

# Efficient Numerical Methods for Pricing American Options under Lévy Models

I N A U G U R A L - D I S S E R T A T I O N

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

SEBASTIAN QUECKE

aus Bonn

Hundt Druck GmbH, Köln

2007

Berichterstatter: Prof. Dr. R. Seydel  
Prof. Dr. C. Tischendorf

Tag der mündlichen Prüfung: 18. April 2007

## Abstract

Two new numerical methods for the valuation of American and Bermudan options are proposed, which admit a large class of asset price models for the underlying. In particular, the methods can be applied with Lévy models that admit jumps in the asset price. These models provide a more realistic description of market prices and lead to better calibration results than the well-known Black-Scholes model. The proposed methods are not based on the indirect approach via partial differential equations, but directly compute option prices as risk-neutral expectation values. The expectation values are approximated by numerical quadrature methods. While this approach is initially limited to European options, the proposed combination with interpolation methods also allows for pricing of Bermudan and American options. Two different interpolation methods are used. These are cubic splines on the one hand and a mesh-free interpolation by radial basis functions on the other hand. The resulting valuation methods allow for an adaptive space discretization and error control. Their numerical properties are analyzed and, finally, the methods are validated and tested against various single-asset and multi-asset options under different market models.

## Zusammenfassung

Es werden zwei neue numerische Verfahren zur Bewertung von amerikanischen Optionen und Bermuda-Optionen vorgeschlagen, die eine große Klasse von Aktienkursmodellen für das zugrundeliegende Wertpapier zulassen. Insbesondere können die Verfahren für Lévy-Modelle angewendet werden, die Sprünge in den Aktienkursen modellieren. Diese Modelle sind realitätsnäher und erlauben eine bessere Kalibrierung an Marktdaten als das bekannte Black-Scholes-Modell. Die vorgeschlagenen Verfahren folgen nicht dem indirekten Bewertungsansatz über partielle Differentialgleichungen, sondern berechnen die Optionspreise direkt als risikoneutrale Erwartungswerte. Die Erwartungswerte werden mit numerischen Quadraturmethoden approximiert. Während dieses Vorgehen zunächst auf europäische Optionen beschränkt ist, können durch die Kombination mit Interpolationsverfahren auch Bermuda-Optionen und amerikanische Optionen bewertet werden. Für die Interpolation werden zwei verschiedene Ansätze vorgeschlagen. Zum einen werden kubische Splines verwendet, zum anderen eine gitterfreie Interpolation mit radialen Basisfunktionen. Die resultierenden Bewertungsmethoden erlauben eine adaptive Diskretisierung und damit eine Fehlersteuerung. Ihre numerischen Eigenschaften werden untersucht und schließlich werden die Methoden nach einer Validierung an verschiedenen ein- und mehrdimensionalen Optionen mit unterschiedlichen Marktmodellen getestet.

# Introduction

An *option* is the right, but not the obligation, to do something. In the context of finance, an option contract is a financial instrument, whose value depends on other securities. More specifically, a *European plain-vanilla call* option is a financial instrument that gives its holder the right to buy an asset at a (contractually) pre-specified time for a pre-specified price. In contrast to European options, *American* options are not restricted to a single point in time but to a period, in which the holder is allowed to exercise the option. A third kind of options are *Bermudan* options, which allow a discrete set of exercise times.

Besides plain-vanilla options, more sophisticated contracts exist. Among these are options with non-standard payoffs, options which decay worthless if the price of the underlying crosses a pre-specified barrier, and options on several underlyings. For example basket options grant the right to buy a portfolio of several pre-specified underlyings, and exchange options grant the right to exchange one pre-specified underlying with another.

In 1973, Black, Scholes [BS73], and Merton [Mer73] published their seminal works which provided the first theoretically consistent framework for pricing options. In the Black-Scholes model, the price of the underlying asset is assumed to follow a geometric Brownian motion. This, among other assumptions, allows for a closed-form pricing formula for European plain-vanilla options. The price of American and Bermudan options, however, depends on an optimal exercise strategy and cannot be determined in closed-form. Thus, Bermudan and American option pricing problems naturally lead to numerical methods.

Although widely used, the geometric Brownian motion is not a perfect description of real asset price dynamics. As a result, market option prices can be explained by the Black-Scholes model only in an inconsistent way, namely by using different volatility parameters for different option contracts on the same underlying (“volatility smile”). Independent of the option pricing problem there is also econometric evidence for a systematic underestimation of the probability of large price movements by the Black-Scholes model (“fat tails” of return densities). During the last decade, alternative models received increasing attention by researchers. One particular class of models that allow jumps in the asset price is the class of *Lévy models*, in which the logarithmic asset price is assumed to follow a stochastic Lévy process. The Black-Scholes model can be seen as a special Lévy model without jumps.

The aim of the present work is to develop efficient numerical methods for pricing options with **arbitrary payoffs** and **American** or **Bermudan** exercise structure under **Lévy models** for **one or several underlyings**.

The pricing methods proposed in this work are not based on an indirect approach via partial differential equations, but directly compute option prices as risk-neutral expectation values. The expectation values can be expressed by recursive integrals, where the recursion depth is given by the number of possible exercise times. A straightforward implementation would lead to exponential computation time and thus would be limited to Bermudan options with only few exercise times. However, the proposed combination of numerical quadrature with interpolation methods offers linear computation time and can hence be applied to Bermudan options with high numbers of exercise times. In this sense, they can also provide good approximations to American option values. A convergence result is established and extrapolation techniques are briefly discussed.

For interpolation in the asset price space, two methods are used. These are cubic splines on the one hand and a mesh-free interpolation by radial basis functions (RBF) on the other hand. The corresponding pricing methods, in the following called “spline” and “RBF” method, allow for an adaptive space discretization and error control. Their numerical properties are analyzed, implementations of both methods are validated against independent results, and finally they are tested for various single- and multi-asset options under different market models.

The work is organized as follows. Chapter 1 gives a short introduction to risk-neutral option pricing and a short overview of classical methods. The more closely related literature is discussed in section 2.5. Chapter 2 proposes an approach for pricing Bermudan and American options based on a combination of numerical quadrature and interpolation methods. This approach opens up a new family of valuation methods. Two particular members, the spline method and the RBF method, are introduced and analyzed in chapters 3 and 4. An important prerequisite to the application of both methods is the specification of a market model. Therefore, some current models of Lévy type are introduced in chapter 5. This chapter is placed *behind* the derivation of the new valuation methods intentionally: The methods themselves do not depend on the choice of a particular model. In fact, the market models are to be understood as “plug-ins” for the two valuation methods. All properties that characterize market models which can be employed by the proposed methods are summarized in form of assumptions in section 2.1. The structural separation of methods and models seems to be unique to this approach. In chapter 6 the methods are applied to a variety of option types to demonstrate their flexibility. Where possible, comparisons to results from other methods are included for validation purposes. Chapter 7 contains a concluding discussion and an outlook to interesting open problems.

At this point I would like to thank my academic advisor Prof. Dr. R. Seydel for his continuous support, for many helpful suggestions, and for providing an excellent research environment. I thank Prof. Dr. C. Tischendorf for serving as a referee for my thesis. Finally, I wish to thank my colleagues for lively discussions and their support.

# Contents

## Introduction

<b>1. Overview of option pricing</b>	<b>1</b>
1.1. Options . . . . .	1
1.2. Arbitrage-free pricing . . . . .	4
1.3. Classical valuation methods . . . . .	7
1.4. Motivation . . . . .	9
<b>2. Quadrature based valuation of Bermudan and American options</b>	<b>10</b>
2.1. Assumptions about the market . . . . .	10
2.2. Pricing European options with arbitrary payoffs . . . . .	13
2.3. Pricing Bermudan options by quadrature . . . . .	14
2.4. Connection to American options . . . . .	19
2.5. Related methods . . . . .	25
2.6. Convolution-based methods . . . . .	29
<b>3. The spline method</b>	<b>30</b>
3.1. Spline interpolation . . . . .	30
3.2. Adaptive quadrature . . . . .	33
3.3. Construction of the valuation method . . . . .	35
3.4. Numerical properties . . . . .	43
3.5. Generalization to multi-asset options . . . . .	58
3.6. Conclusion . . . . .	64
<b>4. The RBF method</b>	<b>65</b>

---

4.1. Motivation . . . . .	65
4.2. Interpolation by radial basis functions . . . . .	68
4.3. Quadrature rules for the model matrix . . . . .	81
4.4. Numerical properties . . . . .	83
4.5. Conclusion . . . . .	87
<b>5. Asset price models</b>	<b>89</b>
5.1. One-factor models . . . . .	89
5.2. Multi-factor models . . . . .	106
5.3. Non-parametric modeling . . . . .	108
<b>6. Numerical examples</b>	<b>110</b>
6.1. Validation . . . . .	110
6.2. Single-asset options . . . . .	119
6.3. Multi-asset options . . . . .	126
6.4. Comparisons . . . . .	140
6.5. Further range of application . . . . .	141
<b>7. Conclusion</b>	<b>143</b>
<b>A. Appendix</b>	<b>145</b>
A.1. Notation . . . . .	145
A.2. Convolution-based methods . . . . .	146
A.3. Implementation notes . . . . .	149
<b>Bibliography</b>	<b>155</b>
<b>Index</b>	<b>161</b>

# 1. Overview of option pricing

This chapter describes the current methodology for option pricing. Section 1.1 introduces the basic terminology, section 1.2 contains a brief summary of the concept of arbitrage-free pricing, and section 1.3 gives an overview of classical numerical pricing methods. A list of symbols and notation used throughout this work can be found in appendix A.1.

## 1.1. Options

Options occur in various types and specifications. The lowest common denominator for a definition is that an option is a financial instrument whose value depends on other securities.<sup>1</sup> Emphasizing this dependence, options are also called *derivatives*. The most frequently used options are the “plain-vanilla” options defined as follows.

**Definition 1.1** (Plain-vanilla option). *A plain-vanilla option is a contract giving its holder the right, without the obligation, to either buy or sell an underlying asset at a predetermined price  $K$  (the exercise or strike price) up to a specified expiration date  $T$ . If the right granted is to buy, the option is a call option. If the right granted is to sell, the option is a put option.*

The underlying asset is typically a stock. If the holder uses his right to sell (or buy) the underlying from (or to) the issuer, he *exercises* the option. Options which can be exercised at any time until expiry are called *American* options. There are also contracts which cannot be exercised at any time  $t \in [0, T]$  but only at *maturity*  $T$ . Those options are called *European* options. Options that can be exercised at a (pre-specified) discrete set of times  $\mathcal{T} = \{t_1, \dots, t_m\}$  are called *Bermudan* options. Today the most frequently traded options are of American type ( $\mathcal{T} = [0, T]$ ).

Although definition 1.1 fits well the put and call contracts that are traded most commonly at the financial markets, for this work it is suitable to use the following, more general definition.

**Definition 1.2** (Option). *Let  $x = x(t) \in \mathbb{R}^d$  denote the price of the underlying at time  $t \in [0, T]$ , let  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  be a function of the price of the underlying, and let  $\mathcal{T} \subset [0, T]$*

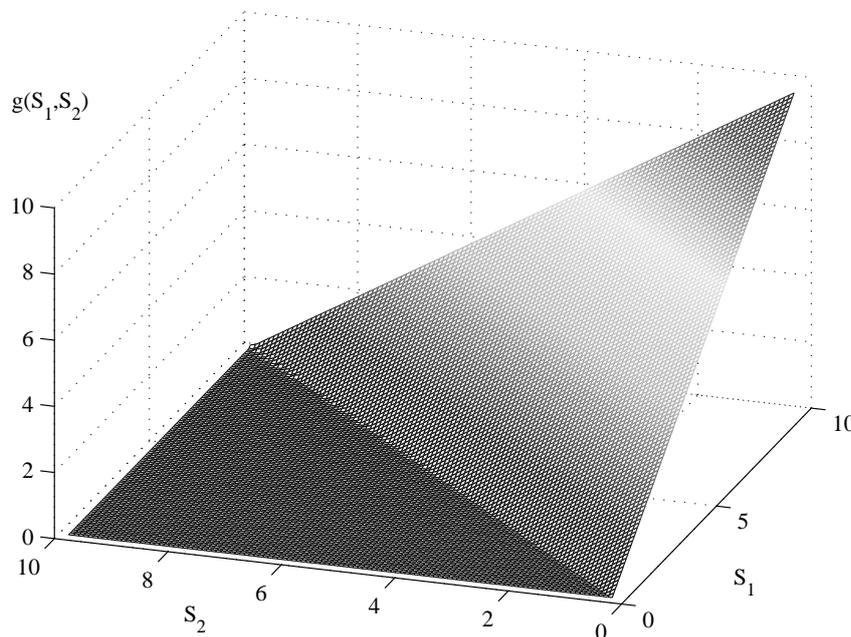
---

<sup>1</sup>The option value may also depend on values and events which are only *observable*. These observable items do not need to be market prices. They can be e.g. weather data. This work focuses on underlyings that can be described by the asset price models presented in chapter 5; and these are typically stocks.

be a set of times. An option with payoff  $g$  and exercise structure  $\mathcal{T}$  is a contract that gives the holder the right to exercise the option at a time  $t \in \mathcal{T}$  and thus to receive the payoff value  $g(x(t))$  from the issuer.

It is convenient to use the log-price  $x := \log S$  instead of the price  $S$  of the underlying. This notation is used throughout this work. Furthermore, functions that depend on the price of the underlying, e.g., the payoff, are sometimes regarded as functions of  $x$  and sometimes as functions of  $S$ . It should be clear from the context, which notation is used. How does definition 1.2 relate to 1.1? For an option on a single stock is  $d = 1$  and  $x = \log S$ , where  $S$  is the price of the stock. In the case of a put option the payoff is  $g(x) := (K - S)^+ = (K - e^x)^+$ . This agrees with definition 1.1. Obviously the first definition is a special case of the second one, admitting solely the payoffs  $g(x) := (S - K)^+$  for call options and  $g(x) := (K - S)^+$  for put options. Furthermore, definition 1.2 is not restricted to the one dimensional case.<sup>2</sup> For example, for a two-asset exchange option is  $d = 2$  and  $x = (x_1, x_2) = (\log S_1, \log S_2)$ , where  $S_1$  and  $S_2$  are the prices of the two underlyings. In this case the payoff is  $g(x) := (c_1 S_1 - c_2 S_2)^+$  with parameters  $c_1$  and  $c_2 \in \mathbb{R}$  describing the amounts of both assets that can be exchanged by exercising the option. An illustration is given in figure 1.1.

Figure 1.1.: Payoff  $g(S_1, S_2) = (S_1 - S_2)^+$  of a two-asset exchange option.



<sup>2</sup>In this work the “dimension” of an option refers to the dimension of the stochastic process describing the underlying. Time is not counted as a dimension. This can lead to confusions: For example, the Black-Scholes equation for *one-dimensional* European plain-vanilla options is a *two-dimensional* PDE.

**Remark 1.3** (Path independence). *Definition 1.2 defines a path-independent option, i.e. an option whose payoff only depends on the price of the underlying at exercise time and not on the path of the price until exercise time. This work is restricted to path-independent options. A possible application to Asian (i.e. path-dependent) options is mentioned later in section 6.5.*

**Remark 1.4** (Bermudan/American options). *The terms Bermudan and American are not used consistently throughout the literature. One reason is that the American option can be seen as the limit case of Bermudan options with  $m$  equidistant exercise times for  $m \rightarrow \infty$  (see section 2.4). Another aspect is that American options are much more important than Bermudan options with respect to trading volume. Some authors use the weaker term “American-style security” or “option with early exercise feature” instead of “Bermudan”. In this work the terms are used precisely.*

## The price of an option

As the issuer of an option takes an obligation, it is clear that he has to be paid for signing the contract. If this premium – the option price – is very high, many investors want to sell the option. If the price is very low, many want to buy the option. The price-dependent demand leads to a *market price* characterized by an equilibrium of sellers and buyers.

One approach to derive a *fair price* of an option is to construct a hedging portfolio, i.e., a portfolio that replicates the value of the option perfectly. Then, following the *no-arbitrage principle*<sup>3</sup>, the costs for every hedging portfolio must be equal and thus give a fair price of the option. This approach has already been used in 1973 by Merton [Mer73] and is known as *dynamic replication*. The possibility to represent every contingent claim as the final value of a self-financing strategy characterizes *complete* markets.<sup>4</sup> The market model used by Black and Scholes in [BS73] is complete and consequently every option under this model has a uniquely determined fair price. Unfortunately, real markets are *incomplete*, as well as the most of the newer market models, e.g., models with jumps. This is not surprising as market completeness is an unstable property. Given a complete market model, the addition of a small jump risk destroys its completeness.<sup>5</sup>

A weaker alternative to the assumption of completeness is the assumption of absence of arbitrage. This is a reasonable assumption for any market model and is – in contrast to market completeness – not considered an artificial restriction.

**Remark 1.5** (Rational bounds). *One can derive “rational” bounds for option values from the no-arbitrage principle. Such bounds can be found, e.g., in [Kwo98]. They are more useful for validation purposes than to determine the price of an option exactly.*

<sup>3</sup>The no-arbitrage principle is the assumption that the market is free of arbitrage opportunities, i.e., possibilities to make a risk free profit.

<sup>4</sup>This definition of complete models can be found in [Shr04], p. 231.

<sup>5</sup>See [CT03], p. 319.

## 1.2. Arbitrage-free pricing

In order to outline the basic methodology for pricing in arbitrage-free markets, this section gives a brief overview following [CT03]. A rigorous treatment of option pricing theory is rather technical and goes beyond the scope of this work. It can be found in the original literature [HK79], [HP81], [HP83], [DS94], and [DS98]. It is assumed that the reader is familiar with the usual notation from probability theory. A good introduction to probability theory in the context of finance can be found in [Shr04].

### Connection between pricing rules and probability measures

Let  $(\Omega, \mathcal{F})$  be the sample space describing the possible market scenarios in the time period  $[0, T]$ . Let  $(\mathcal{F}_t)_{t \in [0, T]}$  denote the filtration generated by the market history up to time  $t$ . The prices of underlying assets may then be described by a non-anticipating process<sup>6</sup>

$$\begin{aligned} S : [0, T] \times \Omega &\rightarrow \mathbb{R}^d \\ (t, \omega) &\mapsto (S_t^1(\omega), \dots, S_t^d(\omega)). \end{aligned}$$

In this case any European contingent claim with maturity  $T$  can be fully described by specifying its terminal payoff  $G(\omega)$  for each scenario  $\omega \in \Omega$ . For example, for the European plain-vanilla put this is  $G = (K - S_T)^+$ . A *pricing rule*  $\Pi$  is a map which attributes to each contingent claim  $G$  a value  $\Pi_t(G)$  at each point in time. Obviously, any reasonable pricing rule needs to fulfill some technical requirements:

- (i)  $\Pi_t(G)$  should be non-anticipating, i.e., the value  $\Pi_t(G)$  can be determined without information about the future market development.
- (ii)  $\Pi$  should be positive, i.e., a non-negative payoff should have a non-negative value.
- (iii)  $\Pi$  should be additive:  $\Pi_t(\sum_{i \in I} G_i) = \sum_{i \in I} \Pi_t(G_i)$ ,  $I$  being an arbitrary index set.

For any event  $A \in \mathcal{F}$ , the random variable  $\mathbf{1}_A$  is a payoff of a contingent claim, which pays 1 at  $T$ , if  $A$  occurs and 0 otherwise. In particular  $\mathbf{1}_\Omega$  corresponds to a zero coupon bond<sup>7</sup> paying 1 at time  $T$ . Assuming a constant discount factor  $r$ , its value is  $\Pi_t(\mathbf{1}_\Omega) = e^{-r(T-t)}$ . Now the definition

$$\begin{aligned} \mathbb{Q} : \mathcal{F} &\rightarrow \mathbb{R} \\ A &\mapsto e^{rT} \Pi_0(\mathbf{1}_A) \end{aligned}$$

<sup>6</sup>A stochastic process  $(S_t)$  is *non-anticipating* with respect to the filtration  $(\mathcal{F}_t)$  if the random variable  $S_t$  is  $\mathcal{F}_t$ -measurable for every  $t$ .

<sup>7</sup>A *zero coupon bond* is a bond which does not pay periodic interests (coupons).

yields a probability measure on the scenario space  $(\Omega, \mathcal{F})$ . Conversely, every probability measure  $\mathbb{Q}$  yields a pricing rule  $\Pi$  by setting

$$\Pi_0(G) := e^{-rT} \mathbb{E}^{\mathbb{Q}}(G) \quad (1.1)$$

for random payoffs of the form  $G = \sum c_i \mathbf{1}_{A_i}$  and extending this measure (under an additional continuity property of  $\Pi$ ) to arbitrary random payoffs. This short motivation indicates the connection between pricing rules and probability measures. It is important that the measure  $\mathbb{Q}$  does not describe the actual probability of market scenarios, but can be used for pricing via (1.1).

## Arbitrage-free pricing rules and martingale measures

Any reasonable pricing rule should be free of arbitrage opportunities, i.e., the resulting prices should not allow arbitrage. The absence of arbitrage is also called “no free lunch with vanishing risk” condition.<sup>8</sup> Technically, arbitrage can be defined as follows.<sup>9</sup>

**Definition 1.6** (Arbitrage). *An arbitrage is a value process  $V(t)$  of a portfolio that is managed using a self-financing strategy with  $V(0) = 0$  and for some time  $t > 0$*

$$\mathbb{P}[V(t) \geq 0] = 1 \quad \text{and} \quad \mathbb{P}[V(t) > 0] > 0.$$

Further analysis of the connection between the measures  $\mathbb{Q}$  and pricing rules  $\Pi$  reveals that arbitrage-free pricing rules are connected to martingale measures, also called *risk-neutral* measures.

**Definition 1.7** (Equivalent martingale measure). *A probability measure  $\mathbb{Q}$  is said to be an equivalent martingale measure (EMM) to  $\mathbb{P}$  if*

- (i)  $\mathbb{Q}$  is equivalent to  $\mathbb{P}$ , i.e.,  
 $\mathbb{Q} \sim \mathbb{P} := \Leftrightarrow \forall A \in \mathcal{F} : \mathbb{Q}(A) = 0 \Leftrightarrow \mathbb{P}(A) = 0$ , and
- (ii) the discounted stock prices  $e^{-rt} S_t^i$  are martingales under the measure  $\mathbb{Q}$ , i.e.,  
 $\forall i = 1, \dots, d : \mathbb{E}^{\mathbb{Q}}(e^{-rT} S_T^i | \mathcal{F}_t) = e^{-rt} S_t^i$ .

The following result establishes a link between arbitrage-free pricing rules and equivalent martingale measures.<sup>10</sup>

**Theorem 1.8** (Risk-neutral pricing). *In a market described by a probability measure  $\mathbb{P}$  on scenarios, any arbitrage-free linear pricing rule  $\Pi$  can be represented as*

$$\Pi_t(G) = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}}(G | \mathcal{F}_t), \quad (1.2)$$

where  $\mathbb{Q}$  is an equivalent martingale measure to  $\mathbb{P}$ .

<sup>8</sup>For example, the “NFLVR” condition in [DS98] is a no-arbitrage condition with subtle modifications (p. 467).

<sup>9</sup>See [Shr04], p. 230.

<sup>10</sup>See [CT03], p. 298.

This theorem is central to the approach used in this work. The following chapters assume that a martingale measure  $\mathbb{Q}$  describing the pricing rule is given and  $\Pi_t(G)$  is numerically approximated via the discounted expectation value in (1.2). A more general theorem establishing a one-to-one correspondence between arbitrage-free pricing rules and equivalent martingale measures is known as *first fundamental theorem of asset pricing*. Unfortunately, it is “quite hard to give a precise version of this theorem”.<sup>11</sup> Details can be found in [DS94].

**Remark 1.9** (Discounting). *In this work a constant discount rate is assumed to simplify notation. In a more general setting the discount factor  $e^{-rt}$  must be replaced by a general discount factor  $D(t)$ .*

## Implications of market (in)completeness

The following result gives an interesting characterization of complete markets. It is also known as the second fundamental theorem of asset pricing.<sup>12</sup>

**Theorem 1.10** (Second fundamental theorem of asset pricing). *Consider a market model that has a risk-neutral probability measure. The model is complete if and only if the risk-neutral probability measure is unique.*

This implies that in complete markets there is only one possible (arbitrage-free) choice for option prices. On the other hand in incomplete markets option prices are not uniquely determined until an equivalent martingale measure is chosen. At first sight this seems not to be a favorable characteristic of incomplete market models, but in fact this only reflects that complete models systematically underestimate the risk inherent in writing an option. In reality as well as in incomplete market models, perfect hedges do not exist.

## Choice of an equivalent martingale measure

Incomplete market models require either the choice of the equivalent martingale measure or the use of nonlinear pricing rules. Some approaches that can be found in the literature are the following.

- (i) **Drift correction:** The drift of the Brownian motion is changed, but all other ingredients are left unchanged. This choice has been proposed by Merton in [Mer76].
- (ii) **Optimal measure:** It is possible to choose the martingale measure by solving an optimization problem. The objective function can, e.g., measure the deviation of  $\mathbb{Q}$  from  $\mathbb{P}$ . Such a measure is discussed in [FM03].

---

<sup>11</sup>See [CT03], p. 299.

<sup>12</sup>See [Shr04], p. 232.

- (iii) **Super-hedging:** The option price is chosen to be the cost of the cheapest hedging strategy that surely dominates the payoff of the option. This approach is discussed in [Kra96]. It leads to nonlinear pricing rules as the worst scenario is different for each option. Thus, the pricing rules cannot be described by a martingale measure in the sense of theorem 1.8.
- (iv) **Indifference pricing:** An investor is assumed to be an expected utility maximizer (equipped with a utility function) and the indifference price for the investor is chosen as option price. This approach replaces the choice of a martingale measure by the choice of a utility function. Most utility functions lead to nonlinear pricing rules.<sup>13</sup>

While approaches (i) and (ii) lead to *linear* pricing rules, approaches (iii) and (iv) lead to *nonlinear* pricing rules. It is evident that most market participants assume linear pricing rules.<sup>14</sup> Therefore, this work also assumes a linear pricing rule and, accordingly, that the pricing rule can be described by an equivalent martingale measure. The corresponding EMM is chosen by the drift correction approach (i), which leads to the *martingale conditions* in chapter 5. The latter choice is not a restriction of the methods proposed in this work. They can be used for arbitrary choices of the EMM.

### 1.3. Classical valuation methods

This section gives a short overview of existing (classical) numerical pricing methods in order to point out the difference between these methods and the methods proposed in this work. Classical methods mostly assume that the market can be described by the famous Black-Scholes model (defined in section 5.1.2). Some of them can be extended to cover incomplete market models. Without going into details, the numerical approaches to option pricing can roughly be divided into the following classes:

- (i) **PDE based methods:** An application of Itô's lemma to a hedging portfolio can be used to derive a partial differential equation (PDE) that describes the risk-neutral price of European options. The price of American options can be represented as solution of a linear complementarity problem. Methods that are based on the solution of these problems can be called *PDE based* methods. Among these methods are finite differences and finite-element methods and their generalizations. Market models with jumps lead to partial integro-differential equations (PIDE) for European options and partial integro-differential inequalities (PIDI) for American options.<sup>15</sup> A resulting PIDI based numerical method for American options is proposed in [MNS05]. A PIDE based method that uses an additional penalty term

<sup>13</sup>The only utility function that leads to linear pricing rules is  $u(x) = -x^2$  (i.e. quadratic hedging).

<sup>14</sup>If a single option is sold for  $V$ ,  $n$  options of the same type are usually sold for  $nV$ .

<sup>15</sup>See e.g., [BL84].

to enforce the optimal exercise strategy is discussed in the working paper [CF05]. The main advantage of PDE based methods is their accuracy and that they are deterministic. PDE methods are based on a rich theoretical foundation.

- (ii) **Tree methods:** Usually, tree methods construct a binomial or trinomial tree that approximates the evolution of the price of the underlying and then trace back in time the risk-neutral value of the option. Supported by the central limit theorem, this approach is fast and easy to implement for the Black-Scholes model. As it is difficult to construct trees for more general market models, tree methods seem to be limited to the Black-Scholes setting. The first binomial tree method has been proposed in [CRR79]. A more recent approach calibrating trees to implied volatilities is proposed in [DK94].
- (iii) **Monte Carlo methods:** Monte Carlo methods randomly draw a large number of elements  $\{\omega_1, \dots, \omega_n\}$  from the sample space  $\Omega$ , evaluate the realized payoff  $G(\omega)$  for each scenario and take the mean as estimate for the discounted expectation value in (1.2). It is evident that this blueprint relies on an efficient numerical evaluation of  $G(\omega_i)$ . While this is possible for European options, it is not for Bermudan and American options, whose value depends on the optimal exercise strategy. Approaches to make Monte Carlo viable for Bermudan options have been proposed in [BG97], [BG04] and [LS01]. All of these extensions estimate the hold value of the option<sup>16</sup> to decide, whether (and when) it is optimal to exercise the option. Concerning the model of the underlying, Monte Carlo methods are surely the most flexible methods for option pricing.<sup>17</sup> Another advantage is that they allow a straightforward implementation in cases where  $G(\omega_i)$  can be evaluated directly. On the other hand, there are numerous drawbacks, from low rates of convergence over discontinuous price surfaces to being non-deterministic and thus providing only confidence intervals for option prices.<sup>18</sup> Nevertheless, for high dimensional or technically involved models of the underlying, Monte Carlo methods are often the only available methods.
- (iv) **“Analytical” methods:** In various cases a *closed-form*<sup>19</sup> solution of the option pricing problem is available. The most prominent example is the Black-Scholes formula for European options. A collection of further “pricing formulas” can be

<sup>16</sup>The hold value  $V^H(S, t_i)$  of a Bermudan option at a possible exercise time  $t_i$  is the value that the option had if it could not be exercised at  $t_i$  (see definition on p. 14).

<sup>17</sup>The pure simulation of trajectories for an arbitrary model is usually not a problem.

<sup>18</sup>Various extensions of Monte Carlo methods exist but cannot be discussed here. For example, using quasi random numbers can improve the rate of convergence.

<sup>19</sup>The question, what a “closed-form solution” is, is a bit philosophical. The Black-Scholes formula involves the Gauss error function; other pricing formulas involve infinite series (e.g. the formula for European plain-vanilla option prices under the Merton model in [Hau97]) or improper integrals (e.g. Heston’s “closed-form solution” from [Hes93]).

found in [Hau97]. Typically, closed-form solutions exist only for European options. Another collection of methods that is labeled “analytical” is the collection of analytically tractable approximations to option pricing problems, e.g. [GJ84]. Such methods can provide (rough) approximations of the value of Bermudan or American options.

Of course this classification is not complete, nor does it fit every valuation method. In particular the methods proposed in this work do not fall into one of the above classes. They can be called *quadrature based methods*. Before this class is introduced in detail, the following section gives a short motivation for a new class of methods in form of construction goals. An overview of the more closely related literature and a fine classification of the methods proposed in this work is deferred to section 2.5.

## 1.4. Motivation

This section sets the goals for a new valuation approach for American options. Desirable properties of a valuation method for American options are the following:

- (i) The method is flexible with respect to the model for the underlying. In particular it can price options under exponential Lévy models.
- (ii) The model for the underlying is separated from the valuation algorithm and can be easily exchanged.
- (iii) The method is deterministic.
- (iv) The method is able to price American/Bermudan options on several underlyings.
- (v) The accuracy of the solution can be controlled.

These properties are to be fulfilled by the methods proposed in chapters 3 and 4. The only “classical” methods that can meet all these goals are the most recent, technically demanding PIDE based methods, e.g., the extended finite difference scheme proposed in [CF05].

## 2. Quadrature based valuation of Bermudan and American options

This chapter introduces the quadrature approach for pricing Bermudan options. It is the basis for the pricing methods proposed in chapters 3 and 4. After specifying general assumptions about the market in section 2.1, a method for pricing European options with arbitrary payoffs is presented in section 2.2. Based on this method, a framework for the valuation of Bermudan options is introduced in section 2.3. It can also be used to approximate the value of American options, which is discussed in section 2.4. Finally, section 2.5 gives an overview of the most recent related literature and thus discusses the current state of research in quadrature methods. A short digression in section 2.6 introduces a convolution-based approach for Bermudan options, which is a possible alternative to the quadrature approach and compares both approaches qualitatively.

### 2.1. Assumptions about the market

Instead of sticking to a particular market model, quadrature methods can be based on a few, rather general assumptions on the structure of the underlying market model. These assumptions and their consequences are discussed in this section. They are met by the Black-Scholes model but admit also a wide range of other market models. Some specific models of Lévy type are presented later in chapter 5.

**Assumption 2.1** (Equivalent martingale measure). *Let the market model be described by a filtered probability space  $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_t)$  and let  $(X_t) = (\log S_t)$  denote the stochastic process of log-prices. It is assumed that the pricing rule can be described by an equivalent martingale measure  $\mathbb{Q}$ .*

This central assumption does not seem to bear severe restrictions. By theorem 1.8 it is satisfied for any arbitrage-free linear pricing rule.

**Assumption 2.2** (Conditional probability density). *It is assumed that  $X_t$  has a conditional probability density with respect to  $X_s$  for every  $t > s$ . The conditional probability density function (PDF) is denoted by  $f = f^{X_t|X_s=x}(\xi)$ . It is furthermore assumed that  $f$  can be evaluated numerically in an “efficient” way.*

Assuming the pure existence of a probability density function is not a restriction in practice. For instance in the one-dimensional case every distribution with absolutely

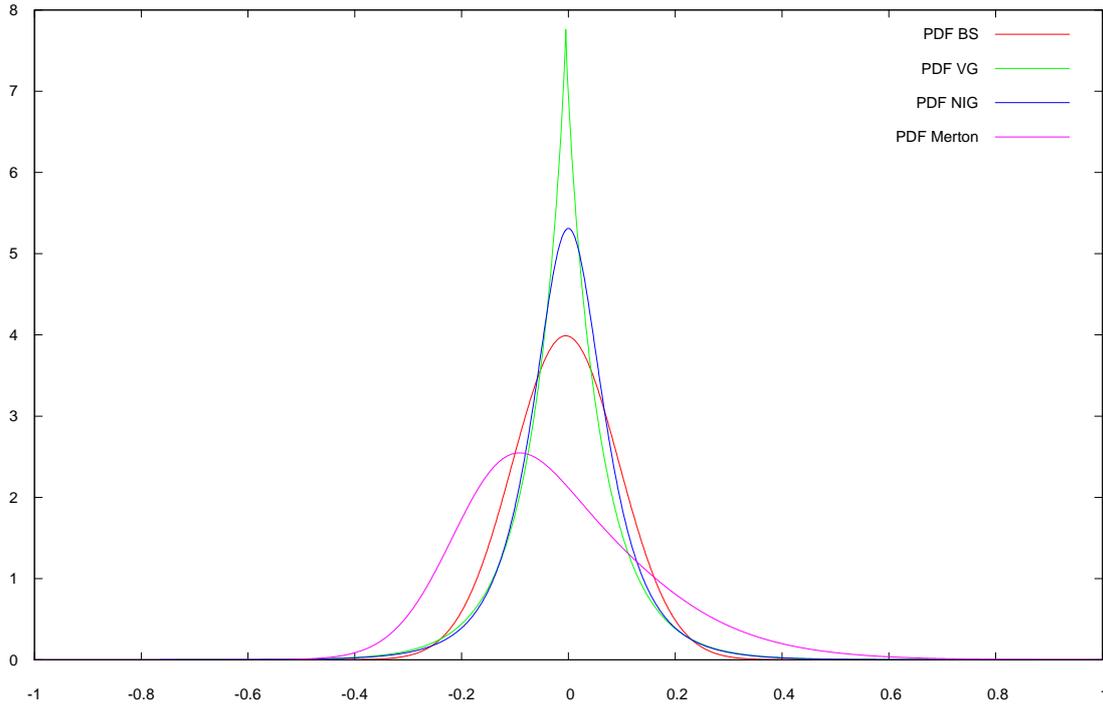


Figure 2.1.: Conditional probability density functions  $f^{X_t|X_0=0}(\xi)$  for different models. The parameter values used for this plot are given in table 5.1 (p. 106).

continuous cumulative distribution function  $F(x)$  has a density: Then,  $F$  is almost everywhere differentiable, and its derivative is the probability density of the distribution. Examples for models that have conditional probability densities are the Black-Scholes model, variance gamma (VG), normal inverse Gaussian (NIG), and the Merton model.

As the density function is later used for the computation of option prices, it is furthermore necessary that it allows for an efficient numerical evaluation. The models introduced in chapter 5 possess such density functions; but in general this could impose restrictions on the market model.<sup>1</sup>

**Remark 2.3** (Density function/Characteristic function). *For Lévy processes assumption 2.2 can be replaced by the assumption that the conditional characteristic function of the Lévy process allows for efficient numerical evaluation. Both functions are related and, e.g., O’Sullivan demonstrates in [O’S05] that the probability density function can be computed efficiently out of the characteristic function via fast Fourier transform. The method is further discussed in section 2.5.*

<sup>1</sup>An example for a stochastic process without known closed form probability density is the “tempered stable process”.

**Assumption 2.4** (Convergence to Dirac delta function). *It is assumed that the conditional probability density function  $f^{X_t|X_s=x}$  converges to the Dirac delta function  $\delta_x$  for  $t \rightarrow s$  in the sense that pointwise*

$$\lim_{t \rightarrow s} \int_{\mathbb{R}^d} f^{X_t|X_s=x}(\xi) g(\xi) d\xi = g(x) \quad \text{for all } x \in \mathbb{R}^d$$

for every continuous function  $g \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$  (such that the arising integrals exist).

This assumption is required for the proof of convergence of the Bermudan option value to the American option value (lemma 2.14). It is not needed for the plain valuation of a Bermudan option with an arbitrary but fixed number of exercise times. For the Black-Scholes model, where  $f$  is the normal density function, this assumption is obviously fulfilled. For the more general class of Lévy processes, it is guaranteed by the stochastic continuity property.<sup>2</sup>

**Assumption 2.5** (Space-homogeneity, optional). *The price process is assumed to be space-homogeneous, i.e.*

$$\forall c \in \mathbb{R}^d, 0 \leq s \leq t \leq T : f^{X_t+c|X_s+c=0} \equiv f^{X_t|X_s=0}. \quad (2.1)$$

**Lemma 2.6.** *Under assumption 2.5 the translation of the random variable  $X_t$  by  $c$  can be passed over to the argument:*

$$\forall \xi \in \mathbb{R}^d : f^{X_t-c|X_s=0}(\xi) = f^{X_t|X_s=0}(\xi + c), \quad (2.2)$$

as the probability densities for  $X_t - c = \xi$  and  $X_t = \xi + c$  are the same. Consequently, assumption 2.5 implies

$$f^{X_t|X_s=c} \equiv f^{X_t+c|X_s=0}.$$

*Proof.*  $\forall \xi : f^{X_t|X_s=0}(\xi) \stackrel{(2.1)}{=} f^{X_t-c|X_s-c=0}(\xi) = f^{X_t-c|X_s=c}(\xi) \stackrel{(2.2)}{=} f^{X_t|X_s=c}(\xi + c)$   
 $\Rightarrow \forall \zeta := \xi + c : f^{X_t|X_s=c}(\zeta) = f^{X_t|X_s=0}(\zeta - c) \stackrel{(2.2)}{=} f^{X_t+c|X_s=0}(\zeta). \quad \square$

The (optional) assumption of space-homogeneity of the log-price seems very natural in the context of stock prices. It means that the conditional distribution of prices is invariant under re-scaling (e.g., switching from a dollar-based quote to a cent-based quote).

To illustrate that this assumption is natural, if the underlying is a stock, consider two scenarios  $S_0 = 8\$$  and  $S_0 = 16\$$ . The assumption of space-homogeneity implies that for both scenarios the probability of the price being at least doubled in a given time interval is equal: In the first scenario we have  $X_0 = \log_2 S_0 = 3$ , in the second one  $X_0 = 4$ . Space-homogeneity says  $F^{X_T|X_0=4} = F^{X_T-1|X_0=5}$ . Thus, the probability for  $X_T \geq 4$

<sup>2</sup>See definition 5.1, property (iii): “ $\forall \varepsilon > 0 : \lim_{\Delta t \rightarrow 0} \mathbb{P}(|X_{t+\Delta t} - X_t| \geq \varepsilon) = 0$ ”.

( $S_T \geq 16\$ = 2S_0$ ) in the first scenario is equal to that for  $X_T \geq 5$  ( $S_T \geq 32\$ = 2S_0$ ) in the second scenario.

This assumption is labeled “optional,” as it is not necessary for the following methods. It is useful as it leads to simpler notation, allows for considerable speed-up in some special cases (see section 4.4.2), and is necessary for the convolution-based approach in section 2.6. The assumption holds for many important models, in particular for those proposed in chapter 5.

**Assumption 2.7** (Minor conventions). *Besides the assumptions above, some minor conventions simplify the notation. They are to be considered preliminary, as they are not technically required.*

- (i) *A process  $X_t$  has stationary increments, if  $f^{X_t|X_{t+\Delta t}=x} \equiv f^{X_s|X_{s+\Delta t}=x}$  for all times  $s, t \in \mathbb{R}$ , time increments  $\Delta t > 0$ , and log-prices  $x \in \mathbb{R}^d$ . The price process is assumed to have stationary increments.*
- (ii) *A constant discount rate  $r > 0$  is assumed. Throughout this work, discount factors are expressed by  $e^{-r\Delta t}$  for every time period of length  $\Delta t \geq 0$ .*
- (iii) *The underlying pays no dividend. This is only for notational convenience. Both discrete and continuous dividends can be incorporated easily.*

**Remark 2.8** (Beyond Lévy processes). *Two important properties of Lévy processes are the stationarity and independence of increments.<sup>3</sup> It is important to note that both restrictions are not necessary for the quadrature methods proposed in this work. Assumptions 2.5 and 2.7(i) are made only for notational convenience. An example for a process which has neither stationary nor independent increments is the CIR process.*

## 2.2. Pricing European options with arbitrary payoffs

The first step in the construction of valuation methods for Bermudan options is the valuation of European options with arbitrary payoffs (as in definition 1.2) under the assumptions from section 2.1.

**Lemma 2.9** (Integral representation of European option prices). *Let  $V(x, t)$  denote the fair value of a European option on the underlying  $X_t$  maturing at  $T$ . Let  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  denote its (arbitrary) payoff function. Under the assumptions above the fair price is*

$$V(x, 0) = e^{-rT} \int_{\mathbb{R}^d} f^{X_T|X_0=x}(\xi) g(\xi) \, d\xi, \quad (2.3)$$

where  $f$  denotes the conditional probability density function.

<sup>3</sup>Lévy processes are defined later in definition 5.1 (p. 90).

*Proof.* Following the risk-neutral pricing theorem 1.8, the price under the equivalent martingale measure from assumption 2.1 can be written as

$$\Pi_0(G) = e^{-rT} \mathbb{E}^{\mathbb{Q}}(G | \mathcal{F}_0).$$

In the case of a (path-independent) European option and fixed start price  $X_0 := x$  this reduces to a discounted conditional expectation of the terminal payoff  $g(X_T)$ . Following assumption 2.2 this expectation can be expressed as the weighted integral above.  $\square$

It is possible to approximate this integral numerically and thus to compute the fair value of any European option with arbitrary payoff  $g$  in any market model satisfying the assumptions above. Methods based on this approach can be called “quadrature based methods for European options”.

**Remark 2.10** (Black-Scholes formula). *For a plain-vanilla European option in a Black-Scholes setting, one can rewrite the integral using the Gauss error function<sup>4</sup>. This leads to the well-known analytic Black-Scholes formula.*

**Remark 2.11** (Arbitrary payoff and density). *The payoff  $g$  does not need to be continuous or even smooth. The integral in equation (2.3) has to be finite, but this is not a restriction in practice. The density function  $f$  prescribed by the model is usually smooth. In cases of non-smooth functions  $g$  or  $f$  the numerical quadrature procedure has to decompose the domain  $\mathbb{R}^d$  according to the discontinuities to retain accuracy.*

## 2.3. Pricing Bermudan options by quadrature

This section describes the construction of a valuation procedure for Bermudan options out of a given valuation procedure for European options with arbitrary payoffs.

