Adaptive Trust-Region POD Methods and their Application in Finance

Dissertation

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

Dem Fachbereich IV der Universität Trier
vorgelegt von

Matthias Schu

Trier, 2012

Eingereicht am 13.8.2012

Gutachter: Prof. Dr. Ekkehard Sachs
Prof. Dr. Stefan Volkwein

Tag der mündlichen Prüfung: 30.11.2012

Contents

German Summary (Zusammenfassung)

Bei der Preisberechnung von Finanzderivaten bieten sogenannte Jump-diffusion-Modelle mit lokaler Volatilität viele Vorteile. Aus mathematischer Sicht jedoch sind sie sehr aufwendig, da die zugehörigen Modellpreise mittels einer partiellen Integro-Differentialgleichung (PIDG) berechnet werden. Wir beschäftigen uns mit der Kalibrierung der Parameter eines solchen Modells. In einem kleinste-Quadrate-Ansatz werden hierzu Marktpreise von europäischen Standardoptionen mit den Modellpreisen verglichen, was zu einem Problem optimaler Steuerung führt.

Ein wesentlicher Teil dieser Arbeit beschäftigt sich mit der Lösung der PIDG aus theoretischer und vor allem aus numerischer Sicht. Die durch ein implizites Zeitdiskretisierungsverfahren entstandenen, dicht besetzten Gleichungssysteme werden mit einem präkonditionierten GMRES-Verfahren gelöst, was zu beinahe linearem Aufwand bezüglich Orts- und Zeitdiskretisierung führt.

Trotz dieser effizienten Lösungsmethode sind Funktionsauswertungen der kleinste-Quadrate-Zielfunktion immer noch teuer, so dass im Hauptteil der Arbeit Modelle reduzierter Ordnung basierend auf Proper Orthogonal Decomposition Anwendung finden. Lokale a priori Fehlerabschätzungen für die reduzierte Differentialgleichung sowie für die reduzierte Zielfunktion, kombiniert mit einem Trust-Region-Ansatz zur Globalisierung liefern einen effizienten Algorithmus, der die Rechenzeit deutlich verkürzt. Das Hauptresultat der Arbeit ist ein Konvergenzbeweis für diesen Algorithmus für eine weite Klasse von Optimierungsproblemen, in die auch das betrachtete Kalibrierungsproblem fällt.

Acknowledgements

First of all, I would like to express my gratitude to Prof. Ekkehard W. Sachs, my adviser, for his constant support and the many suggestions during my research and for giving me the opportunity to write this thesis. I am also thankful to Prof. Stefan Volkwein not only for serving as examiner but also for his former research that provides a basis for several results shown here.

This work was funded by the Deutsche Forschungsgemeinschaft within the priority program SPP 1253 'optimization with partial differential equations' and in parts by the research center FoRUmstat at the University of Trier, whom I also thank.

Moreover, I would like to thank my colleagues at the Department of Mathematics for many fruitful discussions and the great atmosphere they provided. Particular mention deserve the coworkers in the research group of Prof. Sachs Bastian Groß, Dr. Timo Hylla, Dr. Christoph Käbe, Andre Lörx, Marina Schneider, Matthias Wagner and Xuancan Ye as well as my colleagues Ulf Friedrich, Benjamin Rosenbaum, Dr. Claudia Schillings, Dr. Stephan Schmidt, Martin Siebenborn, Roland Stoffel, Dirk Thomas and Christian Wagner.

Further, I am deeply indebted to Christine and my family, especially my parents Gertrud and Hermann, without whom this would not have been possible.

Matthias Schu
Trier, 2012

Chapter 1

Introduction

So-called jump-diffusion models with local volatility provide many advantages concerning the pricing of financial derivatives. However, from a mathematical point of view they are quite complex since the corresponding model prices have to be calculated via a partial integro-differential equation (PIDE). Here, we are dealing with the calibration of the parameters of such a model. We compare market prices of standard European options with the model prices in a least-squares approach yielding an optimal control problem.

A substantial part of this thesis is the solution of the PIDE where the focus is on the numerical part. The dense linear systems of equations arising from an implicit time discretization scheme are solved with a preconditioned GMRES method leading to an almost linear complexity regarding time and space discretization.

Despite this efficient solution, evaluations of the least-squares objective function are still expensive such that in the main part of this thesis reduced order models based on proper orthogonal decomposition (POD) are used. Local a priori error estimates for the reduced differential equations as well as for the corresponding reduced objective function combined with a globalizing trust-region framework yield an efficient algorithm that clearly reduces the computing time. The main result of this thesis is a convergence proof for this algorithm for a wide class of optimization problems to which the considered calibration problems belong to as well.

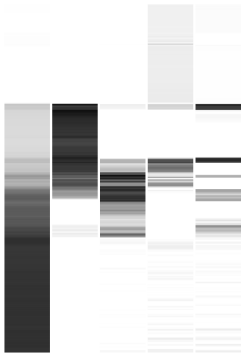
We start by giving a short motivation for reduced order models including an example from image compression, and afterwards describe the outline of the thesis.

1.1 Motivation

We first want to motivate the main topic of this thesis namely the application of reduced order models. In the field of partial differential equations, the use of reduced order models is well-known. The idea is to replace the common finite element basis functions of a space discretization by only a few problem-dependent basis functions in a Galerkin approach. The finite element basis is usually a discretization of the underlying function space, for instance the space H_0^1. However, the solutions of a specific partial differential equation for different control parameters mostly evolve in a lower-dimensional subspace of this function space. Thus, we use more complex, problem-dependent basis functions which only span this subspace and not the whole function space. The concept can be illustrated by the compression of an image similar to an idea in Antoulas et al. (2001).

Figure 1.1(a) shows a black and white image with a resolution of 1536×2048 pixels. Mathematically, we are dealing with a matrix in $[0,1]^{1536\times 2048}$ since the grey values are stored as

(a) Image with resolution 1536 × 2048



(b) First five advanced basis functions

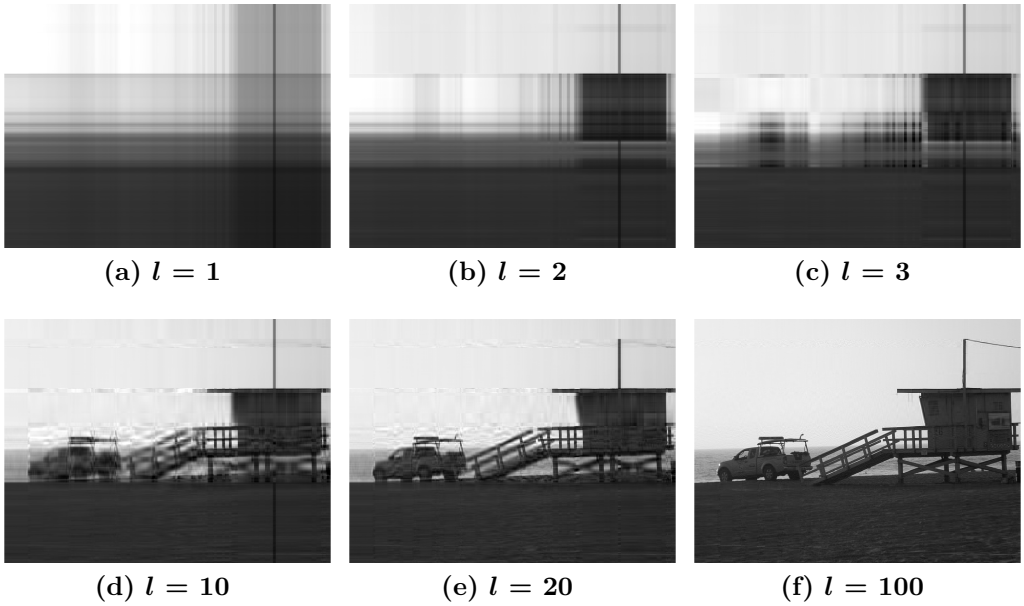
Figure 1.1: Extracting significant information from an image

numbers between 0 (black) and 1 (white), leading to a total of 3.1 million numbers.

We can now interpret each column vector of this matrix as a linear combination of simple unit vectors. The question now is whether we can find some better vectors to represent the columns of the image. The next figure 1.1(b) shows how such basis functions could look like. For instance, the most important information about the image is the light top and the dark bottom what is obviously stored in the first of the five vectors. The information about the hut is given in the second one. Those basis functions, which contain the most significant information, are extracted from the image using a singular value decomposition.

However, they can now be used to reconstruct the image with much less information by representing each column as a linear combination of this new basis. If we use only the first basis function in figure 1.1(b), we have to store this basis function (1536×1 numbers) and the weighting coefficients for each column (2048×1 numbers), what is 0.1% of the original memory requirements. The corresponding compressed image is shown in figure 1.2(a). Increasing the number of basis functions used to recalculate the image leads to the results shown in the next figures (b)-(f). The last one with 100 complex basis functions shows a pretty good approximation to the original image, but it still needs only a fraction of the original memory (11%).

The idea presented above even leads to significantly better approximations when we apply it to a parabolic differential equation, e.g. the heat equation or the option pricing problem that we consider further below. The solution in one time step then corresponds to one column in the motivating example and the unit vectors correspond to the simple finite element basis functions of the spatial discretization. Usually, a very small basis is sufficient to represent all time steps since these solutions are quite smooth, in particular compared to the image above.



(a) $l = 1$ (b) $l = 2$ (c) $l = 3$

(d) $l = 10$ (e) $l = 20$ (f) $l = 100$

Figure 1.2: Reconstruction of the image in figure 1.1(a) using different numbers, l, of basis functions

1.2 Outline

The thesis is mainly divided into four parts. The problem description, the numerical solution of the problem, the introduction of reduced order models with fixed control parameter and the use of these reduced order models in optimization where the control changes permanently. We briefly describe the four parts.

Chapter 2
This chapter is dedicated to option pricing models and their calibration. As a financial derivative, the value of an option depends on the value of some underlying. Beside the formal definition of the options in focus, European calls, the first section 2.1.1 describes different approaches to model this underlying value. Starting with the famous Black-Scholes model, we name its weaknesses and introduce more advanced models. As our model of choice we use a jump-diffusion process with an additional local volatility function and the following dynamics written as a stochastic differential equation:

$$dS_t = \mu S_{t\_}\, dt + \sigma(t, S_{t\_}) S_{t\_}\, dW_t + S_{t\_}\, d\Big(\sum_{j=1}^{N_t} (e^{Y_j} - 1)\Big).$$

The corresponding option prices can be computed by solving a partial integro-differential equation (cf. section 2.1.3). In view of the calibration problem that is discussed below, we focus on the Dupire-like version of the PIDE where the constants maturity T and strike price

K appear as variables:

$$\tilde{D}_T(T,K) - \frac{1}{2}\sigma^2(T,K)K^2\tilde{D}_{KK}(T,K) + r(T)K\tilde{D}_K(T,K)$$
$$-\lambda \int_{-\infty}^{+\infty} \Big(e^y(\tilde{D}(T,Ke^{-y}) - \tilde{D}(T,K)) + K(e^y-1)\tilde{D}_K(T,K)\Big)f(y)\,dy = 0 \qquad (1.1)$$
$$(T,K) \in [t_0, T_{max}) \times (0,\infty)$$
$$\tilde{D}(t_0,K) = \max\{S_0 - K, 0\}, \quad K \in (0,\infty).$$

Taking a closer look at this equation, it contains several parameters and parameter functions, respectively. These are in first place the local volatility function $\sigma(T,K)$, the jump intensity λ and the jump size distribution function $f(y)$.

Section 2.2 then states the corresponding calibration problem in a least-squares formulation as

$$\min_{\tilde{D},\sigma,\lambda,f} J(\tilde{D},\sigma,\lambda,f) := \frac{1}{2}\sum_{i=1}^{M}\Big(\tilde{D}(T_i,K_i) - D_i^M\Big)^2$$
$$\text{s.t. PIDE (1.1)},$$

i.e. we adjust the parameters in such a way that the model prices fit some given market prices D_i^M at M couples (T_i,K_i). We further briefly discuss some mathematical challenges arising in the calibration process.

Chapter 3

We have introduced the calibration problem in chapter 2, and the following chapter deals with its numerical solution. After having proven existence and uniqueness of weak solutions to the partial integro-differential equation in weighted function spaces, the main focus of the chapter is the introduction of a numerical method for its solution. The discretization of the space variable via a finite element approach leads to a dense stiffness matrix what is due to the non-local integral term in the PIDE. For the time discretization, an implicit method would be desirable since the problem is known to be very stiff and explicit methods are restricted by strong CFL conditions. A Crank-Nicolson scheme is used with an additional smoothing of the non-smooth initial condition yielding a second-order convergence in time. However, the linear systems of equations for such implicit methods are dense such that we propose to use a preconditioned GMRES method for their solution. This approach works quite well in numerical tests (section 3.2.4) and, thus, is an alternative to the few existing solvers for implicitly discretized PIDEs.

In section 3.3, we then focus on the numerical solution of the calibration problem that can be written in an abstract vectorial form with state variable y and control u as:

$$\min_{y\in W, u\in\mathcal{U}} J(y,u) := \frac{1}{2}\sum_{i=1}^{D}||Cy(\hat{t}_i) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2}||u||^2 \qquad (1.2)$$

$$s.t. \quad \dot{y}(t) + A(u;t)y(t) - l(u;t) = 0 \ , \quad t \in (0,T]$$
$$y(0) = y_0.$$

We discuss the implications of both the 'first optimize' and 'first discretize' approach with focus on the time discretization. As the least-squares objective function involves pointwise observations of the PIDE solution, the corresponding adjoint equations contain delta Dirac functions (or at least approximations) leading to oscillations in the numerical solution if we do not apply a smoothed time discretization scheme.

Concerning the optimization of the objective function, a brief comparison of the quasi-Newton and Gauss-Newton approach to calculate first- and second-order information is given in section 3.3.2.

The chapter ends in section 3.3.3 with further numerical results on the adjoint solution, and we address a typical calibration problem with real-world market data.

Chapter 4
Chapter 4 provides an introduction to reduced order models based on proper orthogonal decomposition. In section 4.1, POD is defined as a way of extracting the most significant information from a set of given functions or vectors (called 'snapshots' in the following). To be precise, we want to find those basis functions which represent the set of snapshots better than any other basis. Mathematically, this can be written as a constrained optimization problem, and we also recall the methods of solving it, which are well-known in literature. It turns out that the POD basis functions are given by eigenfunctions of a specific eigenvalue problem. And if we only use a part of this POD basis, the average approximation error for the snapshots can be expressed by a sum over those eigenvalues λ_j whose eigenfunctions are not used.

Section 4.2 now shows how POD can be used in the context of parabolic differential equations. Here, the snapshots are given by the solution of this differential equation at different time instances. We present a priori error estimates for time-dependent elliptic operators what is an extension of the results in Kunisch and Volkwein (2001). Since our main aim is the use of POD in our optimal control problem (1.2), the error between a discretized objective function based on a finite element space, $f(u) = J(y^{FE}(u), u)$, and an objective function based on a POD approximation with rank l, $f_l(u) = J(y^l(u), u)$, has to be estimated. Section 4.2.2 establishes a relation between this error and the sum over the remaining eigenvalues using the previous results. For fixed but arbitrary u and a $k_1 > 0$, we obtain

$$|f(u) - f_l(u)|^2 \leq k_1 \sum_{j=l+1}^{r} \lambda_j$$

if the POD basis contains the snapshots from the state solution at u. A similar result is then obtained for the gradient of f. However, we need to include snapshots from the adjoint

solution as well to get for a $k_2 > 0$:

$$||\nabla f(u) - \nabla f_l(u)||^2 \leq k_2 \sum_{j=l+1}^{r} \lambda_j.$$

The k_2 has several dependencies, mainly a proper weighting of adjoint and state snapshots has to be guaranteed. Numerical results supporting the previous theoretical statements are shown in the last section 4.3.

Note that all results presented in this chapter are local in the sense that they only hold true if the control u is fixed.

Chapter 5

A globalization of the POD model is then achieved in chapter 5 by embedding the reduced order model into a trust-region framework. We begin by recalling the idea of a basic trust-region approach with a quadratic model function. Several generalizations of this concept based on the work of Fahl (2000) complete the first section ending with a global convergence proof for non-quadratic model functions under the main assumption of a sufficient gradient approximation of the model function at the trust region centerpoint.

The main result of this thesis is given in section 5.2 where the error estimates of section 4.2.2 are combined with the generalized trust-region approach of section 5.1.2, leading to a convergence proof for the trust-region POD algorithm. Here it is vital that the POD model function can fulfill the assumptions on the gradient accuracy in the sense of Carter (1991):

$$||\nabla f(u_k) - \nabla m_k^l(u_k)|| \leq \Big(k_2 \sum_{j=l+1}^{r} \lambda_j\Big)^{\frac{1}{2}} \leq \zeta ||\nabla m_k^l(u_k)||.$$

The size of the POD basis can be managed adaptively by comparing reduced and exact gradient until the condition above is satisfied. Numerical results in which the adaptive trust-region POD algorithm is used to solve the calibration problem discussed earlier show the efficiency of the algorithm.

All numerical results in this thesis have been produced by a MATLAB code on a desktop PC with Intel®Core™2 Duo (3.00 GHz) processor and 4GB RAM.

Chapter 2

Calibration Problems in Option Pricing

This chapter will provide basic information about options, option pricing models and the calibration of parameters arising therein.

After having dealt with the concept of options as a special kind of financial derivative (section 2.1.1), we focus on how we can calculate their value, i.e. the fair price that one has to pay for them. The first step here is the stochastic modeling of the uncertain underlying value by a stochastic differential equation (section 2.1.2). By now, there exist a lot of models which are mainly extensions of the famous Black-Scholes model (cf. Black and Scholes (1973), Merton (1973)). Our focus lays on a special kind of extension called 'jump-diffusion models'. The stochastic differential equations can further be transformed to partial differential equations or – depending on the model – even to partial integro-differential equations (section 2.1.3).

The above mentioned models include several parameters which strongly influence the option prices. Hence, the calibration of these parameters is an important issue and therefore subject of section 2.2.

2.1 Option Pricing

An 'option' is a financial derivative, i.e. its value is derived from the value of some underlying. The usage of such derivatives has increased significantly in practice, where the two main purposes are hedging, i.e. the mitigation of risks in the underlying, and speculation, i.e. they are used to make leveraged profits by betting on a certain behavior of the underlying.

2.1.1 Introduction

As it is implied by the name, the owner of such a financial product has the right to do something but not the obligation. We start by giving a formal definition of a special kind of option that is considered in this thesis.

Definition 2.1.1. ***(European call option)***
A European call option is a contract that gives its owner the right but not the obligation to buy an underlying at a prescribed time in the future T (maturity) for a certain price K (strike price).

According to this definition, it is clear what happens at maturity. Let S_T be the value of the underlying, e.g. a stock, at the expiration date T. If $S_T > K$, then the option holder exercises the option and purchases the underlying for K. Since the market price is S_T, he

immediately can sell it and obtains the payoff $S_T - K$. On the other hand, if $S_T < K$, then he would not exercise, because he could buy the underlying for only S_T at the market. Then the payoff is zero. This leads to the so-called hockey stick function $C(S_T) = \max\{S_T - K, 0\}$ illustrated in figure 2.1.

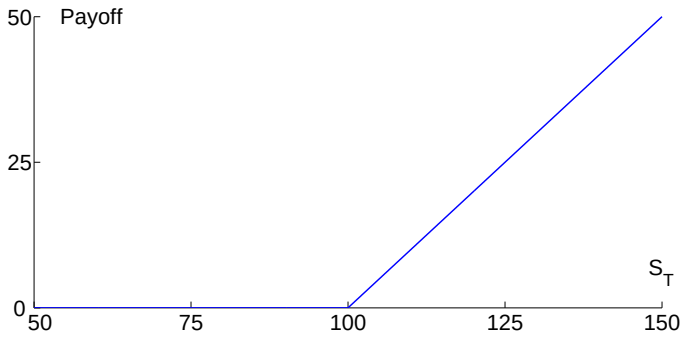


Figure 2.1: Payoff of a European call option with strike $K = 100$ at maturity depending on the underlying price S_T

There are several ways to classify options. The two main classifications are implied by definition 2.1.1. The first one is to distinguish between the above mentioned 'call' and so-called 'put options'. As already specified, the owner of a call option has the right to buy the underlying. In contrast, the owner of a put option has the right to sell. The payoff function in this case would be $P(S_T) = \max\{K - S_T, 0\}$.

Another way to categorize options is the time at which they can be exercised. European style options can only be exercised at the maturity date, 'Bermudian' can be exercised at several specified dates before maturity and 'American' options at every date before expiration.

For the sake of completeness, we have to mention the class of 'exotic' options. Here, the payoff functions are usually more complex, for example depending on an average underlying price (Asian option) or including bounds (barrier, knock-out options).

Due to its relevance especially in the world of finance, there is a vast literature on options. References for an introduction to options are, e.g., Wilmott et al. (1993), Chriss (1997) and Hull (2008).

Taking a closer look at figure 2.1, we see that the owner of a European call option has a nonnegative payoff at maturity. Hence, it is clear that one has to pay a certain price for the derivative at the purchase date. Otherwise, there would be the chance for arbitrage, i.e. the possibility of a riskless profit. So, the important question arises, what is the fair price for an option?

To answer this question, option pricing models make assumptions on the behavior of the uncertain underlying price. For example, Cox et al. (1979) have proposed a discrete-time model where the stock price follows a multiplicative binomial process over discrete time instances. When the time discretization is refined, this model converges to the Black-Scholes model that has been introduced by Black and Scholes (1973) and Merton (1973). Here, the uncertainty of the underlying process is modeled by a geometric Brownian motion, what can be written in terms of a stochastic differential equation. This model, its strengths, its

weaknesses and especially some improvements known as Lévy models are covered by the next section.

2.1.2 Option Pricing Models

This section deals with the modeling of the uncertain underlying process. Although we use some notations borrowed from stochastics, we do not introduce and explain every detail here since this would go beyond the scope of this thesis. On the other hand, it is necessary to mention some stochastical concepts to understand the features of different option pricing models.

For example, Chung and Williams (1990), Karatzas and Shreve (2000) Øksendal (2003), provide fundamentals in stochastic processes that are used here in parts. For an introduction to the stochastic ideas used in the context of option pricing, we refer to Lamberton and Lapeyre (1996), Elliott and Kopp (2005) and especially Schoutens (2003) and Cont and Tankov (2004a).

The most famous option pricing model has been introduced by Black and Scholes (1973) and Merton (1973). Here, the uncertain stock price is modeled by a geometric Brownian motion.

Definition 2.1.2. ***(Standard Brownian motion)***
A stochastic process $W = \{W_t, t \geq 0\}$ *adapted to a filtration* $\mathcal{F} = \{\mathcal{F}_t, t \geq 0\}$ *is called standard Brownian motion on a probability space* $(\Omega, \mathcal{F}, P)$ *if*

a) $W_0 = 0$ *almost sure,*

b) W *has independent increments, i.e.* $W_t - W_s$ *and* $\mathcal{F}_s$ *are independent for* $0 \leq s < t$,

c) W *has stationary and normally distributed increments, i.e.* $W_t - W_s \sim N(0, t-s)$ *for* $0 \leq s < t$, [1]

d) the mapping $t \to W_t$ *is continuous almost sure.*

The second item in its definition implies the so-called Markov property of the Brownian motion. Simplified, the behavior of the stochastic process after time t is only affected by the present value W_t and not by the past history. In fact, an attribute linked to stock prices as well. The last point will also be of interest later on. It says that almost all realizations of the stochastic process – which are usually called 'paths' – are continuous. So, jumps or discontinuities do not occur.

Black and Scholes (1973) and Merton (1973) do not use the Brownian motion to model the uncertainty of the stock price itself, but they propose to model the uncertainty of the return of a stock. Given a stock price S_t, the change of the price $\Delta S_t = S_{t+\Delta t} - S_t$ in a small interval Δt is given by an expected increase $S_t \mu \Delta t$ – here μ is the expected rate of return – added by a random part $S_t \sigma \Delta W_t$ where σ, the volatility, describes how much the stock price fluctuates. In total, we get

$$\Delta S_t = S_t(\mu \Delta t + \sigma \Delta W_t), \quad S_0 > 0. \tag{2.1}$$

[1] $N(\mu, \sigma)$ is the normal distribution with mean μ and variance σ

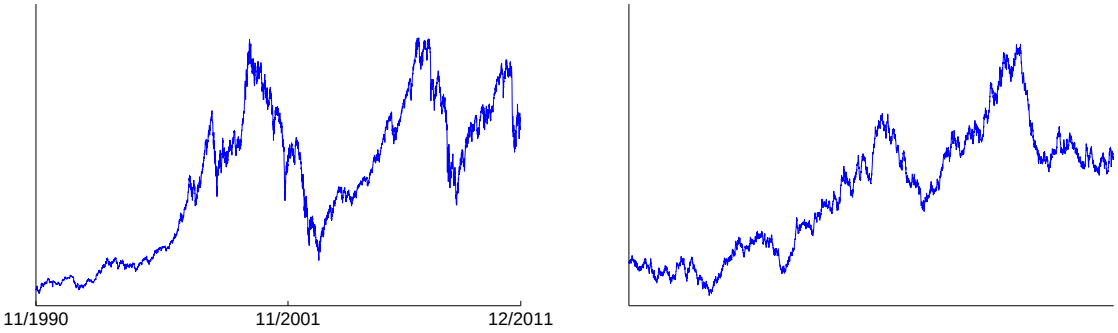
In the limit, as $\Delta t \to 0$, this leads to the following stochastic differential equation (SDE)

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad S_0 > 0. \tag{2.2}$$

The unique solution of this SDE is called 'geometric Brownian motion' (cf. Øksendal (2003)):

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}.$$

Since we know from definition 2.1.2 that the increments of W_t are normally distributed, it is clear that the increments of S_t are log-normally distributed.



(a) DAX history **(b) geometric Brownian motion**

Figure 2.2: Comparison between DAX history (1990–2011) and a sample path of a geometric Brownian motion

Figure 2.2 provides a motivation for the use of a geometric Brownian motion. It shows on the left-hand side a chart of the German stock index DAX from 1990 to 2011 (source: `www.finance.yahoo.com`) and on the right-hand side a sample path of a geometric Brownian motion. For the untrained eye, it would be hard to say which is the original DAX. Thus, the model seems to be well suited at first sight.

To make some statements regarding the option price, when the asset price is driven, e.g., by a geometric Brownian motion, one needs to have an additional riskless asset, called bond B_t, which exhibits the following behavior:

$$dB_t = rB_t dt. \tag{2.3}$$

Given a B_0, the solution to this ordinary differential equation is $B_t = B_0 e^{rt}$, i.e. the money is continuously compounded with a constant interest rate r.

Further, we make some simplifying assumptions on the market:

Assumption 2.1. ***(Market assumptions)***

a) No market friction,

b) no default risk,

c) short selling is permitted,

d) market participants act rational,

e) no arbitrage,

f) no dividends.

This means that we assume an idealized market, where we, e.g., can deposit and borrow money without transaction costs and taxes. Further details can be found in Schoutens (2003) for instance.

Given this market, the price of a European call option can be expressed in terms of an expected value with respect to a certain measure Q, called equivalent martingale measure.

Definition 2.1.3. ***(Pricing formula)***
Let S_t be modeled by a geometric Brownian motion as in (2.2). Then the current price of a European call option $C(t, S_t)$ with payoff function $h(S_T) := \max\{S_T - K, 0\}$ is given by

$$C(0, S_0) = e^{-rT}\mathbb{E}_Q[h(S_T)]. \tag{2.4}$$

In other words, the price of a call option today is the discounted expected value of the payoff at maturity under a certain risk-neutral measure Q. Further details can be found in Föllmer and Schied (2004, pp. 223ff).

In case of the Black-Scholes model, there even exists a closed-form solution for the call price.

Remark 2.1.4. ***(Black-Scholes closed-form solution)***
For given maturity T, strike price K, interest rate r and volatility σ the price of a European call option with asset price S at time t in a Black-Scholes model is given by

$$C^{BS}(t, S) = S\mathcal{N}(d_1) - Ke^{-r(T-t)}\mathcal{N}(d_2), \tag{2.5}$$

where $\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi^2}{2}} d\xi$ is the cumulative probability distribution for the standardized normal distribution, and $d_1 = \frac{1}{\sigma\sqrt{T-t}}\Big(\ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)\Big)$ and $d_2 = d_1 - \sigma\sqrt{T-t}$.

Proof. See, e.g., Black and Scholes (1973) or Hull (2008). □

This closed-form solution is one of the reasons for the success of the Black-Scholes model in practice. Note that μ, the expected rate of return of our stock (see (2.2)), does not occur in the solution formula above because under the risk-neutral measure Q, it is replaced by the riskless interest rate r implying risk neutrality.

The formula for the current call price $C^{BS}(0, S_0)$ rather uses the constants strike K and maturity T, which are defined when the contract is concluded. The interest rate r as well as

the spot price S_0 are also known a priori from market data. The only parameter that cannot be observed directly is the volatility σ. So its influence on the call price is very interesting. It is easy to see that there is a strictly monotone increasing correlation between σ and the call price. Thus, the term 'implied volatility' introduced in the following remark makes sense.

Remark 2.1.5. ***(Implied volatility)***
Let maturity T, strike price K, interest rate r and current asset price S_0 be given and let $C^* \in](S_0 - Ke^{-rT})^+, S_0[$ be the price of a European call. Then there exists a unique volatility σ^* with

$$C^* = C^{BS}(0, S_0;\ \sigma^*), \tag{2.6}$$

where $C^{BS}(t, S; \sigma)$ is the corresponding Black-Scholes price with constant volatility σ.
σ^* is called 'implied volatility'.

Proof. See, e.g., Cont and Tankov (2004a). Just note that if $C^* \notin](S_0 - Ke^{-rT})^+, S_0[$, then there is a chance for arbitrage. □

Note that there exists no closed-form solution, so the implied volatility has to be calculated via numerical methods. However, this implied volatility can now be used to illustrate one of the main weaknesses of the Black-Scholes model: the constancy of the parameters. Since the shortcomings of the model are well-known, there is a vast literature regarding this topic. Derman and Kani (1994), Andersen and Andreasen (2000), Schoutens (2003) and Cont and Tankov (2004a) could be mentioned here.

Empirical studies show that given one asset and several options an this asset with differing strike prices K and maturities T, the corresponding implied volatilities $\sigma^*(T, K)$ are not flat as suggested by the Black-Scholes model. Hence, the appearance of this so-called 'implied volatility surface' $\sigma^*(T, K)$ is of interest The dependence on the strike price is known as volatility smile or skew. Typically, 'at-the-money' calls ($K \approx S_0$) have a lower implied volatility as calls that are 'in-the-money' ($K < S_0$) or 'out-of-money' ($K > S_0$). In this case, the curve $\sigma^*(K)$ looks like a smile. If the dependence is decreasing, it is a skew. There is also a strong dependence of σ on the maturity of the call. It has been observed that the smile or skew effect flattens out as maturity increases. An example of such a surface is displayed in figure 3.8 in the next chapter.

These observations give rise to new, more advanced models, which can be grouped into three main ideas (cf. Andersen and Andreasen (2000)).

Dupire (1994) and Derman and Kani (1994) propose the so-called 'local volatility models'. Here, the constant volatility of the Black-Scholes model is replaced by a deterministic function of stock price S and time t:

$$dS_t = \mu S_t dt + \sigma(t, S_t) S_t dW_t.$$

Although there is no closed-form solution available as in the Black-Scholes case, the model is easy to handle from a numerical point of view. Since $\sigma(t, S)$ is a function, we can fit the model precisely to many quoted call prices. Beside these advantages, there are also some drawbacks (cf. Andersen and Andreasen (2000) and the references cited therein). Especially

the fitting to typical skews for short-term calls requires an unrealistic heavily twisting of the local volatility surface.

The second one to mention is the 'stochastic volatility model'. There are different specific approaches by, e.g., Hull and White (1987), Stein and Stein (1991) and the most famous by Heston (1993). The latter models the dynamics of the stock price by a Brownian motion as well as the Black-Scholes model, but in addition, the volatility as the driving force of the call price is modeled by a stochastic process, namely an Ornstein-Uhlenbeck process, too. Written as a stochastic differential equation, we have

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^1 \\ dv_t &= \kappa(\theta - v_t)dt + \alpha\sqrt{v_t} dW_t^2, \end{aligned}$$

where the increments of the two Brownian motions W^1 and W^2 are correlated with coefficient ρ (cf. Heston (1993)). There are parameter combinations where the implied volatilities of the corresponding Heston prices show the typical skew or smile that is observed in market data. But often the correlation ρ between stock price and volatility has to be chosen unrealistically high to get the desired result. On the other hand, an important advantage of the Heston model is the existence of a closed-form solution.

The third approach requires the introduction of a new stochastic process since the uncertainty is no longer modeled solely by a continuous Brownian motion. Merton (1976) suggests to add random jumps as an additional source of uncertainty that mainly model rare large market movements. The more general process compared to the Brownian motion is called Lévy process:

Definition 2.1.6. ***(Lévy process)***
A stochastic cadlag [2] process $X = \{X_t, t \geq 0\}$ adapted to a filtration $\mathcal{F} = \{\mathcal{F}_t, t \geq 0\}$ is called Lévy process on a probability space $(\Omega, \mathcal{F}, P)$ if

a) $X_0 = 0$ almost sure,

b) X has independent increments, i.e. $X_t - X_s$ and $\mathcal{F}_s$ are independent for $0 \leq s < t$,

c) X has stationary increments, i.e. the distribution of $X_{t+h} - X_t$ is independent of t,

d) X is stochastic continuous, i.e. $\forall\ \epsilon > 0$, $\lim_{h \to 0} P(|X_{t+h} - X_t| \geq \epsilon) = 0$ for $t \geq 0$.

There are mainly two differences compared to the Brownian motion. First, the distribution of the increments does not need to be the normal distribution. And second, the continuity of the paths of X is generalized to stochastic continuity, i.e. jumps in a path can occur, but only at random times t which are not predefined. It can be easily shown that the following examples fit in the Lévy process framework.

[2] cadlag: right-continuous and with left limits

Example 2.1.7. ***(Lévy processes)***

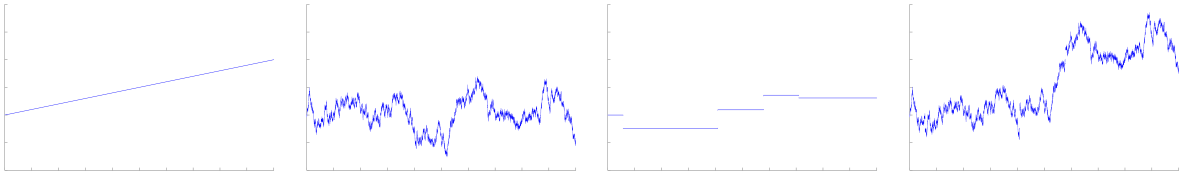
a) ***Brownian motion****: $X_t = W_t$, so Lévy process is a generalization of this concept.*

b) ***Compounded Poisson process****: $X_t = \sum_{i=1}^{N_t} Y_i$. Here, N_t is a Poisson process with intensity $\lambda > 0$ – i.e. it is a counting process, where N_t is the number of random events that occurred up to time t with expected value $\mathbb{E}(N_t) = \lambda t$ – and the Y_i's are independent and identically distributed. This means, N_t describes the occurrence of jumps and the Y_i's determine the jump size (figure 2.3(c) provides an illustration).*

The option pricing models based on general exponential Lévy processes can be divided into two categories. One is called 'jump-diffusion models'. Here, as proposed by Merton (1976) jumps are added to the Brownian motion to model large movements of the asset. The second type are so-called 'infinite activity models', where the Brownian motion is omitted and more or less replaced by an infinite number of small jumps. Barndorff-Nielsen (1997), Carr et al. (2002) can be named as references for models of the last-mentioned type. In this thesis we will focus on jump-diffusion models driven by the following dynamics:

$$dS_t = \mu S_{t\_}\, dt + \sigma S_{t\_}\, dW_t + S_{t\_}\, d\Big(\sum_{j=1}^{N_t} (e^{Y_j} - 1)\Big) \tag{2.7}$$

with $\sigma > 0$. The first two parts on the right-hand side are equal to the Black-Scholes model.



(a) Drift **(b) Brownian motion** **(c) Compounded Poisson process** **(d) = (a) + (b) + (c)**

Figure 2.3: $X_t = \mu t + \sigma W_t + \sum_{j=1}^{N_t} Y_j$: Composition of a typical path of a jump-diffusion process used to model the log-price

A drift and a Brownian motion. What is new, though, is the third term, where jumps are added to the process by a compounded Poisson process (independent of the Brownian motion) that has already been introduced in example 2.1.7. As an illustration, figure 2.3 shows a typical path of a jump-diffusion process and how it is composed of the three parts described above. Due to the discontinuities, the notation $S_{t\_} = \lim_{s \nearrow t} S_s$ is required. Let τ^j be a point in time, where a jump occurs. Then $S_{\tau^j_-}$ is the value right before the jump and the following holds

$$S_{\tau^j} = S_{\tau^j_-}\, e^{Y_j},$$

this means e^{Y_j} is the ratio of the asset price before and after the jump, or in other words Y_j is the jump in the rate of return of the stock price (Glasserman (2004)).

Jump-diffusion models differ in the distribution of the jump sizes Y_j. There are two popular examples.

Example 2.1.8. ***(Jump-diffusion models)***

a) ***Merton (1976)****: Y_i are normally distributed with well-known density function $f^M(y) = \frac{1}{\sqrt{2\pi}\sigma_J} exp\{-\frac{(y-\mu_J)^2}{2\sigma_J^2}\}$ (with $\mu_J \in \mathbb{R}$, $\sigma_J > 0$).*

b) ***Kou (2002)****: Y_i has an asymmetric double exponential distribution with density $f^K(y) = p\lambda^+ e^{-\lambda^+ y}\mathbb{1}_{\{y\geq 0\}} + (1-p)\lambda^- e^{\lambda^- y}\mathbb{1}_{\{y<0\}}$ (with $p \in [0,1]$, $\lambda^+ > 1$, $\lambda^- > 0$).*

The main advantage of these new models for the stock price dynamics is that a skew in the implied volatility, especially for short-term options, can be produced quite easily by setting the mean jump size to be negative. This has been the main weakness of the approaches mentioned up to now. On the other hand, the models by Merton and Kou as well as the stochastic volatility models mentioned above only involve few parameters that are to be calibrated to market prices. Thus, given many market prices, the calibration problem is underdetermined and errors between market and model prices can be too large. Further, jump-diffusion models are known to be difficult to handle from a numerical point of view, although at least for the Merton model (see Example 2.1.8), there exists a semi-analytical solution in terms of a series of Black-Scholes solutions.

Remark 2.1.9. ***(Merton model)***
Given the Merton jump-diffusion model (see (2.7) and example 2.1.8) with volatility $\sigma > 0$, jump intensity λ, mean jump size μ_J and volatility of the jump size σ_J. Then, for given maturity T, strike price K, interest rate r and current stock price S_0, today's price of a European call option is given by

$$C^M = \sum_{n=0}^{\infty} \frac{e^{-\bar{\lambda}T}(\bar{\lambda}T)^n}{n!} C^{BS}(0, S_0;\ \bar{r}, \bar{\sigma}),$$

where $\bar{\lambda} = \lambda(1+\mu_J)$, $\bar{r} = r - \lambda\mu_J + \frac{n\ln(1+\mu_J)}{T}$, $\bar{\sigma}^2 = \sigma^2 + \frac{n\sigma_J^2}{T}$ and $C^{BS}(0, S_0;\ \bar{r}, \bar{\sigma})$ is the today's Black-Scholes price with interest rate $\bar{r}$ and volatility $\bar{\sigma}$ (see remark 2.1.4).

Proof. See, e.g., Merton (1976). □

Since all three generalization of the Black-Scholes model described above still have weaknesses, there is a vast literature on combinations of the different approaches. Bates (1996) combines stochastic volatility with jump-diffusion. Said (1999) and Lipton (2002) propose a stochastic local volatility model, where Lipton (2002) also includes jumps, i.e. a combination of all three approaches. A jump-diffusion model with local volatility was suggested by Andersen and Andreasen (2000). In the further course of this thesis this model will be used since it is suitable to produce typical volatility skews with its jump part and fit prices precisely with the local volatility function. Hence, our stock price is modeled by the following stochastic differential equation:

$$dS_t = \mu S_{t\_}\, dt + \sigma(t, S_{t\_}) S_{t\_}\, dW_t + S_{t\_}\, d\Big(\sum_{j=1}^{N_t}(e^{Y_j} - 1)\Big). \tag{2.8}$$

Due to the fact that especially in case of local volatility functions, there is no analytical solution available, we have to worry about a numerical solution of the problem. There are two main approaches. Monte Carlo simulation of the stochastic differential equation or the transformation of the SDE into a partial differential equation that can be solved numerically. This will be discussed in the next section.

2.1.3 Option Pricing with Partial Differential Equations

There are several methods to compute option prices based on the different models introduced in the last section. As already mentioned there are closed-form (remark 2.1.4) or at least semi-analytical solutions (remark 2.1.9) available for a few of them. If the characteristic function of the stochastic process is known analytically one may use fast Fourier Transformation as described in, e.g., Carr and Madan (1999). These approaches are known to be fast, but they are only feasible for certain models. On the other hand, Monte Carlo methods, which discretize the stochastic differential equation of the underlying process, as described, e.g., in Glasserman (2004), can be used for most of the models and many types of options, but they are known to be slow since the rate of convergence is poor.

Thus, we concentrate on another approach. The pricing formula for a European call option in definition 2.1.3 with an underlying price modeled by a stochastic differential equation can be transformed to a partial differential equation. Without entering into details, this is done using stochastic arguments. For the jump-diffusion model in (2.8), there is the following result for European call options.

Theorem 2.1.10. ***(Backward PIDE)***
Given a jump-diffusion model as in (2.8), the price of a European call option, $C(t,S)$, with asset price S, maturity T, strike K at time t can be calculated via the partial integro-differential equation (PIDE)

$$\begin{aligned} &C_t(t,S) + \frac{1}{2}\sigma^2(t,S)S^2C_{SS}(t,S) + r(t)SC_S(t,S) - r(t)C(t,S) \\ &+\lambda \int_{-\infty}^{+\infty} \Big(C(t,Se^y) - C(t,S) - S(e^y-1)C_S(t,S)\Big) f(y)\, dy = 0 \qquad (2.9) \\ &(t,S) \in [0,T) \times (0,\infty) \\ &C(T,S) = \max\{S-K,0\}, \quad S \in (0,\infty), \end{aligned}$$

where λ is the jump intensity and $f(y)$ is the density function of the jump size distribution.

Proof. This result can be found in Andersen and Andreasen (2000) and Achdou and Pironneau (2005). □

So, to calculate the call prices, a partial differential equation with an additional integral term has to be solved. Note that the price we are usually interested in is $C(t_0,S_0)$ with $t_0 = 0$, the price today with current stock price S_0. If the jump intensity λ is set to zero – i.e. there are no jumps in the underlying price –, equation (2.9) reduces to the well-known

parabolic Black-Scholes PDE:

$$C_t(t,S) + \frac{1}{2}\sigma^2(t,S)S^2C_{SS}(t,S) + r(t)SC_S(t,S) - r(t)C(t,S) = 0$$
$$(t,S) \in [0,T) \times (0,\infty)$$
$$C(T,S) = \max\{S-K,0\}, \quad S \in (0,\infty).$$

Remark 2.1.11. ***(Boundary condition)***
Since the spatial domain of the PIDE is restricted to $(0,\infty)$*, especially from a numerical point of view, a boundary condition for* $C(t,0)$*,* $t \in [0,T]$ *would be preferable. It is easy to show by, e.g., economical arguments that*

$$C(t,0) = 0, \quad t \in [0,T]$$

is the right choice. However, from a analytical point of view this condition is redundant since it is implied by the differential equation and the final condition. See Cont and Tankov (2004a, p.387) for a more detailed discussion on this.

