T \\ J_c(x_k)^T & 0 & 0 \\ J_d^*(x_k)^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \lambda_{k+1} \\ \mu_{k+1}^* \end{pmatrix} = - \begin{pmatrix} \nabla_x F(x) \\ c(x) \\ d^*(x) \end{pmatrix}$$

which in turn are the KKT conditions of the quadratic problem

$$\begin{aligned} \min_{\Delta x_k} \; & \nabla F(x_k)^T \Delta x_k + \tfrac{1}{2} \Delta x_k^T H_k \Delta x_k \\ \text{s.t.} \quad & J_c(x_k)^T \Delta x_k + c(x_k) = 0 \\ & J_d^*(x_k)^T \Delta x_k + d^*(x_k) = 0. \end{aligned}$$

As the set of active inequality constraints is unknown at the very beginning of the optimization, the idea of the SQP algorithm lies in the solution of the corresponding

quadratic problem including all constraints

$$\begin{aligned} & \min_{\Delta x_k} \nabla F(x_k)^T \Delta x_k + \tfrac{1}{2} \Delta x_k^T H_k \Delta x_k \\ \text{s.t.} \quad & J_c(x_k)^T \Delta x_k + c(x_k) = 0 \\ & J_d(x_k)^T \Delta x_k + d(x_k) \leq 0. \end{aligned} \tag{3.15}$$

Algorithm 1 shows the resulting pseudocode of the SQP method.

Algorithm 1 SQP Method

```
Choose (x_0, λ_0, μ_0) ∈ R^p × R^m × R^n
Set k=0
while Convergence is not satisfied do
    Solve the quadratic problem (3.15) and receive Δx_k, λ_{k+1}, μ_{k+1}
    Set x_{k+1} = x_k + Δx_k and k = k + 1
end while
```

Unfortunately this SQP method converges only locally, i.e. only for starting values close enough to a stationary point. To globalize this algorithm two classes of techniques can be utilized, trust-region and line-search based methods. The idea of trust-region is to add an additional constraint of the form $\|W\Delta x\| \leq \Delta$ to the quadratic problem. Δ is the trust-region radius and W a scaling matrix. A more detailed introduction is given for instance in Conn et al. [2000]. The line-search framework has been chosen in this work, where the iterates are calculated via

$$x_{k+1} = x_k + \alpha_k d_k$$

where d_k is a direction in $\mathbb{R}^P$ and α_k the size of the step that is taken in this direction.

The additional problem of ensuring feasibility is solved by adding a penalty term to the objective function. In terms of (3.13), this means that the *merit function*

$$\Theta_\gamma(x) := F(x) + \gamma \left(\|e(x)^\#\| \right)$$

with $e(x) = (c(x), d(x))^T \in \mathbb{R}^{m+n}$ the combined vector of equality and inequality constraints and

$$e_i(x)^\# = \begin{cases} c_i(x) & i = 1, ..., m \\ \max(d_{i-m}(x), 0) & i = m+1, ..., n \end{cases}$$

replaces the original objective $F(x)$. By definition, this function penalizes infeasibility as the values of $\|c(x)\|$ as well as $\|\max(d(x), 0)\|$ increase with increasing degree of infeasibility. Bonnans et al. [2003], p. 295 suggest an adaptive choice of the penalty parameter γ, showed in algorithm 2.

Algorithm 2 Penalty Parameter Update

Choose $\overline{\gamma} > 0$
if $\gamma_{k-1} \geq 1.1(\|\gamma_k\| + \overline{\gamma})$ **then**
    $\gamma_k = \frac{1}{2}(\gamma_{k-1} + \|\lambda_k\| + \overline{\gamma})$
else
    if $\gamma_{k-1} \geq \|\lambda_k\| + \overline{\gamma}$ **then**
        $\gamma_k = \gamma_{k-1}$
    else
        $\gamma_k = \max(1.5\gamma_{k-1}, \|\lambda_k\| + \overline{\gamma}$
    end if
end if

This rule takes into account, that the penalty parameter has to fullfill the condition

$$\gamma_k \geq \|\lambda_k\|$$

to make d_k a descent direction (Bonnans et al. [2003], Proposition 17.1, p. 293). In fact, this condition has to be imposed with some safeguard, i.e.

$$\gamma_k \geq \|\lambda_k\| + \overline{\gamma}$$

for some $\overline{\gamma} > 0$. The constants 1.1 and 1.5 can be replaced by any constant greater 1.

Furthermore, the step size α_k is chosen adaptively to decrease the merit function Θ_γ. In particular, α_k is calculated by the *Armijo* step size rule, defined in algorithm 3. The interpretation of this step size choice is as follows. If the initially chosen

Algorithm 3 Armijo

The iterates x_k and direction d_k are given
Choose $\alpha_{\max} > 0$ and $\beta, \xi \in (0, 1)$
if $\Theta_\gamma(x_k + \alpha_{\max} d_k) - \Theta_\gamma(x_k) \leq \xi \alpha_{\max} \nabla\Theta_\gamma(x_k)^T d_k$ **then**
    $\alpha_k = \alpha_{max}$
else
    Set $l_k = 1$
    while $\Theta_\gamma(x_k + \alpha_{\max}\beta^{l_k} d_k) - \Theta_\gamma(x_k) > \xi \alpha_{\max}\beta^{l_k} \nabla\Theta_\gamma(x_k)^T d_k$ **do**
        $l_k = l_k + 1$
    end while
    $\alpha_k = \alpha_{\max}\beta^{l_k}$
end if

step $x_k + \alpha_{\max} d_k$ decreases Θ_γ sufficiently, this step is being taken. If not, the initial step size $\alpha_{\max}$ is decreased by the factor β^{l_k} until a sufficient decrease of the merit function has been reached. The sufficiency is tested via the so called Armijo condition

$$\Theta_\gamma(x_k + \alpha_{\max} d_k) - \Theta_\gamma(x_k) \leq \xi \alpha_{\max} \nabla\Theta_\gamma(x_k)^T d_k.$$

All together, the line-search SQP algorithm is defined in algorithm 4.

Algorithm 4 Line-Search SQP Method

1: Choose $(x_0, \lambda_0, \mu_0) \in \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^n$, $\alpha_{\max} > 0$ and $\beta, \xi \in (0,1)$ for Armijo and $\overline{\gamma} > 0$ for the penalty update
2: Calculate $\nabla F(x_0)$, $J_c(x_0)$, $J_d(x_0)$ and $H_0 \approx \nabla^2_{xx} L(x_0, \lambda_0, \mu_0)$
3: Set k=0
4: **while** Convergence is not satisfied **do**
5: Solve (3.15) and receive $(\Delta x_k, \lambda_{k+1}, \mu_{k+1})$
6: Adapt γ_k with algorithm 2
7: Choose α_k with algorithm 3
8: Set $x_{k+1} = x_k + \alpha_k \Delta x_k$ and $k = k + 1$
9: Calculate $\nabla F(x_k)$, $J_c(x_k)$, $J_d(x_k)$
10: **end while**

In any case the main effort of the algorithm will be the evaluation of the objective function of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ and its gradient since any evaluation of $f_{M,\Delta t,\epsilon}$ requires to perform M numerical solutions of the stochastic differential equations in $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$. In this context note that it is sufficient for the evaluation of the objective to simulate the SDEs once until $t = T$ and to pick the stock prices at the maturities T_i (see figure 3.2), instead of simulating them again for each call option C^i. As a consequence, all option prices — for different maturities and strikes — can be calculated in one sweep from one path. The effect of this technique can be estimated by $\frac{1}{2} T_I (T_I + 1)$, which is a factor of 15 for maturities between 0 and 5 years for example.

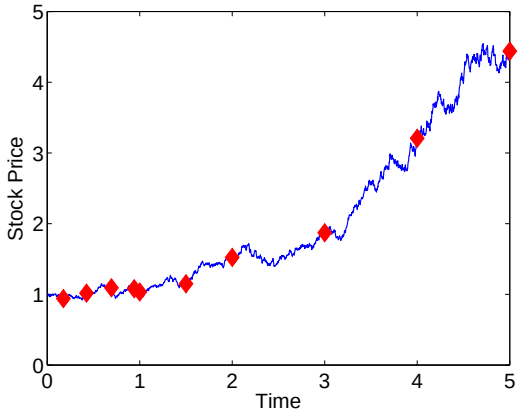

Figure 3.2: Graphical illustration of one simulated stock price path (blue line) and those prices (red diamonds) which can be picked along the path to evaluate the functions $C^i_{M,\Delta t,\epsilon}$.

Chapter 4

Convergence of the Approximating Problem

In the previous chapter the discretized optimization problem ($P_{M,\Delta t,\epsilon}$) has been derived by approximating the expectation functionals via Monte Carlo, discretizing the SDEs with the Euler-Maruyama scheme and smoothing out the non-differentiabilities of the objective function. This results in three sources of errors, namely the Monte Carlo error, the time discretization error and the smoothing error. Though it follows from intuition, that a solution of ($P_{M,\Delta t,\epsilon}$) is an approximation of (P), this statement does not hold for arbitrary problems in mathematical theory.

Consequently, this chapter analyzes the convergence behavior of a solution of ($P_{M,\Delta t,\epsilon}$) towards a solution of (P) and is structured as follows. The first part deals with the uniqueness of solutions of the stochastic differential equation under various assumptions. The second part contains a pointwise convergence analysis in a simplified optimization problem framework. On the basis of this, the uniform convergence of a solution of ($P_{M,\Delta t,\epsilon}$) to a solution of (P) can be shown, which finally allows to prove first order optimality.

4.1 Uniqueness of Solutions to Stochastic Differential Equations

In section 3.1 the stochastic differential equation (3.1) on page 23

$$dY_t(x) = a(x, Y_t(x))dt + b(x, Y_t(x))dW_t\,,\ Y_0 \in (0,\infty)\,,\ 0 \le t \le T$$

has been introduced. As noted very briefly in section 3.1, a and b have to fulfill conditions, that a solution of (3.1) exists. The pure existence has been addressed

in Theorem 2.8 under relatively nonrestrictive assumptions like the path continuity. But like in the theory of ordinary differential equations, the uniqueness of such a solution is desirable. Consider therefore the following definition:

Definition 4.1. *Let* $(Y_t)_{t\in[0,T]}$ *be a solution of* (3.1). *If for every second solution* $(\tilde{Y}_t)_{t\in[0,T]}$*:*

$$P\left(\sup_{0\leq t\leq T}\left\|\tilde{Y}_t - Y_t\right\| > 0\right) = 0,$$

Y_t *is a* pathwise unique solution.

Unfortunately the assumptions of Theorem 2.8 do not seem to allow for a uniqueness proof. Kloeden and Platen [1999] showed the existence (Theorem 4.5.3, pp. 131 ff.) of a pathwise unique solution under heavier assumptions, like the Lipschitz continuity of the coefficient functions. This is being addressed in section 4.1.1. However, there exist models with coefficient functions which do not fulfill a Lipschitz condition. Yamada and Watanabe [1971] provide a uniqueness proof under relaxed conditions (section 4.1.2). These conditions cover for instance, the case with Lipschitz continuous drift and Hölder continuous diffusion. Unfortunately, this result is restricted to the case of an indeed multidimensional model but with autonomous components. In this manner, a uniqueness result can be derived from the weak convergence proof of Mikulevicius and Platen [1991] in section 4.1.3.

As a first step, models with drift and diffusion that provide uniformly bounded Lipschitz constants are observed.

4.1.1 Lipschitz Continuous Coefficients

The crucial assumption for the standard existence and uniqueness result is the Lipschitz continuity of the coefficient functions. Though not every model has Lipschitz continuous coefficients, there is a wide variety of models that do fulfill this assumption, e.g. the models of Stein and Stein [1991] or Vasicek [1977]. Furthermore, the smoothed version of a square-root process, which has been introduced above, obtains Lipschitz continuous coefficients for a positive smoothing parameter $\epsilon > 0$, due to the fact that the smoothing polynomial keeps the process away from reaching zero. Figure 4.1 displays the Lipschitz property of the function $f_\epsilon(x) = \sqrt{\pi_\epsilon(x)}$ with π_ϵ from (3.7). A simple but tedious calculation shows, that f_ϵ has a Lipschitz constant of order $\mathcal{L}(\epsilon) = \mathcal{O}(\epsilon^{-\frac{1}{2}})$. Consequently Kloeden and Platen's theorem for the existence of a pathwise unique solution assuming the Lipschitz property is stated below.

Theorem 4.2. *Let* $(Y_t)_{t\in[0,T]}$ *be a solution of*

$$dY_t = a(x, Y_t)dt + b(x, Y_t)dW_t\,,\ Y_0 \in (0,\infty). \tag{4.1}$$

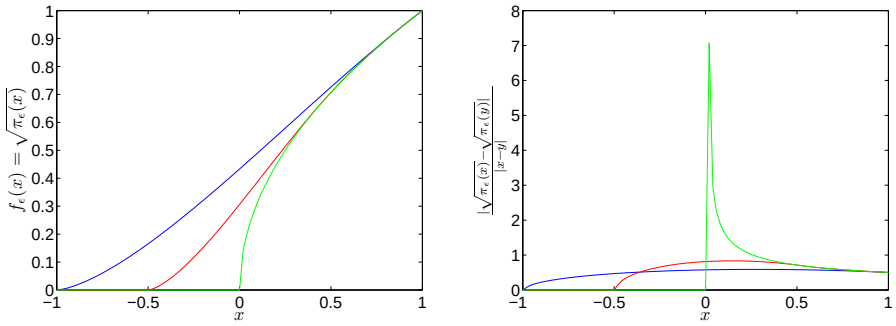

Figure 4.1: $f_\epsilon(x) = \sqrt{\pi_\epsilon(x)}$ for $\epsilon = 1.0$ (blue line) , $\epsilon = 0.5$ (red line) and $\epsilon = 0.0$ (green line) on the left and corresponding upper bound $\frac{|\sqrt{\pi_\epsilon(x)} - \sqrt{\pi_\epsilon(x+h)}|}{h}$ on the right.

If the Lipschitz condition:

There exists a constant $\mathcal{L} > 0$ *such that* $\forall t \in [0, T]$ *and* $y, \overline{y} \in \mathbb{R}^L$

$$\|a(x, y) - a(x, \overline{y})\| + \|b(x, y) - b(x, \overline{y})\| \leq \mathcal{L}\|y - \overline{y}\|$$

holds, Y_t *on* $[0, T]$ *is a pathwise unique solution.*

As the Lipschitz constant may depend on the smoothing parameter ϵ, the proof provided by Kloeden and Platen [1999], pp. 131 ff., is modified and presented to clarify this dependence.

Proof. Let $(Y_t)_{t\in[0,T]}$ and $(\tilde{Y}_t)_{t\in[0,T]}$ be two solutions of (4.1), i.e.

$$\begin{aligned} Y_t &= Y_0 + \int_0^T a(x, Y_t)dt + \int_0^T b(x, Y_t)dW_t \\ \tilde{Y}_t &= Y_0 + \int_0^T a(t, \tilde{Y}_t)dt + \int_0^T b(t, \tilde{Y}_t)dW_t. \end{aligned}$$

The goal is to show that $E(\|\tilde{Y}_t - Y_t\|^2) = 0$. As it yet may occur that the second moments are not finite the following truncation procedure will be used:

$$I_N(t) = \begin{cases} 1 & ; \quad \|Y_s(\omega)\|, \|\tilde{Y}_s(\omega)\| \leq N \text{ for } 0 \leq s \leq t \\ 0 & ; \quad \text{otherwise.} \end{cases}$$

It holds by definition that

$$\begin{aligned} & E\left(\left|I_N(t)\|\tilde{Y}_t - Y_t\|\right|^2\right) \\ = \; & E\left(\left|I_N(t)\int_0^t I_N(s)\|a(x,\tilde{Y}_s) - a(x,Y_s)\|ds\right.\right. \\ + \; & \left.\left. I_N(t)\int_0^t I_N(s)\|b(x,\tilde{Y}_s) - b(x,Y_s)\|dW_s\right|^2\right). \end{aligned}$$

Making use of the fact that $(a+b)^2 \leq 2(a^2+b^2)$ and the Ito isometry (Lemma 2.5 (i)) it follows

$$\begin{aligned} & E\left(\left|I_N(t)\|\tilde{Y}_t - Y_t\|\right|^2\right) \\ \leq \; & 2E\left(\left|I_N(t)(\int_0^t I_N(s)\|a(x,\tilde{Y}_s) - a(x,Y_s)\|ds\right|^2\right) \\ + \; & 2E\left(\left|I_N(t)(\int_0^t I_N(s)\|b(x,\tilde{Y}_s) - b(x,Y_s)\|dW_s\right|^2\right) \\ \leq \; & 2\int_0^t E\left(\left|I_N(s)\|a(x,\tilde{Y}_s) - a(x,Y_s)\|\right|^2\right)ds \\ + \; & 2\int_0^t E\left(\left|I_N(s)\|b(x,\tilde{Y}_s) - b(x,Y_s)\|\right|^2\right)ds. \end{aligned}$$

Now the Lipschitz continuity provides

$$E\left(\left|I_N(t)\|\tilde{Y}_t - Y_t\|\right|^2\right) \leq 4\mathcal{L}^2\int_0^t E\left(\left|I_N(s)\|\tilde{Y}_s - Y_s\|\right|^2\right)ds. \tag{4.2}$$

Finally an application of the Gronwall inequality (Lemma 2.10) with $L = 4\mathcal{L}^2$, $\alpha(t) = E(|I_N(t)\|\tilde{Y}_t - Y_t\||^2)$ and $\beta(t) = 0$ leads to

$$E\left(\left|I_N(t)\|\tilde{Y}_t - Y_t\|\right|^2\right) = 0.$$

This means that $I_N(t)\|Y_t\| = I_N(t)\|\tilde{Y}_t\|$ (a.s.) for each $t \in [0,T]$ due to Jensen's inequality (Theorem 2.11). As Lemma 2.5 (ii) provides the continuity of the sample paths, they are bounded almost surely. Thus, choosing N sufficiently large provides

$P(Y_t \neq \tilde{Y}_t) = 0$ for each $t \in [0, T]$ and consequently

$$P(\{t \in \mathcal{D}; Y_t \neq \tilde{Y}_t\}) = 0$$

where $\mathcal{D}$ is a countable and dense subset in $[0, T]$. As the solutions are continuous and coincide on any countable and dense subset of $[0, T]$ they must coincide almost surely on $[0, T]$. □

Consequently, if one considers for instance the Heston model, where the coefficients are Lipschitz continuous for $\epsilon > 0$, this uniqueness result certainly suffices for a fixed smoothing parameter. Unfortunately, this does not hold true for the unsmoothed version, and thus not in the limit $\epsilon \to 0$. So other results would be desired, if one would like to have uniqueness for instance in the context of the convergence analysis in chapter 4.

4.1.2 Uniqueness under Yamada's Condition

Yamada and Watanabe [1971] presented an alternative uniqueness result for a SDE solution. This result differs from the one presented above in the sense that the Lipschitz assumption could be relaxed. The new assumption is called the *Yamada condition*.

Definition 4.3. (Yamada Condition) *Let $(W_t)_{t\in[0,T]} = (W_t^1, ..., W_t^L)_{t\in[0,T]}$ be a L-dimensional Brownian motion and $(Y_t)_{t\in[0,T]}$ the solution of the L-dimensional system of SDEs*

$$dY_t = a(x, Y_t)dt + b(x, Y_t)dW_t \tag{4.3}$$

with $a : X \times \mathbb{R}^L \to \mathbb{R}^L$ and $b : X \times \mathbb{R}^L \to \mathbb{R}^L \times \mathbb{R}^L$ such that $a(x, Y_t) = (a^1(x, Y_t^1), ..., a^L(x, Y_t^L))^T$ as well as

$$\begin{aligned} b(x, Y_t) &= \operatorname{diag}(b^1(x, Y_t^1), ..., b^L(x, Y_t^L)) \\ &= \begin{pmatrix} b^1(x, Y_t^1) & 0 & 0 & ... & 0 \\ 0 & b^2(x, Y_t^2) & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & ... & b^L(x, Y_t^L) \end{pmatrix}. \end{aligned}$$

If there exists a positive increasing function $\beta : [0, \infty) \to [0, \infty)$ with

$$|b^i(x, y) - b^i(x, \overline{y})| \leq \beta(|y - \overline{y}|) \quad \forall y, \overline{y} \in \mathbb{R}, \quad i = 1, ..., L$$

and

$$\int_0^\delta \beta^{-2}(z)dz = \infty$$

with an arbitrarily small $\delta > 0$*, and a positive increasing concave function* $\alpha : [0,\infty) \to [0,\infty)$ *such that*

$$|a^i(x,y) - a^i(x,\overline{y})| \leq \alpha(|y-\overline{y}|) \quad \forall y,\overline{y} \in \mathbb{R}, \quad i = 1,...,L$$

with

$$\int_0^\delta \alpha^{-1}(z)dz = \infty$$

with an arbitrarily small $\delta > 0$*, the SDE* (4.3) *is said to fulfill the* Yamada Condition.

With the help of this condition Yamada and Watanabe could proof the following uniqueness result.

Theorem 4.4. *In the situation of Definition 4.3 the pathwise uniqueness holds for the solution of the stochastic differential equation* (4.3)*.*

Proof. Yamada and Watanabe [1971] pp. 164 ff. □

Note in this context that the mapping $f_\epsilon(x) = \sqrt{\pi_\epsilon(x)}$ is globally Hölder continuous with factor $\frac{1}{2}$ for all $\epsilon \geq 0$:

Remark 4.5. *The maximum function and the smoothed maximum function are obviously Lipschitz continuous, whereas the Lipschitz constant of the plain maximum function deals as an upper bound for the smoothed maximum function (see also figure 3.1). In particular, the Lipschitz constant is 1. Thus it holds for* $x, y \in \mathbb{R}$

$$|\pi_\epsilon(x) - \pi_\epsilon(y)| \leq |x-y|, \quad \forall \epsilon \geq 0.$$

Moreover, the square root function is Hölder continuous with factor $\frac{1}{2}$*, i.e.*

$$\left(\sqrt{x}-\sqrt{y}\right)^2 = \left|\sqrt{x}-\sqrt{y}\right|\left|\sqrt{x}-\sqrt{y}\right| \leq \left|\sqrt{x}-\sqrt{y}\right|\left|\sqrt{x}+\sqrt{y}\right| = |x-y|$$

and thus $|\sqrt{x}-\sqrt{y}| \leq \sqrt{|x-y|}$*. Summarizing, the mapping* $f(x) = \sqrt{\pi_\epsilon(x)}$ *is Hölder continuous with factor* $\frac{1}{2}$*, as it holds*

$$|\sqrt{\pi_\epsilon(x)} - \sqrt{\pi_\epsilon(y)}| \leq \sqrt{|\pi_\epsilon(x) - \pi_\epsilon(y)|} \leq \sqrt{|x-y|}, \quad \forall \epsilon > 0.$$

Consequently the square root fulfills the assumption for the function β and linear functions the assumption for α. Thus, a model following the dynamics of (4.3) with a Hölder continuous diffusion and a Lipschitz continuous drift possesses

a pathwise unique solution. The special square root case of the constant elasticity of variance-model (Cox [1996])

$$dS_t = (r - \delta)S_t dt + \sigma\sqrt{S_t}dW_t$$

is an example for such a model. Nevertheless, the assumption of L uncoupled SDEs is very restrictive. Multidimensional financial market models often obtain quite naturally a dependence of the single SDE components, as it can be observed for instance in the case of variance (e.g. Heston (3.2)) or interest rate processes (e.g. Vasicek [1977]).

As a further alternative, a uniqueness result can be derived from the weak convergence proof of the Euler scheme by Mikulevicius and Platen [1991].

4.1.3 Uniqueness by Mikulevicius and Platen

Mikulevicius and Platen [1991] proved a weak convergence result for the Euler-Maruyama scheme applied to SDEs of the form (3.1). A similar proof allows a uniqueness result, as it will be shown subsequently. First of all consider the following definition:

Definition 4.6. *Let $l \in (0,1) \cup (1,2) \cup (2,3)$ and $\mathcal{H}_T^l$ the space of continuous functions u on $[0,T] \times \mathbb{R}^L$ posessing continuous derivatives $\frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}$ for all $2r + s < l$ such that*

$$\begin{aligned}\|u\|_T^l \quad := \quad & \sum_{2r+s\leq\lfloor l\rfloor} \sup_{(v,x)\in[0,T]\times\mathbb{R}^L} \left|\frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}u(v,x)\right| \\ + \quad & \sum_{2r+s=\lfloor l\rfloor} \sup_{(v,x),(v,x')\in[0,T]\times\mathbb{R}^L} \frac{\left|\frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}u(v,x) - \frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}u(v,x')\right|}{|x-x'|^{l-\lfloor l\rfloor}} \\ + \quad & \sum_{0<l-2r-s<2} \sup_{(v,x),(v',x)\in[0,T]\times\mathbb{R}^L} \frac{\left|\frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}u(v,x) - \frac{\partial^r}{\partial t}\frac{\partial^s}{\partial x}u(v',x)\right|}{|v-v'|^{\frac{1}{2}(l-2r-s)}}.\end{aligned}$$

$\mathcal{H}^l$ denotes the corresponding space for functions that are time independent and $\|u\|^l$ the corresponding norm.

Now, with the help of the sets $\mathcal{H}_T^l$ and $\mathcal{H}^l$ the uniqueness result can be stated.

Theorem 4.7. *Let $(Y_t)_{t\in[0,T]}$ a solution of*

$$dY_t = a(x, Y_t)dt + b(x, Y_t)dW_t\,,\ Y_0 \in (0,\infty) \tag{4.4}$$

and $B(x,y) := b(x,y)b(x,y)^T$. *If*

$$\langle B(x,y)\eta,\eta\rangle \geq c|\eta|^2,\ c>0,\ \forall \eta, y \in \mathbb{R}^L \tag{4.5}$$

$$a, b \in \mathcal{H}_T^l \text{ for } l \in (0,1)\cup(1,2)\cup(2,3) \tag{4.6}$$

$$g \in \mathcal{H}^l \text{ for } l \in (0,1)\cup(1,2)\cup(2,3) \tag{4.7}$$

$(Y_t)_{t\in[0,T]}$ *is a pathwise unique solution.*

Proof. Let $(Y_t)_{t\in[0,T]}$ and $(\tilde{Y}_t)_{t\in[0,T]}$ be two solutions of (4.4), i.e.

$$\begin{aligned} Y_t &= Y_0 + \int_0^T a(x, Y_t)dt + \int_0^T b(x, Y_t)dW_t \\ \tilde{Y}_t &= Y_0 + \int_0^T a(x, \tilde{Y}_t)dt + \int_0^T b(x, \tilde{Y}_t)dW_t \end{aligned}$$

and $\mathcal{D}$ the diffusion operator

$$\mathcal{D} := \sum_{i=1}^{L} a_i \frac{\partial}{\partial y_i} + \frac{1}{2}\sum_{i,j=1}^{L} B_{ij}\frac{\partial^2}{\partial y_i y_j^2}.$$

Due to assumptions (4.5)-(4.7) It follows from Ladyzenskaja et al. [1968] (Theorem 5.2, p. 320) that there exists a unique solution $v \in \mathcal{H}_T^{l+2}$ of the parabolic partial differential equation

$$\frac{\partial}{\partial t}v + \mathcal{D}v = 0 \tag{4.8}$$

with final condition

$$v(T, y) = g(y) \tag{4.9}$$

and

$$\|v\|_T^{l+2} \leq K|g|^{l+2}.$$

An application of the Ito formula (Theorem 2.9) provides

$$\begin{aligned} v(t, Y_t) = v(0, Y_0) + \int_0^t \tfrac{\partial}{\partial s}v(s, Y_s) + \sum_{i=1}^{L} a_i(x, Y_s)\tfrac{\partial}{\partial y_i}v(s, Y_s) \\ +\tfrac{1}{2}\sum_{i,j=1}^{L} B_{ij}(x, Y_s)\tfrac{\partial^2}{\partial y_i y_i^2}v(s, Y_s)ds + \int_0^t \sum_{i=1}^{L} b_i(x, Y_s)\tfrac{\partial}{\partial y_i}v(s, Y_s)dW_s \end{aligned}$$

and the same for $\tilde{Y}_t$

$$v(t, \tilde{Y}_t) = v(0, Y_0) + \int_0^t \frac{\partial}{\partial s} v(s, \tilde{Y}_s) + \sum_{i=1}^{L} a_i(x, \tilde{Y}_s) \frac{\partial}{\partial y_i} v(s, \tilde{Y}_s)$$
$$+ \frac{1}{2} \sum_{i,j=1}^{L} B_{ij}(x, \tilde{Y}_s) \frac{\partial^2}{\partial y_i y_i^2} v(s, \tilde{Y}_s) ds + \int_0^t \sum_{i=1}^{L} b_i(x, \tilde{Y}_s) \frac{\partial}{\partial y_i} v(s, \tilde{Y}_s) dW_s.$$

Inserting this result in $E(|v(t, Y_t) - v(t, \tilde{Y}_t)|)$ in combination with an application of the triangle inequality provides

$$E\left(\left|v(t, Y_t) - v(t, \tilde{Y}_t)\right|\right) \leq$$
$$E\left(\left|\int_0^t \frac{\partial}{\partial s} v(s, Y_s) + \mathcal{D}v(s, Y_s) ds - \int_0^t \frac{\partial}{\partial s} v(s, \tilde{Y}_s) + \mathcal{D}v(s, \tilde{Y}_s) ds\right|\right)$$
$$+ E\left(\left|\int_0^t \sum_{i=1}^{L} \left(b_i(x, Y_s) \frac{\partial}{\partial y_i} v(s, Y_s) - b_i(x, \tilde{Y}_s) \frac{\partial}{\partial y_i} v(s, \tilde{Y}_s)\right) dW_s\right|\right).$$

As the first term on the right side is equal to zero because of (4.8) and the expected value of an Ito integral is equal to zero one receives

$$E\left(\left|v(t, Y_t) - v(t, \tilde{Y}_t)\right|\right) = 0.$$

The final condition (4.9) provides

$$E\left(\left|g(Y_T) - g(\tilde{Y}_T)\right|\right) = E\left(\left|v(T, Y_T) - v(T, \tilde{Y}_T)\right|\right) = 0.$$

As one is free to choose $g(x) = x$ and T has been chosen arbitrarily, this means that $Y_t = \tilde{Y}_t$ (a.s.) for each $t \in [0, T]$. Thus it holds

$$P(\{t \in \mathcal{D}; Y_t \neq \tilde{Y}_t\}) = 0$$

where $\mathcal{D}$ is a countable and dense subset in $[0, T]$. As the solutions are continuous (Lemma 2.5 (ii)) and coincide on any countable and dense subset of $[0, T]$ they must coincide almost surely on $[0, T]$. □

Note that this theorem contains two critical assumptions, i.e. (4.5) and (4.6). Considering Heston's model (3.2), (4.5) requires that

$$\left\langle b(x, y) b(x, y)^T \eta, \eta \right\rangle \geq c|\eta|^2,\ c > 0,\ \forall \eta, y \in \mathbb{R}^L.$$

The possibly smoothed version of Heston's model yields

$$b_\epsilon(x, y) = \begin{pmatrix} y_1 \sqrt{\pi_\epsilon(y_2)} & 0 \\ \sigma\rho\sqrt{\pi_\epsilon(y_2)} & \sigma\sqrt{1-\rho^2}\sqrt{\pi_\epsilon(y_2)} \end{pmatrix}$$

and thus

$$b_\epsilon(x,y)b_\epsilon(x,y)^T = \begin{pmatrix} y_1^2\pi_\epsilon(y_2) & \sigma\rho y_1\pi_\epsilon(y_2) \\ \sigma\rho y_1\pi_\epsilon(y_2) & \sigma^2\pi_\epsilon(y_2) \end{pmatrix}.$$

The left side of (4.5) is defined as the quadratic form of $b_\epsilon(x,y)^T b_\epsilon(x,y)$ which is by definition equal to the quadratic form of $b_\epsilon(x,y)b_\epsilon(x,y)^T$.