### Reduction to European options with arbitrary payoffs

The first idea is to consider the Bermudan option piecewise as European options with special payoffs. It leads to the following representation of the Bermudan option value, which is also called *Snell envelope*.

**Lemma 2.12** (Reduction principle). *Let  $V(x, t)$  be the value of a Bermudan option with  $m$  discrete exercise times  $t_1 = 0, \dots, t_m = T$ . Under the assumptions above the value of the Bermudan option at each exercise time  $t_i$ ,  $1 \leq i < m$  can be written as*

$$\begin{aligned} V(x, t_i) &= \max \left( g(x), V^H(x, t_i) \right), \text{ with} \\ V^H(x, t_i) &:= e^{-r(t_{i+1}-t_i)} \int_{\mathbb{R}^d} f^{X_{t_{i+1}} | X_{t_i}=x}(\xi) V(\xi, t_{i+1}) \, d\xi, \text{ (“hold value”)} \end{aligned}$$

and  $V(x, t_m) := g(x)$  at maturity.

<sup>4</sup>The Gauss error function is defined e.g. in [AS65], p. 297.

*Proof.* In each interval  $[t_i, t_{i+1}]$  the option can only be exercised at  $t_i$  and  $t_{i+1}$ . Thus in the interval  $(t_i, t_{i+1}]$  the value of the Bermudan option equals the value of a European option with payoff  $g(x) := V(x, t_{i+1})$  at  $t_{i+1}$ , which can be written in integral form (2.3). At  $t_i$  the option can be exercised. The optimal exercise strategy is to exercise, if and only if the value of the European option at  $t_i$  is below the payoff and otherwise hold the option. Thus the value at  $t_i$  is the maximum of payoff and European option price.  $\square$

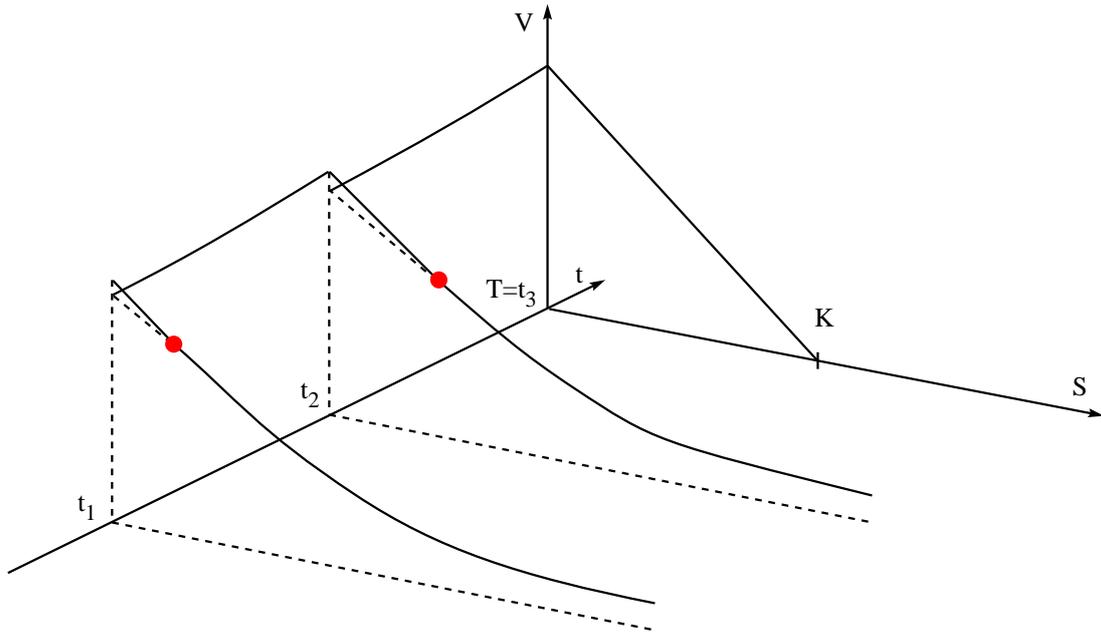


Figure 2.2.: Bermudan plain-vanilla put option sliced into several European options

**Example 2.13.** Figure 2.2 illustrates the value of a Bermudan put option with exercise times  $t_1, t_2$ , and  $t_3$ . The hold value  $V^H(S, t_2)$  at time  $t_2$  (dotted/solid line) corresponds to the value of a European plain vanilla put with payoff  $g(S) := (K - S)^+$  at  $T = t_3$ . As at  $t_2$  the holder may choose to exercise, the value  $V(S, t_2)$  of the Bermudan option at time  $t_2$  is the maximum of the hold value and the payoff value (solid line). The optimal exercise point is marked. At time  $t_1$ , the hold value  $V^H(S, t_1)$  cannot be described by a European plain vanilla put but by a European option with maturity date  $t_2$  and payoff  $g^*(S) := V(S, t_2)$ . As the holder may exercise at  $t_1$ , the resulting option value is  $V(S, t_1) = \max(g(S), V^H(S, t_1))$ .

The optimal exercise curve for American options corresponds to a set of discrete optimal exercise times for Bermudan options. It is evident that the value of a Bermudan option is discontinuous with respect to  $t$  at the exercise times, and that the partial derivative  $\partial V / \partial S$  is discontinuous with respect to  $S$  at the optimal exercise points. The latter

*discontinuity requires a special treatment in numerical approximations of the integrals in lemma 2.12.*

While this reduction is quite obvious, it is not easy to transfer it into an efficient numerical method because of the recursive structure of this representation. For instance the valuation of Bermudan options proposed in [BG97] does nothing else but use Monte Carlo quadrature for each of the integrals. Because of the recursion it has an exponential runtime with respect to the number of exercise times and can be applied only for small numbers of exercise times.<sup>5</sup> Another example for the difficulties caused by the recursive structure is the analytic formula for American puts proposed by Geske and Johnson [GJ84]. This formula is an approximation obtained from a Richardson extrapolation of the exact values of three Bermudans with  $m = 2, 3, 4$  exercise times. Already for  $m = 4$  this involves the evaluation of a three-dimensional integral and with every additional exercise time the integral dimension increases. Consequently, this formula is impracticable for large  $m$ .

## Interpolation of the arising payoffs

What makes the reduction approach viable for efficient numerical valuation schemes is the use of interpolation for the “inner payoff”  $V(x, t_{i+1})$  in each time slice. This separates the quadrature rules for different time slices and allows for independent and adaptive placement of quadrature points. The interpolation breaks down the recursion and guarantees that the computational complexity increases only modestly with the number  $m$  of exercise times.<sup>6</sup>

Algorithm 2.1 gives a sketch of an adaptive pricing method using interpolation and quadrature. It is written in an unusually abstract style:  $f, g_i, V_i^H$  are functions but are treated like variables. This can be implemented by using either function pointers (e.g., in the programming language C) or function classes (a certain design pattern for object-oriented languages like C++). The implementation of the subroutines “quadrature” and

---

<sup>5</sup>This is meant as an example only. The problem with the exponential runtime can be resolved, e.g., in the stochastic mesh method in [BG04], which can be a good choice for high-dimensional pricing problems.

<sup>6</sup>Obviously the number of instructions is the same for each time slice. If one could also show that the computation time in each single time step does not depend on  $m$ , the total time complexity would be linear. This is not possible, as the conditional density function changes with  $\Delta t = \frac{T}{m}$  and thus the number of instructions required for the same accuracy can potentially increase with  $m$ .

“interpolation” as well as the grid refinement is left open at this point.

**Algorithm 2.1:** BERMUDAN( $f, g_m, m, \Delta t, r$ )

```

for  $i \leftarrow m - 1$  to 1
  do
    choose  $n := n_0$  nodes  $x_1, \dots, x_n \in \mathbb{R}^d$  as start discretization
    repeat
       $q_i \leftarrow$  QUADRATURE( $f, g_{i+1}, \Delta t, r, x_1, \dots, x_n$ )
       $V_i^H \leftarrow$  INTERPOLATION( $q_i, x_1, \dots, x_n$ )
      where necessary, refine discretization ( $n; x_1, \dots, x_n$ )
    until local discretization errors below tolerance
     $g_i \leftarrow \max(g_m, V_i^H)$ 
return ( $g_1$ )

```

**Algorithm 2.2:** QUADRATURE( $f, g, \Delta t, r, x_1, \dots, x_n$ )

```

for  $j \leftarrow 1$  to  $n$ 
  do
     $R_j \leftarrow e^{-r\Delta t} \int_{\mathbb{R}^d} f^{X_{\Delta t}|X_0=x_j}(\xi)g(\xi) d\xi$  (evaluate integral)
return (result vector  $R$ )

```

**Algorithm 2.3:** INTERPOLATION( $q, x_1, \dots, x_n$ )

Construct interpolating function to values in vector  $q$  and abscissae  $x_i$ .  
**return** (interpolant)

To keep notation simple the algorithm is formulated for Bermudan options with  $m$  *equidistant* exercise times (step size  $\Delta t$ ). The other formal parameters are the density function  $f$ , terminal payoff  $g_m$ , and risk-free interest rate  $r$ .  $n$  denotes the number of nodes in the space discretization,  $x_1 < \dots < x_n$ ,  $q_i$  denotes the hold value evaluated at each of the spline nodes,  $V_i^H$  denotes the interpolating function, and  $g_i$  the value of the Bermudan option at time step  $i$ . The function “Bermudan” returns an approximation of the current value of the Bermudan option. For the spline method the subroutines are specified later in section 3.3. For the RBF method, interpolation and quadrature are joined (see section 4.1) and the two subroutines do not occur in the explicit form above.

## Advantages of the quadrature approach

The main advantage of the quadrature approach lies in its flexibility. It “decouples” the pricing algorithm from the underlying model: The pricing algorithm for Bermudan

options uses the pricing algorithm for European options as a subroutine (algorithm 2.2). All information about the market model is contained in this subroutine. Changing the market model is thus equivalent to changing a small subroutine. This fact makes it worthwhile to develop sophisticated methods for Bermudan options based on the reduction principle.

As already mentioned, this approach can handle Lévy models but is not restricted to this class. Virtually any model with sufficiently smooth conditional density functions can be employed, e.g., models for commodity prices, interest rates, temperatures, or other underlyings. Even if there is no parametric model for the densities, a kernel estimation approach is possible. This flexibility seemed to be reserved for Monte Carlo methods up to now.

The approach is also robust with respect to non-smooth (or even discontinuous) payoffs. The state space  $\mathbb{R}^d$  can be decomposed into parts in which the payoff is smooth. In each part one can use a suitable quadrature method obtaining the full order of convergence. However, such decompositions can be difficult for higher dimensions.

There can also be advantages concerning speed and accuracy, but these strongly depend on the implementation of the approach and on the option type (dimension and model of the underlying, payoff, and exercise structure of the option). For example, a very efficient valuation of American single-asset options is possible with a specialized version of the RBF method.<sup>7</sup>

---

<sup>7</sup>A highly efficient RBF based method for this case is described in chapter 4.

## 2.4. Connection to American options

As already mentioned, any valuation method for Bermudan options can also be used to approximate the value of American options. In this section the corresponding convergence result is established, and the order of convergence is estimated in a numerical experiment. The convergence result characterizes the relation between Bermudan and American exercise structures and not a particular numerical method. Consequently, it is inherited by any numerical method that approximates American option values by Bermudan option values for a large number of exercise times. This section can be skipped by readers who are mainly interested in the valuation of Bermudan options.

### Convergence result

**Lemma 2.14** (Convergence to American options). *Let  $V_m(x, t)$  denote the value of a Bermudan option with  $m$  equidistant exercise times in  $[0, T]$  and payoff  $g$ . Let  $V$  denote the value of an American option in  $[0, T]$  with the same specifications (underlying, payoff, etc.) as  $V_m$ . Under the assumptions in section 2.1*

$$V(x, t) = \lim_{m \rightarrow \infty} V_m(x, t) \text{ for all } (x, t) \in \mathbb{R}^d \times [0, T].$$

*Proof.* A proof of convergence in the case that the underlying is driven by a geometric Brownian motion is provided by Ekström [Eks04]. In the following the proof is extended to the more general setting of the assumptions in section 2.1.

Let  $\mathcal{T} = \{t_0, \dots, t_m\}$  denote the set of possible exercise times for the Bermudan option. Assume that in a given scenario  $\omega \in \Omega$  it is optimal to exercise the American option at time  $t = t_\omega$  and price of the underlying  $x = x_\omega$ . Let  $L_m$  denote the expected loss arising from the exercise restriction of the Bermudan option. This gives rise to a map  $L_m : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^+$  mapping each possible optimal exercise point  $(x, t)$  to the corresponding expected loss. The value of the American option at point  $(x, t)$  is the payoff  $g(x)$ . The value of the Bermudan option at this point corresponds to the value of a European option with payoff  $V(\cdot, \phi_m(t))$  at maturity time  $\phi_m(t)$ , where

$$\phi_m(t) := \inf\{t_j \in \mathcal{T} | t_j > t, j = 1, \dots, m\}$$

denotes the next possible exercise time of the  $m$ -Bermudan option. The loss  $L_m$  is the difference of both values. Using lemma 2.9 the expected loss can be written as

$$\begin{aligned} L_m(x, t) &= g(x) - e^{-r(\phi_m(t)-t)} \int_{\mathbb{R}^d} \underbrace{f^{X_{\phi_m(t)} | X_t=x}(\xi)}_{\geq 0} \underbrace{V(\xi, \phi_m(t))}_{\geq g(\xi)} d\xi \\ &\leq g(x) - e^{-r(\phi_m(t)-t)} \int_{\mathbb{R}^d} f^{X_{\phi_m(t)} | X_t=x}(\xi) g(\xi) d\xi \quad (\text{cf. lemma 2.12}) \\ &=: \tilde{L}_m(x, t). \end{aligned}$$

What happens for  $m \rightarrow \infty$ ? First, as  $\phi_m(t) \rightarrow t$  the discounting factor converges to 1. Second, assumption 2.4 assures that the conditional probability density function in the integrand converges to the Dirac delta function  $\delta_x$ . This implies

$$\tilde{L}_m(x, t) \rightarrow g(x) - \int_{\mathbb{R}^d} \delta_x(\xi) g(\xi) \, d\xi = g(x) - g(x) = 0 \Rightarrow \lim_{m \rightarrow \infty} L_m = 0 \text{ pointwise.}$$

We now come back to the American and Bermudan option. Let  $V$  and  $V_m$  denote their respective fair values. Following the risk-neutral pricing approach these values are discounted expectations of the future payoffs under the risk-neutral measure  $\mathbb{Q}$ . Thus we can write their difference as discounted expectation, as well:<sup>8</sup>

$$0 \leq V - V_m = \mathbb{E}^{\mathbb{Q}}[e^{-rt_\omega} L_m(x_\omega, t_\omega)] \leq \max(1, e^{-rT}) \mathbb{E}^{\mathbb{Q}}[L_m],$$

where  $(x_\omega, t_\omega)$  denotes the optimal American exercise point corresponding to a scenario  $\omega \in \Omega$ . It is sufficient to show the convergence  $\mathbb{E}^{\mathbb{Q}}[L_m] \rightarrow 0$  for  $m \rightarrow \infty$ . The expectation value is defined as an integral with respect to the risk-neutral measure. In this situation Lebesgue's dominated convergence theorem can be applied in its general form.<sup>9</sup>

$L_m$  is a function on the domain  $\mathbb{R}^d \times [0, T]$ . The canonical map  $\psi : \Omega \rightarrow \mathbb{R}^d \times [0, T]$  is introduced that assigns the optimal exercise point  $(x, t)$  of the American option to each scenario  $\omega$ . This makes  $L_m \circ \psi : \Omega \rightarrow \mathbb{R}^+$  a map on the domain  $\Omega$  of all scenarios.  $(L_m \circ \psi)$  is a sequence of functions on  $\Omega$ . The sequence converges pointwise to the function 0 as shown above. The limit function 0 is  $\mathbb{Q}$ -measurable. The payoff defines a  $\mathbb{Q}$ -integrable function  $g \circ \psi : \Omega \rightarrow \mathbb{R}^+$ . The consequence from Lebesgue's theorem is that the sequence of integrals converges to the integral of the pointwise limit function, i.e.  $\mathbb{E}^{\mathbb{Q}}[L_m] \rightarrow 0$ .  $\square$

This convergence result is implicitly assumed in many papers which derive valuation methods for Bermudan options in order to approximate American option values.

## Order of convergence

In the following the rate of the convergence  $V_m \rightarrow V$  is estimated numerically. Readers who are already satisfied by the qualitative convergence result in lemma 2.14 can skip this section.

For the interpretation of methods for Bermudan options as an approximation to American option values, it is important to estimate the time discretization error  $\|V - V_m\|_\infty$ . In the following this error is quantified in an empirical sense. As test options the Bermudan options specified in table 2.1 (p. 22) are used. The number of exercise times for

<sup>8</sup>Theorem 1.8 (p. 5) describes this approach for European options. Here it is used in the case of a payoff at time  $t_\omega$  due to the early exercise feature. In this case the discounting factor has to be written into the expectation as it depends on  $\omega$ .

<sup>9</sup>The theorem can be found in, e.g., [Els02] or [Alt85].

each of these options is a power of two,  $m := 2^i$  ( $i \in \mathbb{N}$ ). The option parameters and the model of the underlying are chosen to match the example in [Sey02], p. 143. For the valuation of these Bermudan options the RBF method, which is introduced later in chapter 4 is used.<sup>10</sup> This method gives accurate results for the Bermudan options. More details about this method are not necessary at this point. In fact, for this experiment it could be replaced by any other valuation method for Bermudan options.

It is evident from table 2.1 that the values  $V_m$  are strictly increasing, and that the difference quotients

$$Q_i := \frac{V_{2^i} - V_{2^{i-1}}}{V_{2^{i-1}} - V_{2^{i-2}}}$$

are approximately the same. Assuming such a quotient  $q := \frac{1}{2}$  implies

$$V_{2^i} = (1 + q)V_{2^{i-1}} - qV_{2^{i-2}}.$$

The solutions of this linear difference equation are of the form

$$V_{2^i} = c_0 + c_1 q^i$$

with constants  $c_0, c_1$ . Lemma 2.14 assures the convergence

$$V_{2^i} \xrightarrow{i \rightarrow \infty} V \Rightarrow V = c_0, \text{ and consequently } V - V_{2^i} = -c_1 q^i.$$

Inserting  $q = \frac{1}{2}$  and  $m = 2^i$  yields the convergence rate

$$|V - V_m| = \mathbf{O}\left(\frac{1}{m}\right).$$

A graphical illustration of the empirical convergence result is given in figure 2.3. Repeating this experiment with other options leads to similar results, thus supporting the following hypothesis.

**Hypothesis 2.15** (Order of convergence). *The values  $V_m$  of the Bermudan options with  $m = 2^i$  equidistant exercise times converge linearly in the time step size  $\Delta t := \frac{1}{m}$  to the value  $V$  of the corresponding American option:*

$$|V(x, 0) - V_m(x, 0)| = \mathbf{O}(\Delta t) \text{ for all } x \in \mathbb{R}^d$$

Although the empirical result only indicates pointwise convergence, there is every reason to believe that the stronger uniform convergence  $\|V - V_m\|_\infty = \mathbf{O}(\Delta t)$  applies, too.

**Remark 2.16** (Independence from RBF-method). *The experiment above does not depend on the method used for the valuation of the Bermudan options. Other efficient valuation techniques for Bermudan options could have been used. Hypothesis 2.15 characterizes the relation between Bermudan and American options and not the properties of a particular numerical method.*

<sup>10</sup>For the RBF method, a space discretization of 3200 equidistant points in the log-price interval  $[-1, 6]$  is used. This discretization is fine enough for the space discretization error to be negligible.

## Extrapolations

### Linear Extrapolation

Hypothesis 2.15 justifies linear extrapolation to obtain the value of an American option. Let for this section  $V(\Delta t)$  denote the value of the Bermudan option with time step size  $\Delta t$  and let  $V(0)$  denote the value of the American option. In the case of two Bermudan options with time step sizes  $\Delta t$  and  $\frac{\Delta t}{2}$  the linear extrapolation specializes to

$$V(\Delta t) = V(0) + a\Delta t, \quad a := \frac{V(\frac{\Delta t}{2}) - V(\Delta t)}{\frac{\Delta t}{2} - \Delta t}$$

$$\Rightarrow V(0) = 2V(\frac{\Delta t}{2}) - V(\Delta t)$$

Table 2.2 shows these extrapolations based on the results in table 2.1.

### Repeated Richardson extrapolation

For example, Chang et al. [CCS01] propose repeated Richardson extrapolation in the context of the approximation of American option values by corresponding Bermudan option values. As this is a numerical standard technique (e.g. [BD72]), it is not discussed here. It can be applied to any approximation of a function value  $V(0)$  by  $V(\Delta t)$ ,  $\Delta t > 0$ , where the expansion of  $V$  is

$$V(\Delta t) = a_0 + a_1\Delta t^1 + a_2\Delta t^2 + a_3\Delta t^3 + \dots$$

with unknown but constant coefficients. Then the function

$$W_n(\Delta t) := W_{n-1}(\Delta t) + \frac{W_{n-1}(\Delta t) - W_{n-1}(q^{-1}\Delta t)}{q^{1-n} - 1}, \quad W_1(\Delta t) := W(\Delta t)$$

Table 2.1.: Values  $V_m$  of Bermudan options with equidistant exercise times

parameter	value	$i$	$m$	$V_{2^i} = V_m(K, 0)$	$D_i = V_{2^i} - V_{2^{i-1}}$	$Q_i = D_i/D_{i-1}$
payoff	put	1	$2^1$	1.79819989814934		
$K$	10	2	$2^2$	1.83986252549659	0.04166262734725	
$T$	1	3	$2^3$	1.86044549274970	0.02058296725311	0.494039108
model	B-S	4	$2^4$	1.87113830098689	0.01069280823719	0.519497899
$r$	0.25	5	$2^5$	1.87645648277666	0.00531818178978	0.497360625
$\delta$	0.2	6	$2^6$	1.87909470521520	0.00263822243853	0.496076017
$\sigma$	0.6	7	$2^7$	1.88040178370384	0.00130707848864	0.495439077
method	RBF	..	..	..	..	..
$n$	3200	12	$2^{12}$	1.88165177423294	0.00003967770763	0.498045902
$[a, b]$	[-1,8]	13	$2^{13}$	1.88167154177403	0.00001976754109	0.498202701
$n_{Gauss}$	175	14	$2^{14}$	1.88168142286171	0.00000988108768	0.499864279

has the expansion

$$W_n(\Delta t) = a_0 + a_n^{(n)} \Delta t^n + a_{n+1}^{(n)} \Delta t^{n+1} + \dots,$$

where  $a_n^{(n)}$ ,  $a_{n+1}^{(n)}$ , ... denote other constant coefficients. This extrapolation gives similar results as a linear extrapolation. Table 2.3 contains results based on table 2.1. This result is unexpected. It indicates that repeated Richardson extrapolation is not able to improve the accuracy substantially over a linear extrapolation.

Table 2.2.: Linear extrapolation for the American option  $V$

$i$	$V_i^{lin} := 2V_{2^i} - V_{2^{i-1}}$	$ V_i^{lin} - V_{i-1}^{lin} $
2	1.88152515284384	
3	1.88102846000281	$5.0 \times 10^{-4}$
4	1.88183110922407	$8.0 \times 10^{-4}$
5	1.88177466456644	$5.6 \times 10^{-5}$
6	1.88173292765373	$4.2 \times 10^{-5}$
7	1.88170886219248	$2.4 \times 10^{-5}$
..	..	..
11	1.88169176329395	$7.6 \times 10^{-7}$
12	1.88169145194056	$3.1 \times 10^{-7}$
13	1.88169130931512	$1.4 \times 10^{-7}$
14	1.88169130394939	$5.4 \times 10^{-9}$

Table 2.3.: Repeated Richardson extrapolation for the American option  $V$ . Each row contains the extrapolation value  $W_i$  based on the values  $V_{2^1}, \dots, V_{2^i}$  in table 2.1.

$i$	$W_i$	$ W_i - W_{i-1} $
2	1.8815251528	$8.3 \times 10^{-2}$
3	1.8808628957	$6.6 \times 10^{-4}$
4	1.8822751966	$1.4 \times 10^{-3}$
5	1.8816689889	$6.1 \times 10^{-4}$
6	1.8817156705	$4.7 \times 10^{-5}$
7	1.8816963598	$1.9 \times 10^{-5}$
8	1.8817024253	$6.1 \times 10^{-6}$

**Empirical finding: linear dependence of  $V_m$  and  $\frac{1}{m}$**

The above extrapolation results are surprising. They suggest the hypothesis that the Bermudan option values  $V_m$  is linear in the time step size  $\frac{1}{m}$ :

$$V_m = V_\infty - a \frac{1}{m}$$

with a constant  $a > 0$ . This hypothesis can be supported by the graphical analysis in figure 2.3. A linear dependence would be of practical relevance, as it would imply that highly accurate approximations to American option values can be obtained from highly accurate approximations to Bermudan options with only a moderate number of exercise times. A future analysis of this finding could be interesting for future research.

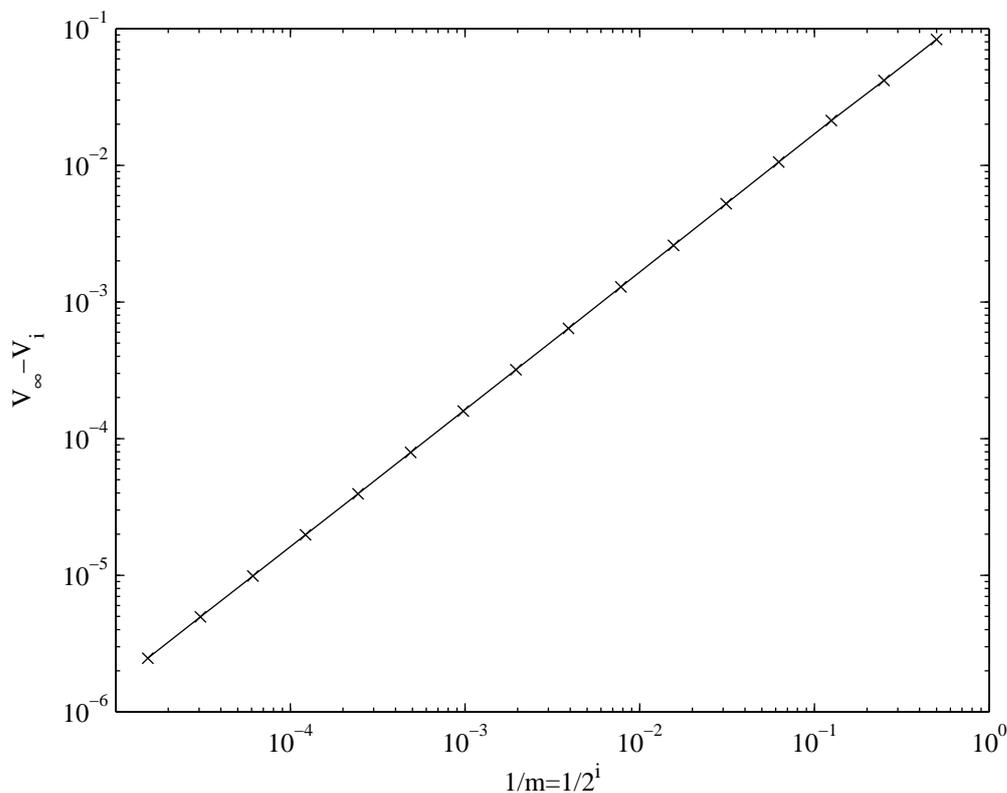


Figure 2.3.: A practical test indicates not only linear convergence of  $V_m \rightarrow V_\infty$  in the step size  $1/m$ , but even linear *dependence* over several orders of magnitude. For this illustration, the Bermudan values from table 2.1 have been compared to the best extrapolation from table 2.2 for the American option value, i.e.,  $V_\infty \approx 1.88169130394939$ .

## 2.5. Related methods

This section discusses related deterministic<sup>11</sup> methods from the current literature, i.e., methods which are also based on the reduction principle in lemma 2.12. This applies to the following articles: [Sul00], [AWDN03], and [O’S05]. Some remarks on existing literature beyond this list are given at the end of this section.

### Sullivan’s method (2000)

Sullivan [Sul00] proposes a method for the valuation of American plain-vanilla puts in a Black-Scholes setting. He uses the reduction approach to value Bermudan options for several numbers of exercise times. A Richardson extrapolation then approximates the American value. In this work, the non-smooth pasting at the exercise points of Bermudan options is already treated by splitting up the integrals. The integrals are approximated by Gauss-Legendre quadrature over a finite interval (truncating large values of  $x$ ). In each time step the early exercise boundary is determined by a bisection method.<sup>12</sup> Unfortunately, an analysis of the numerical properties of this method is lacking. The choice of  $p = 8$  nodes for the Chebyshev interpolation seems heuristical. Although Sullivan observes a stability problem,<sup>13</sup> he does not address it, e.g., by adapting the quadrature scheme to the integrand. A numerical experiment included in the article indicates that an adequate accuracy can be reached with costs comparable to the binomial tree method.<sup>14</sup> Taking into account the striking simplicity of the binomial method, this is not a very convincing result.

**Remark 2.17.** *The heading “multidimensional approximation” in [Sul00] (p. 84ff) may be confusing. In the corresponding section, an interpolant of the option price is constructed for a multidimensional parameter region:  $(S, T, \sigma) \in [32, 52] \times [0.1, 5.0] \times [0.1, 0.5]$ . As expected, the evaluation of such an interpolant is faster than any other method (row “four-dimensional” in table 5, p. 92). The model of the underlying is not “multidimensional”, but the standard Black-Scholes model.*

### The method of Andricopoulos et al. (2003)

A similar quadrature based method has been proposed by Andricopoulos et al. in [AWDN03]. The Gaussian quadrature in Sullivan’s method is replaced by a compound

<sup>11</sup>Non-deterministic methods are not included in this discussion. A non-deterministic method based on the reduction principle is, e.g., the method of Longstaff and Schwartz [LS01].

<sup>12</sup>A bisection method is apparently what “by trial and error” ([Sul00], p. 79) means.

<sup>13</sup>“A larger number of exercise dates require[s] a higher degree of quadrature” ([Sul00], p. 82).

<sup>14</sup>Experiments indicate that “the extrapolated quadrature method [i.e., a four-point Richardson extrapolation for  $m = 1, 3, 9$ , and 27 exercise times] has the best accuracy-speed combination for [a certain] set of option parameters, performing 15% faster than the binomial method while producing lower errors on average” ([Sul00], p. 84).

Simpson rule, which results in a lower convergence order of  $\mathbf{O}(\Delta x^4)$ . This method does not allow to control the accuracy. Error estimates are lacking. The non-smooth pasting at the exercise points is treated by placing a node of the discretization at each optimal exercise point of the Bermudan option.<sup>15</sup>

From the numerical point of view this method is not an improvement over Sullivan’s method. The main contribution of the paper [AWDN03] is to emphasize the universal applicability of quadrature based methods by several examples.

### Extension to multiple assets in a Black-Scholes model

In the working paper [ADNW06] Andricopoulos et al. try to extend the method to higher dimensions. The approach is a straightforward generalization of the one-dimensional case in [AWDN03]. Unfortunately, the authors do not mention which quadrature rule is used for the multi-dimensional integrals. As they use Simpson’s rule in the one-dimensional case, it is presumably a product rule based on Simpson’s rule. A brief review of the results given in [ADNW06], table 4, follows:

- The “2-underlying down-and-out barrier option” has a discretely monitored barrier with 12 observation times. The computation of a discrete “out-barrier” option value is technically the same as pricing a Bermudan option with corresponding exercise times.
- The “3-underlying American call” is a call with discrete dividends at two dates. There is no continuous dividend paid between the discrete dividend dates. In this case, it is not optimal to exercise the American option between the dividend dates. Consequently, this option reduces to a Bermudan option on three underlyings and two exercise times. This “American” option turns out to be nearly a European one.
- The “2-underlying Bermudan put” has 12 exercise times.

It can be summarized that the authors apply their method to Bermudan options in a Black-Scholes setting with  $d = 2$  underlyings and  $m \leq 12$  exercise times or  $d = 3$  underlyings and  $m = 2$  exercise times. The method seems not suitable for pricing Bermudan options with a higher number of exercise times.<sup>16</sup> This limitation is probably caused by the increasing peakedness of the multivariate Gaussian normal density function with decreasing time step size  $\Delta t = \frac{1}{m}$ .<sup>17</sup> Non-adaptive low order quadrature rules like Simpson’s rule are not suited for highly peaked integrands. The methods that are proposed in

<sup>15</sup>“The method benefits from the simple and exact placement of ‘nodes’ on boundaries [...], thus removing any nonlinearity error” ([AWDN03], p. 469).

<sup>16</sup>The same problem can already be observed in the one-dimensional case, although less significant: The maximal number of exercise times in the examples is  $m = 125$  in [AWDN03], table 2, p. 467f, and even only  $m = 20$  in the derived paper [O’S05], table 1, “Q-FFT, R20”.

<sup>17</sup>The increasing peakedness is implied by assumption 2.4 (convergence of PDF to Dirac delta).

the following two chapters do not exhibit problems with peaked integrands. The spline method uses an adaptive quadrature rule and the RBF method uses a quadrature rule with high polynomial order.<sup>18</sup>

### O’Sullivan’s extension to Lévy processes (2005)

O’Sullivan [O’S05] was the first who proposed a quadrature based pricing method in the context of Lévy processes. The method is a combination of the method of Andricopoulos et al. with an FFT based evaluation of the conditional probability density function. The PDF  $f^{X_{\Delta t}|X_0=x}$  can be constructed as follows out of the conditional characteristic function (CCF) of the stochastic process. The CCF is defined as

$$\Psi(\phi, \Delta t; x) := \mathbb{E}^{\mathbb{Q}}[\exp(i\phi X_{\Delta t})|X_0 = x] = \int_{-\infty}^{\infty} \exp(i\phi\xi) f^{X_{\Delta t}|X_0=x}(\xi) d\xi.$$

The conditional cumulative density function is given by the inverse Fourier transform

$$\begin{aligned} \mathbb{P}(X_{\Delta t} < y|X_0 = x) &= \int_{-\infty}^y f^{X_{\Delta t}|X_0=x}(\xi) d\xi \\ &= \frac{1}{2} - \frac{1}{\pi} \int_0^{\infty} \mathbf{Re} \left( \frac{1}{i\phi} \exp(-i\phi y) \Psi(\phi, \Delta t; x) \right) d\phi. \end{aligned}$$

By definition, the conditional probability density function  $f$  is the derivative of the above expression with respect to  $y$  and thus

$$f^{X_{\Delta t}|X_0=x}(y) = \frac{1}{\pi} \int_0^{\infty} \mathbf{Re} (\exp(-i\phi y) \Psi(\phi, \Delta t; x)) d\phi.$$

This expression can be evaluated by a fast Fourier transform. The details are skipped here; but it is important that this technique can be used with any quadrature method, if the CCF is known in closed form.

O’Sullivan’s extension provides a method for one-dimensional<sup>19</sup> Bermudan options under Lévy processes with time complexity  $\mathbf{O}(mn^2)$ , where  $m$  is the number of exercise times and  $n$  the number of nodes in the asset space. Although an analysis of accuracy is still lacking, the paper [O’S05] includes numerical results that are similar to reference solutions obtained by binomial tree and lattice based methods.

<sup>18</sup>For example, to obtain the numerical results in chapter 6, a quadrature rule with polynomial order 350 is used in the one-dimensional case and with polynomial order 50 in the higher-dimensional case.

<sup>19</sup>The dimension refers to the number of underlyings. Time is not counted as a dimension.

## Further remarks

- **Parkinson’s method:** O’Sullivan writes: “Option pricing using numerical integration methods was first introduced by Parkinson (1977) [Par77]”.<sup>20</sup> This is misleading. Although Parkinson uses the reduction principle 2.12 and a rectangle approximation to the integral for the derivation of his method,<sup>21</sup> the resulting method is a trinomial tree method. Consequently it has different properties than quadrature methods.

Quadrature methods allow in particular to choose the time discretization and the (price-) space discretization *independently*. They can by design price a Bermudan option with arbitrary precision using only the time discretization implied by the Bermudan exercise structure. This is not possible with trinomial tree methods, where higher precision is by design accompanied by a finer time discretization.

- **Huang, Subrahmanyam, and Yu’s method:** The method proposed in [HSY96] uses an integral equation for the American put value. This integral equation can be solved recursively for each time step to obtain an approximation to the early exercise boundary. The approximation can in turn be used to evaluate the American put. Although this paper is cited by Andricopoulos et al. as a method “using univariate integration”,<sup>22</sup> it is not a quadrature method in the above sense.
- **Kim’s method:** Kim [Kim90] derives an integral equation for the optimal exercise boundary of a one-dimensional plain-vanilla American option in the Black-Scholes model. The solution of the integral equation can then be used in a second step to calculate the value of the option.<sup>23</sup> As the derivation uses the closed-form solution of the Black-Scholes equation, this approach is limited to one-dimensional plain-vanilla options in the Black-Scholes model. Kim’s method is not a quadrature method in the above sense, as it does not follow the construction principle in lemma 2.12.
- **Longstaff-Schwartz method:** We did not include the method proposed by Longstaff and Schwartz [LS01] in the historic review above because it is nondeterministic. But in principle it relies on the same ideas. It uses a Monte Carlo method as valuation procedure for European puts with arbitrary payoffs. Of course Monte Carlo approximation leads to non-smooth option prices and thus prohibits the use of interpolation methods. To resolve this problem Longstaff and Schwartz relax the interpolation to a least-squares regression.

---

<sup>20</sup>[O’S05], p. 2

<sup>21</sup>[Par77], p. 28

<sup>22</sup>[AWDN03], p. 450

<sup>23</sup>Kim uses an integration approach for the second step, but other numerical approaches could also be used, e.g., finite differences: Once the optimal exercise boundary is known, the free-boundary problem simplifies to a boundary value problem.

## 2.6. Convolution-based methods

The reduction principle in section 2.3 allows the construction of valuation methods for Bermudan options out of valuation methods for European options with arbitrary payoffs. Although this work focuses on quadrature based methods, the reduction principle is not restricted to these methods. It can also be applied to any other valuation method for European options. An alternative idea is the use of convolution methods. In the following this approach is briefly compared to quadrature based methods. For European options, a convolution-based method has been proposed by Carr and Madan in [CM99]. As the application of convolution methods to Bermudan options cannot be found in the recent literature, it is described in appendix A.2. Some differences and similarities between possible convolution-based methods and quadrature methods are the following.

- The convolution approach requires translation invariance of the conditional probabilities, i.e. assumption 2.5. This assumption holds for most of the current stock price models. An example *without* translation invariance is the Heston model with stochastic volatility. In this model the volatility process is not translation invariant. Another important example for which assumption 2.5 fails are interest rate models. The mean-reversion incorporated in models like CIR or Vasicek excludes translation invariance. The requirement of translation invariance can be seen as a restriction inherent to convolution-based methods.
- Some integrability conditions, e.g. assumption A.1 (p. 148), can impose further restrictions on the model or option type in practice.
- The computational complexity of the fast Fourier transform is  $\mathbf{O}(n \log n)$ , where  $n$  denotes the number of nodes in the discretization. Applying this method in the context of pricing Bermudan options, the resulting complexity will be  $\mathbf{O}(mn \log n)$ , where  $m$  denotes the number of exercise points of the Bermudan option. This is a good result. However, the RBF method can lead to a complexity of  $\mathbf{O}(mn)$  in the same special case of one underlying and equidistant nodes (see section 4.4.2).
- The application to higher dimensional pricing problems would require multivariate fast Fourier transforms.

In short, the introduction of convolution methods for Bermudan options is given in appendix A.2. The convolution approach bears some technical difficulties (as the choice of damping parameters). If these difficulties can be solved, the computational complexity is competitive. The method is restricted to translation invariant models. In addition an extension to multivariate options requires multivariate FFT methods.

## 3. The spline method

This chapter proposes a new method for pricing Bermudan options based on quadrature and spline interpolation. After a short review of spline interpolation and suitable numerical quadrature methods, the *spline method* is introduced in section 3.3 for single-asset options. Its numerical properties are analyzed in section 3.4, and a possible extension to multi-asset options is discussed in section 3.5. Numerical results are deferred to chapter 6.

### 3.1. Spline interpolation

The main motivation for the spline method is online accuracy control. Controlling the local discretization error allows to minimize the computational costs for a pre-specified accuracy. Spline interpolation allows for an adaptive placement of nodes. The space discretization can be refined locally until the local error falls below a given tolerance.

In contrast to other interpolation techniques, as, e.g., polynomial interpolation or Chebyshev interpolation, splines can be constructed in  $\mathbf{O}(n)$  ( $n$  = number of nodes) and evaluated in  $\mathbf{O}(1)$ . Fast evaluation of the interpolant is very important, as the quadrature rule requires possibly a large number of evaluations. For sufficiently smooth functions the spline converges with decreasing fill distance  $\|\Delta\|$  to the interpolated function, and the interpolation error is of order  $\mathbf{O}(\|\Delta\|^4)$ . In the following some aspects of splines are briefly reviewed. A more comprehensive introduction can be found in [Gre69].

#### Definition

A cubic spline is a  $\mathcal{C}^2$ -function which is defined piecewise by cubic polynomials:

**Definition 3.1** (Cubic spline). *Let  $\Delta = \{a = x_0 < x_1 < \dots < x_n = b\}$  be a partition of the interval  $[a, b]$ . A cubic spline with respect to  $\Delta$  is a function  $S_\Delta \in \mathcal{C}^2[a, b]$  which is a cubic polynomial on each interval  $[x_i, x_{i+1}]$  ( $i = 0, \dots, n - 1$ ).*

Although this definition can be generalized to piecewise polynomials of higher orders, we use only cubic splines in this work.

**Definition 3.2** (Interpolating spline). *Let  $f$  be a function<sup>1</sup> on  $[a, b]$ . The spline  $S_\Delta$  is called interpolating spline of  $f$ , if  $S_\Delta(x_i) = f(x_i)$  for every  $i \in \{0, \dots, n\}$ .*

---

<sup>1</sup>Notational convention: In this section we use  $f$  to denote the interpolated function, not the probability density function as in the rest of this work.

## Boundary conditions

In the context of interpolation it can be observed that each of the  $n$  cubic polynomials introduces four degrees of freedom. The  $\mathcal{C}^2$ -condition at each of the  $n - 1$  inner nodes requires three degrees of freedom (matching  $S_\Delta$ ,  $S'_\Delta$ , and  $S''_\Delta$ ). The interpolation condition requires one degree at each of the  $n + 1$  nodes. This leaves  $4n - 3(n - 1) - (n + 1) = 2$  degrees of freedom. A possible boundary condition is the *natural*<sup>2</sup> condition  $S''(a) = S''(b) = 0$ . It is not optimal in this case as of interest is the convergence of the spline instead of minimal curvature. Therefore

$$S'(a) = f'(a) \text{ and } S'(b) = f'(b) \quad (3.1)$$

is chosen to determine the interpolating spline uniquely.

## Construction

The construction of interpolating cubic splines can be divided into two steps. One can show that a spline is uniquely determined by its *moments*

$$m_i := S''_\Delta(x_i) \quad (i = 0, \dots, n),$$

and that these moments are the solution of a tridiagonal system.

## Representation by moments

$S''$  is by definition linear in each interval  $[x_i, x_{i+1}]$ :

$$S''(x) = m_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + m_{i+1} \frac{x - x_i}{x_{i+1} - x_i}$$

Integration gives

$$\begin{aligned} S'(x) &= -m_i \frac{(x_{i+1} - x)^2}{2(x_{i+1} - x_i)} + m_{i+1} \frac{(x - x_i)^2}{2(x_{i+1} - x_i)} + c_i \\ S(x) &= m_i \frac{(x_{i+1} - x)^3}{6(x_{i+1} - x_i)} + m_{i+1} \frac{(x - x_i)^3}{6(x_{i+1} - x_i)} + c_i(x - x_i) + d_i \end{aligned} \quad (3.2)$$

with integration constants  $c_i$  and  $d_i$ . They can be determined using the interpolation conditions  $S(x_i) = f(x_i)$  and  $S(x_{i+1}) = f(x_{i+1})$ :

$$\begin{aligned} c_i &= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{6} (m_{i+1} - m_i) \\ d_i &= f(x_i) - m_i \frac{(x_{i+1} - x_i)^2}{6} \end{aligned}$$

---

<sup>2</sup>This condition is called “natural” because the resulting spline  $S$  minimizes  $\|S''\|_{\mathcal{L}^2[a,b]}^2$  under all interpolating functions in  $\mathcal{C}^2[a,b]$ . For small  $f'$  this value can be seen as an approximation of the curvature.