Note that the assumption of a constant interest rate r in the Black-Scholes model (cf. (2.3)) has been generalized to a time-dependent interest rate $r(t)$, what is closer to market behavior. This interest rate curve is usually not related to the specific underlying and is given as a market value. Further, the strike price K and the maturity T are defined by the call option contract. Hence, the parameters that are not known in (2.9) are the volatility function $\sigma(t,S)$, the jump intensity λ and the density function of the jump size distribution $f(y)$. The proper calibration of these parameters is essential if we want to use this option pricing model for the calculation of new option prices. To calibrate the option pricing model for a certain underlying, we compare model prices with given market prices and adjust the parameters such that the error is minimized (details in section 2.2). Market prices are usually given for call options with different maturities T_i and different strike prices K_i $(i = 1,\dots,M)$. Thus, the PIDE above has to be solved newly for each pair (T_i,K_i) to get today's call price $C(0,S_0;\ T_i,K_i)$. In a least-squares formulation, a function evaluation itself would require a tremendous amount in computing time to solve a whole family of the type (2.9).

A similar problem already occurs when we consider the extension of the volatility in the original Black-Scholes equation from a constant to a function in Dupire (1994). Here, the problem is solved by formulating a PDE where T and K occur as variables and the current time t and stock price S show up in the initial condition of the PDE. Therefore, only one PDE has to be solved for a function evaluation in a least-square formulation.

For the PIDE case, there is a similar variant of Dupire's equation by Andersen and Andreasen (2000).

Theorem 2.1.12. ***(Forward PIDE)***
Given the solution $\tilde{D}(T,K)$ *of the PIDE*

$$\tilde{D}_T(T,K) - \frac{1}{2}\sigma^2(T,K)K^2\tilde{D}_{KK}(T,K) + r(T)K\tilde{D}_K(T,K)$$

$$-\lambda \int_{-\infty}^{+\infty} \Big(e^y(\tilde{D}(T, Ke^{-y}) - \tilde{D}(T,K)) + K(e^y - 1)\tilde{D}_K(T,K) \Big) f(y)\ dy = 0 \tag{2.10}$$

$$(T,K) \in [t_0, T_{max}) \times (0, \infty)$$

$$\tilde{D}(t_0, K) = \max\{S_0 - K, 0\}, \quad K \in (0, \infty)$$

the following holds true:

$$\tilde{D}(T,K) = C(t_0, S_0), \tag{2.11}$$

where $C(t,S)$ is the solution of (2.9) with maturity T and strike price K.

Proof. See Andersen and Andreasen (2000) or Pironneau (2007). □

It is noticeable that, in contrast to the former PIDE, this new equation is of forward type.

Usually we are only interested in the call price today, so in the following – without loss of generality – we assume $t_0 = 0$. Further can the integral part in the PIDE above be simplified in the following way.

Remark 2.1.13. *Setting $\zeta = \zeta(f) := \int_{-\infty}^{+\infty}(e^y - 1)f(y)\ dy$, (2.10) is equivalent to*

$$\tilde{D}_T - \frac{1}{2}\sigma^2(T,K)K^2\tilde{D}_{KK} + (r(T) - \lambda\zeta)K\tilde{D}_K + \lambda(1+\zeta)\tilde{D} \tag{2.12}$$
$$-\lambda \int_{-\infty}^{+\infty} \tilde{D}(T, Ke^{-y})e^y f(y)\ dy = 0,$$
$$(T,K) \in [0, T_{max}) \times (0, \infty)$$
$$\tilde{D}(0,K) = \max\{S_0 - K, 0\}, \quad K \in (0, \infty).$$

Proof. Crucial are the following identities

$$\int_{-\infty}^{+\infty} (e^y - 1)K\tilde{D}_K(T,K)f(y)\ dy = \zeta K\tilde{D}_K(T,K)$$
$$\int_{-\infty}^{+\infty} e^y\tilde{D}(T,K)f(y)\ dy = \zeta\tilde{D}(T,K) + \tilde{D}(T,K) = (1+\zeta)\tilde{D}(T,K)$$

and that $\int_{-\infty}^{+\infty} f(y)\ dy = 1$ since f is a density function of a probability measure. □

Now, a variable transformation is used to eliminate the K's in the coefficients of the equation above to have a nicer PIDE from a numerical point of view.

Remark 2.1.14. *Let $D(T,x)$ be the solution of*

$$D_T - \frac{1}{2}\bar{\sigma}^2(T,x)D_{xx} + \Big(r(T) + \frac{1}{2}\bar{\sigma}^2(T,x) - \lambda\zeta\Big)D_x + \lambda(1+\zeta)D \tag{2.13}$$
$$-\lambda \int_{-\infty}^{+\infty} D(T,x-y)e^y f(y)\ dy = 0,$$
$$(T,x) \in [0,T_{max}) \times (-\infty,\infty)$$
$$D(0,x) = \max\{1-e^x, 0\} =: D_0(x), \quad x \in (-\infty,\infty).$$

Then, with $\sigma(T,K) = \bar{\sigma}(T, \ln(\frac{K}{S_0}))$,

$$\tilde{D}(T,K) = S_0 D(T, \ln(\frac{K}{S_0}))$$

solves (2.12).

Proof. Here, the variable transformation $x = \ln(\frac{K}{S_0})$ leads to

$$\tilde{D}_T = S_0 D_T\ , \quad \tilde{D}_K = S_0 K^{-1} D_x\ , \quad \tilde{D}_{KK} = S_0 K^{-2} D_{xx} - S_0 K^{-2} D_x$$

and $K = S_0 e^x$. Together with a scaling of the PIDE and the initial condition by S_0^{-1} we get the desired result. □

In the following, for the sake of simplicity the bar on the log-transformed volatility function will be omitted if the domain of definition is clear.

Note that in literature, the quotient $\frac{K}{S_0}$ is called the 'moneyness' of the option and consequential $\ln(\frac{K}{S_0})$ is called 'log-moneyness'.

The last-mentioned PIDE will be the one that we solve numerically in chapter 3. However, we mention another transformation of (2.13) that includes a reduction of the convection term. Since the convection term includes the space dependent volatility function, it can not be eliminated totally. But we will see in section 4.3 that a reduction can lead to significant changes in the numerical results.

Remark 2.1.15. *Let $\hat{D}(T,\hat{x})$ be the solution of*

$$\hat{D}_T - \frac{1}{2}\hat{\sigma}^2(T,\hat{x})\hat{D}_{\hat{x}\hat{x}} + \Big(r(T) + \frac{1}{2}\hat{\sigma}^2(T,\hat{x}) - \lambda\zeta - c\Big)\hat{D}_{\hat{x}} + \lambda(1+\zeta)\hat{D} \tag{2.14}$$
$$-\lambda \int_{-\infty}^{+\infty} \hat{D}(T,\hat{x}-y)e^y f(y)\ dy = 0,$$
$$(T,\hat{x}) \in [0,T_{max}) \times (-\infty,\infty)$$
$$\hat{D}(0,\hat{x}) = \max\{1-e^{\hat{x}}, 0\}, \quad \hat{x} \in (-\infty,\infty)$$

for $c \in \mathbb{R}$. Then, with $\bar{\sigma}(T,x) = \hat{\sigma}(T, x - cT)$

$$D(T,x) = \hat{D}(T, x - cT)$$

solves (2.13).

Proof. The variable transformation $\hat{x} = x - cT$ leads to

$$D_T = \hat{D}_T - c\hat{D}_{\hat{x}}, \quad D_x = \hat{D}_{\hat{x}} \ , \quad D_{xx} = \hat{D}_{\hat{x}\hat{x}},$$

what, taking into account that the initial condition does not change for $T = 0$, directly yields the proposition. □

Note that the existence and uniqueness of strong solutions to the problems above is not discussed here. Results on this can be found in Cont and Voltchkova (2005) and the references cited therein. Instead, section 3.1 deals with weak solutions. Prior to that, the next section provides some fundamentals about the calibration of the model parameters.

2.2 Calibration of Model Parameters

We have seen in the previous section that option pricing models involve several parameters which have to be set in a meaningful manner. As already mentioned, $r(T)$ is given at the market independent of the certain underlying. However, considering the jump-diffusion model, the volatility function $\sigma(T,x)$, the jump intensity λ and the density function of the jump sizes $f(y)$ are not known a priori.

These parameters shall contain the latest market information. Thus, in practice, the parameters are chosen such that the corresponding model prices fit to frequently traded standard derivatives, e.g. European call and put options. Afterwards, the fitted model can be used to price new options, even exotic ones via Monte Carlo simulation.

To be more precise, what is known today are market prices D_i^M for calls on the specific underlying with certain maturities T_i and certain strike prices K_i, $i = 1, \ldots, M$.

Today – i.e. at $t_0 = 0$ – the price of the underlying S_0 is given, thus, the forward equation introduced in remark 2.1.13 can be used to calculate all model prices corresponding to the different market prices in one sweep.

Dupire (1994) proposes in his local volatility model without jumps to solve the forward equation (2.12) (with $\lambda = 0$) for the volatility function, i.e.

$$\sigma^2(T,K) = \frac{2\tilde{D}_T(T,K) + 2r(T)K\tilde{D}_K(T,K)}{K^2\tilde{D}_{KK}(T,K)}.$$

Assuming that market prices for every strike and every maturity are given, the volatility function could be identified easily with the formula above. If jumps are included in the model, and again market prices for every strike and a certain maturity are known, Cont and Tankov (2004b) propose a technique to deduce the (here constant) volatility and the Lévy measure from the characteristic function of the underlying price. However, both approaches use market prices and their derivatives with respect to K and/or T to obtain the parameters.

In practice, the number of given market prices is usually limited to only a few combinations (T_i, K_i). This leads to the necessity of extra- and interpolation, what – in connection with observation errors – can lead to even increased mismatches in the derivatives.

So, usually the identification of the model parameters is done in a least-squares formulation (see Andersen and Andreasen (2000) or Cont and Tankov (2004b) for instance), where we compare the given market prices with the corresponding model prices.

The mathematical formulation of this approach is given in the following definition.

Definition 2.2.1. ***(Calibration problem)***
The calibration problem consists of finding parameters $\sigma(\cdot,\cdot)$, λ and $f(\cdot)$ that solve the following constrained minimization problem

$$\min_{\tilde{D},\sigma,\lambda,f} J(\tilde{D},\sigma,\lambda,f) := \frac{1}{2}\sum_{i=1}^{M}\left(\tilde{D}(T_i,K_i) - D_i^M\right)^2 \tag{2.15}$$

$$\begin{aligned} s.t. \quad & \tilde{D}_T - \frac{1}{2}\sigma^2(T,K)K^2\tilde{D}_{KK} + (r(T)-\lambda\zeta)K\tilde{D}_K + \lambda(1+\zeta)\tilde{D} \\ & - \lambda\int_{-\infty}^{+\infty}\tilde{D}(T,Ke^{-y})e^y f(y)\,dy = 0, \\ & (T,K) \in [0,T_{max}) \times (0,\infty) \\ & \tilde{D}(0,K) = \max\{S_0 - K, 0\}, \quad K \in (0,\infty), \end{aligned}$$

where D_i^M are market prices for European call options with strike K_i and maturity T_i, $i = 1,\dots,M$.

Thus, the calibration problem is a PIDE constrained optimization problem. Note that for one function evaluation of J, the PIDE constraint has to be solved only once, what is due to the existence of the forward equation.

Since only a finite number of observations, M, is given at the market, and the volatility $\sigma(\cdot,\cdot)$ and the density $f(\cdot)$ are functions, the problem above is clearly underdetermined. This can be eased by parameterizing the local volatility and the density function. One can assume a certain distribution of the jump sizes, for instance the approaches by Merton (1976) or Kou (2002), with two or three parameters, respectively (cf. example 2.1.8). It is similar concerning the local volatility function. In the case of an implied volatility smile, the local volatility function may be parameterized by a quadratic function in space direction. We mention also linear and cubic spline interpolations with a limited number of grid points. In time direction, piecewise linear and also piecewise constant functions are used. For a survey on this methods, we refer to Lörx (2012).

But even in a parametric approach with only few variables, the optimization problem is still ill-conditioned. Cont and Tankov (2004b) provide a good example regarding Merton's jump diffusion model with constant volatility, i.e. four parameters in total. They observed a tradeoff between jump intensity λ and volatility σ. Given a certain optimization error, a higher volatility σ combined with a lower jump intensity λ – and vice versa – leads to similar results, i.e. a comparable error level. Hence, it is necessary to take care of this problem.

Otherwise two calibration runs with slightly different market data may lead to the same error but totally different parameters.

They propose to add a regularization term that compares the Lévy measure that is to be calibrated with a predetermined Lévy measure. For instance, this can be a result from a statistical analysis or from a former calibration run to guarantee that the values do not oscillate.

Andersen and Andreasen (2000) have split the calibration of the jump-diffusion model into two stages. They first keep the volatility fixed and constant, wherein they use an average value of the given implied volatility surface, and then, after having calculated the jump parameters, these are kept fixed and the volatility function is fitted to further reduced the least-square error. Using an appropriate regularization term, this procedure can also be merged to one stage.

Although a proper regularization and parameterization is very important to get a reliable model, this topic exceeds the scope of this thesis, and in the numerical results in section 3.3.3 where a real-world example is studied, these issues are not the main focus.

Chapter 3

Numerical Solution of the Calibration Problem

This chapter is devoted to the numerical solution of the calibration problem that has been introduced in the previous chapter. As partial integro-differential equations (PIDE) are often hard to solve, their efficient solution will play the most important role.

PIDEs arise in several fields of research. We mention for example biological applications discussed in Armstrong et al. (2006) or Gerisch (2010). However, the occurrence in finance is also a field of recent research, e.g. in Andersen and Andreasen (2000), Matache et al. (2004) Cont and Voltchkova (2005) Briani et al. (2007), Sachs and Strauss (2008). The most challenging part here is the nonlocal integral term since it yields dense matrices when applying a spatial discretization like finite elements or finite differences. Thus, a fully implicit time discretization – which is preferable in terms of numerical stability – is rather challenging. Using the special structure of the integral term, we are able to solve implicit methods like the Crank-Nicolson scheme by a preconditioned GMRES algorithm with a complexity of $\mathcal{O}(n_x \log_2 n_x)$ per time step, an approach that is – to the knowledge of the author – quite new in this context and also competitive compared to other known approaches as is shown in the numerical results.

Given an efficient numerical method for the solution of the PIDE constraint, we turn to the numerical solution of the corresponding calibration problem (2.15). The calibration from a numerical point of view is topic of several articles and books. We refer to Achdou and Pironneau (2005), Düring et al. (2008) and especially Andersen and Andreasen (2000) where jump-diffusion models are addressed. There are many interesting issues, e.g. a proper parameterization of the parameter functions that are to be calibrated or proper regularization terms. However, we focus on two different aspects. Firstly, we are interested in the difference of the two approaches that are frequently discussed: 'first discretize, then optimize' or vice versa. The influence on the time discretization scheme and the gradient accuracy are of special interest. Secondly, taking the constrained optimization problem (2.15) as an 'reduced' unconstrained optimization problem, we are interested in the efficiency of two different optimization algorithms, namely a Gauß-Newton and a quasi-Newton algorithm. Both issues are analyzed by means of numerical results.

The outline is as follows. In the first section 3.1 we discuss in detail the existence and uniqueness of weak solutions for the partial integro-differential equation described in (2.13). We also introduce artificial boundary conditions to localize the problem to a bounded domain. Section 3.2 deals with the numerical solution of the PIDE by the method of lines. Since a weak formulation has been derived, the spatial variable is discretized by a finite

element approach and afterwards, the time is discretized by implicit θ-schemes. The integral part in the PIDE combined with an implicit time discretization leads to dense linear systems of equations, thus, we take care of this problem in section 3.2.3 using a preconditioned GMRES algorithm and present numerical results subsequently. In section 3.3 we study the calibration problem, a constrained optimization problem. We discuss the difference between 'first discretize, then optimize' or vice versa, and quasi-Newton or Gauß-Newton method, respectively.

3.1 Weak Formulation of the PIDE

The first section of this chapter deals with the variational formulation and the existence and uniqueness of weak solutions to the option pricing PIDE introduced in section 2.1.3:

$$\begin{aligned} D_T - \frac{1}{2}\bar{\sigma}^2(T,x)D_{xx} + \Big(r(T) + \frac{1}{2}\bar{\sigma}^2(T,x) - \lambda\zeta\Big)D_x + \lambda(1+\zeta)D \\ -\lambda \int_{-\infty}^{+\infty} D(T, x-y)e^y f(y)\, dy = 0, \\ (T,x) \in [0, T_{max}) \times (-\infty, \infty) \\ D(0,x) = \max\{1 - e^x, 0\} =: D_0(x), \quad x \in (-\infty, \infty). \end{aligned} \tag{3.1}$$

To be able to discretize the spatial variable by a finite element approach, the first step is a variational formulation of the problem. There is a vast literature on this topic. For instance, Dautray and Lions (1992) provide some fundamental results that are used here in parts. Regarding partial integro-differential equations arising in finance Matache et al. (2004) derive a weak formulation of a slightly different problem.

First, one makes the observation that the initial condition of (2.13) is not $L^2(\mathbb{R})$-integrable as $D_0(x) = \max\{1 - e^x, 0\} \overset{x\to-\infty}{\longrightarrow} 1$. So, we need to work with weighted function spaces. Matache et al. (2004) have derived a variational formulation for the case of time-independent coefficients for the backward PIDE, i.e. in the variables S and t, not K and T. This concept will be extended to suit the Dupire-like forward PIDE with time- and space-dependent coefficient functions.

Definition 3.1.1. ***(Weighted function spaces)***

a) $L^2_{-\mu}(\mathbb{R}) := \{v \in L^1_{loc}(\mathbb{R}) : v(\cdot)e^{-\mu|\cdot|} \in L^2(\mathbb{R})\}$
with inner product $\langle v, w\rangle_{L^2_{-\mu}} := \int_{\mathbb{R}} v(x)w(x)e^{-2\mu|x|}dx$,

b) $H^1_{-\mu}(\mathbb{R}) := \{v \in L^1_{loc}(\mathbb{R}) : v(\cdot)e^{-\mu|\cdot|}, v'(\cdot)e^{-\mu|\cdot|} \in L^2(\mathbb{R})\}$
with inner product $\langle v, w\rangle_{H^1_{-\mu}} := \langle v, w\rangle_{L^2_{-\mu}} + \langle v', w'\rangle_{L^2_{-\mu}}$.

Remark 3.1.2. *Together with their induced norms* ($||\cdot|| := \sqrt{\langle\cdot,\cdot\rangle}$) *the spaces in definition 3.1.1 are Hilbert spaces.*

It is clear that $D_0(\cdot) \in H^1_{-\mu}(\mathbb{R})$ for all $\mu > 0$. Now, we motivate the variational formulation of (3.1) in the following lines. First we multiply the PIDE by $w(x)e^{-2\mu|x|}$ and integrate over

$\mathbb{R}$, where $\mu > 0$ and w is an arbitrary function in $C_0^\infty(\mathbb{R})$.

$$\int_{\mathbb{R}} D_T(T,x)w(x)e^{-2\mu|x|}dx - \int_{\mathbb{R}} \frac{\sigma^2(T,x)}{2} D_{xx}(T,x)w(x)e^{-2\mu|x|}dx$$
$$+ \int_{\mathbb{R}} \left(r(T) + \frac{\sigma^2(T,x)}{2} - \lambda\zeta \right) D_x(T,x)w(x)e^{-2\mu|x|}dx$$
$$+ \int_{\mathbb{R}} \lambda(1+\zeta)D(T,x)w(x)e^{-2\mu|x|}dx - \lambda \int_{\mathbb{R}} \int_{\mathbb{R}} D(T,x-y)w(x)e^{-2\mu|x|}e^y f(y)dy\, dx = 0.$$

If the second term of this equation is integrated by parts, we obtain the following equation

$$\int_{\mathbb{R}} D_T(T,x)w(x)e^{-2\mu|x|}dx + a^{-\mu}(T; D(T,\cdot), w(\cdot)) = 0,$$

where the bilinear form $a^{-\mu}$ is defined as:

Definition 3.1.3. ***(Bilinear form $a^{-\mu}$)***
Let λ, ζ be given constants and assume that $r(T)$, $\sigma(T,\cdot)$, $\sigma(T,\cdot)_x$ are continuous and bounded functions on $\mathbb{R}$. For each constant $\mu > 0$ and $T > 0$ the bilinear form

$$a^{-\mu}(T;\cdot,\cdot) : H^1_{-\mu}(\mathbb{R}) \times H^1_{-\mu}(\mathbb{R}) \to \mathbb{R}$$

is defined as

$$a^{-\mu}(T;\ v,w) := \int_{\mathbb{R}} \frac{\sigma^2(T,x)}{2} v'(x)w'(x)e^{-2\mu|x|}dx \tag{3.2}$$
$$+ \int_{\mathbb{R}} \left(r(T) + \frac{\sigma^2(T,x)}{2} - \lambda\zeta + \frac{(\sigma^2(T,x))_x}{2} + \sigma^2(T,x)\mu\ \operatorname{sgn}(x) \right) v'(x)w(x)e^{-2\mu|x|}dx$$
$$+ \int_{\mathbb{R}} \lambda(1+\zeta)v(x)w(x)e^{-2\mu|x|}dx - \lambda \int_{\mathbb{R}} \int_{\mathbb{R}} v(x-y)w(x)e^{-2\mu|x|}e^y f(y)dy\, dx,$$

where $\operatorname{sgn}(x)$ *denotes the sign-function.*

Regarding the initial condition we proceed analogously:

$$\int_{\mathbb{R}} D(0,x)w(x)e^{-2\mu|x|}\, dx = \int_{\mathbb{R}} D_0(x)w(x)e^{-2\mu|x|}\, dx.$$

We further introduce a space that is of interest in terms of existence and uniqueness of solutions to the variational formulation:

Definition 3.1.4. ***(Solution space)***
$W([a,b],V) := \left\{ u : u \in L^2((a,b),V), \quad u' \in L^2((a,b),V^*) \right\} \quad a,b \in \mathbb{R}$*, where V is a Hilbert space with its dual V^*.*

Hence the variational formulation of the PIDE (3.1) can be expressed in the following form:

Definition 3.1.5. ***(Weak formulation of the PIDE)***
The variational formulation of the PIDE (3.1) consists of finding $D \in W([0, T_{max}], H^1_{-\mu}(\mathbb{R}))$ *such that for all* $T \in (0, T_{max}]$

$$\frac{d}{dT}\langle D(T,\cdot), w(\cdot)\rangle_{L^2_{-\mu}} + a^{-\mu}(T; D(T,\cdot), w(\cdot)) = 0 \quad \forall\ w \in H^1_{-\mu}(\mathbb{R}) \tag{3.3}$$

holds with initial condition

$$\langle D(0,\cdot), w(\cdot)\rangle_{L^2_{-\mu}} = \langle D_0(\cdot), w(\cdot)\rangle_{L^2_{-\mu}} \ \forall\ w \in H^1_{-\mu}(\mathbb{R}). \tag{3.4}$$

Next we take care of the solvability of the problem above. For this purpose, we need to introduce some assumptions on the coefficient functions.

Assumption 3.1. *For each* $T \in [0, T_{max}]$*, let* $\sigma(T,\cdot)$ *be continuously differentiable on* $\mathbb{R}$*. Furthermore, let* $r(\cdot)$*,* $\sigma(\cdot, x)$ *and* $\sigma_x(\cdot, x)$ *be uniformly Lipschitz-continuous functions in the variable* T *with Lipschitz constants* $r_{lip}, \sigma_{lip}, \sigma_{x,lip}$*. We assume that there are constants* $r_{max}, \sigma_{min}, \sigma_{max}, \sigma_{der}$ *satisfying*

$$\begin{aligned} 0 \le r(T) &\le r_{max} && \forall\ T \in [0, T_{max}], \\ 0 < \sigma_{min} \le \sigma(T,x) &\le \sigma_{max} && \forall\ (T,x) \in [0, T_{max}] \times \mathbb{R}, \\ |\sigma_x(T,x)| &\le \sigma_{der} && \forall\ (T,x) \in [0, T_{max}] \times \mathbb{R}. \end{aligned}$$

In the following theorem we prove that the bilinear form defined in definition 3.1.3 is bounded and that Gårding's inequality holds. For this to hold, we further need an assumption on the asymptotic decay of the function f.

Assumption 3.2. *For some* $\mu > 0$ *assume that* $\int\limits_{\mathbb{R}} e^{y+\mu|y|} y f(y)\ dy < \infty$.

Later we will show that this condition is usually satisfied for common choices of f in finance.

Theorem 3.1.6. ***(Properties of*** $a^{-\mu}$***)***
If assumptions 3.1 and 3.2 hold, then there exist constants $c_1, c_2 > 0$ *and* $c_3 \in \mathbb{R}$ *independent of* $T \in [0, T_{max}]$*, such that the following inequalities hold for the bilinear form (3.2):*

a) Continuity: $|a^{-\mu}(T;\ v, w)| \le c_1\ ||v||_{H^1_{-\mu}} ||w||_{H^1_{-\mu}} \qquad \forall\ T \in [0, T_{max}]$

b) Gårding inequality: $a^{-\mu}(T;\ v, v) + c_3\ ||v||^2_{L^2_{-\mu}} \ge c_2\ ||v||^2_{H^1_{-\mu}} \qquad \forall\ T \in [0, T_{max}]$

c) Lipschitz continuity in time:

$$|a^{-\mu}(T_1;\ v, w) - a^{-\mu}(T_2;\ v, w)| \le c_{lip}\ |T_1 - T_2|\ ||v||_{H^1_{-\mu}} ||w||_{H^1_{-\mu}} \qquad \forall\ v, w \in H^1_{-\mu},\ T_1, T_2 \in [0, T_{max}]$$

Proof. In order to prove the continuity of the bilinear form, we estimate the terms in $a^{-\mu}$ separately. First,

$$\Big| \int_{\mathbb{R}} \frac{\sigma^2(T,x)}{2} v'(x) w'(x) e^{-2\mu|x|} \, dx \Big| \leq \frac{\sigma_{max}^2}{2} ||v||_{H^1_{-\mu}} ||w||_{H^1_{-\mu}}. \tag{3.5}$$

If we set $k_1 = |r_{max} + \frac{\sigma_{max}^2}{2} + \lambda\zeta + \sigma_{max}\sigma_{der} + \mu\sigma_{max}^2|$, we obtain for the next term of the bilinear form:

$$\begin{aligned} &\Big| \int_{\mathbb{R}} \Big(r(T) + \frac{\sigma^2(T,x)}{2} - \lambda\zeta + (\frac{\sigma^2(T,x)}{2})_x + \sigma^2(T,x)\mu \ \operatorname{sgn}(x) \Big) v'(x) w(x) e^{-2\mu|x|} \, dx \Big| \\ &\leq k_1 \langle v'(x), w(x) \rangle_{L^2_{-\mu}} \leq k_1 ||v'||_{L^2_{-\mu}} ||w||_{L^2_{-\mu}} \leq k_1 ||v||_{H^1_{-\mu}} ||w||_{H^1_{-\mu}}. \end{aligned} \tag{3.6}$$

The two remaining terms are merged using $1 + \zeta = \int_{\mathbb{R}} e^y f(y) dy$ (see remark 2.1.13)

$$\begin{aligned} g(v,w) :=& \lambda \Big| \int_{\mathbb{R}} \Big((1+\zeta) v(x) w(x) \ - \int_{\mathbb{R}} v(x-y) e^y f(y) \ dy \ w(x) \Big) e^{-2\mu|x|} dx \Big| \\ &= \lambda \Big| \int_{\mathbb{R}} \int_{\mathbb{R}} (v(x) - v(x-y)) e^y f(y) \ dy \ w(x) e^{-2\mu|x|} \ dx \Big| \\ &= \lambda \Big| \int_{\mathbb{R}} \int_{\mathbb{R}} \int_0^1 v'(x - \xi y) \ y \ d\xi \ e^y f(y) \ dy \ w(x) e^{-2\mu|x|} \ dx \Big|. \end{aligned}$$

We rearrange the order of integration, use the Cauchy-Schwarz inequality and a variable transformation by introducing $z = x - \xi y$

$$\begin{aligned} g(v,w) &= \lambda \Big| \int_0^1 \int_{\mathbb{R}} \int_{\mathbb{R}} v'(x - \xi y) w(x) e^{-2\mu|x|} \ dx \ y \ e^y f(y) \ dy \ d\xi \Big| \\ &\leq \lambda \Big| \int_0^1 \int_{\mathbb{R}} \Big(\int_{\mathbb{R}} v'^2(x - \xi y) e^{-2\mu|x|} \ dx \Big)^{1/2} ||w||_{L^2_{-\mu}} \ y \ e^y f(y) \ dy \ d\xi \Big| \\ &\leq \lambda \Big| \int_{\mathbb{R}} \Big(\int_{\mathbb{R}} v'^2(z) e^{-2\mu|z|} e^{2\mu|y|} \ dz \Big)^{1/2} ||w||_{L^2_{-\mu}} \ y \ e^y f(y) \ dy \Big| \\ &\leq \lambda \Big| \int_{\mathbb{R}} y \ e^{\mu|y| + y} f(y) \ dy ||v'||_{L^2_{-\mu}} ||w||_{L^2_{-\mu}} \Big|. \end{aligned} \tag{3.7}$$

By assumption 3.2 we can define a constant $k_2 = \int_{\mathbb{R}} e^{y+\mu|y|} y f(y) \ dy$ and finally obtain

$$g(v,w) \leq \lambda k_2 ||v'||_{L^2_{-\mu}} ||w||_{L^2_{-\mu}} \leq \lambda k_2 ||v||_{H^1_{-\mu}} ||w||_{H^1_{-\mu}}. \tag{3.8}$$

Collecting the estimates from (3.5), (3.6) and (3.8) the continuity of the bilinear form is

proven

$$|a^{-\mu}(T;\ v,w)| \leq \Big(\frac{\sigma_{max}^2}{2} + k_1 + \lambda k_2\Big)||v||_{H^1_{-\mu}}||w||_{H^1_{-\mu}}.$$

Next we prove Gårding's inequality for the bilinear form. We estimate the first term by

$$\begin{aligned}\int_{\mathbb{R}} \frac{\sigma^2(T,x)}{2} v'^2(x) e^{-2\mu|x|}\ dx &\geq \frac{\sigma_{min}^2}{2}\int_{\mathbb{R}} v'^2(x)e^{-2\mu|x|}\ dx \\ &= \frac{\sigma_{min}^2}{2}||v'||^2_{L^2_{-\mu}} = \frac{\sigma_{min}^2}{2}||v||^2_{H^1_{-\mu}} - \frac{\sigma_{min}^2}{2}||v||^2_{L^2_{-\mu}}. \end{aligned} \tag{3.9}$$

Due to (3.6), (3.7) we obtain for the remaining terms

$$\begin{aligned}&\int_{\mathbb{R}} \Big(r(T) + \frac{\sigma^2(T,x)}{2} - \lambda\zeta + (\frac{\sigma^2(T,x)}{2})_x + \sigma^2(T,x)\mu\ \operatorname{sgn}(x)\Big)v'(x)v(x)e^{-2\mu|x|}\ dx + \\ &\qquad \lambda\int_{\mathbb{R}} \Big((1+\zeta)v(x)w(x)\ - \int_{\mathbb{R}} v(x-y)e^y f(y)\ dy\ v(x)\Big)e^{-2\mu|x|}dx \\ &\geq -(k_1+k_2\lambda)||v'||_{L^2_{-\mu}}||v||_{L^2_{-\mu}} \geq -\frac{k_{arb}^2}{4}||v'||^2_{L^2_{-\mu}} - \frac{(k_1+k_2\lambda)^2}{k_{arb}^2}||v||^2_{L^2_{-\mu}} \\ &\geq -\frac{k_{arb}^2}{4}||v||^2_{H^1_{-\mu}} - \frac{(k_1+k_2\lambda)^2}{k_{arb}^2}||v||^2_{L^2_{-\mu}} \end{aligned} \tag{3.10}$$

for an arbitrary constant $k_{arb} > 0$. If we choose $k_{arb} = \sigma_{min}$, then the estimates (3.9) and (3.10) lead to

$$a^{-\mu}(T;\ v,v) \geq \frac{\sigma_{min}^2}{4}||v||^2_{H^1_{-\mu}} - \Big(\frac{(k_1+k_2\lambda)^2}{\sigma_{min}^2} + \frac{\sigma_{min}^2}{2}\Big)||v||^2_{L^2_{-\mu}}.$$

Finally, we address the Lipschitz continuity of the bilinear form. Similar to the proof of the continuity we divide the bilinear form into three different parts. Since the term $g(v,w)$ is independent of time, it can be ignored. For the next term we obtain

$$\Big|\int_{\mathbb{R}} \frac{\sigma^2(T_1,x) - \sigma^2(T_2,x)}{2} v'(x)w'(x)e^{-2\mu|x|}\ dx\Big| \leq |T_1 - T_2|\frac{\sigma_{lip}\sigma_{max}}{2}||v||_{H^1_{-\mu}}||w||_{H^1_{-\mu}}, \tag{3.11}$$

where σ_{lip} is the Lipschitz constant of $\sigma(\cdot,x)$. The remaining term can be treated analogously:

$$\begin{aligned}&\Big|\int_{\mathbb{R}} \Bigg(\Big(r(T_1) + \frac{\sigma^2(T_1,x)}{2} - \lambda\zeta + (\frac{\sigma^2(T_1,x)}{2})_x + \sigma^2(T_1,x)\mu\ \operatorname{sgn}(x)\Big) \\ &\quad - \Big(r(T_2) + \frac{\sigma^2(T_2,x)}{2} - \lambda\zeta + (\frac{\sigma^2(T_2,x)}{2})_x + \sigma^2(T_2,x)\mu\ \operatorname{sgn}(x)\Big)\Bigg)v'(x)w(x)e^{-2\mu|x|}\ dx\Big|\end{aligned}$$

$$\leq |T_1 - T_2|(r_{lip} + \sigma_{lip}\sigma_{max} + \sigma_{lip}\sigma_{der} + \sigma_{x,lip}\sigma_{max} + 2\sigma_{lip}\sigma_{max}\mu)\ ||v||_{H^1_{-\mu}}||w||_{H^1_{-\mu}}, \quad (3.12)$$

which completes the proof. □

It is well-known that the boundedness of $a^{-\mu}$ together with the weak coercivity yields the existence of a unique solution of the variational equality.

Theorem 3.1.7. ***(Existence and Uniqueness of a weak solution)***
If assumptions 3.1 and 3.2 hold, then there exists a unique solution $D \in W([0, T_{max}], H^1_{-\mu}(\mathbb{R}))$ of the problem specified in definition 3.1.5.

Proof. Under the given assumptions, we can define a new bilinear form on $L^2_{-\mu}$

$$\tilde{a}^{-\mu}(T;\ v, w) = a^{-\mu}(T;\ v, w) + c_3\langle v, w\rangle_{L^2_{-\mu}},$$

which satisfies all the assumptions on the boundedness, the coercivity and Lipschitz continuity as spelled out previously.

According to theorem 3.1.6 this bilinear form is uniformly bounded and coercive on the Hilbert space $H^1_{-\mu}$. Further notice that the function spaces $V = H^1_{-\mu}$ and $H = L^2_{-\mu}$ form a Gelfand triple with dense embeddings ($V \hookrightarrow H = H^* \hookrightarrow V^*$). By Dautray and Lions (1992, pp. 509 ff.) there exists a unique solution $\tilde{D} \in W([0, T_{max}], H^1_{-\mu}(\mathbb{R}))$ of the variational equality for all $T \in (0, T_{max}]$

$$\frac{d}{dT}\langle \tilde{D}(T, \cdot), w(\cdot)\rangle_{L^2_{-\mu}} + \tilde{a}^{-\mu}(T; \tilde{D}(T, \cdot), w(\cdot)) = 0 \quad \forall\ w \in H^1_{-\mu}(\mathbb{R}) \quad (3.13)$$

with initial condition

$$\langle \tilde{D}(0, \cdot), w(\cdot)\rangle_{L^2_{-\mu}} = \langle D_0(\cdot), w(\cdot)\rangle_{L^2_{-\mu}}\ \forall\ w \in H^1_{-\mu}(\mathbb{R}). \quad (3.14)$$

Then it is easy to verify that

$$D(T, x) = e^{-c_3 T}\tilde{D}(T, x)$$

satisfies the desired variational equality (3.3) with initial condition (3.4). □

Note that the coefficient functions $\sigma(T, x)$ and $r(T)$ (assumption 3.1) or the bilinear form (theorem 3.1.6 c)), respectively, do not need to be Lipschitz continuous in time to show the result above. The Lipschitz continuity will be of importance later on.

After the existence and uniqueness of a weak solution of the PIDE has been shown, we return to assumption 3.2 and show that this requirement is not too restrictive since it is fulfilled by several popular models (see example 2.1.8).

Remark 3.1.8. *The following models specified by the functions $f(y)$ defined below satisfy the requirement $\int_{\mathbb{R}} e^{y+\mu|y|} y f(y)\ dy < \infty$.*

a) Merton (1976): $f(y) = \dfrac{1}{\sqrt{2\pi}\sigma_M} exp\Big\{ - \dfrac{(y - \mu_M)^2}{2\sigma_M^2}\Big\}$,

b) Kou (2002): $f(y) = p\ \lambda^+\ e^{-\lambda^+ y}\ 1_{\{y \geq 0\}} + (1 - p)\ \lambda^-\ e^{\lambda^- y}\ 1_{\{y<0\}}$
with $\lambda^+ > 1$, $\lambda^- > 0$ and $p \in [0, 1]$.

Proof. Regarding the Merton model, the finiteness of the integral term is clear since we have a (shifted) $-y^2$-term in the density function.

For the proof in the case of the Kou model, we divide the integral into two terms. First, for a given $\lambda^+ > 1$ there is a $\mu > 0$ sufficiently small, such that $-\lambda^+ + \mu + 1 < 0$. Hence, we obtain

$$\int_0^{+\infty} p\ \lambda^+\ e^{-\lambda^+ y} e^{\mu y + y} y\ dy \int_0^{+\infty} p\ \lambda^+ e^{(-\lambda^+ + \mu + 1)y} y\ dy < \infty.$$

For the other half of the integration interval, let $\mu > 0$ be small enough such that $\lambda^- - \mu + 1 > 0$, which can be achieved since $\lambda^- > 0$ by assumption. Then

$$\int_{-\infty}^0 (1-p)\ \lambda^-\ e^{\lambda^- y} e^{-\mu y + y} y\ dy = \int_{-\infty}^0 (1-p)\ \lambda^- e^{(\lambda^- - \mu + 1)y} y\ dy < \infty.$$

□

Note that infinite activity models like Variance Gamma or CGMY with $\sigma = 0$ do not fit the theory presented here and need to be analyzed differently.

Localization

For a numerical solution, the infinite space domain has to be restricted. For this process, which is often called 'localization' in literature, the behavior of the solution at the boundaries has to be studied. We do this from an economical point of view. When we want to know, how the transformed call price $D(T, x)$ acts for x very small or very large, respectively, we better think about the behavior of $\tilde{D}(T, K)$ before the variable transformation, where K is close to zero or K is large. Since there is a no-arbitrage assumption in our model, the price of a call is equal to the value of a perfect hedge. So, if the strike K is close to zero and in particular much smaller than the current underlying price, then the probability for the call holder to exercise the call option is close to one. Therefore, the call issuer will most likely have to deliver the underlying. To hedge his risk when the contract is concluded at $t_0 = 0$, he needs to buy an underlying for the price S_0. When the call option is exercised at maturity T, the issuer receives the (small) strike price K, which is worth $Ke^{-\int_0^T r(\tau)d\tau}$ – discounted with the given interest rate r – today. Thus, the value of the hedge or the price of the call option, respectively, is $\tilde{D}(T, K) \overset{K \to 0}{\longrightarrow} S_0 - Ke^{-\int_0^T r(\tau)d\tau}$.

On the other hand, if the strike price K is very large compared to the current underlying price, the probability for the underlying to reach the strike tends to zero. So, the probable payoff of our call as well as the corresponding price is zero or – written mathematically – $\tilde{D}(T, K) \overset{K \to +\infty}{\longrightarrow} 0$. Given a small lower bound $\underline{x}$, a large upper bound $\overline{x}$ and the variable transformation $x = \ln\left(\frac{K}{S_0}\right)$, we get

$$D(T, \underline{x}) \approx 1 - e^{\underline{x}} e^{-\int_0^T r(\tau)d\tau}\ , \qquad D(T, \overline{x}) \approx 0. \tag{3.15}$$

In the case of a PDE, these values would be used as Dirichlet boundary conditions for $[\underline{x}, \overline{x}]$ and error estimates could be derived using the maximum principle. Due to the additional

integral term, which is nonlocal, we need information on the domain $\{z = x - y : x \in [\underline{x}, \overline{x}], y \in supp(f)\}$, where f is the density function of the jump sizes that typically has support $supp(f) = \mathbb{R}$. Hence, the boundaries have to be defined not only for the points $\underline{x}$ and $\overline{x}$ but on the intervals $(-\infty, \underline{x}]$ and $[\overline{x}, \infty)$. Also error estimates get more complicated.

If $D^b(T,x) \in W([0,T_{max}], H^1_{-\mu})$ denotes a function fulfilling the boundary conditions (3.15) on $(-\infty, \underline{x}]$ and $[\overline{x}, \infty)$, respectively, it is easy to verify that the weak formulation (3.3), (3.4) can be transformed in the following way.

Lemma 3.1.9. *Let $D^b(T,x) \in W([0,T_{max}], H^1_{-\mu})$ be given as described above. Further, let $D^h \in W([0,T_{max}], H^1_{-\mu}(\mathbb{R}))$ be the solution of*

$$\frac{d}{dT}\langle D^h(T,\cdot), w(\cdot)\rangle_{L^2_{-\mu}} + a^{-\mu}(T; D^h(T,\cdot), w(\cdot)) = L^{-\mu}(T;\ w(\cdot)) \quad \forall\ w \in H^1_{-\mu}(\mathbb{R}) \tag{3.16}$$

with initial condition

$$\langle D^h(0,\cdot), w(\cdot)\rangle_{L^2_{-\mu}} = \langle D^h_0(\cdot), w(\cdot)\rangle_{L^2_{-\mu}} \ \forall\ w \in H^1_{-\mu}(\mathbb{R}), \tag{3.17}$$

where $L^{-\mu}(T;\ \cdot) := -\frac{d}{dT}\langle D^b(T,x), \cdot\rangle_{L^2_{-\mu}} - a^{-\mu}(T; D^b(T,x), \cdot) \in (H^1_{-\mu}(\mathbb{R}))^*\ \forall\ T \in (0, T_{max}]$ and $D^h_0(\cdot) := D_0(\cdot) - D^b(0,\cdot)$.
Then $D(T,x) := D^h(T,x) + D^b(T,x)$ solves (3.3), (3.4).

The superscript h indicates homogeneous boundary conditions, i.e. that D^h tends to zero for $x \to \pm\infty$. Hence, a localized approximation to the problem above can be defined on the Sobolev space $H^1_0(\underline{x}, \overline{x})$ as follows.

Definition 3.1.10. ***(Localized problem)***
A localized variational formulation of the PIDE (3.1) consists of finding a function $\bar{D} \in W([0,T_{max}], H^1_0(\underline{x}, \overline{x}))$ such that for all $T \in (0, T_{max}]$

$$\frac{d}{dT}\langle \bar{D}(T,\cdot), w(\cdot)\rangle_{L^2} + a(T; \bar{D}(T,\cdot), w(\cdot)) = L(T;\ w(\cdot)) \quad \forall\ w \in H^1_0(\underline{x}, \overline{x}) \tag{3.18}$$

holds with initial condition

$$\langle \bar{D}(0,\cdot), w(\cdot)\rangle_{L^2} = \langle D^h_0(\cdot), w(\cdot)\rangle_{L^2} \ \forall\ w \in H^1_0(\underline{x}, \overline{x}), \tag{3.19}$$

where $a(T;\cdot,\cdot) : H^1_0(\underline{x}, \overline{x}) \times H^1_0(\underline{x}, \overline{x}) \to \mathbb{R}$ denotes the restriction of $a^0(T;\ v, w)$ for functions v, w with $v(x) = w(x) = 0\ \forall\ x \in \mathbb{R}\backslash(\underline{x}, \overline{x})$. Analogously $L(T;\ \cdot) : H^1_0(\underline{x}, \overline{x}) \to \mathbb{R}$ is defined via $L^0(T,\ \cdot)$.