However for the determinant of this matrix it holds that

$$det(b_\epsilon(x,y)b_\epsilon(x,y)^T) = \sigma^2\sqrt{1-\rho^2}y_1^2\pi_\epsilon(y_2)^2$$

which is not strictly positive for all $y \in \mathbb{R}^2$ and thus (4.5) is not arbitrarily fulfilled. Nevertheless this theorem implies an alternative uniqueness results in comparison to section 4.1.1 and 4.1.2.

4.2 Convergence to a Stationary Point of the True Problem

The discretization of the true optimization problem (P) raises the question if a solution of the resulting discretized problem ($P_{M,\Delta t,\epsilon}$) is an approximation of a solution of (P) for appropriately chosen number of Monte Carlo Simulations, discretization step size and smoothing parameter. To be more precise, if $x_k \in X$ is a solution derived by solving ($P_{M,\Delta t,\epsilon}$) with the triplet $(M_k, \Delta t_k, \epsilon_k) \in \mathbb{N} \times \mathbb{R}_+ \times \mathbb{R}_+$, the so obtained sequence $(x_k)_k$ has a subsequence which converges to a limit point $x^* \in X$ for $M_k \uparrow \infty, \Delta t_k \downarrow 0$ and $\epsilon_k \downarrow 0$, due to the fact that X is compact. It is desirable, that this limit point x^* is a solution of problem (P). The following example shows, that this unfortunately does not hold in general.

Example 4.8. *Consider for instance the minimization of $f(x) = x^2$ over $[-1;1]$ (see figure 4.2). f attains its global minimum at $x^* = 0$. By contrast, the approximating objective $f_M(x) = x^2 - 2M^{-1}\sin(Mx^2)$ possesses many local minima. The number even increases with increasing parameter M, as it is shown in figure 4.2. Consequently, minimization with increasing M may lead to a sequence that does not converge to x^*, though a uniform convergence of the objective functions, namely $\sup_{x\in\mathbb{R}} |f_M(x) - f(x)| \leq M^{-1}$, can be observed. It is shown later, that not only the uniform convergence of the objectives but also of the gradients is a crucial assumption. This assumption is violated here, as $\sup_{x\in\mathbb{R}} |\nabla f_M(x) - \nabla f(x)| = \sup_{x\in\mathbb{R}} |4\cos(Mx^2)| = 4$. Furthermore, this example emphasizes, that solving ($P_{M,\Delta t,\epsilon}$) for fixed M, Δt and ϵ can lead to local minimizers that are not close to local minimizers of* (P). *An application of e.g.* fminsearch *in MatLab to find a minimum of f_{10} starting at $x_0 = -1$ finds a solution at $x_{10}^* = -0.8562$.*

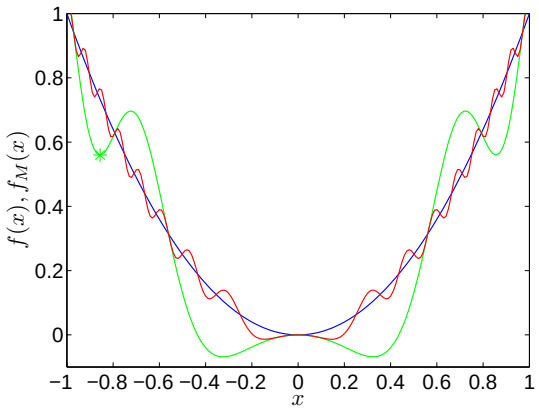

Figure 4.2: $f(x) = x^2$ (blue line) , $f_M(x) = x^2 - 2M^{-1}\sin(Mx^2)$ for $M = 10$ (green line) and $M = 50$ (red line) and minimum of $f_{10}(x)$ (green star) found by *fminsearch* in MatLab.

Monte-Carlo techniques are well-known tools and have been applied in many different fields of applications. Consequently the literature provides a large amount of results on the convergence of a solution of a Monte Carlo based optimization problem to a solution of the corresponding expected value problem. Considering for instance

$$\min_{x\in X}\{h(x) := E(H(x,\omega))\}$$

and the corresponding SAA problem

$$\min_{x\in X}\{h_{\mathrm{M}}(x) := \frac{1}{M}\sum_{m=1}^{M} h(x,\omega_m)\}$$

Rubinstein and Shapiro [1993] show that h_{M} converges uniformly to h on X under the assumption that $h(\cdot,\omega)$ is almost surely dominated integrable (see also Definition 4.15 below) and continuous on X. Based on this they were able to demonstrate the convergence of the sequence of solution of h_{M} to a solution of h in the sense of a first order critical point. Shapiro [2000] proves convergence under the assumption that the optimization problem produces a global minimum. The case of an optimization problem that produces a complete set of solutions has been examined by Robinson [1996]. In comparison to these results, Bastin et al. [2006] additionally considers second order optimality conditions and stochastic constraints.

However, the approximation of (P) via ($\mathrm{P}_{\mathrm{M},\Delta t,\epsilon}$) in this work depends on three errors, Monte-Carlo, discretization and smoothing error. It will turn out that a uniform convergence of the objective functions and corresponding gradients of (P) and ($\mathrm{P}_{\mathrm{M},\Delta t,\epsilon}$) with respect to these three errors allows to prove optimality conditions. Thus, the subsequent analysis is structured as follows. Section 4.2.1 deals with the

analysis under which assumptions a pointwise convergence of the approximating to the original objective function holds. Secondly, the uniform convergence of solutions of (P) and ($P_{M,\Delta t,\epsilon}$) can be shown, where this pointwise convergence is one of the keypoints. Section 4.2.3 then addresses first order optimality.

Before the pointwise convergence and uniform convergence analysis can be stated, this is examined for a simplified problem compared to (P) (page 25), namely

$$\begin{aligned} & \min_{x\in X} g(x) := E\left(G(x,\omega)\right) \\ \text{where} \quad & G(x,\omega) = \max(S_T(x,\omega) - K, 0) \\ \text{s.t.} \quad & dY_t(x,\omega) = a(x, Y_t(x,\omega))dt + b(x, Y_t(x,\omega))dW_t(\omega). \end{aligned} \tag{P^1}$$

This problem corresponds to (P) in the sense that the market prices and thus the least squares differences have been skipped. Additionally the objective now only contains one call price. As it will be usefull, the dependence of the random variables on the random vector ω is denoted explicitly here. Applying the smoothing technique to (P^1) leads to

$$\begin{aligned} & \min_{x\in X} g_\epsilon(x) := E\left(G_\epsilon(x,\omega)\right) \\ \text{where} \quad & G_\epsilon(x,\omega) = \pi_\epsilon(S_{T,\epsilon}(x,\omega) - K) \\ \text{s.t.} \quad & dY_{t,\epsilon}(x,\omega) = a_\epsilon(x, Y_{t,\epsilon}(x,\omega))dt + b_\epsilon(x, Y_{t,\epsilon}(x,\omega))dW_t(\omega). \end{aligned} \tag{P^1_ϵ}$$

If one now discretizes the stochastic differential equation with the Euler-Maruyama scheme, one receives

$$\begin{aligned} & \min_{x\in X} g_{\Delta t,\epsilon}(x) := E\left(G_{\Delta t,\epsilon}(x,\omega)\right) \\ \text{where} \quad & G_{\Delta t,\epsilon}(x,\omega) = \pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) \\ \text{s.t.} \quad & y_{n+1,\epsilon}(x,\omega) = y_{n,\epsilon}(x,\omega) + a_\epsilon(x, y_{n,\epsilon}(x,\omega))\Delta t_n \\ & \qquad + b_\epsilon(x, y_{n,\epsilon}(x,\omega))\Delta W_n(\omega). \end{aligned} \tag{$P^1_{\Delta t,\epsilon}$}$$

An additional approximation of the expected value function with Monte Carlo finally provides

$$\begin{aligned} & \min_{x\in X} g_{M,\Delta t,\epsilon}(x) := \tfrac{1}{M}\sum_{m=1}^{M}\left(G_{\Delta t,\epsilon}(x,\omega_m)\right) \\ \text{where} \quad & G_{\Delta t,\epsilon}(x,\omega_m) = \pi_\epsilon(s_{N,\epsilon}(x,\omega_m) - K) \\ \text{s.t.} \quad & y_{n+1,\epsilon}(x,\omega_m) = y_{n,\epsilon}(x,\omega_m) + a_\epsilon(x, y_{n,\epsilon}(x,\omega_m))\Delta t_n \\ & \qquad + b_\epsilon(x, y_{n,\epsilon}(x,\omega_m))\Delta W_n(\omega_m). \end{aligned} \tag{$P^1_{M,\Delta t,\epsilon}$}$$

As a first step, the pointwise convergence of the objective function of ($P^1_{M,\Delta t,\epsilon}$) to the objective of ($P^1$) will be shown in the next section.

4.2.1 Pointwise Convergence of the Objective Functions

This part deals with the pointwise convergence of $g_{M,\Delta t,\epsilon}(x)$ to $g(x)$ with respect to M,Δt and ϵ. Therefore the total approximation error can be split up into three parts,

$$\begin{aligned} |g_{M,\Delta t,\epsilon}(x) - g(x)| &\leq |g_{M,\Delta t,\epsilon}(x) - g_{\Delta t,\epsilon}(x)| &=: \mathcal{E}_1 \\ &+ |g_{\Delta t,\epsilon}(x) - g_{\epsilon}(x)| &=: \mathcal{E}_2 \\ &+ |g_{\epsilon}(x) - g(x)| &=: \mathcal{E}_3 \end{aligned}$$

namely the Monte Carlo error $\mathcal{E}_1$, the discretization error $\mathcal{E}_2$ and the smoothing error $\mathcal{E}_3$. In the following, these three error components will be analyzed and even a convergence order will be presented. Note that this analysis is restricted to the case of Lipschitz continuous coefficients. Thus, the following assumption is stated.

(A.4)
There exist constants $\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon) > 0$ such that
$\forall t \in [0, T],\ y, \overline{y} \in \mathbb{R}^L :\ \|a_\epsilon(x, y) - a_\epsilon(x, \overline{y})\| \leq \mathcal{L}_{a,y}(\epsilon)\|y - \overline{y}\|$
$\forall t \in [0, T],\ y, \overline{y} \in \mathbb{R}^L :\ \|b_\epsilon(x, y) - b_\epsilon(x, \overline{y})\| \leq \mathcal{L}_{b,y}(\epsilon)\|y - \overline{y}\|.$

Additionally, the coefficients have to fulfill a growth condition

(A.5)
There exists a constant $\mathcal{G} > 0$ such that $\forall x \in X$ and $y \in \mathbb{R}^L$
$\|a_\epsilon(x, y)\| + \|b_\epsilon(x, y)\| \leq \mathcal{G}(1 + \|y\|)$.

4.2.1.1 Smoothing Error

The first aim is to analyze the error $\mathcal{E}_3 = |g_\epsilon(x) - g(x)|$ of the smooth approximation, which is done in the following theorem. Consider therefore the stochastic differential equation (3.9) formulated as integral equation

$$Y_{t,\epsilon} = Y_0 + \int_0^t a_\epsilon(x, Y_{s,\epsilon})ds + \int_0^t b_\epsilon(x, Y_{s,\epsilon})dW_s. \tag{4.12}$$

Theorem 4.9. *If the Lipschitz property* (A.4) *and the error estimation* (A.2) *hold, the smoothing error can be bounded by*

$$\mathcal{E}_3 = |g_\epsilon(x) - g(x)| \leq C\sqrt{\left(1 + \mathcal{L}_y^2(\epsilon)\right)\psi(\epsilon)} + C\epsilon^2,$$

for a suitable choice of the constant C, $\mathcal{L}_y(\epsilon) := \max(\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon))$ *and* $\psi(\cdot)$ *from* (A.2).

Proof. As it follows from the definition of (P^1) and (P^1_ϵ) that

$$\mathcal{E}_3 = |g_\epsilon(x) - g(x)| = |E(\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K)) - E(\pi(S_T(x,\omega) - K))|$$

the triangle inequality yields by inserting $\pi(S_{T,\epsilon}(x,\omega) - K)$

$$\begin{aligned} & |E(\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K)) - E(\pi(S_T(x,\omega) - K))| \\ \leq\ & E(|\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K) - \pi(S_{T,\epsilon}(x,\omega) - K)|) \\ +\ & E(|\pi(S_{T,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|). \end{aligned}$$

The first term can be bounded from above as Lemma 3.3 shows that

$$\|\pi_\epsilon(x) - \pi(x)\|_\infty = \sup_{x\in\mathbb{R}} |\pi_\epsilon(x) - \pi(x)| = \frac{3}{16}\epsilon \tag{4.13}$$

and thus

$$E(|\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K) - \pi(S_{T,\epsilon}(x,\omega) - K)|) = \mathcal{O}(\epsilon^2) \tag{4.14}$$

where the fact has been exploited that there is an $\mathcal{O}(\epsilon)$ probability that smoothing is required, i.e. $S_{T,\epsilon} \in [-\epsilon, \epsilon]$. For the second term let for a fixed $x \in X$

$$Z(T) = E\left(\|Y_{T,\epsilon}(x,\omega) - Y_T(x,\omega)\|^2\right).$$

Without loss of generality, one may assume the existence of the second order moments as one could make otherwise use of a truncation technique like in the proof of Theorem 4.2. If one now exploits the fact that $(a+b)^2 \leq 2(a^2+b^2)$ and the Ito isometry (Lemma 2.5(i)) one receives by inserting the integral equation (4.12)

$$\begin{aligned} Z(T) \ \leq\ & 2\int_0^T E(\|a_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - a(x, Y_t(x,\omega))\|^2)dt \\ +\ & 2\int_0^T E(\|b_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - b(x, Y_t(x,\omega))\|^2)dt. \end{aligned}$$

The triangle inequality provides furthermore that

$$\begin{aligned} Z(T) \ \leq\ & 4\int_0^T E(\|a_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - a_\epsilon(x, Y_t(x,\omega))\|^2)dt \\ +\ & 4\int_0^T E(\|a_\epsilon(x, Y_t(x,\omega)) - a(x, Y_t(x,\omega))\|^2)dt \\ +\ & 4\int_0^T E(\|b_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - b_\epsilon(x, Y_t(x,\omega))\|^2)dt \\ +\ & 4\int_0^T E(\|b_\epsilon(x, Y_t(x,\omega)) - b(x, Y_t(x,\omega))\|^2)dt. \end{aligned}$$

Due to the Lipschitz property (A.4) it holds true that

$$\begin{aligned} & 4\int_0^T E(\|a_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - a_\epsilon(x, Y_t(x,\omega))\|^2)dt \\ + \quad & 4\int_0^T E(\|b_\epsilon(x, Y_{t,\epsilon}(x,\omega)) - b_\epsilon(x, Y_t(x,\omega))\|^2)dt \\ \leq \quad & 8\mathcal{L}_y^2(\epsilon)\int_0^T E\left(\|Y_{t,\epsilon}(x,\omega) - Y_t(x,\omega)\|^2\right)dt \end{aligned}$$

where $\mathcal{L}_y(\epsilon) := \max(\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon))$. Exploiting assumption (A.2) provides

$$\begin{aligned} & 4\int_0^T E(\|a_\epsilon(x, Y_t(x,\omega)) - a(x, Y_t(x,\omega))\|^2)dt \\ + \quad & 4\int_0^T E(\|b_\epsilon(x, Y_t(x,\omega)) - b(x, Y_t(x,\omega))\|^2)dt \\ \leq \quad & 4T\psi(\epsilon). \end{aligned}$$

Thus one has in summary

$$Z(T) \leq 8\mathcal{L}_y^2(\epsilon)\int_0^T Z(t)dt + C_1\psi(\epsilon).$$

One can now apply Gronwall's Lemma (Lemma 2.10), which provides

$$Z(T) \leq C_1\psi(\epsilon) + 8\mathcal{L}_y^2(\epsilon)e^{LT}C_1\psi(\epsilon)\int_0^T e^{-Lt}tdt.$$

Thus $Z(T)$ can be bounded from above by

$$Z(T) \leq C_2\left(\left(1 + \mathcal{L}_y^2(\epsilon)\right)\psi(\epsilon)\right).$$

Consequently, an application of Jensen's inequality (Theorem 2.11) in combination with the Lipschitz continuity of $\pi(\cdot)$ provides

$$E(|\pi(S_{T,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|) = \mathcal{O}\left(\sqrt{\left(1 + \mathcal{L}_y^2(\epsilon)\right)\psi(\epsilon)}\right).$$

Summarizing the two estimates leads to

$$\begin{aligned}\mathcal{E}_3 = |g_\epsilon(x) - g(x)| &= |E(\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K)) - E(\pi(S_T(x,\omega) - K))| \\ &= \mathcal{O}\left(\sqrt{\left(1 + \mathcal{L}_y^2(\epsilon)\right)\psi(\epsilon)} + \epsilon^2\right).\end{aligned}$$

□

4.2.1.2 Discretization Error

The second error term $\mathcal{E}_2$ occurs due to the discretization of the stochastic differential equation. One has by definition that

$$\mathcal{E}_2 = |g_{\Delta t,\epsilon}(x,\omega) - g_\epsilon(x,\omega)| = |E\left(\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K)\right) - E\left(\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K)\right)|.$$

Kloeden and Platen [1999] provide a convergence proof assuming a Lipschitz condition for a simple discretization of $S_T(x,\omega)$. Following this, it is essential that

$$E\left(|s_{N,\epsilon}(x,\omega) - S_{T,\epsilon}(x,\omega)|\right) \leq C\Delta t^{\frac{1}{2}},$$

for an additionally considered and fixed smoothing parameter $\epsilon \geq 0$ under the Lipschitz assumption (A.4). Together with the Lipschitz property of π_ϵ the discretization error $\mathcal{E}_2$ can thus be bounded from above by

$$\mathcal{E}_2 \leq C\Delta t^{\frac{1}{2}}. \tag{4.15}$$

As C from (4.15) depends on the corresponding Lipschitz constants $\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon)$ from assumption (A.4) which itself depends on the smoothing parameter, $\mathcal{E}_2$ is expected to reveal a relation $\mathcal{O}(I(\epsilon)\Delta t^{\frac{1}{2}})$, with some functional I. The following contains a detailed analysis on the discretization error to determine this functional.

Consider therefore the stochastic differential equation formulated as integral equation (4.12). The associated Euler-Maruyama discretized version can be interpolated continuously in the following way:

$$y_{t,\epsilon}(x,\omega) = Y_0 + \int_0^t a_\epsilon(x, y_{\chi(s),\epsilon}(x,\omega))ds + \int_0^t b_\epsilon(x, y_{\chi(s),\epsilon}(x,\omega))dW_s(\omega) \tag{4.16}$$

where $\chi(s) = n$, $\forall s \in [\tau_n, \tau_{n+1})$ and $n = 0, ..., N-1$. In this case it is $y_{\tau_n,\epsilon} = y_{n,\epsilon}$ as it holds true that

$$\begin{aligned}y_{\tau_n,\epsilon} - y_{\tau_{n-1},\epsilon} &= \int_{\tau_{n-1}}^{\tau_n} a_\epsilon(x, y_{\chi(s),\epsilon})ds + \int_{\tau_{n-1}}^{\tau_n} b_\epsilon(x, y_{\chi(s),\epsilon})dW_s \\ &= a_\epsilon(x, y_{n-1,\epsilon})\Delta t_{n-1} + b_\epsilon(x, y_{n-1,\epsilon})\Delta W_{n-1}.\end{aligned}$$

Figure 4.3 shows the continuity of the interpolation for a Black-Scholes example.

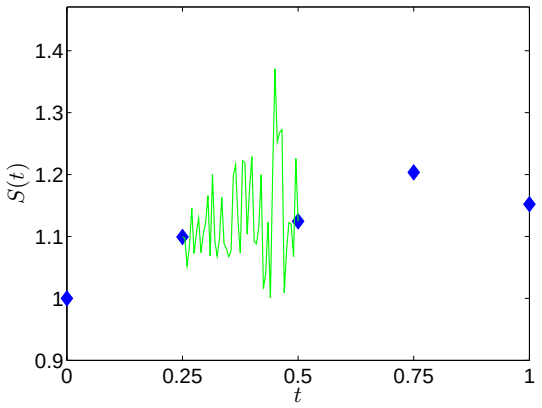

Figure 4.3: Discrete Black Scholes path (blue diamonds) with $\mu = 0.1$, $\sigma = 0.2$, $\Delta t = 0.25$ and interpolated values (green line) for $t \in [0.25, 0.5]$.

To proof an order of $\mathcal{E}_2$, the following Lemma by Kloeden and Platen [1999] is required.

Lemma 4.10. *Suppose that the growth condition* (A.5) *and the Lipschitz continuity* (A.4) *hold. Then* $Y_{t,\epsilon}$ *the solution of* (4.12) *satisfies*

$$E(\|Y_{t,\epsilon} - Y_0\|^2) \leq C_2(1 + \|Y_0\|^2)te^{C_1 t}$$

for $t \in [0, T]$, $\epsilon \geq 0$ *and positive constants* C_1 *and* C_2.

Proof. Kloeden and Platen [1999] provide the proof in Theorem 4.5.4, where in fact the constants C_1 and C_2 do not depend on the inserted smoothing parameter ϵ. □

Now the theorem considering $\mathcal{E}_2$ can be stated.

Theorem 4.11. *Suppose that the Lipschitz continuity* (A.4) *and the growth condition* (A.5) *hold. The discretization error can be estimated in the following way:*

$$\mathcal{E}_2 = |g_{\Delta t,\epsilon}(x) - g_\epsilon(x)| \leq C\mathcal{L}_y(\epsilon)\Delta t^{\frac{1}{2}}$$

where $\mathcal{L}_y(\epsilon) := \max(\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon))$.

Proof. It holds true by definition that

$$|g_{\Delta t,\epsilon}(x) - g_\epsilon(x)| = |E(\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K)) - E(\pi_\epsilon(S_{T,\epsilon}(x,\omega) - K))|$$

which can be estimated in the following way

$$|E(\pi_\epsilon(s_{N,\epsilon}(x,\omega)-K))-E(\pi_\epsilon(S_{T,\epsilon}(x,\omega)-K))| \le E(|s_{N,\epsilon}(x,\omega)-S_{T,\epsilon}(x,\omega)|)$$

as π_ϵ is Lipschitz continuous where the Lipschitz constant of π deals as an upper bound (see also figure 3.1). Now let

$$Z(T) := E\left(\|y_{N,\epsilon}(x,\omega)-Y_{T,\epsilon}(x,\omega)\|^2\right)$$

where again the existence of the second order moments can be assumed without loss of generality (see also the proofs of Theorem 4.2 and Theorem 4.9). Inserting the solution of $y_{N,\epsilon}$ and $Y_{T,\epsilon}$ as well as skipping the dependency from x and ω to facilitate notation leads to

$$\begin{aligned} Z(T) &= E\left(\left\|\int_0^T a_\epsilon(x,Y_{t,\epsilon})-a_\epsilon(x,y_{\chi(t),\epsilon})dt\right.\right. \\ &+ \left.\left.\int_0^T b_\epsilon(x,Y_{t,\epsilon})-b_\epsilon(x,y_{\chi(t),\epsilon})dW_t\right\|^2\right). \end{aligned}$$

Ito isometry (Lemma 2.5(i)) and $(a+b)^2 \le 2(a^2+b^2)$ together yield

$$\begin{aligned} Z(T) &\le 2\int_0^T E\left(\left\|a_\epsilon(x,Y_{t,\epsilon})-a_\epsilon(x,y_{\chi(t),\epsilon})\right\|^2\right)dt \\ &+ 2\int_0^T E\left(\left\|b_\epsilon(x,Y_{t,\epsilon})-b_\epsilon(x,y_{\chi(t),\epsilon})\right\|^2\right)dt. \end{aligned}$$

The assumed Lipschitz property of a_ϵ and b_ϵ allows the following estimate:

$$Z(T) \le 2\mathcal{L}_{a,y}^2(\epsilon)\int_0^T E\left(\left\|Y_{t,\epsilon}-y_{\chi(t),\epsilon}\right\|^2\right)dt + 2\mathcal{L}_{b,y}^2(\epsilon)\int_0^T E\left(\left\|Y_{t,\epsilon}-y_{\chi(t),\epsilon}\right\|^2\right)dt$$

which can be combined to

$$Z(T) \le 4\mathcal{L}_y^2(\epsilon)\int_0^T E\left(\left\|Y_{t,\epsilon}-y_{\chi(t),\epsilon}\right\|^2\right)dt$$

with $\mathcal{L}_y(\epsilon) := \max(\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon))$. After all Lemma 4.10 yields similar to Kloeden

and Platen [1999] pp. 343 f.

$$Z(T) \leq C_1 \mathcal{L}_y^2(\epsilon) \Delta t e^{C_2 \Delta t}.$$

If one now estimates the exponential term with the first two terms of the exponential series, namely $1 + C_2 \Delta t$, this expression can be bounded from above as Δt is assumed to converge to 0. Hence $Z(T) \leq C_1 \mathcal{L}_y^2(\epsilon) \Delta t$. From the definition of $Z(T)$ it follows with Jensen's inequality (Theorem 2.11), namely with

$$E(\|y_{N,\epsilon}(x,\omega) - Y_{T,\epsilon}(x,\omega)\|)^2 \leq E(\|y_{N,\epsilon}(x,\omega) - Y_{T,\epsilon}(x,\omega)\|^2)$$

that

$$E(\|y_{N,\epsilon}(x,\omega) - Y_{T,\epsilon}(x,\omega)\|) = \mathcal{O}(\mathcal{L}_y(\epsilon) \Delta t^{\frac{1}{2}}).$$

Due to the Lipschitz continuity of π_ϵ it follows finally that

$$\mathcal{E}_2 = |g_{\Delta t,\epsilon}(x) - g_\epsilon(x)| = \mathcal{O}(\mathcal{L}_y(\epsilon) \Delta t^{\frac{1}{2}})$$

which means the proof of the statement. □

4.2.1.3 Monte Carlo Error

The *Central Limit Theorem* is the crucial result to analyze the Monte Carlo Error.