Substituting the constants in equation (3.2), it is seen that the moments  $m_i$  ( $i = 0, \dots, n$ ) uniquely determine the spline function piecewise on each interval  $[x_i, x_{i+1}]$  and thus globally on  $[a, b]$ . Given the moments  $m_i$  we can evaluate the spline function in  $\mathbf{O}(1)$  at any point  $x \in [a, b]$  using equation (3.2) for the subinterval  $[x_i, x_{i+1}] \ni x$ .<sup>3</sup>

### Computation of the moments

The smooth pasting conditions  $S'(x_i^-) = S'(x_i^+)$  at each inner node  $x_i$  introduce  $n - 1$  linear equations for the inner moments  $m_1, \dots, m_{n-1}$ .

$$\mu_i m_{i-1} + 2m_i + \lambda_i m_{i+1} = b_i$$

for  $i = 1, \dots, n - 1$  with

$$\begin{aligned} \lambda_i &:= \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}, \\ \mu_i &:= 1 - \lambda_i, \\ b_i &:= \frac{6}{x_{i+1} - x_{i-1}} \left( \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \right). \end{aligned}$$

The boundary conditions (3.1) provide two additional equations of the same form for  $i \in \{0, n\}$  with coefficients  $\mu_0 := 0$ ,  $\lambda_0 := 1$ ,  $\mu_n := 1$ ,  $\lambda_n := 0$ , and right hand sides

$$\begin{aligned} b_0 &:= \frac{6}{x_1 - x_0} \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} - f'(x_0) \right), \\ b_n &:= \frac{6}{x_n - x_{n-1}} \left( f'(x_n) - \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \right). \end{aligned}$$

Together we have the tridiagonal linear system

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \cdot & \cdot & \cdot & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \cdot \\ \cdot \\ m_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \cdot \\ \cdot \\ b_n \end{pmatrix}$$

Gerschgorin's circle theorem implies that the system has a unique solution for every partition  $\Delta$ .<sup>4</sup> Because of the tridiagonal structure it can be solved in  $\mathbf{O}(n)$  by Gaussian elimination.

<sup>3</sup>In fact it is not trivial to obtain  $\mathbf{O}(1)$ : For an arbitrary point  $x$  one needs to search the whole partition  $\Delta$  for the subinterval  $[x_i, x_{i+1}]$  which contains  $x$ . A simple binary search would lead to  $\mathbf{O}(\log n)$  for each evaluation. Combining binary search with a "hint parameter" in the quadrature method, one can reduce the overall search time for each integration to  $\mathbf{O}(n)$ : Although the worst case complexity for each single searching is  $\mathbf{O}(n)$ , it can be seen that the total complexity for  $n$  searchings is also  $\mathbf{O}(n)$  in this case.

<sup>4</sup>Theorem can be found in [SB00], §6.9; here:  $\mu_i \in (0, 1)$ ,  $\lambda_i \in (0, 1)$  for  $1 \leq i \leq n - 1$  and  $\lambda_0 = \mu_n = 1$ .

## Interpolation error

For present purposes the following estimate of the interpolation error is a bit theoretical, as it expresses the interpolation error only by means of the fill distance  $\|\Delta\|$ . In practice we do not consider  $\|\Delta\| \rightarrow 0$  but refine the spline locally, based on a heuristic error estimate. The following result can be seen as a worst-case estimate.

**Theorem 3.3** (Interpolation error of cubic splines). *Let  $f \in \mathcal{C}^4[a, b]$  and  $\|f^{(4)}\|_\infty \leq L$ . Let  $\Delta = \{a = x_0 < x_1 < \dots < x_n = b\}$  be a partition of  $[a, b]$  and  $S_\Delta$  the corresponding interpolating cubic spline with  $S'_\Delta(a) = f'(a)$  and  $S'_\Delta(b) = f'(b)$ . Then*

$$\|f - S_\Delta\|_\infty \leq \frac{5}{384}L\|\Delta\|^4,$$

where  $\|\Delta\| := \max\{x_{i+1} - x_i | i = 1, \dots, n - 1\}$  denotes the fill distance.

*Proof.* A proof can be found in [HM76]. □

## 3.2. Adaptive quadrature

Our aim is to construct a quadrature based pricing method with accuracy control. This requires an adaptive quadrature rule. The integrand is the product of a spline function and a smooth function and thus piecewise  $\mathcal{C}^\infty[x_i, x_{i+1}]$ -smooth. As the function is only  $\mathcal{C}^2[a, b]$ -smooth globally, it does not make sense to use high-order quadrature schemes for the whole interval  $[a, b]$ . It is necessary to split up the integration according to the partition  $\Delta$  of the spline function. The subintervals  $[x_i, x_{i+1}]$  are already small and it can be expected that only a few quadrature points suffice to achieve high accuracy. The use of adaptive Romberg quadrature for each subinterval is decided upon.

**Remark 3.4** (Special case for Black-Scholes). *In the Black-Scholes case the conditional density function has the form  $a_0 e^{-a_1(x+a_2)^2}$  with certain constants  $a_0, a_1, a_2$ . The integrand is piecewise a product of this function with a cubic polynomial. It is possible to develop a specialized quadrature rule for integrands of the form  $a_0 e^{-a_1(x-x_0+a_2)^2} (x^3 + a_3x^2 + a_4x + a_5)$ . In this work the focus is on a method that is suitable for other market models and therefore a “general purpose” quadrature rule is used.*

## Romberg quadrature

Romberg quadrature is an application of the repeated Richardson extrapolation (RRE) mentioned in section 2.4 to the trapezoid rule. Assume that it is wished to approximate an integral over a function  $f$ :

$$\int_a^b f(x) dx, \quad \text{where } f \in \mathcal{C}^\infty(a, b) \text{ is a smooth function.}$$

In this context,  $f$  denotes an arbitrary function and not the probability density as in chapter 2. Let  $T(h)$  be the value of the trapezoid rule

$$T(h) := h\left[\frac{1}{2}f(a) + f(a+h) + f(a+2h) + \dots + f(b-h) + \frac{1}{2}f(b)\right]$$

with step size  $h$  ( $b-a = kh$  for an integer  $k$ ). Using the Euler Maclaurin formula, one can show that  $f \in \mathcal{C}^{2m+2}$   $T$  has the expansion

$$T(h) = \int_a^b f(x) dx + a_1 h^2 + a_2 h^4 + a_3 h^6 + \dots + \mathbf{O}(h^{2m+2})$$

with constants  $a_i$  (independent of  $h$ ). Consequently one can apply RRE to approximate  $T(0)$ . Let  $(h_i)$  be a sequence of decreasing step sizes and  $T_{i,0} := T(h_i)$  denote the corresponding trapezoid rule approximations of the integral. Then RRE defines the approximations

$$T_{i,k} := T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{(h_{i-k}/h_i)^2 - 1}, \text{ for } 1 \leq k \leq i \leq m.$$

**Theorem 3.5** (Error estimate for a special Romberg sequence). *Let  $f \in \mathcal{C}^{2k+2}$  and the sequence of step sizes be  $h_i := h_{i-1}/2$ ,  $h_0 := b-a$ . Then the integration error is*

$$T_{i,k} - \int_a^b f(x) dx = (b-a) \left( \prod_{j=0}^k h_{i-j}^2 \right) \frac{(-1)^k B_{2k+2}}{(2k+2)!} f^{(2k+2)}(\xi)$$

for a  $\xi \in (a, b)$ , where  $B_{2k+2}$  denotes the  $(2k+2)$ th Bernoulli number.

*Proof.* See [BRS63]. □

This error estimate could be used to estimate the error for a given integrand  $f$  analytically. The analysis would depend on both the option type and the model of the underlying. In practice, a heuristic terminating condition gives satisfactory results. It observes several successive approximations  $T_{k-2,k-2}$ ,  $T_{k-1,k-1}$ ,  $T_{k,k}$  for each  $k$  to decide if the required accuracy is already achieved. Of course, this simple terminating condition can later be replaced by a more sophisticated one.

### 3.3. Construction of the valuation method

This section applies spline interpolation in the context of the reduction principle 2.12 (p. 14). The coarse framework for the valuation method has already been described by algorithm 2.1 (p. 17). The program of this section is, roughly speaking, to specify the subroutines for quadrature and interpolation and discuss technical details of their implementation. One goal of the construction is to allow error control. The method involves three types of errors:

- interpolation,
- truncation, and
- quadrature errors.

Each of these errors has to be controlled in every time step. This section is structured according to the three types of errors: Subsection 3.3.1 describes the adaptive discretization of the spline, which controls the interpolation error. Subsection 3.3.2 describes the truncation strategy, which determines the truncation error. Finally, subsection 3.3.3 briefly describes how the quadrature error is controlled.

This section considers only a single time step. To simplify notation, the dependence of all variables on the time step number  $i$  is omitted. (That means  $n$  is used instead of  $n(i)$ ,  $a_S$  instead of  $a_S(i)$ , etc.) Variables from the previous time step are referred to with the subscript “prev”.

We focus on the one-dimensional situation ( $d = 1$ ). The notation follows that in chapter 2, i.e.,  $f$  denotes the conditional density function,  $g$  the payoff, etc.

#### 3.3.1. Adaptive spline discretization

The discretization of the log price space  $\mathbb{R}$  takes place on two levels.

- The first level is the discretization by the nodes of the interpolating spline. These nodes are denoted by  $x_1 < \dots < x_n$ .<sup>5</sup>
- The second, finer discretization level is constructed by the quadrature method.

In this subsection only the first discretization level is considered. The second level is discussed later in subsection 3.3.3.

---

<sup>5</sup>Although this is not indicated in the algorithm sketch 2.1, the number of nodes  $n$  may vary for each time step, i.e.,  $n = n_i$ . This detail is omitted in the following to simplify notation.

### Start discretization

The spline is constructed for a finite interval  $[a_S, b_S] \subset \mathbb{R}$ . The choice of  $a_S$  and  $b_S$  is connected to the truncation error and discussed in subsection 3.3.2. As start discretization chosen were  $n$  equidistant nodes  $a_S = x_1 < \dots < x_n = b_S$ .<sup>6</sup> Because of the following adaptive refinement procedure, the initial choice of  $n$  is not important. In fact the minimal start discretization  $n := 2$ ,  $x_1 := a_S$ ,  $x_2 := b_S$  gives good results. The refinement is not sensitive with respect to the start discretization if the option value function is convex as it is the case for plain-vanilla payoffs. For other payoffs, a more conservative choice could be based on the final number of nodes that were used in the previous time slice, e.g.,  $n := \sqrt{n_{prev}}$ , with an arbitrary number for the first time slice, e.g.,  $n := 100$ . This choice is suitable to reduce the risk of premature termination of the refinement procedure.

### Adaptive refinement of the spline nodes

The adaptive refinement is based on a heuristic error estimate. It is assumed that the maximal interpolation error between two successive spline nodes  $x_j$  and  $x_{j+1}$  can be estimated by the interpolation error at the “test point”  $\frac{x_j + x_{j+1}}{2}$ :

**Assumption 3.6** (Estimation of the interpolation error). *Let  $g$  denote the spline interpolant of  $f$  with respect to the nodes  $x_1 < \dots < x_n$ . The interpolation error between two nodes  $x_j$  and  $x_{j+1}$  is estimated as follows:*

$$\left| f\left(\frac{x_j + x_{j+1}}{2}\right) - g\left(\frac{x_j + x_{j+1}}{2}\right) \right| \approx \max_{\xi \in [x_j, x_{j+1}]} (|f(\xi) - g(\xi)|) \quad (\forall j = 1, \dots, n-1)$$

If the error estimate exceeds the given tolerance, an additional node is placed at  $\frac{x_j + x_{j+1}}{2}$ . This leads to a new discretization  $x'_1 < \dots < x'_{n'}$  with  $n' \geq n$  points. If  $n' = n$ , the iterative refinement terminates (cf. the corresponding repeat-until-loop in algorithm 2.1). If  $n' > n$ , the refinement procedure is repeated.

**Remark 3.7** (Termination of refinement). *In practice the procedure terminates after a few iterations. To guarantee termination by means of theorem 3.3, it is required that the fill distance  $\Delta$  converges to 0. This is not necessarily the case for the refinement procedure described above. Therefore, theoretically, the following modification is required: All intervals  $[x_j, x_{j+1}]$  with  $|x_{j+1} - x_j| > \frac{b-a}{\sqrt{n}}$  are refined ( $\Rightarrow \Delta \rightarrow 0$  for  $n \rightarrow \infty$ ).<sup>7</sup>*

<sup>6</sup>As  $x$  denotes log-prices, the start nodes are *not* equidistant with respect to the price  $S$  of the underlying. The step size increases with increasing  $S$ .

<sup>7</sup>In practice this modification does not have a strong impact on the algorithm. When using adaptive nodes, the actual interpolation error is far below the guaranteed error bound from theorem 3.3, which only estimates the error by means of the fill-distance.

### 3.3.2. Control of truncation errors

The subroutine “quadrature” from algorithm sketch 2.1 evaluates the following integral for all spline nodes ( $j = 1, \dots, n$ ):

$$V^H(x_j, t_i) := e^{-r(t_{i+1}-t_i)} \int_{\mathbb{R}} f^{X_{t_{i+1}}|X_{t_i}=x_j}(\xi) V(\xi, t_{i+1}) d\xi, \quad (3.3)$$

where  $f$  is the conditional PDF and  $V(\cdot, t_{i+1})$  is the option value at the next exercise time  $t_{i+1}$ . As the integrand has unbounded support, the integral is improper. It is approximated by truncating the integral to a finite interval  $[a, b]$  and using adaptive Romberg quadrature for  $[a, b]$ . This subsection discusses a suitable choice of  $a$  and  $b$ .

As the integrand depends on the node  $x_j$ , the truncation points  $a = a(x_j) = a_j$  and  $b = b(x_j) = b_j$  also depend on  $j$ . In the following we fix one single node  $x_j$  and only write  $a, b$  instead of  $a_j, b_j$  to simplify notation.

#### Truncation of the improper integrals

As shown later in lemma 5.10 (p. 104), it is possible for all models considered in chapter 5 and all admissible parameter sets to find a constant  $\alpha > 1$  such that  $f(x) = \mathbf{O}(e^{-\alpha|x|})$  for  $|x| \rightarrow \infty$ .<sup>8</sup> In addition to that the following (weak) assumption is needed.

**Assumption 3.8.** *The option price  $V(x, t)$  grows at most exponentially in  $x$  for all  $t$ :*

$$V(x, t) = \mathbf{O}(e^{|x|}) \text{ for } |x| \rightarrow \infty.$$

This holds for all option types which arise in practice.<sup>9</sup> It is possible to conclude the following error estimate for a truncation point  $b \gg 0$  (large enough):

$$\int_b^\infty f(\xi) V(\xi) d\xi \leq \int_b^\infty c e^{(1-\alpha)\xi} d\xi = \frac{c}{\alpha-1} e^{b(1-\alpha)} \quad (3.4)$$

with an unknown constant  $c > 0$ . The residual of the integral decays exponentially with increasing  $b$ . For the other truncation point  $a < 0$  we get an even better result. Taking into account that the option price is bounded for  $\xi \ll 0$  for all usual option types, we can estimate the error caused by truncation at  $a \ll 0$  as follows.

$$\int_{-\infty}^a f(\xi) V(\xi) d\xi \leq \int_{-\infty}^a c e^{-\alpha|\xi|} d\xi = \frac{c}{\alpha} e^{\alpha a} \quad (3.5)$$

<sup>8</sup>The expression “ $= \mathbf{O}(\dots)$  for  $|x| \rightarrow \infty$ ” means that the function is in this class for both limit cases  $x \rightarrow \infty$  and  $x \rightarrow -\infty$ .

<sup>9</sup>This is a well-known fact for plain-vanilla options in a Black-Scholes setting. For example, for convex payoffs (plain-vanilla payoffs are convex) this assumption can be reduced to an assumption for European options and then be verified by the limit behavior of the analytic solution of the Black-Scholes formula. For all other options that arise in practice, this assumption obviously holds. An exact specification of these options would introduce an unnecessary formalism.

with an unknown constant  $c > 0$ . Together, this leads to the total error estimate for truncation at  $a \ll 0$  and  $b \gg 0$ :

$$\int_{-\infty}^{\infty} f(\xi)V(\xi) \, d\xi = \int_a^b f(\xi)V(\xi) \, d\xi + \mathbf{O}(e^{\alpha a} + e^{b(1-\alpha)})$$

### Basic truncation strategy

Suitable truncation parameters  $(a, b)$  can be found by a simple search strategy. We evaluate the integrand at a sequence of points  $(a_\nu, b_\nu) := (-2^{\nu-1}, 2^{\nu-1})$ , until  $f(a_\nu)V(a_\nu) < \varepsilon$  and  $f(b_\nu)V(b_\nu) < \varepsilon$  with a sufficiently small tolerance  $\varepsilon \approx 0$ . This heuristic strategy works well in practice.

### Improved truncation strategy

An improved truncation strategy can be derived by assuming the following limit behavior for the integrand:

$$h(x) := f(x)V(x) \approx ce^{(1-\alpha)x} \text{ for } x \gg 0$$

with unknown constants  $c$  and  $\alpha$ .<sup>10</sup> The same doubling sequence  $(a_\nu, b_\nu)$  as above is used. Evaluating  $h$  at two points  $b_\nu$  and  $b_{\nu+1} := 2b_\nu$  yields

$$\begin{aligned} h_\nu &= ce^{(1-\alpha)b_\nu}, \\ h_{\nu+1} &= ce^{(1-\alpha)2b_\nu} \\ \Rightarrow \alpha &= 1 + \frac{1}{b_\nu} \log \frac{h_\nu}{h_{\nu+1}}, \quad c = \frac{h_\nu^2}{h_{\nu+1}}. \end{aligned}$$

Substituting  $\alpha$  and  $c$  into estimate (3.4) of the truncation error leads to the following terminating condition for the truncation point  $b_\nu$ :

$$\frac{c}{\alpha-1} e^{b_\nu(1-\alpha)} < \frac{\varepsilon}{2},$$

where  $\varepsilon$  is a prescribed tolerance for the truncation error. The lower truncation point  $a$  can be treated in a similar way by using the approximation  $h(x) \approx ce^{\alpha x}$  for  $x \ll 0$ :

$$\begin{aligned} h_\nu &= ce^{\alpha a_\nu}, \\ h_{\nu+1} &= ce^{2\alpha a_\nu} \\ \Rightarrow \alpha &= -\frac{1}{a_\nu} \log \frac{h_\nu}{h_{\nu+1}}, \quad c = \frac{h_\nu^2}{h_{\nu+1}} \end{aligned}$$

Substituting into eq. (3.5) leads to the terminating condition for the truncation point  $a_\nu$ :

$$\frac{c}{\alpha} e^{\alpha a_\nu} < \frac{\varepsilon}{2}$$

<sup>10</sup>This assumption fails obviously at the first time step of a Bermudan option with plain-vanilla payoff: At maturity is  $V(x) = 0$  for  $x \ll 0$  (call) or for  $x \gg 0$  (put); but in this case the corresponding truncation is not necessary. For all subsequent times  $V(x) > 0$  is given for all  $x$ , as the probability density function  $f(x)$  is strictly positive for all  $x$ .

### Choice of the spline interval

For all usual option types (and assuming  $r > 0$ ) there is one optimal exercise point  $x^*$ , i.e., exactly one solution of

$$V^H(x^*) = g(x^*). \quad (3.6)$$

In the following we discuss the case of a put (plain-vanilla or binary put).<sup>11</sup> Figure 3.1 gives an illustration of a time step  $t_{i+1} \rightarrow t_i$  in this case. For puts we have

$$V^H(x) < g(x) \text{ for } x < x^*.$$

Consequently, the option is exercised for  $x < x^*$ . The first node of the spline can be chosen as  $a_S := x^*$  and the interpolation function can be continued arbitrarily, e.g., by setting  $V^H(x) := 0$  on  $(-\infty, a_S)$ . As the option is exercised for  $x < a_S$ , the value of the option does not depend on the hold value in this region. The value  $a_S$  is then defined by eq. (3.6) and can be approximated numerically by a Newton method. (Both functions are smooth in the neighborhood of  $a_S$ ; a good start value is given by  $a_S^{prev}$  from the previous time step.) The choice of  $b_S$  and the continuation of the spline on  $(b_S, \infty)$  are a bit more complicated. The hold value has the limit behavior  $V^H(x) \rightarrow 0$  for  $x \rightarrow \infty$ . This justifies the continuation  $V^H(x) := 0$  for  $x > b_S$ . To prevent this choice from introducing new errors, it must be guaranteed that

$$b_S > B := \max_{j \in \{1, \dots, n\}} (b_j). \quad (3.7)$$

As  $B$  is not known in advance when the spline is being constructed, it is assumed  $b_S \approx b_S^{prev}$  and the guess

$$b_S := 2b_S^{prev}$$

with a safety factor 2 is used. Later, in the quadrature routine, the new truncation points  $b_j$  are known and condition (3.7) can be verified a posteriori. If it is violated, the spline is rejected and a new spline is constructed with  $b_S := B$ .

---

<sup>11</sup>Call options can be handled analogously.

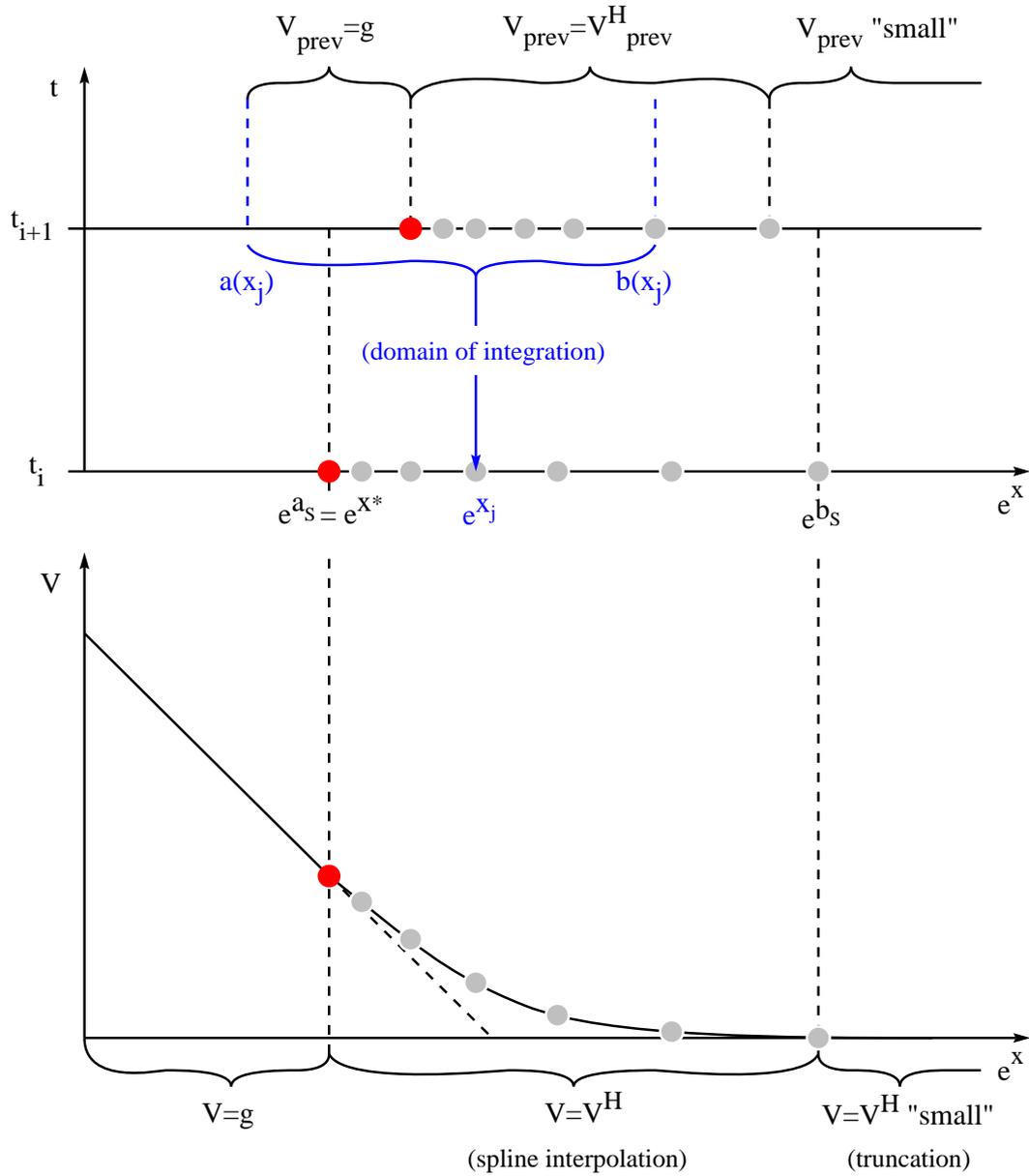


Figure 3.1.: Interpolation and integration in each time step of the spline method.

### 3.3.3. Control of quadrature errors

This subsection briefly describes the discretization for the quadrature method, which controls the quadrature error.

#### Splitting up integration

Consider the integrand in (3.3)

$$h(\xi) := f(\xi)V_{prev}(\xi) = f(\xi) \max(V_{prev}^H(\xi), g(\xi)).$$

This function is only piecewise  $\mathcal{C}^\infty$ -smooth.<sup>12</sup> To retain the full (adaptive) order of the Romberg quadrature, it is necessary to split the interval  $[a, b]$  at the spline nodes  $x_1^{prev} < \dots < x_n^{prev}$  from the previous time slice. Let  $x_A^{prev}$  denote the first and  $x_B^{prev}$  the last node in  $[a, b]$ . The truncated integral can then be divided into

$$\int_a^b h(\xi) \, d\xi = \int_a^{x_A} h(\xi) \, d\xi + \sum_{\nu=A}^{B-1} \int_{x_\nu}^{x_{\nu+1}} h(\xi) \, d\xi + \int_{x_B}^b h(\xi) \, d\xi.$$

The integrand is  $\mathcal{C}^\infty$ -smooth on each of the intervals  $(a, x_A)$ ,  $(x_A, x_{A+1})$ ,  $\dots$ ,  $(x_B, b)$ . For each of these subintervals, Romberg quadrature can be applied.

#### Adaptive refinement of the discretization (second level)

The discretization implied by the spline interpolation is refined adaptively to control the quadrature error. On each interval  $[x_\nu, x_{\nu+1}]$  the Romberg method uses  $2^{N_\nu}$  equidistant quadrature points. The discretization size  $N_\nu$  is determined by the error estimate in the Romberg scheme. It is based on the comparison of integral approximations with several consecutive discretizations (with  $2^{N_\nu}$ ,  $2^{N_\nu-1}$ ,  $2^{N_\nu-2}$ ,  $\dots$  points, respectively). The discretization is considered fine enough, if the difference of several consecutive approximations is below the given error tolerance.

### 3.3.4. Summary

In this section the subroutines “quadrature” and “interpolation” have been specified. By construction the three types of errors can be controlled in each time step. A missing link is the smoothness of the hold value  $V^H$ . The next section fills this gap and analyzes the numerical properties of the spline method.

---

<sup>12</sup>Here, the density function  $f$  is assumed to be piecewise smooth,  $g$  is assumed to be piecewise smooth. It is shown in section 3.4 that  $V^H \in \mathcal{C}^\infty$  under rather general conditions.

### Illustration of adaptive spline nodes

Figure 3.2 illustrates the adaptive spline nodes generated by the spline method for a Bermudan plain-vanilla put option. The optimal exercise boundary is given by the leftmost nodes  $a_S(i)$ . The rightmost nodes are the truncation points  $b_S(i)$  of the spline interpolation. The discrete structure of the node distances is caused by the adaptive spline discretization described in 3.3.1, which inserts new nodes between two existing nodes. The figure illustrates well that the adaptive spline method automatically places more nodes near the optimal exercise boundary and near the strike than, e.g., far out of money.

The option parameters can be found in table 6.7 (p. 124). The exercise structure is Bermudan with  $m = 100$  equidistant exercise times, error tolerance is  $\varepsilon = 10^{-6}$ . The spline method required for this example 9656 nodes, i.e., on average only 96 nodes per time step. The number of required Romberg integrations for this problem is of order  $100 \times 96^2 \approx 1000000$ .

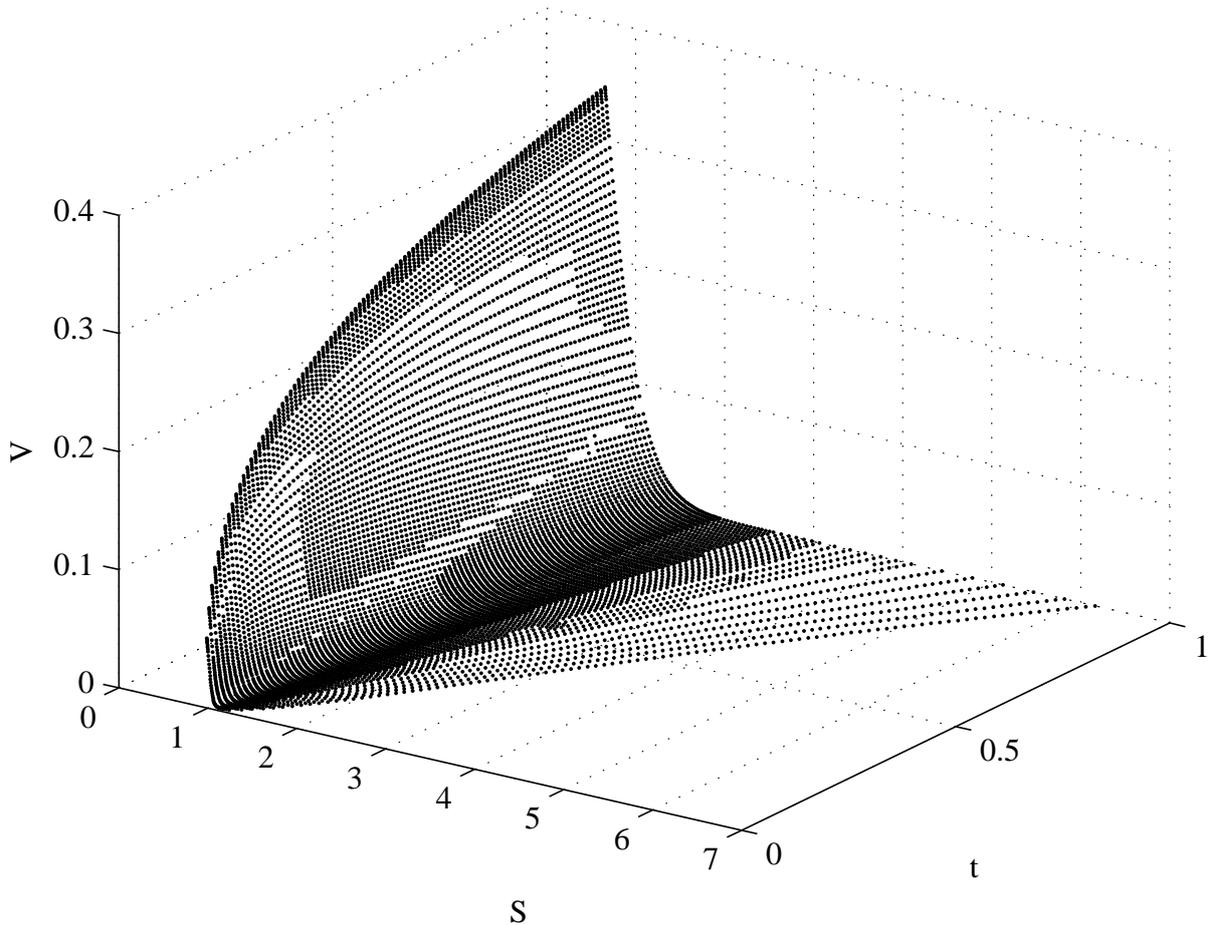


Figure 3.2.: Adaptive spline nodes for a Bermudan option.

## 3.4. Numerical properties

This section analyzes the numerical properties of the spline method (error control, stability, and computational complexity). Subsection 3.4.1 shows that the method can be used to value Bermudan options up to an arbitrary prescribed error bound  $\varepsilon$ . This implies the convergence of the method for  $\varepsilon \rightarrow 0$ . Subsection 3.4.2 unleashes theorems 3.3 and 3.5 by proving that under rather general conditions the hold value of a Bermudan option is  $\mathcal{C}^\infty$ -smooth. Subsection 3.4.3 discusses further stability issues of the valuation algorithm. The computational complexity and numerical efficiency is analyzed in subsection 3.4.4.

### 3.4.1. Convergence/error control

The spline method contains two adaptive mechanisms to control the accuracy of the valuation method. The first one controls the interpolation errors, and the second one controls quadrature errors. The following result describes how the method can be used to compute an option value with a given error tolerance.

**Theorem 3.9** (Error control). *Under the general assumptions from section 2.1, the spline method can be used to approximate the value  $V(x, t)$  of Bermudan plain-vanilla options within a predetermined absolute error bound  $\varepsilon > 0$ , i.e., the resulting approximation  $W(x, t)$  satisfies*

$$\|V(x, t) - W(x, t)\|_\infty \leq \varepsilon \text{ for all } t \in \mathcal{T},$$

where  $\mathcal{T}$  denotes the set of exercise times.

*Proof.* By construction: Let  $\mathcal{T} = \{t_1, \dots, t_m\}$  denote the set of exercise times of the Bermudan option including maturity  $t_m = T$ . It is sufficient to show that  $\forall t_i \in \mathcal{T}$

$$\|V(x, t_i) - W(x, t_i)\|_\infty \leq \frac{(m-i)\varepsilon}{m}. \quad (3.8)$$

Induction  $i+1 \rightarrow i$ : Assume (3.8) for  $i+1$ . Writing briefly  $D := e^{-r(t_{i+1}-t_i)}$  for the discount factor and  $f(\xi, x) := f^{X_{t_{i+1}}|X_{t_i}=x}(\xi)$  for the conditional density, the exact price  $V(x, t_i)$  is (following lemma 2.12):

$$V(x, t_i) = \max \left( g(x), D \int_{-\infty}^{\infty} f(\xi, x) V(\xi, t_{i+1}) \, d\xi \right)$$

Let  $W_0$  denote

$$W_0(x, t_i) := \max \left( g(x), D \int_{-\infty}^{\infty} f(\xi, x) W(\xi, t_{i+1}) \, d\xi \right)$$

The absolute error of  $W_0$  summarizes the error arising from previous time steps and can be estimated as follows.

$$\begin{aligned}
|V(x, t_i) - W_0(x, t_i)| &\leq \left| D \int_{-\infty}^{\infty} f(\xi, x) V(\xi, t_{i+1}) \, d\xi - D \int_{-\infty}^{\infty} f(\xi, x) W(\xi, t_{i+1}) \, d\xi \right| \\
&\leq \int_{-\infty}^{\infty} f(\xi, x) |V(\xi, t_{i+1}) - W(\xi, t_{i+1})| \, d\xi \\
&\leq \int_{-\infty}^{\infty} f(\xi, x) \frac{(m-i-1)\varepsilon}{m} \, d\xi \quad (\text{induction}) \\
&= \frac{(m-i-1)\varepsilon}{m} \underbrace{\int_{-\infty}^{\infty} f(\xi, x) \, d\xi}_{=1} \\
&= \frac{(m-i-1)\varepsilon}{m}
\end{aligned}$$

for all  $x$ . This implies that additive errors in  $W$  are not amplified. The value  $W$  at time  $t_i$  can be written as

$$W(x, t_i) = \max \{g(x), D \cdot \text{Quad}_a^b[f(\xi, x)W_{spline}(\xi, t_{i+1})]\},$$

where  $W_{spline}$  denotes the interpolating function and  $\text{Quad}_a^b$  the quadrature method. The additional errors introduced by interpolation, truncation, and quadrature at exercise time  $t_i$  are:

1. Interpolation error:<sup>13</sup>

$$\begin{aligned}
\varepsilon_1(x) &:= \left| \int_{-\infty}^{\infty} f(\xi, x) W(\xi, t_{i+1}) \, d\xi - \int_{-\infty}^{\infty} f(\xi, x) W_{spline}(\xi, t_{i+1}) \, d\xi \right| \\
&\leq \|W - W_{spline}\|_{\infty}
\end{aligned}$$

2. Truncation error:

$$\varepsilon_2(x) := \left| \int_{-\infty}^{\infty} f(\xi, x) W_{spline}(\xi, t_{i+1}) \, d\xi - \int_a^b f(\xi, x) W_{spline}(\xi, t_{i+1}) \, d\xi \right|$$

3. Quadrature error:

$$\varepsilon_3(x) := \left| \int_a^b f(\xi, x) W_{spline}(\xi, t_{i+1}) \, d\xi - \text{Quad}_a^b[f(\xi, x)W_{spline}(\xi, t_{i+1})] \right|$$

As discussed in section 3.3, each of these errors can be controlled.<sup>14</sup> The interpolation error  $\varepsilon_1$  is controlled using the adaptive refinement strategy described in section 3.3.1. The truncation error  $\varepsilon_2$  is controlled by the truncation strategy described in section 3.3.2. Finally, the quadrature error  $\varepsilon_3$  is controlled by the second level discretization of the

<sup>13</sup>Here implicitly enters assumption 3.6 (p. 36) for the estimates of the local interpolation errors.

<sup>14</sup>The assumption  $V^H \in \mathcal{C}^{\infty}$  is verified in section 3.4.2.

Romberg method mentioned in section 3.3.3. The composite error can be estimated as follows:

$$\begin{aligned} |W_0(x, t_i) - W(x, t_i)| &\leq \left| \int_{-\infty}^{\infty} f(\xi, x)W(\xi, t_{i+1}) \, d\xi - \text{Quad}_a^b[f(\xi, x)W_{\text{spline}}(\xi, t_{i+1})] \right| \\ &\leq \varepsilon_1(x) + \varepsilon_2(x) + \varepsilon_3(x) \end{aligned}$$

All three types of errors can be controlled<sup>15</sup> to satisfy  $\|\varepsilon_{\{1,2,3\}}\|_{\infty} < \frac{\varepsilon}{3m}$  and consequently

$$\begin{aligned} \|V(x, t_i) - W(x, t_i)\|_{\infty} &\leq \|V(x, t_i) - W_0(x, t_i)\|_{\infty} + \|W_0(x, t_i) - W(x, t_i)\|_{\infty} \\ &\leq \frac{(m-i-1)\varepsilon}{m} + \|\varepsilon_1\|_{\infty} + \|\varepsilon_2\|_{\infty} + \|\varepsilon_3\|_{\infty} \\ &\leq \frac{(m-i)\varepsilon}{m}. \end{aligned}$$

□

This proof is based on the fact that for *sufficient smoothness* of the option's hold value in each time slice the errors made by interpolation and quadrature can be controlled. The smoothness of the hold value is analyzed in the next subsection.

### 3.4.2. Smoothness of the hold value $V^H$

This section analyzes the hold value of a Bermudan option at exercise time  $t_i$

$$V^H(x, t_i) := e^{-r(t_{i+1}-t_i)} \int_{\mathbb{R}^d} f^{X_{t_{i+1}}|X_{t_i}=x}(\xi) V(\xi, t_{i+1}) \, d\xi \quad (\text{cf. lemma 2.12})$$

and shows that  $V^H(x, t_i) \in \mathcal{C}^{\infty}(\mathbb{R}^d)$  for each  $t_i \in \mathcal{T}$  under rather general conditions. This is a property of the valuation *problem* and not a property of the valuation *method*. Nevertheless, this result  $V^H(x, t_i) \in \mathcal{C}^{\infty}(\mathbb{R}^d)$  is important to guarantee convergence of the *method*, as the error estimates in theorem 3.3 and 3.5 apply only to sufficiently smooth functions.

The integral representation 2.9 suggests the notation of convolutions. Unfortunately, payoffs and option prices are in general not absolutely integrable functions.<sup>16</sup> Therefore we need to generalize some results for integrable functions to *locally integrable* functions.

**Definition 3.10** (Local integrability). *A function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is locally integrable, if it is integrable on any compact set  $K \subset \mathbb{R}^d$ . The set of locally integrable functions on  $\mathbb{R}^d$  is denoted by  $\mathcal{L}_{loc}^1(\mathbb{R}^d)$ .*

<sup>15</sup>Using assumptions that will be discussed in the next subsections.

<sup>16</sup>For example, the payoff of a plain-vanilla call is not absolutely integrable.

The integral term in the hold value  $V^H$  can be written as convolution of the conditional density function  $f$  and the option value  $V$  at the subsequent exercise time  $t_{i+1}$ , and by lemma 2.6 (p. 12):

$$\begin{aligned} \int_{\mathbb{R}^d} f^{X_{t_{i+1}}|X_{t_i}=x}(\xi)V(\xi, t_{i+1}) \, d\xi &= \int_{\mathbb{R}^d} \underbrace{f^{X_{t_{i+1}}|X_{t_i}=0}(\xi-x)}_{=: \tilde{f}(x-\xi) \geq 0} \underbrace{V(\xi, t_{i+1})}_{=: g(\xi) \geq 0} \, d\xi \quad (\text{lemma 2.6}) \\ &=: \tilde{f} * g(x), \end{aligned}$$

where  $\tilde{f}$  denotes the mirror function of  $f$ . This technical detail is only required to stay consistent with the usual notation of convolution. As properties like integrability, smoothness, boundedness, etc., are invariant under mirroring, we can omit the tilde from  $\tilde{f}$  in the following and simply write  $f$ . The convolution can be defined formally as follows.

**Definition 3.11** (Convolution). *For  $f, g \in \mathcal{L}_{loc}^1(\mathbb{R}^d)$  the convolution  $f * g$  is given by*

$$f * g(x) := \int_{\mathbb{R}^d} f(x-y)g(y) \, dy,$$

*if the function  $f(x-\cdot)g(\cdot)$  is integrable and by  $f * g(x) := 0$  otherwise.*

For the (standard) case  $f, g \in \mathcal{L}^1(\mathbb{R}^d)$  one can show  $f * g \in \mathcal{L}^1(\mathbb{R}^d)$  as well as some basic properties of the convolution like commutativity, distributivity, and associativity.<sup>17</sup> The case  $f \notin \mathcal{L}^1$  is not covered by textbooks on functional analysis, but in the following it is shown that  $f * g \in \mathcal{C}^\infty(\mathbb{R}^d)$  even for  $f \notin \mathcal{L}^1$  under the additional premises of theorem 3.16. First, some prerequisites for the proof of theorem 3.16 are introduced. These prerequisites are modified versions of theorems from standard textbooks.

**Theorem 3.12** (Continuity of translation). *Let  $1 \leq p < \infty$  and  $f \in \mathcal{L}^p(\mathbb{R}^d)$ . Then the mapping*

$$\begin{aligned} \theta : \mathbb{R}^d &\rightarrow \mathcal{L}^p(\mathbb{R}^d) \\ y &\mapsto f(\cdot - y) \end{aligned}$$

*is uniformly continuous.*

*Proof.* A proof of the one-dimensional case  $d = 1$  can be found in [Rud87], p. 182, theorem 9.5. It is based on the fact that the set of continuous functions with compact support  $\mathcal{C}_c(\mathbb{R}^d) \subset \mathcal{L}^p(\mathbb{R}^d)$  is dense. The proof can be generalized to  $d > 1$  as follows:

Fix  $\varepsilon \in (0, 1)$ . As  $f \in \mathcal{L}^p$ , there exists a continuous function  $g \in \mathcal{C}_c$  with compact support, such that

$$\|f - g\|_p < \varepsilon.$$

---

<sup>17</sup>See e.g., [Els02], p. 193f.