Note that $(D_0(\cdot) - D^b(0,\cdot))\big|_{(\underline{x},\overline{x})} \in H^1_0(\underline{x}, \overline{x})$ by definition of D^b. At this point, it further has to be stressed that the linear operator $L^{-\mu}$ is well-defined for $\mu = 0$, what is not so obvious at first sight. But due to the specific definition of the function D^b at the boundary, the time and the spatial derivatives as well as the application of the integral term lead to exponentially decreasing functions for $x \to \pm\infty$, which are $L^2(\mathbb{R})$-integrable.

Remark 3.1.11. ***(Localization error)***
It is clear that the localization leads to an approximation error. Noting that the artificial

boundaries above are identical to the initial condition $D_0(x)$ if $r = 0$ in our case, we at least can give a reference for a proof of an exponentially decreasing localization error. Namely, for the backward PIDE case, Matache et al. (2004) assume $r = 0$, use the initial condition to restrict the spatial domain and show an exponential rate of convergence for increasing boundaries.

The problem defined in (3.18) and (3.19) is discretized in the next section.

Prior to this we notice that (3.18) and (3.19) can be rewritten in vectorial form providing some advantages in terms of a simpler notation.

Remark 3.1.12. ***(Vectorial problem formulation)***
Defining $V := H_0^1(\underline{x}, \overline{x})$, there exist unique operators $A(T) \in L(V, V^*)$ and $l(T) \in V^*$ ($T \in (0, T_{max}]$) such that (3.18) and (3.19) can be rewritten as

$$\begin{aligned} &\dot{y}(T) + A(T)y(T) = l(T) \quad \forall\ T \in (0, T_{max}] \\ &y(0) = D_0^h \end{aligned}$$

in the sense of $L^2(V^*)$.

3.2 Discretization of the PIDE

The discretization of the system described in definition 3.1.10 is the focus of this section. For this purpose we use the method of lines. To be more precise, the spatial variable or the function space $H_0^1(\underline{x}, \overline{x})$, respectively, is discretized by a finite dimensional function space and the resulting system of ordinary differential equations is discretized by implicit finite difference methods. The dense linear systems of equations of the fully discretized system are solved via a preconditioned GMRES algorithm with an overall complexity of $\mathcal{O}(n_t n_x \log_2 n_x)$, an approach not used in this context so far.

3.2.1 Spatial Discretization

The separable Hilbert space $H_0^1(\underline{x}, \overline{x})$ is to be approximated by a finite element space. This approach is well-known in literature, Brenner and Scott (2008), Larsson and Thomée (2003) or Dautray and Lions (1992) can be mentioned here for fundamentals.

Definition 3.2.1. ***(Space of piecewise linear functions)***
Let $\underline{x} =: x_0 < x_1 < \ldots < x_{n_x+1} := \overline{x}$ be an equidistant partition of $(\underline{x}, \overline{x})$ with $\Delta x = x_{i+1} - x_i$ ($i = 0, \ldots, n_x$). Then $H^{n_x} := span\{\Phi_1, \ldots, \Phi_{n_x}\}$ with

$$\Phi_i(x) = \begin{cases} \Delta x^{-1}(x - x_{i-1}) & , x_{i-1} \leq x \leq x_i \\ \Delta x^{-1}(x_{i+1} - x) & , x_i < x \leq x_{i+1} \\ 0 & , else, \end{cases}$$

$i = 1, \ldots, n_x$, is called the space of piecewise linear functions.

Now the Φ_i are used as trial and test functions in (3.18), (3.19), i.e.

$$\bar{D}(T,x) \approx \sum_{i=1}^{n_x} \alpha_i(T)\Phi_i(x).$$

The coefficient functions $\alpha_i(\cdot)$ are to be determined. The semi-discretized problem can then be described as follows:

Definition 3.2.2. ***(Semi-discretized problem)***
Given the Galerkin approximation H^{n_x} to $H_0^1(\underline{x},\overline{x})$ as described above, the semi-discretized solution to (3.18) and (3.19) consists of finding $\alpha_i(\cdot) \in H^1([0,T_{max}])$ $(i=1,\dots,n_x)$ such that for all $T \in (0,T_{max}]$

$$\sum_{i=1}^{n_x} \dot{\alpha}_i(T)\langle\Phi_i,\Phi_j\rangle_{L^2} + \sum_{i=1}^{n_x} \alpha_i(T)a(T;\ \Phi_i,\Phi_j) = L(T;\ \Phi_j) \quad \forall\ j=1,\dots,n_x \tag{3.20}$$

holds with initial condition

$$\sum_{i=1}^{n_x} \alpha_i(0)\langle\Phi_i,\Phi_j\rangle_{L^2} = \langle D_0^h,\Phi_j\rangle_{L^2} \quad \forall\ j=1,\dots,n_x. \tag{3.21}$$

Remark 3.2.3. *(3.20), (3.21) can be written in equivalent matrix form as*

$$\begin{aligned} &M\dot{\alpha}(T) + A(T)\alpha(T) = L(T) \quad , \qquad T \in (0,T_{max}] \\ &M\alpha(0) = B, \end{aligned} \tag{3.22}$$

where $M \in \mathbb{R}^{n_x\times n_x}$ with $M_{ji} = \langle\Phi_i,\Phi_j\rangle_{L^2}$, $A \in \mathbb{R}^{n_x\times n_x}$ with $A_{ji}(T) = a(T;\ \Phi_i,\Phi_j)$, $L(T),B,\alpha(T) \in \mathbb{R}^{n_x}$ with $L(T) = L(T;\ \Phi_j)$ and $B_j = \langle D_0^h,\Phi_j\rangle_{L^2}$, resp. $(i,j=1,\dots,n_x)$.

After the spatial discretization, the problem to solve, (3.22), is a linear system of differential equations of first order. Before these ODEs are discretized in time, we take a closer look at the matrices occurring in remark 3.2.3 since their structure is determining for the computational effort.

Using an equidistant grid with step size Δx in definition 3.2.1, the mass matrix M is a symmetric, tridiagonal matrix with

$$M_{ii} = \tfrac{2}{3}\Delta x, \quad M_{i,i+1} = M_{i+1,i} = \tfrac{1}{6}\Delta x,$$

which has nice numerical properties. For instance, matrix-vector products with general tridiagonal matrices and their memory capacity are of complexity $\mathcal{O}(n_x)$. The tridiagonal structure is due to the local support of the basis functions Φ_i.

The time-dependent stiffness matrix $A(T)$ turns out to be more complicated. For a better analysis we split $A(T)$ into two parts. The time-independent part A^I containing the terms resulting from the double integral in the bilinear form and the rest called $A^{NI}(T)$:

$$A(T) = A^{NI}(T) + A^I. \tag{3.23}$$

It is well-known that the matrix $A^{NI}(T)$ is of tridiagonal structure, too, since it is the discretization of the convection-diffusion term. For $\sigma(T,x)$ depending on x an appropriate numerical integration has to be used.

In order to get an idea of how the entries of $A^{NI}(T)$ look like, you find below the result for the simplest Gaussian integration on intervals $[x_i, x_{i+1}]$:

$$\begin{aligned} A^{NI}_{ii}(T) &= (\tfrac{1}{2\Delta x} + \tfrac{1}{2})(\sigma^2_{i-1}(T) + \sigma^2_i(T)) + \tfrac{1}{2}(\sigma^2_{x,i-1}(T) + \sigma^2_{x,i}(T)) + \tfrac{2}{3}\lambda(1+\zeta), \\ A^{NI}_{i,i+1}(T) &= -\tfrac{1}{2\Delta x}\sigma^2_i(T) + \tfrac{1}{2}(r(T) + \tfrac{1}{2}\sigma^2_i(T) - \lambda\zeta + \tfrac{1}{2}\sigma^2_{x,i}(T)) + \tfrac{1}{6}\Delta x\lambda(1+\zeta), \\ A^{NI}_{i+1,i}(T) &= -\tfrac{1}{2\Delta x}\sigma^2_i(T) - \tfrac{1}{2}(r(T) + \tfrac{1}{2}\sigma^2_i(T) - \lambda\zeta + \tfrac{1}{2}\sigma^2_{x,i}(T)) + \tfrac{1}{6}\Delta x\lambda(1+\zeta), \end{aligned} \tag{3.24}$$

where $\sigma^2_{i-1}(T) := \sigma^2(T, x_{i-1} + \frac{\Delta x}{2})$ and $\sigma^2_{x,i-1}(T) := (\sigma^2(T, x_{i-1} + \frac{\Delta x}{2}))_x$.

Unfortunately, the matrix A^I is not of sparse type, at least not in the sense that most entries are equal to zero. But it has a special structure called 'Toeplitz'.

Definition 3.2.4. ***(Toeplitz matrix)***
A matrix $T \in \mathbb{R}^{n\times n}$ is called Toeplitz matrix if $T_{ij} = t_{j-i}$ $(i,j = 1,\dots,n)$ and $t_{-n+1},\dots,t_{n-1} \in \mathbb{R}$, thus:

$$T = \begin{pmatrix} t_0 & t_1 & \cdots & t_{n-1} \\ t_{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{-n+1} & \cdots & t_{-1} & t_0 \end{pmatrix}.$$

Remark 3.2.5. *The computational effort of a matrix-vector multiplication with a Toeplitz matrix is of complexity $\mathcal{O}(n \log_2 n)$ using fast Fourier transformation (FFT) (cf. Golub and van Loan (1996)). The memory capacity is $\mathcal{O}(n)$ since only the first column and the first row have to be stored. This can be interpreted as a different kind of sparsity.*

Regarding matrix A^I, there is the following result:

Remark 3.2.6. *For equidistantly distributed grid points, the matrix A^I with its entries*

$$A^I_{ij} = -\lambda \int_{x_{i-1}}^{x_{i+1}} \int_{x-x_{j+1}}^{x-x_{j-1}} \Phi_j(x-y)\Phi_i(x)e^y f(y)dy\,dx. \tag{3.25}$$

has Toeplitz structure.

Proof. Using the variable transformation $\bar{x} = x - \Delta x$ in (3.25) and the fact that $\Phi_{i+1}(z + \Delta x) = \Phi_i(z)$ for equidistantly distributed grid points, it is easy to show that $A^I_{ij} = A^I_{i+1,j+1}$. □

Since A^I has Toeplitz structure, a matrix-vector product can be calculated with $\mathcal{O}(n_x \log_2 n_x)$ flops (remark 3.2.5). However, the matrix usually has no zero entries. This is due to the translation in $\Phi_j(x-y)$ and the inner integral, and the fact that the density function $f(\cdot)$ in general has $supp(f) = \mathbb{R}$. Therefore the equidistant grid, which leads to the Toeplitz structure, is essential to keep the computational cost manageable.

The double integrals in (3.25) have to be approximated via numerical integration. The two-dimensional area can be divided into four different squares of size $\Delta x \times \Delta x$, where we again use Gaussian quadrature of order one. For example, denoting $g(y) := e^y f(y)$ we get $\int_{x_{i-1}}^{x_i} \int_{x-x_{j+1}}^{x-x_j} \Phi_j(x-y)\Phi_i(x)e^y f(y)dy\,dx \approx \frac{1}{4}g((i-j-1)\Delta x)$ and thus:

$$A_{ij}^I = -\lambda\tfrac{\Delta x^2}{4}\Big(g((i-j-1)\Delta x) + 2g((i-j)\Delta x) + g((i-j+1)\Delta x)\Big).$$

Numerical results have shown that a higher order integration does not lead to an improvement of the results.

Although A^I has Toeplitz structure and $A^{NI}(T)$ is sparse, the matrix $A(T) = A^{NI}(T) + A^I$ has neither of the two properties since the space-dependent volatility function destroys the constancy on the diagonals in $A^{NI}(T)$. Thus $A(T)$ is a dense matrix, what has to be taken into account in the further numerical solution. However, using the splitting above a matrix-vector-product can be realized in $\mathcal{O}(n_x \log_2 n_x)$ flops as will be discussed further below.

At this point, the work of Matache et al. (2004) has to be mentioned. In their numerical solution of the backward PIDE, they propose a different spatial discretization using special wavelet basis functions instead of the finite element basis functions, which are used above. That way, they are able to reduce the number of nonzero elements in their stiffness matrix to $\mathcal{O}(n_x \log_2 n_x)$ by ignoring very small values, leading to the same complexity regarding matrix-vector multiplications as the approach using Toeplitz matrices.

Before proceeding with the time discretization of (3.22), just note that the inhomogeneous term $F(T) := F(T;\ \Phi_j)$ involves integrals that are not restricted to a bounded interval:

$$-\lambda \int_{x_{i-1}}^{x_{i+1}} \int_{x-\underline{x}}^{+\infty} D^b(x-y)\Phi_j(x)e^y f(y)\ dy\ dx.$$

and therefore need a special treatment by, e.g., a variable transformation.

3.2.2 Time Discretization

For the discretization of the initial value problem, (3.22), in time, schemes of the θ-method are used. Given an ordinary differential equation $\dot{y}(t) = f(t, y(t))$ with initial condition $y(0) = y_0$ and $\theta \in [0,1]$, the discretization via a θ-method is given by:

$$\frac{y_{k+1} - y_k}{\Delta t} = \theta f(t_{k+1}, y_{k+1}) + (1-\theta) f(t_k, y_k), \tag{3.26}$$

where Δt is the step size, $t_k := k\Delta t$ and $y_k \approx y(t_k)$. Special cases are the forward Euler ($\theta = 0$), the backward Euler ($\theta = 1$) and the Crank-Nicolson method ($\theta = 0.5$), where all methods with $\theta \neq 0$ are implicit. In general, we use the Crank-Nicolson method, the implicit Euler scheme or – as we will see – a combination of both. After the time discretization is introduced, the fully discretized system can be specified:

Definition 3.2.7. ***(Fully discretized problem)***
Given $\theta \in [0,1]$, $T_k := k\Delta T$ $(k = 0, \ldots, n_T)$ and $T_{n_T} = T_{max}$. Then the fully discretized

solution to (3.22) consists of finding $\alpha^k \in \mathbb{R}^{n_x}$ $(k = 0, \ldots, n_T)$ *such that*

$$\begin{aligned}(M + \Delta T \theta A(T_{k+1}))\alpha^{k+1} &= \\ &= (M - \Delta T(1-\theta)A(T_k))\alpha^k + \Delta T(\theta F(T_{k+1}) + (1-\theta)F(t_k)) \qquad (3.27)\\ M\alpha^0 &= B. \qquad (3.28)\end{aligned}$$

It is well-known that problem (3.22) is very stiff, so an unstable forward Euler method would be restricted by a strong CFL condition. Among the stable schemes Crank-Nicolson would be the method of choice regarding the error of the time discretization since the central difference quotient is of order $\mathcal{O}(\Delta t^2)$. But although it is A-stable, non-smooth initial conditions often lead to oscillations. Rannacher (1984) proposes to use two half-time steps of the strongly A-stable backward Euler scheme to smooth the initial condition and proceed with Crank-Nicolson to preserve second order convergence. In Giles and Carter (2006), four backward Euler full- or four quarter time steps are named as an alternative to the original approach.

This Rannacher approach will be of special interest when we solve the adjoint equation, which is introduced in section 3.3.

Regardless of whether (3.27) is solved for $\theta = 0.5$ or $\theta = 1$, the corresponding linear systems of equations $\tilde{A}x = b$ involve a dense matrix $\tilde{A}$. A direct solver would be of complexity $\mathcal{O}(n_x^3)$.

Thus, in PIDE literature, there are primarily two sophisticated time discretization methods available that avoid the dense linear systems of equations in the context of financial applications. Andersen and Andreasen (2000) use an alternating directions implicit (ADI) method. Here, one time step is divided into two half-time steps. In the first half-time step the sparse part is treated with an implicit Euler and the dense part with an explicit Euler. In the second half-time step it is vice versa, i.e. explicit Euler is applied to the sparse part and the implicit Euler to the dense part. In total, this leads to a computational cost of $\mathcal{O}(n_x \log n_x)$ if fast Fourier transformation is used to handle the dense part.

The second approach by Cont and Voltchkova (2005) or Briani et al. (2007) uses an implicit-explicit (IMEX) splitting scheme, where the stiffness matrix is split into a sparse and a dense part analog to (3.23). As already mentioned, a fully explicit method would be restricted by a strong CFL condition, so the sparse part is treated implicitly and only the dense part explicitly by a higher-order Runge-Kutta method. This weakens the restriction on the time steps ΔT versus Δx. An example would be the so-called Midpoint-122 rule (cf. Briani et al. (2007)), which applies an explicit midpoint scheme to the integral and convection terms and an implicit midpoint scheme to the sparse diffusion part. Stability is guaranteed for $\Delta T = \mathcal{O}(\Delta x^{4/3})$, but the CFL condition is not eliminated totally. The Midpoint-122 and the ADI method lead to a second order convergence in time.

Despite these approaches to avoid the dense system, we will solve the discretization (3.27) even for fully implicit methods. Of course, this can only be done by an iterative method and will be topic of the next section.

3.2.3 Efficient Solution of the Fully Discretized PIDE

In (3.27), we need to solve a linear system of equations, $\tilde{A}x = b$, in each time step. In general, the matrix $\tilde{A}$ is not sparse because of the integral part in the stiffness matrix, and,

as a consequence of the space-dependent volatility function, it is not Toeplitz and furthermore not symmetric. At first sight, using an iterative method is not practicable, since matrix-vector multiplications with $\tilde{A}$ seem to be very expensive. But given the splitting of the stiffness matrix $A(T)$ (cf. (3.23)) a matrix vector product can be realized in $\mathcal{O}(n_x \log_2 n_x)$:

$$\tilde{A}x = \big(M + \Delta T\theta A(T_{k+1})\big)x = \underbrace{M\ x}_{\mathcal{O}(n_x)} + \Delta t\theta\Big(\underbrace{A^{NI}(T_{k+1})\ x}_{\mathcal{O}(n_x)} \quad + \underbrace{A^I\ x}_{\mathcal{O}(n_x \log_2 n_x)}\Big).$$

Beside the references mentioned above, where the integral part is mostly treated explicitly, Almendral and Oosterlee (2005) and Toivanen (2008) also work with fully implicit schemes. They both use splitting techniques, e.g. the Jacobi-method or a tridiagonal splitting to solve the systems iteratively.

In contrast, Sachs and Strauss (2008) use the conjugate gradient method to solve their dense systems. However, this is not applicable here because the matrix $\tilde{A}$ is not symmetric. The main reason is the generally non-symmetric density function $f(y)$. But even in the special case of the Merton model, when $\mu_J = 0$ and therefore $f(y)$ is symmetric, our Dupire-like PIDE involves an additional factor e^y destroying the symmetry.

For the solution of non-symmetric linear systems of equations Saad and Schultz (1986) proposed the generalized minimum residual (GMRES) algorithm. Like the conjugate gradient method for symmetric linear systems of equations, GMRES is an iterative Krylov subspace method and only needs matrix-vector products $\tilde{A}x$. For details, we refer to the books of Kelley (1995) and Saad (2003).

We will show some numerical results for the Merton jump-diffusion model using the following model constants and parameters:

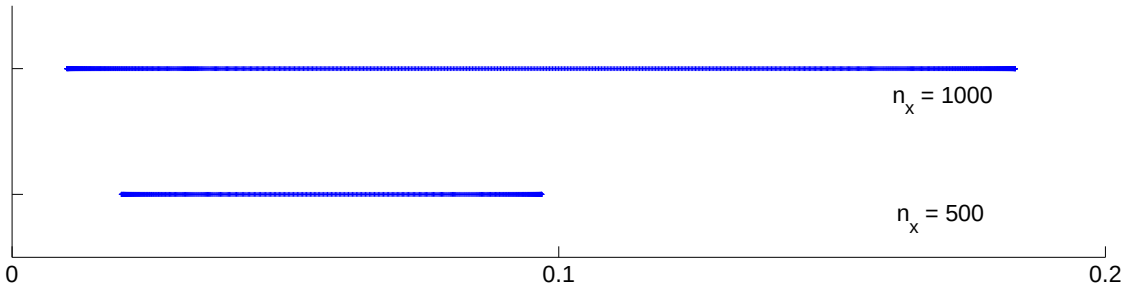
$$\begin{gathered}\underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 2\ y,\ r \equiv 3\%,\ \Delta T = 0.02, \\ \sigma \equiv 30\%,\ \lambda = 100\%,\ \mu_J = 0\%,\ \sigma_J = 50\%.\end{gathered} \tag{3.29}$$

The condition number and the eigenvalue distribution of the system play an important role with regard to the convergence speed. The second column of table 3.1 shows the condition number of the matrix $\tilde{A}$ for the setting above. $\kappa_2(\tilde{A})$ grows quadratically to the inverse of Δx, i.e. when the space step is halved, then the condition is quadruplicated. Figure 3.1(a)

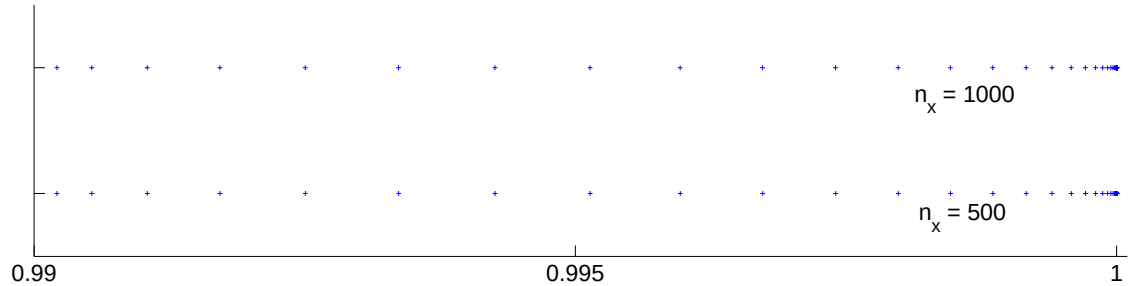
Δx	$\kappa_2(\tilde{A})$	$\kappa_2(P^{-1}\tilde{A})$
0.04	1.47	1.01
0.02	4.85	1.01
0.01	18.37	1.01
0.005	72.39	1.01
0.0025	288.42	1.01

Table 3.1: Condition number of the unpreconditioned ($\kappa_2(\tilde{A})$) and preconditioned system ($\kappa_2(P^{-1}\tilde{A})$) for different step sizes Δx

shows the eigenvalue distribution of $\tilde{A}$ for the cases $n_x = 500$ and $n_x = 1000$ (i.e. $\Delta x = 0.02$ and $\Delta x = 0.01$, resp.). Since $\tilde{A}$ in general is not symmetric, complex eigenvalues may occur.



(a) Unpreconditioned matrix $\tilde{A}$ (top: $n_x = 1000$, bottom: $n_x = 500$)



(b) Preconditioned matrix $P^{-1}\tilde{A}$ (top: $n_x = 1000$, bottom: $n_x = 500$)

Figure 3.1: Eigenvalue distribution of the preconditioned and unpreconditioned matrix $P^{-1}\tilde{A}$ and $\tilde{A}$, resp., for different Δx

In the examples shown here, the complex part is insignificantly small ($< 10^{-13}$), thus, it is omitted. We can see that the eigenvalues are distributed uniformly over an interval in the unpreconditioned case. Increasing the number of discretization points, n_x, leads to an expansion of the interval and a reduction of the distance to the origin.

Having made these observations, it is clear that a preconditioner is needed. There are two requirements for a preconditioner P. First, it has to approximate $\tilde{A}$ in some sense, e.g. to get $||I - P^{-1}\tilde{A}|| < \rho < 1$ for a left preconditioner. And second, the linear system $Px = c$ should be easy to solve. A matrix fulfilling these properties is $P := M + \Delta T \theta A^{NI}(T_{k+1})$. As mentioned above M and $A^{NI}(T)$ are tridiagonal matrices, thus, linear systems can be solved in $\mathcal{O}(n_x)$. It further is a good approximation to $\tilde{A}$ as we can see in the numerical results. Table 3.1 shows in the third column the condition number of the preconditioned system $P^{-1}\tilde{A}$. Note that it is close to one and seems to be mesh-independent. Regarding the eigenvalues in figure 3.1(b), one makes the observation that they are clustering around one and that their location is not influenced by a mesh refinement. Notice here the different scalings in figure 3.1(a) and 3.1(b).

Beside these technical observations, the efficiency of the preconditioner is illustrated by the numerical results in the next section.

There is one more thing to mention at this point, showing that it seems to be a good idea to use an iterative method. Since we solve a parabolic problem, the solution in time step T_k will not differ much from the solution in the last time step T_{k-1}. So this solution can be taken as an excellent initial guess for the new linear system of equations.

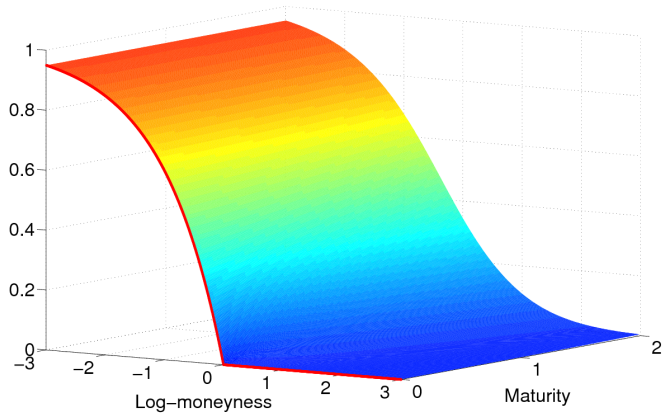


Figure 3.2: FE solution of the Merton model (highlighted in red: non-smooth initial condition)

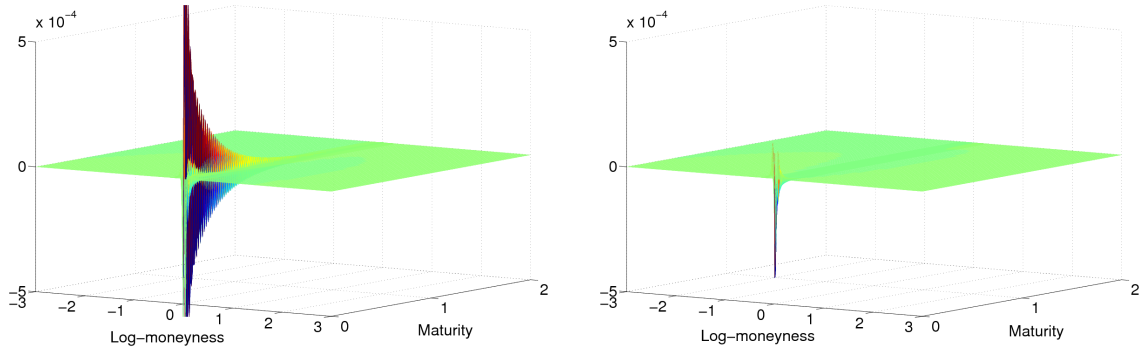
3.2.4 Numerical Results

This section presents some numerical results on the solution of the PIDE. For the convenience of the reader, let us first sum up the numerical solution that has been presented in the previous section. It can be divided into three parts:

a) Space discretization: finite element method with equidistantly distributed basis functions.
b) Time discretization: Crank-Nicolson method with Rannacher smoothing (the first Crank-Nicolson step is replaced by four implicit Euler quarter steps).
c) The dense linear systems of equations $\tilde{A}x = b$ in each time step are solved with the GMRES method; the sparse part of $\tilde{A}$ is used as a preconditioner.

Figure 3.2 shows a typical solution of the PIDE (3.1), where the parameter setting is the same as specified in (3.29) in the last section with $n_x = 2000$ and $n_T = 200$. At $T = 0$, the non-smooth initial condition $D_0(x) = \max\{1 - e^x, 0\}$ is highlighted in red. Since we consider a parabolic problem, the non-differentiable point at $x = 0$ is smoothed out in time. In the parameter setting we use a constant volatility $\sigma(\cdot, \cdot) \equiv 30\%$, so we are able to compare the numerical results of the Merton model with the closed-form solution introduced in remark 2.1.9 in this particular case.

Figure 3.3(a) shows the corresponding error of the finite element solution with an ordinary Crank-Nicolson time discretization. As expected, there are oscillations due to the non-smooth initial condition. If we replace the first Crank-Nicolson step by four implicit Euler quarter steps, i.e. applying the Rannacher smoothing, the result illustrated in figure 3.3(b) produce a far better approximation, especially at $x = 0$, a region of great importance in practical applications. Note that the additional computational effort is negligible. Tables 3.2 and 3.3 support the observations of figure 3.3. They show the error between a numerical solution with Crank-Nicolson and Rannacher time-stepping, respectively, and the closed-form solution for Merton's model. The L^∞- and L^2-error is evaluated at the time instances



(a) Error of Crank-Nicolson method **(b) Error with Rannacher smoothing (four quarter steps)**

Figure 3.3: Error between FE solution and closed-form solution for the Merton model ($\Delta x = 0.005$, $\Delta T = 0.01$)

discretization		$L^\infty(\Omega)$-error				$L^2(\Omega)$-error			
Δx	ΔT	$T=1$	ratio	$T=2$	ratio	$T=1$	ratio	$T=2$	ratio
0.00125	0.08	2.20e-3		1.51e-3		1.73e-4		1.06e-4	
	0.04	1.01e-3	2.2	6.51e-4	2.3	6.26e-5	2.8	3.73e-5	2.8
	0.02	4.13e-4	2.4	2.50e-4	2.6	2.21e-5	2.8	1.31e-5	2.8
	0.01	1.41e-4	2.9	8.45e-5	3.0	7.63e-6	2.9	4.24e-6	3.1

Table 3.2: $L^2(\Omega)$- and $L^\infty(\Omega)$-error (for $T = 1$ and $T = 2$) between finite element solution with Crank-Nicolson and closed-form solution for different time step sizes ΔT and fixed Δx

$T = 1$ and $T = 2$ on the spatial domain $\Omega = [-3, 3]$. The columns captioned by 'ratio' show the factor of decrease in the error when the number of discretization steps in time is doubled. It is observable that the Crank-Nicolson method does not show a quadratic convergence what would be indicated by a ratio of four. However, this ratio can be found in table 3.3 for the Rannacher smoothing.

Since we want to know whether the method proposed above is competitive, the Midpoint-122-scheme (cf. Briani et al. (2007)) has been implemented and table 3.4 contains the according results. The spatial step size Δx is kept fixed at a fine level to achieve that the errors are mainly caused by the time discretization. The first two columns show the step sizes in space and in time. The next four columns focus on different errors. Again, we set $\Omega = [-3, 3]$ and take a look at the $L^2(\Omega)$- and the $L^\infty(\Omega)$-error at the time instances $T = 1$ and $T = 2$. The last column shows the computational time required for solving one PIDE. The first thing to mention is the divergence for large time steps ΔT, because the CFL condition $\Delta T = \mathcal{O}(\Delta x^{4/3})$ is violated. If the step size is refined, the method converges up to sufficiently small errors. Table 3.5 is designed in the same way, but now the Crank-Nicolson method with Rannacher smoothing is used instead of the IMEX-scheme. Note that the error tolerance of the GMRES method is set to 10^{-8}. As expected, the numerical results show

discretization		$L^\infty(\Omega)$-error				$L^2(\Omega)$-error			
Δx	ΔT	$T=1$	ratio	$T=2$	ratio	$T=1$	ratio	$T=2$	ratio
0.00125	0.08	2.39e-5		1.11e-5		2.48e-5		1.37e-5	
	0.04	6.41e-6	3.7	2.77e-6	4.0	6.41e-6	3.9	3.42e-6	4.0
	0.02	1.63e-6	3.9	6.79e-7	4.1	1.61e-6	4.0	8.58e-7	4.0
	0.01	4.24e-7	3.8	1.59e-7	4.3	4.14e-7	3.9	2.21e-7	3.9

Table 3.3: $L^2(\Omega)$- and $L^\infty(\Omega)$-error (for $T=1$ and $T=2$) between finite element solution with Rannacher smoothing and closed-form solution for different time step sizes ΔT and fixed Δx

discretization		$L^\infty(\Omega)$-error		$L^2(\Omega)$-error		effort
Δx	ΔT	$T=1$	$T=2$	$T=1$	$T=2$	Time (sec.)
0.0025	0.02	2.05e+1	2.94e+6	1.95e+0	3.53e+5	0.26
	0.01	6.27e-2	3.04e+1	6.95e-3	4.41e+0	0.48
	0.005	1.27e-4	1.94e-4	1.62e-5	2.99e-5	0.95
	0.0025	7.44e-8	1.89e-7	6.10e-8	2.22e-7	1.86

Table 3.4: Computing times and $L^2(\Omega)$- and $L^\infty(\Omega)$-error (for $T=1$ and $T=2$) between finite element solution with Midpoint-122 rule and closed-form solution for different time step sizes ΔT and fixed Δx

the quadratic convergence $\mathcal{O}(\Delta T^2)$ in time.

The next to last column shows the computational time and we observe that the method is competitive compared to the Midpoint-122-scheme and, in contrast to the IMEX-scheme, there is convergence even for a coarse time grid.

The last column of the table shows the average number of GMRES iterations per time step. The algorithm needs about four iterations on a coarse time grid. When the time grid is refined, the initial guess of the GMRES method gets better, and the number of iterations can be reduced to an average of three although the linear systems of equations in each time step have size 4000×4000 for $\Delta x = 0.0025$.

After having studied the time discretization, we now turn to the spatial variable. Table 3.6 shows the corresponding results. Now, the step size in time, ΔT, is kept fixed at a fine

discretization		$L^\infty(\Omega)$-error		$L^2(\Omega)$-error		effort	
Δx	ΔT	$T=1$	$T=2$	$T=1$	$T=2$	Time(sec.)	Øiter
0.0025	0.04	6.47e-6	2.70e-6	6.46e-6	3.43e-6	0.48	4.2
	0.02	1.69e-6	6.35e-7	1.66e-6	8.89e-7	0.87	3.8
	0.01	4.88e-7	2.37e-7	4.62e-7	3.08e-7	1.61	3.0
	0.005	1.86e-7	1.89e-7	1.69e-7	2.25e-7	3.16	3.0

Table 3.5: Computing times, Ø-GMRES iterations per time step and $L^2(\Omega)$- and $L^\infty(\Omega)$-error (for $T=1$ and $T=2$) between finite element solution with Rannacher smoothing and closed-form solution for different time step sizes ΔT and fixed Δx

discretization		$L^\infty(\Omega)$-error		$L^2(\Omega)$-error		effort	
Δx	ΔT	$T=1$	$T=2$	$T=1$	$T=2$	Time(sec.)	Øiter
0.04	0.005	2.18e-5	4.60e-5	2.05e-5	5.46e-5	0.38	2.3
0.02		5.54e-6	1.18e-5	5.23e-6	1.38e-5	0.55	2.7
0.01		1.46e-6	2.99e-6	1.37e-6	3.46e-6	0.88	3.0
0.005		4.42e-7	7.53e-7	4.03e-7	8.70e-7	1.52	3.0
0.0025		1.86e-7	1.89e-7	1.69e-7	2.25e-7	3.16	3.0

Table 3.6: Computing times, Ø-GMRES iterations per time step and $L^2(\Omega)$- and $L^\infty(\Omega)$-error (for $T=1$ and $T=2$) between finite element with Rannacher smoothing and closed-form solution for different spatial step sizes Δx and fixed ΔT

level and Δx is varied. Taking a look at the errors in columns three to six, the quadratic convergence in space, $\mathcal{O}(\Delta x^2)$, is observable. The number of average GMRES iterations in the last column imply the mesh-independence of the preconditioned linear systems resulting in a linear growth of the computing times for refined meshes.

Hence, after the numerical solution of the PIDE has been derived and the numerical results have shown the efficiency of the implementation, we now study the optimization problem.

3.3 Solving the Optimization Problem

The proper calibration of the parameters occuring in the PIDE discussed in the last section is of great importance in practice. We recall the optimization problem under a PIDE constraint introduced in definition 2.2.1:

$$\begin{aligned}
&\min_{\tilde{D},\sigma,\lambda,f} \; J(\tilde{D},\sigma,\lambda,f) := \frac{1}{2}\sum_{i=1}^{M}\left(\tilde{D}(T_i,K_i) - D_i^M\right)^2 \qquad (3.30)\\
&s.t. \quad \tilde{D}_T - \frac{1}{2}\sigma^2(T,K)K^2\tilde{D}_{KK} + (r(T)-\lambda\zeta)K\tilde{D}_K + \lambda(1+\zeta)\tilde{D}\\
&\qquad\qquad - \lambda\int_{-\infty}^{+\infty}\tilde{D}(T,Ke^{-y})e^y f(y)\,dy = 0,\\
&\qquad\qquad (T,K)\in[0,T_{max})\times(0,\infty)\\
&\qquad \tilde{D}(0,K) = \max\{S_0-K,0\}, \quad K\in(0,\infty).
\end{aligned}$$

The problem is a so-called optimal control problem. There is a vast literature on this topic – for a basic introduction we refer to, e.g., Lions (1971), Hinze et al. (2009) or Tröltzsch (2010) – and there are also several ways to handle the problem from a numerical point of view. In the context of option pricing models, we mention Düring et al. (2008), where sequential quadratic programming is used to solve the constrained optimization problem. A different way of handling it is to transform it into an unconstrained problem, what can be done because the PIDE constraint is uniquely solvable. This unconstrained problem can then be minimized by a gradient-based method. Achdou and Pironneau (2005) provide an

introduction to this topic and further references.

In this thesis, we follow the latter approach, but before we solve the calibration problem numerically, we rewrite it in a more abstract way, in which the PIDE is replaced by its weak formulation according to remark 3.1.12. For instance, the spaces L^2 and H_0^1 are replaced by general Hilbert spaces H and V and so on, such that the following results may also be applied to other optimal control problems with parabolic constraints. Further, this has some advantages in terms of a more simple notation. For this purpose, we first define some operators.

We denote by V and H two real, separable Hilbert spaces with $V \hookrightarrow H = H^* \hookrightarrow V^*$, and by $\mathcal{U}$, a suitable closed, convex subset of a Hilbert space U, the space of control variables. For a fixed but arbitrary control $u \in \mathcal{U}$ and time $t \in [0,T]$, $A(u;t) \in \mathcal{L}(V,V^*)$ is a time- and control-dependent elliptic operator, which is Fréchet-differentiable with respect to the control variable u. The Fréchet derivative is denoted by $A'(u;t)$. Analog, we write for the right-hand side of the equation $l(u;t) \in V^*$ for all $u \in \mathcal{U}$ and $t \in [0,T]$ with corresponding Fréchet derivative $l'(u;t)$.

Market data $d_i \in \mathcal{H}$ are available at certain maturities $\hat{t}_i$, $i = 1,\dots,D$, where $\mathcal{H}$ is a Hilbert space with $\mathcal{H}^* = \mathcal{H}$. For instance, $\mathcal{H} = \mathbb{R}^5$ if we have data available for five strike prices at maturity $\hat{t}_i$. We assume $\hat{t}_i < \hat{t}_j$ for $i < j$ and define $\hat{t}_0 = 0$ and $\hat{t}_D = T$. $C \in \mathcal{L}(H,\mathcal{H})$ denotes the observation operator.

Given this setting we can define an abstract optimal control problem.

Definition 3.3.1. ***(Constrained optimization problem)***
For given market data d_i at $\hat{t}_i$ ($i = 1,\dots,D$), find solutions $y \in W([0,T],V)$ and $u \in \mathcal{U}$, which satisfy

$$\begin{aligned} \min_{y\in W, u\in\mathcal{U}} \; & J(y,u) := \frac{1}{2}\sum_{i=1}^{D} ||Cy(\hat{t}_i) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2}||u||_U^2 \\ s.t. \quad & \dot{y}(t) + A(u;t)y(t) - l(u;t) = 0 \quad , \quad t \in (0,T] \\ & y(0) = y_0. \end{aligned} \tag{3.31}$$

Although this is not the main topic of this thesis, we add a simple regularization term to the objective function in the problem formulation above, just to make clear that such a term – in this form or probably a more complicated one – usually can not be neglected in applications.

We briefly want to discuss how the problem (3.30) fits in this abstract setting. As it has been already discussed in section 3.1, the PIDE in its weak formulation can be transformed as desired (cf. remark 3.1.12). Setting the function spaces $H = L^2(\underline{x},\overline{x})$ and $V = H_0^1(\underline{x},\overline{x})$, then the pointwise observations in (3.30) turn out to be a challenge as C is assumed to be in $\mathcal{L}(L^2,\mathcal{H})$. To stress that our problem (3.30) can, however, be written in this abstract form above, we need to take care of this issue.

Remark 3.3.2. ***(Pointwise observations)***
In (3.30), the objective function involves pointwise observations. Assuming $\mathcal{H} = \mathbb{R}^1$ – i.e. market data is given for only one strike price $\hat{x}$ at maturity $\hat{t}_i$ –, then the observation operator C would include a Dirac delta function, which is known to be not L^2-integrable. To avoid the

involvement of distribution theory, we address the numerical approximation of C already at this stage and interpret C as an L^2-approximation, $\delta_{\hat{x}}^{\Delta x}$, of the Dirac delta function. Scott (1973) proposed and analyzed a concrete representation that is equivalent to $\delta_{\hat{x}}$ for a finite element space H^{n_x} in the sense that $\langle \delta_{\hat{x}}^{\Delta x}, v \rangle_{L^2} = \delta_{\hat{x}}(v)$, $v \in H^{n_x}$. We have in this special case

$$Cv = \langle \delta_{\hat{x}}^{\Delta x}, v \rangle_{L^2}, \quad v \in L^2,$$
$$C^* z = \delta_{\hat{x}}^{\Delta x} z, \quad z \in \mathbb{R}.$$

Note that – although it is not indicated in the following – C depends on Δx in this case.

Since the bilinear form $A(u;t)$ is assumed to be coercive and continuous, it is clear that for every $u \in \mathcal{U}$, the parabolic constraint admits a unique solution $y(u;\cdot) \in W([0,T],V)$. Hence, the problem specified in definition 3.3.1 can be written as an unconstrained optimization problem, in literature also known as 'reduced problem' [1].

Remark 3.3.3. ***(Unconstrained optimization problem)***
The problem specified in definition 3.3.1 can be written as

$$\min_{u \in \mathcal{U}} f(u) := J(y(u), u). \tag{3.32}$$

We describe the problem as unconstrained since the PIDE constraint is only involved implicitly. But to be precise, the space of controls $\mathcal{U}$ may be bounded, e.g. if the volatility function is bounded away from zero or has bounded derivatives in our application. Thus, we would again have a constrained optimization problem. Especially in a parameterized finite-dimensional space of controls, those constraints are usually given by box constraints.

3.3.1 First Discretize, then Optimize or vice versa?

Considering the discretization of the optimal control problem, there are mainly two approaches common in literature: 'Optimize-then-discretize' or 'discretize-then-optimize'.

In remark 3.3.2, we have explained that pointwise observations occurring in the calibration problem lead to Dirac delta functions, which we would like to avoid. So we first discretize the spatial variable. However, the further calculations in this section include only operators that might be interpreted either as discretized matrices or infinite-dimensional operators.

It remains the question whether to discretize in time or to optimize first. We briefly discuss both approaches and show some numerical results subsequently.

First Optimize

In order to derive a gradient representation, $\nabla f(u)$, for the unconstrained problem, which is needed in an optimization algorithm, we first define the Lagrange function for problem (3.31).

[1] Not to be confused with the reduced order models described later in this thesis.