Theorem 4.12. (Central Limit Theorem) *Let $(X_m)_m$ a sequence of independent and identically distributed, square integrable real valued random variables with expectation μ variance σ^2. Then it holds true that*

$$\sqrt{M} \left(\frac{1}{M} \sum_{m=1}^{M} X_m - \mu \right) \Rightarrow N(0, \sigma^2).$$

Proof. Bauer [2002] □

This means that for a fixed variance σ^2 with increasing number of simulations $(\frac{1}{M} \sum_{m=1}^{M} X_m - \mu)$ decreases faster than $\sqrt{M}$ increases which provides the well known result that the Monte Carlo approximation behaves asymptotically like $\mathcal{O}(1/\sqrt{M})$. Consider now the first error term $\mathcal{E}_1 = |g_{M,\Delta t,\epsilon}(x) - g_{\Delta t,\epsilon}(x)|$ with

$$g_{\mathrm{M},\Delta t,\epsilon}(x) = \frac{1}{M} \sum_{m=1}^{M} \pi_\epsilon(s_{N,\epsilon}(x) - K)$$

and

$$g_{\Delta t,\epsilon}(x) = E\left(\pi_\epsilon(s_{N,\epsilon}(x) - K)\right).$$

For fixed $\Delta t, \epsilon \geq 0$ it holds by an application of the central limit theorem, that

$$\sqrt{M}\,|g_{\mathrm{M},\Delta t,\epsilon}(x) - g_{\Delta t,\epsilon}(x)| \Rightarrow N(0, \sigma^2_{\Delta t,\epsilon}),$$

where $\sigma^2_{\Delta t,\epsilon}$ is the variance of $\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K)$ which certainly depends on the chosen Δt and ϵ. Thus, as the goal is to decrease the overall error with increasing number of simulations as well as decreasing discretization step size and smoothing parameter, one has to make sure that $\sigma^2_{\Delta t,\epsilon}$ is at least bounded for $\Delta t \to 0$ and $\epsilon \to 0$. This is supported by the following Lemma

Lemma 4.13. *Under the Lipschitz assumption* (A.4) *and the coefficient error assumption* (A.2) *the variance can be bounded by*

$$\mathrm{Var}\left(\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)\right) \leq C\left((1 + \mathcal{L}_y^2(\epsilon))\psi(\epsilon) + \mathcal{L}_y^2(\epsilon)\Delta t + \epsilon^4\right)$$

for a suitably chosen constant $C > 0$.

Proof. The variance is defined as

$$\begin{aligned} & \mathrm{Var}\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|\right) \\ = \;& E\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|^2\right) \\ - \;& E\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|\right)^2. \end{aligned} \tag{4.17}$$

As the second term on the right side is nonnegative, the variance can be bounded from above by simply considering the first term:

$$\begin{aligned} & \mathrm{Var}\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|\right) \\ \leq \;& E\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|^2\right). \end{aligned}$$

As $\pi(\cdot)$ is Lipschitz continuous with Lipschitz constant 1, one can estimate the term on the right side exploiting this fact together with $(a+b)^2 < 2a^2 + 2b^2$, Jensen's inequality and (4.14) by

$$\begin{aligned} & E\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|^2\right) \\ \leq \;& 2E\left(|\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(s_{N,\epsilon}(x,\omega) - K)|^2\right) \\ + \;& 2E\left(|\pi(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)|^2\right) \\ \leq \;& 2E\left(|s_{N,\epsilon}(x,\omega) - S_T(x,\omega)|^2\right) + C\epsilon^4. \end{aligned}$$

An application of the triangle inequality provides furthermore

$$\begin{aligned} & E\left(|s_{N,\epsilon}(x,\omega) - S_T(x,\omega)|^2\right) \\ \leq \; & 2E\left(|s_{N,\epsilon}(x,\omega) - S_{T,\epsilon}(x,\omega)|^2\right) + 2E\left(|S_{T,\epsilon}(x,\omega) - S_T(x,\omega)|^2\right). \end{aligned}$$

Making use of Jensen's inequality allows to estimate the first term with the squared discretization error $\mathcal{E}_2$ and the second with the squared smoothing error $\mathcal{E}_3$, already bounded from above in section 4.2.1.1 and 4.2.1.2. Thus

$$\mathrm{Var}\left(\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)\right) = \mathcal{O}\left((1 + \mathcal{L}_y^2(\epsilon))\psi(\epsilon) + \mathcal{L}_y^2(\epsilon)\Delta t + \epsilon^4\right).$$

which proofs the statement. □

Hence one has in summary under suitable assumptions the following result:

Corollary 4.14. *Considering that the assumptions* (A.2), (A.4) *and* (A.5) *hold and that additionally the Lipschitz constants from* (A.4) *are uniformly bounded, i.e. bounded for all* $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ *and* $y \in \mathbb{R}^L$, *the total approximation error* $|g_{M,\Delta t,\epsilon}(x) - g(x)|$ *has the following order:*

$$|g_{M,\Delta t,\epsilon}(x) - g(x)| = \mathcal{O}\left(1/\sqrt{M} + \sqrt{\Delta t} + \sqrt{\psi(\epsilon)}\right). \tag{4.18}$$

Proof. The boundedness of the Lipschitz constants provides that

$$\lim_{\substack{\Delta t \to 0 \\ \epsilon \to 0}} \mathrm{Var}\left(\pi_\epsilon(s_{N,\epsilon}(x,\omega) - K) - \pi(S_T(x,\omega) - K)\right) = 0$$

due to Lemma 4.13 as the terms $\mathcal{L}_y(\epsilon)\psi(\epsilon)$ as well as $\mathcal{L}_y(\epsilon)\Delta t$ now converge to zero. Thus, the proof follows directly from an additional application of Theorem 4.9, Theorem 4.11. □

In the special case of coefficient functions with a linear structure such that $\psi(\epsilon) = C\epsilon^2$ and uniformly bounded constants $\mathcal{L}_y(\epsilon)$, the total error is hence of the order $\mathcal{O}(1/\sqrt{M} + \Delta t^{\frac{1}{2}} + \epsilon)$. In this situation the choice $\Delta t_k = c_1/M_k$, $\epsilon_k = c_2/\sqrt{M_k}$ drives all three error components with the same speed to zero, and hence does not waste numerical effort in the reduction of one error, while another one still dominates the total error. This will numerically be confirmed for the test case of the Stein-Stein model in chapter 7.

4.2.2 Uniform Convergence

With the help of the pointwise convergence from the previous section, a uniform convergence can be shown, if two additional propositions hold, namely the continuity

of g and the epicontinuity of $g_{M,\Delta t,\epsilon}$. For the latter an additional Lipschitz condition will be required:

$$\begin{aligned} &\text{There exist constants } \mathcal{L}_{a,x}(\epsilon), \mathcal{L}_{b,x}(\epsilon) > 0 \text{ such that} \\ (A.6) \qquad &\forall t \in [0,\, T],\, x, \overline{x} \in \mathbb{R}^P :\, \|a_\epsilon(x,y) - a_\epsilon(\overline{x},y)\| \leq \mathcal{L}_{a,x}(\epsilon)\|x - \overline{x}\| \\ &\forall t \in [0,\, T],\, x, \overline{x} \in \mathbb{R}^P :\, \|b_\epsilon(x,y) - b_\epsilon(\overline{x},y)\| \leq \mathcal{L}_{b,x}(\epsilon)\|x - \overline{x}\| \end{aligned}$$

The first Lemma deals with the continuity of g where the dominated integrability of $\{\pi_\epsilon(S_T(x,\omega) - K), x \in X\}$ is a crucial issue:

Definition 4.15. *A family $\{F(x,\omega), x \in X\}$ is dominated by a Q-integrable function, if there exists a function $\overline{F}(\omega)$ with $E_Q(\overline{F}(\omega)) < \infty$ and $|F(x,\omega)| \leq \overline{F}(\omega)$ for all $x \in X$ and Q-almost every ω.*

Under this assumption, the continuity of g can be easily proved:

Lemma 4.16. *Consider that assumption* (A.3)*, namely the continuity of π_ϵ holds and furthermore, that $\{\pi_\epsilon(S_T(x,\omega) - K), x \in X\}$ is dominated integrable. Then, g is already continuous.*

Proof. Given a sequence $(x_k)_k$ with $x_k \to x^*$ one has by definition that

$$\lim_{k\to\infty} g(x_k) = \lim_{k\to\infty} E\left(\pi_\epsilon(S_T(x_k,\omega))\right).$$

An application of Lebesgue's dominated convergence theorem (Theorem 2.14) yields due to the dominated convergence of $\pi_\epsilon(S_T(x,\omega) - K)$

$$\lim_{k\to\infty} E\left(\pi_\epsilon(S_T(x_k,\omega))\right) = E\left(\lim_{k\to\infty} \pi_\epsilon(S_T(x_k,\omega))\right).$$

Finally, the continuity of $\pi_\epsilon(\cdot)$, in particular

$$E\left(\lim_{k\to\infty} \pi_\epsilon(S_T(x_k,\omega))\right) = E\left(\pi_\epsilon(S_T(x^*,\omega))\right) = g(x^*)$$

proves the statement. □

Besides this Lemma, the epicontinuity of $g_{M,\Delta t,\epsilon}$ will be necessary to be able to proof the desired uniform convergence result. This requires harder assumptions, as it will be shown in the following Lemma:

Lemma 4.17. *Considering that the Lipschitz continuity* (A.4) *and* (A.6) *hold with uniformly bounded Lipschitz constants in $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ and $y \in \mathbb{R}^L$, the mapping $g_{M,\Delta t,\epsilon}$ is epicontinuous, i.e. for M large enough and for all $\Delta t, \epsilon \geq 0$:*

$$\lim_{\delta\to 0} \sup_{x \in U(x^0,\delta)} \left|g_{\mathrm{M},\Delta t,\epsilon}(x) - g_{\mathrm{M},\Delta t,\epsilon}(x^0)\right| = 0$$

where $U(x^0, \delta)$ is a neighborhood of $x^0 \in X$ with radius δ.

Proof. Choose $x, x^0 \in X$ and consider

$$E\left(\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right).$$

Inserting the interpolated integral equation (4.16) yields

$$\begin{aligned} & E\left(\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right) \\ = \quad & E\left(\left\|\int_0^T a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))dt\right.\right. \\ + \quad & \left.\left.\int_0^T b_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - b_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))dW_t\right\|^2\right). \end{aligned}$$

Making use of the Ito isometry (Lemma 2.5(i)) and the fact that $(a+b)^2 \leq 2(a^2+b^2)$ provides

$$\begin{aligned} & E\left(\left\|\int_0^T a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))dt\right.\right. \\ + \quad & \left.\left.\int_0^T b_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - b_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))dW_t\right\|^2\right) \\ \leq \quad & 2\int_0^t E(\|a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))\|^2)dt \\ + \quad & 2\int_0^t E(\|b_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - b_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))\|^2)dt. \end{aligned}$$

In the latter only the first term on the right side is considered, as the second one can be treated analogously. The triangle inequality yields

$$\begin{aligned} & 2\int_0^t E(\|a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))\|^2)dt \\ \leq \quad & 4\int_0^t E(\|a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x,\omega))\|^2)dt \\ + \quad & 4\int_0^t E(\|a_\epsilon(x^0, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))\|^2)dt. \end{aligned}$$

The assumed Lipschitz continuity of a_ϵ with respect to both variables leads to

$$\begin{aligned} & 4\int_0^t E(\|a_\epsilon(x, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x,\omega))\|^2)dt \\ + \quad & 4\int_0^t E(\|a_\epsilon(x^0, y_{\chi(t),\epsilon}(x,\omega)) - a_\epsilon(x^0, y_{\chi(t),\epsilon}(x^0,\omega))\|^2)dt \\ \leq \quad & 4\int_0^T \mathcal{L}_{a,x}^2(\epsilon)\left|x - x^0\right|^2 dt + \int_0^T \mathcal{L}_{a,y}^2(\epsilon) 4E\left(\left\|y_{\chi(t),\epsilon}(x,\omega)) - y_{\chi(t),\epsilon}(x^0,\omega)\right\|^2\right) dt. \end{aligned}$$

Summarizing where b is treated in analogy to a yields

$$\begin{aligned} & E\left(\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right) \\ \leq \quad & 8T\mathcal{L}_x^2(\epsilon)\left|x - x^0\right|^2 + 8\mathcal{L}_y^2(\epsilon)\int_0^T E\left(\left\|y_{\chi(t),\epsilon}(x,\omega)) - y_{\chi(t),\epsilon}(x^0,\omega)\right\|^2\right) dt \end{aligned}$$

where $\mathcal{L}_x(\epsilon) := \max(\mathcal{L}_{a,x}(\epsilon), \mathcal{L}_{b,x}(\epsilon))$ and $\mathcal{L}_y(\epsilon) := \max(\mathcal{L}_{a,y}(\epsilon), \mathcal{L}_{b,y}(\epsilon))$. An application of the Gronwall Lemma in combination with the boundedness of the Lipschitz constants thus leads to

$$E\left(\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right) \leq C_1\left|x - x^0\right|^2.$$

which provides with help of the Doob inequality that

$$\begin{aligned} & E\left(\sup_{x\in U(x^0,\delta)}\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right) \\ \leq \quad & 4\sup_{x\in U(x^0,\delta)} E\left(\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|^2\right) \\ \leq \quad & C_2\sup_{x\in U(x^0,\delta)}\left|x - x^0\right|^2. \end{aligned}$$

Thus, with Jensen inequality, it is essential that

$$E\left(\sup_{x\in U(x^0,\delta)}\left\|y_{N,\epsilon}(x,\omega) - y_{N,\epsilon}(x^0,\omega)\right\|\right) \leq C\sup_{x\in U(x^0,\delta)}\left|x - x^0\right|^2. \tag{4.20}$$

If one now considers

$$\sup_{x\in U(x^0,\delta)}\left|g_{M,\Delta t,\epsilon}(x) - g_{M,\Delta t,\epsilon}(x^0)\right|$$

it follows from the definition of $g_{\mathrm{M},\Delta t,\epsilon}(x)$ that

$$\begin{aligned} & \sup_{x\in U(x^0,\delta)} \left| g_{\mathrm{M},\Delta t,\epsilon}(x) - g_{\mathrm{M},\Delta t,\epsilon}(x^0) \right| \\ = & \sup_{x\in U(x^0,\delta)} \left| \frac{1}{M}\sum_{m=1}^{M} \pi_\epsilon(s_{N,\epsilon}(x,\omega_m) - K) - \frac{1}{M}\sum_{m=1}^{M} \pi_\epsilon(s_{N,\epsilon}(x^0,\omega_m) - K) \right|. \end{aligned}$$

Again, the triangle inequality in combination with $\sup(a+b) \leq \sup(a) + \sup(b)$ provides

$$\begin{aligned} & \sup_{x\in U(x^0,\delta)} \left| \frac{1}{M}\sum_{m=1}^{M} \pi_\epsilon(s_{N,\epsilon}(x,\omega_m) - K) - \frac{1}{M}\sum_{m=1}^{M} \pi_\epsilon(s_{N,\epsilon}(x^0,\omega_m) - K) \right| \\ \leq & \frac{1}{M}\sum_{m=1}^{M} \sup_{x\in U(x^0,\delta)} \left| \pi_\epsilon(s_{N,\epsilon}(x,\omega_m) - K) - \pi_\epsilon(s_{N,\epsilon}(x^0,\omega_m) - K) \right|. \end{aligned}$$

As π_ϵ is Lipschitz continuous, it holds true that

$$\begin{aligned} & \frac{1}{M}\sum_{m=1}^{M} \sup_{x\in U(x^0,\delta)} \left| \pi_\epsilon(s_{N,\epsilon}(x,\omega_m) - K) - \pi_\epsilon(s_{N,\epsilon}(x^0,\omega_m) - K) \right| \\ \leq & \frac{1}{M}\sum_{m=1}^{M} \sup_{x\in U(x^0,\delta)} \left| s_{N,\epsilon}(x,\omega_m) - s_{N,\epsilon}(x^0,\omega_m) \right| \end{aligned}$$

which converges due to the Law of Large Numbers (Theorem 2.15):

$$\begin{aligned} & \frac{1}{M}\sum_{m=1}^{M} \sup_{x\in U(x^0,\delta)} \left| s_{N,\epsilon}(x,\omega_m) - s_{N,\epsilon}(x^0,\omega_m) \right| \\ \xrightarrow[M\to\infty]{} & E\left(\sup_{x\in U(x^0,\delta)} \left| s_{N,\epsilon}(x,\omega) - s_{N,\epsilon}(x^0,\omega) \right| \right). \end{aligned}$$

With the help of (4.20) this term can be estimated in the following way:

$$E\left(\sup_{x\in U(x^0,\delta)} \left| s_{N,\epsilon}(x,\omega) - s_{N,\epsilon}(x^0,\omega) \right| \right) \leq C \left| x - x^0 \right|.$$

Taking the limit over δ finally provides for M large enough and for all $\Delta t, \epsilon \geq 0$

$$\begin{aligned} & \lim_{\delta\to 0} \sup_{x\in U(x^0,\delta)} \left| g_{\mathrm{M},\Delta t,\epsilon}(x) - g_{\mathrm{M},\Delta t,\epsilon}(x^0) \right| \\ \leq & \lim_{\delta\to 0} E\left(\sup_{x\in U(x^0,\delta)} \left| s_{N,\epsilon}(x,\omega) - s_{N,\epsilon}(x^0,\omega) \right| \right) \\ = & 0. \end{aligned}$$

□

Now, the uniform convergence can be shown in the following Theorem.

Theorem 4.18. *Consider that the assumptions* (A.2)-(A.6) *hold. If additionally the Lipschitz constants are uniformly bounded in* $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ *and* $y \in \mathbb{R}^L$ *and* $\{\pi_\epsilon(S_T(x,\omega) - K), x \in X\}$ *is dominated integrable it is essential that*

$$\lim_{k\to\infty} \sup_{x\in X} |g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g(x)| = 0.$$

Proof. Given an arbitrary $x^0 \in X$ and a neighborhood $U(x^0,\delta^0)$ it holds true that

$$\begin{aligned}\sup_{x\in U(x^0,\delta^0)} |g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g(x)| \quad &\leq \quad \sup_{x\in U(x^0,\delta^0)} \left|g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x^0)\right| \\ &+ \quad \left|g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x^0) - g(x^0)\right| \\ &+ \quad \sup_{x\in U(x^0,\delta^0)} \left|g(x^0) - g(x)\right|.\end{aligned}$$

Due to Corollary 4.14, Lemma 4.16 and Lemma 4.17 all three terms can be bounded from above. Moreover it holds certainly true that $\cup_{x^0\in X} U(x^0,\delta^0) \supset X$. As X is convex there exists a finite number of points $x_1, ..., x_J$ and a corresponding finite covering of X, namely $U(x_1,\delta_1), ..., U(x_J,\delta_J)$ such that it is essential for every $U(x_j,\delta_j)$ that

$$\lim_{k\to\infty} \lim_{\delta_j\to 0} \sup_{x\in U(x_j,\delta_j)} |g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g(x)| = 0,\ j = 1, ..., J.$$

Thus, it holds true that

$$\begin{aligned}&\lim_{k\to\infty} \sup_{x\in X} |g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g(x)| \\ \leq \quad &\sum_{j=1}^{J} \lim_{k\to\infty} \lim_{\delta_j\to 0} \sup_{x\in U(x_j,\delta_j)} |g_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x) - g(x)| \\ = \quad &0.\end{aligned}$$

□

This Theorem proves the uniform convergence of $g_{\mathsf{M},\Delta t,\epsilon}$ to g. Certainly, this is desired for $f_{\mathsf{M},\Delta t,\epsilon}$ and f which in particular can be deduced from the above Theorem, as already mentioned in the very beginning of this section.

Remark 4.19. *Consider the objective functions of the problems* (P) *and* ($\mathsf{P}_{\mathsf{M},\Delta t,\epsilon}$).

Due to the binomial formula it holds true that

$$\begin{aligned} |f_{M,\Delta t,\epsilon}(x) - f(x)| &= \left| \sum_{i=1}^{I} \left(C^i_{M,\Delta t,\epsilon}(x) - C^i_{\text{obs}} \right)^2 - \sum_{i=1}^{I} \left(C^i(x) - C^i_{\text{obs}} \right)^2 \right| \\ &\leq \sum_{i=1}^{I} 2 \left| C^i_{\text{obs}} \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right| \\ &+ \sum_{i=1}^{I} \left| C^i_{M,\Delta t,\epsilon}(x)^2 - C^i(x)^2 \right|. \end{aligned}$$

An application of the triangle inequality provides furthermore

$$\begin{aligned} |f_{M,\Delta t,\epsilon}(x) - f(x)| &\leq \sum_{i=1}^{I} 2 \left| C^i_{\text{obs}} \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right| \\ &+ \sum_{i=1}^{I} \left| C^i_{M,\Delta t,\epsilon}(x) \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right| \\ &+ \sum_{i=1}^{I} \left| C^i(x) \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right| \end{aligned}$$

which in summary means that

$$|f_{M,\Delta t,\epsilon}(x) - f(x)| \leq \sum_{i=1}^{I} \left| \overline{C}(x) \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right| \tag{4.21}$$

where $\overline{C}(x) = \max_{i=1,\dots,I}(\max(C^i_{M,\Delta t,\epsilon}(x), C^i(x), C^i_{\text{obs}}))$.

Thus, the uniform convergence of $f_{\text{M},\Delta t,\epsilon}$ to f requires the uniform convergence of $C^i_{M,\Delta t,\epsilon}(x)$ to $C^i(x)$ for every $i = 1, ..., I$ and the continuity of $\overline{C}(x)$. The first one can be ensured by applying Theorem 4.18 for every option $i = 1, ..., I$. As $C^i_{M,\Delta t,\epsilon}(x)$ and $C^i(x)$ are additionally continuous (Assumption (A.3) and Lemma 4.16) the following corollary can be stated:

Corollary 4.20. *Consider that the assumptions* (A.2)-(A.6) *hold. If additionally the Lipschitz constants are uniformly bounded in* $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ *and* $y \in \mathbb{R}^L$ *and* $\{\pi_\epsilon(S_{T_i}(x, \omega) - K_i), x \in X\}$ *are dominated integrable for* $i = 1, ..., I$ *it is essential that*

$$\lim_{k \to \infty} \sup_{x \in X} |f_{\text{M}_k,\Delta t_k,\epsilon_k}(x) - f(x)| = 0.$$

Proof. From Theorem 4.18 it follows that

$$\lim_{k \to \infty} \sup_{x \in X} \left| C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i(x) \right| = 0, \; i = 1, ..., I.$$

As Remark 4.19 shows that

$$\lim_{k \to \infty} \sup_{x \in X} |f_{\text{M}_k,\Delta t_k,\epsilon_k}(x) - f(x)| \leq \lim_{k \to \infty} \sup_{x \in X} \sum_{i=1}^{I} \left| \overline{C}(x) \right| \left| C^i_{M,\Delta t,\epsilon}(x) - C^i(x) \right|$$

with $\overline{C}(x) = \max_{i=1,\dots,I}(\max(C^i_{M,\Delta t,\epsilon}(x), C^i(x), C^i_{\text{obs}}))$ this proves the statement due to Assumption (A.3) and Lemma 4.16. □

As the next section deals with first order optimality, the same result for the objectives gradients $\nabla f_{\mathsf{M}_k,\Delta t_k,\epsilon_k}(x)$ and $\nabla f(x)$ will be required. Obviously, making analogous assumptions as for Corollary 4.20 will allow for a similar result. However, as already discussed in section 3.3, the mapping $\pi(\cdot)$ is not differentiable for $S_T = K$. But as the event $\{S_T = K\}$ has probability zero (e.g. Glasserman [2003] p. 388), $\pi(S_{T_i}(\cdot,\omega) - K_i)$ is at least almost surely differentiable.

For additional assumptions in analogy to (A.2), (A.4), (A.5) and (A.6) consider the gradient components

$$\frac{\partial}{\partial x_p} C^i_{\mathsf{M},\Delta t,\epsilon}(x) = \frac{1}{M}\sum_{m=1}^{M} \pi'_\epsilon(s^m_{N_i,\epsilon}(x,\omega) - K_i)\frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}(x,\omega)$$

where

$$\begin{aligned}
\tfrac{\partial}{\partial x_p} y^m_{n+1,\epsilon} &= \tfrac{\partial}{\partial x_p} y^m_{n,\epsilon} + \left[\tfrac{\partial}{\partial y} a_\epsilon(x, y^m_{n,\epsilon})\tfrac{\partial}{\partial x_p} y^m_{n,\epsilon} + \tfrac{\partial}{\partial x} a_\epsilon(x, y^m_{n,\epsilon})\Delta x\right]\Delta t_n \\
&\quad + \left[\tfrac{\partial}{\partial y}\left(b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n\right)\tfrac{\partial}{\partial x_p} y^m_{n,\epsilon} + \tfrac{\partial}{\partial x}\left(b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n\right)\Delta x\right] \\
\eta^m_0 &= 0, \quad n = 0, \dots, N-1, \quad m = 1, \dots, M, \quad N := \max_{i=1,\dots,I} N_i.
\end{aligned}$$

The following assumptions are consequently stated:

(A.7)
$$\begin{aligned}
&\|\tfrac{\partial}{\partial x_p} a_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} a(x,y)\|^2_\infty + \|\tfrac{\partial}{\partial x_p} b_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} b(x,y)\|^2_\infty < \psi'_x(\epsilon) \\
&\text{with } \psi'_x : \mathbb{R} \to \mathbb{R} \text{ and } \lim_{\epsilon \to 0} \psi'_x(\epsilon) = 0 \text{ for } p = 1, \dots, P. \\
&\|\tfrac{\partial}{\partial y} a_\epsilon(x,y) - \tfrac{\partial}{\partial y} a(x,y)\|^2_\infty + \|\tfrac{\partial}{\partial y} b_\epsilon(x,y) - \tfrac{\partial}{\partial y} b(x,y)\|^2_\infty < \psi'_y(\epsilon) \\
&\text{with } \psi'_y : \mathbb{R} \to \mathbb{R} \text{ and } \lim_{\epsilon \to 0} \psi'_y(\epsilon) = 0 \text{ for } p = 1, \dots, P.
\end{aligned}$$

(A.8)
$$\begin{aligned}
&\text{There exist constants } \mathcal{L}'_y(\epsilon), \mathcal{L}'_x(\epsilon) > 0 \text{ such that} \\
&\forall t \in [0, T], y, \overline{y} \in \mathbb{R}^L, x, \overline{x} \in \mathbb{R}^P, p = 1, \dots, P : \\
&\|\tfrac{\partial}{\partial x_p} a_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} a_\epsilon(x,\overline{y})\| + \|\tfrac{\partial}{\partial x_p} b_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} b_\epsilon(x,\overline{y})\| \\
&+ \|\tfrac{\partial}{\partial y} a_\epsilon(x,y) - \tfrac{\partial}{\partial y} a_\epsilon(x,\overline{y})\| + \|\tfrac{\partial}{\partial y} b_\epsilon(x,y) - \tfrac{\partial}{\partial y} b_\epsilon(x,\overline{y})\| \\
&\leq \mathcal{L}'_y(\epsilon)\|y - \overline{y}\| \text{ and} \\
&\|\tfrac{\partial}{\partial x_p} a_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} a_\epsilon(\overline{x},y)\| + \|\tfrac{\partial}{\partial x_p} b_\epsilon(x,y) - \tfrac{\partial}{\partial x_p} b_\epsilon(\overline{x},y)\| \\
&+ \|\tfrac{\partial}{\partial y} a_\epsilon(x,y) - \tfrac{\partial}{\partial y} a_\epsilon(\overline{x},y)\| + \|\tfrac{\partial}{\partial y} b_\epsilon(x,y) - \tfrac{\partial}{\partial y} b_\epsilon(\overline{x},y)\| \\
&\leq \mathcal{L}'_x(\epsilon)\|x - \overline{x}\|
\end{aligned}$$

(A.9) There exists constants $\mathcal{G}'_y, \mathcal{G}'_x > 0$ for $p = 1, ..., P$ such that
$\forall x \in X$ and $y \in \mathbb{R}^L : \|\frac{\partial}{\partial x_p} a_\epsilon(x, y)\| + \|\frac{\partial}{\partial x_p} b_\epsilon(x, y)\| \leq \mathcal{G}'_x(1 + \|y\|)$
$\forall x \in X$ and $y \in \mathbb{R}^L : \|\frac{\partial}{\partial y} a_\epsilon(x, y)\| + \|\frac{\partial}{\partial y} b_\epsilon(x, y)\| \leq \mathcal{G}'_y(1 + \|y\|)$.