As the support of  $g$  is compact, there exists an  $A > 0$  such that  $g(x) = 0$  for all  $x$  with  $\|x\| > A$ . By uniform continuity of  $g$  there exists a  $\delta \in (0, A)$  such that  $\|s - t\| < \delta$  implies

$$|g(s) - g(t)| < (3A)^{-d/p} \varepsilon.$$

Then for  $\|s - t\| < \delta$  follows that

$$\begin{aligned} \|g(\cdot - s) - g(\cdot - t)\|_p &= \left( \int_{\mathbb{R}^p} |g(x - s) - g(x - t)|^p dx \right)^{1/p} \\ &= \left( \int_{B(s,A) \cup B(t,A)} |g(x - s) - g(x - t)|^p dx \right)^{1/p} \\ &< \left( \int_{B(s,A) \cup B(t,A)} (3A)^{-d} \varepsilon^p dx \right)^{1/p} \\ &< (3A)^{-d/p} \varepsilon (2A + \delta)^{d/p} < \varepsilon, \end{aligned}$$

where  $B(s, A)$  is the ball with center  $s$  and radius  $A$ . As  $\mathcal{L}^p$ -norms are translation-invariant, then

$$\begin{aligned} \|f(\cdot - s) - f(\cdot - t)\|_p &\leq \|f(\cdot - s) - g(\cdot - s)\|_p + \|g(\cdot - s) - g(\cdot - t)\|_p + \\ &\quad \|g(\cdot - t) - f(\cdot - t)\|_p \\ &= \|f - g\|_p + \|g(\cdot - s) - g(\cdot - t)\|_p + \|g - f\|_p < 3\varepsilon \end{aligned}$$

for  $\|s - t\| < \delta$ . □

**Corollary 3.13.** *If  $1 \leq p \leq \infty$ ,  $\frac{1}{p} + \frac{1}{q} = 1$ ,  $f \in \mathcal{L}^p(\mathbb{R}^d)$  and  $g \in \mathcal{L}^q(\mathbb{R}^d)$ , then  $f * g$  is uniformly continuous.<sup>18</sup>*

*Proof.* Let  $p < \infty$  (w.l.o.g.). Fix  $\varepsilon > 0$ . Then for  $x, x' \in \mathbb{R}^d$

$$\begin{aligned} |f * g(x) - f * g(x')| &= \left| \int_{\mathbb{R}^d} f(x - y)g(y) dy - \int_{\mathbb{R}^d} f(x' - y)g(y) dy \right| \\ &= \left| \int_{\mathbb{R}^d} [f(x - y) - f(x' - y)]g(y) dy \right| \\ &\leq \| [f(x - \cdot) - f(x' - \cdot)]g(\cdot) \|_1 \\ &\leq \|f(x - \cdot) - f(x' - \cdot)\|_p \|g\|_q \quad (\text{Hölder's inequality}) \end{aligned}$$

By theorem 3.12 there exists a  $\delta > 0$  such that  $\|f(x - \cdot) - f(x' - \cdot)\|_p < \varepsilon / \|g\|_q$  for all  $x, x'$  with  $\|x - x'\| < \delta$ . □

---

<sup>18</sup>It is useful to formally include the cases  $p = \infty$  and  $q = \infty$ .

**Definition 3.14.** *In the following let for sufficiently partially differentiable  $f : \mathbb{R}^d \rightarrow \mathbb{R}$*

$$D^\alpha f := \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \circ \dots \circ \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}} f$$

*denote the partial derivatives of  $f$ , where  $\alpha := (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$  is a multi-index.*

**Lemma 3.15** (Smoothness of convolution). *Let  $g \in \mathcal{L}^1(\mathbb{R}^d)$  be bounded and  $f \in \mathcal{C}^\infty(\mathbb{R}^d, \mathbb{R})$ . Furthermore, let all derivatives  $D^\alpha f$  be uniformly continuous and bounded. Then*

$$f * g \in \mathcal{C}^\infty(\mathbb{R}^d, \mathbb{R}) \quad \text{and} \quad D^\alpha(f * g) = (D^\alpha f) * g \quad \text{for all } \alpha.$$

*Proof.* Let  $D_j f := \frac{\partial}{\partial x_j} f$  denote the partial derivative with respect to  $x_j$ . Because of uniform continuity of  $D_j f$  for all  $\varepsilon > 0$  there exists a  $\delta > 0$ , such that

$$|D_j f(u) - D_j f(v)| < \varepsilon \quad \text{for all } u, v \in \mathbb{R}^d$$

with  $\|u - v\| < \delta$ . Denoting the  $j$ -th unit vector of  $\mathbb{R}^d$  by  $e_j$  we have for all  $0 \neq t \in \mathbb{R}$ ,  $|t| < \delta$  and for all  $x \in \mathbb{R}^d$ :

$$\begin{aligned} & \left| \frac{1}{t} [f * g(x + te_j) - f * g(x)] - (D_j f) * g(x) \right| \\ &= \left| \int_{\mathbb{R}^d} g(y) \frac{1}{t} \int_0^t (D_j f(x - y + se_j) - D_j f(x - y)) \, ds \, dy \right| \leq \varepsilon \|g\|_1. \end{aligned}$$

Consequently,  $f * g$  is partially differentiable and  $D_j(f * g) = (D_j f) * g$  for all  $j = 1, \dots, d$ . Following corollary 3.13 ( $D_j f \in \mathcal{L}^\infty$ ,  $g \in \mathcal{L}^1$ ) this function is uniformly continuous. As all partial derivatives exist and are continuous, the function  $f * g$  is continuously differentiable. The same arguments can be applied repeatedly to the functions  $D_j f$  and  $g$  in order to complete the proof for arbitrary higher derivatives  $D^\alpha(f * g)$ .  $\square$

**Theorem 3.16** (Smoothness of hold value). *Let  $f \in \mathcal{C}^\infty(\mathbb{R}^d, \mathbb{R})$  and let all derivatives  $D^\alpha f$  be uniformly continuous and bounded. Let  $g : \mathbb{R}^d \rightarrow \mathbb{R}_0^+$  be locally bounded and locally  $\mathcal{L}^1$ -integrable, and let  $f$  be fast declining with respect to  $g$ , that means let*

$$\text{the integrals in } (D^\alpha f) * g \text{ and consequently } |D^\alpha f| * g \text{ exist for all } \alpha. \quad (3.9)$$

**Then:**

$$\begin{cases} f * g \in \mathcal{C}^\infty \text{ and} \\ D^\alpha(f * g) = (D^\alpha f) * g \end{cases} \quad (3.10)$$

*Proof.* The proof is structured as follows. We construct a sequence  $(g_n)$  of functions and then prove:

- a)  $D^\alpha(f * g_n) = (D^\alpha f) * g_n$  for all  $\alpha$  and all  $n$ ,
- b)  $|D^\alpha f| * g_n \rightarrow |D^\alpha f| * g$  pointwise for  $n \rightarrow \infty$  for all  $\alpha$ ,  
(and consequently  $(D^\alpha f) * g_n \rightarrow (D^\alpha f) * g$  pointwise)
- c)  $(D^\alpha f) * g$  is continuous for all  $\alpha$ ,  
(and consequently  $|D^\alpha f| * g$  continuous)
- d)  $|D^\alpha f| * g_n \rightarrow |D^\alpha f| * g$  locally uniformly for  $n \rightarrow \infty$  for all  $\alpha$ ,
- e)  $D^\alpha(f * g_n) \rightarrow D^\alpha(f * g)$  pointwise for  $n \rightarrow \infty$  for all  $\alpha$ .

Then a), b), and e) imply  $D^\alpha(f * g) = (D^\alpha f) * g$  which together with c) implies  $f * g \in \mathcal{C}^\infty$ . d) is a lemma needed to show e).

Now defining a suitable sequence  $(g_n)$  is the first step. Let  $g_n := g \cdot \mathbf{1}_{\{x \in \mathbb{R}^d, \|x\| \leq n\}}$  denote the sequence of functions equal to  $g$  on the ball with center 0 and radius  $n$ , and equal to 0 outside. As  $g$  is locally  $\mathcal{L}^1$ -integrable, we have  $g_n \in \mathcal{L}^1(\mathbb{R}^d)$  for all  $n$ .

**ad a)**

For every  $n \in \mathbb{N}$  the function  $g_n$  is bounded being a restriction of a locally bounded function  $g$  to a compact set. Consequently, lemma 3.15 applies to  $f$  and  $g_n$ .

**ad b)**

This can be proved by Lebesgue's dominated convergence theorem. For every fixed  $x \in \mathbb{R}^d$ ,  $\alpha \in \mathbb{N}^d$  let  $(h_n)_{n \geq 1}$  denote the sequence of functions defined by

$$h_n(\xi) := |D^\alpha f(x - \xi)| g_n(\xi).$$

As mentioned above,  $g_n$  is bounded. Furthermore,  $g_n \in \mathcal{L}^1(\mathbb{R}^d)$ ,  $D^\alpha f \in \mathcal{L}^\infty(\mathbb{R}^d)$ , and thus<sup>19</sup>  $h_n \in \mathcal{L}^1(\mathbb{R}^d)$ . Let

$$h(\xi) := |D^\alpha f(x - \xi)| g(\xi).$$

Then  $h_n \rightarrow h$  pointwise, as  $g_n \rightarrow g$  pointwise for  $n \rightarrow \infty$ . Function  $h$  dominates the sequence and is integrable itself by condition 3.9. Consequently (by dominated convergence):

$$\begin{aligned} \int_{\mathbb{R}^d} h(\xi) \, d\xi &= \lim_{n \rightarrow \infty} \int_{\mathbb{R}^d} h_n(\xi) \, d\xi \quad (\forall x) \\ \Leftrightarrow |D^\alpha f| * g_n &\rightarrow |D^\alpha f| * g \text{ pointwise for } n \rightarrow \infty. \end{aligned}$$

---

<sup>19</sup>By Hölder's inequality.

This also implies

$$\begin{aligned} & |D^\alpha f| * (g - g_n) \rightarrow 0 \\ \Rightarrow & |(D^\alpha f) * (g - g_n)| \rightarrow 0 \\ \Leftrightarrow & (D^\alpha f) * g_n \rightarrow (D^\alpha f) * g \quad \text{pointwise for } n \rightarrow \infty. \end{aligned}$$

**ad c)**

It is sufficient to show that for each compact set  $K \subset \mathbb{R}^d$

$$\exists c > 0 \forall x, x' \in K : |(D^\alpha f) * g(x) - (D^\alpha f) * g(x')| \leq c \|x' - x\|.$$

By definition we have

$$\begin{aligned} & |(D^\alpha f) * g(x) - (D^\alpha f) * g(x')| \\ = & \left| \int_{\mathbb{R}^d} D^\alpha f(x - y) g(y) \, dy - \int_{\mathbb{R}^d} D^\alpha f(x' - y) g(y) \, dy \right| \\ = & \left| \int_{\mathbb{R}^d} [D^\alpha f(x - y) - D^\alpha f(x' - y)] g(y) \, dy \right|. \end{aligned}$$

Next we use an integral representation for the difference of the function values  $D^\alpha f$  at  $(x - y)$  and  $(x' - y)$  along the path  $\gamma(t) := t(x' - y) + (1 - t)(x - y) = tx' + (1 - t)x - y$ ,  $t \in [0, 1]$ . With  $\dot{\gamma}(t) = x' - x$  follows:

$$= \left| \int_{\mathbb{R}^d} g(y) \int_0^1 [\nabla D^\alpha f(\gamma(t))]^{tr} (x' - x) \, dt \, dy \right|$$

Integration over  $t$  and  $y$  can be swapped (a consequence of Fubini's and Tonelli's theorems):

$$= \left| \int_0^1 \int_{\mathbb{R}^d} [g(y) \nabla D^\alpha f(\gamma(t))]^{tr} (x' - x) \, dy \, dt \right|$$

As  $(x' - x)$  does not depend on  $y$  and  $t$ :

$$= \left| \int_0^1 \int_{\mathbb{R}^d} [g(y) \nabla D^\alpha f(\gamma(t))]^{tr} \, dy \, dt (x' - x) \right|$$

With  $D^{\alpha_1} f := \frac{\partial}{\partial x_1} D^\alpha f$ , ...,  $D^{\alpha_d} := \frac{\partial}{\partial x_d} D^\alpha f$  this can be written as

$$\begin{aligned} & = \left| \int_0^1 \left( \int_{\mathbb{R}^d} g(y) D^{\alpha_1} f(\gamma(t)) \, dy, \dots, \int_{\mathbb{R}^d} g(y) D^{\alpha_d} f(\gamma(t)) \, dy \right)^{tr} dt (x' - x) \right| \\ & = \left| \left( \int_0^1 h_1(t) \, dt, \dots, \int_0^1 h_d(t) \, dt \right)^{tr} (x' - x) \right| \end{aligned}$$

with  $h_j(t) := (D^{\alpha_j} f) * g(tx' + (1 - t)x)$  for  $j = 1, \dots, d$ . It can be shown that the functions  $(D^{\alpha_j} f) * g$  are bounded on  $K$ . The proof is short but a bit technical and is deferred to

lemma 3.17 (p. 52). Now, with  $c_j := \sup_{x \in K} |(D^{\alpha_j} f) * g(x)|$  and by triangle inequality the estimation below results.

$$\begin{aligned} &\leq \sum_{j=1}^d c_j \cdot |(x' - x)_j| \\ &\leq \sum_{j=1}^d c_j \|x' - x\|_{\infty} \\ &\leq \underbrace{d \cdot \max(c_1, \dots, c_d)}_{=: c} \|x' - x\|_{\infty} = c \|x' - x\| \end{aligned}$$

**ad d)**

The space  $\mathbb{R}^d$  is locally compact. In locally compact spaces local uniform convergence is the same as compact convergence.<sup>20</sup> Thus it is sufficient to show that for every compact subset  $K \subset \mathbb{R}^d$

$$(|D^{\alpha} f| * g_n)|_K \rightarrow (|D^{\alpha} f| * g)|_K \text{ uniformly,}$$

which is a consequence of Dini's theorem.  $K$  is compact,  $|D^{\alpha} f| * g$  is continuous as well as  $|D^{\alpha} f| * g_n$  is continuous for all  $n$ ,  $(|D^{\alpha} f| * g_n)_n$  is a monotonically increasing sequence of functions, and  $|D^{\alpha} f| * g = \sup_{n \geq 1} |D^{\alpha} f| * g_n$  pointwise. Dini's theorem implies uniform convergence.

**ad e)**

First, locally uniform convergence of  $D^{\alpha}(f * g_n) = (D^{\alpha} f) * g_n$  is shown for all  $\alpha$ .

$$\begin{aligned} &(D^{\alpha} f) * g_n \rightarrow (D^{\alpha} f) * g \quad (\text{locally uniformly}) \\ \Leftrightarrow &(D^{\alpha} f) * g_n - (D^{\alpha} f) * g \rightarrow 0 \quad (\text{l.u.}) \\ \Leftrightarrow &|(D^{\alpha} f) * g_n - (D^{\alpha} f) * g| \rightarrow 0 \quad (\text{l.u.}) \\ \Leftrightarrow &|(D^{\alpha} f) * (g_n - g)| \rightarrow 0 \quad (\text{l.u.}) \\ \Leftarrow &|D^{\alpha} f| * (g - g_n) \rightarrow 0 \quad (\text{l.u.}), \text{ given by d).} \end{aligned}$$

The sequence of functions  $f * g_n \in \mathcal{C}^{\infty}$  converges pointwise by b). Its partial derivatives converge locally uniformly as shown above. Then the limit  $f * g$  is continuously differentiable and

$$D^{\alpha}(f * g) = \lim_{n \rightarrow \infty} D^{\alpha}(f * g_n) \text{ for all } \alpha \in \mathbb{N}^d.$$

This completes the proof of theorem 3.16. □

<sup>20</sup>Definitions: A space is locally compact  $:\Leftrightarrow$  every point has a neighborhood whose closure is compact.  
 $f_n \rightarrow f$  locally uniformly  $:\Leftrightarrow$  for every point there exists a neighborhood on which  $f_n \rightarrow f$  uniformly.  
 $f_n \rightarrow f$  compactly  $:\Leftrightarrow f_n \rightarrow f$  uniformly on every compact set.

**Lemma 3.17** (Technical supplement to step **c**) of the above proof). *Under the conditions of theorem 3.16, the function  $(D^\alpha f) * g$  is bounded on each compact set  $K \subset \mathbb{R}^d$  for all  $\alpha \in \mathbb{N}^d$ .*

*Proof.*  $(D^\alpha f) * g_n$  is in  $\mathcal{C}^\infty$  for each  $n$  (lemma 3.15) and thus bounded on  $K$ . Let  $c_n := \sup_{x \in K} |(D^\alpha f) * g_n(x)|$ . It is sufficient to show that  $c := \sup\{c_n\} < \infty$ . Indirect proof: If not, then there exists a sequence  $(x_n) \subset K$  such that  $|D^\alpha f * g_n(x_n)| > n$ . As  $K$  compact, w.l.o.g.,<sup>21</sup>  $\exists x^* \in K : x_n \rightarrow x^*$ . The resulting situation is:

$$\forall n : |(D^\alpha f) * g_n(x_n)| > n \quad \mathbf{and} \quad |(D^\alpha f) * g_n(x_*)| \leq |D^\alpha f| * g(x_*) < \infty. \quad (3.11)$$

$(D^\alpha f) * g_n$  is differentiable and by a)

$$D^\beta(D^\alpha f * g_n) = D^\beta D^\alpha(f * g_n) = (D^{\alpha+\beta} f) * g_n \text{ for all } \beta \in \mathbb{N}^d.$$

Furthermore is

$$|(D^{\alpha+\beta} f) * g_n| \leq |D^{\alpha+\beta} f| * g_n \leq |D^{\alpha+\beta} f| * g$$

and thus

$$|D^\beta[(D^\alpha f) * g_n](x_*)| \leq |(|D^{\alpha+\beta} f| * g)(x_*)| < \infty$$

by condition (3.9). This means that all partial derivatives of  $(D^\alpha f) * g_n$  in  $x_*$  are bounded in contradiction to (3.11).  $\square$

### Application to European options

Theorem 3.16 is an important tool. It obviously covers a large set of payoffs  $g$  and models of the underlying  $f$ . For example, it is applicable in the Black-Scholes case for plain-vanilla puts and calls as well as for binary options. It is also applicable to some exponential Lévy models, e.g., the NIG model and Merton's model (introduced in chapter 5), and for high dimensional underlyings, e.g., rainbow options. In all these cases theorem 3.16 can be used to show that the value of a European option at any time  $t < T$  is  $\mathcal{C}^\infty$ -smooth in  $x$ .

### Application to Bermudan options

The theorem can also be applied in the context of Bermudan options, starting at maturity and then proceeding repeatedly backwards in time. Thus, it can be shown for each interval  $[t_i, t_{i+1}]$  between two subsequent exercise times that the hold value  $V^H$  of the option is  $\mathcal{C}^\infty$ -smooth (with respect to  $x$ ) for every  $t < T$ . In the following it is briefly discussed under which circumstances this "induction" is viable. Given the conditions of

<sup>21</sup>After switching over to a subsequence, if necessary.

theorem 3.16 for the payoff  $g$  and density  $f$ , we get a smooth hold value  $V^H = e^{-r\Delta t} f * g$  at the last exercise time before maturity. At this time the value of the Bermudan option jumps due to the early exercise opportunity:

$$V(x, t_1) = \max ( g(x), V^H(x, t_1) ).$$

Next we have to apply the theorem to the same density function  $f$  but with the new “payoff” function  $\tilde{g} := V(\cdot, t_1)$ . The question is now, whether the conditions for the application of the theorem are fulfilled. In particular:

- (i) Is  $\tilde{g}$  locally bounded and locally  $\mathcal{L}^1$ -integrable?
- (ii) Is  $f$  fast declining with respect to  $\tilde{g}$ ? I.e. do the integrals in  $(D^\alpha f) * \tilde{g}$  exist?

The answer to (i) is yes, as both  $V^H(\cdot, t_1) \in \mathcal{C}^\infty(\mathbb{R}^d)$  and  $g$  are locally bounded and locally  $\mathcal{L}^1$  integrable; and so is the maximum of both functions  $\tilde{g}$ .

The answer to (ii) is more difficult and must be handled for each market model individually. The following consideration may help to make (ii) plausible. For example, in the case of the Black-Scholes model, the density looks like  $f(x) := e^{-x^2}$ . The option value function  $\tilde{g}(x)$  grows only polynomially with  $x \rightarrow \pm\infty$ . It is then clear that for all derivatives  $f^{(\alpha)}$ , the function  $f^{(\alpha)} \cdot \tilde{g}$  is integrable (and thus  $D^\alpha f * \tilde{g}$  exists).

### 3.4.3. Further stability issues

Some potential stability problems have been neglected in the previous analysis. The reasons for them lie in several assumptions made in the construction of the spline method. Although they do not cause problems in practice, they are briefly mentioned in this section.

- (i) The local interpolation errors are estimated by a heuristic rule based on assumption 3.6. It is intuitively clear that this heuristic works well for the interpolation of globally convex or concave functions and that it works poorly for oscillatory functions. Option prices are seldom oscillatory functions. In fact, oscillations are damped under diffusion.
- (ii) The maximal number of iterations of the adaptive Romberg quadrature is usually limited to ensure termination of the quadrature routine. Such limitations can result in errors that exceed the prescribed error bound. In practice, extremely high numbers of iterations cannot be observed.
- (iii) Enforcing a minimal distance between neighboring spline nodes can result in errors that exceed the prescribed error bound. In practice, extremely low distances between the adaptively placed spline nodes cannot be observed.
- (iv) The spline does not interpolate the exact hold value function  $V^H(\cdot, t_i) \in \mathcal{C}^\infty(\mathbb{R})$  but an approximation  $W$  carrying errors. These errors do not need to be smooth; It is possible that  $W \notin \mathcal{C}(\mathbb{R})$ . This can theoretically lead to spline oscillations and thus instability, as is discussed in the following subsection.

#### Potential problem: spline oscillations

The error control technique above requires that for fill distance  $\Delta \rightarrow 0$  the spline interpolant converges to the interpolated function. This is provided by theorem 3.3 (p. 33) for the interpolation of  $V^H \in \mathcal{C}^\infty$ , but in practice the values at the nodes of the spline are given by approximations  $W(x_i)$  to the function  $V^H(x_i)$ .  $W(x_i)$  carries an additive error bounded by  $tol := \frac{(m-i)\varepsilon}{m}$  (theorem 3.9). Consequently, the function  $W$  lies within an  $V^H \pm tol$  band but is not necessarily continuous. In the worst case this can lead to increasing oscillations of the spline for decreasing minimal step size  $h_{min} \rightarrow 0$ . Figure 3.3 illustrates a spline interpolant  $W_1$  and its refinement  $W_2$ . Increasing oscillations are clearly visible.

In order to avoid oscillations it seems reasonable to prevent further refinements, if the minimal step size  $h_{min}$  falls below, e.g.,  $tol * 10$ . Fortunately this condition does not occur in practice.<sup>22</sup> There are two possible reasons:

---

<sup>22</sup>In fact, if problems due to spline oscillations occur, this strongly indicates bugs in the implementation of the quadrature or truncation procedures.

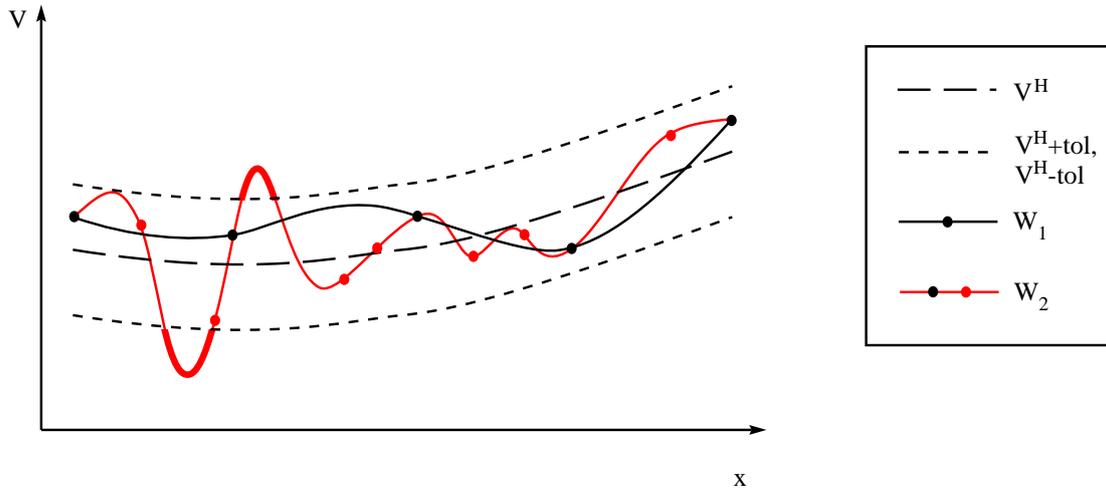


Figure 3.3.: Illustration of possible spline oscillations.

- The error actually behaves more smoothly than indicated in figure 3.3.
- The refinement loop is terminated early enough by the regular termination condition, i.e., the minimal step size that is required to fulfill the local error bound is large enough ( $h_{min} \gg tol$ ).

Although oscillations obviously do not occur in practice, this remains a potential stability problem of the spline method.

#### 3.4.4. Efficiency/computational complexity

As for most adaptive methods it is difficult to give a strict estimate of the computational complexity for the spline method. In the following, the complexity is analyzed under the assumptions that the spline interpolant in each time step has the same number of  $n$  nodes and that each Romberg quadrature uses the same number of function evaluations.  $m$  denotes the number of exercise times of the Bermudan option. The computation time  $T(m, n)$  is then linear in  $m$  by construction, as each time step requires the same number of operations:

$$T(m, n) = \mathbf{O}(mT(1, n)). \quad (3.12)$$

In each time step an adaptive discretization is constructed. The further analysis can be simplified by the (realistic) assumption that  $\log_2(n)$  refinements are necessary, i.e.,  $\log_2(n)$  cycles of the repeat-until loop in algorithm 2.1.

For each cycle the quadrature routine is called once. As the maximal number of nodes processed by the quadrature routine is  $n$ , there are  $\mathbf{O}(n \log_2(n))$  evaluations of the integral in algorithm 2.2 for each time step. The proposed quadrature rule is a compound

Romberg quadrature rule over  $n$  subintervals. Consequently, the computational complexity is that of  $n$  Romberg quadratures. To simplify the analysis, the assumption is made that every Romberg call has the same constant costs  $c$ .<sup>23</sup> Consequently, the total costs for quadratures are

$$T_{quad}(1, n) = \mathbf{O}(cn^2 \log_2 n) = \mathbf{O}(n^2 \log_2 n).$$

Additionally,  $\log_2(n)$  splines are constructed in each time step. Each interpolant requires the solution of a tridiagonal  $n \times n$ -system (linear complexity in  $n$ ). This leads to the following costs for interpolations

$$T_{interpol}(1, n) = \mathbf{O}(n \log_2 n),$$

which are dominated by the quadrature costs. In total, the computation time for each time step is

$$T(1, n) = \mathbf{O}(n^2 \log_2 n)$$

and thus, following (3.12), for all  $m$  time steps

$$T(m, n) = \mathbf{O}(mn^2 \log_2 n). \tag{3.13}$$

As this result alone yields no information on the *efficiency*<sup>24</sup> of the spline method, it is useful to relate (3.13) to the convergence properties. Under the (worst-case) assumption of equidistant spline nodes, theorem 3.3 (p. 33) implies convergence  $\mathbf{O}(n^{-4})$  of the spline interpolation. Stated conversely, a given error bound  $\varepsilon$  requires  $n = \mathbf{O}(\varepsilon^{-1/4})$  nodes and by (3.13) a time complexity of

$$T_\varepsilon(m) = \mathbf{O}(m\varepsilon^{-1/2} \log_2(\varepsilon^{-1/4})). \tag{3.14}$$

This efficiency results for Bermudan options. For American options section 2.4 indicates linear convergence in  $m$ , i.e.,  $m \approx \varepsilon^{-1}$ . Thus, the efficiency result for American options is:

$$T_\varepsilon = \mathbf{O}(\varepsilon^{-3/2} \log_2(\varepsilon^{-1/4})). \tag{3.15}$$

The result (3.14) for Bermudan options is competitive. In practice one can expect even better results because of the adaptivity of the spline method. The efficiency (3.15) for American options suffers from the linear convergence order of the time discretization.

---

<sup>23</sup>This assumption is rather realistic. Empirical tests show that only a few number of Romberg iterations is necessary. A typical number of function evaluations in each Romberg call is 16. Each function evaluation is essentially a spline evaluation with complexity  $\mathbf{O}(1)$ .

<sup>24</sup>Here, “efficiency” means the relation between accuracy and computation time.

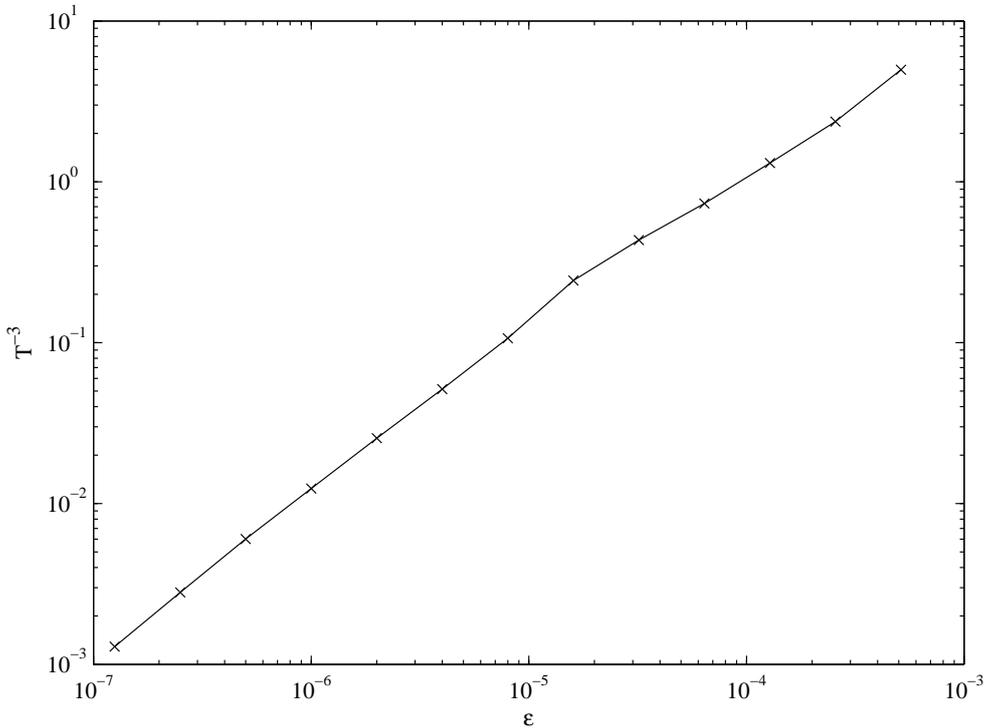


Figure 3.4.: Practical efficiency measurements for the spline method.

Figure 3.4 shows practical results<sup>25</sup> that indicate a computational efficiency of the spline method for Bermudan options of  $T_\varepsilon(m) = \mathbf{O}(m\varepsilon^{-1/3})$ . To illustrate the result clearly, the figure shows  $T^{-3}$  against  $\varepsilon$  in a log/log plot, where  $T$  is the measured computation time in seconds and  $\varepsilon$  is the error bound of the spline method. This empirical result is one order better than the theoretical result (3.14) which was derived by neglecting the adaptivity of the spline method.

<sup>25</sup>To obtain these results an efficient implementation in C++ was used on an Intel T2600 CPU with 2.16 GHz. The option parameters are the same as for the example in figure 3.2 (p. 42). The measured CPU times are between 0.5s and 9s.

## 3.5. Generalization to multi-asset options

Up to now, the spline method can only be used for single-asset options. This section discusses possible generalizations to multi-asset options. An efficient generalization turns out to be difficult, as efficient multivariate interpolation and quadrature are demanding problems. Although the extension of the spline method to multi-asset options in this section works, the resulting method is not very efficient and the RBF method introduced in chapter 4 is far superior. Thus, this section is kept short.

Section 3.5.1 discusses suitable multivariate interpolation techniques, section 3.5.2 suitable quadrature techniques, and section 3.5.3 explains the adaptive refinement strategy for the multivariate case, which is slightly more complicated than the bisection strategy from section 3.3.1.

### 3.5.1. A suitable multivariate interpolation technique

There are several possible generalizations of one-dimensional splines to higher dimensions. Two examples are briefly introduced in the following.

#### Tensor product splines

This is the most natural generalization of splines to higher dimensions. Two one-dimensional spline discretizations  $x_1 < \dots < x_{n_1}$  and  $y_1 < \dots < y_{n_2}$  define a rectangular grid on  $A := [x_1, x_{n_1}] \times [y_1, y_{n_2}]$ . The *bivariate tensor product spline* of degree  $d_1$  in  $x$  and  $d_2$  in  $y$  is a function  $s(x, y)$  that

- coincides on each subrectangle  $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$  of the grid with a polynomial of degree  $d_1$  in  $x$  and  $d_2$  in  $y$ , and
- is continuous on  $A$ , and whose
- partial derivatives  $\partial^{i+j}s/\partial x^i \partial y^j \in \mathcal{C}(A)$  are continuous for  $i \leq d_1, j \leq d_2$ .

Obviously, tensor product splines are restricted to fairly simple grids, like smooth transformations of axis-parallel grids. This restriction disqualifies tensor product splines for adaptive interpolations.

#### Powell-Sabin splines

This generalization of univariate splines is defined over a given triangulation  $\Delta$ . It was introduced in [PS77]. For a certain refinement  $\Delta^*$  that subdivides each triangle of the triangulation  $\Delta$  into six special triangles it is possible to prove existence and uniqueness

for the interpolation problem

$$\begin{aligned} s(x_i) &= y_i, \\ \frac{\partial s}{\partial x}(x_i) &= u_i, \text{ and} \\ \frac{\partial s}{\partial x}(x_i) &= v_i \quad (\text{for all nodes } x_i \text{ in the triangulation } \Delta^*) \end{aligned}$$

in the function space

$$\mathcal{S} = \{ s \in \mathcal{C}(\mathbb{R}^2, \mathbb{R}) : \forall \text{ triangles } \rho \in \Delta^* \exists p \in \Pi_2 : s|_{\rho} = p \},$$

where  $\Pi_2$  denotes the space of bivariate polynomials with total degree less or equal two. This approach is not useful for a generalization of the valuation method, as the risk-neutral valuation approach does not provide any information on the partial derivatives  $\partial V^H / \partial x$  and  $\partial V^H / \partial y$  of the hold value.<sup>26</sup>

### Thin plate splines

The two spline generalizations above indicate the problem of all such generalizations. They are restricted to structured meshes or require information about derivatives. Therefore, an alternative interpolation approach is needed. In the following an interpolation with radial basis functions is briefly introduced, which gives up the restriction that the interpolant is piecewise polynomial. The so-called “thin plate splines”<sup>27</sup> are interpolants of the form

$$s(x) := \sum_{j=1}^n \lambda_j \phi(x - x_j), \tag{3.16}$$

with interpolation coefficients  $\lambda_j \in \mathbb{R}$  and  $\phi(x) := \|x\|_2^2 \cdot \log \|x\|_2$ . This is a special case of an interpolation with radial basis functions (RBF). Interpolation with radial basis functions is introduced in chapter 4. The interpolation condition

$$s(x_j) = f(x_j) =: y_j \quad (\text{for } 1 \leq j \leq n)$$

is equivalent to the linear system  $B\lambda = y$  with an interpolation matrix

$$B = (b_{ij}) \in \mathcal{M}^{n \times n} \text{ defined by } b_{ij} := \phi(\|x_i - x_j\|).$$

Existence and uniqueness of the solution as well as the condition of the interpolation matrix and the interpolation error are discussed, e.g., in [Wen05]. As the efficiency of the multivariate spline method is not analyzed in the present work, these results are not introduced. In principle, thin plate splines represent a (with reservations) suitable interpolation technique for the present purpose.

<sup>26</sup>Numerical differentiation is prohibitively inefficient for this purpose.

<sup>27</sup>The name of this interpolation method refers to a physical analogy of bending a thin plate of metal.

**Remark 3.18** (Costs). *The construction of the interpolant (3.16) for a given set of nodes  $x_1, \dots, x_n \in \mathbb{R}^d$  has computational complexity  $\mathbf{O}(n^3)$  due to the linear system  $B\lambda = y$  and each evaluation of this interpolant has complexity  $\mathbf{O}(n)$ . This is expensive compared to cubic splines, which require only  $\mathbf{O}(n)$  for their construction and  $\mathbf{O}(1)$  for each evaluation (see section 3.1).*

### 3.5.2. Multivariate numerical quadrature

Several approaches to construct multivariate numerical quadrature rules are discussed in, e.g., [DR75]. A straightforward generalization of univariate quadrature methods is the *product rule*. It is constructed as follows.

**Definition 3.19** (Bivariate product rule). *Given two univariate rules on the nodes  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}$  and  $Y = \{y_1, \dots, y_m\} \subset \mathbb{R}$  with corresponding weights  $w_1, \dots, w_n$  and  $v_1, \dots, v_m$ , the bivariate product rule on the nodes  $X \times Y$  is defined by the weights  $w_{ij} := w_i v_j$  for all  $(x_i, y_j) \in X \times Y$ . The integral over a function  $f : [x_1, x_n] \times [y_1, y_m] \rightarrow \mathbb{R}$  is then approximated by*

$$R(f) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} f(x_i, y_j) \approx \int_{x_1}^{x_n} \int_{y_1}^{y_m} f(x, y) \, dy \, dx.$$

**Theorem 3.20** (Exactness of bivariate product rule). *If the univariate rule  $R$  integrates  $f(x)$  exactly over  $A \subset \mathbb{R}$  and the univariate rule  $S$  integrates  $g(y)$  exactly over  $B \subset \mathbb{R}$ , then  $R \times S$  will integrate  $h(x, y) := f(x)g(y)$  exactly over  $A \times B$ .*

For the implementation of the multivariate spline method, a quadrature rule is used. An error estimate for product rules is given in [Hab70]. It is not required here, as the efficiency of the multivariate spline method is not discussed in detail.

#### Requirement of adaptive quadrature

Error bounds for the quadrature error can only be given for sufficiently smooth integrands. By construction, the integrands  $I := \max(g, V^H)$  in the spline method are *not* smooth. For highly accurate quadrature rules it is thus necessary to decompose the domain of integration in such a way that  $I$  is sufficiently smooth on each subdomain. While this decomposition was trivial in the univariate case, it is a difficult problem in the multivariate case.

For the implementation of the multivariate spline method here, the domain of integration is *not* adaptively decomposed. Thus, strict error bounds cannot be given. In this sense, the development of the multivariate spline method in the present work must be regarded provisional.

**Remark 3.21** (Possible extension: sparse grid quadrature). *Quadrature rules on hierarchical “sparse” grids have been proposed by Smolyak in [Smo63]. Error bounds for integrands in the space  $C^k(\mathbb{R}^d)$  are given in [NR97]. However, they are exponentially in the dimension  $d$ . The exponential dependence of the error bound on the dimension can be reduced by restricting the class of integrands to Sobolev spaces  $W_\infty^k(\mathbb{R}^d)$  as noted in [GG98]. The use of sparse grid quadrature can become interesting for applications to higher dimensions.*

### 3.5.3. Adaptive refinement

In the multivariate case, an adaptive refinement is even more important than in the univariate case. Although the interpolation with thin plate splines is mesh-free, the following refinement strategy is based on triangulations. As in the one-dimensional method, it is assumed that the improper integrals can be approximated by truncation to a finite (rectangular) region. The initial triangulation is then obtained by a subdivision of the rectangle into two triangles.

The refinement strategy is a straightforward generalization of the bisection strategy in the univariate case (section 3.3.1), in which a “test point” between two successive spline nodes is used to estimate the interpolation error. In the bivariate case the midpoint of the longest edge of the triangle is chosen as test point. If the estimated local interpolation error at the test point is too large, the corresponding triangle is divided into two triangles at the line between the test point and its opposite vertex. This choice guarantees that the resulting triangles cannot degenerate. Another advantage of this choice is that the function value at the test point can be reused for the refined interpolant.

The refinement procedure can lead to “hanging vertices” as illustrated in figure 3.5, step (5)  $\rightarrow$  (6). As the thin plate spline interpolation does not use the vertex information of the triangulation, this does not cause discontinuities in the interpolant. Nevertheless, avoiding hanging vertices leads to a more uniform distribution of interpolation nodes. Therefore, the corresponding neighboring triangles are subdivided, if a hanging node is detected (as illustrated by dotted lines in figure 3.5). A criterion for hanging vertices can be based on the “depth” of each triangle in a binary tree representation of the triangulation. It is easy to see that hanging nodes occur, if and only if the depths of neighboring triangles differ by more than one. (For example, in figure 3.5, step (5)  $\rightarrow$  (6), a triangle with depth 4 is divided into two triangles with depth 5 neighboring a triangle with depth 3.)

#### Illustration

Figure 3.6 shows an adaptively refined triangulation for a two-asset put option. The value  $V$  denotes the option value at a time  $t < T$  near maturity. Figure 3.7 shows the same results after graphical post-processing. For this example, the number of triangles

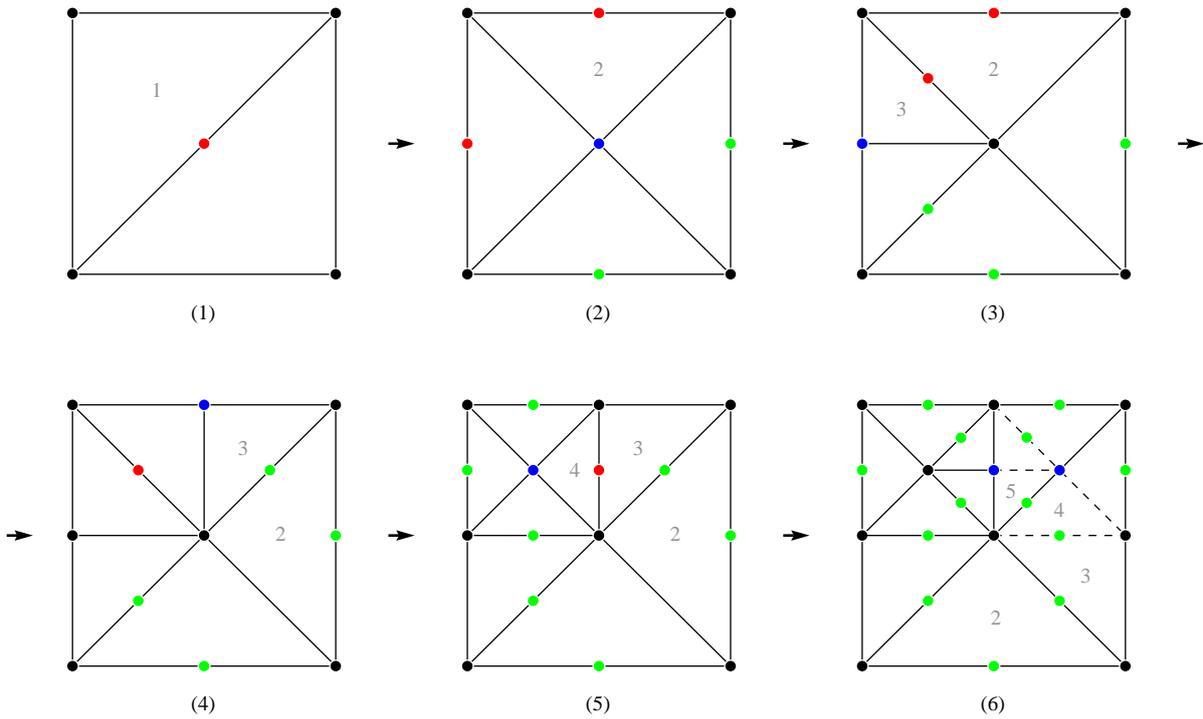


Figure 3.5.: Illustration of the adaptive refinement procedure. The test points at which the interpolation error is still too large are marked red. The test points at which the desired accuracy has been reached are marked green. In each step, the triangulation is refined at one of the red nodes. An additional refinement takes place if hanging nodes have been detected, e.g., at the dashed lines in step (6). The numbers in the triangles indicate the depth of the triangles in the corresponding binary tree representation.

generated by the spline method has been reduced artificially to simplify printing. (Some printers exhibit extreme delays when printing pages with more than 30000 triangles.)