Given appropriate Lagrange multipliers p^i, $i = 1, \ldots, D$, we define

$$\mathcal{L}(y, u, p^1, \ldots, p^D) := J(y, u) + \sum_{i=1}^{D} \int_{\hat{t}_{i-1}}^{\hat{t}_i} \langle p^i(t), \dot{y}(t) + A(u;t)y(t) - l(u;t) \rangle_{V,V^*} dt \tag{3.33}$$

and are now able to derive heuristically the corresponding optimality conditions. They consist of the state equation

$$\begin{aligned} &\dot{y}(t) + A(u;t)y(t) - l(u;t) = 0 \ , \qquad t \in (0, T] \\ &y(0) = y_0, \end{aligned} \tag{3.34}$$

where the initial condition $y(0) = y_0$ is given explicitly since we have not introduced an additional Lagrange multiplier in (3.33).

For $i = D$, the adjoint equation can be specified as

$$\dot{p}^D(t) - A^*(u;t)p^D(t) = 0 \ , \qquad t \in [\hat{t}_{D-1}, T) \tag{3.35}$$

$$p^D(T) = -C^*(Cy(T) - d_D) \tag{3.36}$$

and for $i = 1, \ldots, D-1$

$$\dot{p}^i(t) - A^*(u;t)p^i(t) = 0 \ , \qquad t \in [\hat{t}_{i-1}, \hat{t}_i) \tag{3.37}$$

$$p^i(\hat{t}_i) = -C^*(Cy(\hat{t}_i) - d_i) + p^{i+1}(\hat{t}_i) \tag{3.38}$$

Note that in (3.38) $p^{i+1}(\hat{t}_i)$ is known since we start solving the adjoint equations backwards at $t = T$, i.e. we first solve equation (3.35) backwards with end condition (3.36). It can be shown easily that $p^i \in W([\hat{t}_{i-1}, \hat{t}_i], V)$, $i = 1, \ldots, D$.

The last partial derivative of the Lagrange function with respect to the control u along a feasible direction δu leads to:

$$\alpha \langle u, \delta u \rangle_U + \sum_{i=1}^{D} \int_{\hat{t}_{i-1}}^{\hat{t}_i} \langle p^i(t), A'(u;t)\delta u \ y(t) - l'(u;t)\delta u \ \rangle_{V,V^*} dt \geq 0 \tag{3.39}$$

To show formally that (3.39) is the gradient of the unconstrained optimization problem, we introduce the sensitivity $z(t) = \frac{\partial y(u;t)}{\partial u}\delta u$. It holds the following result:

Lemma 3.3.4. ***(Sensitivity equation)***
Given a fixed but arbitrary $u \in \mathcal{U}$, the corresponding solution $y(u;\cdot) \in W([0,T], V)$ and a feasible direction δu. Further let z be the unique solution of

$$\begin{aligned} &\dot{z}(t) + A(u;t)z(t) + A'(u;t)\delta u \ y(u;t) - l'(u;t)\delta u = 0 \ , \ t \in (0, T] \\ &z(0) = 0. \end{aligned} \tag{3.40}$$

Then $z \in W([0,T], V)$ is the Fréchet derivative of $y(u;\cdot)$ with respect to u along direction δu.

Proof. Existence and uniqueness of a solution to (3.40) is guaranteed since A' and l' are

bounded linear operators. Setting $w(t) := y(u+\delta u;t) - y(u;t) - z(t)$, the following equation holds:

$$\begin{aligned}
&\dot{w}(t) + A(u;t)w(t) + \big(A(u+\delta u;t) - A(u;t) - A'(u;t)\delta u\big)y(u+\delta u;t) \\
&\quad + A'(u;t)\delta u\big(y(u+\delta u;t) - y(u;t)\big) - \big(l(u+\delta u;t) - l(u;t) - l'(u;t)\delta u\big) = 0 \\
&w(0) = 0.
\end{aligned}$$

Using the Fréchet-differentiability of A and l, it is easy to show that for every $\epsilon > 0$, there exists a finite $c > 0$ and $\Delta > 0$ with $||\delta u||_{\mathcal{U}} < \Delta$ such that $||w(t)||_H^2 \leq c\epsilon||\delta u||_{\mathcal{U}}^2$. Thus, the definition of a Fréchet derivative is fulfilled. □

Theorem 3.3.5. ***(Directional derivative of f(u))***
The derivative of $f(u)$ (defined in (3.32)) along a feasible direction δu is given by

$$f'(u)\delta u = \alpha\langle u, \delta u\rangle_U + \sum_{i=1}^{D}\int_{\hat{t}_{i-1}}^{\hat{t}_i} \langle p^i(t), A'(u;t)\delta u\ y(t) - l'(u;t)\delta u\ \rangle_{V,V^*}dt, \tag{3.41}$$

where y solves (3.34) and the p^i solve (3.35), (3.36) and (3.37), (3.38), respectively.

Proof. Differentiating $f(u)$ with respect to u in direction δu leads to

$$f'(u)\delta u = \alpha\langle u, \delta u\rangle_U + \sum_{i=1}^{D}\langle C^*(Cy(u;\hat{t}_i) - d_i), z(\hat{t}_i)\rangle_H.$$

For the second summand we get by using (3.36), (3.38) and $z(0) = 0$:

$$\begin{aligned}
&\sum_{i=1}^{D}\langle C^*(Cy(u;\hat{t}_i) - d_i), z(\hat{t}_i)\rangle_H = \\
&\quad = \langle -p^D(\hat{t}_D), z(\hat{t}_D)\rangle_H + \sum_{i=1}^{D-1}\langle -p^i(\hat{t}_i) + p^{i+1}(\hat{t}_i), z(\hat{t}_i)\rangle_H + \langle p^1(0), z(0)\rangle_H \\
&\quad = -\sum_{i=1}^{D}\Big(\langle p^i(\hat{t}_i), z(\hat{t}_i)\rangle_H - \langle p^i(\hat{t}_{i-1}), z(\hat{t}_{i-1})\rangle_H\Big).
\end{aligned} \tag{3.42}$$

Because $p^i \in W([\hat{t}_{i-1}, \hat{t}_i], V)$ $(i = 1, \ldots, D)$ integration by parts can be applied to every summand in (3.42). If further (3.35), (3.37) and (3.40) are used, we get for $i = 1, \ldots, D$:

$$\begin{aligned}
&\langle p^i(\hat{t}_i), z(\hat{t}_i)\rangle_H - \langle p^i(\hat{t}_{i-1}), z(\hat{t}_{i-1})\rangle_H = \int_{\hat{t}_{i-1}}^{\hat{t}_i}\Big(\langle z(t), \dot{p}^i(t)\rangle_{V,V^*} + \langle p^i(t), \dot{z}(t)\rangle_{V,V^*}\Big)dt \\
&= \int_{\hat{t}_{i-1}}^{\hat{t}_i}\Big(\langle z(t), A^*(u;t)p^i(t)\rangle_{V,V^*} + \langle p^i(t), \dot{z}(t)\rangle_{V,V^*}\Big)dt \\
&= \int_{\hat{t}_{i-1}}^{\hat{t}_i}\langle p^i(t), A(u;t)z(t) + \dot{z}(t)\rangle_{V,V^*}dt = -\int_{\hat{t}_{i-1}}^{\hat{t}_i}\langle p^i(t), A'(u;t)\delta u\ y(u;t) - l'(u;t)\delta u\rangle_{V,V^*}dt,
\end{aligned}$$

what directly shows the proposition. □

In order to solve the optimization problem numerically, the next step is the discretization. Section 3.2 describes the solution of the state equation in detail. This procedure can be applied analogously to the adjoint equations (3.35), (3.36) and (3.37),(3.38), respectively.

So we might assume that we know approximate solutions of the state and adjoint equations at certain time steps. For this let $\Delta t = T/n_t$ be the step size of a time discretization and $t_k = k\ \Delta t$ $(k = 0, \dots, n_t)$ the corresponding grid points. We denote by $y_k \approx y(t_k)$ and $p_k^i \approx p^i(t_k)$ (of course p^i and p_k^i only exist on $[\hat{t}_{i-1}, \hat{t}_i]$). For simplicity, we set $\{\hat{t}_i\}_{i=0}^D \subset \{t_i\}_{i=0}^{n_t}$, i.e. we assume that the time instances where market data is available are a subset of the grid, and there exist subindices k_i such that $\hat{t}_i = t_{k_i}$.

Definition 3.3.6. ***(Derivative approximation (FO))***
For given weights ω_k^i $(k = k_{i-1}, \dots, k_i,\ i = 1, \dots, D)$ we define the gradient approximation

$$f'_{FO}(u)\delta u = \Delta t \sum_{i=1}^{D} \sum_{k=k_{i-1}}^{k_i} \omega_k^i \langle p_k^i, A'(u;t_k)\delta u\ y_k - l'(u,t_k)\delta u \rangle_{V,V^*} + \alpha \langle u, \delta u \rangle_{\mathcal{U}}. \tag{3.43}$$

Remark 3.3.7. *Note that the weights ω_k^i determine the numerical integration rule. We here mention, e.g., the 'composite trapezoidal rule', where $\omega_{k_{i-1}}^i = \omega_{k_i}^i = 0.5$ and $\omega_k^i = 1$ for all other k.*

For any given gradient approximation in the form of (3.43), the approximation error can be divided into an integral approximation error depending on the method chosen to set the weights ω_k^i, and the error of the state solution, y_k, and the errors of the adjoint variables, p_k^i, compared to the continuous solutions, $y(t_k)$, $p^i(t_k)$, respectively.

First Discretize

We now want to discretize first and use the θ-scheme for the time discretization. For this, we define a time grid $t_k = k\ \Delta t$ $(k = 0, \dots, n_t)$ with $\hat{t}_i = t_{k_i}$ as above. As an abbreviation, we set for the finite difference quotient

$$\bar{\partial} y_i := \frac{y_i - y_{i-1}}{\Delta t}. \tag{3.44}$$

Definition 3.3.8. *For given market data d_i at $\hat{t}_i$ $(i = 1, \dots, D)$, find $\{y_k\}_{k=0}^{n_t} \subset V$ and $u \in \mathcal{U}$, solving the optimization problem*

$$\min_{\{y_k\}_{k=0}^{n_t} \subset V, u \in \mathcal{U}} \tilde{J}(y,u) := \frac{1}{2} \sum_{i=1}^{D} ||Cy_{k_i} - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2} ||u||_U^2 \tag{3.45}$$

$$\begin{aligned} s.t. \quad & \bar{\partial} y_{k+1} + \theta A(u;t_{k+1}) y_{k+1} + (1-\theta) A(u;t_k) y_k \\ & - \theta l(u;t_{k+1}) - (1-\theta) l(u;t_k) = 0 \quad , \quad k = 0, \dots, n_t - 1, \\ & y_0 = \hat{y}. \end{aligned} \tag{3.46}$$

A reduced cost function $f_{FD}(u) = \tilde{J}(y(u), u)$ can be defined as in the previous section.

Together with the corresponding adjoint equation,

$$\begin{aligned} &- \bar{\partial} p_{k+1} + \theta A^*(u, t_k) p_k + (1-\theta) A^*(u, t_k) p_{k+1} \\ &\quad + \sum_{i=1}^{D-1} C^*(C y_{k_i} - d_i) \mathbb{1}_{k=k_i} = 0 \quad , \quad k = n_t - 1, \ldots, 1 \\ &p_{n_t} + \Delta t \theta A^*(u, t_{n_t}) p_{n_t} = -\Delta t C^*(C y_{n_t} - d_D), \end{aligned} \tag{3.47}$$

which again can be derived via the Lagrangian approach, a gradient representation for the discrete reduced problem can be verified via a discrete sensitivity equation.

Theorem 3.3.9. ***(Derivative approximation (FD))***
The derivative of $f_{FD}(u)$ (as defined above) along a feasible direction δu is given by

$$\begin{aligned} f'_{FD}(u)\delta u = \sum_{k=0}^{n_t-1} &\langle p_{k+1}, \theta A'(u; t_{k+1})\delta u \; y_{k+1} + (1-\theta) A'(u; t_k)\delta u \; y_k \\ &- \theta l'(u; t_{k+1})\delta u - (1-\theta) l'(u; t_k)\delta u \rangle_{V, V^*} dt + \alpha \langle u, \delta u \rangle_{\mathcal{U}}, \end{aligned} \tag{3.48}$$

where y solves (3.46) and the p solves (3.47).

We discuss the difference between both approaches by means of numerical results in section 3.3.3.

3.3.2 Optimization Methods

Regardless of whether we discretize or optimize first, we solve the optimization problem,

$$\min_u f(u) \; ,$$

by a gradient-based method. The gradient of the problem can be calculated efficiently by means of the adjoint equation (3.41). However, second-order information is more complicated and the calculation of the exact Hessian is usually not reasonable.

Quasi-Newton methods use gradient information of the iterations that have already been carried out to approximate the Hessian matrix. Update formulas as 'symmetric-rank-one' or 'BFGS' and some further convergence results can be found in, e.g., Nocedal and Wright (1999).

Beside these methods, the special structure of our optimal control problem with its least-squares formulation as in (3.31) meets the requirements for the application of a 'Gauß-Newton' approach. The objective function f can be written as

$$f(u) := \tfrac{1}{2} ||R(u)||^2 \qquad \text{with} \quad R : \mathbb{R}^{n_p} \to \mathbb{R}^{n_m},$$

where $R_j(u)$, $j = 1, \ldots, n_m$, is the difference between market value j and the corresponding

model value. Defining by $J : \mathbb{R}^{n_m \times n_p} \to \mathbb{R}^{n_m}$ the Jacobian of R, we get

$$\nabla f(u) = J(u)^T R(u)$$
$$\nabla^2 f(u) = J(u)^T J(u) + \sum_{j=1}^{n_m} R_j(u)\ \nabla^2 R_j(u) \approx J(u)^T J(u)$$

because $R_j(u)$ is hopefully a very small value.

Assuming that the functions, which are to be calibrated, are parameterized, i.e. the number of parameters, n_p, is small, the Gauß-Newton method often yields a faster convergence. Regarding calibration problems in finance, a detailed comparison between quasi-Newton and Gauß-Newton method can, e.g., be found in Lörx (2012). In the numerical results below, we will compare the Gauß-Newton and quasi-Newton method applied to the calibration of a jump-diffusion model.

3.3.3 Numerical Results

Adjoint Equation and Gradients

In this section we discuss the numerical calculation of the gradient of our problem. Here, the focus is on numerical challenges in the time discretization arising in the solution of the adjoint equation. The numerical results presented below are based on the Merton model (cf. example 2.1.8) and the following setting:

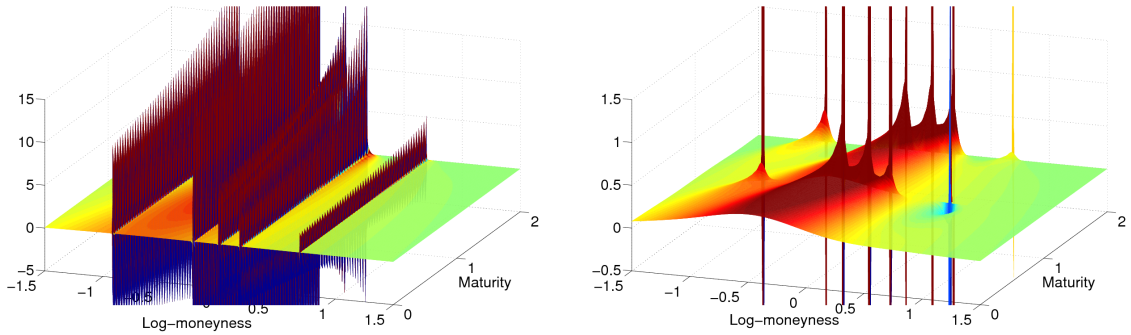
$$\begin{gathered} \underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 2y,\ r \equiv 3\%,\ \alpha = 0. \\ \text{Market data given at: } (T_i, K_i) \in \{\{1,2\} \times \{40\%, 80\%, 100\%, 120\%, 200\%\}\}. \\ \text{Four parameters for Merton's model: } u = (\lambda, \mu_J, \sigma_J, \sigma^2) \in \mathbb{R}^4. \end{gathered} \tag{3.49}$$

The market data call prices are produced with $\tilde{u} = (50\%, 0\%, 50\%, 30\%^2)$ and we choose as a sample parameter $u = (60\%, -80\%, 40\%, 30\%^2)$ to calculate gradients and adjoints.

We have already noticed that the pointwise observations in the objective function of the calibration problem (2.15) lead to high-frequency end conditions in the backward adjoint equations.

If we first optimize, we are free in the choice of an appropriate discretization method. As has been shown in section 3.2, a Crank-Nicolson method loses its quadratic convergence in case of non-smooth initial conditions. Regarding the adjoint equation, the problem is far more extreme. Given the setting above, (3.49), a standard Crank-Nicolson method applied to the adjoint equation (3.35), (3.36) and (3.37), (3.38) leads to the result illustrated in Figure 3.4(a) ($\Delta x = 0.0025$, $\Delta T = 0.02$). According to the notation of section 3.3.1, the adjoint is formally divided into two parts, p^1, p^2, with end conditions at $T = 1$ and $T = 2$, where market data is available. The peaks occurring in these end conditions are not smoothed out, but oscillate strongly over the whole time domain. Analog to the state equation, Rannacher smoothing can be applied to the adjoint at each end condition. The corresponding figure 3.4(b) now shows functions p^1, p^2 that are smooth in time. Please note the different scaling of figure 3.4(a) and 3.4(b), causing a cut of the peaks at $T = 1$ and $T = 2$ in (b).

We now turn to the first discretize approach, where we use a Rannacher time stepping



(a) Crank-Nicolson **(b) Rannacher time stepping**

Figure 3.4: First optimize: solutions of the adjoint equation ($\Delta T = 0.02$, $\Delta x = 0.0025$)

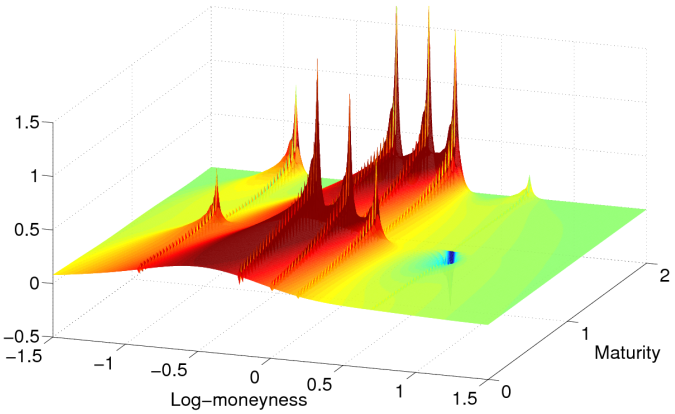


Figure 3.5: First discretize: solution of the adjoint equation ($\Delta T = 0.02$, $\Delta x = 0.0025$)

scheme for the state equation as proposed in section 3.3.1. Figure 3.5 shows the numerical solution of the corresponding adjoint in the sense of (3.47). Note here that the Rannacher method in the state equation yields a Crank-Nicolson method for the the adjoint except for the last time step before $T = 0$, where four implicit Euler quarter steps are used. Two points are remarkable here. First we note that the peaks at $T = 1$ and $T = 2$ are not as pronounced as in the first optimize approach. This is due to the fact that the end condition,

$$p_{n_t} + \Delta t \theta A^*(u, t_{n_t}) p_{n_t} = -\Delta t C^*(C y_{n_t} - d_D),$$

contains a kind of built-in smoothing through the elliptic operator weighted with step size Δt. Hence, the greater the step size, i.e. the more instable the Crank-Nicolson scheme for non-smooth end conditions, the more pronounced is the smoothing. However, - and this is the second point - there are still small oscillations observable through the whole time domain, avoiding a quadratic convergence of the scheme.

discretization		$L^2(\Omega)$-error at $T = 0$					
Δx	ΔT	FO (Rann.)	ratio	FO (C.-N.)	ratio	FD	ratio
0.0025	0.04	4.69e-5		1.61e+0		2.55e-3	
	0.02	1.19e-5	3.9	1.05e+0	1.5	1.29e-3	2.0
	0.01	2.99e-6	4.0	3.91e-1	2.7	6.47e-4	2.0
	0.005	7.47e-7	4.0	1.22e-1	3.2	3.24e-4	2.0
	0.0025	1.85e-7	4.0	1.21e-1	1.0	1.62e-4	2.0

Table 3.7: $L^2(\Omega)$-error at $T = 0$ of the adjoint solution for the three approaches in figure 3.4 and 3.5, resp., for different time step sizes ΔT and fixed Δx (Reference solution calculated with FO (Rann.) and ΔT = 3.125e-4, $\Delta x = 0.0025$)

This is shown in table 3.7, where we compare the numerical solution of the adjoint equation at the last time instance $T = 0$ with a reference solution calculated with first optimize Rannacher on a very fine time grid (ΔT = 3.125e-4, $\Delta x = 0.0025$). This is done for the three methods shown in figure 3.4 and 3.5, and for different step sizes ΔT (Δx fixed).

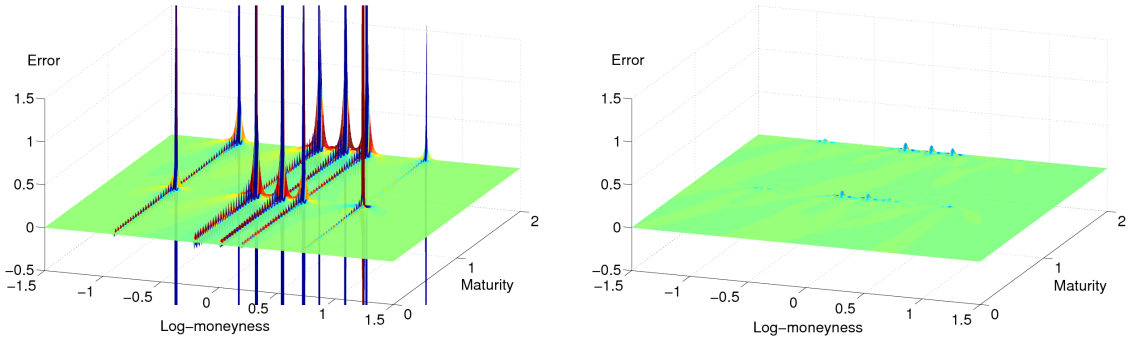
The term 'ratio' again shows the factor of decrease in the $L^2(\Omega)$-error when the number of discretization steps in time is doubled ($\Omega = [-3, 3]$). In the last column of the table, the ratio implies only a linear convergence with respect to the step size ΔT. However, the first optimize approach using Crank-Nicolson shows nearly no improvement of the error for a refined time grid. As expected, the Rannacher smoothing steps in the first optimize approach preserve the quadratic convergence, where the difference in the order of magnitude of the error compared to the first discretize approach is significant.

In addition, the error results of table 3.7 are visualized in figure 3.6. We omit the result for first optimizing with Crank-Nicolson since this is not competitive. Again we calculate the reference solution for the adjoint on a fine grid (ΔT = 3.125e-4, $\Delta x = 0.0025$). Figure 3.6(a) then shows the pointwise error of the adjoint on a coarse time grid ($\Delta T = 0.02$, $\Delta x = 0.0025$), where we first discretize the state equation with Rannacher time stepping and the adjoint equation is then solved by a Crank-Nicolson scheme with four implicit Euler quarter steps in the last time step. The oscillations that are visible in figure 3.5 are now observable more clearly. However, figure 3.6(b) shows the error when we first optimize and then use the Rannacher smoothing for the adjoint, which is remarkably smaller.

We further want to point out the effect of the four implicit quarter steps when we discretize first. Figure 3.7 shows the error at time $T = 0.02$, i.e. the last time step before the implicit steps are applied. Where the first discretize approach (continuous line) shows strong oscillations, the first optimize approach (dotted line) is quite smooth. But the oscillations are then smoothed out in the the last time step, observable in 3.7(b).

This observation also motivates a different approach that can be found in Goll et al. (2011). They propose to change the discretization scheme for the state equation in such way that the resulting scheme for the adjoint equation in the first discretize approach automatically leads to a stabilized version. To be precise, one would discretize the state equation by Rannacher time stepping and additionally apply four implicit Euler quarter time steps before (!) each time step where market data is available.

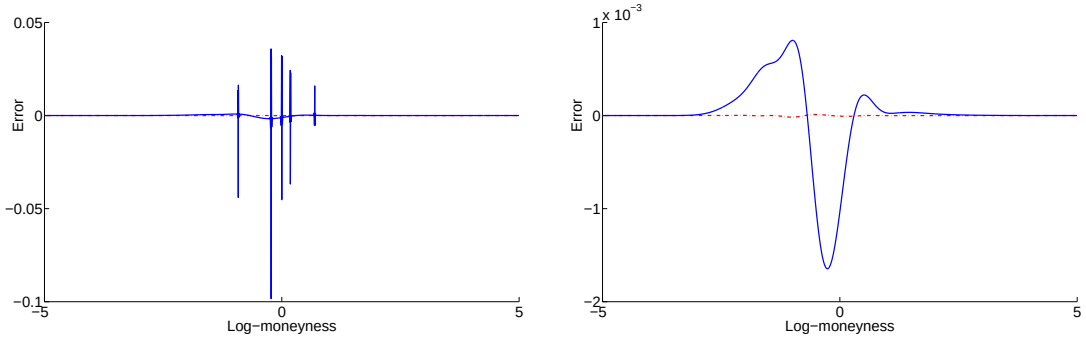
Lastly, we are of course interested in the gradient of our problem. Given the parameter vector $u \in \mathbb{R}^4$ for the Merton model, a reference gradient, ∇f_{ref}, is calculated on a fine grid



(a) Error adjoint: first discretize (Rannacher), then optimize

(b) Error adjoint: first optimize, then discretize (Rannacher time stepping)

Figure 3.6: Difference between reference adjoint and the adjoints on coarser grids for first discretize (a) and first optimize (b), resp.



(a) Error at $T = 0.02$

(b) Error at $T = 0$

Figure 3.7: Difference between reference adjoint and the adjoints on coarser grids for first discretize and first optimize (dotted line), resp., at time $T = 0$ and at time $T = 0.02$

discretization		$\|\|\nabla f_{ref} - \nabla f_{disc}\|\|_2 / \|\|\nabla f_{ref}\|\|_2$					
Δx	ΔT	FO (Rann.)	ratio	FO (C.-N.)	ratio	FD	ratio
0.0025	0.04	3.50e-5		6.53e-1		6.72e-4	
	0.02	8.48e-6	4.1	1.96e-1	3.3	1.83e-3	0.4
	0.01	2.12e-6	4.0	2.78e-1	0.7	3.23e-4	5.7
	0.005	5.27e-7	4.0	2.95e-1	0.9	1.22e-5	26.5
	0.0025	1.38e-7	3.8	2.94e-1	1.0	5.43e-7	22.5

Table 3.8: Relative gradient errors for the three approaches of figure 3.4 and 3.5, resp., and different time step sizes ΔT and fixed Δx with control $u \in \mathbb{R}^4$

strike K	maturity T									
	.175	.425	.7	.95	1.0	1.5	2.0	3.0	4.0	5.0
85%	.190	.177	.172	.171	.171	.169	.169	.168	.168	.168
90%	.168	.155	.157	.159	.159	.160	.161	.161	.162	.164
95%	.133	.138	.144	.149	.150	.151	.153	.155	.157	.159
100%	.113	.125	.133	.137	.138	.142	.145	.149	.152	.154
105%	.102	.109	.118	.127	.128	.133	.137	.143	.148	.151
110%	.097	.103	.104	.113	.115	.124	.130	.137	.143	.148
115%	.120	.100	.100	.106	.107	.119	.126	.133	.139	.144
120%	.142	.114	.101	.103	.103	.113	.119	.128	.135	.140
130%	.169	.130	.108	.100	.099	.107	.115	.124	.130	.136
140%	.200	.150	.124	.110	.108	.102	.111	.123	.128	.132

Table 3.9: Implied volatility table with interest rate $r = 5\%$ and no dividends (a slightly modified test example on S&P 500 options according to Andersen and Brotherton-Ratcliffe (1998))

via the first optimize approach with Rannacher smoothing (ΔT = 3.125e-4, $\Delta x = 0.0025$). Note that the relative difference between the reference gradient based on first optimize-Rannacher and first discretize-Rannacher is 7.47e-009, i.e it is negligible. Table 3.8 shows the relative errors between this reference gradient and the gradient for the three approaches on several coarser time grids. It is observable that, especially for very coarse time steps ΔT, the first optimize approach with Rannacher smoothing is by far the best one. However, discretizing first also leads to acceptable results, especially for finer grids. In fortunate circumstances, the oscillations that are observable in the adjoint equation seem to sum up to zero.

Calibration Example

We now turn to a concrete calibration problem. We use the MATLAB solver 'fmincon' for its solution and provide gradient and second-order information either via a Gauß-Newton or a quasi-Newton approach, respectively, as it has been proposed in section 3.3.2. Let us first specify the example that is to be solved.

We have given market data in terms of implied volatilities for ten different strike prices and ten different maturities on S&P 500 options. We follow an academic example presented in Andersen and Brotherton-Ratcliffe (1998) that is only slightly modified. The implied volatility table is shown in table 3.9. In addition to this table 3.9, the corresponding surface is illustrated in figure 3.8. It is observable that this example provides the typical smile curve known from many empirical studies especially for the short-term options.

We now want to calibrate the Merton model with three jump parameters, λ, μ_J, σ_J, and a constant or parameterized volatility, respectively. To be precise, we do not calibrate the volatility but the squared volatility because the bilinear form defined in (3.2) only involves the squared function or its derivative, respectively. This also means that we parameterize the squared volatility function further below.

For completeness, we specify the remaining constants of the discretization that have been

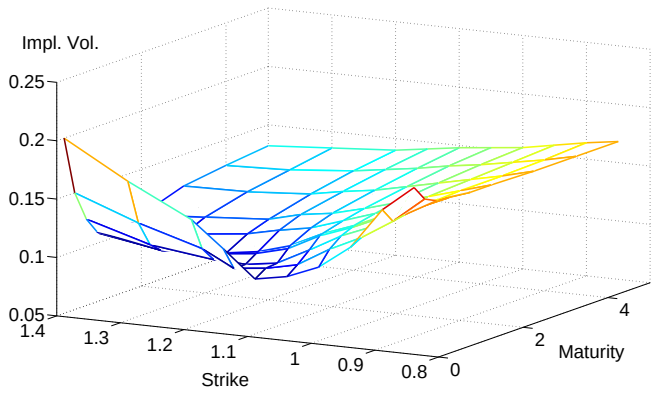


Figure 3.8: Visualization of the implied volatility surface of table 3.9

set as follows:

$$\begin{gathered} \underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 5\ y,\ r \equiv 5\%,\ \Delta T = 0.0125,\ \Delta x = 0.0025, \\ \text{starting vector: } u = (\lambda, \mu_J, \sigma_J, \sigma^2) = (40\%, 0\%, 40\%, 40\%^2). \end{gathered} \tag{3.50}$$

Due to the ill-posedness of the problem, we further introduce a regularization term where we penalize the difference between the squared volatility and a predefined mean squared volatility σ^2_{mid}. Thus, we replace the simple regularization term $\frac{\alpha}{2}||u||^2$ that has been added to the objective function previously by $\frac{\alpha}{2}||\sigma^2 - \sigma^2_{mid}||^2$, where $\sigma_{mid} = 13\%$ is chosen as an average of the implied volatilities between the strikes 95% and 120%. The weighting factor α is set to 0.01.

Table 3.10 shows the corresponding results for both algorithms where the volatility function is assumed to be constant. Stopping at a comparable error level, both algorithms need approximately the same computing time. Although the number of iterations is higher for the quasi-Newton method, one iteration is cheaper since the gradient is calculated via adjoints which is a computational effort of about two and a half PIDEs (state plus adjoint plus the summation in the gradient). On the other hand, one Gauß-Newton iteration requires the computation of five PIDEs but also provides second order information.

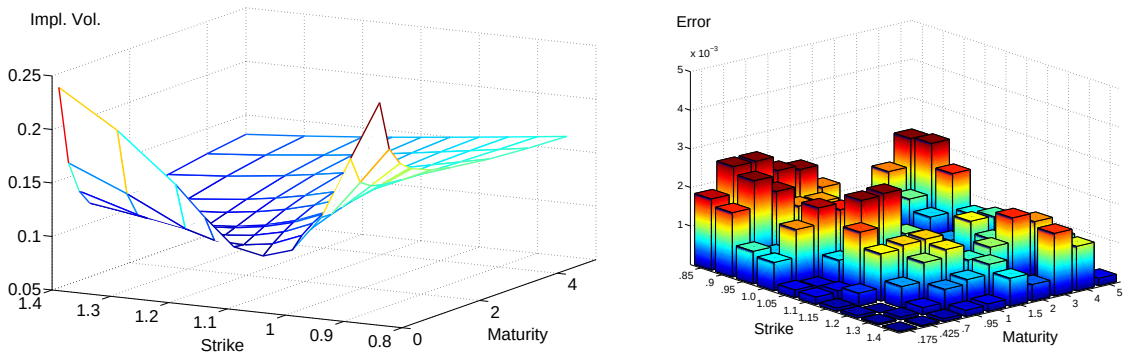
Since we calibrate four parameters to fit 100 market prices, the problem is clearly underdetermined. However, taking a look at the implied volatilities corresponding to the model prices in the optimal point of the Gauß-Newton algorithm in figure 3.9(a), we observe that the smile is already approximated quite well. This effect of jump-diffusion models has been discussed earlier in section 2.1.2 and is due to a negative mean jump size. For instance, the approximate solution using the Gauß-Newton approach yields an optimal control $u_{opt} = (\lambda, \mu_J, \sigma_J, \sigma^2) = (15.9\%, -22.1\%, 21.0\%, 8.9\%^2)$.

But, on the other hand, the error between market and model prices in 3.9(b) is still too big.

To further reduce this error, we use a parameterized volatility function. For this, the squared volatility is parameterized using linear splines in space with five degrees of freedom

algorithm	time (sec.)	#iter	#PIDE-eval	f_{opt}	$\|\|\nabla f_{opt}\|\|_2$
Gauß-Newton	573	39	201	6.07e-5	1.46e-4
quasi-Newton	554	54	184	6.35e-5	1.84e-4

Table 3.10: Computing times, number of iterations, function evaluations and gradient evaluations for the quasi-Newton and Gauß-Newton method; optimization with constant volatility (four parameters)



(a) Implied volatility surface at the optimal solution

(b) Error between market data and model data at the optimal solution

Figure 3.9: Optimal solution of the Gauß-Newton method for Merton's model with constant volatility (four parameters)

(grid points at 90%, 100%, 110%, 120%, 130%), and parameterized with four piecewise constant parts in time (grid points: 0y, 0.425y, 0.95y, 3y, 5y). On the left-hand side of the leftmost grid point of the spatial parameterization (90%), the squared volatility is assumed to be constant. The same holds true for the rightmost grid point (130%). Thus, instead of one constant volatility, we now have 20 degrees of freedom. Together with jump insensity λ, mean jump size μ_J and volatility of the jump size σ_J, we have 23 parameters in total.

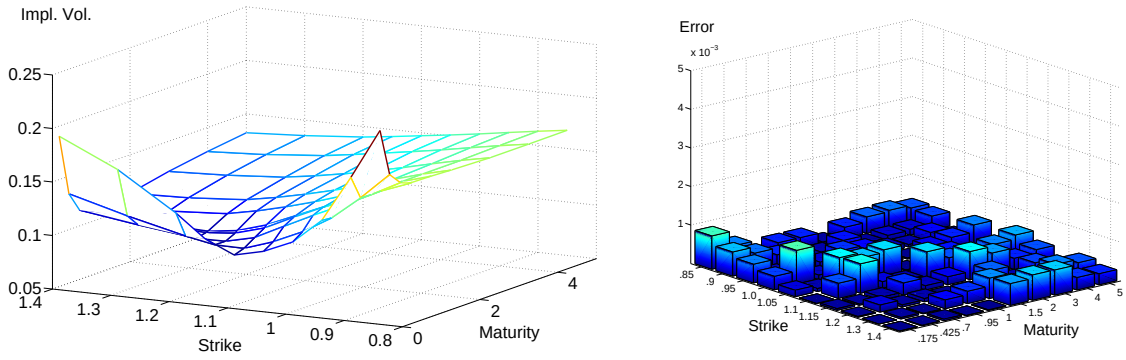
Let us first take a look at the consequences regarding the computational effort. Table 3.11 shows the corresponding results. In both algorithms, the quasi-Newton and the Gauß-Newton approach, the time increases. But it is clearly observable that the quasi-Newton algorithm is less sensitive to the number of parameters than the Gauß-Newton. This is due to the fact that one gradient evaluation grows linearly with the number of parameters in the latter case, but it does not increase significantly when using the adjoint approach.

Taking again a look at the optimal values of the Gauß-Newton algorithm, we see that the implied volatility surface in its optimal point is quite similar to the target surface in figure 3.8. And even more clearly, the error between market and model prices has decreased.

Thus, we have learned that the quasi-Newton approach is the method of choice when we have many parameters to calibrate. A Gauß-newton method providing second-order information would be more reasonable if the function evaluations – this means the solution

algorithm	time (sec.)	#iter	#PIDE-eval	f_{opt}	$\|\|\nabla f_{opt}\|\|_2$
Gauß-Newton	2983	43	1057	1.06e-5	1.08e-4
quasi-Newton	888	82	274	9.62e-6	8.29e-4

Table 3.11: Computing times, number of iterations, function evaluations and gradient evaluations for the quasi-Newton and Gauß-Newton method; optimization with local volatility (23 parameters)



(a) Implied volatility surface at the optimal solution

(b) Error between market data and model data at the optimal solution

Figure 3.10: Optimal solution of the Gauß-Newton method for Merton's model with local volatility (23 parameters)

of a PIDE – are less expensive. To reduce the cost of one PIDE evaluation is now the objective of the next chapter where we introduce reduced order models.

Chapter 4

Model Order Reduction via POD

The general idea of model order reduction (MOR) is to replace a large mathematical problem – in our case a discretized partial differential equation – by a small one. The error between the original model and the reduced model should be small, however, the computational effort is supposed to be reduced significantly.

The so-called 'reduced basis' method (cf. Grepl (2005) or Grepl and Patera (2005) and the references cited therein) is based on the observation that solutions of parameter-dependent differential equations are not arbitrary functions of the solution space. Usually they are all contained in a lower-dimensional subspace. In a finite element approach (cf. chapter 3) the space of basis functions is an approximation of the whole solution space (e.g. the space $H_0^1(\Omega)$). The reduced basis approach proposes to use basis functions that only span the aforementioned lower-dimensional subspace which is related to the characteristics of the problem.

Antoulas et al. (2001) or Antoulas (2005) provide a survey of several model reduction techniques for linear dynamical systems in state space form. The most famous ones are 'balanced truncation' and 'proper orthogonal decomposition' (POD). Both methods have a close connection to 'singular value decomposition' (SVD). Balanced truncation is only applicable to linear time-invariant systems and therefore cannot be used for the option pricing problem analyzed in the previous chapter. However, POD is even used for nonlinear systems and will be our method of choice.

To be precise, POD is used to find basis functions approximating an ensemble of observations in a certain optimal sense. So, when we say 'Model Order Reduction via POD' in the title of the chapter, we mean that we use proper orthogonal decomposition to extract certain basis functions from a given set of information. These basis functions are then used in a Galerkin approach as a reduced basis in the above-mentioned sense.

Hence, one of the main tasks in the construction of a reduced order model is to get appropriate information concerning our problem in terms of functions or vectors – we will call them 'snapshots' in the following –, where POD can be applied to. These can either be determined from experimental data or – and this will be done here – from numerical experiments.

In Kahlbacher and Volkwein (2007, 2012), POD is used to derive a low-order model of a parameter-dependent elliptic problem. Here, one needs to calculate high-order solutions for several parameters and these solutions are then used as snapshots. In time-dependent differential equations, the time variable is treated as a parameter. For instance, Kunisch and Volkwein (1999) study POD applied to the unsteady Burgers equation and in Kunisch and Volkwein (2001) general parabolic problems are addressed.

There is a vast literature on further applications, where POD is used. Lumley (1967), Sirovich (1987) and Holmes et al. (1997) can be named as the early references for POD with application in fluid flow, coherent structures and turbulences,. It is also used in signal processing, data compression and pattern recognition (cf. Holmes et al. (1996)).

The application of reduced order models in financial applications, e.g. the solution of partial integro-differential equations resulting from jump-diffusion option pricing models, is a quite new issue first described in Sachs and Schu (2008) where POD is used, and by Pironneau (2009) using a reduced basis approach with basis functions based on Black-Scholes solutions. It has been further investigated in Sachs and Schu (2010) and Cont et al. (2011).

Since usually global, control-independent a priori error estimates are lacking, the use of POD in optimal control is a quite difficult task. However, it has already been done in several articles. Afanasiev and Hinze (2001) build a POD model based on a certain control and then calculate a suboptimal reduced control. For the suboptimal control, new snapshots are computed and added to the former ones to get a better POD basis. In Ravindran (2002), a POD model is updated in each SQP iteration with application to an optimal control of the Navier-Stokes equation. The so-called optimality system-POD (OS-POD) has been introduced by Kunisch and Volkwein (2008). They include the optimality of the POD basis in the optimality system and split the optimization procedure in optimizing the control under a reduced model and optimizing the POD model under a given control.

In this chapter, the adjoint equation of the optimal control problem turns out to be of great importance. Its influence on POD has already been investigated in Hinze and Volkwein (2008) and Tröltzsch and Volkwein (2009) where error estimates for the suboptimal control are presented in terms of the POD error for the state and for the adjoint equation. However, the POD basis is not updated during the optimization procedure but only extended.

This chapter is now organized as follows. Section 4.1 presents an introduction to proper orthogonal decomposition in general. It is shown how we can extract significant information – that is then stored in a POD basis – from a given set of information. Here, eigenvalue problems play an important role and the POD basis consists of certain eigenfunctions. The corresponding projection errors are given in terms of a sum over those eigenvalues whose eigenfunctions are not part of the POD basis.

In section 4.2, we are interested in further error estimates of POD reduced order models. First, POD is applied to an abstract parabolic differential equation with time-dependent bilinear form in section 4.2.1, where the set of information is a given solution of the parabolic problem for a fixed control variable. Again, error estimates can be derived in terms of a sum over the remaining eigenvalues. We then turn to the application of POD in optimal control problems in section 4.2.2. We can use the results of section 4.2.1 to estimate the error between the 'true' objective function value and the objective function value based on a POD model. However, to make statements on the corresponding gradient error, we have to include information of the adjoint equation. 4.3 shows numerical experiments confirming the theoretical results.

Note that all theoretical results of this section only hold true for a fixed control variable. But numerical results show also the positive effect of one combined basis for state and adjoint equation when we veer away from the initial control. A globalization of the POD technique is then achieved through embedding into a trust-region method in chapter 5.

4.1 Proper Orthogonal Decomposition

According to Sirovich (1987), proper orthogonal decomposition has been mentioned first by Lumley (1967) in the context of turbulent flows. In other fields of research, e.g. statistics, it is also known as 'Karhunen-Loève transformation' or 'principal component analysis'. The idea is to build an orthonormal basis for a given set of information. This orthonormal basis must be optimal in the sense that there does not exist another basis of the same size that represents the set of information better. In this context, Sirovich (1987) has established the expression 'method of snapshots'.

There is a vast literature on POD. However, the following introduction is mainly based on Volkwein (2001).

Let H be a real, separable Hilbert space with inner product $\langle \cdot, \cdot \rangle_H$ and the induced norm $|| \cdot ||_H := \sqrt{\langle \cdot, \cdot \rangle_H}$. Further, let elements $u_i \in H$, $i = 1, \dots, n$, be given. The space spanned by these 'snapshots' has dimension $r > 0$, i.e. $\dim(\text{span}(u_1, \dots, u_n)) = r$. Thus, at least one snapshot is assumed to be nonzero. Proper orthogonal decomposition consists of finding elements $\Psi_j \in H$, $j = 1, \dots, r$, that build an orthonormal basis of $\text{span}(u_1, \dots, u_n)$ and have the following additional property: Considering the partial basis $\Psi_1, \dots, \Psi_l$ for an arbitrary $l \in \{1, \dots, r\}$, there are no other orthonormal basis functions $\Phi_1, \dots, \Phi_l$, which approximate an 'average' element of $\text{span}(u_1, \dots, u_n)$ in a better way.