On the basis of these, a result in analogy to Corollary 4.20 can be stated:

Corollary 4.21. *In the situation of Corollary 4.20 consider that additionally the families* $\{\frac{\partial}{\partial x_p}\pi_\epsilon(S_{T_i}(x, \omega) - K_i), x \in X\}$ *are almost surely dominated integrable for* $i = 1, ..., I$ *and* $p = 1, ..., P$ *and that the above stated assumptions for the coefficients derivatives* (A.7)-(A.9) *hold true. If additionally* $\mathcal{L}_y(\epsilon)$ *and* $\mathcal{L}'_y(\epsilon)$ *are uniformly bounded in* $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ *and* $y \in \mathbb{R}^L$, *it is thus essential that*

$$\lim_{k\to\infty} \sup_{x\in X} \|\nabla f_{M_k,\Delta t_k,\epsilon_k}(x) - \nabla f(x)\| = 0.$$

Proof. An application of the triangle inequality provides for every $\frac{\partial}{\partial x_p} f_{M,\Delta t,\epsilon}$ and $p = 1, ..., P$ that

$$\begin{aligned}
& \left| \frac{\partial}{\partial x_p} f_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} f(x) \right| \\
= & \left| \sum_{i=1}^{I} 2 \left(C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i_{\text{obs}} \right) \frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) \right. \\
- & \left. \sum_{i=1}^{I} 2 \left(C^i(x) - C^i_{\text{obs}} \right) \frac{\partial}{\partial x_p} C^i(x) \right| \\
\leq & 2 \sum_{i=1}^{I} \left| C^i_{M_k,\Delta t_k,\epsilon_k}(x) \frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i(x) \frac{\partial}{\partial x_p} C^i(x) \right| \\
+ & 2 \sum_{i=1}^{I} \left| \left(\frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} C^i(x) \right) C^i_{\text{obs}} \right|.
\end{aligned}$$

For the first summand, one can derive the estimate

$$\begin{aligned}
& \left| C^i_{M_k,\Delta t_k,\epsilon_k}(x) \frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i(x) \frac{\partial}{\partial x_p} C^i(x) \right| \\
\leq & \left| C^i_{M_k,\Delta t_k,\epsilon_k}(x) \right| \cdot \left| \frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} C^i(x) \right| \\
+ & \left| \frac{\partial}{\partial x_p} C^i(x) \right| \cdot \left| C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i(x) \right|.
\end{aligned}$$

Summarizing, this yields

$$\begin{aligned}
&\left|\frac{\partial}{\partial x_p} f_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} f(x)\right| \\
\leq\ & 2\left|C^i_{M_k,\Delta t_k,\epsilon_k}(x)\right| \cdot \left|\frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} C^i(x)\right| \\
+\ & 2\left|\frac{\partial}{\partial x_p} C^i(x)\right| \cdot \left|C^i_{M_k,\Delta t_k,\epsilon_k}(x) - C^i(x)\right| \\
+\ & 2\sum_{i=1}^{I}\left|\left(\frac{\partial}{\partial x_p} C^i_{M_k,\Delta t_k,\epsilon_k}(x) - \frac{\partial}{\partial x_p} C^i(x)\right) C^i_{\text{obs}}\right|.
\end{aligned}$$

As Lemma 4.16 together with the dominated integrability assumptions ensured the continuity of $\frac{\partial}{\partial x_p} C^i(x)$ these three terms converge uniformly to zero as the assumptions here allow the application of Corollary 4.20 also to the gradients such that one obtains in summary

$$\lim_{k\to\infty} \sup_{x\in X} \left|\left|\nabla f^i_{M_k,\Delta t_k,\epsilon_k}(x) - \nabla f(x)\right|\right|_2 = 0 \quad (a.s.).$$

□

As the final step, the optimality will be analyzed in the next section.

4.2.3 First Order Optimality

To answer the question on first order optimality reconsider the necessary optimality condition from Theorem 2.28

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad \forall x \in X$$

and let $(x_k)_{k\in N} \subset X$ a sequence of points derived by solving $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ with $(M_k, \Delta t_k, \epsilon_k)$, where $(M_k)_k \subset \mathbb{N}_+$, $(\Delta t_k)_k \subset \mathbb{R}_+$ and $(\epsilon_k)_k \subset \mathbb{R}_+$ are sequences with $\mathrm{M}_k \to \infty$, $\Delta t_k \to 0$, $\epsilon_k \to 0$. Each of these points fulfills the variational inequality following Theorem 2.28. Due to computational error, i.e. running an optimization algorithm on the computer only leads to an approximation of the true minimizer, this optimality condition is only satisfied approximately. Thus, setting $(\gamma_k)_k \subset \mathbb{R}_+$, $\gamma_k \to 0$ a sequence of error tolerances one has

$$\nabla f_{M_k,\Delta t_k,\epsilon_k}(x_k)^T (x - x_k) \geq -\gamma_k, \quad \forall x \in X, \quad k = 1, \ldots \quad . \tag{4.25}$$

Note that, since $f_{M_k,\Delta t_k,\epsilon_k}$ depends on the random Brownian increments ΔW^m_n, the iterates x_k are also random variables. However, this dependence of x_k on the random samples is not expressed explicitly to facilitate notation. As X is convex

and compact, there exists a subsequence $(x_{k_l})_l$ which has a limit point x^* in X. The following theorem shows that x^* almost surely is a critical point of first order for f.

Theorem 4.22. *Let $(M_k)_k \subset \mathbb{N}_+$, $(\Delta t_k)_k \subset \mathbb{R}_+$, $(\epsilon_k)_k \subset \mathbb{R}_+$ and $(\gamma_k)_k \subset \mathbb{R}_+$ be given sequences with $M_k \to \infty$, $\Delta t_k \to 0$, $\epsilon_k \to 0$ and $\gamma_k \to 0$ and assume that $(x_k)_{k\in N} \subset X$ is a sequence of points satisfying* (4.25)*. Suppose Assumptions* (A.1)-(A.9) *hold true and additionally $\{\pi_\epsilon(S_{T_i}(x,\omega) - K_i), x \in X\}$ as well as $\{\frac{\partial}{\partial x_p}\pi_\epsilon(S_{T_i}(x,\omega) - K_i), x \in X\}$ are dominated integrable for $i = 1, ..., I$ and $p = 1, ..., P$. If the Lipschitz constants are uniformly bounded in $\epsilon \in \mathbb{R}_+$, $x \in \mathbb{R}^P$ and $y \in \mathbb{R}^L$, every limit point $x^* \in X$ of $(x_k)_k$ satisfies the first order optimality condition*

$$\nabla f(x^*)^T(x - x^*) \geq 0 \quad \forall\, x \in X$$

for problem (P).

Proof. Let x^* be a limit point of $(x_k)_k$ and assume that $(x_{k_l})_{l\in N}$ is a subsequence converging to x^*. The existence of such a limit point is ensured by Assumption (A.1), namely the compactness of X. In the following, to facilitate notation, it will not be distinguished between x_k and the corresponding subsequence converging to x^*.

As a first step Corollary 4.20 shows, that $f_{M,\Delta t,\epsilon}$ converges uniformly to f on X (a.s.) . To be more precise one obtains

$$\lim_{k\to\infty} \sup_{x\in X} |f_{M_k,\Delta t_k,\epsilon_k}(x) - f(x)| = 0 \quad (a.s.).$$

In analogy Corollary 4.21 leads to

$$\lim_{k\to\infty} \sup_{x\in X} \left|\left|\nabla f^i_{M_k,\Delta t_k,\epsilon_k}(x) - \nabla f(x)\right|\right|_2 = 0 \quad (a.s.).$$

Hence, for all $\delta_1 > 0$ one can choose $K_1 > 0$ such that

$$||\nabla f_{M_k,\Delta t_k,\epsilon_k}(x_k) - \nabla f(x_k)||_2 < \delta_1 \quad \forall\, k > K_1 \quad (a.s.). \tag{4.26}$$

Furthermore, the continuity of ∇f implies that

$$\forall\, \delta_2 > 0\, \exists\, K_2 > 0 \text{ such that } ||\nabla f(x_k) - \nabla f(x^*)||_2 < \delta_2 \quad \forall\, k > K_2. \tag{4.27}$$

Thus, based on (4.26) and (4.27), one obtains for all $k > K := \max(K_1, K_2)$

$$\begin{aligned}
&||\nabla f_{M_k,\Delta t_k,\epsilon_k}(x_k) - \nabla f(x^*)||_2 \\
&\qquad \leq \|\nabla f_{M_k,\Delta t_k,\epsilon_k}(x_k) - \nabla f(x_k)\|_2 + ||\nabla f(x_k) - \nabla f(x^*)||_2 \\
&\qquad < \delta_1 + \delta_2 =: \delta \quad \forall k > K \quad (a.s.)
\end{aligned}$$

But this means that taking limits ($k \to \infty$) on both sides of the inequality (4.25) provides

$$\nabla f_{M_k,\Delta t_k,\epsilon_k}(x_k)^T (x - x_k) \geq -\gamma_k \quad \forall x \in X \quad (a.s.),$$

which implies that

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad \forall x \in X.$$

Thus, x^* is a first order critical point of f. □

Summarizing, the convergence of a sequence of first order critical points, derived by solving ($\mathrm{P}_{\mathrm{M},\Delta t,\epsilon}$) with the triplet $(M_k, \Delta t_k, \epsilon_k) \in \mathbb{N} \times \mathbb{R}_+ \times \mathbb{R}_+$, to a critical point first order of (P) is ensured under reasonable assumptions like the Lipschitz continuity of the SDE's coefficient functions and their first order derivatives. Section 4.2.1 provides an order of this convergence, namely $\mathcal{O}(1/\sqrt{M} + \sqrt{\Delta t} + \sqrt{\psi(\epsilon)})$. Numerical results will be provided for the example of the Stein-Stein model in chapter 7, which confirm the theoretical results of this chapter. Note that the Heston model (3.2), introduced in the beginning of this thesis, does not fulfill the required assumptions.

Chapter 5

Efficient Calculation of the Objective's Gradient

In any gradient based optimization method the algorithm requires for the computation of the derivative of the residual vector for the solution of the least squares problem. Unless the gradient can be calculated analytically, a simple but expensive way is the use of finite difference approximations. Especially in cases where the calculation of the objective function is expensive, as in a Monte Carlo framework like the one under observation in this work, this method results in a high computational effort. Furthermore, an inadequate choice of the step size for the finite difference quotient can lead to severe problems.

Initially, an overview to finite differences is given in section 5.1. As a first improvement section 5.2 introduces the sensitivity equation. Unfortunately, this approach leads in fact to the exact gradient but suffers from the same computational effort as the finite difference approach. Thus, it will be shown in section 5.3 how the calculation can be sped up with an adjoint method. The fourth part approves this numerically. To round the topic out, alternative approaches like automatic differentiation are briefly explained and discussed in the last part.

5.1 Gradient Calculation and Finite Differences Approximation

As mentioned in section 3.4, the optimization problem ($P_{M,\Delta t,\epsilon}$) is solved with a line-search SQP algorithm. This method is based on first and second order derivative

information of the objective function

$$f_{M,\Delta t,\epsilon}(x) := \sum_{i=1}^{I} \left(C^i_{M,\Delta t,\epsilon}(x) - C^i_{\text{obs}}\right)^2.$$

In Lemma 3.4, $f_{M,\Delta t,\epsilon}$ has already been transformed to the squared 2-norm of a residual vector $R : \mathbb{R}^P \to \mathbb{R}^I$:

$$R(x) = [R_i(x)]_{i=1}^I = \left[C^i_{M,\Delta t,\epsilon}(x) - C^i_{\text{obs}}\right]_{i=1}^I. \tag{5.1}$$

Thus, the objective function can be written as

$$f_{M,\Delta t,\epsilon}(x) = \|R(x)\|_2^2.$$

Defining the Jacobi matrix of the residual vector $J_R : \mathbb{R}^P \to \mathbb{R}^{I\times P}$ as

$$J_R(x) := \left[\frac{\partial}{\partial x_p} R_i(x)\right]_{i,p=1}^{I,P},$$

the gradient of the objective function can be calculated through

$$\nabla f_{M,\Delta t,\epsilon}(x) = 2J_R(x)^T R(x) \tag{5.2}$$

as described in Lemma 3.4. Thus the calculation of $\nabla f_{M,\Delta t,\epsilon}$ has been boiled down to the calculation of the Jacobian J_R. A simple, but expensive way is the use of finite difference approximations.

Lemma 5.1. *Let $R : \mathbb{R}^P \to \mathbb{R}^I$ be the vector valued function*

$$R(x) = \left[e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i}(x) - K_i)\right) - C^i_{\text{obs}}\right]_{i=1}^{I},$$

and e_p denote the p-th unit vector $(0, ..., 0, 1, 0, ..., 0)^T \in \mathbb{R}^P$. Given Assumption (A.3) *and $h > 0$*

$$\left[\frac{\partial}{\partial x_p} R_i(x)\right]_{i,p=1}^{I,P} \approx \left[\frac{R_i(x + he_p) - R_i(x)}{h}\right]_{i,p=1}^{I,P}, \tag{5.3}$$

is an approximation of the true Jacobian with order $\mathcal{O}(h)$.

Proof. A Taylor series expansion (see Theorem 2.25) provides

$$R_i(x + he_p) = R_i(x) + h\frac{\partial}{\partial x_p} R_i(x) + \mathcal{O}(h)$$

and an isolation of $\frac{\partial}{\partial x_p} R_i$ leads to

$$\frac{\partial}{\partial x_p} R_i(x) = \frac{R_i(x + he_p) - R_i(x)}{h} + \mathcal{O}(h).$$

□

Both, the above introduced forward scheme as well as the corresponding backward scheme:

$$\left[\frac{\partial}{\partial x_p} R_i(x)\right]_{i,p=1}^{I,P} \approx \left[\frac{R_i(x) - R_i(x - he_p)}{h}\right]_{i,p=1}^{I,P},$$

overestimate a change of gradient on one side of the point under observation. Therefore the *central* finite difference approximation is often considered:

$$\left[\frac{\partial}{\partial x_p} R_i(x)\right]_{i,p=1}^{I,P} \approx \left[\frac{R_i(x + he_p) - R_i(x - he_p)}{2h}\right]_{i,p=1}^{I,P}.$$

This scheme is more stable compared to the forward or backward scheme, but the computational effort is doubled as $R_i(x)$ is now replaced by $R_i(x - he_p)$. Thus this work focuses on the forward scheme. The computational effort is addressed in the following remark.

Remark 5.2. *As $\nabla f_{M,\Delta t,\epsilon}(x) = 2J_R(x)^T R(x)$ due to Lemma 3.4, the computational effort for $J_R = [\frac{\partial}{\partial x_p} R_i(x)]_{i,p=1}^{I,P}$ instead of $\nabla f_{M,\Delta t,\epsilon}$ is considered. Every single $\frac{\partial}{\partial x_p} R_i(x)$ requires the calculation of $R_i(x)$ and $R_i(x + he_p)$ for $p = 1, ..., P$. Keeping $R_i(x)$ in memory, each of these $P + 1$ residual vectors itself requires the solution of the underlying stochastic differential equation which results in L multiplications for $a_\epsilon(x, y_n^m(x))\Delta t_n$ and L^2 multiplications for $b_\epsilon(x, y_n^m(x))\Delta W_n^m$ for each of the M simulations and each of the N time steps. As it could be motivated at the end of section 3.4 that solving the SDE only once provides all option prices, the resulting computational complexity of the forward finite difference scheme is of order $\mathcal{O}((P + 1)MN(L + L^2))$.*

Consequently, the computation time for the finite difference approximation scales linearly in the number of parameters. This will also be shown in the numerical results in section 5.4.

Besides this high computational effort, the correct choice of the step size $h > 0$ can be a critical issue, as already mentioned in the very beginning. The rate of convergence $\mathcal{O}(h)$ suggests to choose h as small as possible. On the other hand, a widely known optimal choice for h is the square root of the machine accuracy divided by the second derivative (see e.g. Nocedal and Wright [1999], pp. 166 ff.), where the machine accuracy is 10^{-15} on a double precision system. However,

in practice both approaches may lead to severe problems. Table 5.1 shows results for two finite differences based gradient approximations in direction of the mean reversion speed parameter θ in the Heston model (3.2) for a varying set of parameter values. Following the rule for the optimal choice of h explained above would lead

h	Finite Differences	h	Finite Differences
1.0e-01	4.5702644e-01	1.0e-01	1.2364214e+00
1.0e-02	2.1809812e-01	1.0e-02	1.6345058e+00
1.0e-03	-8.6920392e-01	1.0e-03	2.1654201e+00
1.0e-04	-2.6063369e+00	1.0e-04	7.2584657e-01
1.0e-05	3.0741435e+01	1.0e-05	-1.2348351e+01
1.0e-06	1.8494888e+02	1.0e-06	-3.3469893e+01
1.0e-07	4.4476847e+02	1.0e-07	3.0128138e+02
1.0e-08	-1.1492117e+02	1.0e-08	2.6195569e+03
1.0e-09	-4.5257191e+02	1.0e-09	1.5295545e+03
1.0e-10	-2.9722253e+03	1.0e-10	-1.1503288e+04
1.0e-11	-4.2431821e+02	1.0e-11	-2.8594760e+04
8.2e-12 *	-3.5369871e+02	1.0e-12	-3.5064154e+03
1.0e-12	-1.8970325e+01	8.0e-13 *	-2.6152544e+03
1.0e-13	3.6540770e+01	1.0e-13	7.9288603e+02
1.0e-14	7.0637940e+01	1.0e-14	1.2934459e+03
Exact	3.6826311e+01	Exact	1.3332680e+03

Table 5.1: Derivative evaluation via finite differences for the volatility in the Heston model with 10,000 simulations for varying sets of parameter values.

to $h = 8.2 \times 10^{-12}$ for the test case on the left side and to $h = 8.0 \times 10^{-13}$ on the right side. Both choices lead to derivative values which are totally inaccurate in comparison to the exact derivative. Additionally, it is also not possible to find a preliminary fixed h which would lead to a good approximation for both test cases.

These results illustrate on the one hand the problem of finding an optimal step size h. On the other hand they additionally exhibit the enormous fluctuation range of a finite differences based gradient approximation. The proposed convergence behavior from Lemma 5.1 seems to be restricted to a small interval of step size values. In fact, such severe instabilities may lead to a breakdown of the calibration algorithm before convergence has been reached, especially when solving ill-posed problems. A first improvement of the gradient calculation in this manner is the *sensitivity equation*.

5.2 Exact Derivative via the Sensitivity Equation

One of the mentioned two crucial disadvantages of the finite difference method introduced above, is that it is only an approximation. Furthermore the quality of the approximation is uncertain and like shown above even severe instabilities might occur. Thus, an exact method to calculate the derivatives is desired.

Reconsider, that the derivatives of $R_i(x)$ in direction x_p for $i = 1, ..., I$ and $p = 1, ..., P$ are required. By definition it holds true that

$$\frac{\partial}{\partial x_p} R_i(x) = e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \pi'_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}(x). \tag{5.4}$$

As π'_ϵ can be calculated analytically:

$$\pi'_\epsilon(x) := \begin{cases} 0 & , \ x \leq -\epsilon \\ -\frac{1}{4\epsilon^3}x^3 + \frac{3}{4\epsilon}x + \frac{1}{2} & , \ -\epsilon < x < \epsilon \\ 1 & , \ x \geq \epsilon, \end{cases} \tag{5.5}$$

only $\frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}(x)$, i.e. the derivative of the SDE solution with respect to the parameters, is required. These expressions are called sensitivities. The corresponding stochastic differential equation providing $\frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}$ as its solution is consequently denoted as *sensitivity equation*:

Definition 5.3. (Sensitivity Equation) *Consider the EMS discretized stochastic differential equation*

$$\begin{aligned} &y^m_{n+1}(x) = y^m_n(x) + a_\epsilon(x, y^m_n(x))\Delta t + b_\epsilon(x, y^m_n(x))\Delta W^m_n, \\ &y^m_0 = Y_0, \quad n = 0, ..., N-1, \quad m = 1, ..., M, \quad N := \max_{i=1,...,I} N_i. \end{aligned}$$

Taking derivatives with respect to x in direction Δx in this SDE leads to the sensitivity equation

$$\begin{aligned} \eta^m_{n+1}(x) = \eta^m_n(x) &+ \left[\frac{\partial}{\partial y} a_\epsilon(x, y^m_n(x))\eta^m_n(x) + \frac{\partial}{\partial x} a_\epsilon(x, y^m_n(x))\Delta x\right] \Delta t_n \\ &+ \left[\frac{\partial}{\partial y}\left(b_\epsilon(x, y^m_n(x))\Delta W^m_n\right)\eta^m_n(x) + \frac{\partial}{\partial x}\left(b_\epsilon(x, y^m_n(x))\Delta W^m_n\right)\Delta x\right] \\ \eta^m_0 = 0, \quad n = 0, ..., N-1, \quad m = 1, ..., M, \quad N := \max_{i=1,...,I} N_i \end{aligned} \tag{5.6}$$

where $\eta^m_n \in \mathbb{R}^L$ is defined as

$$\begin{aligned} \eta^m_n(x) &:= \left[\xi^m_n(x), \eta^{2,m}_n(x), ..., \eta^{L,m}_n(x)\right]^T \\ &:= \left[\frac{\partial}{\partial x} s^m_n(x)\Delta x, \frac{\partial}{\partial x} y^{2,m}_n(x)\Delta x, ..., \frac{\partial}{\partial x} y^{L,m}_n(x)\Delta x\right]^T \end{aligned}$$

in analogy to the definition of y^m_n and the quantities $\frac{\partial}{\partial x} a_\epsilon$, $\frac{\partial}{\partial x}\left(b_\epsilon \Delta W\right) \in \mathbb{R}^{L\times P}$ as well as $\frac{\partial}{\partial y} a_\epsilon$, $\frac{\partial}{\partial y}\left(b_\epsilon \Delta W\right) \in \mathbb{R}^{L\times L}$ denote the Jacobians of a_ϵ, $b_\epsilon \Delta W$ with respect to the variables x and y.

Obviously, (5.4) in combination with the sensitivity equation (5.6) provides the Jacobi matrix:

Theorem 5.4. *Let $R : \mathbb{R}^P \to \mathbb{R}^I$ be the vector valued function*

$$R(x) = \left[e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon(s^m_{N_i}(x) - K_i) \right) - C^i_{\text{obs}} \right]_{i=1}^{I}$$

with

$$y^m_{n+1}(x) = y^m_n(x) + a_\epsilon(x, y^m_n(x))\Delta t + b_\epsilon(x, y^m_n(x))\Delta W^m_n,$$
$$y^m_0 = Y_0, \quad n = 0, ..., N-1, \quad m = 1, ..., M, \quad N := \max_{i=1,...,I} N_i.$$

Given Assumption (A.3)*, the derivative of R_i can be computed via*

$$\frac{\partial}{\partial x_p} R_i(x) = e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \pi'_\epsilon(s^m_{N_i,\epsilon}(x) - K_i)\xi^m_{N_i,\epsilon}(x)$$

with $\xi^m_{N_i,\epsilon}(x)$ the first component of the solution of (5.6)

Proof. The derivative of $R_i(x)$ in direction of an increment Δx can be expressed as $\frac{\partial}{\partial x} R_i(x)\Delta x$ where particularly for $\Delta x = e_p$ with $e_p \in \mathbb{R}^P$ the p-th unit vector it holds true that

$$\frac{\partial}{\partial x} R_i(x) e_p = \frac{\partial}{\partial x_p} R_i(x).$$

Thus, solving (5.6) with $\Delta x = e_p$, $p = 1, ..., P$ provides $\frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}$ and together with (5.4) the gradient. □

By definition, Theorem 5.4 provides the exact gradient which is a clear advantage in comparison to the finite difference method. Unfortunately, the computational effort of both methods is almost identical:

Remark 5.5. *Following Theorem 5.4 the calculation of*

$$\frac{\partial}{\partial x_p} R_i(x) = e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \pi'_\epsilon(s^m_{N_i,\epsilon}(x) - K_i)\xi^m_{N_i,\epsilon}(x)$$

requires the solution of the sensitivity equation (5.6) *for every $\Delta x = e_p$, $p = 1, ..., P$. $\frac{\partial}{\partial y} a_\epsilon(x, y^m_n(x)) \in \mathbb{R}^{L \times L}$ and $\eta^m_n \in \mathbb{R}^L$ lead to L^2 multiplications to build their product. The same holds for $\frac{\partial}{\partial y} \left(b_\epsilon(x, y^m_n(x))\Delta W^m_n \right) \eta^m_n(x)$. As Δx is chosen as e_p, the results of $\frac{\partial}{\partial x} a_\epsilon(x, y^m_n(x))\Delta x$ as well as $\frac{\partial}{\partial x}(b_\epsilon(x, y^m_n(x))\Delta W^m_n)\Delta x$ are simply the p-th columns of $\frac{\partial}{\partial x} a_\epsilon(x, y^m_n(x)) \in \mathbb{R}^{L \times P}$ respectively $\frac{\partial}{\partial x}(b_\epsilon(x, y^m_n(x))\Delta W^m_n) \in \mathbb{R}^{L \times P}$. Finally $\frac{\partial}{\partial y} a_\epsilon(x, y^m_n(x))\eta^m_n \in \mathbb{R}^L$ and $\frac{\partial}{\partial x} a_\epsilon(x, y^m_n(x))\Delta x \in \mathbb{R}^L$ are multiplied with the scalar Δt_n leading to L multiplications for each. Thus the total complexity of solving the sensitivity equation is of order $\mathcal{O}(L^2 + L)$. This has to be done P times, i.e. for each parameter e_p. Again, like in Remark 5.2,* (5.6) *can be solved*

in one sweep. Neglecting the evaluation of π' for (5.4) *leads consequently to a computational effort of order $\mathcal{O}(PMN(L^2+L))$ which scales linearly in P similar to the finite difference scheme.*

Concluding, it would be desirable to have an exact method which produces a significantly less computational effort. This will be provided by the adjoint equation in the next section.

5.3 Adjoint Equation

An efficient method to calculate the gradient which is well known from optimization with partial differential equations is the *adjoint equation*. It has been introduced into finance literature by Giles and Glasserman in their paper *Smoking Adjoints* (Giles and Glasserman [2006]). In this, the adjoint equation has been used to calculate sensitivities in a Libor market model. Additionally Giles [2007] uses the adjoint in an automatic differentiation framework (see also section 5.5.3). In the following theorem, the adjoint equation will be derived for the optimization problem $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ and consequently applied in a calibration framework with stochastic differential equations.