**Remark 3.22** (Implementation note). *An efficient implementation of triangulations requires a representation by tree data structures. Such implementations require pointer arithmetic. This is one reason, why the spline method should be implemented using a programming language that is compiled to native code.*

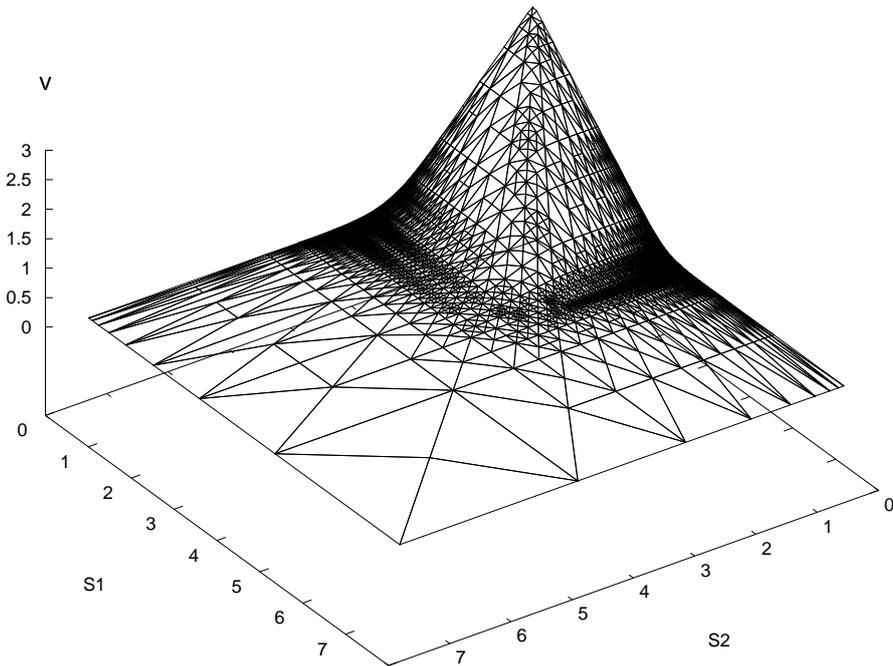


Figure 3.6.: Adaptive triangulation for a Bermudan “two-color better-of put” with payoff  $g(S_1, S_2) = (K - \max(c_1 S_1, c_2 S_2))^+$ .

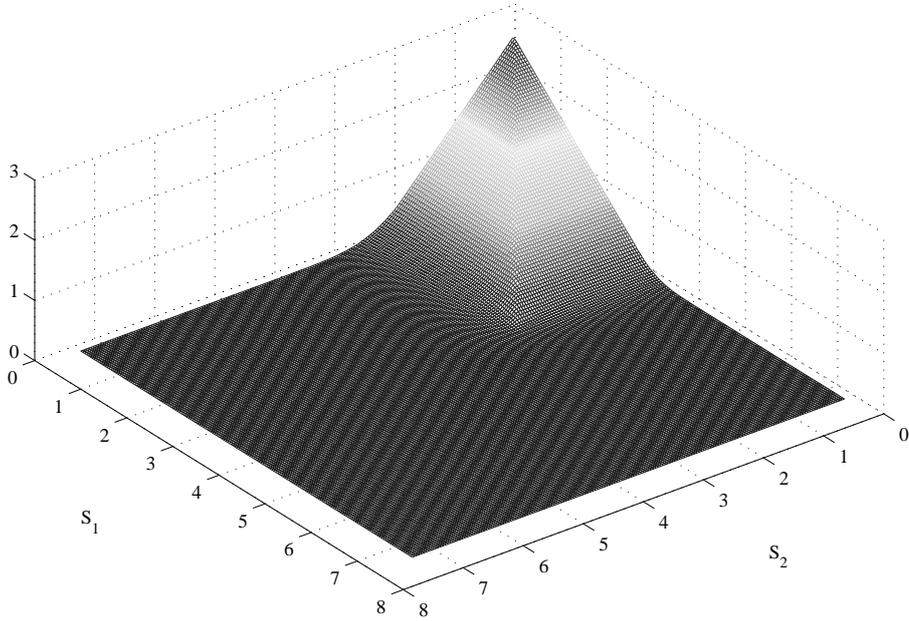


Figure 3.7.: The same results graphically post-processed with MATLAB.

### 3.5.4. Numerical properties

The analysis of numerical properties can be carried out analogously to the one-dimensional case, as the important smoothness result from theorem 3.16 is not limited to the univariate case. Further requirements for a multivariate convergence theorem analogous to theorem 3.9 would be estimates of the quadrature and interpolation error.

Nevertheless, the multivariate convergence analysis is not carried out here, as the used interpolation and quadrature methods seem provisional: The computational complexity of thin plate spline interpolation is by a factor  $n$  higher than the cubic spline interpolation. Additionally, the number of nodes  $n$  itself vastly increases for the additional dimension. A two-dimensional mesh with the same fill distance as a one-dimensional mesh with  $n$  equidistant nodes requires  $\mathbf{O}(n^2)$  nodes. An alternative interpolation technique could be based on radial basis functions with compact support as analyzed in [Wen96], but an efficient implementation goes beyond the scope of this work.<sup>28</sup>

## 3.6. Conclusion

A first method following the reduction principle in chapter 2 has been derived – the “spline” method. By construction, this method is capable of pricing Bermudan and American options with arbitrary payoffs and fairly arbitrary models for the underlying (in the sense of section 2.1). Its adaptivity makes this method highly suitable for options with non-smooth payoffs or other sources of discontinuities (e.g., barriers). Another advantage is that both building blocks of the univariate method, namely, splines and Romberg quadrature, are well understood numerical tools.

The spline method has also some drawbacks. It does not generalize naturally to higher dimensions as seen in section 3.5. Of practical relevance is that an efficient implementation is only possible using compiled languages like C++ or Fortran and not with interpreter based environments like MATLAB. The reason is the interaction of interpolation and quadrature: The quadrature method requires  $\mathcal{C}^\infty$ -smooth integrands and splines are  $\mathcal{C}^\infty$ -smooth only between neighboring nodes. That requires all integrals to be split up along the spline nodes which in turn leads to a high number of elementary operations. Interpreter based languages are efficient only for operations of large granularity (such as operations on vectors or matrices) and are inefficient for operations of small granularity (such as scalar operations like function evaluations).

---

<sup>28</sup>An efficient implementation requires efficient data structures for the representation of neighborhood between nodes and fast search algorithms. These aspects are discussed e.g. in [Wen05], §14.

## 4. The RBF method

The generalization of the spline method in section 3.5 indicates that the combination of interpolation and quadrature in several dimensions can be computationally expensive. This chapter proposes a method that exploits the structure of the interpolant for an adapted quadrature method. There are two important gains. First, the interpolant does not need to be evaluated at random points anymore. Second, the quadrature problems are reduced to globally smooth integrands.

The short discussion of multivariate interpolation methods in section 3.5 suggests an interpolation by radial basis functions (RBF). In contrast to splines, this interpolation technique generalizes naturally to higher dimensions. Furthermore, it is known to offer spectral convergence properties for certain integrands and basis functions.

The pricing method proposed in this chapter is based on radial basis functions and thus called *RBF method*. A first motivation is given in section 4.1. Some aspects of the interpolation by radial basis functions are discussed in section 4.2. As the following derivation will show, the choice of numerical quadrature routines for the RBF method is not as critical as for the spline method. Section 4.3 discusses possible approaches. The numerical properties of the RBF pricing method are analyzed in section 4.4. Practical validations and examples are given in chapter 6.

### 4.1. Motivation

The main idea of the method proposed in this chapter is to use RBF interpolation to solve the recursive integrals in the option price representation of the reduction principle for Bermudan options (lemma 2.12, p. 14).

In the following one time step  $t_{k+1} \rightarrow t_k$  between two subsequent exercise times of a Bermudan option is considered. The time index in this derivation is changed to  $k$ . This agreement releases the index  $i$ , which can now be used for the space discretization, i.e., for node numbering.

Let  $X := \{x_1, \dots, x_n\} \subset \mathbb{R}^d$  denote a fixed set of nodes in the asset price space. Then, an interpolant  $s$  of the option price function  $V(\cdot, t_{k+1})$  can be constructed by a function

$$s(x) = s_{V,X}(x) := \sum_{j=1}^n \lambda_j^{(k+1)} \phi_j(x - x_j),$$

with *interpolation coefficients*  $\lambda_j^{(k+1)} \in \mathbb{R}$  (for  $j = 1, \dots, n$ ) and *radial basis functions*

$\phi_j : \mathbb{R}^d \rightarrow \mathbb{R}$ , which are specified later (in section 4.2). The coefficients  $\lambda_j^{(k+1)}$  are determined by the interpolation condition

$$s(x_i) = \sum_{j=1}^n \lambda_j^{(k+1)} \underbrace{\phi_j(x_i - x_j)}_{=: b_{ij}} = V(x_i, t_{k+1}) \quad (\text{for } i = 1, \dots, n). \quad (4.1)$$

This linear system suggests abbreviations

$$\begin{aligned} B &:= (b_{ij})_{i,j=1\dots n}, \\ \lambda^{(k+1)} &:= (\lambda_1^{(k+1)}, \dots, \lambda_n^{(k+1)})^{tr}, \text{ and} \\ v_{k+1} &:= (V(x_1, t_{k+1}), \dots, V(x_n, t_{k+1}))^{tr}. \end{aligned}$$

The interpolation condition (4.1) can now be written as

$$B\lambda^{(k+1)} = v_{k+1}. \quad (4.2)$$

As  $B$  only depends on the basis functions  $\phi_j$  (and the location of the nodes  $x_1, \dots, x_n$ , but this is considered fixed here), it will be referred to as the **basis matrix** (in the literature on RBF interpolation, this matrix is also called interpolation matrix). By lemma 2.12 (p. 14), the hold value at  $t_k$  is given as discounted expectation value. Abbreviating the discount factor by  $a := e^{-r(t_{k+1}-t_k)}$ , we get

$$\begin{aligned} \frac{1}{a}V^H(x, t_k) &= \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x}(\xi)V(\xi, t_{k+1}) \, d\xi \\ &\approx \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x}(\xi)s(\xi) \, d\xi \end{aligned} \quad (4.3)$$

$$\begin{aligned} &= \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x}(\xi) \left[ \sum_{j=1}^n \lambda_j^{(k+1)} \phi_j(\xi - x_j) \right] \, d\xi \\ &= \sum_{j=1}^n \lambda_j^{(k+1)} \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x}(\xi) \phi_j(\xi - x_j) \, d\xi \end{aligned} \quad (4.4)$$

$$\Rightarrow \frac{1}{a}V^H(x_i, t_k) = \sum_{j=1}^n \lambda_j^{(k+1)} \underbrace{\int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x_i}(\xi) \phi_j(\xi - x_j) \, d\xi}_{=: m_{ij}} \quad (i = 1, \dots, n).$$

Of course, the decisive step “ $\approx$ ” will be subject to an error analysis later (in section 4.4). With

$$M := (m_{ij})_{i,j=1\dots n} \quad \text{and} \quad v_k^H := (V^H(x_1, t_k), \dots, V^H(x_n, t_k))^{tr}$$

the above linear system can be written as

$$v_k^H = aM\lambda^{(k+1)}. \quad (4.5)$$

As  $M$  depends on the model of the underlying (represented by its conditional probability density  $f$ ), it is referred to as the *model matrix*. By equations (4.2) and (4.5) the relation between the hold values  $V^H(x_i, t_k)$  and option values  $V(x_i, t_{k+1})$  is

$$v_k^H = aMB^{-1}v_{k+1}. \quad (4.6)$$

At time  $t_k$ , the option can be exercised and its value jumps to

$$V(x, t_k) = \max(g(x), V^H(x)),$$

where  $g$  denotes the option's payoff function. The discrete version of this equation is

$$v_k = \max(g, v_k^H),$$

where  $g$  is to be understood as vector  $g = (g(x_1), \dots, g(x_n))^{tr}$  and the maximum is to be understood componentwise. Inserting (4.6) yields

$$v_k = \max(g, aMB^{-1}v_{k+1}). \quad (4.7)$$

By this procedure, the value of the Bermudan option can be computed iteratively backwards in time, starting with the payoff  $v_m = g$ . Of course, the approximation error of “ $\approx$ ” in (4.3) and the stability of this method have still to be analyzed.

### Foreseeable advantages and difficulties

At this point, some key advantages of the RBF method can be foreseen:

- + All integration problems are relocated to the model matrix, which remains constant as long as the location of nodes remains unchanged. For fixed nodes, the model matrix needs only to be computed once.
- + The integrands of the integration problems have the form  $f \cdot \phi$  and are likely to be “good behaving” functions that facilitate the (numerical) quadratures during the setup of the model matrix.
- + The interpolant does not need to be evaluated at random points as in the spline method. Thus, the  $\mathbf{O}(n)$ -costs for evaluations are irrelevant to the computational costs of the RBF pricing method.
- + Concerning the model for the underlying, the RBF method is as flexible as the spline method. The density function  $f$  describing the conditional probability densities enters the setup of the model matrix.

Some difficulties that can be foreseen at this point are:

- The estimation of the approximation error “ $\approx$ ” can be involved.
- The interpolated functions  $V(\cdot, t_k)$  are not globally smooth, as is known from section 3.5. This can affect the approximation error.
- It is known for RBF interpolations that the condition of  $B$  can be problematic.

## 4.2. Interpolation by radial basis functions

A recent overview of radial basis function methods can be found in [Buh03]. For the RBF based pricing proposed in this work, the interpolation aspect is most important. Generally, a *radial basis function interpolant* for a function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  has the form

$$s_{f,X}(x) := \sum_{j=1}^n \lambda_j \phi_j(x - x_j), \quad (4.8)$$

where  $X := \{x_1, \dots, x_n\}$  denotes the set of *centers*,  $\lambda_1, \dots, \lambda_n \in \mathbb{R}$  the interpolation coefficients, and  $\phi_j : \mathbb{R}^d \rightarrow \mathbb{R}$  are given functions, called radial basis functions. In the following we consider only the case of a single basis function for all nodes  $x_j$ , i.e.,  $\phi_j = \phi$ . Usually,  $\phi$  is constructed radially symmetric:  $\phi(x) := g(\|x\|_2)$  with a continuous function  $g : \mathbb{R}^+ \rightarrow \mathbb{R}$ . Typical choices of  $g$  are the following:

$g(r)$	parameters	interpolation type
$r^2 \log r$		thin plate splines
$\sqrt{r^2 + c^2}$	$(c > 0)$	multiquadrics
$e^{-\alpha r^2}$	$(\alpha > 0)$	Gaussian
$[(1 - r)^+]^4(4r + 1)$		compactly supported RBF (example)

For the purpose of RBF based option pricing, the Gaussian basis function is preferable, as this function decays to zero for  $r \rightarrow \infty$  and, furthermore, this choice simplifies the integrals arising in the model matrix. The use of compactly supported RBF could be an interesting alternative but is not considered in the present work. For thin plate splines and multiquadrics, the step “ $\approx$ ” in (4.3) does not work. Thus, the following discussion focuses on the properties of Gaussian interpolation.

As mentioned in the previous section,<sup>1</sup> the interpolation conditions  $s_{f,X}(x_i) = f(x_i)$  ( $i = 1, \dots, n$ ) give rise to a linear system

$$\vec{f} = B\lambda \quad (4.9)$$

where  $\vec{f} := (f(x_1), \dots, f(x_n))^{tr}$  contains function values of the interpolated function  $f$ ,  $\lambda := (\lambda_1, \dots, \lambda_n)^{tr}$  denotes the coefficient vector, and  $B$  the basis matrix defined as follows.

**Definition 4.1** (Basis matrix/interpolation matrix). *The matrix*

$$B := (b_{ij})_{i,j=1\dots n} \text{ with } b_{ij} := \phi(x_i - x_j)$$

*is called basis matrix or interpolation matrix.*

**Remark 4.2** (Shape parameter  $\alpha$ ). *The Gaussian basis function uses a shape parameter  $\alpha > 0$ . This parameter is important for the condition of  $B$ . Its choice is discussed in section 4.2.3.*

<sup>1</sup>See equation (4.2), p. 66.

### 4.2.1. Unique existence of the interpolant

This section briefly discusses the unique existence of the interpolation, i.e., the question, whether  $B$  is nonsingular. Theorem 4.4 can be found in [Buh03] (prop. 2.1, p. 13). It was shown first by Schoenberg.

**Definition 4.3** (Complete monotonicity). *A function  $g \in C^\infty(\mathbb{R}^+)$  is completely monotonic if and only if for all  $l \in \mathbb{N}$  and for all  $t \geq 0$ :  $(-1)^l g^{(l)}(t) \geq 0$ .*

In particular, the exponential function  $g(t) = e^{-\alpha t}$  is completely monotonic for  $\alpha \geq 0$ .

**Theorem 4.4** (Schoenberg). *Let  $g : \mathbb{R}^+ \rightarrow \mathbb{R}$  be a continuous, completely monotonic function. Then, for all finite  $X \subset \mathbb{R}^d$  of distinct points and all  $n$ , the matrix  $B$  from definition 4.1 is positive definite for  $\phi(x) = g(\|x\|_2^2)$ , unless  $g$  is constant. In particular, the matrix  $B$  is nonsingular.*

*Proof.* Can be found, e.g., in [Buh03]. □

Theorem 4.4 immediately implies that the Gaussian basis function  $\phi(x) = e^{-\alpha \|x\|_2^2}$  leads to an invertible interpolation matrix for any  $\alpha > 0$ . Thus, a unique interpolant exists for any choice of the centers.

### 4.2.2. Error bounds/convergence

To establish error estimates for any interpolation method, it is necessary to start with assumptions on the interpolated functions. In the case of radial basis function interpolation, the interpolated functions are assumed to come from the so-called *native space*  $\mathcal{F}_\phi$  corresponding to the kernel  $\phi(\cdot - \cdot)$ . A definition of this space would introduce unnecessary notations here.<sup>2</sup> Instead, we directly give a characterization of the native space.

**Definition 4.5** (Positive definite functions). *A function  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  is positive definite, if for any set of pairwise distinct centers  $x_1, \dots, x_n \in \mathbb{R}^d$  the matrix  $B := (b_{ij})_{i,j=1\dots n}$  with  $b_{ij} := \phi(x_i - x_j)$  is a positive definite matrix.*

Schoenberg's theorem 4.4 implies that the Gaussian basis function is positive definite. The native space of positive definite functions can be characterized as follows:<sup>3</sup>

**Theorem 4.6** (Characterization of the native space). *Let  $\phi \in C(\mathbb{R}^d) \cap \mathcal{L}^1(\mathbb{R}^d)$  be a real-valued positive definite function. Define*

$$\mathcal{G} := \left\{ f \in \mathcal{L}^2(\mathbb{R}^d) \cap C(\mathbb{R}^d) \mid \widehat{f} / \sqrt{\widehat{\phi}} \in \mathcal{L}_2(\mathbb{R}^d) \right\},$$

<sup>2</sup>The construction of native spaces is explained in [Wen05], §10.

<sup>3</sup>See [Wen05], th. 10.12, p. 139.

where  $\widehat{f}$  and  $\widehat{\phi}$  denote the Fourier transforms of  $f$  and  $\phi$ , respectively. Let this space be equipped with the bilinear form

$$(f, g)_{\mathcal{G}} := (2\pi)^{-d/2} \left( \widehat{f}/\sqrt{\widehat{\phi}}, \widehat{g}/\sqrt{\widehat{\phi}} \right)_{\mathcal{L}^2(\mathbb{R}^d)} = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \frac{\widehat{f}(\omega)\overline{\widehat{g}(\omega)}}{\widehat{\phi}(\omega)} d\omega.$$

Then  $\mathcal{G}$  is a real Hilbert space with inner product  $(\cdot, \cdot)_{\mathcal{G}}$  and reproducing kernel  $\phi(\cdot - \cdot)$ . Hence  $\mathcal{G}$  is the native space of  $\phi$  on  $\mathbb{R}^d$ , and both inner products coincide.

*Proof.* See [Wen05], p. 139. □

Characterization 4.6 shows that the native space for Gaussian basis functions is rather small. The square integrability of the quotient  $\widehat{f}/\widehat{\Phi}$  requires that the Fourier transform of  $f$  decays faster than the Fourier transform of the Gaussian, which is a Gaussian itself, as the Gaussian function is self-dual with respect to the Fourier transform. Therefore, convergence results that are restricted to functions from the native space  $\mathcal{F}_{\phi}$  are not useful for the interpolation of option price functions. In particular the impressive spectral convergence result for Gaussian basis functions is of little use here.<sup>4</sup>

### Convergence result for Sobolev spaces

Schaback introduced the idea to consider convergence results for the interpolation of functions from “foreign” native spaces. In [Sch96], an interpolation of functions from a native space  $\mathcal{F}_0$  of a radial basis function  $\phi_0$  is considered, where a *different* radial basis function  $\phi_1$  is used for the interpolation.<sup>5</sup> It is found that “the  $\phi_1$ -interpolants seem to have more or less the same error on the larger space  $\mathcal{F}_0$  as the optimal  $\phi_0$ -interpolants”.<sup>6</sup> Using this approach, error bounds can be derived that are applicable to the RBF pricing method. In the following the situation  $\mathcal{F}_0 \supseteq \mathcal{F}_1$  is considered, where  $\mathcal{F}_1$  is the native space of the Gaussian basis function

$$\phi_1(x) := e^{-\alpha\|x\|_2^2}$$

and  $\mathcal{F}_0$  is a Sobolev space. As can be shown with theorem 4.6, a radial basis function  $\phi_0$ , whose native space is the Sobolev space  $W_2^k(\mathbb{R}^d)$  is the following:

$$\phi_0(x) := c \cdot \|x\|_2^{k-d/2} K_{k-d/2}(\pi\|x\|_2),$$

where  $c$  is a constant, and  $K_{\nu}$  denotes the modified Bessel function of the second kind. For this special case, the following error bound holds.

<sup>4</sup>The spectral convergence result is established in [MN92].

<sup>5</sup>Similar results can be found in [Yoo01].

<sup>6</sup>See [Sch96].

**Theorem 4.7** (Error bound for Gaussian interpolation in Sobolev spaces). *Let  $s_{f,X}$  be the Gaussian basis function interpolant of the form (4.8) for  $f \in W_2^m(\mathbb{R}^d)$ . Then the error is bounded as follows:*

$$\|f - s_{f,X}\|_{\infty,\Omega} \leq c_0 \cdot \|f\|_{m,2} \cdot h^{m-d/2}$$

where  $c_0 > 0$  is constant and  $h$  denotes a scaling parameter that controls the approximation quality, namely an upper bound to the local fill distances:

$$h_{\rho,X}(x) := \sup_{\{y : \|x-y\|_2 \leq \rho\}} \min_{x_j \in X} \|y - x_j\|_2 \leq h.$$

*Proof.* See [Sch96], theorem 4.5 and §5. □

This result can be applied to usual payoffs by an appropriate restriction to compact supports. E.g. for the plain-vanilla put payoff this theorem guarantees a convergence  $\mathbf{O}(\sqrt{h})$  in the supremum norm. However, practical results indicate a better convergence behavior.

### Practical tests of the convergence behavior

In the following practical tests of the convergence behavior for several typical payoffs are performed. The payoffs are interpolated with Gaussian basis functions on an log-equidistant  $S$ -discretization. The interpolation error is estimated empirically. Concurrently, a European option value for a small time step is approximated via equation (4.6). The exact specification of the model and option parameters is not important here, as the results are qualitative.

Figure 4.1 illustrates the convergence of the interpolant (dashed line) for a plain-vanilla payoff  $g$  (dotted line). The approximation value  $V_n$  of the resulting European option is plotted as a solid line. Apparently, the local interpolation error is maximal near the kink. Table 4.1 shows the interpolation errors (left column). In case of a plain-vanilla put, they decay linearly in the fill distance  $h = \mathbf{O}(1/n)$ . Further, it can be seen that the approximations of the option values (right column) converge quadratically in  $h$ . The error of the approximation  $V_n$  is estimated at the point  $S = 1$ . The errors at the border of the computational domain are due to truncation errors, which are not discussed here.

Figure 4.2 shows the convergence behavior for a binary option as a representative for discontinuous payoffs. The interpolant of the payoff is plotted as a dashed line, the payoff itself as a dotted line. Again, the resulting approximation  $V_n$  of the European option value for a small time step is plotted as a solid line. It is clearly visible that the interpolation error in terms of the supremum norm remains constant, independent of the fill distance. Table 4.1 contains the interpolation errors for the binary payoff and the approximation errors  $|V - V_n|$  at  $S = 1.1$ . As for the plain-vanilla option, the approximations  $V_n$  converge for  $n \rightarrow \infty$  quadratically in the fill distance  $h = \mathbf{O}(1/n)$ .

Similar results for the interpolation of continuous and discontinuous payoffs are found for higher dimensional options. (See figures 4.3 and 4.4 for an illustration.)

Figure 4.1.: Convergence of the RBF interpolant for plain-vanilla payoff: The plots show the payoff and option interpolants for several discretizations.

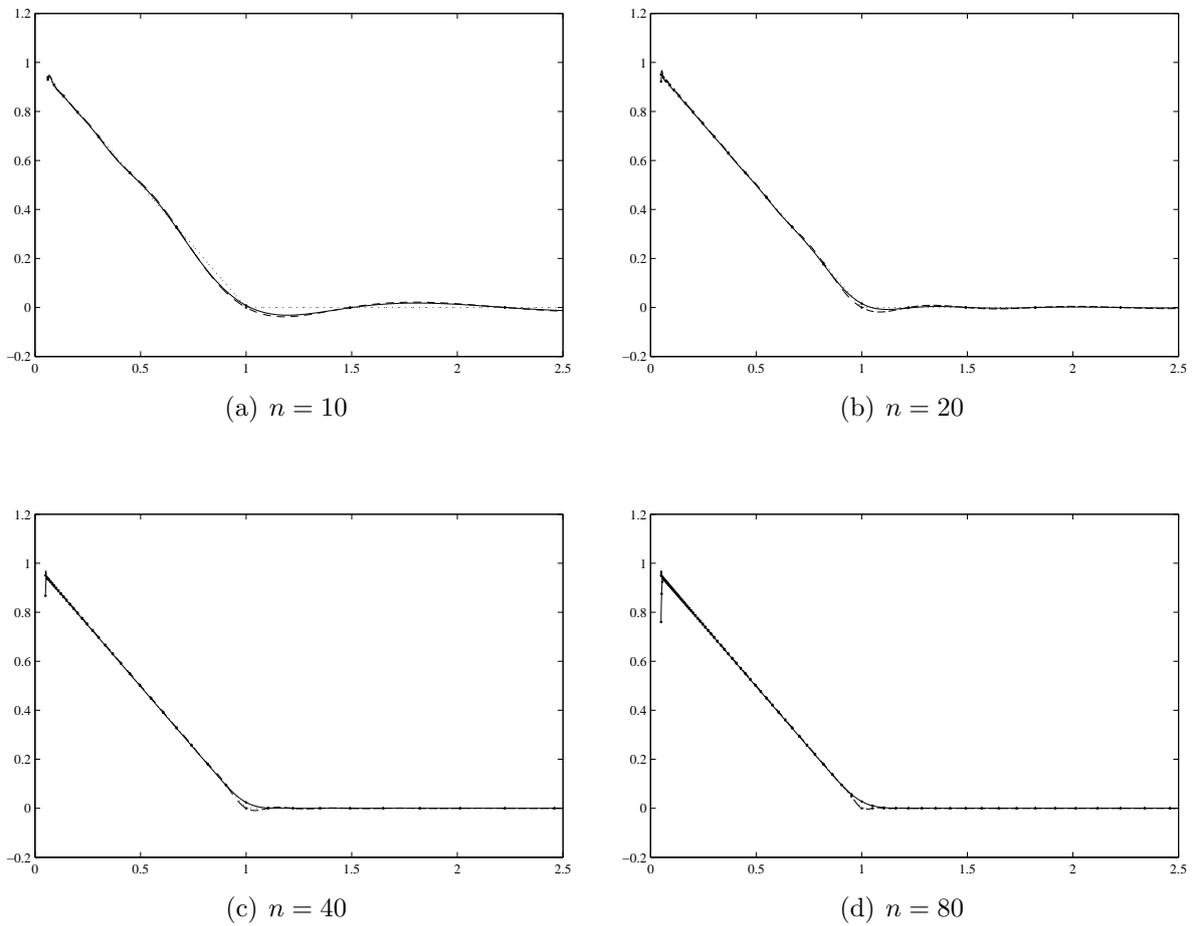
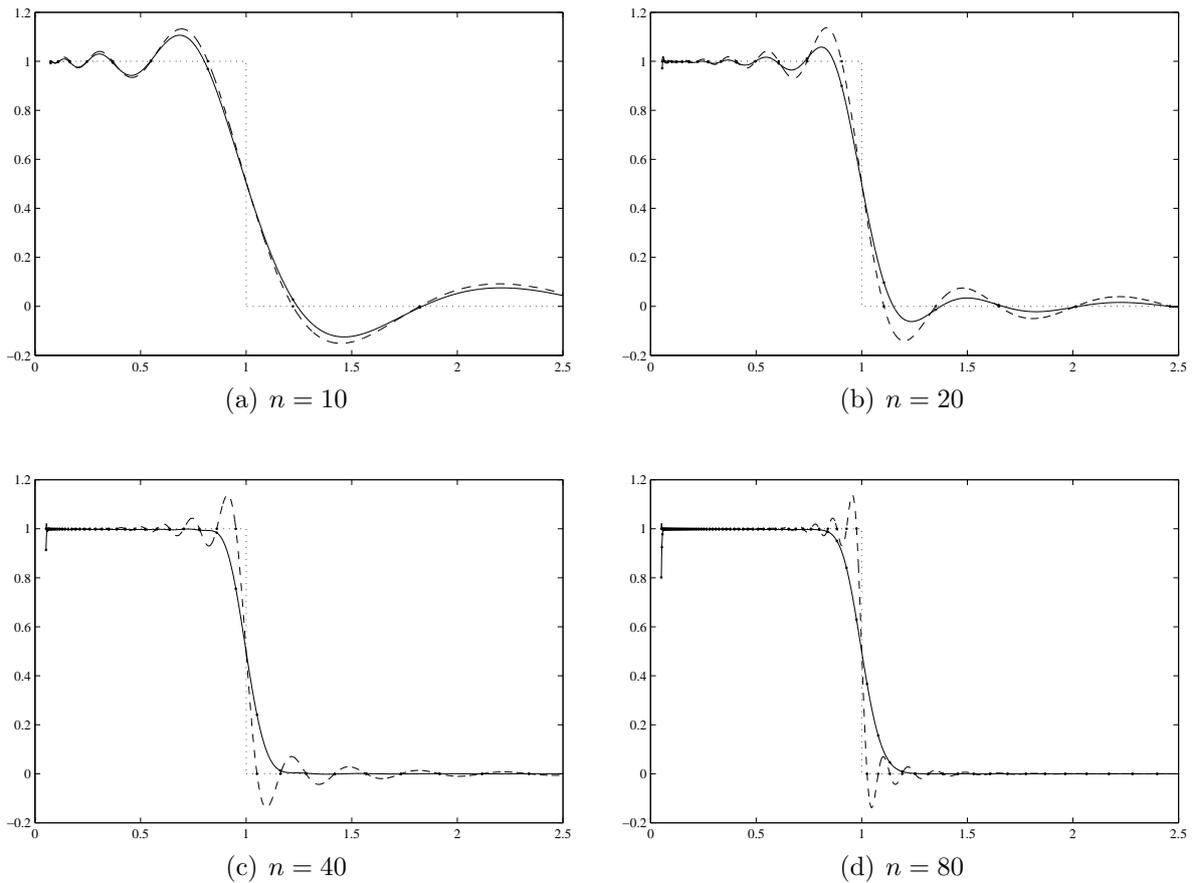


Table 4.1.: Convergence for plain-vanilla payoff (left) and binary payoff (right)

$n$	$\ g - S_{g,X}\ _\infty$	$ V(1) - V_n(1) $	$n$	$\ g - S_{g,X}\ _\infty$	$ V(1.1) - V_n(1.1) $
10	0.0367	0.02995	10	0.1371	0.002455
20	0.0177	0.00883	20	0.1334	0.006609
40	0.0088	0.00212	40	0.1333	0.001660
80	0.0045	0.00052	80	0.1333	0.000408
160	0.0022	0.00013	160	0.1333	0.000101
320	0.0011	0.00003	320	0.1333	0.000025

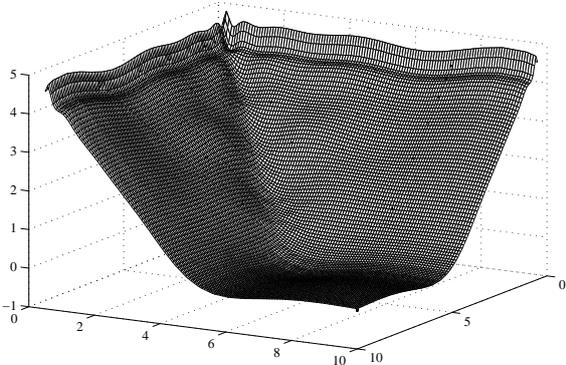
Figure 4.2.: Convergence of the RBF interpolant for binary payoff in 1D: The plots show the payoff and option interpolants for several discretizations ( $n = 10, 20, 40, 80$ ). Apparently, the interpolants of the payoff do not converge with respect to the supremum norm. Nevertheless, the option value (solid line) converges quadratically in  $n$ . The reason is that high frequencies are damped out. (See section 4.4.1.)



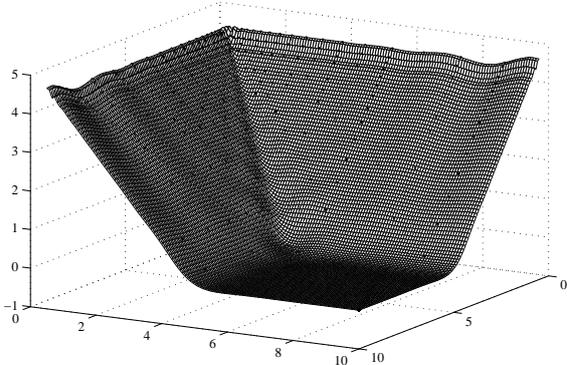
### Summary

Although classical error bounds for Gaussian interpolation are restricted to a rather small class of functions, it is possible to obtain error bounds for Sobolev spaces. However, practical tests indicate a better convergence than the theoretical result. The convergence of the interpolant to the payoff in the supremum norm seems to be linear for continuous payoffs. For discontinuous payoffs, the interpolant does not converge in the supremum norm, but seems to converge in an  $\mathcal{L}^1$ -sense. The approximation error for the option value is of order  $\mathbf{O}(h^2)$ .

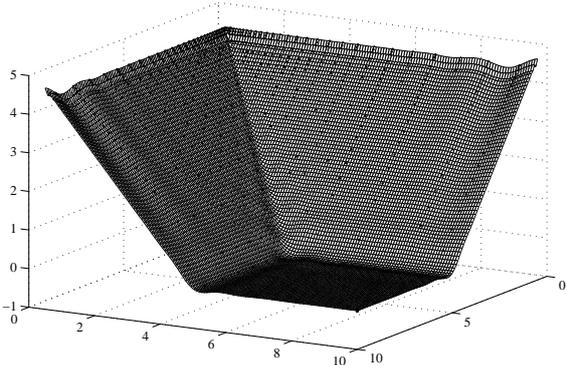
Figure 4.3.: Illustration of interpolants for continuous payoffs ( $d = 2$ ).



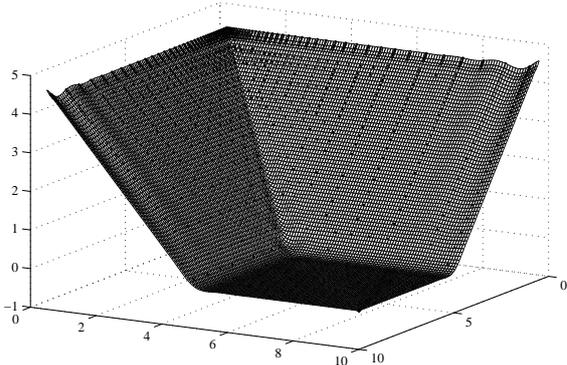
(a)  $n = 10 \times 10$



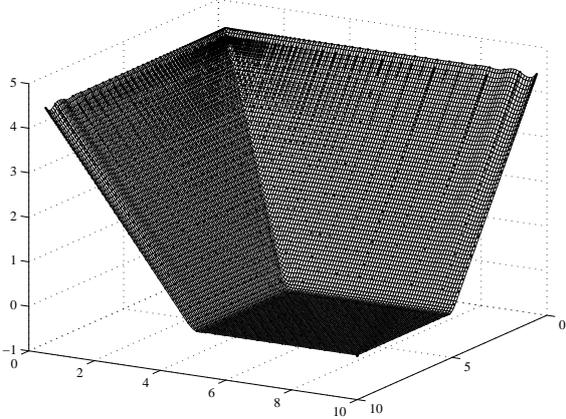
(b)  $n = 20 \times 20$



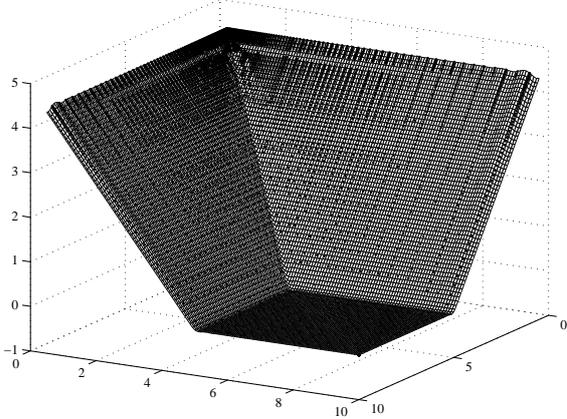
(c)  $n = 30 \times 30$



(d)  $n = 40 \times 40$

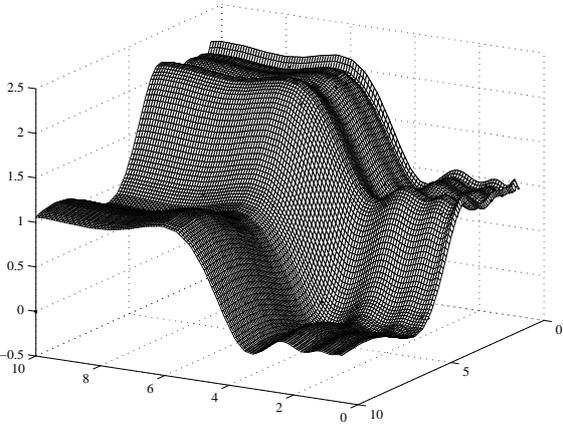


(e)  $n = 60 \times 60$

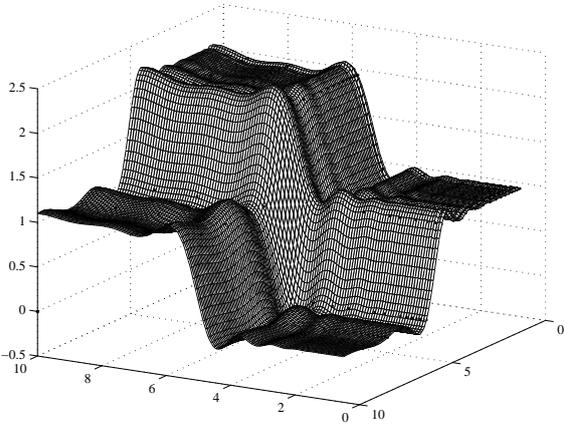


(f)  $n = 80 \times 80$

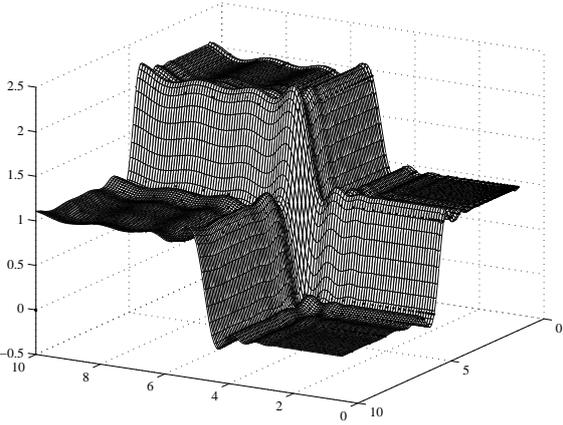
Figure 4.4.: Illustration of interpolants for discontinuous payoffs ( $d = 2$ ).



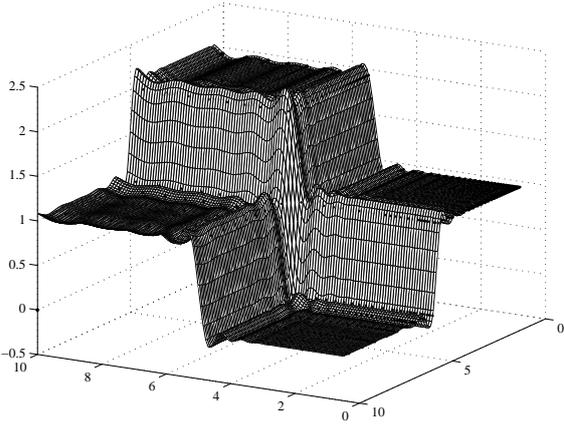
(a)  $n = 10 \times 10$



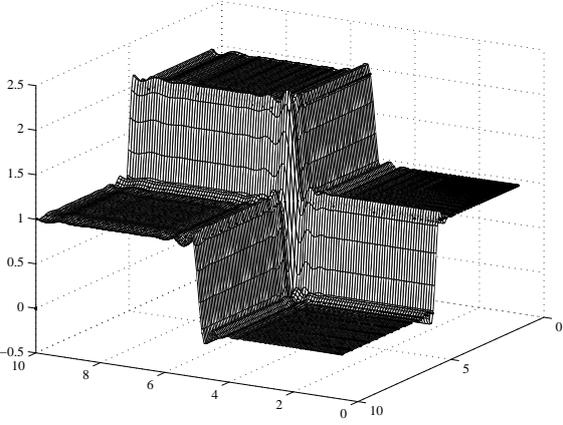
(b)  $n = 20 \times 20$



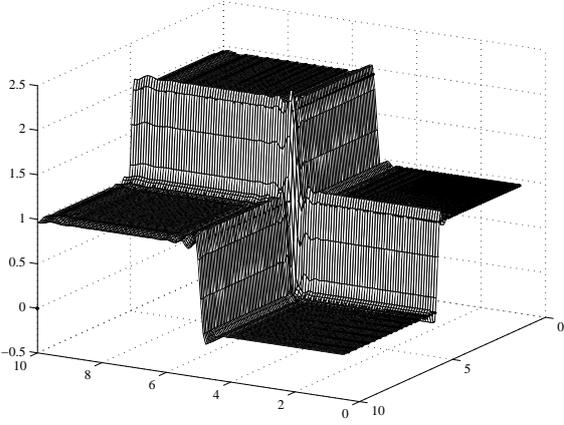
(c)  $n = 30 \times 30$



(d)  $n = 40 \times 40$



(e)  $n = 60 \times 60$



(f)  $n = 80 \times 80$

### 4.2.3. Structure and condition of the basis matrix $B$

An interpolation with radial basis functions involves the solution of system (4.9):

$$\vec{f} = B\lambda, \quad \text{with } B = (b_{ij})_{i,j=1,\dots,n}, \quad b_{ij} := e^{-\alpha\|x_i - x_j\|_2^2} \quad (4.10)$$

Thus, the condition of  $B$  is important for the stability of the interpolation method. The values of  $B$  depend on the location of the centers  $x_j \in X \subset \mathbb{R}^d$  and of the shape parameter  $\alpha > 0$ . In principle, the shape parameter can be chosen individually for each center  $x_j$ , but in the following we focus on the case of a homogeneous shape parameter  $\alpha$ .