The projection of a $v \in \text{span}(u_1, \dots, u_n)$ on the space spanned by arbitrary orthonormal functions $\{\Psi_j\}_{j=1}^l$ can be computed from its Fourier expansion:

$$\tilde{v} = \sum_{j=1}^{l} \langle v, \Psi_j \rangle_H \Psi_j.$$

Hence, the mathematical definition for the POD basis functions is formulated as follows.

Definition 4.1.1. ***(POD basis)***
Given snapshots $u_1, ..., u_n \in H$, find orthonormal functions $\Psi_1, \dots, \Psi_r \in \text{span}(u_1, \dots, u_n)$ by solving the minimization problem

$$\min_{\Psi_1, ..., \Psi_l} \sum_{i=1}^{n} \gamma_i \Big|\Big| u_i - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j \Big|\Big|_H^2 \tag{4.1}$$
$$s.t. \quad \langle \Psi_j, \Psi_k \rangle_H = \delta_{jk} \quad \forall j, k = 1, \dots, l$$

for all $l \in \{1, \dots, r\}$ with weights $\gamma_i > 0$, $i = 1, \dots, n$. The first l vectors $\Psi_1, ..., \Psi_l$ are called a POD basis of rank l. The spanning subspace is denoted by $\mathcal{V}^l = \text{span}(\Psi_1, ..., \Psi_l)$. Here, δ_{ij} denotes the Kronecker delta with $\delta_{ij} = 1$ for $i = j$, $\delta_{ij} = 0$, else.

Choosing the weights in (4.1) as $\gamma_i = \frac{1}{n}$ for all n yields the arithmetic average. However, the weights might be chosen differently depending on the particular problem.

Remark 4.1.2. *Note that the POD basis functions can also be defined in an infinite-dimensional setting, where – given a continuum of snapshots – the sum $\sum_{i=1}^n$ in (4.1) is replaced by an integral (cf. Kunisch and Volkwein (2002)). The finite sum above may then*

be interpreted as a numerical integration where the γ_i's gain importance as the corresponding weights of each function value.

In (4.1) the POD basis functions minimize the projection error, however, the minimization problem can also be written equivalently as a maximization problem. Here, the basis can be interpreted as those functions that capture most of the system energy.

Lemma 4.1.3. *Given snapshots $u_1, ..., u_n \in H$, find orthonormal vectors $\Psi_1, \dots, \Psi_r \in \operatorname{span}(u_1, \dots, u_n)$ by solving the minimization problem:*

$$\max_{\Psi_1,\dots,\Psi_l} \sum_{i=1}^{n} \gamma_i \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H^2 \tag{4.2}$$
$$s.t. \quad \langle \Psi_j, \Psi_k \rangle_H = \delta_{jk} \quad \forall j,k = 1, \dots, l$$

for all $l \in \{1, \dots, r\}$ with weights $\gamma_i > 0$, $i = 1, \dots, n$. Then, $\Psi_1, \dots, \Psi_r$ is a solution of (4.1) and vice versa.

Proof. For every summand of (4.1), we have

$$\begin{aligned}
&\Big\| u_i - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j \Big\|_H^2 = \Big\langle u_i - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j, u_i - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j \Big\rangle_H \\
&= \langle u_i, u_i \rangle_H - 2 \Big\langle u_i, \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j \Big\rangle_H + \Big\langle \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j, \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H \Psi_r \Big\rangle_H \\
&= \langle u_i, u_i \rangle_H - 2 \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \langle u_i, \Psi_j \rangle_H + \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H \underbrace{\langle \Psi_j, \Psi_k \rangle_H}_{\delta_{jk}} \\
&= \langle u_i, u_i \rangle_H - 2 \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H^2 + \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H^2 = \langle u_i, u_i \rangle_H - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H^2,
\end{aligned}$$

which immediately yields the proposition. □

Thus, the POD basis can be determined either via the solution of (4.1) or (4.2). The next task is to find optimality conditions for these optimization problems. For this, we need to define some operators and a weighted scalar product for vectors $v, w \in \mathbb{R}^n$:

$$\langle v, w \rangle_\gamma := \sum_{i=1}^{n} \gamma_i v_i w_i \ ,$$

where the $\gamma_i > 0$, $i = 1, \dots, n$, are the weights as in (4.1).

Definition 4.1.4. *Based on the snapshot set $u_1, \dots, u_n \in H$, the bounded linear operator $\mathcal{Y} \in \mathcal{L}(\mathbb{R}^n, H)$ is defined as*

$$\mathcal{Y} v := \sum_{i=1}^{n} \gamma_i v_i u_i \ , \quad v \in \mathbb{R}^n. \tag{4.3}$$

Remark 4.1.5. *Thus, the adjoint operator $\mathcal{Y}^* \in \mathcal{L}(H, \mathbb{R}^n)$ is given by*

$$\mathcal{Y}^* z = \Big(\langle z, u_1\rangle_H, \ldots, \langle z, u_n\rangle_H\Big)^T , \quad z \in H.$$

Proof. $\langle v, \mathcal{Y}^* z\rangle_\gamma = \sum_{i=1}^{n} \gamma_i v_i \langle z, u_i\rangle_H = \langle \sum_{i=1}^{n} \gamma_i v_i u_i, z\rangle_H = \langle \mathcal{Y} v, z\rangle_H$. □

Definition 4.1.6. *Based on the snapshot set $u_1, \ldots, u_n \in H$, the bounded linear operator $\mathcal{R} \in \mathcal{L}(H, H)$ is defined as*

$$\mathcal{R} z := \mathcal{Y}\mathcal{Y}^* z = \sum_{i=1}^{n} \gamma_i \langle z, u_i\rangle_H u_i , \quad z \in H. \tag{4.4}$$

It is easy to verify that the autocorrelation operator $\mathcal{R}$ has the following properties (cf. Volkwein (2001)):

Remark 4.1.7. *The operator $\mathcal{R} \in \mathcal{L}(H, H)$ as defined in (4.4) is compact, self-adjoint and non-negative.*

These properties allow us to apply the following theorem that can be found, e.g., in Reed and Simon (1980, p. 203) to the operator $\mathcal{R}$.

Theorem 4.1.8. ***(Hilbert-Schmidt theorem)***
Let $\mathcal{D} : H \to H$ be a bounded, self-adjoint, compact operator on a real, separable Hilbert space H. Then there exists a complete orthonormal basis $\{\Phi_k\}_{k=1}^{\infty}$ for H, such that

$$\mathcal{D}\Phi_k = \lambda_k \Phi_k \text{ with } \lambda_k \overset{k\to\infty}{\longrightarrow} 0.$$

It turns out that the application of theorem 4.1.8 to the operator $\mathcal{R}$ provides a way of determining the POD basis functions. Via the formulation of the Lagrange functional for the constrained optimization problem (4.2), the following optimality conditions can be derived.

Theorem 4.1.9. ***(Construction of the POD basis)***
Given the setting above, there exists a complete orthonormal basis $\{\Psi_k\}_{k=1}^{\infty}$ of H and corresponding non-negative numbers $\{\lambda_k\}_{k=1}^{\infty}$, such that

$$\mathcal{R}\Psi_k = \lambda_k \Psi_k \quad \text{with } \lambda_1 \geq \lambda_2 \geq \ldots \tag{4.5}$$

Then $\{\Psi_k\}_{k=1}^{l}$, $l \leq r$, is the solution to (4.1), i.e. the POD basis of rank l.

Proof. A detailed proof is provided in Volkwein (2001). □

Thus, to get the POD basis of rank l for a given set of snapshots $u_1, \ldots, u_n \in H$, we have to solve the eigenvalue problem (4.5), where the POD basis functions are identical to the eigenfunctions Ψ_k corresponding to the largest eigenvalues λ_k.

The size of the eigenvalue problem is equal to the dimension of H that might be infinite-dimensional or at least very high. In this case, we are able to define another operator of dimension n, which is significantly smaller in many applications.

Definition 4.1.10. *Based on the snapshot set $u_1, \dots, u_n \in H$, the bounded linear operator $\mathcal{K} \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$ is defined as*

$$\mathcal{K}z := \mathcal{Y}^*\mathcal{Y}v \quad , \quad v \in \mathbb{R}^n, \tag{4.6}$$

i.e. $\mathcal{K} \in \mathbb{R}^{n\times n}$ with $\mathcal{K}_{ij} = \gamma_i \langle u_j, u_i \rangle_H$.

Analog to remark 4.1.7, $\mathcal{K}$ has the following properties:

Remark 4.1.11. *The operator $\mathcal{K} \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$ as defined in (4.6) is compact, self-adjoint and non-negative with $rank(\mathcal{K}) = r$.*

Proof. Clearly, the matrix $\mathcal{K}$ is a weighted Gramian matrix. Thus, the properties above can be deduced easily. Just notice that due to the weights γ_i the matrix $\mathcal{K}$, in general, is non-symmetric, but it is self-adjoint with respect to the weighted scalar product $\langle \cdot, \cdot \rangle_\gamma$: $\langle \mathcal{K}w, v \rangle_\gamma = \langle \mathcal{Y}w, \mathcal{Y}v \rangle_H = \langle w, \mathcal{K}v \rangle_\gamma$. □

The connection between the operators $\mathcal{K}$ and $\mathcal{R}$ is pointed out in the next lemma.

Lemma 4.1.12. *Given the setting above, there exists a complete orthonormal basis [1] $\{v_k\}_{k=1}^n$ of $\mathbb{R}^n$ and corresponding non-negative numbers $\{\lambda_k\}_{k=1}^n$, such that*

$$\mathcal{K}v_k = \lambda_k v_k \quad \text{with } \lambda_1 \geq \ldots \geq \lambda_r > 0 \text{ and } \lambda_k = 0 \ \ \forall\ k = r+1, \dots, n. \tag{4.7}$$

Then all $\lambda_k > 0$, $k = 1, \dots, r$, are eigenvalues of $\mathcal{R}$, too, and the corresponding orthonormal eigenfunctions are given by

$$\Psi_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{Y} v_k \ , \quad k = 1, \dots, r. \tag{4.8}$$

Proof. The proof naturally falls into two parts. Firstly, using remark 4.1.11, the existence of $\{v_k\}_{k=1}^n$ and $\{\lambda_k\}_{k=1}^n$ satisfying (4.7) is evident (e.g. by applying theorem 4.1.8).

Secondly, for $k \leq r$, we apply the operator $\frac{1}{\sqrt{\lambda_k}}\mathcal{Y}$ to (4.7) and get $\mathcal{Y}\mathcal{Y}^* \frac{1}{\sqrt{\lambda_k}} \mathcal{Y} v_k = \lambda_k \frac{1}{\sqrt{\lambda_k}} \mathcal{Y} v_k$, i.e. (4.8). $\langle \Psi_k, \Psi_j \rangle_H = \frac{1}{\sqrt{\lambda_k \lambda_j}} \langle \mathcal{Y}^*\mathcal{Y} v_k, v_j \rangle_\gamma = \frac{\lambda_k}{\sqrt{\lambda_k \lambda_j}} \langle v_k, v_j \rangle_\gamma = \delta_{kj}$ gives the orthonormality of the Ψ_k's. □

So instead of solving the eigenvalue problem (4.5), which has the same dimension as H, we can now solve problem (4.7) with dimension n, typically much smaller than $\dim(H)$. Afterwards, one would apply $\frac{1}{\sqrt{\lambda_k}}\mathcal{Y}$ to the eigenvectors v_k to get the POD basis functions Ψ_k.

It is clear that, given a POD basis of rank $l < r$, the projection of an arbitrary element $u \in \text{span}(u_1, \dots, u_n)$ on the space spanned by $\{\Psi_k\}_{k=1}^l$ may lead to an approximation error. It turns out that this error strongly depends on the eigenvalues of (4.5) or their decay rate, respectively.

[1] Orthonormal with respect to the weighted scalar product $\langle \cdot, \cdot \rangle_\gamma$

l	1	2	3	10	20	100
$\mathcal{E}_l$	0.9428	0.9745	0.9795	0.9911	0.9952	0.9992

Table 4.1: Energy $\mathcal{E}_l$ for different l corresponding to the picture example (figure 1.2) in chapter 1

Corollary 4.1.13. ***(Truncation error)***
Given snapshots $u_1, ..., u_n \in H$, and the corresponding POD basis $\Psi_1, \dots, \Psi_r \in \operatorname{span}(u_1, \dots, u_n)$ that solves (4.1). Then, for $l \leq r$, the optimal function value is

$$\min_{\Psi_1,\dots,\Psi_l} \sum_{i=1}^{n} \gamma_i \Big\| u_i - \sum_{j=1}^{l} \langle u_i, \Psi_j \rangle_H \Psi_j \Big\|_H^2 = \sum_{k=l+1}^{r} \lambda_k. \tag{4.9}$$

Proof. First, the orthonormality of $\{\Psi_k\}_{k=1}^{l}$ yields

$$\begin{aligned}\lambda_k &= \lambda_k \langle \Psi_k, \Psi_k \rangle_H = \langle \lambda_k \Psi_k, \Psi_k \rangle_H = \langle \mathcal{R}\Psi_k, \Psi_k \rangle_H = \Big\langle \sum_{i=1}^{n} \gamma_i \langle \Psi_k, u_i \rangle_H u_i, \Psi_k \Big\rangle_H \\ &= \sum_{i=1}^{n} \gamma_i \langle \Psi_k, u_i \rangle_H \langle u_i, \Psi_k \rangle_H = \sum_{i=1}^{n} \gamma_i \langle \Psi_k, u_i \rangle_H^2.\end{aligned}$$

Using the ideas of the proof to lemma 4.1.3 and the basis property of $\{\Psi_k\}_{k=1}^{r}$, we get:

$$\begin{aligned}&\sum_{i=1}^{n} \gamma_i \Big\| u_i - \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H \Psi_k \Big\|_H^2 = \sum_{i=1}^{n} \gamma_i \Big(\langle u_i, u_i \rangle_H - \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H^2 \Big) \\ &= \sum_{i=1}^{n} \gamma_i \Big(\Big\langle \sum_{k=1}^{r} \langle u_i, \Psi_k \rangle_H \Psi_k, u_i \Big\rangle_H - \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H^2 \Big) = \sum_{i=1}^{n} \gamma_i \Big(\sum_{k=1}^{r} \langle u_i, \Psi_k \rangle_H^2 - \sum_{k=1}^{l} \langle u_i, \Psi_k \rangle_H^2 \Big) \\ &= \sum_{i=1}^{n} \sum_{k=l+1}^{r} \gamma_i \langle u_i, \Psi_k \rangle_H^2 = \sum_{k=l+1}^{r} \lambda_k,\end{aligned}$$

which establishes formula (4.9). □

In literature (cf. Sirovich (1987)) the sum $\sum_{k=1}^{r} \lambda_k$ is often referred to as 'energy' of the system (or of the set of snapshots, resp.). Thus, the quotient

$$\mathcal{E}_l := \frac{\sum_{k=1}^{l} \lambda_k}{\sum_{k=1}^{r} \lambda_k} \tag{4.10}$$

is of interest as a criterion to choose l such that the first l POD basis functions capture a specific percentage, e.g. $\mathcal{E}_l \geq 99\%$, of the total system energy.

Table 4.1 shows exemplarily the system energy for different numbers of POD basis functions for the motivating example in figure 1.2 in chapter 1.

However, in practice, the value corresponding to the denominator of (4.10), in general, is not available because the eigenvalue problems (4.7) are often not solved completely but only

iteratively for the largest eigenvalues.

4.2 POD Error Estimates

This section deals with error estimates of POD reduced systems in the context of parabolic differential equations and, furthermore, in optimal control problems governed by such PDEs.

4.2.1 A Priori Error Estimates for Parabolic Differential Equations

If we want to apply the POD technique outlined above to a parabolic differential equation – like the pricing PIDE of chapter 3 – we have to specify the set of snapshots. Here, we choose the solution of the problem at fixed time steps $t_0, \ldots, t_n$. We then obtain some orthonormal basis functions containing specific information about the solution of the PIDE in the above-mentioned sense. Approximating the PIDE problem via a POD approach then means that we replace the finite element basis functions by the POD basis functions calculated from the given solution. Since we only need a few basis functions – numerical tests show that 10 is already a sufficient quantity – compared to, e.g., 1000 finite element basis functions, the systems of equations that have to be solved in each time step are considerably smaller.

Since the original problem, the PIDE, is replaced by a smaller one, the POD approximation, we want to estimate the corresponding error. We make the following assumptions on the parabolic problem following Dautray and Lions (1992, pp. 509 ff):

Assumption 4.1. *a) Let V and H be two real, separable Hilbert spaces with the inner products $\langle\cdot,\cdot\rangle_V$ and $\langle\cdot,\cdot\rangle_H$ and the induced norms $||\cdot||_V$ and $||\cdot||_H$, respectively. With the dual spaces V^* and H^* they form a Gelfand triple:*

$$V \hookrightarrow H = H^* \hookrightarrow V^* \tag{4.11}$$

with dense embeddings. Furthermore, assume an $\alpha > 0$ with $||v||_H^2 \leq \alpha ||v||_V^2$ for all $v \in V$. b) Let $a : [0,T] \times (V \times V) \to \mathbb{R}$ for all $t \in [0,T]$ be a uniformly continuous and coercive bilinear form, i.e. there exist constants $\beta, \kappa > 0$ independent of t with

$$|a(t;\ v,w)| \leq \beta ||v||_V ||w||_V \qquad \forall\, v,w \in V\ \forall\, t \in [0,T], \tag{4.12}$$

$$a(t;\ v,v) \geq \kappa ||v||_V^2 \qquad \forall\, v \in V\ \forall\, t \in [0,T]. \tag{4.13}$$

In addition let $a(\cdot;\cdot,\cdot)$ be Lipschitz continuous with respect to t, i.e.

$$|a(t_1;\ v,w) - a(t_2;\ v,w)| \leq c_{lip}|t_1 - t_2|\ ||v||_V ||w||_V \qquad \forall\, v,w \in V. \tag{4.14}$$

c) Let $L :\ [0,T] \times V \to \mathbb{R}$ be a linear form with $L \in L^2(V^*)$ and $c_L > 0$ such that

$$|L(t;\ v)| \leq c_L ||v||_V \quad \forall\, t \in [0,T],\ v \in V. \tag{4.15}$$

With the notation fixed and the assumptions stated, we can formulate the weak form of an abstract parabolic initial value problem.

Definition 4.2.1. ***(Continuous problem)***
For given initial value $y_0 \in H$ find a solution $y \in W([0,T],V)$ which satisfies

$$\frac{d}{dt}\langle y(t),v\rangle_H + a(t;\ y(t),v) = L(t;\ v) \quad \forall\, v \in V,\ t \in (0,T) \tag{4.16}$$

and initial condition

$$\langle y(0),v\rangle_H = \langle y_0,v\rangle_H \quad \forall\, v \in V. \tag{4.17}$$

Again $\bar{\partial}$ is used as an abbreviation for the finite difference quotients as already defined in (3.44). Thus, problem (4.16) discretized in time on a subspace $\mathcal{V}^l$ of V with equidistant time steps $t_0,\ldots,t_m$ (i.e. $\Delta t = t_i - t_{i-1}\ \forall i = 1,\ldots,m$) looks as follows:

Definition 4.2.2. ***(Discretized problem)***
For given initial value $y_0 \in H$ and some $\theta \in [0,1]$ find $\{y_i^l\}_{i=0}^n \subset \mathcal{V}^l$ with

$$\begin{aligned}\langle \bar{\partial} y_i^l, v\rangle_H + \theta \cdot a(t_i;\ y_i^l, v) + (1-\theta)\cdot a(t_{i-1};\ y_{i-1}^l, v) = \\ \theta \cdot L(t_i;\ v) + (1-\theta)\cdot L(t_{i-1}, v) \qquad \forall\, v \in \mathcal{V}^l,\ i = 1,\ldots,n\end{aligned} \tag{4.18}$$

and initial condition

$$\langle y_0^l, v\rangle_H = \langle y_0, v\rangle_H \quad \forall\, v \in \mathcal{V}^l. \tag{4.19}$$

Using the stated assumptions, we can invoke an existence and uniqueness theorem in Dautray and Lions (1992, pp. 512 ff) to conclude that there exists a unique solution for both problems.

Since there are different possibilities to create a POD basis, we want to clarify which ones we use and which errors we address.

Error 1: Average error between the solution $y(t)$ of problem (4.16) and the solution on the POD subspace, discretized in time via the θ-method (this is problem (4.18)). The POD basis functions are calculated in the sense of (4.1) from the snapshots of the solution $y(t)$ and the corresponding difference quotients, i.e. the snapshots are $\bar{y}_i = y(t_{i-1})$, $i = 1,\ldots,n+1$ and $\bar{y}_{i+n+1} = \frac{y(t_i)-y(t_{i-1})}{\Delta t}$, $i = 1,\ldots,n$. To avoid confusion we call the POD solution $y^{l,1}$ and define

$$ERR_1 = \frac{1}{n}\sum_{i=1}^{n} ||y_i^{l,1} - y(t_i)||_H^2. \tag{4.20}$$

Error 2: Average error between the finite element approximation y^{FE} discretized in time and space (this is the solution of problem (4.18) in which we replace the POD space $\mathcal{V}^l$ by the finite element subspace H^{n_x}) and the POD approximation, $y^{l,2}$, discretized in time (problem (4.18)), where the POD basis functions are calculated in the sense of (4.1) from the snapshots of the finite element solution and the corresponding difference quotients. We

define

$$ERR_2 = \frac{1}{n}\sum_{i=1}^{n} ||y_i^{l,2} - y_i^{FE}||_H^2. \tag{4.21}$$

The fact that the difference quotients are included in the calculation of the POD basis is often reported to yield numerically better approximation results. Here, we note that it also facilitates the proof for the error estimates. We further notice that we restrict ourselves to the case where we use the weaker topology H in (4.1). The case using V instead of H is also studied in literature, cf. Kunisch and Volkwein (2001) and leads to slightly different estimates.

In the proof we use two different projections and in the next lemma we show some characteristic properties of these projections.

Definition 4.2.3. ***(Projection operators)***
Let $\mathcal{V}^l$ be a subspace of V. We define the H-projection Π_H^l

$$\Pi_H^l : V \to \mathcal{V}^l \quad \Leftrightarrow \quad \langle \Pi_H^l u - u, v\rangle_H = 0 \quad \forall\, v \in \mathcal{V}^l$$

and the Ritz-projection $\Pi_{a,t}^l$

$$\Pi_{a,t}^l : V \to \mathcal{V}^l \quad \Leftrightarrow \quad a(t;\ \Pi_{a,t}^l u - u, v) = 0 \quad \forall\, v \in \mathcal{V}^l, t \in [0,T].$$

We show a relationship of the Ritz-projection to the H-projection and the Lipschitz continuity for the Ritz-projection with respect to the time variable.

Lemma 4.2.4. *For the projections we have for $u \in V$*

$$||\Pi_{a,t}^l u - u||_V^2 \le \frac{\beta}{\kappa}||v-u||_V^2 \quad v \in \mathcal{V}^l, \quad \textit{in particular} \quad v = \Pi_H^l u \tag{4.22}$$

$$||(\Pi_{a,t}^l - \Pi_{a,s}^l)u||_V \le |t-s|\frac{c_{lip}}{\kappa}||\Pi_{a,s}^l u - u||_V. \tag{4.23}$$

Proof. Using (4.12) and (4.13) one easily verifies the following inequalities:

$$\kappa||\Pi_{a,t}^l u - u||_V^2 \le a(t;\ \Pi_{a,t}^l u - u, \Pi_{a,t}^l u - u) \le a(t;\ v-u, v-u) \le \beta||v-u||_V^2$$

for all $t \in [0,T]$ and $v \in \mathcal{V}^l$. The coercivity (4.13) yields

$$\kappa||(\Pi_{a,t}^l - \Pi_{a,s}^l)u||_V^2 \le a(t;\ (\Pi_{a,t}^l - \Pi_{a,s}^l)u, (\Pi_{a,t}^l - \Pi_{a,s}^l)u)$$

and using the Ritz-projection property as well as the Lipschitz continuity (4.14) we get

$$\begin{aligned} a(t;\ (\Pi_{a,t}^l - \Pi_{a,s}^l)u, (\Pi_{a,t}^l - \Pi_{a,s}^l)u) &= \\ &= a(s;\ \Pi_{a,s}^l u - u, (\Pi_{a,t}^l - \Pi_{a,s}^l)u) - a(t;\ \Pi_{a,s}^l u - u, (\Pi_{a,t}^l - \Pi_{a,s}^l)u) \\ &\le c_{lip}|t-s|\ ||\Pi_{a,s}^l u - u||_V\ ||(\Pi_{a,t}^l - \Pi_{a,s}^l)u||_V. \end{aligned}$$

Combining these two results yields the second statement. □

By assumption 4.1 we have $||v||_H^2 \leq \alpha ||v||_V^2$ for all $v \in V$. A reverse inequality holds if we consider the finite-dimensional subspace $\mathcal{V}^r$:

$$||u||_V \leq \sqrt{||S||_2}||u||_H \quad \forall u \in \mathcal{V}^r \quad \text{with} \quad S \in \mathbb{R}^{r\times r}, \quad S_{ij} = \langle \Psi_j, \Psi_i \rangle_V, \tag{4.24}$$

see e.g. Kunisch and Volkwein (2001, lemma 2).

For the POD error compared to the Ritz-projection we have the following error estimate. Note that we assume $y_0 \in V$ from now on.

Lemma 4.2.5. *For the implicit Euler method ($\theta = 1$) we have*

$$||y_i^{l,1} - \Pi_{a,t_i}^l y(t_i)||_H \leq ||y_{i-1}^{l,1} - \Pi_{a,t_{i-1}}^l y(t_{i-1})||_H + \Delta t ||v_i||_H \tag{4.25}$$

and for the Crank-Nicolson scheme ($\theta = 1/2$)

$$||y_i^{l,1} - \Pi_{a,t_i}^l y(t_i)||_H \leq (1 + 4\xi\Delta t^3)||y_{i-1}^{l,1} - \Pi_{a,t_{i-1}}^l y(t_{i-1})||_H + 2\Delta t ||v_i||_H \tag{4.26}$$

provided $\xi \Delta t^3 < 1/2$ with $\xi = c_{lip}^2 \alpha ||S||_2^2/(32\kappa)$. Here, v_i is defined as

$$v_i = \theta \cdot y_t(t_i) + (1-\theta)\cdot y_t(t_{i-1}) - \bar{\partial}\Pi_{a,t_i}^l y(t_i). \tag{4.27}$$

Proof. Set

$$w_i = y_i^{l,1} - \Pi_{a,t_i}^l y(t_i).$$

In the equalities below, for the first equation we use the definition of w_i, for the second equation recall (4.18) for the first part and definition 4.2.3 for the second part (note that $w_i \in \mathcal{V}^l$). Thus, we have for an arbitrary $\Psi \in \mathcal{V}^l$:

$$\begin{aligned}
&\langle \bar{\partial} w_i, \Psi \rangle_H + \theta \cdot a(t_i;\ w_i, \Psi) + (1-\theta)\cdot a(t_{i-1};\ w_{i-1}, \Psi) \\
&= \langle \bar{\partial} y_i^{l,1}, \Psi \rangle_H + \theta \cdot a(t_i;\ y_i^{l,1}, \Psi) + (1-\theta)\cdot a(t_{i-1};\ y_{i-1}^{l,1}, \Psi) \\
&\quad - \langle \bar{\partial}\Pi_{a,t_i}^l y(t_i), \Psi\rangle_H - \theta \cdot a(t_i;\ \Pi_{a,t_i}^l y(t_i), \Psi) - (1-\theta)\cdot a(t_{i-1};\ \Pi_{a,t_i}^l y(t_{i-1}), \Psi) \\
&= \theta \cdot L(t_i;\ \Psi) + (1-\theta)\cdot L(t_{i-1};\ \Psi) \\
&\quad - \langle \bar{\partial}\Pi_{a,t_i}^l y(t_i), \Psi\rangle_H - \theta \cdot a(t_i;\ y(t_i), \Psi) - (1-\theta)\cdot a(t_{i-1};\ y(t_{i-1}), \Psi).
\end{aligned}$$

Since $y(t)$ is the solution of (4.16) we obtain

$$\begin{aligned}
&\langle \bar{\partial} w_i, \Psi \rangle_H + \theta \cdot a(t_i;\ w_i, \Psi) + (1-\theta)\cdot a(t_{i-1};\ w_{i-1}, \Psi) \\
&= \theta \cdot \langle y_t(t_i), \Psi\rangle_H + (1-\theta)\cdot \langle y_t(t_{i-1}), \Psi\rangle_H - \langle \bar{\partial}\Pi_{a,t_i}^l y(t_i), \Psi\rangle_H = \langle v_i, \Psi\rangle_H
\end{aligned} \tag{4.28}$$

with v_i defined in (4.27).

If we set $\Psi = w_i$ we obtain for the implicit Euler method ($\theta = 1$)

$$\begin{aligned} ||w_i||_H^2 &= \langle w_i, w_{i-1}\rangle_H - \Delta t\ a(t_i;\ w_i, w_i) + \Delta t \langle v_i, w_i\rangle_H \\ &\leq ||w_i||_H ||w_{i-1}||_H - \Delta t \frac{\kappa}{\alpha} ||w_i||_H^2 + \Delta t ||v_i||_H ||w_i||_H \end{aligned}$$

and hence

$$||w_i||_H \leq \frac{1}{1 + \Delta t \frac{\kappa}{\alpha}} (||w_{i-1}||_H + \Delta t ||v_i||_H) \leq ||w_{i-1}||_H + \Delta t ||v_i||_H. \tag{4.29}$$

Before we derive the estimate for the Crank-Nicolson scheme, we show that

$$a(t_i;\ w_i, w_i + w_{i-1}) + a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}) \geq -\frac{c_{lip}^2 \alpha ||S||_2^2}{16\kappa} \Delta t^2 (||w_i||_H + ||w_{i-1}||_H)^2.$$

We use the assumptions on the bilinear form to derive

$$\begin{aligned} &a(t_i;\ w_i, w_i + w_{i-1}) + a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}) \\ &= a(t_i;\ w_i + w_{i-1}, w_i + w_{i-1}) + (a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}) - a(t_i;\ w_{i-1}, w_i + w_{i-1})) \\ &\geq \kappa ||w_i + w_{i-1}||_V^2 - c_{lip} |t_i - t_{i-1}|\ ||w_{i-1}||_V ||w_i + w_{i-1}||_V \end{aligned}$$

and similarly

$$\begin{aligned} &a(t_i;\ w_i, w_i + w_{i-1}) + a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}) \\ &\geq \kappa ||w_i + w_{i-1}||_V^2 - c_{lip} |t_i - t_{i-1}|\ ||w_i||_V ||w_i + w_{i-1}||_V. \end{aligned}$$

First, we add the last two inequalities and divide by two, then use $w_i \in \mathcal{V}^l$ to estimate with (4.24), and finally complete the squares to obtain

$$\begin{aligned} &a(t_i;\ w_i, w_i + w_{i-1}) + a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}) \\ &\geq \kappa ||w_i + w_{i-1}||_V^2 - \frac{1}{2} c_{lip} \Delta t (||w_i||_V + ||w_{i-1}||_V) ||w_i + w_{i-1}||_V \\ &\geq \frac{\kappa}{\alpha} (||w_i + w_{i-1}||_H^2 - \frac{c_{lip} \alpha ||S||_2}{2\kappa} \Delta t (||w_i||_H + ||w_{i-1}||_H) ||w_i + w_{i-1}||_H) \\ &= \frac{\kappa}{\alpha} \Big((||w_i + w_{i-1}||_H - \frac{c_{lip} \alpha ||S||_2}{4\kappa} \Delta t (||w_i||_H + ||w_{i-1}||_H))^2 \\ &- \frac{c_{lip}^2 \alpha^2 ||S||_2^2}{16\kappa^2} \Delta t^2 (||w_i||_H + ||w_{i-1}||_H)^2 \Big) \geq -2\xi \Delta t^2 (||w_i||_H + ||w_{i-1}||_H)^2 \end{aligned}$$

with $\xi := c_{lip}^2 \alpha ||S||_2^2 / (32\kappa)$, which was claimed to be shown above.

We return to formula (4.28) and use $\Psi = w_i + w_{i-1} \in \mathcal{V}^l$ in the Crank-Nicolson case ($\theta = 1/2$)

$$||w_i||_H^2 = ||w_{i-1}||_H^2 - \frac{\Delta t}{2} (a(t_i;\ w_i, w_i + w_{i-1}) + a(t_{i-1};\ w_{i-1}, w_i + w_{i-1}))$$

$$
\begin{aligned}
&+ \Delta t \langle v_i, w_i + w_{i-1} \rangle_H \\
&\leq ||w_{i-1}||_H^2 + \xi \Delta t^3 (||w_i||_H + ||w_{i-1}||_H)^2 + \Delta t ||v_i||_H (||w_i||_H + ||w_{i-1}||_H)
\end{aligned}
$$

and therefore

$$
||w_i||_H - ||w_{i-1}||_H = \frac{||w_i||_H^2 - ||w_{i-1}||_H^2}{||w_i||_H + ||w_{i-1}||_H} \leq \xi \Delta t^3 (||w_i||_H + ||w_{i-1}||_H) + \Delta t ||v_i||_H.
$$

If we assume that Δt is chosen so small that $\xi \Delta t^3 < 1/2$ we obtain

$$
||w_i||_H \leq \frac{1 + \xi \Delta t^3}{1 - \xi \Delta t^3} ||w_{i-1}||_H + \frac{1}{1 - \xi \Delta t^3} \Delta t ||v_i||_H \leq (1 + 4\xi \Delta t^3) ||w_{i-1}||_H + 2\Delta t ||v_i||_H. \tag{4.30}
$$

□

After proving these lemmas, we can state and show the main error estimate. First, we consider error 1 as defined in (4.20).

Theorem 4.2.6. ***(POD error 1)***
Let $y(t)$ be the solution of (4.16), $\{y_i^{l,1}\}_{i=0}^n$ the solution of (4.18). Then with appropriate constants C_i $(i = 0, 1, 2)$, independent of n, we have

$$
\frac{1}{n} \sum_{i=1}^{n} ||y(t_i) - y_i^{l,1}||_H^2 \leq C_0 ||y(t_0) - \Pi_H^l y(t_0)||_H^2 + C_1 \Delta t^j + C_2 ||S||_2 \sum_{j=l+1}^{r} \lambda_j \tag{4.31}
$$

with $j = 2$ for the implicit Euler method assuming $y_{tt} \in L_2([0, T]; H)$
and $j = 4$ for the Crank-Nicolson method, assuming $y_{ttt} \in L_2([0, T]; H)$ and Δt sufficiently small.
Furthermore, for some constant C we have

$$
||y(t_0) - \Pi_H^l y(t_0)||_H^2 \leq n \; C \sum_{j=l+1}^{r} \lambda_j.
$$

Proof. Define the snapshots $\bar{y}_i$:

$$
\begin{aligned}
\bar{y}_i &= y(t_{i-1}) && i = 1, \ldots, n+1 \\
\bar{y}_{i+n+1} &= \bar{\partial} y(t_i) = \frac{y(t_i) - y(t_{i-1})}{\Delta t} && i = 1, \ldots, n.
\end{aligned}
$$

Let $\dim(\text{span}(\bar{y}_1, \ldots, \bar{y}_{2n+1})) = r$. We compute the POD basis $\Psi_1, \ldots, \Psi_r$ with the corresponding eigenvalues $\lambda_1, \ldots, \lambda_r$ using the norm $|| \cdot ||_H$. For simplicity, the weighting factors are set constant, i.e. $\gamma_i = \frac{1}{2n+1} \; \forall \; i$. However, a different choice with $\gamma_i \neq \gamma_j$, for $i \neq j$, would only cause slight modifications. Denote by $\mathcal{V}^l$ the space spanned by $\{\Psi_i\}_{i=1}^l$. Then (4.9)

yields:

$$\frac{1}{2n+1}\sum_{i=0}^{n}\left|\left|y(t_i)-\Pi_H^l y(t_i)\right|\right|_H^2+\frac{1}{2n+1}\sum_{i=1}^{n}\left|\left|\bar{\partial}y(t_i)-\Pi_H^l\bar{\partial}y(t_i)\right|\right|_H^2=\sum_{k=l+1}^{r}\lambda_k. \tag{4.32}$$

Let us define

$$w_i^1=y_i^{l,1}-\Pi_{a,t_i}^l y(t_i)\quad\text{and}\quad w_i^2=\Pi_{a,t_i}^l y(t_i)-y(t_i)$$

so that the triangle inequality yields:

$$\frac{1}{n}\sum_{i=1}^{n}||y_i^{l,1}-y(t_i)||_H^2\leq\frac{2}{n}\sum_{i=1}^{n}||w_i^1||_H^2+\frac{2}{n}\sum_{i=1}^{n}||w_i^2||_H^2. \tag{4.33}$$

Let us first give an estimate for w_i^2. Using the assumption on the norms of the Hilbert spaces in assumption 4.1a), lemma 4.2.4 (4.22), (4.24) and (4.32):

$$\frac{1}{n}\sum_{i=1}^{n}||w_i^2||_H^2\leq\frac{1}{n}\frac{\alpha\beta||S||_2}{\kappa}\sum_{i=1}^{n}\left|\left|y(t_i)-\Pi_H^l y(t_i)\right|\right|_H^2\leq\frac{3\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j. \tag{4.34}$$

Since we included the difference quotients in the set of snapshots we obtain analogously:

$$\frac{1}{n}\sum_{i=1}^{n}||\bar{\partial}y(t_i)-\Pi_{a,t_i}^l\bar{\partial}y(t_i)||_H^2\leq\frac{3\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j, \tag{4.35}$$

a result that will be needed later.

Estimates for w_i^1 are provided in lemma 4.2.5: For the implicit Euler we have

$$||w_i^1||_H\leq||w_{i-1}^1||_H+\Delta t||v_i||_H \tag{4.36}$$

and for Crank-Nicolson

$$||w_i^1||_H\leq(1+4\xi\Delta t^3)||w_{i-1}^1||_H+2\Delta t||v_i||_H \tag{4.37}$$

with $v_i=r_i+z_i$ from (4.27), where

$$r_i:=\theta\cdot y_t(t_i)+(1-\theta)\cdot y_t(t_{i-1})-\bar{\partial}y(t_i)\quad\text{and}\quad z_i:=\bar{\partial}y(t_i)-\bar{\partial}\Pi_{a,t_i}^l y(t_i).$$

If we apply lemma 4.2.7 formulated below to (4.36) and (4.37) this leads to

$$\begin{aligned}\theta=1:\quad&\frac{1}{n}\sum_{i=1}^{n}||w_i^1||_H^2\leq\max_{1\leq i\leq n}||w_i^1||_H^2\leq 2||w_0^1||_H^2+2n\sum_{k=1}^{n}\Delta t^2||v_k||_H^2\\&\leq 2||w_0^1||_H^2+4T\Delta t\sum_{k=1}^{n}(||r_k||_H^2+||z_k||_H^2)\\\theta=\frac{1}{2}:\quad&\frac{1}{n}\sum_{i=1}^{n}||w_i^1||_H^2\leq 2e^{8\Delta t^3\xi n}\Big(||w_0^1||_H^2+\frac{1-e^{-8\Delta t^3\xi n}}{8\Delta t^3\xi}\sum_{k=1}^{n}4\Delta t^2||v_k||_H^2\Big)\end{aligned} \tag{4.38}$$

$$= 2e^{8\Delta t^2 \xi T}\Big(||w_0^1||_H^2 + \frac{1-e^{-8\Delta t^2\xi T}}{2\Delta t\xi}\sum_{k=1}^{n}||v_k||_H^2\Big)$$
$$\leq \tilde{C}_{CN}||w_0^1||_H^2 + C_{CN}\Delta t\sum_{k=1}^{n}(||r_k||_H^2 + ||z_k||_H^2). \tag{4.39}$$

We split z_i as follows and use the results of lemma 4.2.4, i.e. (4.23) and (4.22), to estimate the second summand:

$$\begin{aligned}||z_i||_H^2 &\leq 2||\bar{\partial}y(t_i) - \Pi_{a,t_i}^l\bar{\partial}y(t_i))||_H^2 + 2||\Pi_{a,t_i}^l\bar{\partial}y(t_i) - \bar{\partial}\Pi_{a,t_i}^l y(t_i)||_H^2\\ &= 2||\bar{\partial}y(t_i) - \Pi_{a,t_i}^l\bar{\partial}y(t_i)||_H^2 + \frac{2}{\Delta t^2}||\Pi_{a,t_i}^l y(t_{i-1}) - \Pi_{a,t_{i-1}}^l y(t_{i-1})||_H^2\\ &\leq 2||\bar{\partial}y(t_i) - \Pi_{a,t_i}^l\bar{\partial}y(t_i)||_H^2 + 2\alpha\frac{c_{lip}^2}{\kappa^2}||\Pi_{a,t_{i-1}}^l y(t_{i-1}) - y(t_{i-1})||_V^2\\ &\leq 2||\bar{\partial}y(t_i) - \Pi_{a,t_i}^l\bar{\partial}y(t_i)||_H^2 + 2\alpha\frac{c_{lip}^2}{\kappa^2}\frac{\beta||S||_2}{\kappa}||\Pi_H^l y(t_{i-1}) - y(t_{i-1})||_H^2.\end{aligned}$$

We use (4.35) for the first part and (4.32) for the second to get

$$\frac{1}{n}\sum_{i=1}^{n}||z_i||_H^2 \leq \frac{6\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j + \frac{c_{lip}^2}{\kappa^2}\frac{6\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j. \tag{4.40}$$

With regard to r_i one can easily show the following results:

$$\theta = 1: \quad \sum_{i=1}^{n}||y_t(t_i) - \bar{\partial}y(t_i)||_H^2 \leq \Delta t\int_0^T ||y_{tt}(s)||_H^2\ ds = \bar{C}\Delta t, \tag{4.41}$$

$$\theta = \frac{1}{2}: \quad \sum_{i=1}^{n}\Big|\Big|\frac{1}{2}y_t(t_i) + \frac{1}{2}y_t(t_{i-1}) - \bar{\partial}y(t_i)\Big|\Big|_H^2 \leq \frac{\Delta t^3}{16}\int_0^T ||y_{ttt}(s)||_H^2\ ds = \tilde{C}\Delta t^3 \tag{4.42}$$

under the assumption that $y_{tt} \in L^2([0,T];H)$ for $\theta = 1$ and $y_{ttt}(t) \in L^2([0,T];H)$ for $\theta = \frac{1}{2}$.

Altogether, we obtain for $ERR_1 = \frac{1}{n}\sum_{i=1}^{n}||y_i^{l,1} - y(t_i)||_H^2$ combining (4.33) and (4.34)

$$ERR_1 \leq \frac{2}{n}\sum_{i=1}^{n}||w_i^1||_H^2 + \frac{2}{n}\sum_{i=1}^{n}||w_i^2||_H^2 \leq \frac{2}{n}\sum_{i=1}^{n}||w_i^1||_H^2 + \frac{6\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j.$$

With appropriate constants $d_1, ..., d_4$ we estimate further using (4.38), (4.40), (4.41) or (4.39), (4.40), (4.42) and $j = 2$ for implicit Euler and $j = 4$ for Crank-Nicolson

$$\begin{aligned}ERR_1 &\leq d_1||w_0^1||_H^2 + d_2\Delta t\sum_{k=1}^{n}(||r_k||_H^2 + ||z_k||_H^2) + \frac{6\alpha\beta||S||_2}{\kappa}\sum_{j=l+1}^{r}\lambda_j\\ &\leq d_1||w_0^1||_H^2 + d_3\Delta t^j + d_4\sum_{j=l+1}^{r}\lambda_j\end{aligned}$$

which yields to the proposition. □

The following lemma gives a useful estimate which is being used in the proof of the previous theorem.