Theorem 5.6. *Let $R : \mathbb{R}^P \to \mathbb{R}^I$ be the vector valued function*

$$R(x) = \left[e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon(s_{N_i}^m(x) - K_i) \right) - C_{\text{obs}}^i \right]_{i=1}^{I}$$

with

$$\begin{aligned} &y_{n+1}^m(x) = y_n^m(x) + a_\epsilon(x, y_n^m(x))\Delta t + b_\epsilon(x, y_n^m(x))\Delta W_n^m, \\ &y_0^m = Y_0, \quad n = 0, ..., N-1, \quad m = 1, ..., M, \quad N := \max_{i=1,...,I} N_i. \end{aligned}$$

Given Assumption (A.3) *the derivative of R_i can be computed via*

$$R_i'(x) = \frac{e^{-rT_i}}{M} \sum_{m=1}^{M} \sum_{n=0}^{N_i-1} (\lambda_{n+1}^{m,i})^T \left[\frac{\partial}{\partial x} a_\epsilon(x, y_n^m)\Delta t_n + \frac{\partial}{\partial x} \left(b_\epsilon(x, y_n^m)\Delta W_n^m \right) \right]$$

where $\lambda_n^{m,i} \in \mathbb{R}^L$ results from the adjoint equation

$$\begin{aligned} &\lambda_n^{m,i} = \left[I + \tfrac{\partial}{\partial y} a_\epsilon(x, y_n^m)\Delta t_n + \tfrac{\partial}{\partial y} \left(b_\epsilon(x, y_n^m)\Delta W_n^m \right) \right]^T \lambda_{n+1}^{m,i}, \\ &n = N_i - 1, N_i - 2, ..., 1, \quad m = 1, ..., M, \\ &\lambda_{N_i}^{m,i} = \left[\left(\pi_\epsilon'(s_{N_i}^m(x) - K) \right), 0, ..., 0 \right] \in \mathbb{R}^L. \end{aligned} \tag{5.7}$$

Proof. For the derivation of the adjoint equation, each of the $M \times N$ recursive sensitivity equations (5.6) is initially multiplied with vectors $\lambda_{n+1}^m \in \mathbb{R}^L$, which will be determined later. Summarizing over all time steps $n = 0, ..., N-1$ leads to

$$\begin{aligned} &\sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \eta_{n+1}^m \\ &\quad - \sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \left[I + \tfrac{\partial}{\partial y} a_\epsilon(x, y_n^m) \Delta t_n + \tfrac{\partial}{\partial y} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right] \eta_n^m \\ &= \sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \left[\tfrac{\partial}{\partial x} a_\epsilon(x, y_n^m) \Delta t_n + \tfrac{\partial}{\partial x} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right] \Delta x \\ &\eta_0^m = 0, \quad m = 1, ..., M. \end{aligned} \tag{5.8}$$

Since $\eta_0^m = 0$, the second summation on the left hand side can start at $n = 1$, which is also convenient for the first sum, since an index shift yields

$$\sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \eta_{n+1}^m = \sum_{n=1}^{N} (\lambda_n^m)^T \eta_n^m = \sum_{n=1}^{N-1} (\lambda_n^m)^T \eta_n^m + (\lambda_N^m)^T \eta_N^m.$$

If one uses this equality and merges the two first sums of (5.8) into one, one obtains for (5.8) the equation

$$\begin{aligned} &\sum_{n=1}^{N-1} \left[(\lambda_n^m)^T - (\lambda_{n+1}^m)^T \left(I + \tfrac{\partial}{\partial y} a_\epsilon(x, y_n^m) \Delta t_n + \tfrac{\partial}{\partial y} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right) \right] \eta_n^m \\ &+ (\lambda_N^m)^T \eta_N^m = \sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \left[\tfrac{\partial}{\partial x} a_\epsilon(x, y_n^m) \Delta t_n + \tfrac{\partial}{\partial x} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right] \Delta x \\ &\eta_0^m = 0, \quad m = 1, ..., M, \quad N := \max_{i=1,...,I} N_i. \end{aligned} \tag{5.9}$$

It can be easily seen, that, if it is required that the vectors λ_n^m satisfy recursively the relation

$$\begin{aligned} &\lambda_n^m = \left[I + \tfrac{\partial}{\partial y} a_\epsilon(x, y_n^m) \Delta t_n + \tfrac{\partial}{\partial y} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right]^T \lambda_{n+1}^m \\ &n = N-1, ..., 1, \quad m = 1, ..., M \end{aligned} \tag{5.10}$$

then the first term in brackets in (5.9) vanishes and one obtains

$$(\lambda_N^m)^T \eta_N^m = \sum_{n=0}^{N-1} (\lambda_{n+1}^m)^T \left[\frac{\partial}{\partial x} a_\epsilon(x, y_n^m) \Delta t_n + \frac{\partial}{\partial x} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right] \Delta x. \tag{5.11}$$

Note that in the so-called *adjoint equation* (5.10) the recursion for λ_n^m runs backwards. Hence a final condition for the adjoint variable has to be specified, where one is free to choose this. If one recalls the form of the derivative (5.4), one realizes

that the following choice

$$\lambda_{N_i}^{m,i} = \left[\left(\pi_\epsilon'(s_{N_i}(x) - K)\right), 0, ..., 0\right] \in \mathbb{R}^L$$

substituted in (5.4) together with (5.11) yields the expression

$$R_i'(x)\Delta x = e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon'(s_{N_i}^m(x) - K_i)\right)\xi_{N_i}^m = e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}(\lambda_{N_i}^{m,i})^T\eta_{N_i}^m.$$

Replacing the expression for $\lambda_{N_i}^{m,i}\eta_{N_i}^m$ from equation (5.11) leads to

$$\begin{aligned} R_i'(x)\Delta x &= e^{-rT_i}\tfrac{1}{M}\sum_{m=1}^{M}\sum_{n=0}^{N_i-1}(\lambda_{n+1}^{m,i})^T\left[\tfrac{\partial}{\partial x}a_\epsilon(x, y_n^m)\Delta t_n\right. \\ &+ \left.\tfrac{\partial}{\partial x}\left(b_x(x, y_n^m)\Delta W_n^m\right)\right]\Delta x \end{aligned}$$

which proves the statement. □

The computational effort is as follows.

Remark 5.7. *As shown in Remark 5.5, one adjoint step has the complexity $\mathcal{O}(L^2)$ multiplications, since $\lambda_{n+1}^{m,i} \in \mathbb{R}^L$. A closer look reveals, that the adjoint has to be resolved for every maturity due to the final condition. Following the idea on page 36, namely to simulate the SDE only once for all maturities, one is able to exploit the fact that some maturities T_i and hence N_i may be identical and then so are the adjoint values. Thus, let*

$$\mathcal{I} := \{i : N_i \neq N_j \,\forall\, j = 1, ..., i-1\}$$

be a set of different maturities indices. The computational complexity of (5.7) *is thus $\mathcal{O}(MN|\mathcal{I}|L^2)$. The cost for*

$$R_i'(x) = \frac{e^{-rT_i}}{M}\sum_{m=1}^{M}\sum_{n=0}^{N_i-1}(\lambda_{n+1}^{m,i})^T\left[\frac{\partial}{\partial x}a_\epsilon(x, y_n^m)\Delta t_n + \frac{\partial}{\partial x}\left(b_\epsilon(x, y_n^m)\Delta W_n^m\right)\right] \quad (5.12)$$

is $MN|\mathcal{I}|PL$. Summarizing, the adjoint method yields a computational complexity of $\mathcal{O}(MN|\mathcal{I}|(L^2 + PL))$ which does not scale linearly in P, like the finite difference scheme and the sensitivity equation do, but still scales strongly in P due to the part in brackets in (5.12).

Following Theorem 5.6 and Remark 5.7 one would have to solve the adjoint equation backwards for every varying maturity and the computational effort still scales in P albeit not linearly. However, there is a way to boil down the computational effort even further. The first remark aims at the backward solves.

Remark 5.8. *Consider again the adjoint equation* (5.7) *for the i-th option with maturity T_i and strike K_i:*

$$\begin{aligned}
&\lambda_n^{m,i} = \left[I + \tfrac{\partial}{\partial y} a_\epsilon(x, y_n^m)\Delta t_n + \tfrac{\partial}{\partial y}\left(b_\epsilon(x, y_n^m)\Delta W_n^m\right)\right]^T \lambda_{n+1}^{m,i} \\
&\lambda_{N_i}^{m,i} = \left[\left(\pi_\epsilon'(s_{N_i}^m(x) - K)\right), 0, ..., 0\right] \in \mathbb{R}^L \\
&n = N_i - 1, N_i - 2, ..., 1 \quad i = 1, ..., I.
\end{aligned}$$

As it will be discussed subsequently, one can calculate the adjoints pathwise, so that one may leave out the upper index m for the sake of readability. If one sets for abbreviation

$$G_n := \left[I + \frac{\partial}{\partial y} a_\epsilon(x, y_n)\Delta t_n + \frac{\partial}{\partial y}\left(b_\epsilon(x, y_n)\Delta W_n\right)\right]^T,$$

and

$$\Pi_i' = \pi_\epsilon'(s_{N_i}(x) - K)$$

the adjoint can be written as

$$\begin{aligned}
&\lambda_n^i = G_n \lambda_{n+1}^i \\
&\lambda_{N_i}^i = \Pi_i' e_1^T \\
&n = N_i - 1, N_i - 2, ..., 1,\ i = 1, ..., I.
\end{aligned}$$

Consider again the set of different maturities indices $\mathcal{I}$. Instead of solving this adjoint equation $|\mathcal{I}|$ times, the following is suggested. Let $\mu_n^{i,l} \in \mathbb{R}^L$ be the solution of

$$\begin{aligned}
&\mu_{N_{i-1}}^l = G_{N_{i-1}} ... G_{N_i - 2} G_{N_i - 1} \mu_{N_i}^l \\
&\mu_{N_i}^l = e_l^T \\
&n = N_i - 1, N_i - 2, ..., N_{i-1}
\end{aligned}$$

which is the same recursion as the adjoint, but starting at N_i with the l-th unit vector $\mu_{N_i}^l = e_l^T$ and stopping at the next lower maturity N_{i-1}. This leads to a sequence of basis solutions $\mu_n^i \in \mathbb{R}^L$ defined through

$$\mu_n^l := \left[\mu_0^{l,1}, ..., \mu_{N_1}^{l,1}, ..., \mu_{N_{|\mathcal{I}|-1}+1}^{l,|\mathcal{I}|}, ..., \mu_{N_{|\mathcal{I}|}}^{l,|\mathcal{I}|}\right]^T$$

for $l = 1, ..., L$. All required adjoint variables can now be calculated by simply building linear combinations with these basis solutions as it holds for a given maturity N_i, that

$$\begin{aligned}
&\lambda_n^i = (\lambda_{n+1}^i)^T \mu_n^i \\
&n = 0, ..., N_i - 1 \\
&\lambda_{N_i}^i = \Pi_i' e_1^T
\end{aligned}$$

Note that one has here L instead of $|\mathcal{I}|$ solves of the adjoint equation, where usually

the number of maturities $|\mathcal{I}|$ strongly dominates the dimension of the SDE.

Additionally one can exploit possible structure of the derivatives with respect to x in

$$R_i'(x) = \frac{e^{-rT_i}}{M} \sum_{m=1}^{M} \sum_{n=0}^{N_i-1} (\lambda_{n+1}^{m,i})^T \left[\frac{\partial}{\partial x} a_\epsilon(x, y_n^m) \Delta t_n + \frac{\partial}{\partial x} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) \right]$$

to further decrease the scaling of the computational effort with P, as shown in the next Remark.

Remark 5.9. *Remark 5.7 states that the computational effort for the term in brackets above is $\mathcal{O}(PL)$. This effort can even further be reduced. Consider the case that $x_p = [x_p^1, ..., x_p^B]^T \in \mathbb{R}^B$ with*

$$x_p(t) = x_p^b; \quad t \in [T_{b-1}, T_b); \quad b = 1, ...B \quad \text{and} \quad x(T) = x_p^B.$$

In particular this means that x_p is chosen piecewise constant on B time intervals $[T_{b-1}, T_b)$ for $b = 1, ..., B$. Consequently it holds for the partial derivative with respect to the p-th parameter on the b-th subinterval that

$$\frac{\partial}{\partial x_p^b} a_\epsilon(x, y_n^m) \Delta t_n = \frac{\partial}{\partial x_p^b} \left(b_\epsilon(x, y_n^m) \Delta W_n^m \right) = 0, \ \forall n \text{ with } \tau_n < t_{b-1} \text{ or } \tau_n \geq t_b.$$

Thus on every subinterval $[\tau_{b-1}, ..., _b)$ there exist exactly one $b \in \{1, ..., B\}$ with non vanishing corresponding derivative of the coefficients, i.e. $\frac{\partial}{\partial x_p^b} a_\epsilon(x, y_n^m)$ and $\frac{\partial}{\partial x_p^b}(b_\epsilon(x, y_n^m) \Delta W_n^m)$ have local support. Increasing the number of intervals does consequently not increase the required effort to evaluate $R_i'(x)$.

The following remark deals with the resulting computational complexity

Remark 5.10. *Following Remark 5.7, the computational effort of the adjoint equation is of order $\mathcal{O}(MN|\mathcal{I}|L^2)$. As one now has L instead of $|\mathcal{I}|$ solves, the total effort of the adjoint method is reduced to $\mathcal{O}(MNL(L^2 + PL))$ which should be significantly less, as the number of maturities can be expected to dominate the dimension of the SDE. Furthermore, if one considers in summary P parameters which are chosen piecewise constant on B subintervals resulting from Q different parameter types, i.e. $P = BQ$, the complexity for the adjoint method is $\mathcal{O}(MNL(L^2 + BQL))$. As it has been shown above, that B can be erased from the formula, the effort in summary is $\mathcal{O}(MNL(L^2 + QL))$. Compared to the finite difference scheme the ratio is*

$$\frac{\text{Finite Differences}}{\text{Adjoint Equation}} = \frac{(BQ + 1)MN(L^2 + L)}{MNL(L^2 + QL)}.$$

Omitting the 1 in the first term in brackets in the enumerator and adding a Q to the

right term in brackets in the denominator helps to estimate this term from below with

$$\frac{(BQ+1)MN(L^2+L)}{MNL(L^2+QL)} \geq \frac{BQMN(L^2+L)}{MNLQ(L^2+L)} = B.$$

Thus the number of subintervals is a lower bound for the speed up, provided by the adjoint method in comparison to the finite differences approximation.

Note that the structure of this derivative calculation allows to calculate the adjoints pathwise, meaning that one can calculate the values for y_n^m for fixed m forward and directly afterwards the values for λ_n^m backwards. Hence, this pathwise structure makes it easy to store the Brownian increments ΔW_n^m during the forward sweep and to reuse them immediately afterwards in the backwards computation. In comparison to finite differences this is an additional advantage as it is usually not possible to store the Brownian increments ΔW_n^m for all m and n. See also section 6.3 for a more detailed discussion.

5.4 Numerical Results

To fully assess the potential speedup of the introduced method, time dependent Heston parameters $\kappa_t, \theta_t, \sigma_t$ and ρ_t for $0 \leq t \leq T$ are introduced. In particular piecewise constant parameters on $[t_{b-1}, t_b)$, $b = 1, ..., B$ are chosen, i.e.

$$\begin{aligned}
\kappa_t = \kappa_b \quad ; \quad t \in [t_{b-1}, t_b) \quad ; \quad b = 1, ...B \quad \text{and} \quad \kappa(T) = \kappa_B, \\
\theta_t = \theta_b \quad ; \quad t \in [t_{b-1}, t_b) \quad ; \quad b = 1, ...B \quad \text{and} \quad \theta(T) = \theta_B, \\
\sigma_t = \sigma_b \quad ; \quad t \in [t_{b-1}, t_b) \quad ; \quad b = 1, ...B \quad \text{and} \quad \sigma(T) = \sigma_B, \\
\rho_t = \rho_b \quad ; \quad t \in [t_{b-1}, t_b) \quad ; \quad b = 1, ...B \quad \text{and} \quad \rho(T) = \rho_B,
\end{aligned} \tag{5.13}$$

where $0 = t_0 < t_1 < ... < t_B = T$ is a suitable discretization of the time interval $[0, T]$ into B subintervals. In the examples below the points $t_1, ..., t_B$ will be chosen as maturities T_i of the options listed in Table 7.1. For the time-dependent parameters the notation of a vector $x \in \mathbb{R}^P$ can be retained by arranging the elements of x in the following way

$$x = (v_0, \kappa_1, ..., \kappa_B, \theta_1, ..., \theta_B, \sigma_1, ..., \sigma_B, \rho_1, ..., \rho_B)^T \in \mathbb{R}^P. \tag{5.14}$$

This only changes the calculation of the adjoint equation slightly in that one has to replace the previously constant x_i by its corresponding value on the subinterval.

Table 5.2 shows the computing time for one gradient evaluation via adjoint equation compared to the finite differences scheme for varying number of subintervals B and thus varying number of parameters P. As it could have been expected, the computing time for the finite-difference based gradient evaluation increases linearly whereas the adjoint equation almost stays constant. Thus, the ratio of these two

B	P	Fin. Diff.	Adjoint	Ratio
1	5	15	10	1.5
2	9	25	11	2.3
3	13	35	11	3.2
4	17	46	12	3.8
5	21	56	12	4.7
6	25	66	13	5.1
7	29	75	14	5.4
8	33	85	14	6.1
9	37	96	15	6.4
10	41	107	15	7.1

Table 5.2: Computing time in seconds for one gradient evaluation via adjoint equation compared to the finite differences scheme for B subintervals or P parameters with $M = 100,000$, $\Delta t = 5\times10^{-2}$ and $\epsilon = 3.1\times10^{-3}$.

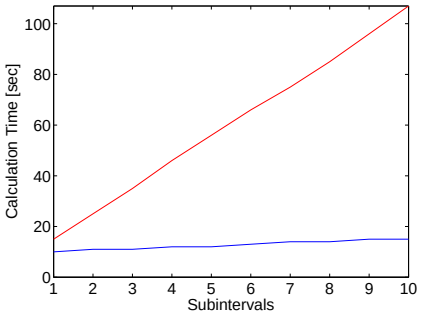

Figure 5.1: Computing time in seconds for one gradient evaluation via adjoint equation (blue line) compared to the finite differences scheme (red line) with $M = 100,000$, $\Delta t = 5\times10^{-2}$ and $\epsilon = 3.1\times10^{-3}$.

increases at an almost linear rate. On 10 subintervals and thus with 41 parameters, the adjoint is 7.1 times faster, than the finite difference scheme. Figure 5.1 illustrates this behavior graphically.

5.5 Alternative Approaches

The above introduced approaches to derive the objectives gradient are certainly not the only existing techniques. Though gradient calculation is certainly a broad topic and though it is not the goal of this work, a brief overview to other approaches including a short discussion on there applicability is given in the following.

5.5.1 Likelihood Ratio Method

The *Likelihood Ratio Method* is based on differentiating the probability density defined by the model for the underlying stock dynamics. An introduction to this method is for instance given in Broadie and Glasserman [1996] or Glasserman [2003] pp. 401 ff. Consider the following example from Broadie and Glasserman [1996] pp. 271-272.

Example 5.11. (Black-Scholes Vega) *Let*

$$C = e^{-rT} E\left(\max(S_T - K, 0)\right) \tag{5.15}$$

be the price of a European Call Option with

$$dS_t = (r - \delta)S_t dt + \sigma S_t dW_t$$

where r is the risk-free rate, δ the dividend yield and σ the volatility. The pricing formula (5.15) *can also be written as*

$$C = e^{-rT} \int_0^\infty \max(x - K, 0) g(x) dx \tag{5.16}$$

where $g(x)$ is the probability density of S_T. If the goal for example is to calculate the Black-Scholes Vega $\partial C/\partial\sigma$, i.e. the derivative of the call price C with respect to the volatility σ, one can under some standard smoothness assumptions interchange the derivative and the integral. Consequently (5.16) *leads to*

$$\frac{\partial C}{\partial \sigma} = e^{-rT} \int_0^\infty \max(x - K, 0) \frac{\partial g(x)}{\partial \sigma} dx.$$

Making use of the fact that $\partial \ln(g) = \frac{\partial g}{g}$ yields

$$\begin{aligned} \frac{\partial C}{\partial \sigma} &= e^{-rT} \int_0^\infty \max(x - K, 0) \frac{\partial \ln(g(x))}{\partial \sigma} g(x) dx \\ &= e^{-rT} E\left(\max(S_T - K, 0) \frac{\partial \ln(g(S_T))}{\partial \sigma} \right). \end{aligned} \tag{5.17}$$

As it is well known that the probability density of S_t is given by

$$\begin{aligned} g(x) &= \frac{1}{x\sigma\sqrt{2T\pi}} e^{-\frac{1}{2}d(x)^2} \\ d(x) &= \frac{\ln(x/S_0) - (r - \delta - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \end{aligned}$$

the derivative of $ln(g(x))$ with respect to σ can easily be calculated. Substituting this result in (5.17) *then leads to the exact Black-Scholes Vega, which only depends on the simulated value for S_T.*

Note that the likelihood ratio method just like the sensitivity or the adjoint equation leads to the exact derivative. Nevertheless, the probability density of the model dynamics is a crucial issue. Unfortunately, this density is not known for many financial market models, such that this method is only feasible for a few chosen situations.

5.5.2 Direct Pathwise Derivatives

A different approach, which is often introduced together with the Likelihood Ratio Method is the *Pathwise Method* (see for instance Glasserman [2003] pp. 386

ff. or Broadie and Glasserman [1996]). The following example from Broadie and Glasserman [1996] shows the functionality.

Example 5.12. *In the situation of Example 5.11, S_T can be calculated via*

$$S_T = S_0 e^{(r-\delta-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z} \tag{5.18}$$

where Z is a standard normal random variable. Thus one has

$$\frac{\partial S_T}{\partial \sigma} = S_T(-\sigma T + \sqrt{T}Z).$$

A simple transformation of (5.18) *provides*

$$\ln(S_T/S_0) = (r - \delta - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z$$

which leads to

$$\frac{\partial S_T}{\partial \sigma} = \frac{S_T}{\sigma}\left(\ln(S_T/S_0) - (r - \delta + \frac{1}{2}\sigma^2)T\right).$$

The Black-Scholes Vega is defined as

$$\frac{\partial C}{\partial \sigma} = \frac{\partial C}{\partial S_T}\frac{\partial S_T}{\partial \sigma}.$$

Though the maximum function is not differentiable for $S_T = K$, this event has probability zero. Thus

$$\frac{\partial C}{\partial S_T} = e^{-rT}1_{\{S_T>K\}} \quad (a.s.).$$

Consequently the Black-Scholes Vega is

$$\frac{\partial C}{\partial \sigma} = \frac{\partial C}{\partial S_T}\frac{\partial S_T}{\partial \sigma} = e^{-rT}\frac{S_T}{\sigma}\left(\ln(S_T/S_0) - (r - \delta + \frac{1}{2}\sigma^2)T\right)1_{\{S_T>K\}}.$$

Thus, with the simulated value of S_T in memory, the derivative can be calculated.

As this example shows, this method is based on the closed-form solution of the model defining the dynamics of the underlying. In absence of such a solution formula, the pathwise method leads to the sensitivity equation, introduced in Section 5.2.

5.5.3 Automatic Differentiation

A totally different but evolving approach is *automatic differentiation*, sometimes also called *algorithmic differentiation*. For a brief explanation consider the following

simple example function $f : \mathbb{R}^2 \to \mathbb{R}$:

$$f(x_1, x_2) = x_1 x_2 + \sin(x_1). \tag{5.19}$$

As a first step one defines the real valued auxiliary variables x_3, x_4, x_5 in the following way:

$$\begin{aligned} x_3 &= x_1 x_2 \\ x_4 &= \sin(x_1) \\ x_5 &= x_3 + x_4. \end{aligned}$$

The derivatives with respect to x_1 and x_2 can now easily be calculated

$$\begin{aligned} \tfrac{\partial}{\partial x_i} x_3 &= x_1 \tfrac{\partial}{\partial x_i} x_1 + x_2 \tfrac{\partial}{\partial x_i} x_2 \\ \tfrac{\partial}{\partial x_i} x_4 &= \cos(x_1) \tfrac{\partial}{\partial x_i} x_1 \\ \tfrac{\partial}{\partial x_i} x_5 &= \tfrac{\partial}{\partial x_i} x_3 + \tfrac{\partial}{\partial x_i} x_4, \quad i = 1, 2. \end{aligned} \tag{5.20}$$

(5.20) thus provides the gradient of f from (5.19). This is denoted as the *forward mode* of AD. As it requires to solve (5.20) for every component of the vector x, its computational effort behaves asymptotically like finite differences and is thus very large. In this context the *reverse mode* reduces this effort significantly. Let $u = (x_1, x_2)$ the vector of independent variables, $y = (x_3, x_4, x_5)$ the vector of dependent variables and define a function $\phi : \mathbb{R}^5 \to \mathbb{R}^3$ with

$$\phi(u, y) = y(u) = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \phi_3(x_1, x_2) \\ \phi_4(x_1, ..., x_3) \\ \phi_5(x_1, ..., x_4) \end{pmatrix} = \begin{pmatrix} \phi_3(u) \\ \phi_4(u, x_3) \\ \phi_5(u, x_3, x_4) \end{pmatrix} \tag{5.21}$$

Clearly f from (5.19) can be written as

$$f(u) = e_3^T y(u)$$

with $e_3 = (0, 0, 1)^T$. Thus the derivative of f with respect to the independent variables u in direction v can be calculated through

$$\frac{\partial f}{\partial u} v = e^T \frac{\partial y}{\partial u} v.$$

From (5.21) it follows

$$\frac{\partial y}{\partial u} v = \frac{\partial \phi}{\partial u} v + \frac{\partial \phi}{\partial y} \frac{\partial \phi}{\partial u} v$$

which yields

$$\frac{\partial y}{\partial u} = \left(I - \frac{\partial \phi}{\partial y} \right)^{-1} \frac{\partial \phi}{\partial u}.$$

If one now lets $\lambda \in \mathbb{R}^3$ the solution of

$$\left(I - \frac{\partial \phi}{\partial y}^T\right) \lambda = e_3 \tag{5.22}$$

it holds

$$\frac{\partial f}{\partial u} = \lambda^T \frac{\partial \phi}{\partial u}. \tag{5.23}$$

Moreover, a closer look on the definition of ϕ in equation (5.21) reveals that $(I - \frac{\partial \phi}{\partial y}^T) \in \mathbb{R}^{3\times 3}$ is an upper triangular matrix. Backward substitution then provides the solution of (5.22) in only 3 steps. However, tests show that the automatically derived codes for the reverse mode of AD are not competitive to the handcoded counterparts (see for instance Giles [2007]).

Chapter 6

Computational Reduction of the Calibration Time

Chapter 5 was related to speeding up the calibration via an adjoint equation. In contrast to this more theoretical method, the following chapter deals with a number of computational methods and techniques to reduce the overall calibration time. The first section deals with methods of variance reduction, as a smaller variance of the Monte Carlo estimator allows for a smaller number of simulations and thus speeds up the calibration. The second section contains a multi layer method, where the idea is to have coarse evaluations of the objective function at the beginning of the optimization and finer ones at the end. The third section then explains the idea of storing the random numbers instead of regenerating them every time they are needed which is finally followed by parallelizing the algorithm on several processors.

6.1 Variance Reduction

The concept of variance reduction is understood to be a group of methods to reduce the variance of the Monte Carlo estimator and thus reduce the number of required Monte Carlo simulations in order to achieve a certain accuracy. In this work antithetic sampling and control variates are explained shortly. For a more detailed information see e.g. Glasserman [2003].

6.1.1 Antithetic Sampling

For a brief explanation of *antithetic sampling* consider the simple Black-Scholes SDE example

$$\begin{aligned} C &= E\left(\max(S_T - K, 0)\right) \\ dS_t &= (r-\delta)dt + \sigma S_t dW_t \end{aligned}$$

where the discounting with e^{-rT} in the call price formula has been skipped for simplicity reasons. By definition, the increments of the Brownian motion are $\mathcal{N}(0, \Delta t)$ distributed. Thus, for the discrete version given by

$$\begin{aligned} C_{\mathsf{M}} &= \frac{1}{M}\sum_{m=1}^{M}\left(\max(S_N^m - K, 0)\right) \\ S_{n+1}^m &= S_n^m(r-\delta)\Delta t + \sigma S_n^m \Delta W_n^m, \quad n = 0, ..., N-1,\ m = 1, ..., M \end{aligned}$$

one simulates $\mathcal{N}(0,1)$ distributed random numbers and multiplies them with $\sqrt{\Delta t}$ to receive $\mathcal{N}(0, \Delta t)$ distributed random numbers for the Brownian increments ΔW_n^m. By definition of the standard normal distribution it holds that, if Z is standard normally distributed so is $-Z$. Furthermore $-Z$ is the reflection of Z around the origin. Consequently, one simulates the antithetic path through replacing ΔW_n^m with its negative counterpart $-\Delta W_n^m$, i.e.

$$\tilde{S}_{n+1}^m = \tilde{S}_n^m(r-\delta)\Delta t + \sigma \tilde{S}_n^m(-\Delta W_n^m).$$

The call price is then the mean of the resulting two prices:

$$C_{\mathsf{M}}^{\mathsf{as}} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left(\max(S_N^m - K, 0) + \max(\tilde{S}_N^m - K, 0)\right).$$

Figure 6.1 displays the effect for this Black-Scholes example. Obviously, the combined estimator (orange line) has a significantly lower variance than the standard estimator (blue line) and is thus a better approximation to the exact price (green line).