#### Special case of equidistant nodes for $d = 1$

In the case of equidistant nodes with fill distance  $h$  it is useful to choose the shape parameter as follows:

$$\alpha := -h^{-2} \log \beta, \quad (4.11)$$

with a control parameter  $\beta \in (0, 1)$ . Then,  $x_{j+1} - x_j = h$ , and the value of the basis function for the node  $x_j$  at the neighboring node  $x_{j+1}$  is

$$\phi(x_i - x_j) = e^{-\alpha(h|i-j|)^2} = e^{(h^{-2} \log \beta)h^2|i-j|^2} = \beta^{|i-j|^2}.$$

Figure 4.5 gives a geometrical motivation for this choice of  $\alpha$ : The dependence between interpolation coefficients  $\lambda_j, \lambda_{j+1}$  for neighboring nodes can be controlled by  $\beta$ . It is intuitively clear that  $\beta \approx 1$  leads to a strong dependence and  $\beta \approx 0$  leads to a weak dependence. This is reflected in a high or low condition number of the basis matrix  $B$ .

For the choice (4.11) of the shape parameter in the one-dimensional, equidistant case the matrix  $B$  has the form

$$B = B_n := \begin{pmatrix} 1 & \beta & \beta^4 & \dots & \beta^{n^2} \\ \beta & 1 & \beta & \dots & \beta^{(n-1)^2} \\ \beta^4 & \beta & 1 & \dots & \beta^{(n-2)^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^{n^2} & \beta^{(n-1)^2} & \beta^{(n-2)^2} & \dots & 1 \end{pmatrix}.$$

The condition  $\text{cond}(B_n)$  of the matrices  $B_n$  depends on  $\beta$  and  $n$ . It is clear that  $\text{cond}(B_n(\beta)) \rightarrow 1$  for  $\beta \rightarrow 0$  and  $\text{cond}(B_n(\beta)) \rightarrow \infty$  for  $\beta \rightarrow 1$  (for  $n \geq 2$ ). For a fixed  $\beta \in (0, 1)$ , the condition converges for  $n \rightarrow \infty$  to a constant  $c = c(\beta)$ . A choice of  $\beta$  that leads to a “moderate” condition is, e.g.,  $\beta = 0.7$ . The convergence is illustrated in figure 4.6. In this case, the limit of the 2-norm condition is  $c(\beta) \approx 505.04$ , which can be considered “moderate”.<sup>7</sup>

<sup>7</sup>Of course, for iterated solutions of eq. (4.10) this condition were to high, but in fact, the iteration step of the RBF pricing method is (4.6) – and this step is stable (see section 4.4.1, p. 83).

Figure 4.5.: Controlling the dependence between interpolation coefficients by  $\beta$ .

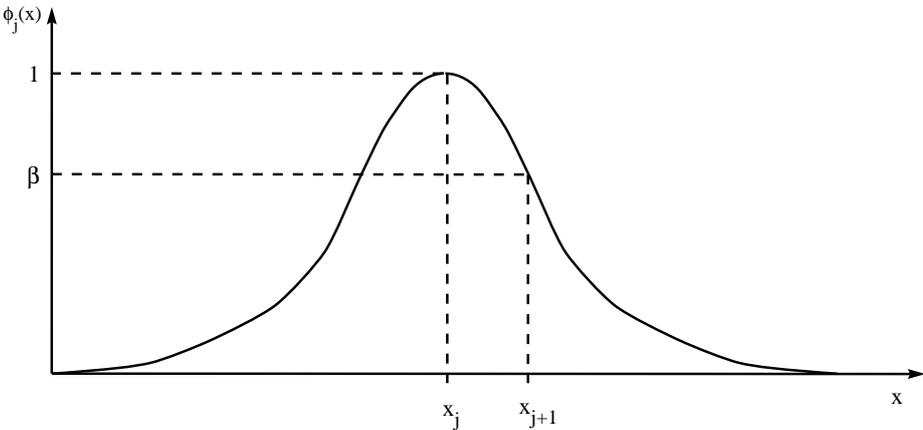
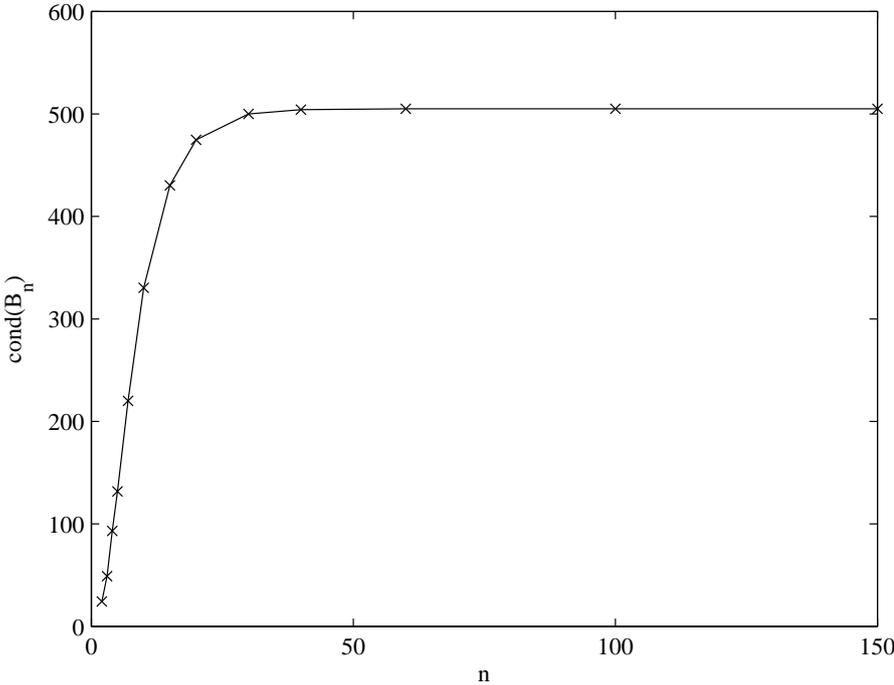


Figure 4.6.: Convergence of condition number  $\text{cond}_2(B_n)$  for  $n \rightarrow \infty$ .



Besides the condition number, the structure of  $B$  is relevant for an efficient solution of system (4.9).

**Lemma 4.8** (Toeplitz structure of the basis matrix). *In the one-dimensional, equidistant case, i.e., for nodes  $x_j = x_1 + h(j-1) \in \mathbb{R}$ , the matrix  $B$  defined in (4.9) has a Toeplitz<sup>8</sup> structure.*

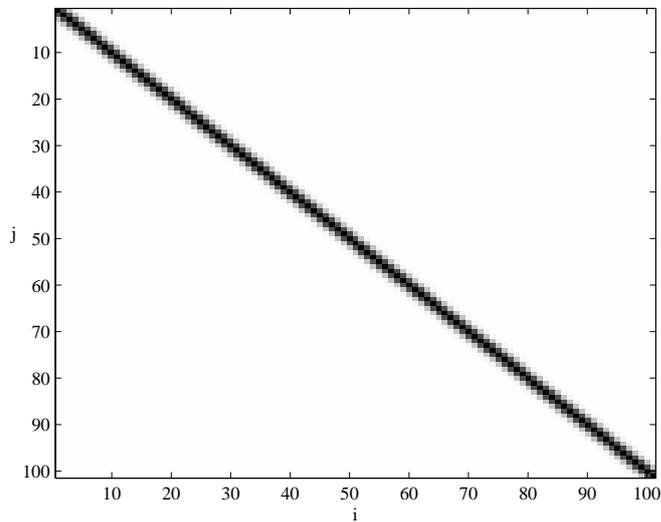
*Proof.* The entries  $b_{ij} = \phi(\|x_i - x_j\|)$  only depend on  $x_i - x_j = h(i-j)$  and thus, only on  $i-j$ . Thus,  $B$  is a Toeplitz matrix.  $\square$

**Lemma 4.9** (Quasi band structure of the basis matrix). *For a one-dimensional model and equidistant centers the matrix  $B$  defined in (4.9) has a “quasi band” structure, i.e.,  $b_{ij} = \mathbf{O}(e^{-|i-j|})$ .*

*Proof.* The entries of  $B$  are by definition  $b_{ij} = e^{-\alpha|x_i - x_j|} = e^{-\alpha|h(i-j)|} = \mathbf{O}(e^{-|i-j|})$ .  $\square$

This property guarantees, that the basis matrix can be treated as a band matrix by cutting off all values below a certain threshold. Figure 4.7 illustrates the typical structure of the basis matrix  $B$  graphically.

Figure 4.7.: Example of the structure of  $B$  for the one-dimensional case ( $d = 1$ ) with equidistant nodes. Large values are displayed black, values near zero are displayed white. Here, for  $n = 100 \times 100$ ,  $\text{cond}_2(B) \approx 505$ .



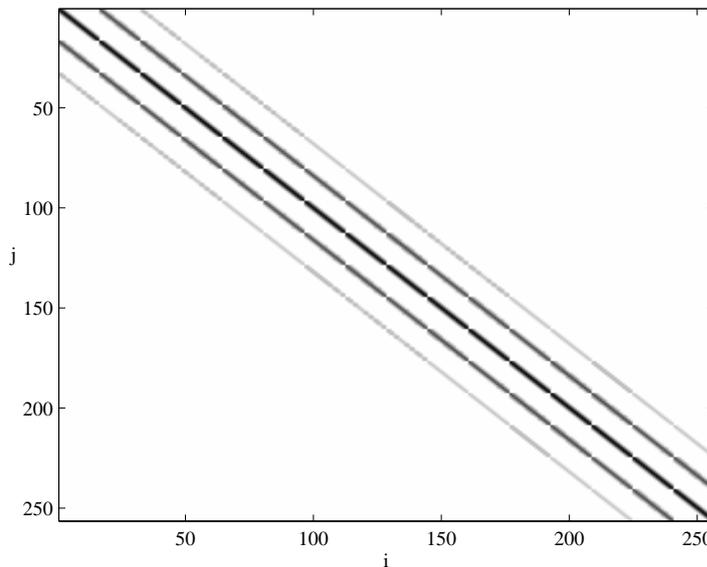
<sup>8</sup>A matrix  $A$  is called *Toeplitz* matrix, if its entries  $a_{ij}$  depend only on the difference  $i-j$ , i.e., all entries on (sub-) diagonals are identical.

### Special case of equidistant nodes for $d > 1$

In the multi-dimensional case  $d > 1$  with a rectangular grid, the structure of  $B$  is more difficult. The structure of  $B$  for such an example is illustrated in figure 4.8. It is a regular pattern of sub-matrices of the same form as the one-dimensional basis matrices. As in the one-dimensional case, it can be observed empirically that the condition converges for  $n \rightarrow \infty$ .

The matrix  $B$  has also a quasi band structure, but the band is by substantially wider than in the one-dimensional case. Consequently, in the multi-dimensional case  $B$  can be regarded (and treated numerically) as a sparse matrix.

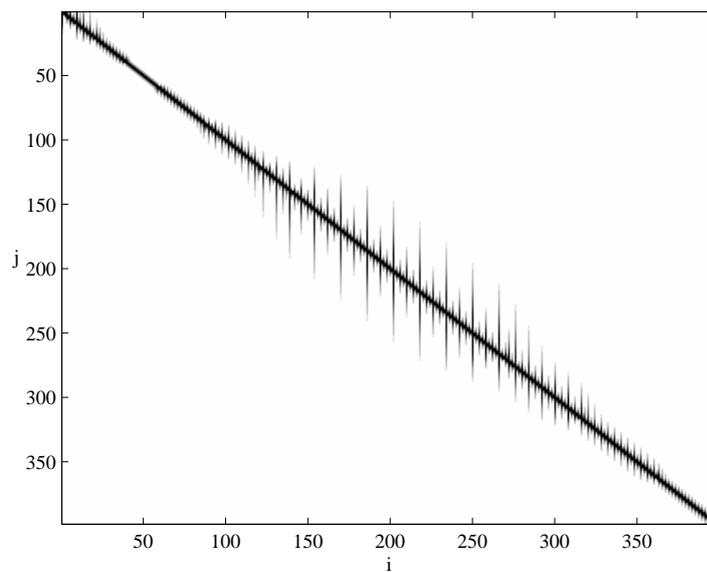
Figure 4.8.: Example of the structure of  $B$  for the two-dimensional case ( $d = 2$ ) with equidistant nodes. Large values are displayed black, values near zero are displayed white. Here, for  $n = 16 \times 16$ ,  $\text{cond}_2(B) \approx 304$ .



### Case of non-equidistant nodes

For an adaptive placement of nodes, the structure of  $B$  depends on the refinement procedure. The refinement strategy described for the spline method in section 3.3.1 can be applied in the context of the RBF method, as well. It leads to basis matrices that can be regarded as wide band matrices. Figure 4.9 shows a basis matrix that has been generated for a binary put option.

Figure 4.9.: Example of the structure of  $B$  for the one-dimensional case ( $d = 1$ ) with adaptive, non-equidistant nodes. Large values are displayed black, values near zero are displayed white. In this case, node dependent shape parameters  $\alpha_j$  are used, which are determined by the level of refinement. Here,  $\text{cond}_2(B) \approx 6.89 \cdot 10^6$  (computed by the condition estimator “condest” in MATLAB). The option used for this example is the same binary put option as in figure 4.2.



### General case

The general structure (and thus condition) of the basis matrix depends on the placement of nodes, choice of shape parameters, and, in the case of adaptive nodes, of the option type. Consequently, general a-priori bounds on the condition number cannot be made. Thus, for practical applications of the RBF method it is advisable to estimate the condition of  $B$  automatically for every run of the pricing routine.

### 4.3. Quadrature rules for the model matrix

The quadrature problems that arise for the extension of the spline method to multi-dimensional underlyings are difficult, as the corresponding integrands are not  $\mathcal{C}^\infty$ -smooth. This is not the case for the RBF method. Due to section 4.1, the integrands occurring in the entries of the model matrix  $M = (m_{ij})_{i,j=1,\dots,n}$  are of the form

$$m_{ij} := \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x_i}(\xi) \phi(\xi - x_j) \, d\xi. \quad (4.12)$$

With the choice  $\phi(x) = e^{-\alpha\|x\|_2^2}$  from (4.10), the second factor of the integrand is  $\mathcal{C}^\infty$ -smooth. The first factor, the density function  $f$ , is determined by the market model. For the Black-Scholes model, the Merton model, and NIG model it is  $\mathcal{C}^\infty$ -smooth, as well, as is shown later (in chapter 5). For the *univariate* VG model the integrand can be split up according to the singularity.<sup>9</sup> The *multivariate* VG model is not considered in the present work. In the following the case  $f \in \mathcal{C}^\infty$  is considered.

#### 4.3.1. Gauss quadrature

Gauss quadrature is efficient, if many integrals have to be evaluated with the same weight function  $\omega$  and if the integrand (without the weight) can be approximated well by a polynomial. Classical Gauss quadrature can be found in numerical analysis textbooks and does not need to be discussed here in detail. The following error representation can be found, e.g., in [Sto99]:

**Theorem 4.10** (Approximation error). *The approximation error of the Gauss quadrature for each function  $f \in \mathcal{C}^{2n}[a, b]$  can be expressed as*

$$\int_a^b \omega(x) f(x) \, dx - \sum_{i=1}^n w_i f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!} (p_n, p_n),$$

where  $\omega$  denotes the weight function,  $w_i$  and  $x_i$  the  $n$  nodes and weights of the quadrature rule,  $p_n$  denotes the orthogonal polynomial of order  $n$  and  $\xi \in (a, b)$ .

For the RBF pricing method, the weight function  $\omega(x) := e^{-x^2}$  (Gauss-Hermite quadrature) is used. It is clear that for an accurate estimation of the quadrature error, the derivative  $f^{(2n)}$  in the error representation above must be estimated. Instead, several Gauss rules with increasing number  $n$  of nodes are used and the remaining error is estimated heuristically by the difference of two successive approximations.

---

<sup>9</sup>See section 5.1.4.

**Implementation for  $d = 1$** 

An implementation of Gauss-Hermite quadrature can be found, e.g., in [PTVF02]. This algorithm has been used for the implementation of the RBF pricing method for one-dimensional underlyings. It allows Gauss quadrature with  $n = 150$  nodes, which corresponds to a polynomial order of about 300. This suffices for the present purposes. Nevertheless, all quadrature results are tested against a second Gauss quadrature with lower polynomial order to ensure that the required accuracy is reached.

**Implementation for  $d > 1$** 

For multivariate quadrature, a product rule based on Gauss-Hermite quadrature has been used. For dimensions  $d = 2$  and  $d = 3$  this turned out to be practicable. For higher dimensions specialized rules for integration over infinite regions with Gaussian weights have been proposed by Genz and Keister [GK96] based on the prior work [Gen86]. An implementation in Fortran is available<sup>10</sup> and has been used for this work.

**4.3.2. Analytical quadrature for Black-Scholes**

The RBF method for the Black-Scholes model can be implemented in a very efficient way. Standard calculus allows for an analytical solution of the integrals (4.12). As the derivation is technical, only the result is presented, which can be incorporated into the RBF valuation algorithm for the one-dimensional Black-Scholes model.

Let  $K_1 \in \mathbb{R}$ ,  $K_2 \geq 0$ ,  $K_3 \geq 0$ ,  $K_4 \in \mathbb{R}$  be some constants which depend on the model's parameters. Let  $x_i, x_j \in X$  be two nodes. For each entry of the model matrix  $M$  we need to solve an integral of the form

$$I := \int_{-\infty}^{\infty} K_4 \exp\{-K_2(K_3\xi + x_j - x_i - K_1)^2 - \frac{1}{2}\xi^2\} d\xi.$$

The analytic solution is

$$I = \frac{K_4\sqrt{\pi}}{\sqrt{K_2K_3^2 + \frac{1}{2}}} \exp\left\{-\frac{\frac{1}{2}K_2[(x_j - x_i)^2 - 2K_1(x_j - x_i) + K_1^2]}{K_2K_3^2 + \frac{1}{2}}\right\}.$$

Analytic solutions also exist for the multi-variate Black-Scholes models, but are not included in this work. For convenience, the Gauss-Hermite product rule from section 4.3.1 has been used for  $d > 1$ .

---

<sup>10</sup>The code can be found on the home page of Alan Genz:  
<http://www.math.wsu.edu/faculty/genz/homepage>.

## 4.4. Numerical properties

This section analyzes the numerical properties of the RBF pricing method.

### 4.4.1. Stability

For each exercise time of the Bermudan option, the RBF method solves step (4.7):

$$v_k = \max(g, aMB^{-1}v_{k+1}),$$

where  $g$  is the payoff vector,  $a$  the discount factor,  $M$  the model matrix,  $B$  the basis matrix,  $v_{k+1}$  the option price approximation at time  $t_{k+1}$ , and  $v_k$  the option price approximation at time  $t_k$ . As taking the maximum is numerically stable, it suffices to verify the stability of the iteration

$$w_k = aMB^{-1}w_{k+1}. \quad (4.13)$$

As seen in section 4.2.3, the condition of  $B$  is “moderate” in most cases. Then, the stability depends on the development of additive errors of  $w_{k+1}$  in each step (4.13), i.e., on the spectral radius of  $A := aMB^{-1}$

$$\rho(A) := \max_{i=1,\dots,n} |\mu_i|,$$

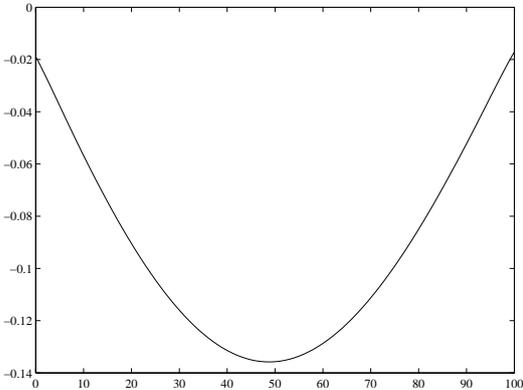
where  $\mu_1 > \dots > \mu_n$  denote the eigenvalues of  $A$ . An analytical estimation of  $\rho(A)$  is not possible, as  $A$  is influenced by numerous factors, e.g., the dimension of the underlying, the number and location of centers  $x_j$ , as well as model type and parameters.

Although it is not proved here analytically, for practical tests, e.g., for all examples in chapter 5, it is found that  $\rho(A) < 1$ . The RBF implementation used for this work monitors the largest eigenvalue automatically. In most cases, the largest eigenvalue is  $\mu_1 \approx 1$  and the following eigenvalues decay fast to zero. In fact, the stability is better than  $\mu_1 \approx 1$  suggests. When using one-dimensional models, the eigenvectors correspond directly to “eigenfunctions”. It can be observed that the largest eigenvector corresponds to an oscillation with a long period, and the smaller eigenvectors correspond to oscillations with short periods. This phenomenon is intuitively clear, as (4.13) approximates one time step of a diffusion and diffusions damp oscillations. Figure 4.10 provides an illustration of eigenvectors for several eigenvalues. The figures have been plotted using a one-dimensional Black-Scholes model with  $n = 101$  equidistant nodes. The consequence of the described phenomenon is, that non-smooth errors, such as rounding errors, are strongly damped down.

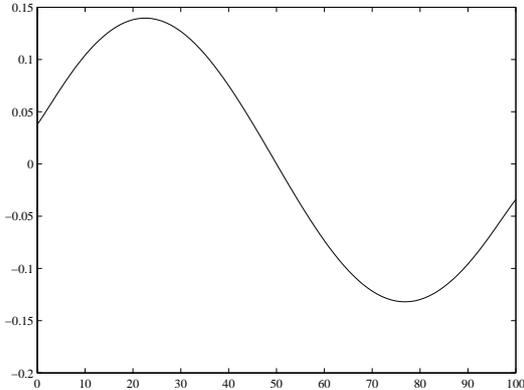
### Summary

The empirical finding of  $\rho(A) < 1$  indicates stability of the RBF method. Although a proof for some special cases seems possible, it is not contained in the present work.

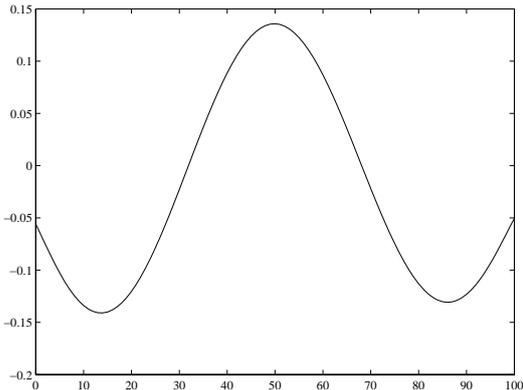
Figure 4.10.: Eigenvectors of  $A = aMB^{-1}$ . All components of the eigenvectors for several eigenvalues  $\mu_i$  are displayed. The  $x$ -axis shows the component index.



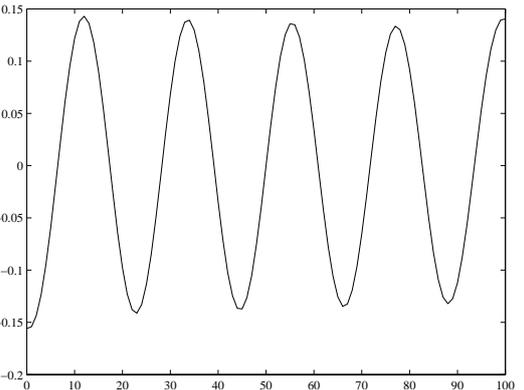
(a)  $\mu_1 = 0.9904$



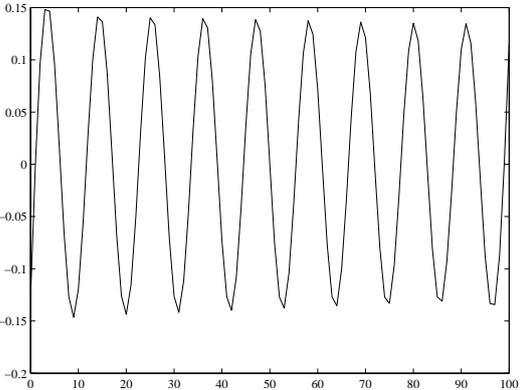
(b)  $\mu_2 = 0.9621$



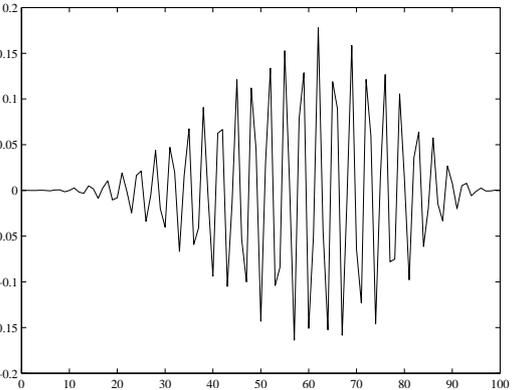
(c)  $\mu_3 = 0.9169$



(d)  $\mu_{10} = 0.3826$



(e)  $\mu_{20} = 0.0226$



(f)  $\mu_{101} = 8.9 \cdot 10^{-15}$

### 4.4.2. Computational complexity

The RBF pricing method for Bermudan options involves two steps, namely the setup of the matrices  $B$  and  $M$ , and the solution of (4.7) for each time step. The costs for the computation of the entries of  $B$  are negligible, but each entry of  $M$  involves the solution of a quadrature problem. In some cases the structure of the model matrix can be used to save a large amount of the quadrature costs.

#### Structure of the model matrix $M$

**Lemma 4.11** (Toeplitz structure of model matrix). *In the one-dimensional case with equidistant centers, the model matrix  $M$  has a Toeplitz structure.*

*Proof.* The entries of the model matrix  $M = (m_{ij})$  are by definition (p. 66):

$$\begin{aligned} m_{ij} &= \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=x_i}(\xi) \phi_j(\xi - x_j) \, d\xi \\ &\stackrel{(2.2)}{=} \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=0}(\xi - x_i) \phi_j(\xi - x_j) \, d\xi \\ &= \int_{\mathbb{R}^d} f^{X_{t_{k+1}}|X_{t_k}=0}(\xi) \phi_j(\xi - (x_j - x_i)) \, d\xi \end{aligned}$$

It should be noted that this is only possible under space homogeneity (assumption 2.5). For the equidistant case  $x_j = x_1 + h(j - 1)$ ,  $\phi_j = \phi$  has been chosen and thus  $m_{ij}$  depends only on  $(x_j - x_i)$ . As the nodes are equidistant, it depends only on  $i - j$  and consequently, the model matrix  $M$  is a Toeplitz matrix.  $\square$

An  $n \times n$  Toeplitz matrix  $M$  has the structure

$$M = \begin{pmatrix} m_1 & m_2 & m_3 & \dots & m_n \\ m_{n+1} & m_1 & m_2 & & m_{n-1} \\ m_{n+2} & m_{n+1} & m_1 & & m_{n-2} \\ \vdots & & & \ddots & \vdots \\ m_{2n-1} & m_{2n-2} & m_{2n-3} & \dots & m_1 \end{pmatrix}.$$

As can be seen directly from this structure, the number of different entries in such a matrix is  $2n - 1$  instead of  $n^2$  in a general matrix. This fact reduces the number of necessary quadratures for the setup of the model matrix by a factor  $n$ , if  $M$  has Toeplitz structure. The costs for each quadrature are the same and thus the computational complexity for setting up the model matrix  $M$  is for the Toeplitz case  $\mathbf{O}(n)$ .

#### Special case: One-dimensional model and equidistant centers

In the special case of an one-dimensional model and equidistant centers  $x_j$  the matrix  $B$  is a Toeplitz quasi band matrix (by lemma 4.8 and 4.9, p. 78). That means, the matrix

can be approximated well by a band matrix  $\tilde{B}$  by “cutting off” the small entries. This reduces the complexity for the solution of the linear system considerably. It is known that the LU-decomposition for band matrices gives two band triangular matrices  $L, U$  ([GVL96], §4.3):

**Theorem 4.12** (Band LU factorization). *Suppose  $A \in \mathbb{R}^{n \times n}$  has an LU factorization  $A = LU$ . If  $A$  has upper bandwidth  $q$  and lower bandwidth  $p$ , then  $U$  has upper bandwidth  $q$  and  $L$  has lower bandwidth  $p$ .*

In addition, for  $n \gg p, q$  Golub/Van Loan give an algorithm for the LU-factorization in  $\mathbf{O}(2npq)$  flops and an algorithm for solving the band triangular systems  $L$  and  $U$  in  $\mathbf{O}(2n(p+q))$  flops.<sup>11</sup> This is *linear* complexity in  $n$ .

### Summary

For one-dimensional models and equidistant nodes the total computational complexity of the RBF pricing method is  $\mathbf{O}(mn)$ , where  $m$  denotes the number of exercise times of the Bermudan option and  $n$  the number of centers.

For multi-dimensional models (or non-equidistant nodes) the costs for an LU-decomposition of  $B$  are  $\mathbf{O}(n^3)$ . The costs for the setup of the model matrix are  $\mathbf{O}(n^2)$ , and the costs for the solution of the linear systems are  $\mathbf{O}(mn^2)$ . Consequently, the total computational complexity is  $\mathbf{O}(mn^2 + n^3)$ .

**Remark 4.13** (Sparsity for higher-dimensional models). *For higher-dimensional models the matrices  $M$  and  $B$  do not have a Toeplitz quasi-band structure but are quasi-sparse, i.e., many entries are close to zero. Neglecting such entries leads to sparse matrices. Of course it is difficult to analyze the speed-up in the higher-dimensional case exactly and this aspect is not discussed here. Nevertheless, the implementation uses sparse matrix structures and can be switched to iterative solvers.*

### 4.4.3. Convergence

The convergence of the RBF method can, in principle, be established analogously to the convergence result of the spline method in section 3.4.1 (p. 43). The three kinds of errors are again interpolation errors, truncation errors, and quadrature errors. In contrast to the spline method, the quadrature errors are negligible for the RBF method, as the integrands (4.12) are  $\mathcal{C}^\infty$ -smooth (p. 81). The truncation errors can be estimated analogously to the estimation of the truncation errors for the spline method. Thus, the convergence depends only on the interpolation error.

The theoretical error bound 4.7 (p. 71) for the Gaussian interpolation can be used to establish qualitative convergence for, e.g., plain-vanilla options. For binary options it is useless, as these payoff functions do not live in  $W_2^m(\mathbb{R}^d)$ . Numerical tests in section

<sup>11</sup> See [GVL96], p. 152f.

4.2.2 indicate that the interpolants do *not* converge in the  $\mathcal{L}^\infty$ -norm but in the  $\mathcal{L}^1$ -norm. This suggests error bounds for Gaussian interpolation of discontinuous functions in an  $\mathcal{L}^1$ -sense, but the development of new error bounds for RBF interpolation is beyond the scope of this work.

To establish a hypothesis on the convergence behavior, numerical experiments have been performed. For Bermudan options, figure 4.11 illustrates some of these experiments. The results indicate quadratic convergence for Bermudan options in the number of nodes  $n$  and thus justify the following hypothesis.

**Hypothesis 4.14** (Convergence). *The RBF based approximation  $V_{m,n}(x)$  of the value of a univariate Bermudan option with  $m$  exercise times and  $n$  equidistant centers converges quadratically in  $n$  for every log-price  $x$  of the underlying for the models from chapter 5.*

**Remark 4.15** (Convergence for multivariate models). *The above hypothesis cannot be extended to multivariate models, as the currently available processing power does not allow for discretizations that are fine enough to draw conclusions about the limit behavior for  $n \rightarrow \infty$ .*

## 4.5. Conclusion

The RBF method is also based on the reduction principle (lemma 2.12, p. 14), and thus closely related to the spline method. The main idea is to use RBF interpolation for the option value function. As a by-product of the interpolation the initial quadrature problem is reduced to a smooth integrand, and thus all errors due to singularities<sup>12</sup> in the option price are shifted from quadrature to interpolation. By step (4.4), p. 66, integration and summation are exchanged and the integration problems are relocated to the model matrix, which can be build in a preliminary setup step. Consequently, the quadrature problems must be solved only once, namely during the setup of the model matrix, and not in every time step as in the spline method. This leads to substantial performance gains for Bermudan options with high numbers of exercise times. In fact, options with, e.g.,  $m = 10000$  time steps can be priced easily.

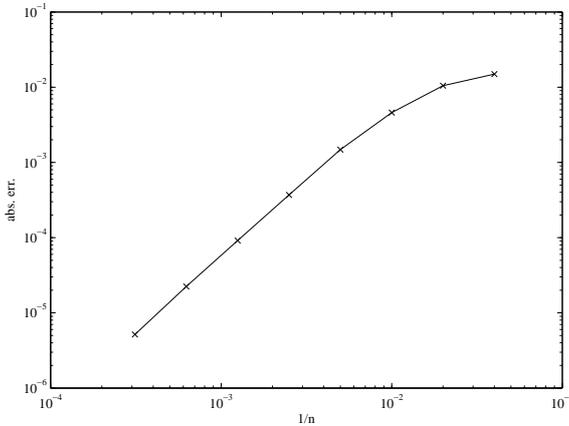
By design, the method can be used for pricing Bermudan and American options with arbitrary payoffs and under fairly arbitrary models. In contrast to the spline method it generalizes naturally to higher dimensions. However, the computational costs still considerably increase with the dimension. A technical aspect is that the RBF method is matrix oriented and thus can be implemented easily in MATLAB or other interpretative numerical environments.

An important property of the RBF method is order of convergence in  $n$ . Empirical tests indicate quadratic convergence for Bermudan options.

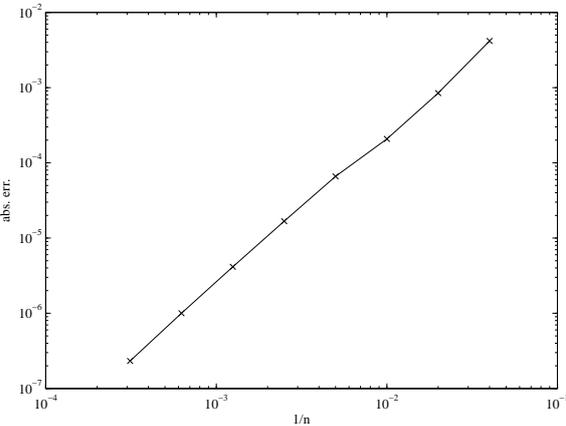
---

<sup>12</sup>“Singularity” here refers to non-differentiability or discontinuity.

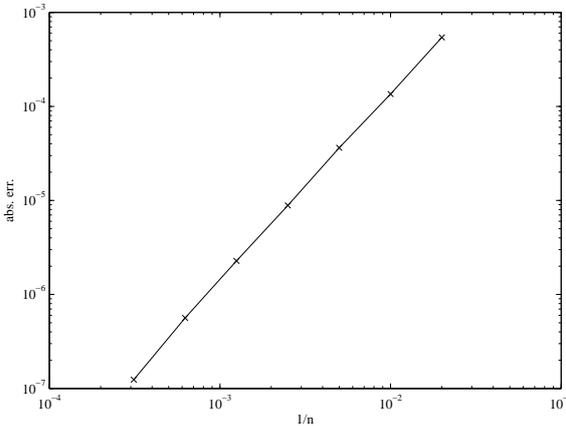
Figure 4.11.: Empirical convergence behavior of the RBF method for Bermudan options with  $m = 100$  exercise times and  $n$  centers,  $n \rightarrow \infty$ . The illustrations show log-log plots of  $1/n$  versus the absolute error of the approximation. (The quadratic convergence can be seen from the asymptotic slope of 2.)



(a) binary option (B/S)



(b) plain-vanilla put (B/S)



(c) plain-vanilla put (Merton)

## 5. Asset price models

As the value of an option depends on the price of the underlying asset, any option pricing model must include an asset price model. This chapter introduces asset price models that can be employed with the spline and RBF method. It is a prerequisite for the application of the new pricing methods.

The presented models assume that the log-price  $X_t$  of the underlying follows a stochastic Lévy process.<sup>1</sup> Such models are called *exponential Lévy models* or just *Lévy models*. Already the geometric Brownian motion (GBM) used by Black, Scholes [BS73], and Merton [Mer73] falls into this class of models. Although this model is widely used, it does not reproduce the phenomenon of price jumps. In the last decade, models with jumps received increasing attention. On the one hand, there is empirical evidence for the occurrence of jumps in real markets.<sup>2</sup> On the other hand, calibration results indicate that models with jumps provide a more realistic description of market prices than pure diffusion models like the geometric Brownian motion.<sup>3</sup>

Section 5.1 introduces several one-factor models with jumps from [Mer76], [MS90], and [Ryd97], and discusses their suitability for the use with quadrature methods. Section 5.2 briefly introduces the multi-factor Black-Scholes model and the notion of copulae, which allows the construction of multi-factor models with a dependence structure between the individual factors. Finally, section 5.3 gives an outlook on non-parametric modeling as an example for a modeling technique that is inaccessible to most other valuation techniques.

### 5.1. One-factor models

In the following several asset price models for single assets are introduced and their relevant properties for an application within quadrature methods are analyzed. Each model is characterized by the conditional probability density function (PDF) of the asset price. The most important properties are smoothness and tail behavior of the PDF and the possibility of its efficient numerical evaluation. The selection of models in this section follows the current literature on option pricing. The question, whether these models are good descriptions of market prices or not, is not an issue here and is not critical to the

---

<sup>1</sup>See definition 5.1. An introduction to Lévy processes for financial modeling can be found in [CT03].

<sup>2</sup>Empirical studies and econometric details can be found, e.g., in [CLM97] and [Pag96].

<sup>3</sup>A calibration of several Lévy models (including B/S, NIG, and VG) to S&P 500 index option prices is discussed in [Mat05].

development of the spline and RBF method. If future research discovers better models, they can be easily attached to existing implementations of the spline or RBF method by just exchanging a single subroutine.

### 5.1.1. General assumptions

Usual option pricing models share several basic assumptions. Some of them can be economically motivated and some of them are purely technical to establish mathematical tractability. Typical assumptions are the following:

- There are no transaction costs, fees, taxes, etc.
- Transactions have no influence on the market price.
- Any new information becomes simultaneously available to all market participants.
- There are no arbitrage opportunities.
- The interest rates for borrowing and lending are equal.
- The market price of the underlying follows a certain stochastic process  $(X_t)_{t \geq 0}$ .

The asset price models introduced in this section describe only one aspect of the option pricing model, namely, the stochastic process for the price of the underlying. This work only considers Lévy processes, which are defined as follows.

**Definition 5.1** (Lévy process). *A stochastic process  $X_t$  on  $(\Omega, \mathcal{F}, \mathbb{P})$  with values in  $\mathbb{R}^d$  such that  $X_0 = 0$  is called a Lévy process, if it has the following properties:*

- (i) *Independent increments: For all times  $t_0 < t_1 \leq s_0 < s_1$  the random variables  $(X_{t_1} - X_{t_0})$  and  $(X_{s_1} - X_{s_0})$  are stochastically independent.*
- (ii) *Stationary increments: For all time increments  $\Delta t$ , the distribution of  $(X_{t+\Delta t} - X_t)$  does not depend on  $t$ .*
- (iii) *Stochastic continuity:  $\forall t, \varepsilon > 0 : \lim_{\Delta t \rightarrow 0} \mathbb{P}(|X_{t+\Delta t} - X_t| \geq \varepsilon) = 0$ .*

In contrast to the continuity of the Brownian motion, the weaker constraint of *stochastic* continuity in (iii) allows jumps in the asset price. Property (iii) implies assumption 2.4, property (ii) corresponds to the stationarity assumption 2.7(i), and property (i) implies that the (optional) space homogeneity assumption 2.5 holds for all Lévy models with density functions.

**Remark 5.2** (Restriction of this work to Lévy models). *Although all models introduced in this section are Lévy models, the spline and RBF methods are not restricted to this*

class of models. One central property of a Lévy process is the independence of increments (i). This property is not required for the application with quadrature methods. Examples for stochastic processes which do not have independent increments are processes with mean reversion, e.g., the Ornstein-Uhlenbeck process proposed for interest rate modeling by Vasicek [Vas77] and by Cox, Ingersoll, and Ross [CIR85]. A popular non-Lévy model that is used for option pricing is the Heston model [Hes93], which is an extension of the Black-Scholes model by a stochastic volatility ( $\sigma_t$ ) that follows an Ornstein-Uhlenbeck process. The valuation by quadrature methods in this case leads to a two-dimensional pricing problem, with one dimension being the asset price ( $S_t$ ) and the other being the stochastic volatility ( $\sigma_t$ ). This additional complexity does also apply to finite difference methods for the Heston model.

### 5.1.2. The Black-Scholes model

The following model was used by Black and Scholes in their seminal work [BS73]. As it is well-known and widely used in practice, this section is kept at a minimum. A more detailed introduction can be found, e.g., in [Sey02]. In this model the asset price  $S_t$  is assumed to follow a geometric Brownian motion with drift  $\tilde{\gamma}$  and infinitesimal variance  $\sigma^2$ , i.e.,

$$dS_t = \tilde{\gamma}S_t dt + \sigma S_t dW_t,$$

where  $W_t$  denotes a Wiener process. This representation of the model is equivalent to the following one for the *log-price*  $X_t := \log S_t$ :<sup>4</sup>

$$X_t = \gamma t + \sigma W_t$$

Together with the stationarity property of the Wiener process, this implies that the *log-returns*  $X_{t+\Delta t} - X_t$  are distributed normally with mean  $\gamma\Delta t$  and variance  $\sigma^2\Delta t$  for all  $t$  and  $\Delta t$ . The conditional probability density function of the log-price  $X_t$  is

$$f^{X_t|X_0=0}(x) = \frac{1}{\sigma\sqrt{2\pi t}} \exp\left\{-\frac{(x-\gamma t)^2}{2\sigma^2 t}\right\}. \quad (5.1)$$

(In the following the superscripts are omitted, if they are obvious from the context.) The density function  $f$  is  $\mathcal{C}^\infty$ -smooth and can be evaluated efficiently in  $\mathbf{O}(1)$ .<sup>5</sup> These properties allow efficient quadrature schemes for the spline and RBF method. The quadrature problems arising in the RBF method when applied to this model can even be solved analytically, as is mentioned in section 4.3.2.

<sup>4</sup>The equivalence can be shown using Itô's lemma; the drift parameter is changed to  $\gamma$ .

<sup>5</sup>"In  $\mathbf{O}(1)$ " means that the computational complexity of the function evaluation is  $\mathbf{O}(1)$ .

### Martingale condition

The Black-Scholes model allows only one choice for the equivalent martingale measure (EMM).<sup>6</sup> The following *risk-neutrality condition* guarantees that the process  $S_t$  adjusted for dividends and discounted with the risk-free interest rate is a martingale:

$$\mathbb{E}(S_t|S_0 = 1) = e^{t(r-q)} \quad (\forall t \geq 0), \quad (5.2)$$

where  $r$  denotes the risk-free interest rate and  $q$  denotes the continuous dividend rate paid by the asset. The risk-neutrality condition is a consequence of the no-arbitrage assumption. It implies for the Black-Scholes model:

$$\begin{aligned} \mathbb{E}(S_t|S_0 = 1) &= \mathbb{E}(e^{X_t}|X_0 = 0) \\ &= \int_{-\infty}^{\infty} e^x f^{X_t|X_0=0}(x) dx = \dots = \\ &= \exp\left\{t\left(\gamma + \frac{\sigma^2}{2}\right)\right\} \stackrel{!}{=} \exp(t(r - q)) \end{aligned}$$

This determines the drift that is necessary to satisfy the risk-neutrality condition:

$$\gamma = r - q - \frac{\sigma^2}{2}.$$

The choice of  $\gamma$  corresponds to the choice of an equivalent martingale measure. In this case the EMM is uniquely determined, as the Black-Scholes model is a complete model, i.e., any security can be hedged in this model.

### Tail behavior

The tail behavior is important for the truncation of quadrature intervals and for a classification of the model. The tail behavior of the Black-Scholes PDF in equation (5.1) can be described as:

$$f(x) = \mathbf{O}(e^{-\alpha x^2}) \text{ for } |x| \rightarrow \infty \quad (5.3)$$

for  $\alpha := \frac{1}{2\sigma^2 t} - \varepsilon > 0$ , with a small  $\varepsilon > 0$ .<sup>7</sup> Tails of this kind are called *thin tails*. Empirical evidence rejects the hypothesis of thin tails for stock and stock index prices. This property is one of the major drawbacks of the Black-Scholes model. An symptom of this misbehavior is the *volatility smile* of implied volatilities.<sup>8</sup>

<sup>6</sup>See e.g., [Shr04], second fundamental theorem of asset pricing (theorem 5.4.9, p. 232).