Lemma 4.2.7. *Assume that $r_i \leq (1+\delta)r_{i-1}+b_i$, $i = 1, \ldots, n$, holds for some given sequence b_i and some r_0. Then*

$$\max_{1\leq i\leq n} |r_i|^2 \leq 2e^{2\delta n}\left(r_0^2 + \frac{1-e^{-2\delta n}}{2\delta}\sum_{k=1}^{n} b_k^2\right) \qquad \text{if} \quad \delta > 0, \tag{4.43}$$

$$\max_{1\leq i\leq n} |r_i|^2 \leq 2r_0^2 + 2n\sum_{k=1}^{n} b_k^2 \qquad \text{if} \quad \delta = 0. \tag{4.44}$$

Proof. We only prove the proposition for $\delta > 0$ since the special case $\delta = 0$ can easily be obtained from this. From the assumption we infer that $r_i \leq e^{\delta i}r_0 + \sum_{k=1}^{i} e^{\delta(i-k)}b_k$. Since $\delta > 0$ an application of the binomial formula as well as the Cauchy-Schwarz inequality and a geometric series argument we obtain that

$$\begin{aligned}\max_{1\leq i\leq n} |r_i|^2 &\leq 2e^{2\delta n}r_0^2 + 2\Big(\sum_{k=1}^{n} e^{\delta(n-k)}b_k\Big)^2 \leq 2e^{2\delta n}r_0^2 + 2\Big(\sum_{k=1}^{n} e^{2\delta(n-k)}\sum_{k=1}^{n} b_k^2\Big)\\ &\leq 2e^{2\delta n}\Big(r_0^2 + 2\frac{1-e^{-2\delta n}}{e^{2\delta}-1}\sum_{k=1}^{n} b_k^2\Big) \leq 2e^{2\delta n}\Big(r_0^2 + 2\frac{1-e^{-2\delta n}}{2\delta}\sum_{k=1}^{n} b_k^2\Big).\end{aligned}$$

□

Thus, the error can be divided into two different parts. On the one hand, the error resulting from the time discretization via the Euler and Crank-Nicolson method (this is the first summand). As expected the latter yields to a better order (Δt^4 vs. Δt^2). On the other hand, we got the error $const \cdot \sum_{j=l+1}^{r} \lambda_j$, resulting from the projection on the POD space. Using the maximum number of POD basis functions this error drops out.

In the next theorem, we take a look at error 2 (cf. (4.21)). Here, we estimate the difference between the POD solution compared to a discretized FE solution.

Theorem 4.2.8. ***(POD error 2)***
Let $\{y_i^{FE}\}_{i=0}^n$ be the finite element solution using the finite element space H^{n_x} in the Galerkin approximation. Let $\{y_i^{l,2}\}_{i=0}^n$ be the solution of problem (4.18) based on the FEM snapshots. Then with appropriate constants $\tilde{C}_0, \tilde{C}_1$ independent of n, we have for the implicit Euler method and, for sufficiently small Δt, also for the Crank-Nicolson method

$$\frac{1}{n}\sum_{i=1}^{n} \big|\big|y_i^{FE} - y_i^{l,2}\big|\big|_H^2 \leq \tilde{C}_0\ ||y_0^{FE} - \Pi_H^l y_0^{FE}||_H^2 + \tilde{C}_1||S||_2 \sum_{j=l+1}^{r} \lambda_j$$

where $||y_0^{FE} - \Pi_H^l y_0^{FE}||_H^2 \leq 3n\sum_{j=l+1}^{r} \lambda_j$.

Proof. The proof is analogous to theorem 4.2.6. Instead of $y(t_i)$ we use the snapshots y_i^{FE}. Defining $w_i^1 = y_i^{l,2} - \Pi_{a,t_i}^l y_i^{FE}$ the main difference is:

$$\begin{aligned}
&\langle \bar{\partial} w_i^1, \Psi \rangle_H + \theta \cdot a(t_i;\ w_i^1, \Psi) + (1-\theta) \cdot a(t_{i-1};\ w_{i-1}^1, \Psi) = \\
&= \langle \bar{\partial} y_i^{l,2}, \Psi \rangle_H + \theta \cdot a(t_i;\ y_i^{l,2}, \Psi) + (1-\theta) \cdot a(t_{i-1};\ y_{i-1}^{l,2}, \Psi) \\
&\quad - \langle \bar{\partial} \Pi_{a,t_i}^l y_i^{FE}, \Psi \rangle_H - \theta \cdot a(t_i;\ \Pi_{a,t_i}^l y_i^{FE}, \Psi) - (1-\theta) \cdot a(t_{i-1};\ \Pi_{a,t_i}^l y_{i-1}^{FE}, \Psi) = \\
&= \theta \cdot L(t_i;\ \Psi) + (1-\theta) \cdot L(t_{i-1};\ \Psi) \\
&\quad - \theta \cdot a(t_i;\ y_i^{FE}, \Psi) - (1-\theta) \cdot a(t_{i-1};\ y_{i-1}^{FE}, \Psi) - \langle \bar{\partial} P_{t_i}^p y_i^{FE}, \Psi \rangle_H \\
&= \langle \bar{\partial} y_i^{FE} - \bar{\partial} \Pi_{a,t_i}^l y_i^{FE}, \Psi \rangle_H =: \langle v_i, \Psi \rangle_H.
\end{aligned}$$

Compared to theorem 4.2.6 the r_i's drop out, which leads immediately to the statement of the theorem. □

If we use the maximal number of POD basis functions, the whole error is equal to zero because the error resulting from the time discretization is present in both solutions.

Note that the norm $||S||_2$, which occurs in the estimates in theorem 4.2.6 and 4.2.8, in general depends on n. However, this factor can be avoided by using the stronger topology V in (4.1) (cf. Kunisch and Volkwein (2001)).

Error 2 seems to be more interesting because in practice we do not have the exact solution $y(t)$ of our PIDE, but only an approximation, e.g. from a finite element method, available.

The Crank-Nicolson method and the implicit Euler scheme may also be combined to achieve a smoothing as proposed by Rannacher (cf. section 3.2.2). We do not show a corresponding theoretical result, but theorem 4.2.8 should still hold true. Actually, numerical results in section 3.2.4 have already shown that the Rannacher smoothing provides a much smoother solution in time direction and therefore the singular values in general decay faster leading to a better convergence with respect to the number of POD basis functions l. Section 4.3.1 will provide numerical results supporting this considerations.

We have learned that we need to compute a full solution of the parabolic equation to be able to build a POD model. Thus, to make this model reasonable in terms of computational efficiency, we have to use it in optimization, where the problem has to be solved repeatedly.

4.2.2 Error Estimates in Optimal Control Problems

We now want to use the model order reduction technique in the context of an optimal control problem as it has been described in section 3.3.1. Thus, we formulate the optimization problem for a reduced order model space $\mathcal{V}^l \subset V$ that will be specified further below.

The reduced objective function is given by

$$f_l(u) = \frac{1}{2} \sum_{i=1}^{D} ||C y_{k_i}^l(u) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2} ||u||_U^2. \tag{4.45}$$

Here, $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ is the POD approximation to the state equation discretized with the

θ-method, i.e

$$\begin{aligned} &\bar{\partial} y_k^l + \theta A(u;\ t_k) y_k^l + (1-\theta) A(u;\ t_{k-1}) y_{k-1}^l = \\ &\qquad \theta l(u;\ t_k) + (1-\theta) l(u;\ t_{k-1}), \qquad k = 1, \dots, n_t \\ &y_0^l = y_0 \end{aligned} \tag{4.46}$$

in the sense of $\mathcal{V}^l$.

The corresponding discretized derivative has also been derived in section 3.3.1, where, for given weights ω_k^i ($k = k_{i-1}, \dots, k_i$, $i = 1, \dots, D$), we have

$$f_l'(u)\delta u = \Delta t \sum_{i=1}^{D} \sum_{k=k_{i-1}}^{k_i} \omega_k^i \langle p_k^{i,l}, A'(u;t_k)\delta u\ y_k^l - l'(u,t_k)\delta u \rangle_{V,V^*} + \alpha \langle u, \delta u \rangle_U. \tag{4.47}$$

$\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ solve (4.46) and the discrete adjoint variables $\{p_k^{i,l}\}_{k=k_{i-1}}^{k_i} \subset \mathcal{V}^l$ are solutions to

$$\begin{aligned} &-\bar{\partial} p_k^{D,l} + \theta A^*(u;\ t_{k-1}) p_{k-1}^{D,l} + (1-\theta) A^*(u;\ t_k) p_k^{D,l} = 0, \qquad k = n_t - 1, \dots, k_{D-1} \\ &p_{n_t}^{D,l} = -C^*(C y_{k_D}^l - d_D), \end{aligned} \tag{4.48}$$

and, for $i = D-1, \dots, 1$,

$$\begin{aligned} &-\bar{\partial} p_k^{i,l} + \theta A^*(u;\ t_{k-1}) p_{k-1}^{i,l} + (1-\theta) A^*(u;\ t_k) p_k^{i,l} = 0, \qquad k = k_i - 1, \dots, k_{i-1} \\ &p_{k_i}^{i,l} = -C^*(C y_{k_i}^l - d_i) + p_{k_i}^{i+1,l} \end{aligned} \tag{4.49}$$

in the sense of $\mathcal{V}^l$.

As in the previous section, we want to estimate the error between the reduced objective function, (4.45), and a reference objective function, and the reduced derivative, (4.47), and a reference derivative, respectively. As reference, we might take the infinite-dimensional solution (as in theorem 4.2.6) or a numerical approximation (as in theorem 4.2.8). In the following, we will restrict ourselves to the latter case where we assume to know finite element solutions $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x} \subset V$ and $\{p_k^{i,FE}\}_{k=k_{i-1}}^{k_i} \subset H^{n_x}$ for the problems (4.46) and (4.48), (4.49), respectively. The main reason is that in most applications – and this also includes the calibration of option pricing models that is considered in this thesis – continuous solutions are not available, and furthermore, we are mainly interested in the error resulting from the reduced order model and not the time discretization errors.

The following two lemmas provide first estimates for the error of the objective function and the derivative. Note that we assume the control u to be fixed but arbitrary.

Lemma 4.2.9. *For fixed but arbitrary $u \in \mathcal{U}$, let $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ be a solution to (4.46) and $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x}$ a solution to (4.46) where we replace $\mathcal{V}^l$ by H^{n_x}. Writing $f_{FE}(u)$ and $f_l(u)$ as the corresponding objective functions in the sense of (4.45), then there exists a constant*

$c > 0$ independent of n_t but dependent of u, such that the following estimate holds:

$$\left|f_{FE}(u) - f_l(u)\right|^2 \leq cD \sum_{i=1}^{D} ||y_{k_i}^{FE}(u) - y_{k_i}^{l}(u)||_H^2. \tag{4.50}$$

Proof. By definition, the left-hand side of (4.50) yields

$$\left|f_{FE}(u) - f_l(u)\right|^2 = \frac{1}{4}\Big(\sum_{i=1}^{D} \left(||Cy_{k_i}^{FE}(u) - d_i||_{\mathcal{H}}^2 - ||Cy_{k_i}^{l}(u) - d_i||_{\mathcal{H}}^2\right)\Big)^2. \tag{4.51}$$

Using the inequality $||x||^2 - ||y||^2 = \langle x, x-y\rangle + \langle x-y, y\rangle \leq ||x-y||(||x|| + ||y||)$, the fact that $C \in \mathcal{L}(H, \mathcal{H})$ and that the solutions $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ and $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x}$ are bounded in H for fixed u, we can estimate the summands as follows:

$$\begin{aligned}
&||Cy_{k_i}^{FE}(u) - d_i||_{\mathcal{H}}^2 - ||Cy_{k_i}^{l}(u) - d_i||_{\mathcal{H}}^2 \\
&\quad \leq ||C(y_{k_i}^{FE}(u) - y_{k_i}^{l}(u))||_{\mathcal{H}}\big(||Cy_{k_i}^{FE}(u) - d_i||_{\mathcal{H}} + ||Cy_{k_i}^{l}(u) - d_i||_{\mathcal{H}}\big) \\
&\quad \leq c_1 ||y_{k_i}^{FE}(u) - y_{k_i}^{l}(u)||_H,
\end{aligned}$$

where $i \in \{1, \dots, D\}$ is arbitrary. Thus, applying the Cauchy-Schwartz inequality, (4.51) yields $\left|f_{FE}(u) - f_l(u)\right|^2 \leq \frac{c_1^2}{4} D \sum_{i=1}^{D} ||y_{k_i}^{FE}(u) - y_{k_i}^{l}(u)||_H^2$. □

Assuming that the operator $C \in \mathcal{L}(H, \mathcal{H})$ contains an approximation of a Dirac delta function, the constant c_1 used in the lemma above depends on the spatial discretization (cf. remark 3.3.2).

However, we see that the error between the objective function based on a finite element discretization and on a POD discretization, respectively, can be estimated by the difference between the state solutions. This will be used later to establish a connection between the error of the function values and the eigenvalues of a POD model as in the previous section.

A similar result as above can be stated for the derivative error.

Lemma 4.2.10. *For fixed but arbitrary $u \in \mathcal{U}$, let $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ be a solution to (4.46) and $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x}$ a solution to (4.46) where we replace $\mathcal{V}^l$ by H^{n_x}. Analogously, $\{p_k^{i,l}\}_{k=k_{i-1}}^{k_i} \subset \mathcal{V}^l$ and $\{p_k^{i,FE}\}_{k=k_{i-1}}^{k_i} \subset H^{n_x}$ $(i = D, \dots, 1)$ are the corresponding adjoint solutions to (4.48) and (4.49), respectively. Writing $f'_{FE}(u)\delta u$ and $f'_l(u)\delta u$ as the corresponding derivatives in the sense of (4.47), then there exist constants $\bar{c}, c_{A'}, c_{l'} > 0$ independent of n_t but dependent of u, such that the following estimate holds for feasible directions δu:*

$$\begin{aligned}
\left|f'_{FE}(u)\delta u - f'_l(u)\delta u\right|^2 \leq c\frac{1}{n_t}\Big(&\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} c_{A'}^2 ||p_k^{i,FE}||_V^2 ||y_k^{FE} - y_k^l||_V^2 \\
&+ \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \left(c_{A'}||y_k^l||_V + c_{l'}\right)^2 ||p_k^{i,FE} - p_k^{i,l}||_V^2\Big)||\delta u||_U^2.
\end{aligned} \tag{4.52}$$

Proof. By definition, the left-hand side of (4.52) yields

$$\begin{aligned}\left|f'_{FE}(u)\delta u - f'_l(u)\delta u\right|^2 = \Big|\Delta t \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \omega_k^i \Big(&\langle p_k^{i,FE}, A'(u;t_k)\delta u\ y_k^{FE} - l'(u,t_k)\delta u\rangle_{V,V^*} \\ &- \langle p_k^{i,l}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*}\Big)\Big|^2. \end{aligned} \tag{4.53}$$

For an arbitrary $k = k_{i-1}, \dots, k_i$ $(i = 1, \dots, D)$, we can estimate the summands as follows:

$$\begin{aligned}
&\langle p_k^{i,FE}, A'(u;t_k)\delta u\ y_k^{FE} - l'(u,t_k)\delta u\rangle_{V,V^*} - \langle p_k^{i,l}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*} \\
&= \langle p_k^{i,FE}, A'(u;t_k)\delta u\ y_k^{FE} - l'(u,t_k)\delta u\rangle_{V,V^*} - \langle p_k^{i,FE}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*} \\
&\quad + \langle p_k^{i,FE}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*} - \langle p_k^{i,l}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*} \\
&= \langle p_k^{i,FE}, A'(u;t_k)\delta u\ (y_k^{FE} - y_k^{l})\rangle_{V,V^*} + \langle p_k^{i,FE} - p_k^{i,l}, A'(u;t_k)\delta u\ y_k^{l} - l'(u,t_k)\delta u\rangle_{V,V^*} \\
&\le \Big(c_{A'}||p_k^{i,FE}||_V||y_k^{FE} - y_k^{l}||_V + ||p_k^{i,FE} - p_k^{i,l}||_V\big(c_{A'}||y_k^{l}||_V + c_{l'}\big)\Big)||\delta u||_U.
\end{aligned}$$

Thus, setting $\bar\omega = \max\{\omega_k^i : \forall\ i,k\}$ and further using $n_t + D \le 2n_t$, (4.53) yields

$$\begin{aligned}
\left|f'_{FE}(u)\delta u - f'_l(u)\delta u\right|^2 &\le 2(n_t + D)\Delta t^2 \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \\
&\quad \bar\omega^2\Big(c_{A'}^2||p_k^{i,FE}||_V^2||y_k^{FE} - y_k^{l}||_V^2 + ||p_k^{i,FE} - p_k^{i,l}||_V^2\big(c_{A'}||y_k^{l}||_V + c_{l'}\big)^2\Big)||\delta u||_U^2 \\
&\le 4T\frac{T}{n_t}\bar\omega^2\Big(\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} c_{A'}^2||p_k^{i,FE}||_V^2||y_k^{FE} - y_k^{l}||_V^2 \\
&\quad + \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \big(c_{A'}||y_k^{l}||_V + c_{l'}\big)^2||p_k^{i,FE} - p_k^{i,l}||_V^2\Big)||\delta u||_U^2,
\end{aligned}$$

which proves the lemma. □

What is striking concerning the two lemmas above is the fact that both errors, the function as well as the gradient error, mainly depend on the difference of the state solutions, $y_k^{FE}(u)$, $y_k^{l}(u)$, and, concerning the gradient, also on the difference of the adjoint solutions $p_k^{i,FE}(u)$, $p_k^{i,l}(u)$ (for the sake of readability we omit the dependence of the state and the adjoint on the fixed control u in the following).

The results of the previous section now tell us how to estimate these differences. On the one hand, to estimate the difference between y_k^{FE} and y_k^{l} in the sense of theorem 4.2.8, the POD subspace $\mathcal{V}^l$ needs to include information of the state equation $\{y_k^{FE}\}_{k=0}^{n_t}$. But it is, on the other hand, also important that the POD basis contains information of the adjoint solutions $\{p_k^{i,FE}\}_{k=k_{i-1}}^{k_i}$ to be able to estimate the error between $p_k^{i,FE}$ and $p_k^{i,l}$ $(i = 1, \dots, D)$. Thus, for a fixed but arbitrary control u, we define the set of snapshots

$$y_k^{FE}, \quad k = 0, \dots, n_t,$$

$$p_k^{i,FE}, \quad k = k_{i-1}, \dots, k_i, \ i = 1, \dots, D,$$

where we also include the corresponding difference quotients for y^{FE} and $p^{i,FE}$ $(i = 1, \dots, D)$. Denoting by r the rank of all snapshots, we write $\mathcal{V}^r$ for the space spanned by them and $\mathcal{V}^l$, $l \leq r$, for the corresponding l-dimensional subspace. We further denote by η_S, $\eta_A > 0$ two weighting factors that manage the importance of the state and adjoint snapshots, respectively, for the computation of the POD basis. This weighting leads to the following POD projection error:

$$\begin{aligned} &\frac{\eta_S}{4n_t + D + 1}\Big(\sum_{i=0}^{n_t} \Big|\Big|y_k^{FE} - \Pi_H^l y_k^{FE}\Big|\Big|_H^2 + \sum_{i=1}^{n_t} \Big|\Big|\bar{\partial} y_k^{FE} - \Pi_H^l \bar{\partial} y_k^{FE}\Big|\Big|_H^2\Big) + \\ &\frac{\eta_A}{4n_t + D + 1}\Big(\sum_{i=1}^{D} \sum_{k=k_{i-1}}^{k_i} \Big|\Big|p_k^{i,FE} - \Pi_H^l p_k^{i,FE}\Big|\Big|_H^2 + \sum_{i=1}^{D} \sum_{k=k_i-1}^{k_{i-1}} \Big|\Big|\bar{\partial} p_k^{i,FE} - \Pi_H^l \bar{\partial} p_k^{i,FE}\Big|\Big|_H^2\Big) = \sum_{j=l+1}^{r} \lambda_j. \end{aligned} \tag{4.54}$$

Of course, the λ_j's strongly depend on the weighting factors, whose choice will be discussed further below. Just note that only the relationship between η_S and η_A is important and not the absolute values. Given the POD space corresponding to the snapshot setting above, we can now estimate the error of the objective function and the derivative in terms of the sum $\sum_{j=l+1}^{r} \lambda_j$.

When we use an extended POD basis including the adjoint snapshots, this has some impacts on the error estimates that have been derived in the previous section. Further, we have to stress that an POD error of the state solution affects the error of the reduced adjoint solution since the state appears in the end conditions. We mention some issues that arise in this context in the following remark.

Remark 4.2.11. ***(POD for state and adjoint)***
For fixed but arbitrary $u \in \mathcal{U}$, let $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ be a solution to (4.46) and $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x}$ a solution to (4.46) where we replace $\mathcal{V}^l$ by H^{n_x}. Analogously, $\{p_k^{i,l}\}_{k=k_{i-1}}^{k_i} \subset \mathcal{V}^l$ and $\{p_k^{i,FE}\}_{k=k_{i-1}}^{k_i} \subset H^{n_x}$ $(i = D, \dots, 1)$ are the corresponding adjoint solutions to (4.48) and (4.49), respectively. Given the assumptions of theorem 4.2.8 and the space $\mathcal{V}^l$ as described above, we can state the following results:

a) The POD error for the state equation can now be estimated by

$$\begin{aligned} \frac{1}{n_t} \sum_{k=1}^{n_t} ||y_k^{FE} - y_k^l||_H^2 &\leq \hat{C}_0 \ ||y_0^{FE} - \Pi_H^l y_0^{FE}||_H^2 + \hat{C}_1 ||S||_2 \frac{1}{\eta_S} \sum_{j=l+1}^{r} \lambda_j \\ &\leq \frac{1}{\eta_S}(5\hat{C}_0 n_t + \hat{C}_1 ||S||_2) \sum_{j=l+1}^{r} \lambda_j, \end{aligned} \tag{4.55}$$

and an error at the initial condition $||y_0^{FE} - y_0^l||_H^2 \leq \frac{1}{\eta_S} 5n_t \sum_{j=l+1}^{r} \lambda_j$.

b) The adjoint equations are backward. A simple change of variables leads to a forward equation in the sense of theorem 4.2.8. However, in the further notation, we keep the backward formulation.

c) Writing $k_i - k_{i-1} = \varpi_i n_t$ $(\varpi_i \in (0, 1])$, the adjoint error can be estimated recursively

($i = D, \dots, 1$) by the adjoint projection error, ϵ_{proj}, plus the error due to a disturbed end condition, ϵ_{end}:

$$\frac{1}{n_t \varpi_i} \sum_{k=k_{i-1}}^{k_i - 1} ||p_k^{i,FE} - p_k^{i,l}||_H^2 \leq \epsilon_{proj}(i) + \epsilon_{end}(i) \tag{4.56}$$

with

$$\epsilon_{proj}(i) = \frac{1}{\eta_A}(5C_0 n_t + C_1 ||S||_2 \frac{1}{\varpi_i}) \sum_{j=l+1}^{r} \lambda_j \tag{4.57}$$

and

$$\epsilon_{end}(i) = \frac{n_t}{\eta_S} \tilde{C}_0 (5\hat{C}_0 n_t + \hat{C}_1 ||S||_2) \sum_{j=l+1}^{r} \lambda_j + 2c||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2, \tag{4.58}$$

where $||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2 = n_t \varpi_{i+1}(\epsilon_{proj}(i+1) + \epsilon_{end}(i+1))$ can be estimated recursively via (4.56) for $i < D$ (we set $\epsilon_{proj}(D+1) = \epsilon_{end}(D+1) = 0$).

We can further estimate the error at the end condition of the backward adjoints for $i = D, \dots, 1$ as follows:

$$||p_{k_i}^{i,FE} - p_{k_i}^{i,l}||_H^2 \leq \frac{n_t}{\eta_A} 10 \sum_{j=l+1}^{r} \lambda_j + \frac{n_t}{\eta_S} C_1 (5\hat{C}_0 n_t + \hat{C}_1 ||S||_2) \sum_{j=l+1}^{r} \lambda_j + ||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2. \tag{4.59}$$

Proof. a) The changes in the constants are due to the inclusion of the adjoint snapshots in the sense of (4.54).

c) We begin by introducing a further solution $\{\tilde{p}_k^{i,l}\}_{k=k_{i-1}}^{k_i} \subset \mathcal{V}^l$ to (4.48) and (4.49), respectively, with a new end condition given by $\tilde{p}_{k_i}^{i,l} = \Pi_H^l[-C^*(Cy_{k_i}^{FE} - d_i) + p_{k_i}^{i+1,FE}]$. Note that in the notation of (4.48) and (4.49), the projection operator Π_H^l is omitted, although it is meant in the sense of $\mathcal{V}^l$. Here, it is not omitted to stress the fact that we are dealing with a solution in $\mathcal{V}^l$. Further note that $p_{k_D}^{D+1,FE} := 0$. We easily derive the following inequality:

$$\frac{1}{n_t \varpi_i} \sum_{k=k_{i-1}}^{k_i - 1} ||p_k^{i,FE} - p_k^{i,l}||_H^2 \leq \frac{2}{n_t \varpi_i} \sum_{k=k_{i-1}}^{k_i - 1} ||p_k^{i,FE} - \tilde{p}_k^{i,l}||_H^2 + \frac{2}{n_t \varpi_i} \sum_{k=k_{i-1}}^{k_i - 1} ||\tilde{p}_k^{i,l} - p_k^{i,l}||_H^2. \tag{4.60}$$

The first summand on the right-hand side, denoted by $\epsilon_{proj}(i)$ in the following, can be estimated by theorem 4.2.8:

$$\epsilon_{proj}(i) \leq C_0 ||p_{k_i}^{i,FE} - \Pi_H^l p_{k_i}^{i,FE}||_H^2 + C_1 ||S||_2 \frac{1}{\eta_A \varpi_i} \sum_{j=l+1}^{r} \lambda_j \tag{4.61}$$

$$\leq \frac{1}{\eta_A}(5C_0 n_t + C_1||S||_2 \frac{1}{\varpi_i}) \sum_{j=l+1}^{r} \lambda_j. \tag{4.62}$$

Using parts of the proofs for theorem 4.2.6 and 4.2.8, especially (4.38) and (4.39), the second summand in (4.60), $\epsilon_{end}(i)$, i.e. the average error between $\tilde{p}_k^{i,l}$ and $p_k^{i,l}$, can be estimated by its inital error:

$$\begin{aligned}
\epsilon_{end}(i) &\leq c||\tilde{p}_{k_i}^{i,l} - p_{k_i}^{i,l}||_H^2 \leq c||\Pi_H^l[-C^*(Cy_{k_i}^{FE} - d_i) + p_{k_i}^{i+1,FE}] - \Pi_H^l[-C^*(Cy_{k_i}^{l} - d_i) + p_{k_i}^{i+1,l}]||_H^2 \\
&\leq c||-C^*(Cy_{k_i}^{FE} - d_i) + p_{k_i}^{i+1,FE} + C^*(Cy_{k_i}^{l} - d_i) - p_{k_i}^{i+1,l}||_H^2 \\
&\leq 2cc_c^2||y_{k_i}^{FE} - y_{k_i}^{l}||_H^2 + 2c||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2 \\
&\leq \frac{n_t}{\eta_S}\tilde{C}_0(5\hat{C}_0 n_t + \hat{C}_1||S||_2) \sum_{j=l+1}^{r} \lambda_j + 2c||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2,
\end{aligned}$$

where the last estimate uses (4.55). The term $||p_{k_i}^{i+1,FE} - p_{k_i}^{i+1,l}||_H^2$ has to be estimated recursively. (4.59) can be easily estimated using the same techniques as above. □

The partition of the adjoint equation – which has been necessary from a theoretical point of view –, where the end condition of one adjoint, $p_{k_i}^{i,FE}$, involves the solution of the previous adjoint in the same time slice, $p_{k_i}^{i+1,FE}$, leads to this recursive representation of the adjoint error as seen above. In case of $D = 1$, we get:

$$\begin{aligned}
&\frac{1}{n_t}\sum_{k=0}^{n_t-1} ||p_k^{FE} - p_k^l||_H^2 \\
&\quad \leq \Big(\frac{1}{\eta_A}(5C_0 n_t + C_1||S||_2) + \frac{n_t}{\eta_S}\tilde{C}_0(5\hat{C}_0 n_t + \hat{C}_1||S||_2)\Big) \sum_{j=l+1}^{r} \lambda_j
\end{aligned} \tag{4.63}$$

with an error of the end condition

$$||p_{n_t}^{FE} - p_{n_t}^l||_H^2 \leq \Big(\frac{n_t}{\eta_A}10 + \frac{n_t}{\eta_S}C_1(5\hat{C}_0 n_t + \hat{C}_1||S||_2)\Big) \sum_{j=l+1}^{r} \lambda_j. \tag{4.64}$$

The highest degree for the factor n_t is two in (4.64) and (4.63). However, the recursive structure yields a factor n_t^{D+1} for arbitrary $D \geq 1$. This is due to the fact that the POD error is given as an average error, and thus, the error for an adjoint in the last time step involves a factor of n_t. The factor n_t^{D+1} seems to be a strong drawback at first sight, however, the numerical results below will show small errors even for small POD bases. Furthermore, n_t^{D+1} occurs with an additional factor $\prod_{i=1}^{D} \varpi_i$, typically very small if D is high.

We are also able to eliminate the factor n_t^{D+1} by considering the adjoint equation based on the first discretize approach. Discontinuities in the right-hand side terms can be ignored, and thus, according to (3.47), we have – instead of D parts – only one adjoint equation:

$$-\bar{\partial}p_{k+1}^l + \theta A^*(u, t_k)p_k^l + (1-\theta)A^*(u, t_k)p_{k+1}^l$$

$$+\sum_{i=1}^{D} C^*(Cy^l_{k_i} - d_i)\mathbb{1}_{k=k_i} = 0 \quad , \quad k = n_t, \dots, 1$$
$$p^l_{n_t+1} = 0.$$

If we want to estimate the corresponding POD adjoint error, we can again split the error into the adjoint projection error and the error due to a disturbed state solution. We just need to rewrite the right-hand side – instead of the disturbed initial condition as above – as $C^*(Cy^l_{k_i} - d_i) = C^*(Cy^{FE}_{k_i} - d_i) + C^*(C(y^l_{k_i} - y^{FE}_{k_i}))$.

We now use the previous results to estimate the error for the reduced function value and the reduced derivative in terms of a sum over the remaining eigenvalues.

Theorem 4.2.12. ***(POD function error)***
For fixed but arbitrary $u \in \mathcal{U}$, let $\{y^l_k\}_{k=0}^{n_t} \subset \mathcal{V}^l$ be a solution to (4.46) and $\{y^{FE}_k\}_{k=0}^{n_t} \subset H^{n_x}$ a solution to (4.46) where we replace $\mathcal{V}^l$ by H^{n_x}. Writing $f_{FE}(u)$ and $f_l(u)$ as the corresponding objective functions in the sense of (4.45) and given the assumptions of remark 4.2.11, then there exist constants $\bar{C}_0, \bar{C}_1 > 0$ independent of n_t but dependent of u, such that the following estimate holds:

$$\begin{aligned} |f_{FE}(u) - f_l(u)|^2 &\leq D\frac{1}{\eta_S} n_t \Big(\bar{C}_0 \ ||y^{FE}_0 - \Pi^l_H y^{FE}_0||^2_H + \bar{C}_1 ||S||_2 \sum_{j=l+1}^{r} \lambda_j \Big) \\ &\leq D\frac{1}{\eta_S} n_t (5\bar{C}_0 n_t + \bar{C}_1 ||S||_2) \sum_{j=l+1}^{r} \lambda_j, \end{aligned} \tag{4.65}$$

where the λ_j's also depend on η_S and η_A.

Proof. Applying lemma 4.2.9, we have to estimate the term $\sum_{i=1}^{D} ||y^{FE}_{k_i}(u) - y^l_{k_i}(u)||^2_H$:

$$|f_{FE}(u) - f_l(u)|^2 \leq cD \sum_{i=1}^{D} ||y^{FE}_{k_i}(u) - y^l_{k_i}(u)||^2_H \leq cD \sum_{k=1}^{n_t} ||y^{FE}_k(u) - y^l_k(u)||^2_H.$$

Equation (4.55) then yields the desired conclusion. □

The first estimate in (4.65) shows a dependence on the number of steps in time direction, n_t, which is due to the fact that the objective function compares values at D certain time instances. If we use an objective function that compares values over the whole time domain, i.e. an integral over the time, this dependence should drop out.

A similar result can be derived for the reduced derivative.

Theorem 4.2.13. ***(POD derivative error)***
For fixed but arbitrary $u \in \mathcal{U}$, let $\{y^l_k\}_{k=0}^{n_t} \subset \mathcal{V}^l$ be a solution to (4.46) and $\{y^{FE}_k\}_{k=0}^{n_t} \subset H^{n_x}$ a solution to (4.46) where we replace $\mathcal{V}^l$ by H^{n_x}. Analogously, $\{p^{i,l}_k\}_{k=k_{i-1}}^{k_i} \subset \mathcal{V}^l$ and $\{p^{i,FE}_k\}_{k=k_{i-1}}^{k_i} \subset H^{n_x}$ $(i = D, \dots, 1)$ are the corresponding adjoint solutions to (4.48) and (4.49), respectively. Writing $f'_{FE}(u)\delta u$ and $f'_l(u)\delta u$ as the corresponding derivatives in the

sense of (4.47) and given the assumptions of remark 4.2.11, there exist constants k_S, $k_A \geq 0$, such that the following estimate holds:

$$\frac{\left|f'_{FE}(u)\delta u - f'_l(u)\delta u\right|^2}{||\delta u||_U^2} \leq \Big(k_S(\frac{1}{\eta_S}, c_p, ||S||_2, n_t) + k_A(\frac{1}{\eta_A}, \frac{1}{\eta_S}, c_y, ||S||_2, n_t)\Big) \sum_{j=l+1}^{r} \lambda_j. \quad (4.66)$$

Here, k_S, k_A depend on $||S||_2$, n_t and bounds, $c_p \geq ||p_k^{i,FE}||_V$, $c_y \geq ||y_k^l||_V$ $(\forall\ i,k)$, for the adjoint and state solution, respectively, as well as on the weighting factors η_S, η_A.

Proof. Let us first apply lemma 4.2.10 to the left-hand side of (4.66). Using the estimate (4.24) and the boundedness of the adjoint solution, $||p_k^{i,FE}||_V \leq c_p$, we can estimate the first sum of (4.52) as follows:

$$\begin{aligned}
&\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} c_{A'}^2||p_k^{i,FE}||_V^2||y_k^{FE} - y_k^l||_V^2 \leq c_{A'}^2||S||_2 \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} ||p_k^{i,FE}||_V^2||y_k^{FE} - y_k^l||_H^2 \\
&\leq 2c_{A'}^2||S||_2 c_p^2\Big(\sum_{k=1}^{n_t} ||y_k^{FE} - y_k^l||_H^2 + ||y_0^{FE} - y_0^l||_H^2\Big). \qquad (4.67)
\end{aligned}$$

Similarly, we can estimate the second sum in (4.52), using $||y_k^l||_V \leq c_y$:

$$\begin{aligned}
&\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \big(c_{A'}||y_k^l||_V + c_{l'}\big)^2||p_k^{i,FE} - p_k^{i,l}||_V^2 \leq ||S||_2\big(c_{A'}c_y + c_{l'}\big)^2 \sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} ||p_k^{i,FE} - p_k^{i,l}||_H^2 \\
&= ||S||_2\big(c_{A'}c_y + c_{l'}\big)^2 \sum_{i=1}^{D}\Big(\sum_{k=k_i-1}^{k_{i-1}} ||p_k^{i,FE} - p_k^{i,l}||_H^2 + ||p_{k_i}^{i,FE} - p_{k_i}^{i,l}||_H^2\Big). \qquad (4.68)
\end{aligned}$$

Combining (4.67) with (4.55), we get:

$$\begin{aligned}
\frac{c}{n_t}\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} c_{A'}^2||p_k^{i,FE}||_V^2||y_k^{FE} - y_k^l||_V^2 &\leq c2c_{A'}^2||S||_2 c_p^2\Big(\frac{1}{\eta_S}\big(5n_t + 5\hat{C}_0 n_t + \hat{C}_1||S||_2\big)\Big) \sum_{j=l+1}^{r} \lambda_j \\
&=: k_S(\frac{1}{\eta_S}, c_p, ||S||_2, n_t) \sum_{j=l+1}^{r} \lambda_j.
\end{aligned}$$

And analogously, (4.68) together with (4.56)-(4.59) yields:

$$\frac{c}{n_t}\sum_{i=1}^{D}\sum_{k=k_{i-1}}^{k_i} \big(c_{A'}||y_k^l||_V + c_{l'}\big)^2||p_k^{i,FE} - p_k^{i,l}||_V^2 \leq k_A(\frac{1}{\eta_A}, \frac{1}{\eta_S}, c_y, ||S||_2, n_t) \sum_{j=l+1}^{r} \lambda_j$$

with appropriate constants k_S, k_A, amongst others depending on the weightings η_S, η_A, on $||S||_2^2$, n_t and the bounds c_p, c_y for the adjoint and state solution, respectively. Thus,

$$\frac{\left|f'_{FE}(u)\delta u - f'_l(u)\delta u\right|^2}{||\delta u||_U^2} \leq \Big(k_S(\frac{1}{\eta_S}, c_p, ||S||_2, n_t) + k_A(\frac{1}{\eta_A}, \frac{1}{\eta_S}, c_y, ||S||_2, n_t)\Big) \sum_{j=l+1}^{r} \lambda_j,$$

which completes the proof. □

Let us comment on some issues concerning the constants, k_S, k_A, arising in (4.66). The dependence on c_p and c_y is stressed here because – at least in our application – the order of magnitude of these two values influences the order of magnitude of the two constants, k_S, k_A, in (4.66). Thus, – as a rule of thumb – if c_p is large compared to c_y, we may choose η_S larger than η_A in a similar way, and vice versa. However, changing the ratio of η_S to η_A also affects the eigenvalues λ_j.

We further point out that the adjoint error is strongly influenced by the state error, and the constant k_A for the adjoint part also involves the weighting factor η_S.

4.3 Numerical Results

This section shows some numerical results confirming the theoretical statements we have seen previously. It is divided into two subsections. Firstly, we study reduced order models for our partial integro-differential equation (cf. chapter 3) including a brief description of how to get the reduced order model. And secondly, we show numerical results concerning the optimal control problem defined in section 3.3.

4.3.1 Partial Integro-Differential Equation

We denote by $\{\Phi_j\}_{j=1}^{n_x}$ the finite element basis, i.e. $H^{n_x} := \text{span}(\Phi_1, \dots, \Phi_{n_x})$, and by $M \in \mathbb{R}^{n_x \times n_x}$ the corresponding mass matrix. On our way to a POD basis, we have snapshots $s_1, \dots, s_n \in H^{n_x}$ given, e.g. from a finite element solution of our PIDE. Hence, we can write

$$s_j(x) = \sum_{k=1}^{n_x} S_{kj} \Phi_k(x)$$

with a coefficient matrix $S \in \mathbb{R}^{n_x \times n}$.

We define the matrix $Y = M^{\frac{1}{2}} S D^{\frac{1}{2}} \in \mathbb{R}^{n_x \times n}$ with weighting matrix $D = diag(\gamma_1, \dots, \gamma_n)$, and can now calculate the matrix-vector product $Y^T Y \in \mathbb{R}^{n \times n}$ in the sense of lemma 4.1.12. Using a solver for eigenvalue problems, we find the l largest eigenvalues $\lambda_1, \dots, \lambda_l$ and corresponding orthonormal [2] eigenvectors $v_1, \dots, v_l \in \mathbb{R}^{n_x}$ to this matrix, i.e.

$$Y^T Y v_j = \lambda_j v_j, \qquad j = 1, \dots, l. \tag{4.69}$$

Using (4.8), we now compute the coefficient matrix $P \in \mathbb{R}^{n_x \times l}$ with

$$P_{\cdot,j} = \frac{1}{\sqrt{\lambda_j}} S D^{\frac{1}{2}} v_j$$

[2] Orthonormal in the sense of $\mathbb{R}^{n_x}$ with the common Euclidean scalar product

such that the POD basis functions are given by

$$\Psi_j(x) = \sum_{k=1}^{n_x} P_{kj}\Phi_k(x). \tag{4.70}$$

The eigenvalue problem (4.69) is equivalent to a singular value decomposition, for which reason POD belongs to the SVD based model reduction techniques. Since we are only interested in the l largest eigenvalues, the problem can be solved by an iterative method. In Matlab, the function 'eigs' uses an Arnoldi-Lanczos method where l can be passed as an additional input parameter. For further details on the POD eigenvalue problem we refer to, e.g., Fahl (2000).

Given the POD basis functions, (4.70), they can be used as test and trial functions in a Galerkin ansatz as described in section 4.2.1. To get a numerical solution for the corresponding reduced order problem, we need to compute the system matrices, M_l, $A_l(T) \in \mathbb{R}^{l\times l}$, and vectors, B_l, $F_l \in \mathbb{R}^l$. It is easy to show that they can be deduced from the finite element matrices and vectors:

$$M_l = P^T M P, \quad A_l(u;T) = P^T A(u;T)P, \quad F_l(u;T) = P^T F(u;T), \quad B_l = P^T B. \tag{4.71}$$

In the notation above, we stressed the time- and parameter-dependence of the matrices and vectors. This implies a great challenge for the reduced order models since it seems to be not very efficient to calculate, for instance, the matrix $A_l(u;T)$ in each time step. In literature, a so-called affine time- and parameter-dependence would be desirable as discussed in Grepl and Patera (2005). Here, the matrices and vectors or bilinear and linear operators, respectively, can be written as an affine combination of time- and parameter-independent operators. If this structure is not available, an empirical interpolation method can be used (cf. Barrault et al. (2004), Chaturantabut and Sorensen (2011)), where nonlinear terms are approximated by affine combinations of time- and parameter-independent operators.

We will see later that the POD basis changes during an optimization. Thus, we cannot split the computational effort in 'online'- and 'offline'-stages as in a typical reduced basis method, and this affine parameter dependence gains importance.

Let us take a look at the stiffness matrix $A(u;T)$. According to (3.23), the stiffness matrix can be split in a sparse and a Toeplitz part, $A(u;T) = A^{NI}(u;T)+A^I(u)$. The dependence of $A^I(u)$ on the parameters is usually highly nonlinear, however, we have to calculate $P^T A^I(u)P$ only once for each parameter setting. This is not possible for the sparse part, $A^{NI}(u;T)$, because the volatility is space-dependent and thus destroys the affine time-dependence except for the case of a separable volatility function, i.e. $\sigma^2(T,x) = \chi(T)\xi(x)$.

Local volatility functions that are piecewise constant in time unfortunately do not satisfy the theoretical assumption of Lipschitz continuity in time, however, they are very popular in practice. Given I different time buckets, the time-dependent part of the reduced stiffness matrix has to be computed only I times. We will see that this approach leads to acceptable computing times at the end of this section.