To analyze the computational complexity of this method, it is important to note preliminarily that the random normal deviate for $\tilde{S}_N^m$, i.e. $-\Delta W_n^m$ can be received by only changing the sign of ΔW_n^m, which has already been calculated for S_N^m. If one neglects this difference in computing time, the computational effort for S_N^m and $\tilde{S}_N^m$ is the same. Thus, the complexity of the antithetic sampling estimator is assumed to be twice compared to the plain estimator. Hence, for the analysis of the effective variance reduction, it is reasonable to consider a plain estimator with

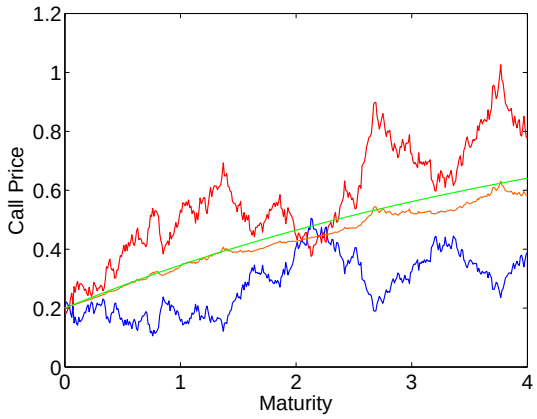

Figure 6.1: Standard (blue line) , antithetic (red line) , combined (orange line) and exact (green line) Black-Scholes call price with $(r - d) = \sigma = 0.2$.

twice the number of simulations

$$C_{\mathrm{M}} = \frac{1}{2M} \sum_{m=1}^{2M} \max(S_N^m - K, 0).$$

For the variance of this estimator it holds true that

$$\begin{aligned} \mathrm{Var}(C_{\mathrm{M}}) &= \mathrm{Var}\left(\frac{1}{2M} \sum_{m=1}^{2M} \max(S_N^m - K, 0) \right) \\ &= \frac{1}{4M^2} \sum_{m=1}^{2M} \mathrm{Var}\left(\max(S_N^m - K, 0) \right) \end{aligned}$$

as the S_N^m are independent identically distributed and $\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$ for a real number a and a random variable X. Certainly $\max(S_N^m - K, 0)$ is equal to zero if the stock price is smaller than the strike price and thus

$$\mathrm{Var}(C_{\mathrm{M}}) = \frac{1}{4M^2} \sum_{m=1}^{2M} \mathrm{Var}\left(S_N^m - K \right) \mathbf{1}_{(S_N^m > K)} = \frac{1}{4M^2} \sum_{m=1}^{2M} \mathrm{Var}\left(S_N^m \right) \mathbf{1}_{(S_N^m > K)}$$

as $\mathrm{Var}(X + a) = \mathrm{Var}(X)$. Without loss of generality it is assumed that the stock prices for all paths are larger than the strike. As the S_N^m are independent identically distributed, it is essential that

$$\mathrm{Var}(C_{\mathrm{M}}) = \frac{1}{2M} \mathrm{Var}\left(S_N \right).$$

Analogously the variance of the antithetic sampling estimator is

$$\mathrm{Var}(C_{\mathrm{M}}^{\mathrm{as}}) = \frac{1}{4M^2}\sum_{m=1}^{M}\mathrm{Var}\left(\max(S_N^m - K, 0) + \max(\tilde{S}_N^m - K, 0)\right).$$

As it holds by definition that $\mathrm{Var}(X+Y) = \mathrm{Var}(X)+\mathrm{Var}(Y)+\mathrm{Cov}(X, Y)$ it follows that

$$\begin{aligned}\mathrm{Var}(C_{\mathrm{M}}^{\mathrm{as}}) &= \frac{1}{4M^2}\sum_{m=1}^{M}\mathrm{Var}\left(S_N^m\right)\mathbf{1}_{(S_N^m>K)} + \mathrm{Var}\left(\tilde{S}_N^m\right)\mathbf{1}_{(\tilde{S}_N^m>K)} \\ &+ \mathrm{Cov}(S_N^m, \tilde{S}_N^m)\mathbf{1}_{(S_N^m>K)}\mathbf{1}_{(\tilde{S}_N^m>K)}.\end{aligned}$$

Again it is assumed that $S_N^m > K$ and $\tilde{S}_N^m > K$ for all $m = 1, ..., M$. Hence

$$\mathrm{Var}(C_{\mathrm{M}}^{\mathrm{as}}) = \frac{1}{2M}\mathrm{Var}(S_N) + \frac{1}{2M}\mathrm{Cov}(S_N, \tilde{S}_N).$$

Thus $\mathrm{Var}(C_{\mathrm{M}}^{\mathrm{as}}) - \mathrm{Var}(C_{\mathrm{M}})$ behaves asymptotically like $\mathrm{Cov}(S_N, \tilde{S}_N)$. The variance of the antithetic sampling estimator is hence less than the variance of the plain estimator, if $\mathrm{Cov}(S_N, \tilde{S}_N)$ is negative. The negativity of the covariance is by definition equivalent to

$$E\left(S_N\tilde{S}_N\right) < E\left(S_N\right)E\left(\tilde{S}_N\right).$$

Thus, the more close the mapping of the SDE is to linear, the higher is the variance reduction via antithetic sampling.

6.1.2 Control Variates

In contrast to antithetic sampling, *control variates* is a more complex variance reduction technique. For the explanation reconsider the Monte Carlo estimator for an arbitrary random variable Y:

$$E(Y) \approx \frac{1}{M}\sum_{m=1}^{M} Y_m. \tag{6.1}$$

Assuming the availability of a second random variable $\tilde{Y}$ with known expected value $E(\tilde{Y})$ which has the same distribution as Y, it is possible to calculate the corresponding sample mean

$$\frac{1}{M}\sum_{m=1}^{M} Y_m - \beta(\tilde{Y}_m - E(\tilde{Y}))$$

with a chosen constant $\beta \in [-1;1]$. Just as (6.1), this estimator is unbiased (see also (2.2) on page 14) as

$$\begin{aligned} E\left(\frac{1}{M}\sum_{m=1}^{M} Y_m - \beta(\tilde{Y}_m - E(\tilde{Y}))\right) &= E\left(\frac{1}{M}\sum_{m=1}^{M} Y_m\right) \\ &- \beta\left(E\left(\frac{1}{M}\sum_{m=1}^{M}\tilde{Y}_m\right) - E(\tilde{Y})\right). \end{aligned}$$

In case of the objective function in ($\mathrm{P}_{\mathrm{M},\Delta t,\epsilon}$) the expression

$$\begin{aligned} C^i_{M,\Delta t,\epsilon} &= e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon} - K_i)\right) \\ y^m_{n+1,\epsilon} &= y^m_{n,\epsilon} + a_\epsilon(x, y^m_{n,\epsilon})\Delta t_n + b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n \end{aligned}$$

is replaced by

$$\begin{aligned} C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon} &:= e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon} - K_i) - \beta\left(\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon} - K_i) - E(\pi(\tilde{S}_{T_i} - K_i))\right)\right) \\ y^m_{n+1,\epsilon} &= y^m_{n,\epsilon} + a_\epsilon(x, y^m_{n,\epsilon})\Delta t_n + b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n \\ \tilde{y}^m_{n+1,\epsilon} &= \tilde{y}^m_{n,\epsilon} + \tilde{a}_\epsilon(x, \tilde{y}^m_{n,\epsilon})\Delta t_n + \tilde{b}_\epsilon(x, \tilde{y}^m_{n,\epsilon})\Delta W^m_n. \end{aligned}$$

Note that for the sampled Monte Carlo control variate process the smoothed version $\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon} - K_i)$ is used whereas the expected value is calculated with the help of the unsmoothed process $\pi(\tilde{S}_{T_i} - K_i)$. One has to accept the resulting approximation error as the objective function has to fulfill the differentiability requirements on the one hand (see section 3.3) and the expected value is only known for the unsmoothed process on the other hand.

It can be shown that for an optimal choice of β the variance of the combined control variate estimator is smaller than for the original one, if the correlation between Y and $\tilde{Y}$, i.e. $\pi_\epsilon(S_{T_i,\epsilon} - K_i)$ and $\pi_\epsilon(\tilde{S}_{T_i,\epsilon} - K_i)$ is high:

Theorem 6.1. *Let $S^m_{t,\epsilon}$, $\tilde{S}^m_{t,\epsilon}$ the first components of the solutions of*

$$y^m_{n+1,\epsilon} = y^m_{n,\epsilon} + a_\epsilon(x, y^m_{n,\epsilon})\Delta t_n + b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n$$

respectively

$$\tilde{y}^m_{n+1,\epsilon} = \tilde{y}^m_{n,\epsilon} + \tilde{a}_\epsilon(x, \tilde{y}^m_{n,\epsilon})\Delta t_n + \tilde{b}_\epsilon(x, \tilde{y}^m_{n,\epsilon})\Delta W^m_n.$$

For the Monte Carlo estimator

$$C^i_{M,\Delta t,\epsilon} = e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon} - K_i)\right)$$

the control variate estimator

$$C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon} = e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon}-K_i)-\beta\left(\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon}-K_i)-E(\pi(\tilde{S}_{T_i}-K_i))\right)\right)$$

has minimal variance for

$$\beta^* = \frac{\mathrm{Cov}(\pi_\epsilon(S^m_{N_i,\epsilon}-K),\pi_\epsilon(\tilde{S}^m_{N_i,\epsilon}-K))}{\mathrm{Var}(\pi_\epsilon(\tilde{S}^m_{N_i,\epsilon}-K))}. \tag{6.2}$$

In particular the variance of this estimator is

$$\mathrm{Var}(C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon}) = \left(1-\rho^2\right)\mathrm{Var}(C^i_{M,\Delta t,\epsilon}) \tag{6.3}$$

where ρ is defined as the correlation coefficient between the two payoffs $\pi_\epsilon(S^m_{N_i,\epsilon}-K)$ and $\pi(\tilde{S}^m_{N_i,\epsilon}-K)$.

Proof. Glasserman [2003], p. 186 f. □

Consequently, if the correlation between $\pi(S_{T_i}-K_i)$ and $\pi(\tilde{S}_{T_i}-K_i)$ is high, one can conclude, that both random variables act similar. Thus, the error in the approximation of $E(\pi(\tilde{S}_{T_i}-K_i))$ should be similar to the error when approximating $E(\pi(S_{T_i}-K_i))$. In the extreme case, where both processes are identical, the covariance would be equal to one and the variance would be zero. Obviously both Monte Carlo processes in the control variate estimation formula would erase each other and $C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon}$ would be equal to $E(\pi(\tilde{S}_{T_i}-K_i))$. In other words, $\frac{1}{M}\sum_{m=1}^{M}\pi(\tilde{S}^m_{T_i}-K_i)-E(\pi(\tilde{S}_{T_i}-K_i))$ serves as a control for the approximation error in $E(\pi(S_{T_i}-K_i))$ weighted with the corresponding correlation. However, the effect of control variates decreases strongly with a decreasing correlation of the two processes because the correlation enters quadratically into the formula. The clear restriction is the knowledge of a process with high correlation to the primary process and well-established expected value formula.

6.1.3 Comments on the Gradient Calculation

Note that the above introduced variance reduction techniques change the objective function. Of course, this change has an effect on the objectives first and second order derivatives which have to be calculated for the solution of the calibration problem. Chapter 5 deals with the topic of these derivatives with respect to the underlying parameters. Reconsidering the objective function of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$

$$C^i_{M,\Delta t,\epsilon}(x) = e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon}(x)-K_i)\right)$$

it holds

$$\frac{\partial}{\partial x_p} C^i_{M,\Delta t,\epsilon}(x) = e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \left(\pi'_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}(x) \right).$$

It turns out, that the introduced variance reduction techniques lead to a similar structure of the objective. Recall the call price formula provided by antithetic sampling

$$C^{i,\mathrm{as}}_{M,\Delta t,\epsilon}(x) = e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{2} \left(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) + \pi_\epsilon(\tilde{s}^m_{N_i,\epsilon}(x) - K_i) \right).$$

Thus the gradient is just a combination of two objective functions similar to the one in $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$. The same observation can be made with control variates. Reconsidering the control variate estimator

$$\begin{aligned} C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon}(x) \quad &:= \quad e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \Bigg(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \\ &- \quad \beta \left(\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon}(x) - K_i) - E(\pi(\tilde{S}_{T_i}(x) - K_i)) \right) \Bigg) \end{aligned}$$

the gradient is

$$\begin{aligned} \frac{\partial}{\partial x_p} C^{i,\mathrm{cv}}_{M,\Delta t,\epsilon}(x) \quad &= \quad e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} \Bigg(\pi'_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \frac{\partial}{\partial x_p} s^m_{N_i,\epsilon}(x) \\ &- \quad \frac{\partial}{\partial x_p} \beta \left(\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon}(x) - K_i) - E(\pi(\tilde{S}_{T_i}(x) - K_i)) \right) \\ &- \quad \beta \Bigg(\pi'_\epsilon(\tilde{s}^m_{N_i,\epsilon}(x) - K_i) \frac{\partial}{\partial x_p} \tilde{s}^m_{N_i,\epsilon}(x) \\ &- \quad \frac{\partial}{\partial x_p} E(\pi(\tilde{S}_{T_i}(x) - K_i)) \Bigg) \Bigg). \end{aligned}$$

Usually $\frac{\partial}{\partial x_p}\beta$ can be calculated with gradient information on $\pi_\epsilon(s^m_{N_i,\epsilon} - K_i)$ and $\pi_\epsilon(\tilde{s}^m_{N_i,\epsilon} - K_i)$ (see also Theorem 6.1). $\frac{\partial}{\partial x_p} E(\pi(\tilde{S}_{T_i} - K_i))$ can be evaluated with finite differences for instance, which only slightly increases the calculation time, as the evaluation of $E(\pi(\tilde{S}_{T_i} - K_i))$ can usually be done very fast.

Summarizing, antithetic sampling and control variates keep the general structure of the objective function. Consequently, the adjoint technique can be applied by adding adjoint equations similar to the standard ones, e.g. for the antithetic SDE. Thus, these variance reduction methods have been neglected for the derivation of the adjoint equation. Of course, in practical implementations they can be applied, if possible and helpful.

The above introduced variance reduction method speeds up the calibration via decreasing the calculation time for the objective function. The Multi-Layer approached in the next section aims on the optimization algorithm.

6.2 Multi Layer

Generally, one can expect to choose an initial value for the optimization algorithm which is far away from the optimum, except from cases of recalibration. Thus, the approximation error of the model prices with respect to the market prices, i.e. the objective function value, can be expected to be comparably large at the beginning of the optimization. Consequently the requirement to evaluate the model prices with high accuracy is relatively weak at the first iterations and increases during the optimization process due to the fact that the overall approximation error of the function evaluation should be dominated by the approximation error of the model prices to the market prices at the beginning of the optimization process. Additionally, the total approximation error can be decomposed into three parts, namely the Monte-Carlo, the discretization and the smoothing error:

as it has already been described in section 6.2. This motivates to start the optimization with relatively few simulations, a large discretization step size and a large smoothing parameter, i.e. a coarse layer, and to increase the accuracy during optimization. The resulting algorithm is shown in algorithm 5.

Algorithm 5 Multi Layer

1: Choose Q layers $(M_q|\Delta t_q|\epsilon_q)$, $q = 1, ..., Q$
2: Start optimization at layer $(M_1, \Delta t_1, \epsilon_1)$ and determine point x_1
3: **for** q=2 to Q **do**
4: Given initial point x_{q-1}, optimize at layer $(M_q|\Delta t_q|\epsilon_q)$
5: and determine an approximately stationary point x_q
6: **end for**

The effect of this method can be described as follows. The optimization on the q-th layer starting with x_{q-1} should lead to a value closer to the optimum than x_{q-1}, namely x_q. Thus the number of iterations on the next finer layer decreases significantly compared to an optimization starting with x_{q-1}. This effect carries on from layer to layer and should reduce the overall calibration time.

Note, that the effect of this method is supported by variance reduction techniques, like antithetic sampling or control variates introduced above, as these methods allow for fewer simulations and thus coarse layers can be chosen even coarser.

6.3 Storing Random Numbers

Solving the stochastic differential equation in ($P_{M,\Delta t,\epsilon}$) requires the generation of random numbers for the brownian increments. As described in section 3.4 the idea of sample average approximation is to keep this random number sequence identical during the optimization. Thus, every time the objective function has to be evaluated, which is at least once per iteration of the optimization algorithm, these random numbers have to be regenerated. Before explaining the idea of storing the numbers, a brief introcution will be given into the topic of their generation. The book of Gentle (Gentle [2003]) gives a detailed overview on this topic.

Reconsider that the solution of the SDE

$$y^m_{n+1,\epsilon} = y^m_{n,\epsilon} + a_\epsilon(x, y^m_{n,\epsilon})\Delta t_n + b_\epsilon(x, y^m_{n,\epsilon})\Delta W^m_n$$

from ($P_{M,\Delta t,\epsilon}$) requires the simulation of the Brownian increments ΔW^m_n for every time step $n = 1, ..., N$ and every simulation $m = 1, ..., M$. By definition of the Brownian motion, these increments are normally distributed with mean 0 and variance Δt_n. To implement the simulation of independently and $\mathcal{N}(0, \Delta t_n)$ distributed random numbers, it is made use of the fact, that if a random number X is $\mathcal{N}(0, 1)$ distributed, it holds that $X\sqrt{\Delta t_n}$ is $\mathcal{N}(0, \Delta t_n)$ distributed. Thus one may set

$$\Delta W^m_n = Z^m_n \Delta t_n, \quad n = 1, ..., N \quad m = 1, ..., M$$

where $Z^m_n \sim \mathcal{N}(0, 1)$. Consequently the simulation of $N \times M$ independently and standard normally distributed random numbers is required.

At first sight, the question arises, how to simulate randomness on a computer. There are some approaches that make use of random events occurring in the real world. For instance, measuring atomic decay leads to a Bernoulli distribution. John Walker implemented this at Fourmilab. A file containing a random sequence can be obtained at `http://www.fourmilab.ch/hotbits`. A similar approach has been followed by Toshiba. They developed a PCI board that measures thermal noise in a semiconductor. This board is called RandomMaster.

However, though the so generated sequences are truly random, the number of distributions is strongly restricted to the distribution defined by a natural event, that can be measured with the help of a computer. Additionally, these techniques seem inapplicable for a daily use. Consequently, practitioners often consider so called *pseudo random number generators*. These are deterministic programs, that calculate a sequence of numbers, with statistical properties, which are quite close to the true randomness. Examples may be found on the numerical recipes homepage: `http://www.nr.com`.

What many generators have in common, is that the numerical effort for the

evaluation of the numbers for the Brownian increments is a multiple of the effort necessary to evaluate the rest of the stochastic differential equation. Consequently, for the calculation of the SDE solution, the random number generation is the main effort. Combining this with the already mentioned fact, that the random number sequence is fixed during optimization raises the expectation that storing and reading the random numbers out of the system memory is significantly faster on a usual Desktop-PC than regenerating them in each function evaluation. Unfortunately, the system memory of such a usual Desktop-PC is limited. Considering for instance a SDE evaluation of a two dimensional model with option maturities up to 5 years, a time step every day and 1,000,000 simulations requires the calculation of 3,650,000,000 numbers. This amounts to approximately 27 GB in a double precision framework. As storing and reading on a hard drive is no alternative due to transfer rate limitations, the idea is to store as many numbers as possible in system memory and regenerate further numbers if required. The effect of this technique than depends on the ratio of the available system memory to the size of the random sequence.

Table 6.1 shows calculation times for one function evaluation, 100 call options in this case, with regeneration (first row) in comparison to storing and reading (second row). In this example, 10,000 simulations and a discretization step size of

	1st evaluation	Further evaluations
Without Storing	21067276	21067276
With Storing	21067276	8391305

Table 6.1: Comparison of calculation time (μs) for one function evaluation, i.e. 100 call prices, with and without storing random numbers for $M = 100,000$ and $\Delta t = 5 \times 10^{-3}$.

$\Delta t = 10^{-3}$ have been used. This amounts to 1563MB of required memory. Thus, as the computer obtains 2GB RAM, all numbers could be stored. In the test case with storing, the numbers are generated and stored during the first evaluation and read during the further evaluations. Consequently, the calculation times for the first evaluation equals in both tests, whereas the further evaluations are 2.5 times faster in the storing and reading case. This effect is expected to decay, when the ratio of required to available memory increases. Section 7.3 will show, how this speed up carries on to the calibration.

6.4 Parallelization

Parallel Computing may be defined as the distribution of a number of jobs on different calculation units. Consider for example the exercise to calculate function values for two different sets of parameters. As these jobs are clearly independent of

each other, they can easily be distributed on two different computers. This would then take half the computation time of the sequential way on one computer, if one neglects for example the overhead of communication between the two computers. This simple example makes clear, that the synchronous computation of large jobs, which only requires few information such that communication overhead is negligible, can be very efficient to save computation time. Especially in times, where the increase of computing power of Desktop PCs has slowed down, parallelization becomes more and more important.

However, it is not always as easy to split a job into several subjobs, like in the simple example above. Usually, one would have to solve the question, which part of the program can be distributed on several processors. In this thesis, the exercise is to solve the calibration problem. On the one hand, due to its sequential structure, the optimization algorithm itself cannot be arranged in several parallel jobs. On the other hand, the Monte-Carlo method is very well suited for the parallel computation, as the different simulations are independent of each other such that one has a *Single Instruction Multiple Data* (SIMD) structure. Instead of e.g. M simulations on one computer, one could easily calculate $\frac{1}{n}$M simulations on each of n computers.

As briefly described above, one differs between the parallelization on several CPUs in several computers and the parallelization on several CPUs in one computer. The first is usually realized with the Message Passing Interface (MPI) standard. The competition lies in passing all necessary information from one to the other computer. This becomes easier, if all CPUs are built in one PC. In this situation, all CPUs share the same memory. Consequently, this is denoted as shared memory parallelization. OpenMP is a common library to realize the shared memory parallelization. See for instance Scott et al. [2005] for a detailed introduction in parallel computing.

As the goal is more a feasibility study than a perfect parallel implementation, this thesis focuses on OpenMP. All parallel tests in this section are done on one computer containing 8 AMD Opteron 870 processors with 2.0GHz each and 16GB RAM. The efficiency of a parallel program can be analyzed by its *parallel efficiency*:

Definition 6.2. *(Parallel Efficiency)*
The parallel efficiency is the ratio of calculation time on 1 CPU and the n-th of the calculation time on n CPUs:

$$\text{Parallel Efficiency} = \frac{\text{Calculation Time on 1 CPU}}{\text{n} \times \text{Calculation Time on n CPUs}}.$$

Consequently, if the parallelization would perfectly scale, i.e. synchronous computation on n CPUs leads to a n times faster calculation time, this would result in a parallel efficiency of 100%.

Test runs (table 6.2) for one function evaluation with 100,000 simulations and a discretization step size of 5×10^{-3} show a strong decrease in computation time

and thus a high parallel efficiency from 98% on 2 CPUs to 88% on 8 CPUs. This

Number of CPUs	Computation Time (μs)	Parallel Efficiency	Speed Up
1	31069451	—	—
2	15834206	98.11%	1.96
3	11362625	91.15%	2.73
4	8502939	91.35%	3.65
5	7203398	86.26%	4.31
6	5771182	89.73%	5.38
7	5036708	88.12%	6.17
8	4376403	88.74%	7.10

Table 6.2: Comparison of calculation time (μs) for one function evaluation, i.e. 100 call prices, on 1 to 8 CPUs for $M = 10,000$ and $\Delta t = 10^{-3}$.

validates the expectation, that the Monte Carlo method is very well suited for parallel computing. Furthermore, these high parallel efficiencies result in a almost linear scaling speed up with increasing number of CPUs of up to 7.1 on 8 CPUs. Section 7.3 deals with the parallel calibration.

Chapter 7

Numerical Results

In this chapter numerical results are presented which underline the performance and the theoretical coherence of the Monte Carlo calibration method developed so far. First the chosen market data and additional settings for this chapter are introduced. Section 7.2 then illustrates for the example of the Stein-Stein model that the solutions of $(P_{M,\Delta t,\epsilon})$ converge to those of (P). The last part is then devoted to a detailed analysis of the speed-ups obtained for the calibration of a lognormal variance model by applying all techniques and methods introduced in this thesis.

7.1 Calibration Set Up

For all test cases the financial market model is calibrated to a set of 100 European call options on the S&P 500 index taken from Andersen and Brotherton-Ratcliffe [1997/1998]. The data is illustrated in table 7.1 in the form of implied volatilities,

K \ T	0.175	0.425	0.695	0.940	1.000	1.500	2.000	3.000	4.000	5.000
0.85	0.190	0.177	0.172	0.171	0.171	0.169	0.169	0.168	0.168	0.168
0.90	0.168	0.155	0.157	0.159	0.159	0.160	0.161	0.161	0.162	0.164
0.95	0.133	0.138	0.144	0.149	0.150	0.151	0.153	0.155	0.157	0.159
1.00	0.113	0.125	0.133	0.137	0.138	0.142	0.145	0.149	0.152	0.154
1.05	0.102	0.109	0.118	0.127	0.128	0.133	0.137	0.143	0.148	0.151
1.10	0.097	0.103	0.104	0.113	0.115	0.124	0.130	0.137	0.143	0.148
1.15	0.120	0.100	0.100	0.106	0.107	0.119	0.126	0.133	0.139	0.144
1.20	0.142	0.114	0.101	0.103	0.103	0.113	0.119	0.128	0.135	0.140
1.30	0.169	0.130	0.108	0.100	0.099	0.107	0.115	0.124	0.130	0.136
1.40	0.200	0.150	0.124	0.110	0.108	0.102	0.111	0.123	0.128	0.132

Table 7.1: Market data: Implied volatilities for S&P 500 index options taken from Andersen and Brotherton-Ratcliffe [1997/1998].

as explained in section 2.2. Like in Andersen and Brotherton-Ratcliffe the riskfree interest rate is chosen as $r = 0.06$, the dividend yield as $\delta = 0.0262$ and it is assumed that the initial stock price is normalized to $S_0 = 1$. Figure 7.1 illustrates the volatility surface graphically. The market data shows a so called *volatility smile*.

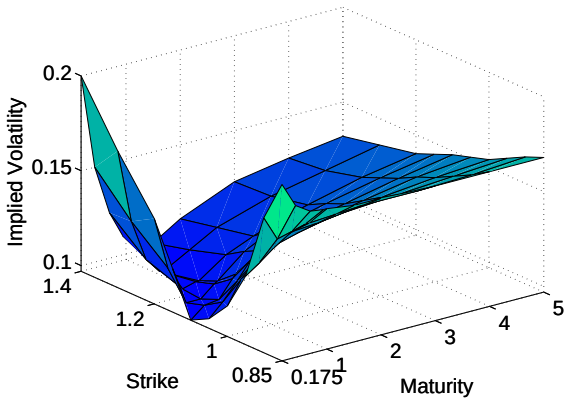

Figure 7.1: Graphical illustration of market data from tabular 7.1.

This means, that the prices of *at-the-money* options, i.e. options with strikes close to the actual stock price (spot), is lower than the price of *in-* or *out-of-the-money* calls, which are options with a strike lower or higher than the actual spot.

All test runs are realized on a desktop PC with an Intel Core2 Duo CPU E7300 with 2.66GHz and 2GB system memory (RAM). Note that both cores are only used in a parallel setting, which will be denoted explicitly. The code has been implemented in C++ and antithetic sampling (section 6.1.1) has been applied in all tests. As a first step, the convergence behavior of solutions of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ to solutions of (P) will be analyzed in the next section.

7.2 Numerical Validation of the Convergence

According to Theorem 4.22 every limit point of approximately stationary points of problem $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ is a stationary point of the true calibration problem (P) if among others (A.1)-(A.9) are satisfied. To verify this convergence behavior, the test case of calibrating the model of Stein and Stein [1991]

$$\begin{aligned} dS_t &= (r-\delta)S_t dt + v_t S_t dW_t^1\,,\ S_0 \in (0,\infty)\,,\ 0 \le t \le T \\ dv_t &= \kappa(\theta - v_t)dt + \sigma(\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2)\,,\ v_0 \in (0,\infty) \end{aligned}$$

to the set of call options listed in Table 7.1 is chosen. This model is particularly suited for the convergence analysis, as it on the one hand fullfills the Lipschitz and

growth assumption required by Theorem 4.22. On the other hand it is possible to derive a closed form solution for the price of call options (see Stein and Stein [1991]), which in turn allows to compare the outcome of a Monte Carlo calibration with an accurate closed-form calibration:

Lemma 7.1. *Define*

$$A = -\frac{\kappa}{\sigma^2} \quad B = \frac{\theta\kappa}{\sigma^2} \quad C = -\frac{x}{\sigma^2 t}.$$

The variable x contained in C is a dummy variable for a subsequent integral definition. Furthermore let

$$\begin{aligned}
a &= \sqrt{(A^2 - 2C)} \\
b &= -\tfrac{A}{a} \\
L &= -A - a\left(\tfrac{\sinh(a\sigma^2 t)+b\cosh(a\sigma^2 t)}{\cosh(a\sigma^2 t)+b\sinh(a\sigma^2 t)}\right) \\
M &= B\left(\tfrac{b\sinh(a\sigma^2 t)+b^2\cosh(a\sigma^2 t)+1-\sigma^2}{\cosh(a\sigma^2 t)+b\sinh(a\sigma^2 t)} - 1\right) \\
N &= \tfrac{a-A}{2a^2}\left(a^2 - AB^2 - B^2 a\right)\sigma^2 t + \tfrac{B^2(A^2-a^2)}{2a^3}\tfrac{(2A+a)+(2A-a)e^{2a\sigma^2 t}}{(A+a+(a-A)e^{2a\sigma^2 t})} \\
&+ \tfrac{2AB^2(a^2-A^2)e^{a\sigma^2 t}}{a^3(A+a+(a-A)e^{2a\sigma^2 t}} - \tfrac{1}{2}\ln\left(\tfrac{1}{2}\left(\tfrac{A}{a}+1\right) + \tfrac{1}{2}\left(1-\tfrac{A}{a}\right)e^{2a\sigma^2 t}\right)
\end{aligned}$$

and

$$I = e^{\left(\frac{Lv_0^2}{2} + Mv_0 + N\right)}.$$

As I depends on x included in C, I is replaced by I(x). For the special case, that the drift $(r-\delta) = 0$, the option price is given by

$$\overline{C}(t, S_t, v_t) = (2\pi)^{-1} S_t^{-\frac{3}{2}} \int_{-\infty}^{\infty} I\left(\left(x^2 + \frac{1}{4}\right)\frac{t}{2}\right) e^{ix\ln(S_t)} dx.$$

and

$$C(t, S_t, v_t) = e^{-(r-\delta)t}\overline{C}(S_t e^{-(r-\delta)t}).$$

Proof. Stein and Stein [1991] pp. 743 ff. □

The calibration problem now consists of identifying the unknown parameters $x = (v_0, \kappa, \theta, \sigma, \rho)^T$. The set X of feasible parameters x is described by suitably chosen lower and upper bounds on the parameters. The imposed lower and upper bounds assure the compactness of the feasible set and limit the parameter combinations to practically relevant values. For the example here the bounds

$$\begin{aligned}
&0.0001 \le v_0 \le 2.0, \quad 0.05 \le \kappa \le 2.0, \quad 0.0001 \le \theta \le 2.0, \\
&0.0001 \le \sigma \le 4.0, \quad -0.985 \le \rho \le 0.985,
\end{aligned} \tag{7.1}$$

where chosen, which in summary leads to a nonempty, convex and compact set

X satisfying Assumption (A.1). Since the parameter $x^1 = v_0$ is the start value of the stochastic variance differential equation, the Stein-Stein dynamics (7.1) at first sight do not seem to fit into the general model framework (3.1). However, the simple transformation $\tilde{v}_t := v_t / v_0$ yields the equivalent model dynamics

$$\begin{aligned} dS_t &= (r-\delta)S_t dt + v_0 \tilde{v}_t^+ S_t dW_t^1, \\ d\tilde{v}_t &= \kappa\left(\tfrac{\theta}{v_0} - \tilde{v}_t^+\right) dt + \sigma\left(\rho dW_t^1 + \sqrt{1-\rho^2} dW_t^2\right), \ \tilde{v}_0 = 1, \end{aligned}$$

with start values that are independent of the model parameters $x \in \mathbb{R}^P$. Though it is clearly true, that the applied positivity preserving scheme full truncation (see section 3.3) is not required due to Lemma 3.1, it is indeed required in the Euler-Maruyama discretized case. Therefore, full truncation has already been involved in the continuous Stein-Stein model.