<sup>7</sup>The notation “ $\mathbf{O}(\dots)$  for  $|x| \rightarrow \infty$ ” means that the function is in this class for both limit cases  $x \rightarrow \infty$  and  $x \rightarrow -\infty$ .

<sup>8</sup>Calibrating the Black-Scholes model to market prices of options for different strikes  $K$  leads to non-constant volatility  $\sigma = \sigma(K)$ . This phenomenon is well-known as “volatility smile,” as the graph of  $\sigma(K)$  resembles a smile. This phenomenon is described, e.g., in [Hul00], §17.

### 5.1.3. The Merton model

The following generalization of the Black-Scholes model was proposed first by Merton [Mer76]. It assumes that the dynamics of the log price can be described partly by diffusion and partly by jumps. Furthermore, the occurrence of jumps over time is assumed to follow a Poisson process, and the jump sizes are assumed to be normally i.i.d. (independent and identically distributed). Thus the log-price  $X_t$  can be written as

$$X_t = \gamma t + \sigma W_t + \sum_{i=1}^{N_t} Y_i,$$

where  $\gamma$  is a drift parameter,  $\sigma$  is a volatility parameter,  $(W_t)$  a Wiener process,  $(N_t)$  a Poisson process counting the jumps in the interval  $[0, t]$ , and  $(Y_i)$  is a family of random variables with  $Y_i \sim \mathcal{N}(\mu, \delta^2)$  describing the size of the  $i$ -th jump. The corresponding probability density function can be derived by combining the PDFs  $(f_k)_{k \geq 0}$ , where each

$$f_k(x) := f^{X_t | X_0=0 \wedge N_t=k}(x)$$

is the conditional PDF of the log-price under the additional condition that  $N_t = k$  jumps occur in  $[0, t]$ .  $f_k$  is the Gaussian PDF augmented by the sum of  $k$  jumps with jump sizes distributed as  $\mathcal{N}(\mu, \delta^2)$ .<sup>9</sup>

$$f_k(x) = \frac{1}{\sqrt{2\pi(\sigma^2 t + k\delta^2)}} \exp\left(-\frac{(x - \gamma t - k\mu)^2}{2(\sigma^2 t + k\delta^2)}\right) \quad (5.4)$$

Each  $f_k$  must be weighted with the probability  $\mathbb{P}(N_t = k)$  for the condition  $N_t = k$ :

$$f^{X_t | X_0=0} = \sum_{k=0}^{\infty} \mathbb{P}(N_t = k) \cdot f_k.$$

The probability for  $k$  Poisson jumps in  $[0, t]$  is

$$\mathbb{P}(N_t = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \quad (5.5)$$

Equations (5.4) and (5.5) imply that the PDF has the following series expansion:<sup>10</sup>

$$f^{X_t | X_0=0}(x) = \sum_{k=0}^{\infty} \frac{e^{-\lambda t} (\lambda t)^k}{k! \sqrt{2\pi(\sigma^2 t + k\delta^2)}} \cdot \exp\left(-\frac{(x - \gamma t - k\mu)^2}{2(\sigma^2 t + k\delta^2)}\right). \quad (5.6)$$

As explained in lemma 2.6 (p. 12), this defines the general conditional PDF for arbitrary  $t > s \geq 0$ ,  $x^* \in \mathbb{R}$  via the stationarity property:

$$f^{X_t | X_s=x^*}(x) = f^{X_{t-s} | X_0=x^*}(x) = f^{X_{t-s}+x^* | X_0=0}(x) = f^{X_{t-s} | X_0=0}(x - x^*)$$

<sup>9</sup>Remember: A sum of  $k$   $\mathcal{N}(\mu, \delta^2)$ -distributed i.i.d. random variables has distribution  $\mathcal{N}(k\mu, k\delta^2)$ .

<sup>10</sup>Compare [CT03], p. 111, eq. (4.12).

Obviously the function  $f$  is  $\mathcal{C}^\infty$ -smooth. The terms of the infinite sum decay very fast to zero for  $k \rightarrow \infty$ . Consequently this function can be approximated by adding up only the first few terms until the required accuracy is reached. As stop criterion it is sufficient to check the size of the next terms. If they do not change the result up to a given tolerance level, the series can be truncated. This naive implementation turns out to be efficient enough in practice.

It is important that for each  $k$  the corresponding term of the series allows an analytic solution of the integral problems arising in connection with the RBF method. The analytic solution is lengthy but involves only the exponential function and can thus be evaluated efficiently. It is not included in this work but can be obtained by standard analysis. As the series converges uniformly, integration and summation can be interchanged and a series expansion for the integral can be obtained.

**Remark 5.3** (Model parametrization). *The parametrization of the Merton model in the literature is not consistent. For example, [Hau97] uses the parameters  $\lambda$  (jump intensity) and  $\gamma$  (“percentage of the total volatility explained by the jumps”<sup>11</sup>). The present work follows the parametrization used in [CT03]:*

- $\sigma > 0$  is the diffusion volatility,
- $\lambda \geq 0$  is the jump intensity,
- $\mu \in \mathbb{R}$  the mean jump size, and
- $\delta \geq 0$  the standard deviation of jump size.

The drift rate  $\gamma \in \mathbb{R}$  is eliminated by the risk-neutrality condition as follows.

### Martingale condition

Proposition 3.14 (p. 92) from [CT03] provides an alternative representation of the expectation value of an exponential Lévy process, namely,

$$\mathbb{E}(e^{X_t}) = e^{t\Psi(-i)},$$

where  $\Psi$  denotes the *characteristic exponent*<sup>12</sup> of the process and  $i$  the imaginary unit. Together with the martingale condition (5.2) this implies

$$\Psi(-i) = r - q. \tag{5.7}$$

The characteristic exponent for the Merton model is ([CT03], table 4.3, p. 112):<sup>13</sup>

$$\Psi_{Merton}(u) = -\frac{\sigma^2 u^2}{2} + i\gamma u + \lambda(e^{-\delta^2 u^2/2 + i\mu u} - 1)$$

<sup>11</sup>[Hau97], p. 8.

<sup>12</sup>The characteristic exponent is the exponent of the characteristic function in the representation  $\Phi(z) = \exp(\Psi(z))$  and is uniquely determined for any Lévy process ([CT03], (3.15), p. 83).

<sup>13</sup>The notation of the drift parameter is inconsistent in [CT03]. It is  $b$  in table 4.3 and  $\gamma$  on p. 111.

Consequently, the risk-neutrality condition for this model is:

$$\gamma = r - q - \frac{\sigma^2}{2} - \lambda(e^{\delta^2/2+\mu} - 1)$$

It suffices to choose the drift  $\gamma$  accordingly and thus eliminate one parameter making the Merton model a four-parametric model.<sup>14</sup> This way to establish the martingale property is called *drift correction*. As the Merton model is incomplete, alternative choices of the EMM would be possible, e.g., by adjusting the mean jump size.

### Tail behavior

**Lemma 5.4** (Infinite tail decay rate for Merton's model). *The Merton model has infinite exponential tail decay rate, i.e., for every  $\alpha > 0$  the tail decays at least as*

$$f(x) = \mathbf{O}(e^{-\alpha|x|}) \text{ for } |x| \rightarrow \infty.$$

*Proof.* Let  $\alpha > 0$  be an arbitrary but fixed number. As  $f \geq 0$ ,  $\limsup e^{\alpha|x|} f(x) \geq 0$  for both  $x \rightarrow \infty$  and  $x \rightarrow -\infty$ . It is sufficient to show  $\limsup e^{\alpha|x|} f(x) < \infty$  for both limit cases. As the series representation of  $f$  in (5.6) converges uniformly, the order of the following limits may be reversed.

$$\begin{aligned} \lim_{x \rightarrow \infty} e^{\alpha|x|} f(x) &= \lim_{x \rightarrow \infty} \lim_{n \rightarrow \infty} e^{\alpha|x|} \sum_{k=0}^n C_k \cdot \exp\left(-\frac{(x - \gamma t - k\mu)^2}{2(\sigma^2 t + k\delta^2)}\right) \\ &= \lim_{n \rightarrow \infty} \lim_{x \rightarrow \infty} e^{\alpha|x|} \sum_{k=0}^n C_k \cdot \exp\left(-\frac{(x - \gamma t - k\mu)^2}{2(\sigma^2 t + k\delta^2)}\right) \\ &= \lim_{n \rightarrow \infty} 0 = 0, \end{aligned}$$

where  $C_k$  denotes constants with respect to  $x$ . The limit case  $x \rightarrow -\infty$  is analogous.  $\square$

Cont and Tankov characterize the tail behavior of the Merton model as follows: "Tails are heavier than Gaussian, but all exponential moments are finite."<sup>15</sup>

#### 5.1.4. The Variance-Gamma model

This type of asset price model has been proposed by Madan and Seneta in [MS90]. It has been used for the valuation of European options, e.g., in [MCC98] and [CM99]. The model is a pure jump model, i.e., the price moves only by jumps (and deterministic drift). This section mainly follows the results of [MCC98], where a closed form of the probability density function is derived.

<sup>14</sup>Not counting the risk-free interest rate  $r$  and the continuous dividend rate  $q$ . Parameters:  $(\sigma, \lambda, \mu, \delta)$ .

<sup>15</sup>[CT03], p. 112.

### Construction

The idea behind the construction of the variance-gamma (VG) model is to evaluate a Brownian motion ( $W_s$ ) with drift at a random time ( $s_t$ ) that is described by a gamma process. The random time process is called *subordinator*. Both processes are assumed to be stochastically independent. A general introduction to the construction of Lévy processes by subordination can be found in [CT03], section 4.2.2.

The gamma process is defined as the homogeneous Lévy process ( $s_t$ ) for which the density of  $s_1$  is determined by the gamma distribution, i.e., the PDF of  $z := s_1$  is assumed to be the following function:

$$f^{s_1}(z; k, \theta) := \frac{z^{k-1} e^{-z/\theta}}{\theta^k \Gamma(k)},$$

where  $k > 0$  and  $\theta > 0$  denote the two parameters of the distribution and  $\Gamma$  denotes the gamma function. It is useful to restrict<sup>16</sup> this class of processes by the assumption  $\mathbb{E}(z) = 1$  and change the parametrization to  $\kappa := \text{var}(z)$ .<sup>17</sup> This corresponds to the choices  $\theta = \kappa = 1/k$  and leads to the density function

$$f^{s_1}(z; \kappa) = \frac{z^{1/\kappa-1} e^{-z/\kappa}}{\kappa^{1/\kappa} \Gamma(1/\kappa)}.$$

In this special case, the resulting process has three parameters: The volatility  $\sigma$  of the Brownian motion, the drift  $\theta$  of the Brownian motion, and the variance  $\kappa$  of the time process. A possible interpretation of this construction is that the calendar time differs from the “business time” at which new events occur at the market. The additional parameters allow control of kurtosis<sup>18</sup> and skewness of the return density and thus can lead to better calibration results. As the gamma process used for subordination is a pure jump process, the resulting variance gamma process is also a pure jump process.

Just as for the Merton model, a drift correction term  $\gamma t$  is added to choose the equivalent martingale measure. This drift correction takes place in the non-subordinated time. Thus, the drift parameter  $\gamma$  has a different meaning than the subordinated drift parameter  $\theta$ . The process for the log-price in the VG model is

$$X_t = \gamma t + \theta s_t + \sigma W_{s_t}.$$

### Properties

From the construction it is obvious that the VG process  $X_t$  reduces to a Brownian motion with drift  $\gamma$  and infinitesimal variance  $\sigma^2$  for  $s_t = t$ , i.e., in the limit case  $\kappa \rightarrow 0$ . In this spirit the VG model can be regarded as a generalization of the Black-Scholes model.

<sup>16</sup>Because of certain scaling properties this is not a restriction of the class of processes. Details can be found in [CT03], p. 116.

<sup>17</sup>Mean and variance of the gamma distribution are  $\mathbb{E}(z) = k\theta$ ,  $\text{var}(z) = k\theta^2$ .

<sup>18</sup>“Kurtosis” is defined as the fourth cumulant divided by the square of the variance of a distribution. It can be interpreted as a shape parameter of the density function.

For the application of quadrature methods, the main ingredient is the conditional PDF of the log-price  $X_t$ . As for the Merton model, the PDF can be derived using the properties of the subordinator process and the subordinated process. The resulting PDF is<sup>19</sup>

$$f^{X_t|X_0=0}(x) = \frac{2 \exp(\kappa(x - \gamma t)/\sigma^2)}{\sigma \sqrt{2\pi} \Gamma(t/\kappa) \kappa^{t/\kappa}} \left( \frac{(x - \gamma t)^2}{2\sigma^2/\kappa + \theta^2} \right)^{\frac{t}{2\kappa} - \frac{1}{4}} \cdot K_{\frac{t}{\kappa} - \frac{1}{2}} \left( \frac{1}{\sigma^2} \sqrt{(x - \gamma t)^2 \left( \frac{2\sigma^2}{\kappa} + \theta^2 \right)} \right), \quad (5.8)$$

where  $K_\nu$  denotes the modified Bessel function of the second kind with index  $\nu := \frac{t}{\kappa} - \frac{1}{2}$  and  $\Gamma$  denotes the gamma function. Without loss of generality it can be assumed that  $\nu \in (-\frac{1}{2}, 0)$ :  $t, \kappa > 0$  and if  $\nu \geq 0$ ,  $n := \lceil \frac{2t}{\kappa} \rceil + 1$  time steps of size  $\frac{t}{n}$  can be performed instead of a single  $t$ -step. It should be noted that the modified Bessel function of second kind is symmetric with respect to  $\nu$ , i.e.  $K_\nu = K_{-\nu}$ .<sup>20</sup>

This density function is not globally  $\mathcal{C}^\infty$ -smooth but piecewise for  $(-\infty, \gamma t)$  and  $(\gamma t, \infty)$ .<sup>21</sup> It has a singularity at  $x = \gamma t$  with  $f(x) \rightarrow \infty$  for  $x \rightarrow \gamma t$ . See (5.9) below for the asymptotic behavior.

### Numerical evaluation

Although technically more involved, this function can be evaluated efficiently in  $\mathbf{O}(1)$ . In the following some aspects of the implementation are mentioned.

- The singularity must be considered in the quadrature method, e.g., by additive separation: The Bessel function  $K_\nu$  has the following asymptotic behavior for the argument  $y \rightarrow 0$ :

$$\text{for } \nu \neq 0: K_\nu(y) \rightarrow \frac{\Gamma(|\nu|)}{2} \left( \frac{2}{y} \right)^{|\nu|} \quad (5.9)$$

The PDF in (5.8) has the structure

$$f(x) = c_1 e^{c_2(x-\gamma t)} |x - \gamma t|^\nu K_\nu(c_3 |x - \gamma t|),$$

where  $c_1, c_2, c_3$  denote constants. Substituting asymptote (5.9) into this function leads to the structure (with other constants  $\tilde{c}_i$ )

$$\tilde{f}(x) = \tilde{c}_1 e^{\tilde{c}_2(x-\gamma t)} |x - \gamma t|^\nu \cdot |x - \gamma t|^{-|\nu|} = \tilde{c}_1 e^{\tilde{c}_2(x-\gamma t)} |x - \gamma t|^{2\nu} \quad (\nu < 0).$$

<sup>19</sup>See e.g., [MCC98], theorem 1; the notation is different: here  $(\sigma, \theta, \kappa) \hat{=} (\sigma, \theta, \nu)$  in [MCC98].

<sup>20</sup>The index symmetry  $K_\nu = K_{-\nu}$  can be found, e.g., in [CT03], appendix A, p. 499, or any formulary.

<sup>21</sup>Figure 5.1 (p. 105) contains an illustration of the VG density.

The integrals arising in the RBF method are of the form (compare section 4.1)

$$\begin{aligned} & \int_{-\infty}^{\infty} e^{-\lambda(\xi-x_i)^2} f(\xi-x_j) d\xi \\ = & \int_{-\infty}^{\infty} \underbrace{e^{-\lambda(\xi-x_i)^2} (f-\tilde{f})(\xi-x_j)}_{(*)} d\xi + \int_{-\infty}^{\infty} \underbrace{e^{-\lambda(\xi-x_i)^2} \tilde{f}(\xi-x_j)}_{(**)} d\xi, \end{aligned}$$

where  $\lambda > 0$  is the shape parameter of the RBF interpolation,  $x_i$  and  $x_j$  are nodes with index  $i$  and  $j$ . By construction of  $\tilde{f}$  the integrand  $(*)$ , which involves  $(f-\tilde{f})$ , does not have a singularity. Thus, the left integral can be evaluated by numerical quadrature. The problem is now reduced to the approximation of the right integral with integrand  $(**)$ , which involves the function  $\tilde{f}$ , i.e., the following integral must be solved for index  $\nu \in (-\frac{1}{2}, 0)$ :

$$\begin{aligned} & \int_{-\infty}^{\infty} e^{-\lambda(\xi-x_i)^2} \tilde{f}(\xi-x_j) d\xi \\ = & \tilde{c}_1 \int_{-\infty}^{\infty} e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} |\xi-x_j-\gamma t|^{2\nu} d\xi \\ = & -\tilde{c}_1 \int_{-\infty}^{x_j+\gamma t} e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} (\xi-x_j-\gamma t)^{2\nu} d\xi \\ & + \tilde{c}_1 \int_{x_j+\gamma t}^{\infty} e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} (\xi-x_j-\gamma t)^{2\nu} d\xi \\ = & +\tilde{c}_1 \int_{-\infty}^{x_j+\gamma t} \frac{-2\lambda\xi + 2\lambda x_i + \tilde{c}_2}{2\nu+1} e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} (\xi-x_j-\gamma t)^{2\nu+1} d\xi \\ & -\tilde{c}_1 \int_{x_j+\gamma t}^{\infty} \frac{-2\lambda\xi + 2\lambda x_i + \tilde{c}_2}{2\nu+1} e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} (\xi-x_j-\gamma t)^{2\nu+1} d\xi \\ & -\tilde{c}_1 |\dots|_{-\infty}^{x_j+\gamma t} + \tilde{c}_1 |\dots|_{x_j+\gamma t}^{\infty} \end{aligned}$$

The last two terms  $|\dots|_{-\infty}^{x_j+\gamma t}$  and  $|\dots|_{x_j+\gamma t}^{\infty}$  vanish; e.g.,

$$|\dots|_{x_j+\gamma t}^{\infty} = \left| e^{-\lambda(\xi-x_i)^2 + \tilde{c}_2(\xi-\gamma t)} \frac{1}{2\nu+1} (\xi-x_j-\gamma t)^{2\nu+1} \right|_{x_j+\gamma t}^{\infty} \stackrel{(\lambda>0)}{=} 0.$$

The remaining integrands do not have singularities (as  $2\nu+1 > 0$ ) and can be solved with numerical quadrature methods.

- After the separation of the singularity, the numerical quadrature must be split up at  $x = x_j + \gamma t$ , and for each part  $x < x_j + \gamma t$  and  $x > x_j + \gamma t$  the improper integral must be approximated separately.

- Computational aspects concerning the modified Bessel function and gamma function can be found in [PTVF02], p. 249ff. Numerical routines for the approximation of the modified Bessel function for arbitrary real indices can be found in [ZJ96].

### Martingale condition

Analog to the derivation of the Merton model the risk-neutrality condition can be derived using the characteristic exponent of the process via equation (5.7)

$$\Psi(-i) = r - q.$$

The characteristic exponent of the VG process is<sup>22</sup>

$$\Psi_{VG}(u) = i\gamma u - \frac{1}{\kappa} \log \left( 1 + \frac{u^2 \sigma^2 \kappa}{2} - i\theta \kappa u \right).$$

And this yields the following risk-neutrality condition for the VG model:

$$\gamma = r - q + \frac{1}{\kappa} \log \left( 1 - \frac{\sigma^2 \kappa}{2} - \theta \kappa \right)$$

Obviously such a value  $\gamma$  exists only for

$$\theta < \frac{1}{\kappa} - \frac{\sigma^2}{2}. \quad (5.10)$$

Otherwise the expectation value  $\mathbb{E}(S_t)$  does not exist (is infinite). In this case the martingale property cannot be restored by an additional drift term. There could be alternative approaches, but for this work the model parameters can be restricted to values which fulfill relation (5.10).<sup>23</sup>

### Tail behavior

The following result can be found in [CT03], §4.4.3 (p. 117).

**Lemma 5.5.** *The log-return probability density of the VG model has exponential tails with decay rates  $B - A$  for  $x \rightarrow \infty$  and  $B + A$  for  $x \rightarrow -\infty$ , with  $A := \frac{\theta}{\sigma^2}$  and  $B := \frac{\sqrt{\theta^2 + 2\sigma^2/\kappa}}{\sigma^2}$ .*

**Lemma 5.6** (Lower bound for right VG tail decay rate). *The right tail of the probability density function  $f$  of the VG model (5.8) decays exponentially with decay rate of at least one, i.e.,  $B - A > 1$ .*

<sup>22</sup>The characteristic exponent can be found in [CT03], table 4.5, p. 117. It must be augmented here by the term  $i\gamma u$ , as the drift correction term is not included in this table.

<sup>23</sup>This should not be a problem in practice either; as the calibration procedure does not converge to parameter values, for which the expectation value of  $S_t$  is infinite.

*Proof.* This is a direct consequence of relation (5.10).

$$\begin{aligned} B - A &= \frac{\sqrt{\theta^2 + 2\sigma^2/\kappa}}{\sigma^2} - \frac{\theta}{\sigma^2} > 1 \\ \Leftrightarrow \sqrt{\theta^2 + 2\sigma^2/\kappa} - \theta - \sigma^2 &> 0 \\ \Leftrightarrow \theta < \frac{1}{\kappa} - \frac{\sigma^2}{2}, \end{aligned}$$

which is given for all admissible parameter values from (5.10). The last equivalence can be shown by case differentiation of  $\theta + \sigma^2 \geq 0$  and  $\theta + \sigma^2 < 0$ .  $\square$

### 5.1.5. The Normal-Inverse-Gaussian model

The Normal-Inverse-Gaussian (NIG) model has been proposed for financial modeling in [BN97] and [Ryd97]. It is closely related to the VG model and can also be represented by Brownian subordination.

#### Construction

For the representation by subordination, an *inverse Gaussian* process is used as subordinator. The inverse Gaussian process is defined as the homogeneous Lévy process  $s_t$  where the density of  $s_1$  is determined by the inverse Gaussian distribution, i.e., where the density of  $z := s_1$  is<sup>24</sup>

$$f^s(z; \delta, \omega) = (2\pi)^{-1/2} \delta \exp(\delta\omega) z^{-3/2} \exp\{-\frac{1}{2}(\delta^2 z^{-1} + \omega^2 z)\},$$

where  $\delta > 0$  and  $\omega > 0$  are parameters of the distribution. Mean and variance are  $\mathbb{E}(z) = \delta/\omega$  and  $\text{var}(z) = \delta/\omega^3$ . As in the VG construction, this two-parameter family of processes is restricted to a one-dimensional family by the assumption  $\mathbb{E}(s_1) = 1$ . This yields  $\delta = \omega$ . In the next step the parametrization is changed to  $\kappa := \text{var}(z)$ , and thus  $\sqrt{\kappa} = 1/\omega$ . Including a new drift correction term  $\gamma t$ , the process for the log-price in the NIG model has the form

$$X_t = \gamma t + \theta s_t + \sigma W_{s_t}$$

with infinitesimal drift parameter  $\theta \in \mathbb{R}$  and variance  $\sigma^2 > 0$  of the Brownian motion.

#### Properties

The NIG model can also be seen as a generalization of the Black-Scholes model. Again, in the limit case  $\kappa \rightarrow 0$  the model reduces to the Black-Scholes model. The conditional

---

<sup>24</sup>See [BN97], eq. (2.5).

PDF for the NIG model is given by:<sup>25</sup>

$$f^{X_t|X_0=0}(x) = \tilde{C} e^{Ax} \frac{K_1(\tilde{B}\sqrt{x^2 + t^2\sigma^2/\kappa})}{\sqrt{x^2 + t^2\sigma^2/\kappa}}, \quad (5.11)$$

where

$$A := \frac{\theta}{\sigma^2}, \quad \tilde{B} := \frac{\sqrt{\theta^2 + \sigma^2/\kappa}}{\sigma^2}, \quad \tilde{C} := \frac{t}{\pi} e^{t/\kappa} \sqrt{\frac{\theta^2}{\kappa\sigma^2} + \frac{1}{\kappa^2}},$$

and  $K_1$  denotes the modified Bessel function of the second kind with index 1. The PDF is  $\mathcal{C}^\infty$ -smooth as  $K_1 \in \mathcal{C}^\infty(0, \infty)$ . It can be evaluated efficiently in  $\mathbf{O}(1)$ .<sup>26</sup>

### Martingale condition

The martingale property can be restored similarly as for the VG model. The characteristic exponent of the NIG model is<sup>27</sup>

$$\Psi_{NIG}(u) = i\gamma u + \frac{1}{\kappa} - \frac{1}{\kappa} \sqrt{1 + u^2\sigma^2\kappa - 2i\theta u\kappa}.$$

Inserting  $u = -i$  and using equation (5.2) yields the following risk-neutrality condition:

$$\gamma = r - q - \frac{1}{\kappa} + \frac{1}{\kappa} \sqrt{1 - \sigma^2\kappa - 2\theta\kappa}$$

The required drift parameter  $\gamma$  has a real value only for

$$\theta \leq \frac{1}{2\kappa} - \frac{\sigma^2}{2}. \quad (5.12)$$

Again, this condition can be interpreted economically: If the drift  $\theta$  of the subordinated process is too large, the risk-neutrality cannot be restored by adding a drift term  $\gamma t$ .

### Tail behavior

The modified Bessel  $K_1$  has the following asymptotic behavior:<sup>28</sup>

$$K_1(y) \rightarrow \sqrt{\frac{\pi}{2y}} e^{-y} \quad \text{for } y \rightarrow \infty$$

Inserting this asymptote for argument  $y = \tilde{B}\sqrt{x^2 + t^2\sigma^2/\kappa}$  into (5.11) yields the tail behavior of the NIG model

$$f(x) \rightarrow \tilde{C} \sqrt{\frac{\pi}{2\tilde{B}(x^2 + t^2\sigma^2/\kappa)^{3/2}}} \exp(Ax - \tilde{B}\sqrt{x^2 + t^2\sigma^2/\kappa}) \quad \text{for } x \rightarrow \pm\infty.$$

<sup>25</sup>This result can be found in [CT03], p. 117

<sup>26</sup>Computational aspects concerning the modified Bessel function can be found in [PTVF02], p. 249ff.

<sup>27</sup>The characteristic exponent can be found in [CT03], table 4.5, p. 117. It must be augmented here by the term  $i\gamma u$  as the drift correction term is not included in this table.

<sup>28</sup>This result can be found e.g., in [AS65], 9.7.4 (p. 378).

The asymptotic behavior is dominated by the exponential term. As

$$y = \sqrt{x^2 + t^2\sigma^2/\kappa} \rightarrow |x| \quad \text{for } |x| \rightarrow \infty$$

it follows that

$$f(x) \in \mathbf{O}(e^{Ax - \tilde{B}|x|}) \quad \text{for } |x| \rightarrow \infty.$$

This means that the NIG model has exponential tails with decay rates  $\tilde{B} - A > 0$  for  $x > 0$  and  $\tilde{B} + A > 0$  for  $x < 0$ . (The case  $\tilde{B} = A$  cannot occur as  $\sigma > 0$ .) These tails are heavier than the tails in the Black-Scholes model.<sup>29</sup> It should be noted that the NIG model is able to reproduce different decay rates for negative and positive returns, thus posing the question, how “heavy” the tails can be. In the following a lower bound for the decay rate  $\tilde{B} - A$  is given.

**Lemma 5.7** (Lower bound for NIG tail decay rate of positive returns).

$$\tilde{B} - A > 1$$

*Proof.* The lower bound is a consequence of relation (5.12), which specifies the admissible parameters:

$$\begin{aligned} \tilde{B} - A &= \frac{\sqrt{\theta^2 + \sigma^2/\kappa}}{\sigma^2} - \frac{\theta}{\sigma^2} \geq 1 \\ \Leftrightarrow \sqrt{\theta^2 + \sigma^2/\kappa} - \theta - \sigma^2 &\geq 0 \\ \Leftrightarrow \theta &\leq \frac{1}{2\kappa} - \frac{\sigma^2}{2} \quad (\text{relation (5.12)}) \end{aligned}$$

Again, the last equivalence can be shown by case differentiation of  $\theta + \sigma^2 \geq 0$  and  $\theta + \sigma^2 < 0$ .  $\square$

**Remark 5.8.** *This is the “best” tail behavior that can be expected for a reasonable model, as section 5.1.7 shows. Interestingly, the corresponding lower bound for the left tail decay rate is zero. Because of the limit behavior  $\theta \rightarrow -\infty \Rightarrow \tilde{B} + A \rightarrow 0$ , the left tail can be made arbitrarily “heavy” (while still decaying exponentially). The possibility to model heavier left tails than right tails is an important feature, as empirical results indicate that large negative jumps occur more frequently than large positive jumps.*

### 5.1.6. Other models

The following list contains some prominent models that are not discussed in this work.

<sup>29</sup>The B/S model has infinite tail decay rates; see e.g., (5.14) on p. 104.

- **The Kou model:** The Kou model falls into the class of jump-diffusion models. It has been proposed in [Kou02]. It is closely related to the Merton model but assumes an asymmetric exponential distribution of the jump sizes. The asymmetric exponential distribution introduces an additional model parameter, making calibration more difficult. We omit this model because of its structural similarity to the Merton model and as there is no closed form of the probability density function  $f$  available in current literature.
- **The Heston model:** This stochastic volatility model was introduced in [Hes93]. It can be used with quadrature method as a two-dimensional model with variables  $(S_t, \sigma_t)$ . The Ornstein-Uhlenbeck process for the volatility exhibits mean reversion and is therefore not a Lévy process (its increments are stochastically dependent). This is not a problem for quadrature methods but for the convolution approach, since the assumption of space-homogeneity fails. The long-term behavior of the Heston model can be described as a one-dimensional model (see [DY02]), but this is a different model which may not be mixed up with the original stochastic volatility model.
- **The CGMY model** has been proposed by Carr et al. in [CGMY02]. It is a generalization of the VG model allowing jumps of both finite and infinite activity.
- **The FMLS model:** The Finite Moment Log Stable (FMLS) process is introduced by Carr and Wu in [CW03]. It assumes a self-similar Lévy process for the asset price. The authors motivate this model by empirical observations for U.S. equity index options. As the proposed process is a Lévy  $\alpha$ -stable process, the conditional return distribution does *not* converge to a normal distribution. In principle, this model can be used with quadrature methods.

**Remark 5.9** (Possible extension: specialized models). *This work only uses models that are available from current literature. For practical applications these models are not necessarily the best choice for quadrature methods. It could be useful to employ specialized asset price models designed to facilitate numerical quadrature or even allow analytic solutions of the arising integrals. When designing such models, it seems important to retain the property of infinite divisibility, which is also given for any Lévy process. Constraints for any density function candidate would be:*

(i) Normalization:  $\|f^{X_t|X_0}\|_{\mathcal{L}^1} = 1$

(ii) Risk-neutrality:  $\|\exp \cdot f^{X_t|X_0}\|_{\mathcal{L}^1} = 1$

(iii) Infinite divisibility:

$$\mathcal{F}(f^{X_t|X_0}) = \mathcal{F}(f^{X_t|X_{(n-1)t/n}}) \cdot \mathcal{F}(f^{X_{(n-1)t/n}|X_{(n-2)t/n}}) \cdot \dots \cdot \mathcal{F}(f^{X_{t/n}|X_0}) \quad (\forall n).$$

### 5.1.7. Summary of tail behavior

Because of the truncation of the computational domain, the tail behavior of the conditional density functions is important for the accuracy of both the spline and RBF method. This section summarizes the tail behavior of the introduced models.

The lowest possible exponential decay rate for the right tail of log-returns  $X_t$  is one for any reasonable model. Otherwise the expectation value  $\mathbb{E}(S_t)$  would be infinite:

Assuming for notational convenience  $S_0 = 1$ , then  $S$  is written for the relative price  $S_t/S_0$ , and  $x$  for the log-return  $\log(S_t/S_0)$ . The transformation theorem leads to the following relation between the density function  $f_{(S)}$  of  $S$  and the density function  $f_{(x)}$  of  $x := \log(S)$ :

$$f_{(x)}(x) = f_{(S)}(e^x)e^x$$

Consequently, the expectation value of  $S$  can be written as

$$\mathbb{E}(S) = \mathbb{E}(e^x) = \int_{-\infty}^{\infty} e^x f_{(x)}(x) dx = \int_{-\infty}^{\infty} e^{2x} f_{(S)}(e^x) dx = \int_0^{\infty} S f_{(S)}(S) dS.$$

Assuming that  $f$  is strictly monotone for  $x \gg 0$ , a necessary condition for this improper integral to exist is

$$f_{(x)}(x) = \mathbf{O}(e^{-x}) \text{ for } x \rightarrow \infty, \text{ or } f_{(S)}(S) = \mathbf{O}(S^{-2}) \text{ for } S \rightarrow \infty, \text{ respectively.} \quad (5.13)$$

Therefore the right tail of the log-return density decays exponentially with a decay rate of at least one in any model with finite expectation of  $S_t$ . For the models introduced in this chapter the decay rate is even strictly greater than one:

**Lemma 5.10** (Common lower bound for exponential decay rates). *The probability density  $f(x)$  of log-returns  $x$  for each of the models {Black-Scholes, VG, and Merton} decays at least exponentially for  $|x| \rightarrow \infty$  with decay rate  $\alpha > 1$ , i.e.,*

$$f(x) = \mathbf{O}(e^{-\alpha|x|}) \text{ for } |x| \rightarrow \infty \text{ for an } \alpha > 1.$$

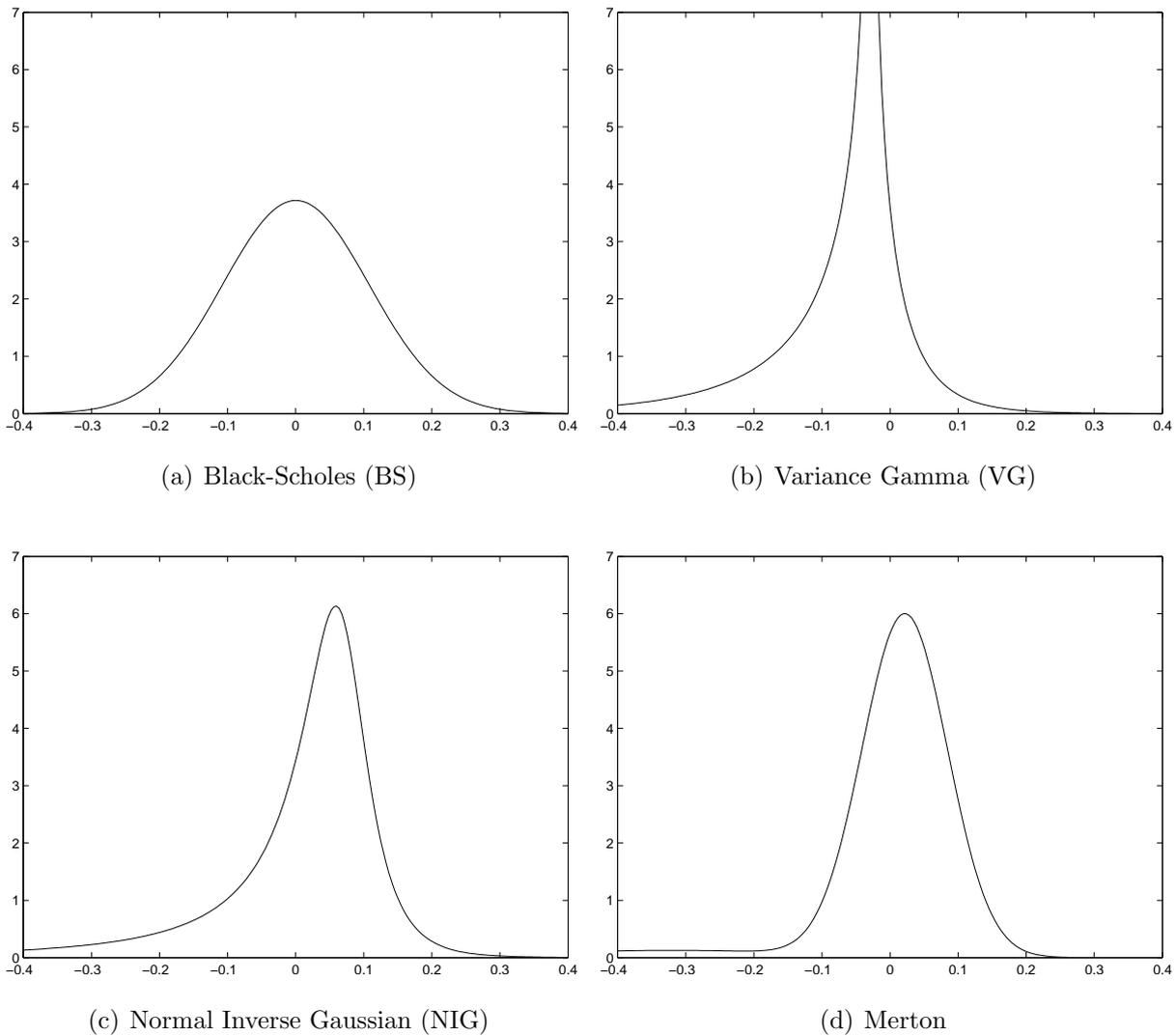
*For the NIG model the tails decay for  $x \rightarrow +\infty$  with decay rate  $\alpha > 1$  and for  $x \rightarrow -\infty$  with decay rate  $\alpha > 0$ .*

*Proof.* For the Black-Scholes model we have for  $|x| \rightarrow \infty$

$$f_{(x)}(x) \stackrel{(5.3)}{=} \mathbf{O}(e^{-\alpha x^2}) \text{ for a } \alpha > 0 \Rightarrow f_{(x)}(x) = \mathbf{O}(e^{-\tilde{\alpha}|x|}) \text{ for all } \tilde{\alpha} > 0, \quad (5.14)$$

particularly for any  $\tilde{\alpha} > 1$ . VG: The decay rate is at least  $B - A$ , and lemma 5.6 ensures  $B - A > 1$ . NIG: The positive tail decay rate is  $\tilde{B} - A$ , and  $\tilde{B} - A > 1$  has been shown in lemma 5.7. For the Merton model the result has already been shown in lemma 5.4.  $\square$

Figure 5.1.: Probability densities  $f(x)$  of log-returns for different models. The parameter values (table 5.1) are close to typical calibration results.



### Tail index

To indicate empirical tail behavior the notion of a *tail index* is introduced in the following.

**Definition 5.11.** *The tail index  $\iota$  of a random variable  $S$  is defined as*

$$\iota := \sup(q \in \mathbb{R}_+ \mid \mathbb{E}(|S|^q) < \infty).$$

parameter	value	parameter	value
$\Delta t$	0.2	$\sigma_{NIG}$	0.24
$r$	0.03	$\theta_{NIG}$	-0.38
$\sigma_{BS}$	0.24	$\kappa_{NIG}$	0.62
$\mu_{BS}$	0.03	$\sigma_{Merton}$	0.14
$\sigma_{VG}$	0.2	$\lambda_{Merton}$	0.32
$\theta_{VG}$	-0.2	$\mu_{Merton}$	-0.34
$\kappa_{VG}$	0.5	$\delta_{Merton}$	0.18

Table 5.1.: Parameter values used for illustration 5.1.

Empirical studies indicate a tail index  $\iota$  of certain stock returns between 2.5 and 5.<sup>30</sup> The tail index  $\iota$  is related to the tail decay rate  $\alpha$  as follows. The tail behavior  $f_{(S)}(S) = \mathbf{O}(S^{-(1+\alpha)})$  or  $f_{(x)}(x) = \mathbf{O}(e^{-\alpha x})$ , respectively, leads to a tail index  $\iota = \alpha + 1$ . Following equation (5.14) the Black-Scholes model has tail index  $\iota = \infty$ , which is a long way from empirical evidence. Other models in this chapter, e.g., the NIG model, are capable to reproduce finite tail indices.<sup>31</sup>

## 5.2. Multi-factor models

For quadrature methods the specification of models for several underlyings corresponds to the specification of the joint conditional probability density function of their prices. It is important that the joint density function is sufficiently smooth and can be evaluated efficiently. How this function is constructed is secondary from the numerical point of view. In this section the multi-dimensional Black-Scholes model is mentioned explicitly. Further, the notion of copulae is introduced to illustrate how multivariate models can be constructed out of one-dimensional models.

### 5.2.1. Multi-factor Black-Scholes

A Black-Scholes model for several underlyings can be used with quadrature methods by “inserting” the multivariate normal density as conditional PDF for the log-price. Using the vector notation  $X_t := (X_1(t), \dots, X_n(t))$  for a process with  $n$  underlyings, the multivariate normal density is

$$f^{X_t|X_0=0}(x) := \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(x - \mu)^{tr} \Sigma^{-1} (x - \mu)\right), \quad (x \in \mathbb{R}^n)$$

where  $\mu = (\mu_1, \dots, \mu_n)^{tr}$  denotes the mean vector and  $\Sigma \in \mathcal{M}^{n \times n}$  denotes the (positive definite) covariance matrix. As in the one-dimensional case, this density function is

<sup>30</sup>This result can be found in [Pag96] for intraday returns.

<sup>31</sup>The NIG model has decay rate  $\lambda = \tilde{B} - A$  and thus tail index  $\iota = \tilde{B} - A + 1 < \infty$ .

optimally suited for the application with quadrature methods: It is  $\mathcal{C}^\infty(\mathbb{R}^n)$ -smooth, can be evaluated efficiently, and all integrals arising in the RBF method can be solved analytically.<sup>32</sup> The martingale condition and tail behavior are analogous to the one-dimensional case.

### 5.2.2. Copulae

The *copula* of a multivariate distribution characterizes the dependence structure between the different random variables. It does not depend on the marginal distributions. An introduction to copulae can be found in [Nel98].

**Definition 5.12** (Copula). *A copula is a multivariate cumulative distribution function defined on  $[0, 1]^n$  such that every marginal distribution is uniform on the interval  $[0, 1]$ .*

Important to the theory of copulae is the following theorem which guarantees their existence and uniqueness for continuous marginal distributions.

**Theorem 5.13** (Sklar's theorem). *Let  $F$  be a distribution function with margins  $F_1, \dots, F_n$ . Then there exists a copula  $C : [0, 1]^n \rightarrow [0, 1]$  such that for all  $x \in \mathbb{R}^n$*

$$F(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)). \quad (5.15)$$

*If all margins  $F_1, \dots, F_n$  are continuous then  $C$  is unique. Conversely, if  $C$  is a copula and  $F_1, \dots, F_n$  are distribution functions, then the function  $F$  defined by (5.15) is an  $n$ -dimensional distribution function with margins  $F_1, \dots, F_n$ .*

**Example 5.14.** *For  $n = 2$  the bivariate Gaussian copula is defined as*

$$C_\rho(x_1, x_2) = N_{2,\rho}(N^{-1}(x_1), N^{-1}(x_2)),$$

*where  $N_{2,\rho}$  denotes the bivariate normal distribution function with correlation  $\rho$  and  $N$  denotes the univariate standard normal distribution function.*

This indicates that a multi-dimensional distribution can be constructed out of one-dimensional distributions by specifying a copula  $C$ . Unfortunately, this construction leaves the class of Lévy processes as the resulting distribution is not necessarily infinitely divisible.<sup>33</sup>

For quadrature methods, however, this is not a problem. Under the assumption of equidistant time steps, the multivariate process described by the conditional density of  $(X_1(\Delta t), \dots, X_n(\Delta t))$  can be calibrated to market prices and used for risk-neutral option valuation.

<sup>32</sup>Despite the availability of an analytic solution, the model matrices for all multi-dimensional examples in chapter 6 have been computed using numerical quadrature.

<sup>33</sup>An approach to utilize the notion of copulae for Lévy processes is discussed in [CT03], §§5.5, 5.6.

### 5.3. Non-parametric modeling

The density function used for quadrature methods must not necessarily be given in a parametric form. For arbitrary underlyings a density estimate can be obtained by the method of *kernel density estimation* proposed in [Par62]. This method constructs a density estimate  $\hat{f}$  from a finite sample of realizations of a random variable.