Again, the following numerical results are based on the Merton jump-diffusion model with

#POD	incl. diff.-quot.		excl. diff.-quot.	
l	ERR_1	$\sum_{k=l+1}^{r}\lambda_k$	ERR_1	$\sum_{k=l+1}^{r}\lambda_k$
5	1.03e-3	5.93e-3	5.13e-5	4.01e-5
6	2.31e-4	1.55e-3	1.09e-5	7.22e-6
7	5.00e-5	3.85e-4	3.62e-6	1.57e-6
8	1.08e-5	9.22e-5	2.52e-6	3.61e-7
9	3.10e-6	2.14e-5	3.52e-6	8.03e-8
10	2.55e-6	4.81e-6	5.61e-6	1.71e-8
11	4.04e-6	1.06e-6	8.41e-6	3.51e-9
12	6.57e-6	2.28e-7	1.13e-5	7.03e-10
13	9.66e-6	4.78e-8	1.36e-5	1.38e-10
14	1.25e-5	9.84e-9	1.47e-5	2.64e-11
15	1.44e-5	1.98e-9	1.45e-5	5.07e-12

Table 4.2: Error $ERR_1 = \frac{1}{n}\sum_{i=1}^{n}\left|\left|y(t_i)-y_i^{l,1}\right|\right|_H^2$ between closed-form- and POD solution of the PIDE (using Crank-Nicolson) and the corresponding sum of remaining eigenvalues for different numbers of POD basis functions, l, including and excluding the difference quotients, resp.

the following parameter setting:

$$\underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 5\ y,\ r \equiv 3\%,\ \Delta T = 0.0125,\ \Delta x = 0.0025, \qquad (4.72)$$
$$\sigma \equiv 30\%,\ \lambda = 50\%,\ \mu_J = 0\%,\ \sigma_J = 50\%.$$

Table 4.2 shows results corresponding to theorem 4.2.6. We used the closed-form solution given in terms of an infinite series to compute snapshots $y(t_0), \ldots, y(t_{n_t}) \in H^{n_x}$ [3] . According to theorem 4.2.6, these are now used to find a POD basis, on the one hand, including the difference quotients (columns 2-3) and, on the other hand, without the difference quotients (columns 4-5). The reduced order solution is then calculated with a Crank-Nicolson time discretization. The table shows the error $ERR_1 = \frac{1}{n}\sum_{i=1}^{n}\left|\left|y(t_i) - y_i^{l,1}\right|\right|_H^2$ and the corresponding sum of remaining eigenvalues $\sum_{k=l+1}^{r}\lambda_k$ for an increasing number of POD basis functions l. First note that in our application the inclusion of difference quotients has minimal influence on the results. We further observe a fast decay of the eigenvalues and also a fast decay of the error. However, the error decay stops at a value of about 1.0e-5. This effect can easily be explained with the summand $C_1\Delta t^4$ occurring in (4.31) that can not be reduced by using more POD basis functions.

This latter effect drops out in the next table 4.3. Here, we take a look at error $ERR_2 = \frac{1}{n}\sum_{i=1}^{n}\left|\left|y_i^{FE} - y_i^{l,2}\right|\right|_H^2$ (cf. theorem 4.2.8). The snaphsots, y_i^{FE}, are now based on a discretized solution in which we use Crank-Nicolson for the time discretization. The corresponding reduced differential equation is solved with a Crank-Nicolson scheme, too. As expected, ERR_2 decays for all l. We further observe that including difference quotients even leads to slightly worse results.

[3] Note that we make two errors in the closed-form solution, though they are insignificant in the total error. First, the infinite sum is truncated, and second, the snapshots are calculated for a finite number of space steps

#POD	incl. diff.-quot.		excl. diff.-quot.	
l	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$
5	1.87e-3	1.16e-2	5.13e-5	4.04e-5
6	7.81e-4	5.24e-3	1.07e-5	7.56e-6
7	3.44e-4	2.48e-3	3.03e-6	1.90e-6
8	1.54e-4	1.19e-3	1.13e-6	6.31e-7
9	6.88e-5	5.66e-4	5.21e-7	2.57e-7
10	3.05e-5	2.67e-4	2.47e-7	1.15e-7
11	1.34e-5	1.25e-4	1.11e-7	5.26e-8
12	5.85e-6	5.81e-5	5.08e-8	2.40e-8
13	2.52e-6	2.68e-5	2.31e-8	1.09e-8
14	1.08e-6	1.23e-5	1.03e-8	4.90e-9
15	4.55e-7	5.54e-6	4.61e-9	2.19e-9

Table 4.3: Error $ERR_2 = \frac{1}{n}\sum_{i=1}^{n} ||y_i^{FE} - y_i^{l,2}||_H^2$ between FE- and POD solution of the PIDE (using Crank-Nicolson) and the corresponding sum of remaining eigenvalues for different numbers of POD basis functions, l, including and excluding the difference quotients, resp.

We have seen in section 3.2.4 that the Rannacher smoothing – i.e. a Crank-Nicolson method where non-smooth initial conditions are smoothed by a few implicit Euler steps – leads to a smoother solution. This finding also influences a POD model as it can be seen in table 4.4. The table has the same structure as the preceding one, however, the finite element solution is not calculated with a Crank-Nicolson time discretization but we applied Rannacher smoothing. We stress that the same technique is then also used for the time discretization of the reduced problem.

The most significant observation is a faster decay of the eigenvalues (for both, including or excluding the difference quotients) what is due to the smoother solution in time direction. This immediately leads to a faster decay of the error.

An important point that has not been mentioned until now is the strong influence of a convection term on the decay rate of the eigenvalues. For this reason, we have introduced the PIDE (2.14) in addition to (2.13). To illustrate the behavior of the eigenvalues and the POD approximation in case of strong convection, we change the variable setting (4.72) to a more extreme case and set

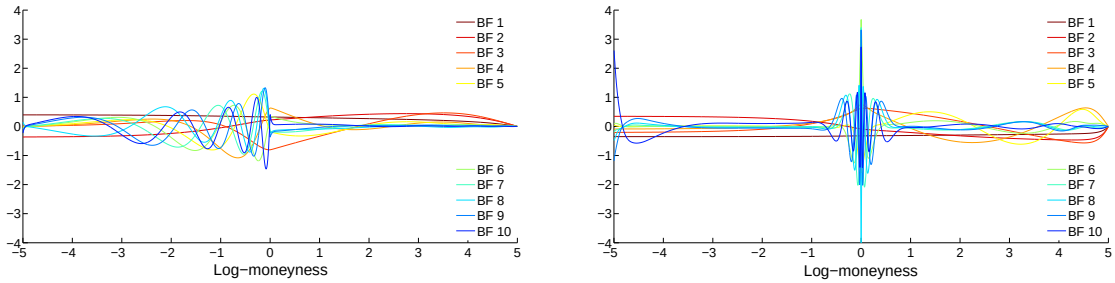
$$\sigma \equiv 30\%,\ \lambda = 100\%,\ \mu_J = 50\%,\ \sigma_J = 100\%.$$

Thus, we define $c := r + \frac{\sigma^2}{2} - \lambda\zeta \approx 1.6433$ in the sense of (2.14) to eliminate the convection, and present the numerical results in table 4.5. We restrict ourselves to the case of not including the difference quotients into the POD basis since this in general has led to better results. Obviously, if the convection is strong, then the eigenvalues decay much slower (see column 3) and, accordingly, the error ERR_2 decays slower (column 2) as in the transformed case (columns 4 - 5). For instance, we need to take 15 POD basis functions instead of only 7 to get a similar error.

Figure 4.1 now shows the first ten POD basis functions for the PIDE with strong con-

#POD	incl. diff.-quot.		excl. diff.-quot.	
l	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$
5	9.85e-04	5.66e-03	5.19e-05	4.02e-05
6	2.27e-04	1.51e-03	1.09e-05	7.25e-06
7	5.27e-05	4.00e-04	2.82e-06	1.56e-06
8	1.24e-05	1.06e-04	7.23e-07	3.56e-07
9	2.93e-06	2.84e-05	1.73e-07	8.11e-08
10	7.01e-07	7.68e-06	4.12e-08	1.85e-08
11	1.71e-07	2.11e-06	1.05e-08	4.29e-09
12	4.24e-08	5.85e-07	2.99e-09	1.02e-09
13	1.07e-08	1.65e-07	9.97e-10	2.47e-10
14	2.78e-09	4.70e-08	3.96e-10	6.13e-11
15	7.46e-10	1.36e-08	1.84e-10	1.57e-11

Table 4.4: Error $ERR_2 = \frac{1}{n}\sum_{i=1}^{n} \left|\left|y_i^{FE} - y_i^{l,2}\right|\right|_H^2$ between FE- and POD solution of the PIDE (using Rannacher time stepping) and the corresponding sum of remaining eigenvalues for different numbers of POD basis functions, l, including and excluding the difference quotients, resp.



(a) Basis functions with convection **(b) Basis functions without convection**

Figure 4.1: Ten sample POD basis functions for the PIDE with and without convection, resp.

vection, 4.1(a), and with removed convection, 4.1(b). Here, single functions do not provide much information. However, it is more interesting to take a look at all functions at a glance. We observe that in case (b), the main activity of the basis functions concentrates on the origin, where the kink in the initial condition is smoothed out. But given a strong convection, the activity seems to drift to the left providing less information in the region of interest around zero.

Beside the results concerning the accuracy of the POD method for the partial integro-differential equation that we have shown above, we are of course interested in the benefit of a reduced computing time. Table 4.6 shows a first result.

Let us first describe the structure of the table. The first two columns show the discretization of the finite element solution. For instance, to show the effect of a refined space mesh, we keep $\Delta T = 0.025$ fixed in the first three lines and divide the step size in space, Δx, in half. The next three lines then show the same results, however, now the step size ΔT is

#POD	with convection		no convection	
l	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$	ERR_2	$\sum_{k=l+1}^{r} \lambda_k$
5	3.67e-03	3.15e-03	2.34e-05	2.00e-05
6	1.31e-03	1.02e-03	3.69e-06	2.26e-06
7	4.54e-04	3.47e-04	9.43e-07	3.61e-07
8	1.68e-04	1.29e-04	3.10e-07	6.58e-08
9	7.11e-05	5.08e-05	1.18e-07	1.50e-08
10	2.98e-05	2.06e-05	2.37e-08	4.61e-09
11	1.43e-05	8.71e-06	3.84e-09	1.29e-09
12	7.33e-06	3.76e-06	1.01e-09	2.84e-10
13	4.15e-06	1.67e-06	3.43e-10	6.19e-11
14	2.53e-06	7.57e-07	1.44e-10	1.39e-11
15	1.65e-06	3.58e-07	1.44e-10	3.28e-12

Table 4.5: Error ERR_2 between FE- and POD solution and the corresponding sum of remaining eigenvalues for different numbers of POD basis functions, l, for the PIDE (2.13) with convection term and the transformed PIDE (2.14) without convection

divided by two, and so on.

Changing the discretization, we are interested in the corresponding computing times. The table contains two blocks. The columns three to six show the computing time using constant parameters r, λ, μ_J, σ_J, σ. We see the time for the solution of one finite element problem in column three, and column six shows the time needed for the calculation of the appropriate POD basis of rank $l = 10$. After having calculated the basis, the computation of the corresponding reduced PIDE can be split in the construction of the reduced system matrices and vectors (column 5) and the solution of the reduced linear systems of equations (column 4). The second block (column 7-10) contains the same timings except that we now assume the volatility to be only piecewise constant in time with ten time buckets. Thus, we have to set up the volatility dependent part of the stiffness matrix ten times instead of only once.

Concentrating on the first block of timings, we again observe a linear time dependence of the finite element solution on the discretization in time and space, respectively (cf. also section 3.2.4). However, the solution of the reduced linear systems of equations is independent of the space discretization since the linear systems of equations have a fixed size of 10×10 in each time step. Since POD only reduces the space dimension, a refinement of the time grid linearly increases the computing time (e.g. $0.008 \rightarrow 0.015 \rightarrow 0.031$). With regard to the construction of the reduced system matrices and vectors, the timings show a dependence on space and time grid as expected from (4.71). Since we assume the parameters to be constant, the mass matrix and the stiffness matrix have to be calculated only once, but the right-hand side vector is time-dependent due to (3.15). However, we can make use of an affine time-dependence for this vector and thus, the dependence on the time discretization is clearly less than linear.

On the other hand, the time needed for the solution of the eigenvalue problem to get the POD basis from the finite element snapshots (column 6) implies a linear dependence on time

discretization		constant parameters FEM (sec.)	POD (sec.)			10 time buckets FEM (sec.)	POD (sec.)		
ΔT	Δx	total	LSE	system	basis	total	LSE	system	basis
0.025	0.005	0.749	0.008	0.017	0.033	0.835	0.008	0.051	0.030
	0.0025	1.683	0.008	0.021	0.040	1.633	0.009	0.087	0.040
	0.00125	3.387	0.008	0.043	0.113	3.531	0.009	0.173	0.080
0.0125	0.005	1.485	0.015	0.021	0.063	1.583	0.016	0.057	0.060
	0.0025	2.985	0.015	0.030	0.120	3.142	0.016	0.094	0.123
	0.00125	6.472	0.015	0.045	0.211	6.707	0.016	0.182	0.237
0.00625	0.005	2.907	0.031	0.033	0.197	2.916	0.032	0.070	0.195
	0.0025	5.992	0.031	0.042	0.355	6.082	0.032	0.108	0.359
	0.00125	12.685	0.031	0.060	0.724	13.319	0.035	0.232	0.700

Table 4.6: Computing times (in sec.) of the FE- and POD solution for several discretizations: overall time for FEM, solving the POD linear systems of equations (LSE), building the POD system matrices and vectors, computation of the POD basis for constant and piecewise constant parameters

and space discretization as well and is the most striking factor within the solution of the reduced order problem. Though the eigenvalue problem itself has dimension n_T, the matrix is constructed via matrix-matrix products with size $n_T \times n_x$.

The second block of the table (columns 7-10) shows slightly increasing finite element timings since the stiffness matrix has to be calculated ten times now. However, relatively, this increase is insignificant compared to the increase of the construction of the reduced system matrices. We stress that the time does not increase by a factor of ten since we again can make use of a partly affine parameter dependence. Because the Toeplitz part of the stiffness matrix is independent of time, this time-consuming task has to be done only once. This clearly shows that an efficient implementation is needed to get a competitive reduced order model. For the sake of completeness, we mention that the timings for solving the LSEs and for constructing the POD basis do almost not change compared to the first block.

We have seen that POD is well-suited for the solution of the PIDE in terms of accuracy and computing time, thus, we turn to the use of model order reduction in optimal control problems.

4.3.2 Optimal Control Problem

We now want to illustrate the statements of section 4.2.2. As a numerical example, we again use Merton's jump-diffusion model with market data as defined in table 3.9. The general setting is as follows:

$$\begin{aligned} \underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 5\ y,\ r \equiv 5\%,\ \alpha = 0, \\ u = (\lambda, \mu_J, \sigma_J, \sigma^2) = (50\%, 0\%, 50\%, 30\%^2) \in \mathbb{R}^4. \end{aligned} \tag{4.73}$$

Until further notice, the parameter vector u is fixed, i.e. it is used for calculating snapshots, POD basis functions and POD approximations.

#POD	basis B_S		basis B_{SA}		basis B_{SwA}	
l	$\epsilon_f^{rel}(u)$	$\sum_{k=l+1}^{r} \lambda_k$	$\epsilon_f^{rel}(u)$	$\sum_{k=l+1}^{r} \lambda_k$	$\epsilon_f^{rel}(u)$	$\sum_{k=l+1}^{r} \lambda_k$
5	3.90e-5	4.60e-5	1.49e-2	7.05e+0	2.41e-3	3.37e-2
6	3.66e-5	8.20e-6	2.86e-3	4.35e+0	6.08e-4	1.20e-2
7	8.62e-6	1.76e-6	3.00e-3	2.48e+0	3.15e-4	7.30e-3
8	4.64e-6	4.04e-7	7.46e-3	1.45e+0	2.18e-4	4.33e-3
9	4.72e-6	9.27e-8	8.16e-4	8.25e-1	4.30e-5	2.14e-3
10	2.73e-7	2.13e-8	2.70e-4	3.57e-1	9.39e-5	1.08e-3
11	5.70e-7	4.95e-9	3.48e-4	2.49e-1	4.62e-5	5.27e-4
12	9.27e-9	1.18e-9	2.37e-4	1.15e-1	1.61e-5	3.74e-4
13	2.39e-7	2.87e-10	4.18e-5	6.53e-2	3.55e-5	2.50e-4
14	6.87e-8	7.16e-11	1.65e-6	4.32e-2	1.41e-5	1.60e-4
15	6.39e-8	1.82e-11	6.69e-5	3.00e-2	2.52e-6	1.07e-4

Table 4.7: Relative error $\epsilon_f^{rel}(u)$ between the objective function based on FE- and POD solution, resp., and the corresponding sum of remaining eigenvalues for different numbers of POD basis functions l, where we use three different POD bases

Of course, including adjoint snapshots into the POD basis as proposed above has a remarkable effect on the approximation quality. To show this, we define three different POD bases depending on different choices of the weights η_S and η_A for a fixed parameter vector u:

$$\begin{aligned} B_S &: \eta_S = 1,\ \eta_A = 0 \text{ (only state snapshots)}, \\ B_{SA} &: \eta_S = 1,\ \eta_A = 1 \text{ (state and adjoint snapshots)}, \\ B_{SwA} &: \eta_S = 1,\ \eta_A = 0.001 \text{ (state and weighted adjoint snapshots)}. \end{aligned} \tag{4.74}$$

We expect a negative effect on the function values for unchanged parameter vector u if η_A is increased. And this is clearly observable in table 4.7. More precisely, the smaller the weight η_A (0, 0.001, 1), the smaller the relative function error

$$\epsilon_f^{rel}(u) = \frac{|f_{FE}(u) - f_l(u)|}{|f_{FE}(u)|}.$$

The table also shows the decreasing error for an increasing number of POD basis functions l and the corresponding decreasing sum over the remaining eigenvalues.

We observe that including the adjoint snapshots leads to a significant slower decrease of the eigenvalues. This effect is explainable with regard to a typical adjoint solution (cf. figure 4.3(b)), which is – at least in our application – not as smooth as the state solution (cf. figure 4.3(a)) due to the point-wise observations at different time instances. However, using a few more basis functions, the inclusion of adjoint snapshots still leads to an acceptable error, especially when they have only a minor weighting. Anyhow, the real advantage of using adjoint snapshots gets clear regarding the gradient approximation via POD.

#POD l	basis B_S $\epsilon_g^{rel}(u)$	basis B_{SA} $\epsilon_g^{rel}(u)$	basis B_{SwA} $\epsilon_g^{rel}(u)$
5	3.76e-2	5.22e-2	2.57e-2
6	3.68e-2	1.32e-2	1.62e-2
7	3.61e-2	1.57e-2	6.50e-3
8	3.60e-2	1.20e-2	3.45e-3
9	3.58e-2	2.41e-2	1.88e-3
10	3.59e-2	2.25e-3	1.79e-3
11	3.60e-2	1.66e-3	1.27e-3
12	3.60e-2	9.89e-4	2.66e-4
13	3.59e-2	4.83e-4	8.84e-5
14	3.60e-2	5.39e-4	1.61e-4
15	3.60e-2	1.90e-4	1.42e-4

Table 4.8: Relative error $\epsilon_g^{rel}(u)$ between the gradient (via adjoints) based on FE- and POD solution, resp., for different numbers of POD basis functions l, where we use three different POD bases

Table 4.8 now shows the relative gradient error,

$$\epsilon_g^{rel}(u) = \frac{||\nabla f_{FE}(u) - \nabla f_l(u)||_2}{||\nabla f_{FE}(u)||_2},$$

where we use the adjoint approach to calculate the gradient. To be precise, we first calculate the finite element state and adjoint solution and the corresponding gradient, $\nabla f_{FE}(u)$, according to (3.43). Using this state and adjoint solution, we calculate three POD bases with the weightings as defined in (4.74) and can then compute the reduced order POD solutions for the state and the adjoint equations based on these three bases. The gradient, $\nabla f_l(u)$, is given by (4.47).

We omit the sum of remaining eigenvalues since they are already given in table 4.7. In the second column of table 4.8 we observe a relative gradient error of about 3.6%, which cannot be improved by adding more basis functions. One may have expected an even worse result since we use a POD basis, B_S, including only information of the state solution, to approximate the adjoint equation. Including adjoint snapshots leads to significantly better results. In particular, we see a generally decreasing relative error for increasing l. It is also observable that in our application a lower weighting of the adjoint snapshots leads to slightly better results.

In numerical mathematics, another way to approximate a gradient – avoiding the use of adjoints – are finite differences. Here, we only use solutions of the state equation to compute the objective function value at u and $u + \Delta e_i$, $i = 1, \ldots, 4$, with unit vectors $e_i \in \mathbb{R}^4$. We choose $\Delta =$ 1.0e-7 and stress that the POD bases, which are based on the parameter vector u, are not changed. Table 4.9 again shows the relative gradient error as in the table above except that we use finite differences instead of adjoints to compute the gradients. The table looks quite similar to table 4.8. Although we do not compute reduced adjoint equations, the inclusion of adjoint snapshots leads to significantly better results. And again, by omitting adjoint snapshots in the second column, the accuracy can not be improved by adding basis

#POD l	basis B_S $\epsilon_g^{rel}(u)$	basis B_{SA} $\epsilon_g^{rel}(u)$	basis B_{SwA} $\epsilon_g^{rel}(u)$
5	3.02e-2	4.96e-2	2.03e-2
6	2.93e-2	1.24e-2	1.12e-2
7	2.85e-2	1.49e-2	6.63e-3
8	2.84e-2	1.11e-2	3.27e-3
9	2.83e-2	2.19e-3	1.83e-3
10	2.84e-2	1.85e-3	1.43e-3
11	2.84e-2	1.61e-3	6.45e-4
12	2.84e-2	9.48e-4	1.94e-4
13	2.84e-2	4.10e-4	9.97e-5
14	2.84e-2	4.43e-4	9.97e-5
15	2.84e-2	1.79e-4	1.72e-4

Table 4.9: Relative error $\epsilon_g^{rel}(u)$ between the gradient (via finite differences) based on FE- and POD solution, resp., for different numbers of POD basis functions l, where we use three different POD bases

step size Δ	basis B_S $\epsilon_f^{rel}(u_\Delta)$	basis B_{SA} $\epsilon_f^{rel}(u_\Delta)$	basis B_{SwA} $\epsilon_f^{rel}(u_\Delta)$
1.0e-5	1.90e-6	6.69e-5	2.54e-6
1.0e-4	1.96e-5	6.69e-5	2.65e-6
1.0e-3	1.96e-4	6.74e-5	3.77e-6
1.0e-2	1.92e-3	6.26e-5	1.33e-5
1.0e-1	1.35e-2	4.36e-3	7.75e-4

Table 4.10: Relative error $\epsilon_f^{rel}(u_\Delta)$ between the objective function based on FE- and POD solution, resp., for varying step size Δ and corresponding control $u_\Delta = u - \Delta \frac{\nabla f(u)}{||\nabla f(u)||_2}$; for three different POD bases with $l = 15$ fixed

functions.

Except for the difference quotients in the last table, we have studied the errors for a fixed control u, where the POD bases are also based on this control. We now want to show that adjoint snapshots also have a positive effect on the function values when we veer away from the start control u without changing the POD model.

The results are shown in table 4.10 and further illustrated in figure 4.2. The idea here is as follows. During a numerical optimization we are usually interested in taking steps in a descent direction. Thus, after having built the bases of fixed rank $l = 15$ according to (4.74) for the control u, we calculate the corresponding gradient and make a step with step size Δ in steepest descent direction,

$$u_\Delta = u - \Delta \frac{\nabla f(u)}{||\nabla f(u)||_2}. \tag{4.75}$$

Then, we compute the relative function value error $\epsilon_f^{rel}(u_\Delta)$, where the POD models are still based on the control u and not u_Δ. What we observe is a better approximation of the basis

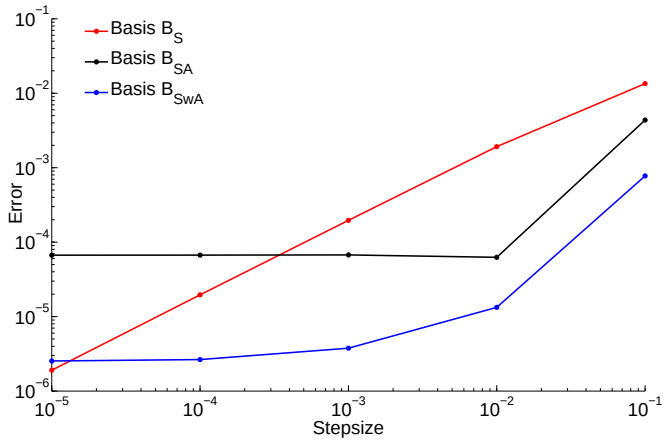


Figure 4.2: Relative error $\epsilon_f^{rel}(u_\Delta)$ between the objective function based on FE- and POD solution, resp., for varying step size Δ and corresponding control $u_\Delta = u - \Delta \frac{\nabla f(u)}{||\nabla f(u)||_2}$; for three different POD bases with $l = 15$ fixed

B_S for very small step sizes Δ = 1.0e-5. However, including adjoint snaphots – especially with a suitable weighting – leads to significantly better results for larger step sizes. For example, setting Δ = 1.0e-2, the error for B_{SwA} is smaller by a factor of 100.

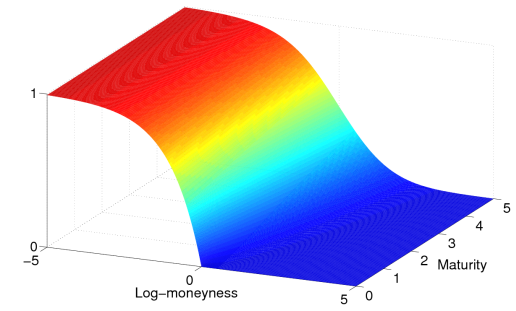
It is clear that the accuracy of the state solution is mainly responsible for the accuracy of the objective function value. Thus, we try to give an explanation for the findings above on the basis of a graphical analysis of the errors of the state solution in figure 4.3.

First, 4.3(a) and (b) show the finite element state and adjoint solution of the PIDE for the control u. Those solutions are used as snapshots for the basis computation where we keep $l = 15$ fixed. The adjoint solution clearly shows peaks at each point where market data is available to which we want to calibrate our option pricing model. Note that we are especially interested in the values of our state solution at these points.

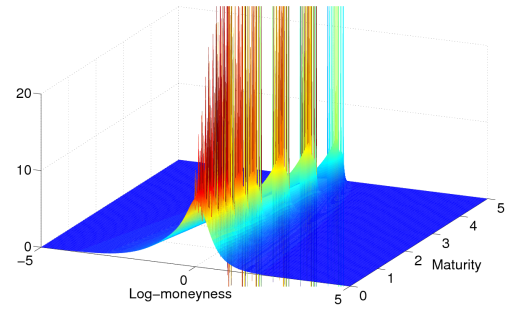
The pointwise difference between the full finite element state solution for the control u, y, and the corresponding reduced state solution based on the basis B_S, $y^{l,S}$ is illustrated in figure 4.3(c). Given the scaling of the graph – that is fixed for the remaining ones –, the error is negligible. (d) shows the same result, but now we use the basis B_{SwA} including adjoint snapshots to compute the POD approximation, $y^{l,SwA}$. As expected, this leads to a larger, observable error especially at the beginning for T close to zero.

However, if we now make a step in steepest descent direction with step size Δ = 1.0e-2 (in the sense of (4.75)), and do not update the POD model, the results are quite different. On the one hand, (e) illustrates a strongly increasing pointwise error compared to (c), unfortunately in a region of great interest. On the other hand, the basis B_{SwA} provides further information in this region, where market data are available and model values have to be fitted, leading to an error that is clearly smaller in figure (f) (cf. also table 4.10).

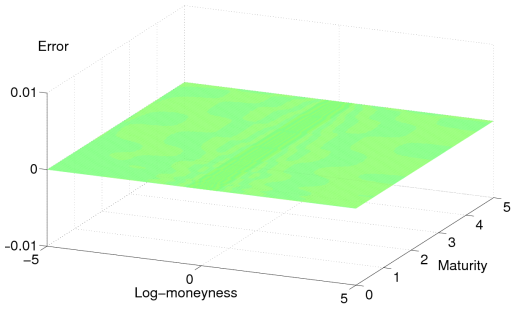
Summarizing the previous results, we have seen that the inclusion of adjoint snapshots in one combined basis with state snaphsots leads to a far better approximation of gradients even when finite differences are used. It further has a positive effect on the otherwise strong locality of a fixed POD basis. The following chapter will now make the next step on the



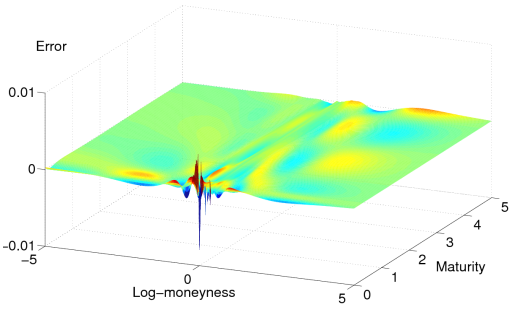
(a) State solution $y(u;\ T,x)$



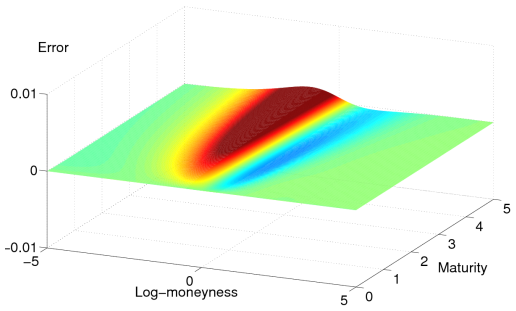
(b) Adjoint solution $p(u;\ T,x)$



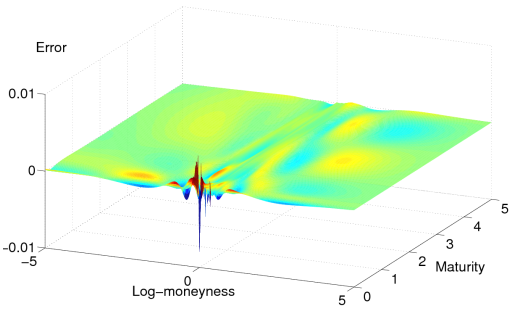
(c) Error $y(u;\ T,x) - y^{l,S}(u;\ T,x)$



(d) Error $y(u;\ T,x) - y^{l,SwA}(u;\ T,x)$



(e) Error $y(u_\Delta;\ T,x) - y^{l,S}(u_\Delta;\ T,x)$



(f) Error $y(u_\Delta;\ T,x) - y^{l,SwA}(u_\Delta;\ T,x)$

Figure 4.3: Influence of adjoint snapshots on the POD state error when the control u is changed and the basis not updated. Figure (c), (d): pointwise error at u; (e), (f): pointwise error at u_Δ ($\Delta = 1.0e-2$)

way to a global optimization technique by embedding the POD approach into a trust-region framework.

Chapter 5

Trust-Region POD

We have learned in the previous chapter that POD is only a local model. This means, if we base our POD model on a certain control, then error estimates only hold true for this control. However, in literature, there are several approaches known where POD is used in optimization, i.e. also for varying controls. In this context, we have already mentioned the work of Afanasiev and Hinze (2001), Ravindran (2002), Kunisch and Volkwein (2008), Hinze and Volkwein (2008) and Tröltzsch and Volkwein (2009) in the introduction to chapter 4.

A different approach has been proposed by Arian et al. (2000) where the local reduced order model is embedded into a trust-region framework. Using this intuitive setting, descent steps of the reduced model are only accepted if they provide an acceptable decrease of the true objective function, too. Further, we are able to review the reduced model in each iteration and adjust the trust region of controls accordingly. Convergence has been proven, but here strong assumptions – that are totally not clear a priori – have to be made. This is exactly the point where this thesis provides an improvement of the present state of research.

The chapter is organized as follows. We start by recalling the idea of trust-region methods in section 5.1. Thus, we first present the standard approach with a quadratic approximation of the objective function. Afterwards, we are interested in several generalizations of this common approach with the most important issue of inexact gradient information. To be able to still show a global convergence result, the inexact model gradient has to satisfy a certain error tolerance introduced by Carter (1991). In section 5.2, we define the adaptive trust-region POD algorithm and then show how the assumptions of the preceding section are fulfilled by a model function based on POD for a class of optimal control problems. The chapter closes with some numerical results confirming the efficiency of reduced order models in finance and the convergence of the proposed algorithm.

5.1 Trust-Region Methods

Let us first specify the problem to be considered. For this, let U be a real Hilbert space. Then we want to find the solution to

$$\min_{u \in U} f(u) \tag{5.1}$$

with $f : U \to \mathbb{R}$.

Given a current iterate $u_k \in U$, the idea of a 'trust-region algorithm' is to build a simple 'model function', m_k, that approximates the objective function, f, in a small region around

u_k. We refer to this region as 'trust region' $\mathcal{B}_k \subset U$ with

$$\mathcal{B}_k := \{u \in U : ||u_k - u|| \leq \Delta_k\}, \tag{5.2}$$

where $\Delta_k > 0$ is the 'trust radius' . Then, the function m_k is minimized approximately in this trust region by finding an appropriate descent direction. Mathematically, the trust-region 'subproblem' in iteration k can be formulated in the following way:

$$\min_{||s|| \leq \Delta_k} m_k(u_k + s).$$

In some sense, the concept is contrary to 'line-search methods' where we first determine a descent direction and then find an appropriate step size.

The next section will explain the basic concept of the standard trust-region approach. Here, the model function is a quadratic approximation of the objective function.

5.1.1 Quadratic Model Functions

Let us assume a sufficiently smooth objective function f. The best-known model function is a quadratic approximation, m_k^{quad}, of f:

$$m_k^{quad}(u_k + s) := f(u_k) + \langle g_k, s\rangle + \tfrac{1}{2}\langle s, H_k s\rangle. \tag{5.3}$$

Here, g_k denotes the gradient of the objective function, $\nabla f(u_k)$, and the self-adjoint operator H_k is an approximation (or equal) to the Hessian $\nabla^2 f(u_k)$.

Then, according to, e.g., Conn et al. (2000) or Fahl (2000), a basic trust-region algorithm can be defined as in algorithm 5.1.

Taking a closer look at this algorithm, it can be divided into four parts.

After the initialization, line 1 can be captioned by 'model definition'. In the case of a quadratic model function, this step consists of calculating the gradient and the Hessian approximation.

Line 2-3 is the 'step calculation'. This part is crucial since a sufficient decrease of the model function needs to be shown to get convergence results.

The 'quality of the trial step' is calculated in line 4-5 by comparing the predicted reduction,

$$pred_k(s_k) := m_k^{quad}(u_k) - m_k^{quad}(u_k + s_k),$$

of the model function with the actual reduction,

$$ared_k(s_k) := f(u_k) - f(u_k + s_k),$$

of the objective function. Note that it is necessary to compute the value $f(u_k + s_k)$ which might be an expensive task in many applications. However, the quotient ρ_k provides a very good measure for the capability of the model function.

This measure is now used in the lines 6 to 12 to decide whether the point $u_k + s_k$ is accepted as a new iterate or not. If ρ_k is greater than $\eta_1 > 0$, we see a decrease in the objective function that is at least a fraction of the predicted decrease. In those cases, the

Algorithm 5.1 Basic trust-region algorithm

Input: $\Delta_0 > 0$, $k = 0$, an initial control $u_0 \in U$ and constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \gamma_3$ satisfying

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3.$$

1: **compute** the model function $m_k^{quad}(u_k + s)$

2: **compute** an approximate solution $s_k \in U$ to

3: $$\min_{||s|| \leq \Delta_k} m_k^{quad}(u_k + s)$$

4: **compute** $f(u_k + s_k)$ **and**

5: $$\rho_k = \frac{f(u_k) - f(u_k + s_k)}{m_k^{quad}(u_k) - m_k^{quad}(u_k + s_k)}$$

6: **if** $\rho_k \geq \eta_2$ **then**

7: **set** $u_{k+1} = u_k + s_k$ **and** $\Delta_{k+1} \in [\Delta_k, \gamma_3 \Delta_k]$

8: **else if** $\eta_1 \leq \rho_k < \eta_2$ **then**

9: **set** $u_{k+1} = u_k + s_k$ **and** $\Delta_{k+1} \in [\gamma_2 \Delta_k, \Delta_k]$

10: **else if** $\rho_k < \eta_1$ **then**

11: **set** $u_{k+1} = u_k$ **and** $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$

12: **end if**

13: **set** $k \leftarrow k + 1$ **and** go to line 1

step s_k is accepted. If further $\rho_k > \eta_2$, a value typically chosen to be close to one, then the model seems to be a good approximation on the trust region and the trust radius may be increased (by the factor γ_3). Otherwise, the radius should be decreased (by the factor γ_2). A step s_k is rejected if $\rho_k < \eta_1$. In that case the model seems to be poor on the trust region and the radius, Δ_k, has to be decreased. This part of the algorithm may be captioned with 'acceptance of the trial step and update of the trust radius'.

The last line 13 implies the iterative structure of the algorithm. Of course, in practice, a stopping criterion that is not specified here would be implemented. A natural choice would be a sufficiently small norm of the gradient, $||\nabla f(u_k)||$, indicating a first-order critical point.

We return to the step calculation for a moment. To find an approximate minimizer, a step along the steepest descent direction seems to be promising. For quadratic model functions, the 'Cauchy point' provides the optimal step size within a given trust region. The following result can, e.g., be found in Conn et al. (2000).

Remark 5.1.1. ***(Cauchy point)***
Given a trust radius Δ_k and a quadratic model function, (5.3), with bounded second derivative such that $\beta_k := 1 + ||H_k||$. Then the Cauchy step $s_k^C = -\lambda_k^C g_k$, that is the unique minimizer

along the steepest descent direction, is determined by

$$\lambda_k^C = \begin{cases} \frac{||g_k||^2}{\langle g_k, H_k g_k\rangle}, & \text{if } \lambda_k^C ||g_k|| \leq \Delta_k \text{ and } \langle g_k, H_k g_k\rangle > 0 \\ \frac{\Delta_k}{||g_k||}, & else, \end{cases} \tag{5.4}$$

and the decrease can be estimated by

$$m_k^{quad}(u_k) - m_k^{quad}(u_k + s_k^C) \geq \frac{1}{2}||g_k|| \min\left\{\frac{||g_k||}{\beta_k}, \Delta_k\right\}. \tag{5.5}$$

The steepest descent method is known to perform poorly in many applications, thus, there are many improvements known in literature, e.g. the 'Dogleg' or 'Double-Dogleg' paths (cf. Conn et al. (2000)). However, as we will see below, it is sufficient from a theoretical point of view.

If the model function is not of quadratic type – and we will consider this case below –, an alternative approach is necessary to guarantee a sufficient decrease in the sense of (5.5).

We now make some assumptions on the objective function f to be able to state a convergence result for algorithm 5.1 according to Conn et al. (2000).

Theorem 5.1.2. ***(Strong global convergence)***
Let $f : U \to \mathbb{R}$ be twice continuously differentiable and bounded below with uniformly bounded Hessian. If the model function m_k^{quad} is given by (5.3) with bounded Hessian H_k and the trust-region subproblems are solved by (5.4), then one has

$$\lim_{k\to\infty} ||\nabla f(u_k)|| = 0.$$

Unfortunately, for our needs, most of the assumptions on the model setting are too restrictive. In the next section, we consider several generalizations. First, the model function is not necessarily of quadratic type (the notation will be m_k instead of m_k^{quad}). Further, the function value of the model function at the trust region center point is not equal to the function value of the objective function, i.e. in general $m_k(u_k) \neq f(u_k)$. The same holds true for the gradient, thus, $\nabla m_k(u_k) \neq \nabla f(u_k)$. In addition to this, only few information about the second derivative are available.

Nevertheless, we want to be able to state a convergence result similar to theorem 5.1.2.

5.1.2 Generalizations

To get an appropriate framework for a model function based on POD, several assumptions that have been made above have to be generalized. Thus, we now recall the main ingredients of the results presented in Fahl (2000) since we want to combine this with the error estimates of the last chapter in section 5.2. As it has been done in Fahl (2000), we restrict ourselves to the case of $U = \mathbb{R}^n$, but a generalization to a general Hilbert space should be straightforward.

In this framework, the objective function of our minimization problem, f, meets some requirements.

Assumption 5.1. ***(Objective function)***
Given $u_0 \in U$, the model function $f : U \to \mathbb{R}$ is Fréchet-differentiable on an open convex

set containing the level set of f at u_0, $\{u \in U : \ f(u) \leq f(u_0)\}$, with Lipschitz continuous gradient ∇f. Further, f is bounded below.

Non-quadratic Model Functions

A first generalization compared to the standard trust-region approach described above is a non-quadratic appearance of the model function, m_k. We can no longer use the Cauchy point as proposed above to solve the trust-region subproblem. Fortunately, Toint (1988) has proposed a more general algorithm to derive a step that fulfills a certain sufficient decrease condition in the sense of (5.5).

Prior to that, we have to introduce some notations and make a few assumptions on the model function.

Assumption 5.2. ***(Model function)***
The model function m_k with trust region center point u_k and trust radius Δ_k is Fréchet-differentiable on an open convex set containing the trust region $\mathcal{B}_k$. Further, we set $g_k := \nabla m_k(u_k)$.

The following algorithm 5.2 can be found in Fahl (2000) and is a slightly modified version of the corresponding algorithm proposed in Toint (1988). The first part of the algorithm

Algorithm 5.2 Step determination algorithm

Input: $\Delta_k > 0$, $u_k \in U$ and constants that satisfy

$$0 < \alpha \leq \beta < 1, \quad 0 < \mu \leq 1, \quad 0 < \nu_1 < 1, \quad \nu_2, \nu_3 > 0$$

1: **compute** λ_k^A such that

2: $$m_k(u_k - \lambda_k^A g_k) \leq m_k(u_k) - \alpha \lambda_k^A ||g_k||^2$$

3: $$||\lambda_k^A g_k|| \leq \Delta_k$$

4: **and** $\lambda_k^A \geq \min\{\nu_1 \Delta_k / ||g_k||, \nu_2\}$ **or** $\lambda_k^A \geq \lambda_k^B$

5: **where** $\lambda_k^B > 0$ (if required) has to satisfy

6: $$m_k(u_k - \lambda_k^B g_k) \geq m_k(u_k) - \beta \lambda_k^B ||g_k||^2$$

7: **if** $\Delta_k \geq \nu_3$ **then**

8: **compute** step s_k such that

9: $$m_k(u_k) - m_k(u_k + s_k) \geq \mu\big(m_k(u_k) - m_k(u_k - \lambda_k^A g_k)\big)$$

10: $$||s_k|| \leq \Delta_k$$

11: **end if**

(lines 1-6) is the search for a generalized Cauchy point. This means, we want to find an appropriate step length within the trust region to get a sufficient decrease in steepest descent direction implied by the Armijo-like condition in line 2. The boundedness away from zero is guaranteed by line 4 and the Goldstein-type condition in line 6, respectively.

It is known from quadratic model functions that the Cauchy point is only taken as a first initial step which is tried to be improved by more advanced methods. The second part of the step determination algorithm (lines 7-11) can be interpreted similarly. Here, for a sufficiently large trust radius, Δ_k, we may leave the steepest descent direction to find another direction, s_k, possibly leading to a greater decrease.

Note that the step determination algorithm produces feasible steps, i.e. the conditions are not incompatible. The following result originally proven by Toint (1988) can also be found in Fahl (2000, lemma 5.9).

Lemma 5.1.3. *Algorithm 5.2 always provides a feasible step s_k.*

A Curvature Measure for Non-quadratic Model Functions

The curvature of the model also plays an important role for the sufficient decrease. This can be seen by regarding the Cauchy decrease, (5.5), where the norm of the second derivative, H_k, is involved. If the Hessian itself and its norm, respectively, are not available, we need another criterion.

The following concept is based on Toint (1988).

Definition 5.1.4. ***(Curvature measure)***
Let $h : U \to \mathbb{R}$ be Fréchet-differentiable. Then

$$\omega(h, u, s) := \frac{2}{||s||^2}(h(u+s) - h(u) - \langle \nabla h(u), s\rangle) \tag{5.6}$$

is defined as a measure for the curvature of h along the step s and based at the point u.

For instance, if we assume the function h to have Lipschitz continuous derivatives with Lipschitz constant L, we can easily estimate (cf. Fahl (2000))

$$\omega(h, u, s) \leq L, \tag{5.7}$$

a result that is repeatedly used in proofs for the statements further below.