In terms of the general model dynamics (3.1), these stochastic differential equations can be expressed by setting $P = 5$, $L = 2$, $x = (v_0, \kappa, \theta, \sigma, \rho)^T$, $y = (y_1, y_2)^T$ and choosing the maps $a : X \times \mathbb{R}^2 \to \mathbb{R}^2$, $b : X \times \mathbb{R}^2 \to \mathbb{R}^2 \times \mathbb{R}^2$ as

$$\begin{aligned} a(x,y) &:= \begin{pmatrix} (r-\delta)y_1 \\ \kappa(\frac{\theta}{v_0} - y_2^+) \end{pmatrix} \\ b(x,y) &:= \begin{pmatrix} v_0 y_2^+ y_1 & 0 \\ \sigma\rho & \sigma\sqrt{1-\rho^2} \end{pmatrix}. \end{aligned}$$

Obviously, the maps a and b are not continuously differentiable on $X \times \mathbb{R}^2$. To eliminate the non-differentiabilities introduced by the square root, the spline function defined in (3.7) is used to obtain the smooth approximations

$$\begin{aligned} a_\epsilon(x,y) &:= \begin{pmatrix} (r-\delta)y_1 \\ \kappa(\frac{\theta}{v_0} - \pi_\epsilon(y_2)) \end{pmatrix} \\ b_\epsilon(x,y) &:= \begin{pmatrix} v_0 \pi_\epsilon(y_2) y_1 & 0 \\ \sigma\rho & \sigma\sqrt{1-\rho^2} \end{pmatrix} \end{aligned}$$

of the maps a, b. Hence the smoothness Assumption (A.3) is fulfilled such that one can make use of derivative-based optimization methods to identify approximately stationary points.

Within each iteration of the optimization algorithm the Jacobian of the residual function (5.1) and hence the gradient of the objective is computed via the adjoint method (Theorem 5.6). For the implementation the Jacobians of a_ϵ and $b_\epsilon \Delta W$ are

necessary:

$$
\begin{aligned}
\tfrac{\partial}{\partial y} a_\epsilon(x,y) &= \begin{pmatrix} r-\delta & 0 \\ 0 & -\kappa\pi'_\epsilon(y_2) \end{pmatrix} \\
\tfrac{\partial}{\partial x} a_\epsilon(x,y) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\kappa\frac{\theta}{v_0^2} & \frac{\theta}{v_0}-\pi_\epsilon(y_2) & \frac{\kappa}{v_0} & 0 & 0 \end{pmatrix} \\
\tfrac{\partial}{\partial y} b_\epsilon(x,y)\Delta W &= \begin{pmatrix} v_0\pi_\epsilon(y_2)\Delta W^1 & v_0\pi'_\epsilon(y_2)y_1\Delta W^1 \\ 0 & 0 \end{pmatrix} \\
\tfrac{\partial}{\partial x} b_\epsilon(x,y)\Delta W &= \begin{pmatrix} \pi_\epsilon(y_2)y_1\Delta W^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma\overline{\Delta W} & \sigma\widetilde{\Delta W} \end{pmatrix}
\end{aligned}
$$

where $\overline{\Delta W} = \rho\Delta W^1 + \sqrt{1-\rho^2}\Delta W^2$ and $\widetilde{\Delta W} = \Delta W^1 - (\frac{\rho}{\sqrt{1-\rho^2}})\Delta W^2$.

The optimization is started with initial values $v_0 = 0.16$, $\kappa = 0.6$, $\theta = 0.16$, $\sigma = 0.4$, $\rho = -0.7$ and iterate with Algorithm 4 until the first order optimality conditions are satisfied with accuracy 10^{-6}. Table 7.2 shows the calibration results for four different sets of Monte Carlo samples, discretization step sizes and smoothing parameters $(M, \Delta t, \epsilon)$ in comparison to the results based on the solution formula. The last two rows contain information on the LSQ value based on the Monte Carlo

x	M=1,000 $\Delta t = 5\times10^{-1}$ $\epsilon = 3.1\times10^{-2}$	M=10,000 $\Delta t = 5\times10^{-2}$ $\epsilon = 1\times10^{-2}$	M=100,000 $\Delta t = 5\times10^{-3}$ $\epsilon = 3.1\times10^{-3}$	M=1,000,000 $\Delta t = 5\times10^{-4}$ $\epsilon = 1\times10^{-3}$	Closed Form
κ	0.79748	1.24537	1.23913	1.24941	1.21877
θ	0.10557	0.11242	0.11660	0.10607	0.10812
σ	0.15374	0.16987	0.17298	0.18026	0.17608
ρ	-0.80274	-0.64010	-0.63283	-0.62404	-0.62356
v_0	0.11549	0.11030	0.11444	0.11838	0.11892
$\mathcal{E}$	6.211e-05	3.226e-05	3.513e-05	3.081e-05	—
$\mathcal{E}^*$	5.424e-04	4.327e-05	3.186e-05	3.108e-05	3.068e-05

Table 7.2: Calibration results for the case of the Stein-Stein model with several Monte Carlo layers and closed form solution.

function evaluations

$$\mathcal{E} := \sum_{i=1}^{I} \left| C^i_{\mathrm{M},\Delta t,\epsilon}(x) - C^i_{obs} \right|^2$$

and the corresponding "true" LSQ value, which results from the evaluation of call prices with the closed form solution on the basis of the calibrated Monte Carlo parameters:

$$\mathcal{E}^* := \sum_{i=1}^{I} \left| C^i(x) - C^i_{obs} \right|^2 .$$

Table 7.2 clearly illustrates the convergence of the solutions of problem $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ as

one increases the number of Monte Carlo simulations M and reduces the discretization step size Δt as well as the smoothing parameter ϵ. To be more precise, the computed stationary points of $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ converge to a stationary point of the true optimization problem (P) computed via a benchmark calibration based on closed form solutions (right column in table 7.2). This is supported by figure 7.2, which shows $\mathcal{E}$ for varying values of the mean reversion speed and level, i.e. κ and θ around the optimal value derived by a closed form based calibration. The remaining pa-

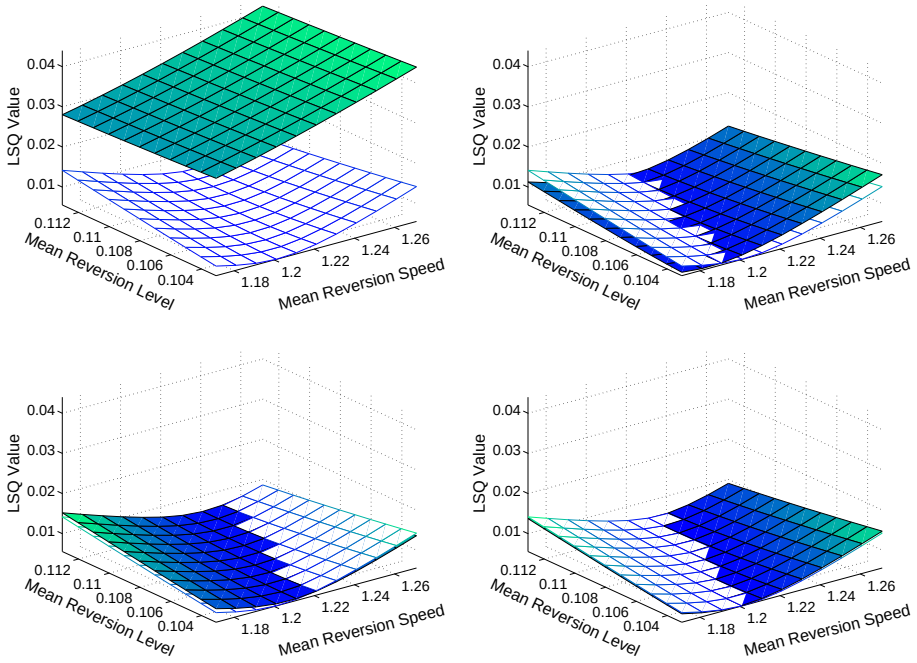

Figure 7.2: Monte Carlo based LSQ values (solid surface) for varying values of mean reversion speed and level around the optimum derived by a closed form calibration in comparison to the LSQ values based on the closed form solution (meshed surface).

rameters are fixed at the closed form optimum. Obviously, the Monte Carlo based LSQ values (solid surface) converges to the closed form LSQs (meshed surface). Consequently, this also numerically confirms the theoretical result of Theorem 4.22.

Furthermore, the least squares error as well as the computed stationary point for the case $M = 10,000$, $\Delta t = 5{\times}10^{-2}$, $\epsilon = 1{\times}10^{-2}$ seems to already lead to a good approximation. However, a closer look reveals, that the "true" LSQ value $\mathcal{E}^*$ becomes slightly worse when increasing the Monte Carlo accuracy to $M = 100,000$, $\Delta t = 5{\times}10^{-3}$ and $\epsilon = 3.1{\times}10^{-3}$, such that the results in the second row are better than it can be generally expected. For practical applications, the approximation in the third row certainly suffices. If one increases the accuracy even further to $M = 1,000,000$, $\Delta t = 5{\times}10^{-4}$, $\epsilon = 1{\times}10^{-3}$, the results listed in Table 7.2 (right)

show virtually no difference to the optimal values obtained via the calibration based on closed form solutions. Moreover these conclusions also hold for the prices of exotic options. Table 7.3 shows prices for a 5 years up-and-out call with spot $S_0 = 1$, strike $K = 0.9$ and barrier $U = 1.2$. Just as for the calibrated parameters, the barrier price for M=100,000, $\Delta t = 1{\times}10^{-3}$ and $\epsilon = 3.1{\times}10^{-3}$ is close to the exact price.

M=1,000 $\Delta t = 5{\times}10^{-1}$ $\epsilon = 3.1{\times}10^{-2}$	M=10,000 $\Delta t = 5{\times}10^{-2}$ $\epsilon = 1{\times}10^{-2}$	M=100,000 $\Delta t = 5{\times}10^{-3}$ $\epsilon = 3.1{\times}10^{-3}$	M=1,000,000 $\Delta t = 5{\times}10^{-4}$ $\epsilon = 1{\times}10^{-3}$	Closed Form
1.5381e-02	1.3314e-02	1.2720e-02	1.3378e-02	1.3163e-02

Table 7.3: Barrier Prices for a 5 years up-and-out call with spot $S_0 = 1$, strike $K = 0.9$ and barrier $U = 1.2$.

To illustrate the order of the above shown convergence derived in section 4.2, table 7.4 displays the error coefficients $|f_k(x_k) - f(x^*)|$, $\frac{|f_k(x_k)-f(x^*)|}{1/\sqrt{\mathrm{M}}}$ as well as $\frac{|f_k(x_k)-f(x^*)|}{1/\mathrm{M}}$. Here $f_k(x_k)$ is the optimal value obtained by solving $(\mathrm{P}_{\mathrm{M},\Delta t,\epsilon})$ with M_k,

Layer	$\lvert f_k(x_k) - f(x^*)\rvert$	$\frac{\lvert f_k(x_k)-f(x^*)\rvert}{1/\sqrt{\mathrm{M}}}$	$\frac{\lvert f_k(x_k)-f(x^*)\rvert}{1/\mathrm{M}}$
M=1,000 $\Delta t = 5{\times}10^{-1}$ $\epsilon = 3.1{\times}10^{-2}$	$3.1430e-05$	$3.1430e-04$	$3.1430e-03$
M=10,000 $\Delta t = 5{\times}10^{-2}$ $\epsilon = 1{\times}10^{-2}$	$1.5800e-06$	$4.9964e-05$	$1.5800e-03$
M=100,000 $\Delta t = 5{\times}10^{-3}$ $\epsilon = 3.1{\times}10^{-3}$	$4.4500e-06$	$4.4500e-04$	$4.4500e-02$
M=1,000,000 $\Delta t = 5{\times}10^{-4}$ $\epsilon = 1{\times}10^{-3}$	$1.3000e-07$	$1.3000e-04$	$1.3000e-01$

Table 7.4: Error analysis for the results in table 7.2.

Δt_k and ϵ_k. Note that Δt and ϵ have been chosen relative to M as described at the end of section 4.2, namely $\Delta t_k = 500/M_k$ and $\epsilon_k = 1/\sqrt{M_k}$. Thus $\mathcal{O}(1/\sqrt{M}) = \mathcal{O}(\Delta t^{\frac{1}{2}}) = \mathcal{O}(\epsilon)$. It can be observed, that $|f_k(x_k) - f(x^*)|$ converges to zero, $\frac{|f_k(x_k)-f(x^*)|}{1/\mathrm{M}}$ converges to infinity and $\frac{|f_k(x_k)-f(x^*)|}{1/\sqrt{\mathrm{M}}}$ converges to a constant which determines the theoretical convergence order result of $\mathcal{O}(1/\sqrt{M} + \Delta t^{\frac{1}{2}} + \epsilon)$.

Having confirmed the theoretical viability of the Monte Carlo calibration method for the benchmark case of the Stein-Stein model as well as the convergence behavior, the following section contains results to analyze the calibration speed.

7.3 Analysis of the Calibration Speed

One of the main advantages of a calibration via Monte Carlo is its flexibility with respect to changes of the model dynamics (3.1). Usually a small change of the Euler discretization code suffices to take the altered dynamics into account. This is also the case for the next test example, a stochastic volatility model with lognormal distribution of the variance

$$\begin{aligned} dS_t &= (r-\delta)S_t dt + \sqrt{v_t^+}S_t dW_t^1, \ S_0 > 0 \\ dv_t &= \kappa(\theta - v_t^+)dt + \sigma v_t^+ \left(\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2\right), \ v_0 > 0. \end{aligned} \tag{7.2}$$

As in the Stein-Stein model (7.1) the lognormal variance model parameters are given in the form of the initial variance v_0, the mean reversion speed κ, the mean reversion level θ, the volatility of variance σ and the correlation ρ. However, although the dynamics of (7.2) and the Stein-Stein model (7.1) or the Heston model (3.2) look very similar, to the knowledge of the author there does not exist a closed-form solution for the price of a European call option in model (7.2). Hence alternative calibration methods like the one proposed in this thesis are necessary. Again the lower and upper bounds (7.1) have been chosen. To clearly illustrate the achieved speed up of the applied techniques, e.g. the adjoint equation, the parameters are chosen to be time constant in a first example and time dependent on 10 subintervals in a second test case (see also (5.13) on page 80). Consequently, the L^2-penalty term

$$\sum_{b=2}^{B}(\kappa_b - \kappa_{b-1})^2 + \sum_{b=2}^{B}(\theta_b - \theta_{b-1})^2 + \sum_{b=2}^{B}(\sigma_b - \sigma_{b-1})^2 + \sum_{b=2}^{B}(\rho_b - \rho_{b-1})^2,$$

multiplied by a suitable regularization parameter $\mu > 0$ has been added to the objective function to reduce the ill-conditioning resulting from the increasing number of parameters with increasing number of subintervals. Table 7.5 shows the calibration times for time constant parameters (B=1) and time dependent parameters on B=10 subintervals. However, as calibration times of several hours or even days are unac-

Method	M=1,000 $\Delta t = 5\times10^{-1}$ $\epsilon = 3.1\times10^{-2}$	M=10,000 $\Delta t = 5\times10^{-2}$ $\epsilon = 1\times10^{-2}$	M=100,000 $\Delta t = 5\times10^{-3}$ $\epsilon = 3.1\times10^{-3}$	M=1,000,000 $\Delta t = 5\times10^{-4}$ $\epsilon = 1\times10^{-3}$
Time Constant: B=1				
Plain MC.	00:00:01	00:01:02	01:30:10	125:41:14
Time Dependent: B=10				
Plain MC.	00:01:38	00:11:17	05:31:33	245:51:37

Table 7.5: Calibration times (hh:mm:ss) for several Monte Carlo layers with time constant and time dependent parameters on B=10 subintervals.

ceptable for practical applications, these results show, that additional techniques to speed up the calibration are strongly required. As a first step, it will be shown, how the methods introduced in chapter 5 and chapter 6 effect the calibration run. The concrete speed up will be displayed in the latter of this section.

Section 6.3 has introduced the idea of storing the random numbers created to simulate the Brownian increments in system memory instead of regenerating them every time they are needed. Table 7.6 shows a calibration run with regenerated random numbers every time they are required on the left side and with stored random numbers on the right side. It can be observed, that the iterations run is

Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$	Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$
0	$3.9390e+00$	$1.1704e+00$	0	$3.9390e+00$	$1.1704e+00$
1	$3.6096e+00$	$8.4981e-01$	1	$3.6096e+00$	$8.4981e-01$
2	$3.3842e+00$	$6.7835e-01$	2	$3.3842e+00$	$6.7835e-01$
3	$2.9785e+00$	$5.4147e-01$	3	$2.9785e+00$	$5.4147e-01$
4	$2.5218e+00$	$4.1864e-01$	4	$2.5218e+00$	$4.1864e-01$
5	$2.0573e+00$	$3.1395e-01$	5	$2.0573e+00$	$3.1395e-01$
⋮	⋮	⋮	⋮	⋮	⋮
29	$1.1177e-05$	$3.3574e-05$	29	$1.1177e-05$	$3.3574e-05$
30	$5.6682e-06$	$3.3574e-05$	30	$5.6682e-06$	$3.3574e-05$
31	$4.2508e-06$	$3.3574e-05$	31	$4.2508e-06$	$3.3574e-05$
32	$2.1867e-06$	$3.3574e-05$	32	$2.1867e-06$	$3.3574e-05$
33	$1.6352e-06$	$3.3574e-05$	33	$1.6352e-06$	$3.3574e-05$
34	$8.4909e-07$	$3.3574e-05$	34	$8.4909e-07$	$3.3574e-05$

Table 7.6: Iterations tabular for calibration with stored random numbers (right side) and with regenerated random numbers (left side) with M=10,000, $\Delta t = 5\times10^{-2}$ and $\epsilon = 1\times10^{-2}$.

identical, as it has been expected.

Certainly, this behavior cannot be observed for a calibration with adjoint equation in comparison to a finite difference based optimization, as shown in tabular 7.7. Nevertheless the iterations tabulars show virtually no difference. Both optimizations converge up to a LSQ value of 3.44×10^{-5} and a 2-norm of the Lagrangian of 2.37×10^{-6}.

Furthermore the resulting solutions, displayed in table 7.8, are identical. Thus, both methods lead to the same solution in the same number of iterations, as it would have been desired.

Table 7.9 illustrates the iterations tabular of a straight forward calibration with M=100,000, $\Delta t = 5\times10^{-3}$ and $\epsilon = 3.1\times10^{-3}$ on the left side taking 40 iterations to find a solution. The right side of this table indeed shows the iterations tabular of a multi layer calibration on 3 layers ($M|\Delta t|\epsilon$), namely $(1,000|5\times10^{-2}|3.1\times10^{-3})$, $(10,000|5\times10^{-2}|3.1\times10^{-3})$ and $(100,000|5\times10^{-2}|3.1\times10^{-3})$. It can be observed, that the optimization on the coarser layers lead to a better starting value for the

Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$	Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$
0	$4.0916e+00$	$1.2075e+00$	0	$4.0916e+00$	$1.2075e+00$
1	$3.7016e+00$	$8.4380e-01$	1	$3.7016e+00$	$8.4380e-01$
2	$3.4594e+00$	$6.9289e-01$	2	$3.4594e+00$	$6.9289e-01$
3	$3.0594e+00$	$5.5514e-01$	3	$3.0594e+00$	$5.5514e-01$
4	$2.6214e+00$	$4.3286e-01$	4	$2.6214e+00$	$4.3286e-01$
5	$2.1781e+00$	$3.2719e-01$	5	$2.1781e+00$	$3.2719e-01$
⋮	⋮	⋮	⋮	⋮	⋮
35	$8.3016e-02$	$1.8364e-04$	35	$8.3016e-02$	$1.8364e-04$
36	$2.0367e-02$	$4.2407e-05$	36	$2.0367e-02$	$4.2407e-05$
37	$2.1715e-03$	$3.4529e-05$	37	$2.1716e-03$	$3.4529e-05$
38	$2.0160e-04$	$3.4427e-05$	38	$2.0165e-04$	$3.4427e-05$
39	$6.7396e-06$	$3.4426e-05$	39	$6.7406e-06$	$3.4426e-05$
40	$2.3719e-07$	$3.4426e-05$	40	$2.3695e-07$	$3.4426e-05$

Table 7.7: Iterations tabular for calibration with gradient evaluation via finite differences (left side) compared to adjoint equation (right side) with M=100,000, $\Delta t = 5\times10^{-3}$ and $\epsilon = 3.1\times10^{-3}$.

x	Fin. Diff.	Adjoint
κ	1.71931	1.71931
θ	0.03253	0.03253
σ	3.04830	3.04828
ρ	-0.72567	-0.72567
v_0	0.01288	0.01288

Table 7.8: Calibration results from a finite differences based optimization in comparison to an adjoint based optimization with M=100,000, $\Delta t = 5\times10^{-3}$ and $\epsilon = 3.1\times10^{-3}$.

finer layers such that finally only 5 iterations on the finest layer are required instead of 40 as in the single layer example. This leads to shorter calibration times, as it will be shown subsequently.

Indeed, table 7.10 now shows calculation times, for combinations of techniques introduced in chapter 5 and chapter 6 measured in hours, minutes and seconds (hh:mm:ss). Initially, the idea of storing random numbers instead of regenerating them has been applied. In section 6.3, where it has been introduced, the speed up per function evaluation was 2.5 if all numbers fit into memory (table 6.1 on page 96). Certainly, this speed up is only a limit of the possible speed up realized during calibration as the numbers are generated and stored during the first function evaluation. Thus, the more iterations the optimization takes, the closer will the speed up be to 2.5. In the concrete test example the calibration speed up lies between 1.7 and 2.0. For example a time constant calibration with $M = 100,000$, $\Delta t = 5\times10^{-3}$ and $\epsilon = 3.1\times10^{-3}$ could be reduced from 01:30 hours to 50 minutes.

Considering the results with an additional application of the adjoint equation one can observe that in the time constant case the calculation time increases slightly which is contrary to intuition. This is due to the fact, that storing the random

Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$	Iter.	$\|\nabla_x L(x)\|_2$	$\|R(x)\|_2^2$
0	$4.0916e+00$	$1.2075e+00$	$M = 1,000$ $\Delta t = 5{\times}10^{-3}$ $\epsilon = 3.1{\times}10^{-3}$		
1	$3.7016e+00$	$8.4380e-01$	0	$2.8285e+00$	$1.8954e+00$
2	$3.4593e+00$	$6.9290e-01$	1	$2.3471e+00$	$1.0302e+00$
3	$3.0593e+00$	$5.5515e-01$	2	$1.8970e+00$	$5.7040e-01$
4	$2.6213e+00$	$4.3286e-01$	⋮	⋮	⋮
5	$2.1780e+00$	$3.2719e-01$	23	$1.2012e-02$	$1.6107e-04$
6	$1.7687e+00$	$2.3730e-01$	24	$5.6549e-04$	$1.5776e-04$
7	$1.4013e+00$	$1.6212e-01$	25	$8.9727e-06$	$1.5775e-04$
8	$1.0772e+00$	$1.0125e-01$	$M = 10,000$ $\Delta t = 5{\times}10^{-3}$ $\epsilon = 3.1{\times}10^{-3}$		
9	$7.9669e-01$	$5.3958e-02$	0	$1.9121e-02$	$2.0904e-04$
10	$2.9423e-01$	$6.0677e-03$	1	$2.3572e-02$	$5.3147e-05$
11	$2.3452e-02$	$6.8155e-04$	2	$2.1980e-02$	$4.7663e-05$
⋮	⋮	⋮	⋮	⋮	⋮
33	$7.5276e-02$	$2.5595e-04$	7	$9.2068e-06$	$2.9952e-05$
34	$7.8941e-02$	$2.2516e-04$	8	$2.3848e-06$	$2.9951e-05$
35	$8.2992e-02$	$1.8358e-04$	9	$4.4286e-07$	$2.9951e-05$
36	$2.0354e-02$	$4.2400e-05$	$M = 100,000$ $\Delta t = 5{\times}10^{-3}$ $\epsilon = 3.1{\times}10^{-3}$		
37	$2.1726e-03$	$3.4529e-05$	0	$5.6441e-02$	$1.0264e-04$
38	$2.0192e-04$	$3.4427e-05$	1	$2.1337e-02$	$4.4687e-05$
39	$6.8887e-06$	$3.4426e-05$	2	$1.0369e-03$	$3.4462e-05$

Table 7.9: Iterations tabular for calibration on 1 Monte Carlo layer with M=100,000, $5{\times}10^{-3}$ and $\epsilon = 3.1{\times}10^{-3}$ (left side) and on 3 layers ($M|\Delta t|\epsilon$), namely $(1,000|5{\times}10^{-2}|3.1{\times}10^{-3})$, $(10,000|5{\times}10^{-2}|3.1{\times}10^{-3})$ and $(100,000|5{\times}10^{-2}|3.1{\times}10^{-3})$ (right side).

numbers is more effective in the finite difference than in the adjoint case as it has already been described after Remark 5.10. This is also reflected by the gradient evaluation time with stored random numbers, which is illustrated in table 7.11. The finite difference method becomes relatively faster in comparison to the results in table 5.2 if the random numbers are stored. Nevertheless, the full speed up of the adjoint equation becomes obvious with an increasing number of parameters through time dependency on e.g. B=10 subintervals. In this example, the adjoint calculation is approximately 4 times faster, which matches the factor in table 7.11.

As it could have been expected from table 7.9, the multi layer method significantly accelerates the calibration. Considering for instance the time dependent case with M=100,000, $\Delta t = 5 \times 10^{-3}$ and $\epsilon = 3.1 \times 10^{-3}$, the calibration time has been reduced from 1:34 hours to 12 minutes.

As a final step, the parallel computation on 2 CPUs further speeds up the optimization process. The concrete acceleration factor varies from test to test, as the parallelization changes the calibration run. Nevertheless, for instance the test calibration with M=100,000, $\Delta t = 5{\times}10^{-3}$ and $\epsilon = 3.1{\times}10^{-3}$ could be reduced from 12 minutes to 6 in the time dependent case or from 1 minute to 5 seconds in the time constant case.

So far, all speed ups can be achieved on a standard Dual Core Desktop PC.

Methods	M=1,000 $\Delta t = 5\times10^{-1}$ $\epsilon = 3.1\times10^{-2}$	M=10,000 $\Delta t = 5\times10^{-2}$ $\epsilon = 1\times10^{-2}$	M=100,000 $\Delta t = 5\times10^{-3}$ $\epsilon = 3.1\times10^{-3}$	M=1,000,000 $\Delta t = 5\times10^{-4}$ $\epsilon = 1\times10^{-3}$
Time Constant: B=1				
Plain MC	00:00:01	00:01:02	01:30:10	125:41:14
+Storing	00:00:01	00:00:37	00:49:25	125:15:25
+Adjoint	00:00:01	00:00:54	01:34:48	121:40:21
+Multi Layer	00:00:01	00:00:25	00:12:19	10:56:12
+2CPUs	00:00:01	00:00:05	00:06:07	7:38:46
Time Dependent: B=10				
Plain MC	00:01:38	00:11:17	05:31:33	245:51:37
+Storing	00:01:21	00:06:42	02:28:30	245:17:12
+Adjoint	00:00:26	00:01:41	00:49:44	81:38:25
+Multi Layer	00:00:08	00:00:56	00:34:55	18:20:15
+2CPUs	00:00:01	00:00:48	00:10:36	14:17:59

Table 7.10: Calculation Times for several Monte Carlo grids with time constant and time dependent parameters on B=10 subintervals and combinations of different methods to speed up the calibration.

B	P	Fin. Diff.	Adjoint	Ratio
1	5	7	9	0.8
2	9	13	10	1.3
3	13	18	10	1.8
4	17	23	11	2.1
5	21	28	12	2.3
6	25	33	12	2.8
7	29	39	12	3.2
8	33	44	13	3.4
9	37	49	13	3.8
10	41	54	14	3.9

Table 7.11: Computing time in seconds for one gradient evaluation via adjoint equation compared to the finite differences scheme for B subintervals or P parameters with $M = 100,000$, $\Delta t = 5 \times 10^{-2}$ and $\epsilon = 3.1 \times 10^{-3}$ and stored random numbers.

Summarizing a time constant calibration with 100,000 simulations could be accelerated from 1.5 hours to 6 minutes or from 5.5 hours to 10 minutes in the time dependent case.

Chapter 8

Extension to Jump Diffusion

Indeed it has been shown so far that adjoints significantly speedup the Monte Carlo calibration of financial market models in a diffusion setting (3.1). However, the method presented in chapter 5 is not immediately applicable if one leaves the model class of diffusion processes and allows for the possibility of jumps. In this setting stock price paths and hence the payoff of the standard options C_i may be not differentiable with respect to parameters like e.g. jump probabilities. On first sight this seems to prevent the application of adjoints in the presence of jump diffusions. However, as it will be seen in the remainder of this section, a suitable decomposition of the sensitivity calculation into the diffusion and jump part may provide the desired smoothness and consequently allows a significant adjoint-based calibration speedup.

8.1 The Bates Model

Without loss of generality the analysis in the following will focus on the Bates model (Bates [1996]). This model has been chosen since it admits a semi-closed form solution for plain vanilla options — just like the Stein-Stein model — which serves as a good benchmark for the sensitivities. Within the Bates model the stock price $(S_t)_t$ under the risk-neutral measure is driven by the stochastic differential equations

$$dS_t = (r - \delta - \lambda_J \beta) S_t dt + \phi \sqrt{v_t} S_t dW_t^1 + S_t d \sum_{d=1}^{D_t} V_j \tag{8.1}$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma \sqrt{v_t}(\rho dW_t^1 + \sqrt{1-\rho^2} dW_t^2) \tag{8.2}$$

where in comparison to the models introduced in chapter 7 ϕ serves as a multiplier that absorbs the initial condition of the variance process. In addition the model allows for the random number of D_t independent jumps V_j up to time t with

lognormal distribution $\ln(1+V_j) \sim N(\mu_J, \sigma_J^2)$, where $(D_t)_t$ denotes an independent Poisson process with intensity λ_J and $\beta = \exp(\mu_J + \sigma_J^2/2) - 1$ is a drift correction factor.

The calibration problem is now composed of choosing the model parameters

$$x = (\kappa, \theta, \sigma, \rho, \phi, \lambda_J, \mu_J, \sigma_J) \tag{8.3}$$

in a suitable set $X \subset \mathbb{R}^8$ such that

$$\begin{aligned}
& \min_{x\in X} f(x) := \sum_{i=1}^{I} \left(C^i(x) - C^i_{\text{obs}}\right)^2 \\
\text{where} \quad & C^i(x) = e^{-rT_i} E_Q\left(\max(S_{T_i}(x) - K_i, 0)\right) \\
\text{s.t.} \quad & dS_t = (r - \delta - \lambda_J\beta)S_t dt + \phi\sqrt{v_t}S_t dW_t^1 + S_t d\sum_{d=1}^{D_t} V_j \\
& dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t}(\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2).
\end{aligned}$$

A combination of Monte Carlo, EMS and smoothing nondifferentiabilities together with a separation of the jump and diffusion with the help of Ito's formula (Theorem 2.9) part leads to the approximation

$$\begin{aligned}
& \min_{x\in X} f_{M,\Delta t,\epsilon}(x) := \sum_{i=1}^{I} \left(C^i_{M,\Delta t,\epsilon}(x) - C^i_{\text{obs}}\right)^2 \\
\text{where} \quad & C^i_{M,\Delta t,\epsilon}(x) := e^{-rT_i}\frac{1}{M}\sum_{m=1}^{M}\left(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i)\right) \\
\text{s.t.} \quad & s^m_{N_i,\epsilon} = u^m_{N_i,\epsilon} e^{\sum_{d=1}^{D^m_{N_i}} \mu_J + \sigma_J Z^m_d}, \; D^m_{N_{i+1}} = D^m_{N_i} + F_i^{-1}(U^m_i) \\
& u^m_{n+1,\epsilon} = u^m_{n,\epsilon} + (r - \delta - \lambda_J\beta)u^m_{n,\epsilon}\Delta t_n + \phi\sqrt{\pi_\epsilon(v^m_{n,\epsilon})}u^m_{n,\epsilon}\Delta W_n^{1,m} \\
& v^m_{n+1,\epsilon} = v^m_{n,\epsilon} + \kappa(\theta - \pi_\epsilon(v^m_{n,\epsilon}))\Delta t_n \\
& \qquad + \sigma\sqrt{\pi_\epsilon(v^m_{n,\epsilon})}\left(\rho\Delta W_n^{1,m} + \sqrt{1-\rho^2}\Delta W_n^{2,m}\right).
\end{aligned}$$

Here, $s^m_{N_i}$ denotes the approximation of S_{T_i} in the m-th path, which is computed in two steps. First the pure diffusion process denoted by u^m_n, v^m_n is simulated on a small step time grid $t_0 = \tau_0 < ... < \tau_N = T$. Secondly, the independent jump term is added for each of the standard option maturities T_i based on a large step simulation of the Poisson process $(D_t)_t$ on the time intervals $(T_{N_i}, T_{N_{i+1}})$. This large step simulation can be obtained by drawing independent uniform random numbers U^m_i and plugging them into the inverse of the distribution function $F_i(\cdot)$ associated with the probability law

$$Q(D_{T_{i+1}} - D_{T_i} = d) = \frac{(\lambda_J(T_{i+1} - T_i))^d}{d!}e^{-\lambda_J(T_{i+1}-T_i)}.$$

The relative size of each of the $D^m_{N_i}$ jumps is determined as $e^{\mu_J + \sigma_J Z^m_d}$ with indepen-

dent $N(0,1)$-distributed deviates Z_d^m.

Unfortunately the cumulative distribution function F_i is not continuous and thus not differentiable. This makes a calibration of the model with efficient algorithms very hard if not impossible. However, to avoid these problems one can make use of a reformulation of the call price functional based on the independence of the jump and diffusion part in the following way:

$$\begin{aligned} C^i(x) &= e^{-rT_i} E_Q\left(\sum_{d=0}^{\infty} \max(S_{T_i}(x) - K_i, 0) 1_{\{D_{T_i}=d\}}\right) \\ &= e^{-rT_i} \sum_{d=0}^{\infty} Q(D_{T_i} = d) E_Q\left(\max(S_{T_i}(x) - K_i), 0 | D_{T_i} = d\right). \end{aligned}$$

Since the probabilities $Q(D_{T_i} = d)$ quickly converge to zero, the first few summands of this series approximate their limit very well. Exploiting this idea in combination with a smoothing of the maximum function leads to the following model reformulation:

$$\begin{aligned} C^i_{M,\Delta t,\epsilon}(x) &= e^{-rT_i} \sum_{d=0}^{\overline{d}} Q(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \pi_\epsilon(s^{m,d}_{N_i,\epsilon}(x) - K_i) \\ s^{m,d}_{N_i,\epsilon} &= u^m_{N_i,\epsilon} e^{\sum_{\nu=1}^{d} \mu_J + \sigma_J Z^m_\nu} \\ u^m_{n+1,\epsilon} &= u^m_{n,\epsilon} + (r - \delta - \lambda_J \beta) u^m_{n,\epsilon} \Delta t_n + \phi \sqrt{\pi_\epsilon(v^m_{n,\epsilon})} u^m_{n,\epsilon} \Delta W^{1,m}_n \\ v^m_{n+1,\epsilon} &= v^m_{n,\epsilon} + \kappa(\theta - \pi_\epsilon(v^m_{n,\epsilon}))\Delta t_n \\ &+ \sigma \sqrt{\pi_\epsilon(v^m_{n,\epsilon})} \left(\rho \Delta W^{1,m}_n + \sqrt{1-\rho^2} \Delta W^{2,m}_n\right). \end{aligned}$$

Since the probabilities $Q(D_{T_i} = d)$ are smooth with respect to the jump intensity, one can conclude that this Monte Carlo estimator is continuously differentiable with respect to all model parameters x_i defined in (8.3). This allows to apply for instance the line-search SQP method from algorithm 4 for the solution of the calibration problem.

8.2 Adjoint Equation

Furthermore, the gradient of the objective function can efficiently be computed with the help of the adjoint method introduced in Theorem 5.6.

Theorem 8.1. *Let $R : \mathbb{R}^P \to \mathbb{R}^I$ be the vector valued function*

$$R(x) = \left[e^{-rT_i} \sum_{d=0}^{\overline{d}} Q(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \pi_\epsilon(s^{m,d}_{N_i}(x) - K_i) - C^i_{\text{obs}} \right]_{i=1}^{I}$$

with

$$\begin{aligned} s^{m,d}_{N_i,\epsilon} &= u^m_{N_i,\epsilon} e^{\sum_{\nu=1}^{d} \mu_J + \sigma_J Z^m_\nu} \\ u^m_{n+1,\epsilon} &= u^m_{n,\epsilon} + (r - \delta - \lambda_J \beta) u^m_{n,\epsilon} \Delta t_n + \phi \sqrt{\pi_\epsilon(v^m_{n,\epsilon})} u^m_{n,\epsilon} \Delta W^{1,m}_n \\ v^m_{n+1,\epsilon} &= v^m_{n,\epsilon} + \kappa(\theta - \pi_\epsilon(v^m_{n,\epsilon}))\Delta t_n \\ &+ \sigma \sqrt{\pi_\epsilon(v^m_{n,\epsilon})} \left(\rho \Delta W^{1,m}_n + \sqrt{1-\rho^2} \Delta W^{2,m}_n \right). \end{aligned}$$

Setting the vector and matrix-valued maps $a_\epsilon : X \times \mathbb{R}^2 \to \mathbb{R}^2$, $b_\epsilon : X \times \mathbb{R}^2 \to \mathbb{R}^2 \times \mathbb{R}^2$,

$$\begin{aligned} a_\epsilon(x,y) &:= \begin{pmatrix} (r - \delta - \lambda_J \beta) y_1 \\ \kappa(\theta - \pi_\epsilon(y_2)) \end{pmatrix} \\ b_\epsilon(x,y) &:= \begin{pmatrix} \phi \sqrt{\pi_\epsilon(y_2)} y_1 & 0 \\ \sigma \sqrt{\pi_\epsilon(y_2)} \rho & \sigma \sqrt{\pi_\epsilon(y_2)(1-\rho^2)} \end{pmatrix}, \end{aligned}$$

the derivative of R_i *can be computed via*

$$R_i'(x) = \frac{e^{-rT_i}}{M} \sum_{m=1}^{M} \sum_{n=0}^{N_i - 1} (\lambda^{m,i}_{n+1})^T \left[\frac{\partial}{\partial x} a_\epsilon(x, y^m_n) \Delta t_n + \frac{\partial}{\partial x} \left(b_\epsilon(x, y^m_n) \Delta W^m_n \right) \right]$$

where $\lambda^{m,i}_n \in \mathbb{R}^L$ *results from the adjoint equation*

$$\begin{aligned} &\lambda^{m,i}_n = \left[I + \tfrac{\partial}{\partial y} a_\epsilon(x, y^m_n) \Delta t_n + \tfrac{\partial}{\partial y} \left(b_\epsilon(x, y^m_n) \Delta W^m_n \right) \right]^T \lambda^{m,i}_{n+1}, \\ &n = N_i - 1, N_i - 2, ..., 1, \quad m = 1, ..., M, \\ &\lambda^{m,i}_{N_i} = \left[\textstyle\sum_{d=0}^{\overline{d}} Q(D_{T_i} = d) \left(\pi_\epsilon'(s^m_{N_i}(x) - K) \right), 0, ..., 0 \right] \in \mathbb{R}^L. \end{aligned} \tag{8.4}$$

Proof. Reconsidering that $\xi^m_{N_i}$ is the first component of $\eta^m_{N_i}$, the final condition in (8.4) allows in analogy to the proof of Theorem 5.6 for

$$\begin{aligned} R_i'(x)\Delta x &= e^{-rT_i} \left(\sum_{d=0}^{\overline{d}} Q(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \right) \right)' \\ &= e^{-rT_i} \sum_{d=0}^{\overline{d}} Q'(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \right) \\ &+ e^{-rT_i} \sum_{d=0}^{\overline{d}} Q(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon'(s^m_{N_i,\epsilon}(x) - K_i) \right) \xi^m_{N_i} \end{aligned}$$

$$
\begin{aligned}
&= e^{-rT_i} \sum_{d=0}^{\overline{d}} Q'(D_{T_i} = d) \frac{1}{M} \sum_{m=1}^{M} \left(\pi_\epsilon(s^m_{N_i,\epsilon}(x) - K_i) \right) \\
&+ e^{-rT_i} \frac{1}{M} \sum_{m=1}^{M} (\lambda^{m,i}_{N_i})^T \eta^m_{N_i}
\end{aligned}
$$

which proves the statement. □

8.3 Numerical Results

This section analyzes the speedup that can be achieved for a gradient evaluation based on the adjoint equation in Theorem 8.1. Just like in section 7.3, a variant of the above introduced Bates model (8.1) with lognormal distribution of the variance and piecewise constant mean reversion speed κ_t, mean reversion level θ_t, volatility of variance σ_t and correlation ρ_t (see also (5.13) on page 80) is introduced, to illustrate the flexibility of the framework.

$$
\begin{aligned}
dS_t &= (r - \delta - \lambda_J \beta) S_t dt + \phi \sqrt{v_t} S_t dW^1_t + S_t d \sum_{d=1}^{D_t} X_j \\
dv_t &= \kappa_t(\theta_t - v_t) dt + \sigma_t \sqrt{v_t} \left(\rho_t dW^1_t + \sqrt{1 - \rho_t^2} dW^2_t \right).
\end{aligned}
$$

Again, though the model dynamics have only been slightly changed to the lognormal distribution of the variance, this now prevents the derivation of a semi-closed form solution for the price of standard calls, such that approximation methods like the one introduced in this thesis are required.

B	P	Fin. Diff.	Adjoint	Ratio
1	8	25	9	2.8
2	12	43	9	4.8
3	16	60	10	6.0
4	20	77	11	7.0
5	24	94	12	7.8
6	28	111	13	8.5
7	32	128	13	9.8
8	36	146	14	10.4
9	40	163	15	10.9
10	44	180	16	11.2

Table 8.1: Computing time in seconds for one gradient evaluation via adjoint equation compared to the finite differences scheme for B subintervals or P parameters with $M = 100,000$, $\Delta t = 5 \times 10^{-2}$ and $\epsilon = 3.1 \times 10^{-3}$.

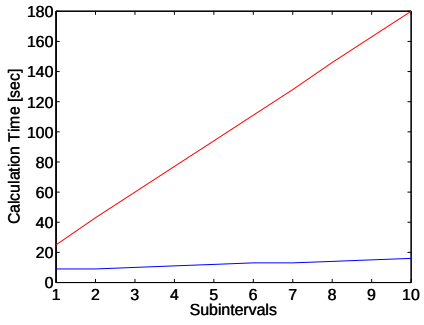

Figure 8.1: Computing time in seconds for one gradient evaluation via adjoint equation (blue line) compared to the finite differences scheme (red line) with $M = 100,000$, $\Delta t = 10^{-3}$ and $\epsilon = 10^{-3}$.

Table 8.1 and figure 8.1 illustrate the computation time for a gradient evaluation

via adjoints compared to a simple finite difference approximation for the parameter vector defined in (8.3)

As expected, the computation time for the gradient evaluation via finite differences grows linearly in the number of model parameters. Thus each iteration of an optimization algorithm solving the calibration problem will become more and more costly as one increases the number of model parameters. In contrast the adjoint framework for the jump diffusion process leads to stable computation times that are nearly independent of the number of model parameters.

Chapter 9

Conclusions

9.1 Summary

A calibration problem for financial market models based on Monte Carlo simulation and discretization of the underlying stochastic differential equation with an Euler-Maruyama scheme has been introduced. As it is desirable to benefit from fast deterministic optimization methods to solve the arising optimization problem, possible non-differentiabilities have been smoothed out with a twice continuously differentiable polynomial. On the basis of the so derived calibration problem, this work was essentially concerned about two issues.

First, it could have been shown, that a sequence of computed stationary points of the sample average approximation problem, derived by increasing the number of Monte Carlo simulations and reducing the discretization step size and the smoothing parameter, converges to a solution of the true problem in the sense of a first order critical point. To show this, initially a pointwise convergence of the two objective's has been shown via a decomposition of the overall approximation error into the Monte Carlo, the discretization and the smoothing error. This result, together with an epicontinuity proof allowed to show a uniform convergence of the approximating and the true objective functions on the feasible set. As a last step, a similar result on the objective's gradients facilitates the optimality proof. In particular, this proof was based on assumptions like the Lipschitz continuity of the SDE coefficient functions. These theoretical results haven been determined by numerical results in chapter 7 for the benchmark example of the 2 dimensional Stein Stein model. Additionally, the theoretically proven convergence order of $\mathcal{O}(1/\sqrt{M} + \Delta t^{\frac{1}{2}} + \epsilon)$ has been confirmed in numerical tests.

The second main task of this work was to speed up the Monte Carlo calibration as computation times of several hours or even days occurring without any special effort are not feasible for practical applications. It turns out that a calculation of

the objective's gradient via an adjoint equation provides a noticeable reduction of the computational effort in comparison to the frequently chosen finite difference method. In particular, this method is independent on an increasing number of parameters, when they are chosen to be piecewise constant on several intervals. In comparison to this, the complexity of a finite difference approximation scale linear in the number of parameters. Thus, the derived speed up scaled at an almost linear rate. Furthermore, this adjoint method yields the exact gradient and thus stabilizes the calibration process. Moreover, several other techniques have been introduced throughout this thesis, that enhance the efficiency of the optimization algorithm. A Multi Layer technique, i.e. starting on a coarse Monte Carlo layer and increasing accuracy during optimization, was very effective in the case, that the chosen initial value is not already close to the solution. Storing instead of regenerating the random numbers required for the Brownian increments in the SDE led to a further speed up. Finally, the parallelization of the option price evaluation proved itself to be very well suited for a parallelization. In particular a combination of this techniques yields a reduction from e.g. 1.5 hours to 6 minutes in the time constant parameters case with $M = 100,000$, $\Delta t = 5\times10^{-3}$ and $\epsilon = 3.1\times10^{-3}$ or from 5.5 hours to 10 minutes for the same setting but with parameters chosen to be time dependent on 10 intervals, which is a significant reduction of the computation time, especially as this speed up could have been achieved on a standard Desktop PC.

9.2 Future Work

The tendency for higher dimensional and more complicated models already described in the introduction leads to stronger requirements on the chosen methods to approximate the corresponding solution. The presented thesis is one step on this road, but nevertheless additional effort reduction techniques are desirable.

Firstly, the fact that storing the random numbers leads to a speed up of the Monte Carlo simulation motivates to concentrate for instance on this part of the option price evaluation, namely the random number generation. A frequently applied method is the so called *Quasi Monte Carlo* where the idea is to replace the pseudo random number generator by an alternative one, which allows for fewer simulations to achieve a certain accuracy. This idea is for instance introduced in Packham and Schmidt [2009] or Glasserman [2003].

Another technique available in the literature, that turned out to be very efficient for option pricing, is the *Multi Level Monte Carlo* introduced in Giles [2006]. This method calculates the expected value of the options future payoff in a telescope sum on several levels, similar to the layers introduced in section 6.2, such that paths with large errors eliminate each other. Additionally the number of Monte Carlo simulations is calculated in dependence of the estimators variance. An immediate

implementation of this method in the presented calibration algorithm could lead to instabilities, as changing parameters would lead to changing levels and number of Monte Carlo simulations which means changing the objective function in every iteration. However, keeping the levels identical during several sequenced iterations, maybe in combination with a Multi Layer approach, should lead to a significant reduction of the calibration time.

Finally, the parallelization of the Monte Carlo option price evaluation was very effective. Thus, increasing the number of processors raises the expectation of a strong decrease in calibration time. In this manner, the parallelization on graphics cards (GPUs) is a hot topic. First tests showed an incredible speed up but also led to problems with single precision while calculating the finite difference approximation of the gradient. This is due to the fact that most of the GPUs are not capable for double precision. Consequently, the implementation of an adjoint equation could prove itself to be helpful.

List of Figures

List of Tables

Bibliography

L. Andersen. Efficient simulation of the heston stochastic volatility model. working paper, January 2007. URL `http://ssrn.com/abstract=946405`.

L. Andersen and R. Brotherton-Ratcliffe. The equity option volatility smile: an implicit finite-difference approach. *The Journal of Computational Finance*, 1(2): 5–38, 1997/1998.

L. Arnold. *Stochastische Differentialgleichungen*. R. Oldenbourg Verlag, 1973.

F. Bastin, C. Cirillo, and P.L. Toint. Convergence theory for nonconvex stochastic programming with an application to mixed logit. *Mathematical Programming Series B*, 108:207–234, 2006. doi: $10.1007/\mathrm{s}10107\text{-}006\text{-}0708\text{-}6$.

S.D. Bates. Jumps and stochastic volatility: Exchange rate process implicit in deutsche mark options. *The Review of Financial Studies*, 9(1):69–107, Spring 1996.

H. Bauer. *Wahrscheinlichkeitstheorie*. De-Gruyter Lehrbuch. Walter de Gruyter, 5th edition, 2002.

H. Bauer. *Maß- und Integrationstheorie*. de Gruyter, 1992.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

P.T. Boggs. *Sequential Quadratic Programming*, volume 4 of *ACTA Numerica 1995*, pages 1–52. Cambridge University Press, 1995.

J.B. Bonnans, J.C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization*. Springer, 2003.

M. Broadie and P. Glasserman. Estimating security price derivatives using simulation. *Managment Science*, 42(2):269–285, 1996.

M. Broadie and O. Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54(2):217–231, 2006.

T.F. Coleman, Y. Li, and A. Verma. Reconstructing the unknown local volatility function. *Journal of Computational Finance*, 2(3):77–102, 1999.

A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust Region Methods*. MPS-SIAM Series on Optimization. SIAM, 2000.

J.C. Cox. The constant elasticity of variance option pricing model. *Journal of Portfolio Management*, 23(1):15–17, December 1996. Special Issue.

J.C. Cox, J.E. Ingersoll, and S.A. Ross. A theory of the terms structure of interest rates. *Econometrica*, 53:385–408, 1985.

G. Deelstra and F. Delbaen. Convergence of discretized stochastic (interest rate) processes with stochastic drift term. *Applied Stochastic Models and Data Analysis*, 14:77–84, 1998.

D. Duffie and P. Glynn. Efficient monte carlo simulation of security prices. *The Annals of Applied Probability*, 5(4):897–905, 1995.

B. Dupire. Pricing with a smile. *Risk*, pages 18–20, 1994.

W. Feller. Two singular diffusion problems. *Annals of Mathematics*, 54:173–182, 1951.

W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, 1970a.

W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, 1970b.

A. Forsgren, P.E. Gill, and M.H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44:525–597, 2002.

O. Forster. *Analysis I*. Vieweg, 1999.

J. Gatheral. Case studies in financial modelling course notes, Fall Term 2004.

C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer, 2002.

J.E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, second edition, 2003.

F. Gerlich, A.M. Giese, J.H. Maruhn, and E.W. Sachs. Parameter identification in stochastic volatility models with time-dependent model parameters. Technical report, Universität Trier, 2006.

A.M. Giese, C. Käbe, J.H. Maruhn, and E.W. Sachs. Efficient calibration for problems in option pricing. *PAMM - Proceedings in Applied Mathematics and Mechanics*, 7(1):1062601–1062602, 2007. doi: $10.1002/\mathrm{pamm}.200701141$. URL `http://www3.interscience.wiley.com/journal/122394337/abstract`.

M.B. Giles. Monte carlo evaluation of sensitivities in computational finance. In *HERCMA - The 8th Hellenic European Research on Computer Mathematics & its Applications Conference*. ACM Digital Library, 2007.

M.B. Giles. Multi-level monte carlo path simulation. Technical Report 06/03, Oxford University Computing Laboratory, 2006.

M.B. Giles and P. Glasserman. Smoking adjoints: Fast monte carlo greeks. Risk Technical Papers, January 2006.

P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.

A. Griewank and G.F. Corlis, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, Philadelphia, 1991. SIAM.

P.S. Hagan, D. Kumar, A.S. Lesniewski, and D.E. Woodward. Managing smile risk. *Wilmott Magazine*, 1:84–108, 2002.

B. Hamida and R. Cont. Recovering volatility from option prices by evolutionary optimization. *Journal of Computational Finance*, 8(4), Summer 2005.

S.L. Heston. A closed-form solution for options with stochastic volatitlity with applications to bond and currency options. *The Review of Financial Studies*, 6 (2):327–343, 1993.

J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 62(2):281–300, Juni 1987.

C. Käbe, J.H. Maruhn, and E.W. Sachs. Adjoint based monte carlo calibration of financial market models. *Journal of Finance and Stochastics*, 13(3):351–379, 2009. doi: $10.1007/\mathrm{s00780\text{-}009\text{-}0097\text{-}9}$. URL `http://www.springerlink.com/content/j27q00u581r01l8p/`.

E. Karatzas and S.E. Shreve. *Methods of Mathematical Finance*, volume 39 of *Applications of Mathematics (New York)*. Springer Verlag, New York, 1998.

I. Karatzas and S.E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 1991.

F. Kilin. Accelerating the calibration of stochastic volatility models. Technical Report 6, Frankfurt School of Finance & Managment, May 2007.

S. Kindermann, P. Mayer, H. Albrecher, and H. Engl. Identification of the local speed function in a levy model for option pricing. *Journal of Integral Equations and Applications*, 20(2):161–200, 2008.

P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 3rd edition, 1999.

O.A. Ladyzenskaja, V.A. Solonikov, and N.N. Uralceva. *Linear and quasilinear equations of parabolic type*. American Mathematical Society, Translations of Mathematical Monographs, 1968.

A.L. Lewis. *Option Valutation under Stochastic Volatility*. Finance Press, March 2000.

R. Lord, R. Koekkoek, and D. Dijk van. A comparison of biased simulation schemes for stochastic volatility models. *Tinbergen Institute Discussion Papers*, 2006.

Xuerong Mao, Aubrey Truman, and Chenggui Yuan. Euler-maruyama approximations in mean-reverting stochastic volatility model under regime-switching. *Journal of Applied Mathematics and Stochastic Analysis*, pages 1–20, 2006. doi: $10.155/\mathrm{JAMSA}/2006/80967$.

S. Mikhailov and U. Nögel. Heston's stochastic volatility model implementation, calibration and some extensions. In P. Wilmott, editor, *The Best of Wilmott 1: Incorporating the Quantitative Finance Review*, pages 401–412. Wilmott, P., 2004.

V. Mikulevicius and E. Platen. Rate of convergence of the euler approximation for diffusion processes. *Mathematische Nachrichten*, 151:233–239, 1991.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

N. Packham and W. Schmidt. Latin hypercube sampling with dependence and applications in finance. *Journal of Computational Finance*, 2009. (accepted).

S.M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21(3):513–528, August 1996.

R.Y. Rubinstein and A. Shapiro. *Discrete Event Systems*. John Wiley, 1993.

E.W. Sachs and M. Schu. Reduced order models (pod) for calibration problems in finance. In K. Kunisch, G. Of, and O. Steinbach, editors, *Proceedings of ENUMATH 2007, the 7th European Conference on Numerical Mathematics and Advanced Applications, Graz, Austria, September 2007*, Numerical Mathematics and Advanced Applications, pages 735–742, September 2007.

E.W. Sachs and A.K. Strauss. Efficient solution of a partial integro-differential equation in finance. *Applied Numerical Mathematics*, 58(58):1687–1703, 2008.

L.R. Scott, T. Clark, and Bagheri.B. *Scientific parallel computing*. Princeton University Press, 2005.

A. Shapiro. Stochastic programming by monte carlo simulation methods. *Stochastic Programming E-Print Series*, 2000.

A.V. Skorokhod. *Studie in the theory of random processes*. Dover Publications, 1965.

E. M. Stein and J. C. Stein. Stock price distributions with stochastic volatility: An analytical approach. *The Review of Financial Studies*, 4:727–752, 1991.

O. Vasicek. An equilibrium charaterization of the term structure. *Journal of Financial Economics*, 5:177–188, 1977.

T. Yamada and S. Watanabe. On the uniquenes of solutions of stochastic differential equations. *Journal of Mathematics of Kyoto University*, 11(1):155–167, 1971.