Lemma 5.1.5. ***(Sufficient model decrease)***
Given a model function, m_k, according to assumption 5.2 with bounded, non-critical gradient g_k at u_k (i.e. $0 < ||g_k|| \leq c_g$). Apply algorithm 5.2 and define

$$\omega_k := \begin{cases} \omega(m_k, u_k, -\lambda_k^B g_k), & \text{if } \lambda_k^B \text{ is required in alg. 5.2} \\ 0, & \text{else.} \end{cases} \tag{5.8}$$

Then $\omega_k \geq 0$ and there exists a constant $c_s > 0$ such that algorithm 5.2 produces a step s_k with

$$m_k(u_k) - m_k(u_k + s_k) \geq c_s ||g_k||^2 \min\left\{\frac{||g_k||^2}{1+\omega_k}, \Delta_k\right\}. \tag{5.9}$$

Proof. For a proof, see Fahl (2000, theorem 5.12). □

Thus, equation (5.9) provides a sufficient decrease condition for the model function when we apply algorithm 5.2.

If we now use the step determination algorithm to compute an approximate minimizer in algorithm 5.1 (lines 2-3), then we can easily deduce the corresponding decrease of the objective function f in case of a successful iteration (see Fahl (2000)).

Corollary 5.1.6. ***(Sufficient objective function decrease)***
Let the assumptions of lemma 5.1.5 be satisfied. Given an arbitrary iteration k of algorithm 5.1 where we use algorithm 5.2 for the 'step calculation', then we have

$$f(u_k) - f(u_k + s_k) \geq \eta_1 c_s ||g_k||^2 \min\left\{\frac{||g_k||^2}{1+\omega_k}, \Delta_k\right\} \tag{5.10}$$

if the iteration is successful.

Of course, it is not yet clear that there exist such successful iterations. This issue will be addressed further below.

Inexact Gradient Information

The next important generalization is an inexact gradient. The gradient of our model function, m_k, will in general not coincide with the real gradient at the trust region center point as it is the case for the quadratic model function as defined in (5.3). Thus, we have $\nabla f(u_k) \neq g_k$. However, to still get a convergence result, the error between $\nabla f(u_k)$ and g_k has to be controlled. The topic is well-known in literature and there are several approaches defining an upper bound for the difference. We briefly mention two of them. For constants $\kappa_1, \kappa_2 > 0$, Toint (1988) has proposed to demand a gradient accuracy in the sense of

$$||\nabla f(u_k) - g_k|| \leq \min\{\kappa_1 \Delta_k, \kappa_2\} \tag{5.11}$$

for all k. On the one hand, it is natural to refine the model gradient for a small trust radius, and this approach also implies ideas for a practical implementation. On the other hand, it may happen that the algorithm finds a critical point for the model, i.e. $||g_k|| = 0$, that is not critical for the objective function, thus, $||\nabla f(u_k)|| > 0$. This has to be avoided in addition to the estimate above.

To overcome this problem, Carter (1991) has recommended a relative error condition

$$\frac{||\nabla f(u_k) - g_k||}{||g_k||} \leq \zeta \tag{5.12}$$

with $\zeta \in (0, 1)$. Then, if we are close to a critical point for the model function, (5.12) appropriately ensures that the error $||\nabla f(u_k) - g_k||$ is small.

A further disadvantage of error condition (5.11) is that, given an unsuccessful iteration with resulting reduced trust radius Δ_k, the gradient g_k may need to be updated to fit the sharper error condition. This can be an expensive task in many applications.

In the following we assume our model function to fulfill the Carter condition, (5.12):

Assumption 5.3. ***(Model gradient)***
The gradient of the model function at the trust region center point u_k, g_k, fulfills condition (5.12) with $\zeta \in (0, 1)$.

This new assumption on the gradient accuracy allows now to make two important statements. Firstly, we can easily proof a connection between critical points for the model and the objective function.

Lemma 5.1.7. *Let assumption 5.3 be fulfilled and further assume a sequence of iterates u_{k_i}, $i = 1, 2, \ldots,$ with*

$$\lim_{i \to \infty} g_{k_i} = 0.$$

Then

$$\lim_{i \to \infty} \nabla f(u_{k_i}) = 0.$$

Proof. Cf. Conn et al. (2000, lemma 8.4.1). □

Secondly, we can make a statement on the success of a trust-region iteration if the descent is calculated via the step determination algorithm.

Lemma 5.1.8. ***(Successful iteration)***
Let assumptions 5.1, 5.2 and 5.3 be satisfied for a fixed iterate u_k with $\zeta \in (0, 1 - \eta_1)$. Applying algorithm 5.2 to get a trial step s_k, and assuming $\omega(m_k, u_k, s_k) \leq c_w$ for a $c_w > 0$. Then, for sufficiently small Δ_k, we have $\rho_k > \eta_1$, and the iteration is successful.

Proof. For a complete proof, we refer to Fahl (2000, theorem 6.2). □

Note that our model function based on a POD model also involves inexact function values at the trust region center point, i.e. $m_k(u_k) \neq f(u_k)$. However, it turns out that this is only implicitly important to ensure convergence.

Convergence of the Generalized Algorithm

Given the framework and assumptions stated above, we are able to state a convergence result for this generalized type of model functions similar to the result in theorem 5.1.2.

Theorem 5.1.9. ***(Generalized strong global convergence)***
Let assumptions 5.1, 5.2 and 5.3 be satisfied with $\zeta \in (0, 1 - \eta_2)$. Further, assume $\{u_k\}$ to be a sequence of iterates produced by algorithm 5.1 using algorithm 5.2 to compute the trial steps $\{s_k\}$. If we have, for $c_b > 0$,

$$b_k := 1 + \max_{0 \leq i \leq k} \{\omega_i, \omega(m_i, u_i, s_i)\} \leq c_b, \tag{5.13}$$

for all k, then

$$\lim_{k \to \infty} ||\nabla f(u_k)|| = 0.$$

Proof. A complete proof can be found in Fahl (2000, theorem 6.3 and theorem 6.4). □

Note that the bounded curvature in condition (5.13) is the counterpart to the bounded Hessian for quadratic model functions.

The next section shows that for a general class of optimal control problems – as they have been discussed in the previous chapters – a model function based on a POD model fits the assumptions on the model function that have been made above to guarantee convergence.

5.2 Trust-Region POD Algorithms

We have seen a convergence result for a generalized trust-region algorithm in the last section. This section is now devoted to the combination of this result with a reduced order model based on POD. We recall the general optimal control problem that has been discussed at several points in this thesis. For given market data d_i at $\hat{t}_i$ $(i = 1, \dots, D)$, find solutions $y \in W([0,T], V)$ and $u \in U$ satisfying

$$\begin{aligned} \min_{y \in W, u \in U} J(y,u) &:= \frac{1}{2} \sum_{i=1}^{D} ||Cy(\hat{t}_i) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2} ||u||^2 \\ s.t. \quad & \dot{y}(t) + A(u;t)y(t) - l(u;t) = 0 \ , \quad t \in (0,T] \\ & y(0) = y_0. \end{aligned} \tag{5.14}$$

As already mentioned, this problem can be written as $\tilde{f}(u) := J(y(u), u)$ such that a trust-region algorithm can be applied. We now want to build a model function which is based on a POD reduced order model and can therefore be solved much more cheaply than the original one.

5.2.1 Derivation

Focussing on a discretized version of (5.14), our problem is given by

$$\min_{u \in U} f(u) = \frac{1}{2} \sum_{i=1}^{D} ||Cy_{k_i}^{FE}(u) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2} ||u||^2, \tag{5.15}$$

where $\{y_k^{FE}\}_{k=0}^{n_t} \subset H^{n_x}$ is the finite element approximation to the state equation discretized with the θ-method, i.e

$$\begin{aligned} &\bar{\partial} y_k^{FE} + \theta A(u;\ t_k) y_k^{FE} + (1-\theta) A(u;\ t_{k-1}) y_{k-1}^{FE} = \\ &\qquad \theta l(u;\ t_k) + (1-\theta) l(u;\ t_{k-1}), \qquad k = 1, \dots, n_t \\ &y_0^{FE} = y_0. \end{aligned} \tag{5.16}$$

Each function evaluation of f requires the solution of the discretized differential equation. The computational effort of solving n_t systems of equations of size $n_x \times n_x$ is quite expensive and should therefore be avoided. Thus, we build a small POD model on the space $\mathcal{V}^l$ with

dimension $l \ll n_x$ for the constraint. This means, for a given control u_k, we define

$$m_k^l(u) = \frac{1}{2}\sum_{i=1}^{D} ||Cy_{k_i}^l(u) - d_i||_{\mathcal{H}}^2 + \frac{\alpha}{2}||u||^2, \tag{5.17}$$

where $\{y_k^l\}_{k=0}^{n_t} \subset \mathcal{V}^l$ is the POD approximation to the state equation discretized in time with the θ-method, i.e

$$\begin{aligned}
&\bar{\partial} y_k^l + \theta A(u;\ t_k)y_k^l + (1-\theta)A(u;\ t_{k-1})y_{k-1}^l = \\
&\qquad\qquad \theta l(u;\ t_k) + (1-\theta)l(u;\ t_{k-1}), \qquad k = 1, \dots, n_t \\
&y_0^l = y_0.
\end{aligned}$$

As we have seen in the last section, a model function needs to fulfill certain properties to guarantee convergence of the corresponding trust-region algorithm. We need to have a Fréchet-differentiable objective function (assumption 5.1) with Lipschitz continuous gradient, thus, we need to assume that the operators $A(u;t)$ and $l(u;t)$ are sufficiently smooth with respect to the control variable. Since the model function uses the same operators, this assumption also implies its sufficient smoothness (assumption 5.2).

For the last main requirement, namely an adequate gradient accuracy, we can now make use of the results of the previous chapter. According to assumption 5.3, we have to guarantee the accuracy of the model gradient at the trust region center point u_k.

Theorem 4.2.13 has estimated the error for a reduced derivative. Combining this with assumption 5.3 yields the following property for the accuracy of the model function:

$$||\nabla f(u_k) - \nabla m_k^l(u_k)|| \leq \Big(c \sum_{j=l+1}^{r} \lambda_j\Big)^{\frac{1}{2}} \leq \zeta ||\nabla m_k^l(u_k)||. \tag{5.18}$$

To have the outer inequality, i.e. the Carter condition, fulfilled, we need to guarantee that the inequality on the right-hand side holds. Increasing the number of POD basis functions, l, reduces the middle term, however, it may also decrease the term on the right-hand side which is depending on l as well. But even for $||\nabla m^l(u_k)|| = 0$, we can – at least theoretically – choose $l = r$ to get (5.18).

So, the crucial point about the a priori error estimates is that we now know that there exists an l such that the condition above is satisfied.

5.2.2 Convergence Proof

Before we state a global convergence proof for a trust-region algorithm based on POD, we first define the adaptive trust-region POD method in algorithm 5.3.

Let us just comment on some issues of the proposed algorithm. The general structure is given by the basic trust-region algorithm 5.1. The task 'compute the model' is now replaced by the lines 1-3. Thus, to get our model we first have to calculate the state and adjoint snapshots for the current control u_k. These are then used to calculate a POD basis where the rank l is chosen such that the inequality in line 3 is satisfied. The Cauchy step for the solution of the trust-region subproblem is again replaced by the step determination

Algorithm 5.3 Adaptive TRPOD algorithm

Input: $\Delta_0 > 0$, $k = 0$, an initial control $u_0 \in U$ and constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \gamma_3, \zeta$ satisfying

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3, \quad 0 < \zeta < 1 - \eta_2.$$

1: **compute** state $y(u_k)$ and adjoint $p(u_k)$ to form the set of snapshots $\mathcal{S}$

2: **compute** for $\mathcal{S}$ the POD basis of rank l and model m_k^l such that

3: $$||\nabla f(u_k) - \nabla m_k^l(u_k)|| \leq \zeta ||\nabla m_k^l(u_k)||$$

4: **compute** an approximate solution $s_k \in U$ to

5: $$\min_{||s|| \leq \Delta_k} m_k^l(u_k + s)$$

6: using algorithm 5.2

7: **compute** $f(u_k + s_k)$ **and**

8: $$\rho_k = \frac{f(u_k) - f(u_k + s_k)}{m_k^l(u_k) - m_k^l(u_k + s_k)}$$

9: **if** $\rho_k \geq \eta_2$ **then**

10: **set** $u_{k+1} = u_k + s_k$ **and** $\Delta_{k+1} \in [\Delta_k, \gamma_3 \Delta_k]$

11: **set** $k \leftarrow k + 1$ **and** go to line 1

12: **else if** $\eta_1 \leq \rho_k < \eta_2$ **then**

13: **set** $u_{k+1} = u_k + s_k$ **and** $\Delta_{k+1} \in [\gamma_2 \Delta_k, \Delta_k]$

14: **set** $k \leftarrow k + 1$ **and** go to line 1

15: **else if** $\rho_k < \eta_1$ **then**

16: **set** $u_{k+1} = u_k$ **and** $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$

17: **set** $k \leftarrow k + 1$ **and** go to line 4

18: **end if**

algorithm. We also stress that in case of an unsuccessful iteration (line 15), we do not have to compute a new POD model. We can keep the old one and only decrease the trust radius.

The main result of the thesis, a global convergence proof for the algorithm above, can now be stated.

Theorem 5.2.1. ***(Strong global convergence of the adaptive TRPOD)*** *Given problem (5.15) with Lipschitz continuous Fréchet derivatives A' and l' in (5.16) and $\zeta \in (0, 1 - \eta_2)$, let the assumptions of theorem 4.2.13 be satisfied. Further, assume $\{u_k\}$ to be a sequence of iterates produced by algorithm 5.3. Then,*

$$\lim_{k \to \infty} ||\nabla f(u_k)|| = 0.$$

Proof. We begin by showing that the assumptions of theorem 5.1.9 are satisfied. First, the function $f(u)$ is clearly bounded below by zero. Further, the Lipschitz continuous operators

A' and l' yield a Lipschitz continuous gradient $\nabla f(u)$ for (5.15), and, due to the similar structure of the reduced model function, (5.17), this directly implies the differentiability of $m_k^l(u)$ and the Lipschitz continuity of $\nabla m_k^l(u)$. Using, in turn, this latter conclusion together with (5.7), we obtain inequality (5.13).

By definition of the algorithm, the Carter condition is fulfilled for each model function, however, the point here is that it can be fulfilled. This is ensured by the inclusion of the adjoint snapshots and the error estimate of theorem 4.2.13, which completes the proof. □

Let us summarize the main assumptions on the problem 5.14 that have to be fulfilled. First, the objective function has to be sufficiently smooth with respect to the control variables, thus, the operators $A(u;t)$ and $l(u;t)$ have to be differentiable and the derivatives have to satisfy a Lipschitz condition. Second, to be able to apply the error estimates to the state equation, the bilinear operator has to be elliptic, and we need to use an implicit discretization scheme as the backward Euler or Crank-Nicolson method. To further use the error estimates for the gradient, the reduced order model needs to include adjoint information.

It is easy to design an example where the necessity of the adjoint snapshots gets clear.

Example 5.2.2. ***(Omit adjoint snapshots)***
Given the optimization problem

$$\begin{aligned} \min_{y\in W, u\in\mathbb{R}^1} \; & J(y,u) := \tfrac{1}{2}||y(T)-d||_H^2 \\ s.t. \quad & \dot{y}(t) - \Delta y(t) - utE = 0 \;, \quad t\in(0,T] \\ & y(0) = 0 \end{aligned} \tag{5.19}$$

with $E \in V^*$. The gradient of the reduced function $f(u) = J(y(u),u)$ is then given by $\nabla f(u) = \int_0^T \langle p(t), -tE\rangle dt$ with adjoint $p \in W$ given by

$$\begin{aligned} \dot{p}(t) + \Delta p(t) &= 0 \\ p(T) &= y(T) - d. \end{aligned}$$

Of course, in general $p \neq 0 \;\forall\; t$, and thus, in general $\nabla f(u) \neq 0$. However, a POD model function solely based on the state snapshots for the particular control $u_k = 0$ would yield an empty POD basis and accordingly, $m_k^{POD}(u) \equiv \frac{1}{2}||d||_H^2$.

This example also illustrates the reason for combining the state and adjoint snapshots in one POD basis instead of building an own POD basis for each of them. Let us assume to have one basis for the state and another one for the adjoint equation. Then, in the example above, the state basis would be an empty set and the model function which is only based on the state solution would be given by $m_k^{POD}(u) \equiv \frac{1}{2}||d||_H^2$ with gradient g_k clearly equal to zero. Given a second basis for the adjoint at u_k, we would be able to sufficiently approximate the gradient $\nabla f(u_k)$, however, this would not be the gradient of our model function m_k^{POD}, and the theory above would not be applicable.

The next remark handles the extreme case when state and adjoint snapshots both are zero.

Remark 5.2.3. ***(Empty set of snapshots)***
It may happen that the state and adjoint snapshots both are zero. In case of an additional regularization term, this does not need to imply $\nabla f(u_k) = 0$*. However, the real gradient and the model gradient are even identical* $\nabla f(u_k) = \alpha u_k = \nabla m_k^{POD}(u_k)$ *because* $m_k^{POD}(u) = \frac{\alpha}{2}||u||^2$*, and we can still make a step in steepest descent direction. In this particular situation, the computation of a POD basis is redundant because the set of snapshots is empty and the local minimizer of the trust-region subproblem with trust radius* Δ_k *can be calculated directly as* $u_{k+1} = u_k - \min\{\Delta_k, ||u_k||\}\frac{1}{||u_k||}u_k$*.*

5.2.3 Managing the POD Error

One main question that arises when the algorithm is to be implemented is how to manage the number of POD basis functions. We observe that, to get convergence, the state and adjoint at u_k have to be computed. This drawback turns out to be an advantage now because in many applications, the exact gradient $\nabla f(u_k)$ can be calculated via the adjoint approach quite cheaply. Given this gradient, we estimate an initial value for the number of POD basis functions, l, and then cheaply compute the corresponding reduced gradient, $\nabla m_k^l(u_k)$, as well as the factor

$$\zeta_k^l := \frac{||\nabla f(u_k) - \nabla m_k^l(u_k)||}{||\nabla m_k^l(u_k)||}. \tag{5.20}$$

If $\zeta_k^l > 1 - \eta_2$, then we need to increase the number of basis functions, l, and compute the new ζ_k^{l+1} iteratively until the condition is satisfied. For this, it would be preferable to be able to stop the eigenvalue solver and restart it if necessary. Or we may compute a few eigenvalues more than we expect to need – this expectation might be a combination of the previous number of basis functions and the previous value ζ_{k+1}^l – to be able to easily increase l.

Further, we have to set the weightings η_S and η_A for the state and adjoint snapshots. In our application, the state equation evolves in the same order of magnitude throughout the calibration. But the adjoint equation tends towards zero when the least-squares error in the objective function is reduced. Thus, in our implementation, the weighting factor η_A is coupled with the inverse of some norm of the adjoint solution.

The error condition above is hard to satisfy in practice when $||\nabla m_k^l(u_k)||$ is very small. This is due to rounding errors which occur if the POD basis is very large and the corresponding eigenvalues are close to zero. On the one hand, due to the inclusion of weighted adjoint snapshots POD is less susceptible to computational rounding errors because the eigenvalues do not drop that fast. On the other hand, in this case, i.e. in case of a large POD basis, it may also be not efficient in the sense of computational effort to use a reduced order model anymore. However, we can then use the current iterate as a starting vector and switch to a traditional minimizer.

5.2.4 Multi-level Strategies

It is clear that the solution of the exact – i.e. discretized on a sufficiently fine finite element grid – state and adjoint equation is a major part of the total computing time. Further,

concerning the construction of the POD basis and of the reduced system matrices, the original spatial discretization is determining for the computational effort. Thus, it seems to be a good idea to combine the POD approach with a multi-level algorithm. This means, if we are far away from the optimal solution of the problem, we do not only use less POD basis functions to compute our model in order to satisfy (5.20), but we also compute the snapshots, i.e. the FE solutions, on a coarser grid.

From a theoretical point of view, we may be able to combine the previous convergence result with a convergence proof for a multi-scale trust-region algorithm in Gratton et al. (2008). However, in the calibration problem that we consider, we restrict ourselves to a nested iteration, i.e. solutions of the TRPOD algorithm on a coarser grid – where we also use the coarse finite element gradient as our reference in (5.20) – are taken as starting point on the finer grid. Using such an implementation, a generalization of the convergence proof above is not necessary.

Note that we do not need to prolongate the control variable in our application since it is parameterized and its dimension does not vary even if the dimension of the PDE constraint is changed. Further note that a fully adaptive refinement of the finite element space is not possible because we are restricted to the use of an equidistant mesh preserving the Toeplitz property of the discretized integral term. Thus, the hierarchical predefined grids that are used just differ in the equidistant step size.

5.3 Numerical Results

As a numerical example showing the efficiency of the algorithm proposed above, we consider the problem discussed in section 3.3.3. Hence, we calibrate Merton's jump-diffusion model with constant or parameterized volatility function to 100 given market prices (table 3.9).

The parameters used in the trust-region algorithm are set as follows:

$$\Delta_0 = 0.1,\ \eta_1 = 0.1,\ \eta_2 = 0.8,\ \gamma_1 = 0.25,\ \gamma_2 = 0.9,\ \gamma_3 = 1.05,$$

whereas the discretization setting and starting vector are given by

$$\underline{x} = -5,\ \overline{x} = 5,\ T_{max} = 5,\ r \equiv 5\%,\ \Delta x = 0.0025,\ \Delta T = 0.0125,$$
$$\text{starting vector: } u = (\lambda, \mu_J, \sigma_J, \sigma^2) = (40\%, 0\%, 40\%, 40\%^2),$$

i.e. the same setting as in (3.50). To get comparable results, we further use the same regularization term as introduced on page 54.

Before we discuss the results, let us comment on some implementation issues. Since the building of one reduced order model is quite expensive and includes the computation of the full gradient, it seems to be not very efficient to only make a step in steepest descent direction as proposed in the first part of the step determination algorithm 5.2. We rather restrict ourselves to the second part of the algorithm and use the MATLAB procedure 'fmincon' to find a descent direction providing a greater decrease. Since PIDE evaluations for the POD model function are far less expensive, we use the Gauß-Newton approach to compute first- and second-order information within the trust-region subproblems.

Starting with the calibration of four parameters, i.e. with constant volatility, table 5.1 now

k	$\|\|\nabla f(u_k)\|\|_2$	$f(u_k)$	$m_k(u_{k+1})$	ρ_k	Δ_k	#POD l	ζ_k^l
0	8.34e+0	1.21e+0	2.43e-1	1.00	0.10	8	0.0024
1	5.81e+0	2.41e-1	4.45e-4	1.00	0.11	10	0.0019
2	1.51e-1	1.32e-4	7.47e-5	1.01	0.11	11	0.0012
3	1.40e-3	7.44e-5	5.84e-5	1.01	0.12	12	0.1368
4	1.80e-3	5.83e-5	5.63e-5	1.09	0.12	18	0.1551
5	4.41e-4	5.61e-5					

Table 5.1: Iterations of a TRPOD calibration run with corresponding gradient norm, function values, ratios ρ_k, trust radius Δ_k, number of used POD basis functions l and ζ_k^l; constant volatility (four control parameters)

algorithm	evaluations		timing (sec.)				optimal values	
	FE	POD	total	FE	POD	basis	$f(u_{opt})$	$\|\|\nabla f(u_{opt})\|\|_2$
TRPOD	12	210	55	36	10	2	5.61e-5	4.41e-4
ML TRPOD	16	135	24	15	4	1	7.62e-5	5.71e-4
quasi-Newton	184	—	554	516	—	—	5.34e-5	1.00e-4

Table 5.2: Comparison between TRPOD, Multi-level TRPOD and a quasi-Newton algorithm: number and timings of FE- and POD evaluations, total computing time, time for POD basis computation, and optimal function value and gradient norm; constant volatility (four control parameters)

shows the corresponding calibration run. Given the iteration counter in the first column, the second column contains the norm of the 'exact' – i.e. based on the full finite element state and adjoint solution – gradient. This value is also used as stopping criterion. The following columns show the 'exact' function value in u_k and the value of the model function in the optimal solution of the trust-region subproblem $m_k(u_{k+1})$. ρ_k is the ratio between actual reduction and predicted reduction as defined in algorithm 5.3 line 8 and Δ_k is the corresponding trust radius. The last two columns contain values directly connected to the POD model function. We see the number of POD functions, l, that is used in the current iteration and the ratio ζ_k^l.

We observe that all iterations are successful and all ratios ρ_k are close to one. The approximation quality of the POD model function is quite good even if we veer away from the trust region center point as it can be seen by comparing, e.g., $m_2(u_3)$ and $f(u_3)$.

The most interesting results can be found in the last two columns. We observe that the POD model satisfies the Carter condition using very few POD basis functions when we are far from the optimal control. And as expected, the closer the optimal solution, i.e. the smaller the model gradient, the more POD basis functions are needed to guarantee $\zeta_k^l < 1 - \eta_2$. However, in this real-world example, the results are acceptable, in particular if we compare them with the quasi-Newton approach of section 3.3.3.

This comparison is shown in table 5.2. We see the performance of different algorithms in terms of evaluations of the discretized differential equations (discretized on a finite element and the POD space, resp.), in terms of itemized computing times and with regard to the function and gradient values in the optimal solution. The algorithms used are the TRPOD

k	$\|\|\nabla f(u_k)\|\|_2$	$f(u_k)$	$m_k(u_{k+1})$	ρ_k	Δ_k	#POD l	ζ_k^l
0	3.58e+0	1.21e+0	2.43e-1	1.00	0.10	10	0.0228
1	1.77e+0	2.44e-1	2.76e-4	1.00	0.11	16	0.0286
2	4.91e-2	3.63e-4	1.29e-5	1.00	0.11	22	0.0216
3	1.43e-3	1.38e-5	1.14e-5	0.83	0.12	28	0.1562
4	1.24e-4	1.18e-5					

Table 5.3: Iterations of a TRPOD calibration run with corresponding gradient norm, function values, ratios ρ_k, trust radius Δ_k, number of used POD basis functions l and ζ_k^l; local volatility (23 control parameters)

algorithm	evaluations		timing (sec.)				optimal values	
	FE	POD	total	FE	POD	basis	$f(u_{opt})$	$\|\|\nabla f(u_{opt})\|\|_2$
TRPOD	10	600	86	30	45	2	1.18e-5	1.24e-4
ML TRPOD	18	768	59	21	29	1	1.19e-5	2.42e-4
quasi-Newton	274	—	888	809	—	—	9.62e-6	8.29e-4

Table 5.4: Comparison between TRPOD, Multi-level TRPOD and a quasi-Newton algorithm: number and timings of FE- and POD evaluations, total computing time, time for POD basis computation, and optimal function value and gradient norm; local volatility (23 control parameters)

algorithm (cf. table 5.1), a multi-level version of the TRPOD algorithm using three different finite element grids (level 1: $\Delta x \times \Delta T = 0.01 \times 0.025$, level 2: 0.005×0.025, level 3: 0.0025×0.0125) and the quasi-Newton approach.

Let us first note that the optimal values in the last two columns have the same order of magnitude and we can thus just compare the computational effort. The first observation is that TRPOD needs 55 sec. for the optimization compared to 554 sec. in the quasi-Newton approach corresponding to a time saving of 90%.

We may also compare the evaluations of the differential equation. The quasi-Newton needs 184 evaluations – containing the state and the adjoint equations – all computed on the full finite element grid. In contrast, the TRPOD algorithm needs $12 + 210$, this means even more, evaluations. However, most of them are computed via the reduced order model and we only need six state and six adjoint solutions on the full grid. In the table, the total computing time is further split into the time needed for the full PIDE solutions, the time needed for the POD solutions and the time for the basis computation, i.e. the solution of the eigenvalue problem. We observe that 12 finite element evaluations require about 36 sec., i.e. three seconds each, and 210 POD evaluations only need ten seconds, i.e. about 0.05 sec. each. The two seconds for the basis computation are negligible. The seven seconds that are not listed in the table ($55 - 36 - 10 - 2 = 7$) are mainly needed to compute the exact gradient from a given state and adjoint solution.

As we have already expected in the previous section, implementing a multi-level strategy yields a further reduction of the computing time, to be precise by a factor of more than two in our application. Although we need more iterations in this case – which is clear taking a look at the number of FE evaluations –, the main time saving is observable in the solution of

the full PIDE. As we need 15 sec. for 16 evaluations, the time per evaluation has decreased to one second in average. However, also the time per POD evaluation has decreased to a mean of 0.03 sec. since the reduced system matrices are also based on the coarser FE system matrices.

We now consider the second example that has already been discussed in section 3.3.3. We again calibrate Merton's jump-diffusion model to the market data as above, but now the volatility is assumed to be parameterized with 20 parameters. Thus, together with the jump intensity λ, the average jump size μ_J and the mean jump size σ_J, there are 23 parameters to be calibrated. Table 5.3 is set up analogously to the preceding table 5.1. We observe a similar convergence as in the case of four parameters, however, we need a few POD basis functions more to be able to satisfy the Carter condition.

Table 5.4 shows – analogously to table 5.2 – that the TRPOD algorithm still saves about 90% of the computing time compared to the quasi-Newton method. Note that the time saving is even more pronounced when we compare the TRPOD which is based on the Gauß-Newton approach for the solution of the trust-region subproblems with the Gauß-Newton results in section 3.3.3.

Chapter 6

Conclusions

Studying different option pricing models in the first part of the thesis, a jump-diffusion model with an additional local volatility function turned out to have many advantages with regard to the modeling of underlying prices. The calibration of such models was done in a least-squares formulation comparing given market prices with our model prices. The model prices were calculated via a Dupire-like partial integro-differential equation, such that we needed to solve an optimization problem with a PIDE constraint, a quite challenging task from a numerical point of view.

In this thesis, we used a preconditioned GMRES algorithm to solve the differential equation constraint reducing the complexity even for implicit time discretization schemes as Crank-Nicolson or the Rannacher smoothing scheme to $\mathcal{O}(n_x \log_2 n_x)$ per time step. However, when an optimization algorithm is applied, the repeated solution of the PIDE is still expensive.

Thus, the main part of this thesis dealt with reduced order models based on proper orthogonal decomposition. This technique is known to be very efficient for smooth parabolic differential equations and their application in finance as shown here is a current field of research. Numerical results proved the approximation quality of the reduced order models and also their efficiency in terms of computational effort. Theoretically, we have shown error estimates for time-dependent parabolic problems based on the work of Kunisch and Volkwein (2001). The extension of these error estimates to optimal control problems was an important result. We estimated the difference between the 'true' objective function and a reduced objective function where the parabolic constraint is based on a reduced order model. The same was done for the gradient of the objective function and its reduced counterpart, however, additional adjoint information was crucial here.

All estimates only hold true for unchanged parameters, but a globalization has been achieved by an embedding in a trust-region framework as proposed by Arian et al. (2000). As the main result of this thesis, we have shown convergence for this trust-region POD algorithm, in which the size of the POD basis has to be adjusted in each iteration such that a certain gradient error tolerance is satisfied.

The theoretical results are not only valid for the calibration problem considered here. The algorithm can be applied to a wide class of optimization problems with general parabolic constraints and this class can probably be extended in future. The main ingredients are error estimates for the constraint and based on this for the gradient of the reduced objective function as shown in chapter 4.

However, the application of the TRPOD algorithm to a certain problem has to be reasonable. In other words, the problem has to be suited for the application of POD. The thesis has addressed many influencing factors, e.g. the unwanted effects of a strong convection

on the decay of the eigenvalues and therefore on the size of the POD basis that is needed to guarantee a predetermined error tolerance. Another important aspect determining the efficiency of the TRPOD algorithm is the affine parameter- and time-dependence of the operators occurring in the partial differential equations. This effect is even more important considering nonlinear problems.

Given an appropriate structure of the problem, the reduced matrices and vectors have to be computed only once per TRPOD iteration. For instance, in the results that we have seen in this thesis, we had to compute the reduced stiffness matrix each time we changed the parameters because the dependence on the jump parameters is nonlinear.

Nevertheless, the results of the TRPOD algorithm are promising in the sense that they provide a significant acceleration of the calibration of jump-diffusion option pricing models.

List of Tables

List of Figures

Bibliography

Achdou, Y. and Pironneau, O. (2005). *Computational Methods for Option Pricing*, SIAM.

Afanasiev, K. and Hinze, M. (2001). Adaptive control of a wake flow using proper orthogonal decomposition, *Lecture Notes in Pure and Applied Mathematics* pp. 317–332.

Almendral, A. and Oosterlee, C. (2005). Numerical valuation of options with jumps in the underlying, *Applied Numerical Mathematics* **53**(1): 1–18.

Andersen, L. and Andreasen, J. (2000). Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, *Review of Derivatives Research* **4**(3): 231–262.

Andersen, L. and Brotherton-Ratcliffe, R. (1998). The equity option volatility smile: an implicit finite-difference approach, *Journal of Computational Finance* **1**(2): 5–37.

Antoulas, A. (2005). *Approximation of Large-Scale Dynamical Systems*, Advances in Design and Control, SIAM.

Antoulas, A., Sorensen, D. and Gugercin, S. (2001). A survey of model reduction methods for large-scale systems, *Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing, Contemporary Mathematics, AMS publications* **280**: 193–219.

Arian, E., Fahl, M. and Sachs, E. W. (2000). Trust-region proper orthogonal decomposition for flow control, *ICASE Report 2000–25, ICASE, NASA Langley Research Center* .

Armstrong, N., Painter, K. and Sherratt, J. (2006). A continuum approach to modelling cell–cell adhesion, *Journal of Theoretical Biology* **243**(1): 98–113.

Barndorff-Nielsen, O. E. (1997). Processes of normal inverse gaussian type, *Finance and Stochastics* **2**: 41–68.

Barrault, M., Maday, Y., Nguyen, N. and Patera, A. (2004). An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus Mathematique* **339**(9): 667–672.

Bates, D. (1996). Jump and stochastic volatility: Exchange rate processes implicit in Deutsche Mark options, *The Review of Financial Studies* **9**: 69–107.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *Journal of Political Economy* **81**(3): 637–654.

Brenner, S. C. and Scott, L. R. (2008). *The Mathematical Theory of Finite Element Methods*, Texts in Applied Mathematics, 3rd edn, Springer.

Briani, M., Natalini, R. and Russo, G. (2007). Implicit–explicit numerical schemes for jump–diffusion processes, *Calcolo* **44**(1): 33–57.

Carr, P., Geman, H., Madan, D. and Yor, M. (2002). The fine structure of asset returns: An empirical investigation, *The Journal of Business* **75**: 305–332.

Carr, P. and Madan, D. B. (1999). Option valuation using the fast Fourier transform, *Journal of Computational Finance* **2**(4): 1–18.

Carter, R. G. (1991). On the global convergence of trust region algorithms using inexact gradient information, *SIAM Journal on Numerical Analysis* **28**(1): 251–265.

Chaturantabut, S. and Sorensen, D. (2011). Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, *Mathematical and Computer Modelling of Dynamical Systems* **17**(4): 337–353.

Chriss, N. (1997). *Black-Scholes and Beyond: Option Pricing Models*, Irwin.

Chung, K. L. and Williams, R. J. (1990). *Introduction to Stochastic Integration*, Birkhäuser.

Conn, A., Gould, N. and Toint, P. (2000). *Trust-Region Methods*, MPS-SIAM series on optimization, SIAM.

Cont, R., Lantos, N. and Pironneau, O. (2011). A reduced basis for option pricing, *SIAM Journal on Financial Mathematics* **2**(1): 287–316.

Cont, R. and Tankov, P. (2004a). *Financial Modelling with Jump Processes*, Chapman & Hall.

Cont, R. and Tankov, P. (2004b). Non-parametric calibration of jump-diffusion option pricing models, *Journal of Computational Finance* **7**(3): 1–49.

Cont, R. and Voltchkova, E. (2005). A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, *SIAM Journal on Numerical Analysis* **43**(4): 1596–1626.

Cox, J. C., Ross, S. A. and Rubinstein, M. (1979). Option pricing: A simplified approach, *Journal of Financial Economics* **7**(3): 229–263.

Dautray, R. and Lions, J.-L. (1992). *Mathematical Analysis and Numerical Methods for Science and Technology, Vol.5: Evolution Problems I*, Springer.

Derman, E. and Kani, I. (1994). The volatility smile and its implied tree, *Quantitative Strategies Research Notes, Goldman Sachs* .

Dupire, B. (1994). Pricing with a smile, *Risk* **7**: 1–10.

Düring, B., Jüngel, A. and Volkwein, S. (2008). Sequential quadratic programming method for volatility estimation in option pricing, *Journal of Optimization Theory and Applications* **139**(3): 515–540.

Elliott, R. and Kopp, P. (2005). *Mathematics of Financial Markets*, Vol. 10 of *Springer finance*, Springer.

Fahl, M. (2000). *Trust-Region Methods for Flow Control Based on Reduced Order Modelling*, PhD thesis, Universität Trier.

Föllmer, H. and Schied, A. (2004). *Stochastic Finance: An Introduction in Discrete Time*, De Gruyter studies in mathematics, Walter de Gruyter.

Gerisch, A. (2010). On the approximation and efficient evaluation of integral terms in PDE models of cell adhesion, *Journal of Numerical Analysis* **30**: 173–194.

Giles, M. and Carter, R. (2006). Convergence analysis of Crank-Nicolson and Rannacher time-marching, *Journal of Computational Finance* **9**(4): 89–112.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, Applications of mathematics, Springer.

Goll, C., Rannacher, R. and Wollner, W. (2011). Goal-oriented mesh adaptation in the space-time finite element approximation of the Black-Scholes equation, *submitted* .

Golub, G. and van Loan, C. (1996). *Matrix Computations*, Johns Hopkins studies in the mathematical sciences, Johns Hopkins University Press.

Gratton, S., Sartenaer, A. and Toint, P. L. (2008). Recursive trust-region methods for multiscale nonlinear optimization, *SIAM Journal on Optimization* **19**(1): 414–444.

Grepl, M. A. (2005). *Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations*, PhD thesis, Massachusetts Institute of Technology.

Grepl, M. A. and Patera, A. T. (2005). A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations, *ESAIM: Mathematical Modelling and Numerical Analysis* **39**(01): 157–181.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options, *Review of Financial Studies* **6**: 327–343.

Hinze, M., Pinnau, R., Ulbrich, M. and Ulbrich, S. (2009). *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications, Springer.

Hinze, M. and Volkwein, S. (2008). Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition, *Computational Optimization and Applications* **39**(3): 319–345.

Holmes, P. J., Lumley, J. L., Berkooz, G., Mattingly, J. C. and Wittenberg, R. W. (1997). Low-dimensional models of coherent structures in turbulence, *Physics Reports* **287**(4): 337–384.

Holmes, P., Lumley, J. and Berkooz, G. (1996). *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press.

Hull, J. (2008). *Options, Futures, and Other Derivatives*, 7th edn, Pearson Prentice Hall.

Hull, J. C. and White, A. D. (1987). The pricing of options on assets with stochastic volatilities, *Journal of Finance* **42**(2): 281–300.

Kahlbacher, M. and Volkwein, S. (2007). Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems, *Discussiones Mathematicae: Differential Inclusions, Control and Optimization* **27**: 95–117.

Kahlbacher, M. and Volkwein, S. (2012). POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems, *ESAIM: Mathematical Modelling and Numerical Analysis* **46**(02): 491–511.

Karatzas, I. and Shreve, S. (2000). *Brownian Motion and Stochastic Calculus*, Graduate texts in mathematics, Springer.

Kelley, C. (1995). *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in applied mathematics, Society for Industrial and Applied Mathematics.

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

Kunisch, K. and Volkwein, S. (1999). Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, *Journal of Optimization Theory and Applications* **102**(2): 345–371.

Kunisch, K. and Volkwein, S. (2001). Galerkin proper orthogonal decomposition methods for parabolic problems, *Numerische Mathematik* **90**(1): 117–148.

Kunisch, K. and Volkwein, S. (2002). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM Journal on Numerical Analysis* **40**(2): 492–515.

Kunisch, K. and Volkwein, S. (2008). Proper orthogonal decomposition for optimality systems, *Mathematical Modelling and Numerical Analysis* **42**(1): 1–23.

Lamberton, D. and Lapeyre, B. (1996). *Introduction to Stochastic Calculus Applied to Finance*, Chapman & Hall.

Larsson, S. and Thomée, V. (2003). *Partial Differential Equations with Numerical Methods*, Texts in applied mathematics, Springer.

Lions, J.-L. (1971). *Optimal Control of Systems Governed by Partial Differential Equations*, Springer.

Lipton, A. (2002). The vol smile problem, *Risk* **February**: 61–65.

Lörx, A. (2012). *Adjoint-based Calibration of Local Volatility Models*, PhD thesis, Universität Trier.

Lumley, J. (1967). The structure of inhomogeneous turbulent flows, *in* A. M. Yaglom and V. I. Tatarski (eds), *Atmospheric turbulence and radio wave propagation*, Nauka, Moscow, pp. 166–178.

Matache, A.-M., von Petersdorff, T. and Schwab, C. (2004). Fast deterministic pricing of options on Lévy driven assets, *Mathematical Modelling and Numerical Analysis* **38**(1): 37–72.

Merton, R. C. (1973). Theory of rational option pricing, *Bell Journal of Economics and Management Science* **4**: 141–183.

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

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*, Springer series in operations research, Springer.

Øksendal, B. (2003). *Stochastic Differential Equations – An Introduction with Applications*, 6th edn, Springer.

Pironneau, O. (2007). Dupire identities for complex options, *Compte Rendu de l'Academie des Sciences I* **344**: 127–133.

Pironneau, O. (2009). Calibration of options on a reduced basis, *Journal of Computational and Applied Mathematics* **232**: 139–147.

Rannacher, R. (1984). Finite element solution of diffusion problems with irregular data, *Numerische Mathematik* **43**: 309–327.

Ravindran, S. (2002). Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition, *SIAM Journal on Scientific Computing* **23**(6): 1924–1942.

Reed, M. and Simon, B. (1980). *Methods of Modern Mathematical Physics: Functional analysis*, Methods of Modern Mathematical Physics, Academic Press.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*, 2nd edn, SIAM.

Saad, Y. and Schultz, M. H. (1986). GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* **7**: 856–869.

Sachs, E. and Schu, M. (2008). Reduced order models (POD) for calibration problems in finance, *in* K. Kunisch, G. Of and O. Steinbach (eds), *Numerical Mathematics and Advanced Applications, ENUMATH 2007*, Springer, pp. 735–742.

Sachs, E. and Schu, M. (2010). Reduced order models in PIDE constrained optimization, *Control and Cybernetics* **39**: 661–675.

Sachs, E. and Schu, M. (2012a). Gradient computation for model calibration with pointwise observations, *submitted*.

Sachs, E. and Schu, M. (2012b). A priori error estimates for reduced order models in finance, *To appear in Mathematical Modelling and Numerical Analysis* .

Sachs, E. and Strauss, A. (2008). Efficient solution of a partial integro-differential equation in finance, *Applied Numerical Mathematics* **58**: 1687–1703.

Said, K. (1999). Pricing exotics under smile, *Risk* **November**: 72–75.

Schoutens, W. (2003). *Lévy-Processes in Finance*, Wiley.

Scott, L. (1973). Finite element convergence for singular data, *Numerische Mathematik* **21**(4): 317–327.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures, part i-iii, *Quarterly of Applied Mathematics* **45**(3): 561 – 571.

Stein, E. M. and Stein, J. C. (1991). Stock price distributions with stochastic volatility: An analytic approach, *Review of Financial Studies* **4**(4): 727–52.

Toint, P. (1988). Global convergence of a class of trust region methods for nonconvex minimization in Hilbert spaces, *IMA Journal of Numerical Analysis* **8**(2): 231–252.

Toivanen, J. (2008). Numerical valuation of European and American options under Kou's jump-diffusion model, *SIAM Journal on Scientific Computing* **30**(4): 1949–1970.

Tröltzsch, F. (2010). *Optimal Control of Partial Differential Equations: Theory, Methods, and Applications*, Graduate studies in mathematics, American Mathematical Society.

Tröltzsch, F. and Volkwein, S. (2009). POD a-posteriori error estimates for linear-quadratic optimal control problems, *Computational Optimization and Applications* **44**(1): 83–115.

Volkwein, S. (2001). Optimal control of a phase-field model using proper orthogonal decomposition, *Zeitschrift für angewandte Mathematik und Mechanik* **81**: 83–97.

Wilmott, P., Dewynne, J. and Howison, J. (1993). *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press.