**Definition 5.15** (Kernel density estimate). *For a sample  $x_1, \dots, x_n \in \mathbb{R}$  of a random variable  $X$ , the kernel density approximation of the probability density function of  $X$  is*

$$\hat{f}(x) := \frac{1}{n} \sum_{i=1}^n \frac{1}{\lambda_i} K\left(\frac{x - x_i}{\lambda_i}\right),$$

where  $K$  is some kernel function and  $\lambda_i > 0$  are smoothing parameters (“bandwidths”). A typical choice of  $K$  is the Gaussian kernel

$$K(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

The method is mentioned here to indicate possible applications of quadrature methods that are inaccessible to other valuation methods. A detailed discussion of kernel density estimation techniques and their convergence properties would go beyond the scope of this work and can be found in the literature.

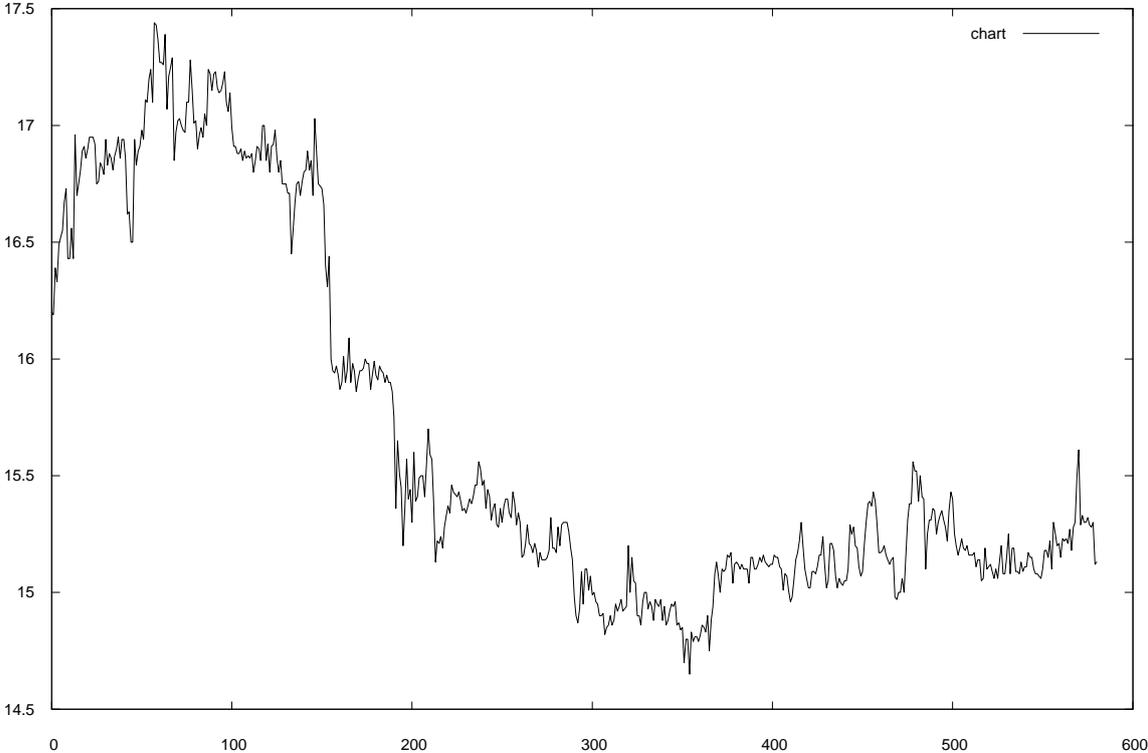
Figure 5.2 illustrates a density estimate obtained from historical prices of “Deutsche Post AG” shares.<sup>34</sup> The density estimates for the first and second half of the time series are plotted separately (“estimate 1” and “estimate 2”) to illustrate the time dependence of the estimate. For this example variable bandwidths  $\lambda_i$  depending on the distance between neighboring sample points have been used. The bandwidths included in the plot are scaled by  $c := \max(\hat{f}(x)) / \max(\lambda_i)$ . The clustering of data points is caused by the decimal notation of stock prices. It requires to limit the bandwidth from below. The resulting density estimate could be used to value fairly arbitrary options for the underlying, including, e.g., American barrier options.

The valuation of such options based on density estimation was restricted to Monte Carlo methods up to now. This possible application shows the flexibility of the quadrature approach quite plainly.

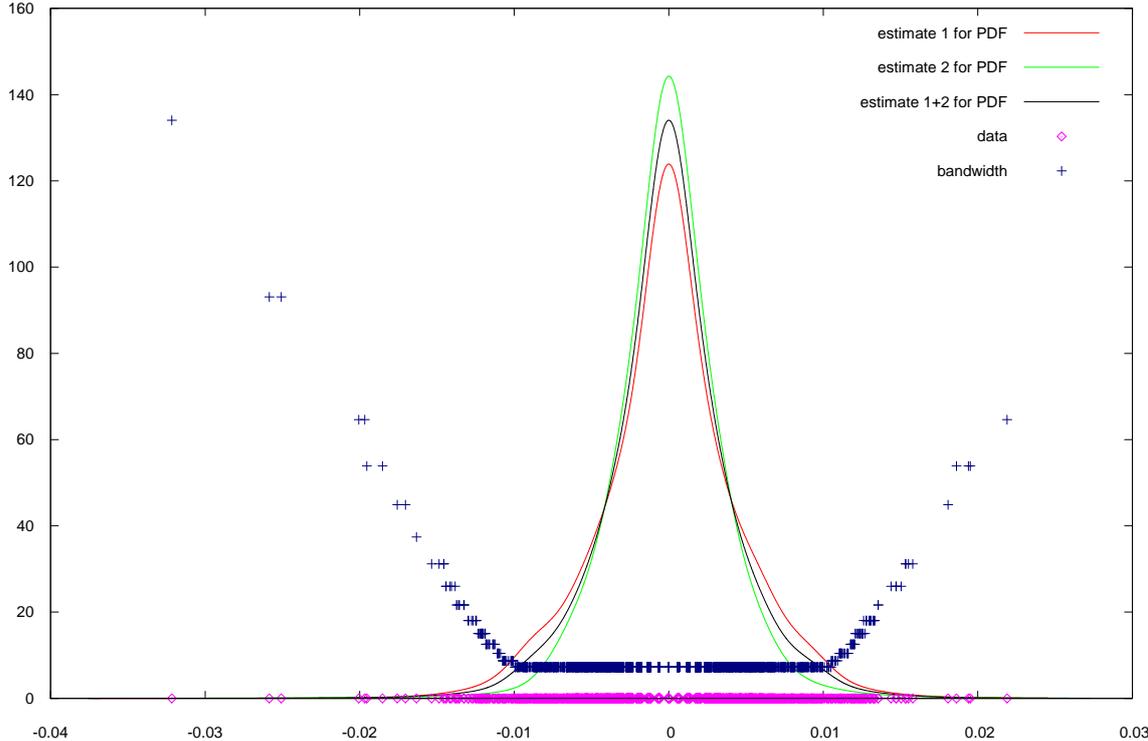
**Remark 5.16.** *Although the time series in figure 5.2 is taken from stock quotes, this is only meant to be an example for any other observable underlying. The time series does not need to represent market prices. It could be, e.g., temperatures, economic variables, error rates, etc.*

<sup>34</sup>Source: <http://www.boersen-zeitung.de>

Figure 5.2.: Example for a non-parametric density estimate from historical prices.



(a) Historical prices of “Deutsche Post AG” 01/2002 - 06/2002



(b) Estimated conditional PDF of daily returns for the first and second half of the time series

## 6. Numerical examples

The purpose of this chapter is threefold. First, it provides a systematic validation of the methods proposed in chapters 3 and 4. Numerical experiments in section 6.1 show that the implementations are correct in the sense that they can reproduce independently obtained results. Second, this chapter illustrates the broad range of possible applications by a collection of examples in sections 6.2 (single-asset options) and 6.3 (multi-asset options). Section 6.4 contains comparisons of the spline and RBF method to other methods. Finally, section 6.5 discusses applications which are not covered by the given examples.

Unfortunately, one important advantage of quadrature methods does not become evident in this chapter. It is the possibility to change the model of the underlying asset by just modifying a single subroutine, namely, the evaluation of the probability density function. This allows a direct application of the spline and RBF method to models with jumps, which leads to a better calibration to market data. As this work does not contain calibration experiments, this advantage cannot not be illustrated clearly. Nevertheless, an example for models with jumps is included in section 6.2.4.

### 6.1. Validation

This section validates the methods in an experimental sense. We choose problems for which highly accurate reference solutions can be provided by other (independently implemented) numerical methods or by analytical solutions and then compare the corresponding solutions of the spline or RBF method to them.

#### 6.1.1. European plain-vanilla put under B/S model

Consider the following European option under a Black-Scholes model:

Option:		Model:	
type	plain-vanilla put	class	GBM
exercise	European	volatility	$\sigma = 0.3$
payoff	$g(S) = (K - S)^+$	return	$r = 0.1$
maturity	$T = 1$	dividend	$q = 0$
strike	$K = 1$		

It represents the simplest type of valuation problem. The payoff is of type “plain-vanilla put”, the exercise right is European, and the model for the underlying security is a Geometric Brownian motion (GBM). For this case the option value solves the Black-Scholes PDE and an “analytic” solution is available: the “Black-Scholes formula”.<sup>1</sup>

Table 6.1.: Results of different methods for a European plain-vanilla put

method	discretization	result $V(1)$	absolute error	CPU time
spline	$n = 140$ (tol. $10^{-5}$ )	0.07217906735137	$3.13 \cdot 10^{-7}$	< 0.01s
spline	$n = 259$ (tol. $10^{-7}$ )	0.07217876221641	$8.36 \cdot 10^{-9}$	< 0.01s
spline	$n = 1037$ (tol. $10^{-10}$ )	0.07217875388664	$2.68 \cdot 10^{-11}$	0.04s
spline	$n = 1695$ (tol. $10^{-11}$ )	0.07217875386158	$1.76 \cdot 10^{-12}$	0.08s
RBF	$n = 100$ (log-eqd.)	0.07182316336727	$3.56 \cdot 10^{-4}$	0.26s
RBF	$n = 1000$ (log-eqd.)	0.07217520417552	$3.55 \cdot 10^{-6}$	0.32s
RBF	$n = 627$ (adaptive)	0.07217875105209	$2.81 \cdot 10^{-9}$	1.97s
RBF	$n = 1473$ (adaptive)	0.07217875294318	$9.17 \cdot 10^{-10}$	5.96s
“analytic”	n/a	0.07217875385982	0	< 0.01s

This first practical test validates the implementation of the methods for the single-asset Black-Scholes case: The results and computation times<sup>2</sup> for different discretizations are given in table 6.1. They obviously approximate the exact solution. Parts of the algorithms which are not covered by this validation are the exercise strategy for Bermudan/American options, the implementation of alternative (non-Black-Scholes) densities from chapter 5, as well as the implementation for multi-dimensional options. These parts are validated in the following, starting with a simple option with American exercise rights.

### 6.1.2. Non-European exercise rights

The example specified in table 6.2 for an American plain-vanilla put is taken from [IV05] (p. 109, example 8.3.2). It is used to validate the exercise strategy for non-European options.

As there is no analytical solution for this problem, the exact solution cannot be obtained as a reference solution. Therefore we use a numerical solution with a very fine

<sup>1</sup>The PDE and its solution can be found, e.g., in [Sey02], §A.3. The solution involves the error function “erf” and has to be evaluated numerically. This has been done here with Maple V.

<sup>2</sup>All computations have been performed on the same CPU, an Intel T2600 with 2.16 GHz. This applies to all CPU times given in this chapter. In table 6.1 the given CPU time for the spline and RBF method is the time needed to compute  $V(S)$  for *all*  $S \in [0.01, 5]$ , while the analytic solution has only been evaluated at  $S = 1$ .

Table 6.2.: American option.

Option:		Model:	
type	plain-vanilla put	class	GBM
exercise	American	volatility	$\sigma = 0.3$
payoff	$g(S) = (K - S)^+$	return	$r = 0.1$
maturity	$T = 1$	dividend	$q = 0$
strike	$K = 1$		

discretization as “exact” reference solution and calculate the absolute error with respect to that solution.

Figure 6.1 shows the absolute error of the RBF method for two different discretizations:  $n = 1000$  and  $n = 2000$  log-equidistant nodes for the space discretization and  $m = 10000$  and  $m = 20000$  time steps, respectively. Figure 6.2 shows the absolute error of the binomial tree method<sup>3</sup> for American options with  $n = 50000$  steps in both space and time. For these discretizations the errors of both methods are similar. Both errors peak at about  $10^{-6}$  near the exercise boundary. The oscillating errors in the exercise region of the option are caused by the truncation of small entries in the model matrix. They could be eliminated by using a lower cutoff threshold, but this is not considered a problem as in the exercise region the value of the option is its payoff. On the other hand the undesirable oscillatory behavior of the binomial tree solution in the hold region is more problematic. It makes extrapolation techniques unfeasible for the binomial tree method.

Table 6.3 shows the results and CPU times needed to compute 500 option values  $V(S)$  for  $S \in [e^{-2}, e^2]$  with different methods. The column “absolute error” reflects the difference to the result of a binomial tree method with a high number of time steps. The column “tolerance” contains the corresponding input parameter for the adaptive spline method.

In case of the binomial method the tree has been rebuilt for each option value. The computation times for the binomial method are given only for the sake of completeness and not as a reference for comparison. An accurate reference solution could have been obtained also by finite difference or finite element methods in a fraction of the time needed for 500 binomial trees.

The results for the spline method indicate that this method is not suitable for an approximation of American options prices. It is efficient only for Bermudan options with a modest number of exercise times, e.g.,  $m = 100$ . The reason is that the computational costs of the spline method are proportional to the number of exercise times. Each step requires the approximation of  $\mathbf{O}(n)$  integrals. Although the RBF method is also linear in the number of exercise times, it requires only the solution of a linear system in each

<sup>3</sup>The binomial tree method is introduced in [CRR79].

time step. All integrations are performed in the setup of the model matrix.

Table 6.3.: Results and computation times for the American option specified in table 6.2

method	discretization	tol.	result $V(1)$	abs. err.	CPU time
spline	$293 \times 100$	$10^{-5}$	<u>0.08325389905</u>	$1.2 \cdot 10^{-4}$	2.0s
spline	$1164 \times 100$	$10^{-6}$	<u>0.08325997100</u>	$1.2 \cdot 10^{-4}$	21.7s
spline	$1903 \times 100$	$10^{-7}$	<u>0.08325997105</u>	$1.2 \cdot 10^{-4}$	55.0s
spline	$311 \times 300$	$10^{-5}$	<u>0.08334177003</u>	$3.5 \cdot 10^{-5}$	2.5s
spline	$543 \times 500$	$10^{-5}$	<u>0.08335334251</u>	$2.3 \cdot 10^{-5}$	19.4s
spline	$1003 \times 1000$	$10^{-5}$	<u>0.08336509932</u>	$1.2 \cdot 10^{-5}$	56.2s
RBF	$1000 \times 1000$	n/a	<u>0.08336424416</u>	$1.3 \cdot 10^{-5}$	2.5s
RBF	$1000 \times 10000$	n/a	<u>0.08337422258</u>	$2.7 \cdot 10^{-6}$	15.3s
RBF	$2000 \times 20000$	n/a	<u>0.08337580976</u>	$1.1 \cdot 10^{-6}$	65.5s
binomial tree	$50000^2$	n/a	<u>0.08337690822</u>	n/a	4312.5s
binomial tree	$100000^2$	n/a	<u>0.08337688124</u>	n/a	17210.6s
binomial tree	$200000^2$	n/a	<u>0.08337686754</u>	n/a	93961.3s

Figure 6.1.: Absolute error estimates for the RBF method with  $n = 1000$  and  $n = 2000$

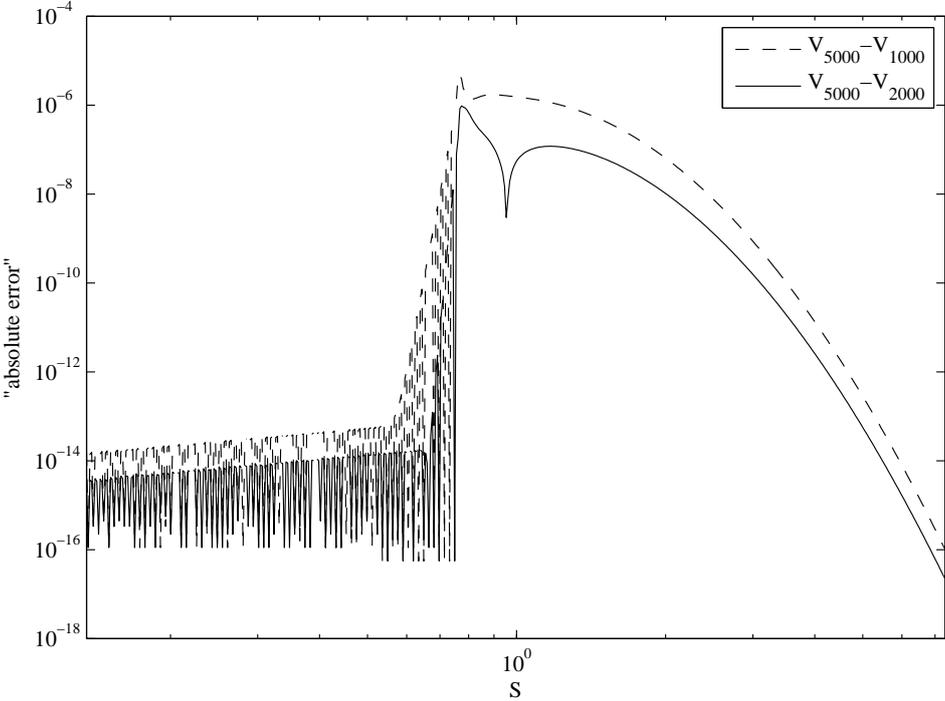
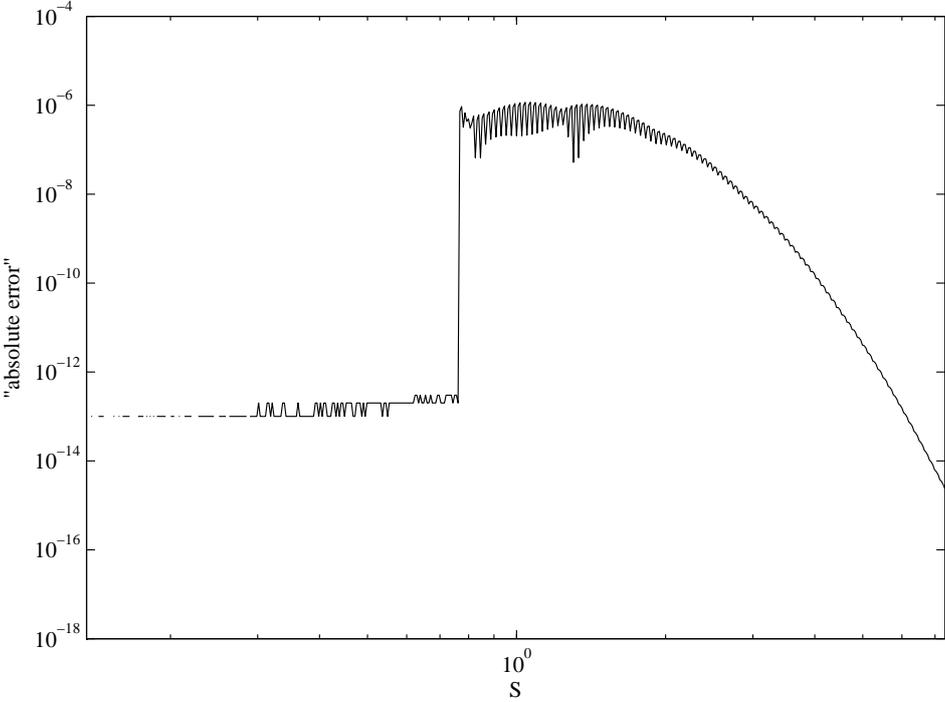


Figure 6.2.: Absolute error estimate for the binomial tree method with  $n = 50000$



### 6.1.3. Non-Black-Scholes models

This section validates the implementation of the non-Black-Scholes models from chapter 5. It starts with a verification of the put-call parity and normalization of the probability density functions. It proceeds with a comparison of results for an explicit example for the Merton model to independent results. Although the validation is done only for European options with plain-vanilla payoffs, this validation in connection with section 6.1.2 implies the validation for non-European options, too, since the implementation of the exercise strategy does not depend on the model for the underlying.

#### Test of put-call parity

A (partial) validation which does not require reference values is based on the put-call parity for European plain-vanilla options.

**Lemma 6.1** (Put-call parity). *For European options under any market model that excludes arbitrage opportunities, the following relation between plain-vanilla put and call prices holds (see, e.g., [Wil98], section 2.12):*

$$V_{call}(S) = V_{put}(S) + S - Ke^{-rT} \quad (6.1)$$

This relation between puts and calls can be derived using purely no-arbitrage arguments and thus holds for any reasonable market model. It can be used for a pre-validation of the implementation for non-Black-Scholes models without requiring any external reference values. Some results are given in table 6.4. The values  $V_{call}(S)$  and  $V_{put}(S)$  have been computed in separate runs of the valuation method. The model parameters are not included as the put-call parity does not depend on parameter values. All values have been computed with the RBF method with  $n = 2000$  log-equidistant nodes. The spline method would lead to similar results. (Actually both methods use the same quadrature code and the same implementation for the density functions.) The small residuals confirm that the put-call parity (6.1) holds for the resulting option prices. This indicates in particular that the drift correction terms are correct.

Table 6.4.: Verification of put-call parity for the implementations of different models.

model class	$S$	$K$	$rT$	$V_{call}(S)$	$V_{put}(S)$	residual of eq. (6.1)
B/S	1000	1000	0.006	45.72310	39.74107	$1.2107 \cdot 10^{-11}$
Merton	1000	1000	0.006	52.90871	46.92667	$9.3365 \cdot 10^{-12}$
Merton	800	1000	0.006	0.76225	194.78021	$6.1555 \cdot 10^{-10}$
VG	1000	1000	0.006	37.32835	31.34631	$1.0105 \cdot 10^{-5}$
NIG	1000	1000	0.006	49.39185	43.40981	$2.8377 \cdot 10^{-8}$

**Remark 6.2** (Computational domains for models with jumps). *The first tests with non-Black-Scholes models indicated that the results were far less accurate than for the Black-Scholes model. The reason is that for the first tests the same computational domains as for the B/S model had been used. After adjusting the computational domains the accuracy was of the same order as for B/S. Although apparent it seems noteworthy that models with jumps require larger computational domains. The reason is the larger influence between distant regions compared to models where jumps do not occur.*

### Test of normalization

The value of an option with payoff  $g \equiv 1$  under any model is  $V \equiv e^{-rT}$ . This option allows a normalization test, which essentially verifies that the  $\mathcal{L}^1$ -norms of the conditional probability density functions are

$$\|f^{X_t|X_0=x}\|_{\mathcal{L}^1} = 1 \text{ for all } x \in \mathbb{R}^d. \quad (6.2)$$

Table 6.5 shows the numerical results of the normalization test using the RBF method for  $r = 0$  and  $x = 0$ .<sup>4</sup> The model parameters used for this test are not included, as the test is independent of these parameters. The discretization was a standard discretization of  $n = 2000$  log-equidistant nodes on the  $S$ -interval  $[1, 10]$ . The results verify the normalization condition (6.2). The larger errors for the VG and NIG norms are probably caused by the implementations of the Gamma function and/or the modified Bessel functions.

Table 6.5.: Norms of the probability density functions for different models.

model class	$\ f\ _{\mathcal{L}^1}$
B/S	1.00000000
Merton	1.00000000
VG	0.99999987
NIG	0.99999919

### Validation of the Merton model

The results of the RBF method for European put and call options on an underlying described by the Merton model are compared to results of the Fast Fourier transform method for European options in [CM99]. The reference results have been obtained from Achim Dahlbokum.<sup>5</sup> The options and the underlying are specified in table 6.6.

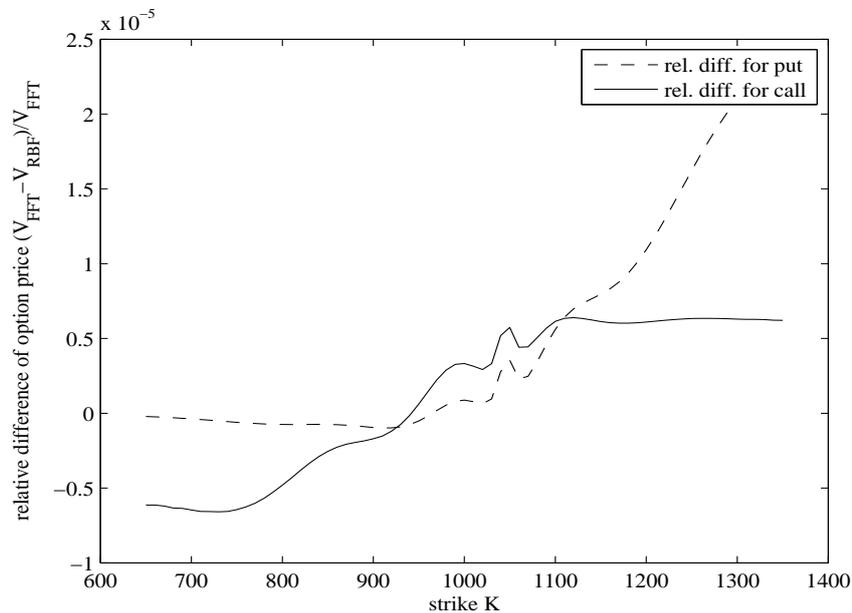
<sup>4</sup>Again, a separate test for the spline method is not required, as the quadrature codes and implementations of the probability density functions are the same as for the RBF method.

<sup>5</sup>Personal communication, November 2006. The model parameters are results of a calibration to market data for certain DAX index options. The calibration has also been performed by Achim Dahlbokum.

Table 6.6.: Test option for the validation of the Merton model.

Put/call option:		Model:	
type	plain-vanilla put	class	Merton (section 5.1.3)
exercise	European	volatility	$\sigma = 0.14$
payoffs	$g_{put}(S) = (K - S)^+$ $g_{call}(S) = (S - K)^+$	return	$r = 0.03$
maturity	$T = 1.5$	jump intensity	$\lambda_M = 0.32$
strikes	$K \in [650, 1350]$	mean jump size	$\mu_M = -0.34$
		std. dev. of jump size	$\delta_M = 0.18$

Figure 6.3 illustrates the relative difference between the RBF solution and the reference values for both options. It is of order  $10^{-5}$ , which is the estimated accuracy of the reference values. As the reference values have been obtained from an independent implementation of a different method, this validates the implementation of the Merton model, i.e., the implementation of the corresponding probability density and proper use of integration techniques.

Figure 6.3.: Relative difference between the RBF and FFT solutions for the price  $V(S)$  of a put and call option evaluated at  $S = 1000$  for different strikes  $K$ .

## Validation of VG and NIG model

We skip the explicit validation of the VG model and NIG model against independent results. Nevertheless they can be regarded as validated by means of the validation of the Merton model. The only difference between the Merton model, VG model, and NIG model are the probability density functions which have been introduced in chapter 5. Although the implementations of the VG and NIG density functions involve the evaluation of the Gamma function and Bessel functions of second kind, both density functions can be evaluated numerically in an efficient way. The tests of the put-call parity and normalization of the density functions exclude virtually any possible errors in the implementation.

### 6.1.4. Options on several underlyings

Options on several underlyings require an additional validation as these types of options involve multi-dimensional quadrature rules. This validation is performed by comparison of the solutions for the two examples in sections 6.3.1 and 6.3.2 to independent results. These results have been obtained by an (independent) implementation of an analytic solution or binomial tree approximation, respectively. Figure 6.4 shows the corresponding screen shots of the software contained in [Hau97].

Figure 6.4.: Screen shots of implementations provided by [Hau97].

	A	B	C	D		A	B	C	
1	<b>Two asset cash-or-nothing options</b>				1	<b>Three dimensional two asset binomial tree</b>			
2					2				
3					3	[3] Options on the minimum of two assets			
4					4				American
5					5				Put
6					6				
7					7	Asset 1 ( $S_1$ )		5,00	
8					8	Asset 2 ( $S_2$ )		5,00	
9					9	Weight asset 1 ( $Q_1$ )		1,00	
10					10	Weight asset 2 ( $Q_2$ )		1,00	
11					11	Strike 1 ( $X_1$ )		5,00	
12					12	Strike 2 ( $X_2$ )		5,00	
13					13	Time to maturity ( $T$ )		1,00	
14					14	Risk-free rate ( $r$ )		10,00%	
15					15	Cost of carry asset 1 ( $b_1$ )		10,00%	
16					16	Cost of carry asset 2 ( $b_2$ )		10,00%	
17					17	Volatility asset 1 ( $\sigma_1$ )		20,00%	
18					18	Volatility asset 2 ( $\sigma_2$ )		30,00%	
19					19	Correlation ( $\rho$ )		0,30	
20					20	Number of time steps ( $n$ )		100	
21					21	Value		0,5218	

(a) Analytic solution for the 2D-binary option from section 6.3.1

(b) Binomial tree solution for the American two-asset rainbow option from section 6.3.2

## 6.2. Single-asset options

### 6.2.1. European binary option

Option:	
type	Binary <sup>6</sup>
exercise	European
payoff	$g(S) = \mathbf{1}_{\{S > K\}}$
maturity	$T = 0.5$
strike	$K = 0.5$

Model:	
class	GBM
volatility	$\sigma = 0.5$
return	$r = 0.1$
dividend	$q = 0$

Method/discretization:	
method	RBF
space discretization	$n = 1000$ , log-eqd.
time discretization	$m = 200$ , eqd.
computational domain	$[0.1, 5]$

Results:	
V(0.5)	<u>0.4622006</u>

This example can be found in the preprint [KVY], section 5.1. The parameters above match the parameters that have been used for figures 4 and 5 in this [KVY] (p. 15). Figure 6.5 shows the evolution of the option price in time.

The results in tables 1 and 2 of the same preprint refer to a *different* option with  $T = 0.25$ ,  $r = 0.05$ , and  $\sigma = 0.2$  (values given on p. 14). The reason for using two different parameter sets may be that the unrealistic high volatility of  $\sigma = 0.5$  leads to a clearer illustration.

### Comparison with analytic solution

The value of this European binary option is known analytically:<sup>7</sup>

$$V(S, t) = e^{-r(T-t)} N(d_2) \quad \text{with} \quad d_2 = \frac{\log(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}},$$

where  $N(\cdot)$  denotes the cumulative standard normal distribution function:

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-\frac{t^2}{2}) dt = \frac{1}{2}(1 + \operatorname{erf}(\frac{x}{\sqrt{2}}))$$

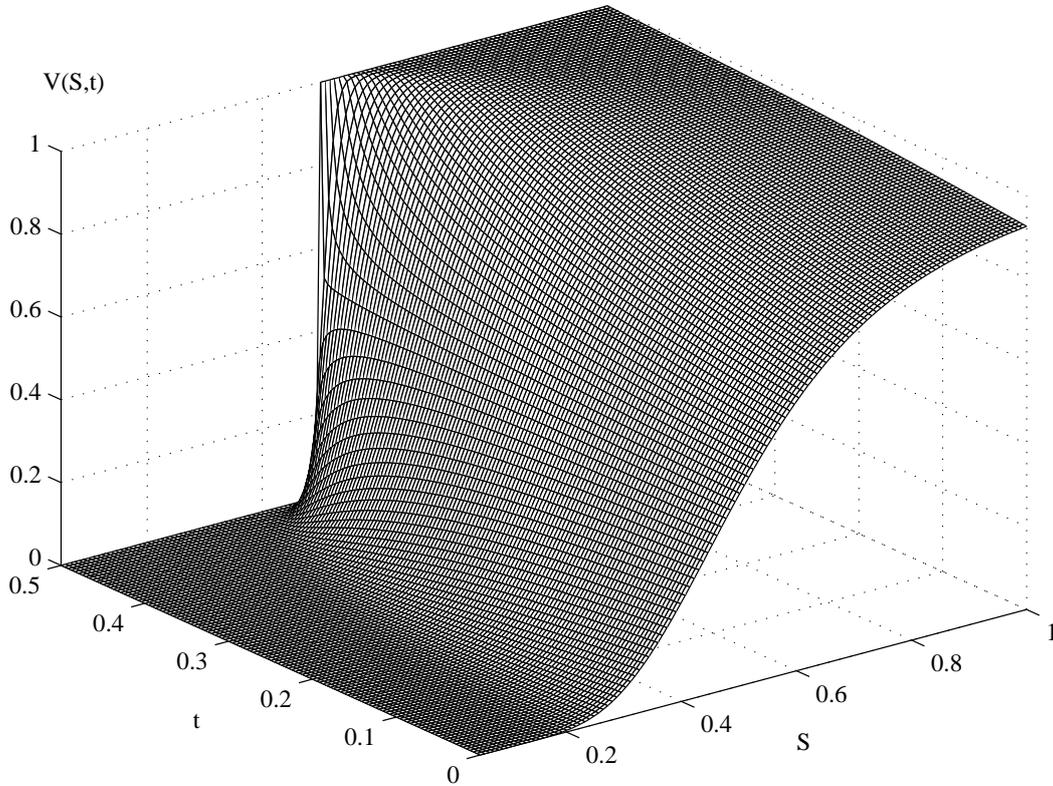
For the parameters ( $T = 0.25$ ,  $r = 0.05$ ,  $\sigma = 0.2$ ) the analytic solution is<sup>8</sup>

$$V(0.5) = 0.52331021191 \quad (\text{all printed digits exact}).$$

<sup>6</sup>An option is called *binary*, if its payoff only takes two different values (here 0/1).

<sup>7</sup>This formula is analogous to the Black-Scholes formula for plain-vanilla puts/calls. Its derivation is straightforward, as the value of this special option corresponds to the probability for  $S(T) \geq K$  at maturity.

<sup>8</sup>Evaluated using “Maple V”.

Figure 6.5.: Evolution in time of the value  $V(S, t)$  of a European binary option.

The RBF method for these parameters with a log-equidistant  $S$ -discretization of 5000 nodes in the interval  $[S_l, S_u] = [0.2, 1]$  results in

$$\tilde{V}(0.5) = \underline{0.52331022462} \text{ (matching digits underlined).}$$

Result for a coarse discretization of 100 log-equidistant nodes on the same interval:

$$\tilde{V}(0.5) = \underline{0.52334208693}.$$

**Remark 6.3** (Placement of nodes). *If the nodes for the RBF method are placed non-adaptive (e.g., log-equidistant), the placement of nodes in the neighborhood of the discontinuity is very important for accuracy. The best approximation is achieved by placing the discontinuity in the middle between two neighboring nodes (experimental experience). This is not an issue, if adaptive discretization is used, as the distance between the nodes in the neighborhood of the discontinuity is many magnitudes smaller than in the non-adaptive case.*

### 6.2.2. American binary option

Option:	
type	Binary
exercise	American
payoff	$g(S) = \mathbf{1}_{\{S > K\}}$
maturity	$T = 0.5$
strike	$K = 0.5$

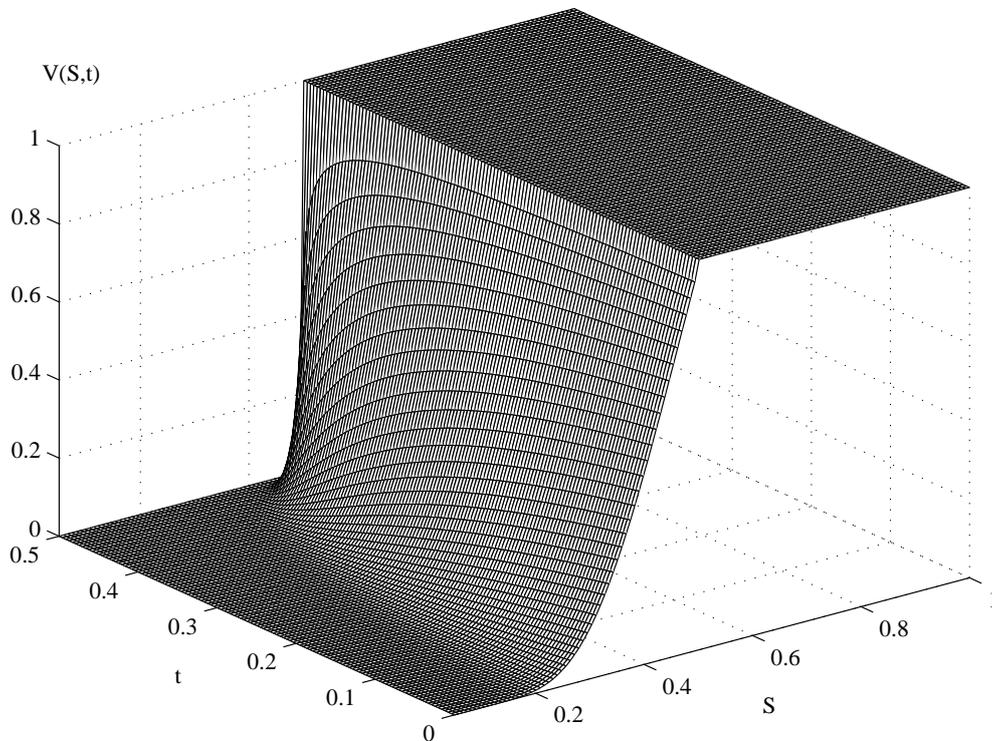
Model:	
class	GBM
volatility	$\sigma = 0.5$
return	$r = 0.1$
dividend	$q = 0$

Method/discretization:	
method	RBF
space discretization	$n = 5000$ , log-eqd.
time discretization	$m = 1000$ , eqd.
computational domain	$[0.05, 1]$

Results:	
$V(0.5)$	<u>1.0000000</u>
$V(0.4)$	0.4944380
$V(0.3)$	0.1318085
$V(0.2)$	0.0079269

This is the same option as in the European case but now with American exercise rights. For this case there is no known analytic solution. The pre-print [KVY] does not provide results for American options, as the Padé schemes are not directly applicable to American options. Figure 6.6 illustrates the results of the RBF method.

Figure 6.6.: Evolution in time of the value  $V(S, t)$  of an American binary option.



### 6.2.3. European double-barrier option

We now come to a more complicated example of a European double-barrier option with two types of barriers. This example is described in the pre-print [KVY]. The outer barriers  $B_1$  and  $B_2$  are of Down-And-Out and Up-And-Out type, respectively. They are continuously monitored. The two inner barriers  $B_3$  and  $B_4$  are discretely monitored and apply at five equidistant times until maturity.

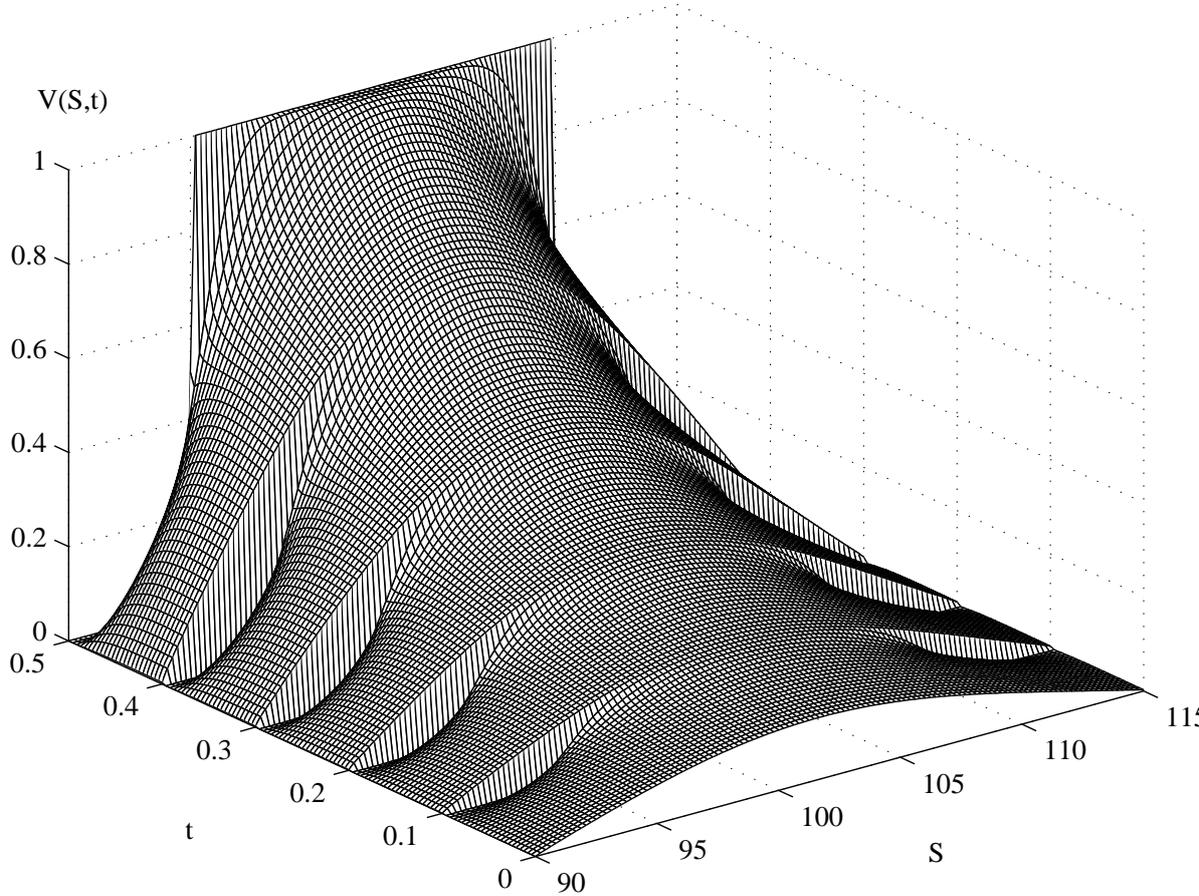
Figure 6.7 shows the evolution of the option value in time. The option for this illustration has been computed with a larger volatility than the result in the table below.

Option:		Model:	
type	“Double-barrier”	class	GBM
exercise	European	volatility	$\sigma = 0.2$
payoff	$g(S) = 1$	return	$r = 0.05$
maturity	$T = 0.5$	dividend	$q = 0$
barriers	continuously monitored: DAO at $B_1 = 90$ , UAO at $B_2 = 115$ . discretely mon. at $t \in \{0.1, 0.2, \dots, 0.5\}$ : DAO at $B_3 = 95$ , UAO at $B_4 = 110$ .		
Method/discretization:		Result:	
method	RBF	V(100)	0.518997
space discretization	$n = 500$ , log-eqd.		
time discretization	$m = 2000$ , eqd.		
computational domain	$[90, 115]$		

**Remark 6.4** (Examples in pre-print [KVY]). *The pre-print [KVY] contains several other examples for European options with different kinds of payoffs in a Black-Scholes setting. All of these options can also be priced by the spline or RBF method. This is not surprising, as the options considered in this pre-print are only “exotic” in the sense of unusual payoffs but not in the sense of path-dependency.*

*Although the pre-print provides results for some of these options, a direct comparison is not possible as the specifications of the corresponding options are not precise enough. For example, for the “double-barrier” option ([KVY], §5.4) the type of payoff (plain-vanilla put/call, binary or other) is not mentioned; it is not specified when the discrete barriers are applied; the apparently continuously applied outer barriers are not mentioned at all.*

Figure 6.7.: Evolution in time of the value  $V(S, t)$  of a European double-barrier option.



### 6.2.4. Options under different market models

This section illustrates the applicability of the proposed methods to a wide range of market models. The same option is valued under different market models from chapter 5. Table 6.7 specifies the test option and three different market models. The results of the two valuation methods are displayed in figure 6.8. It is clearly visible that the option price decays slower (for  $S \rightarrow \infty$ ) for the two models with jumps than for the Black-Scholes model. The decay rates correspond to the tail behavior of the probability density functions of the models. (Compare the left tails on figure 5.1, p. 105.)

Table 6.7.: Specification of the test option and three different market models.

Test option:	
type	plain-vanilla put
exercise	American
payoff	$g(S) = (K - S)^+$
maturity	$T = 1$
strike	$K = 1$

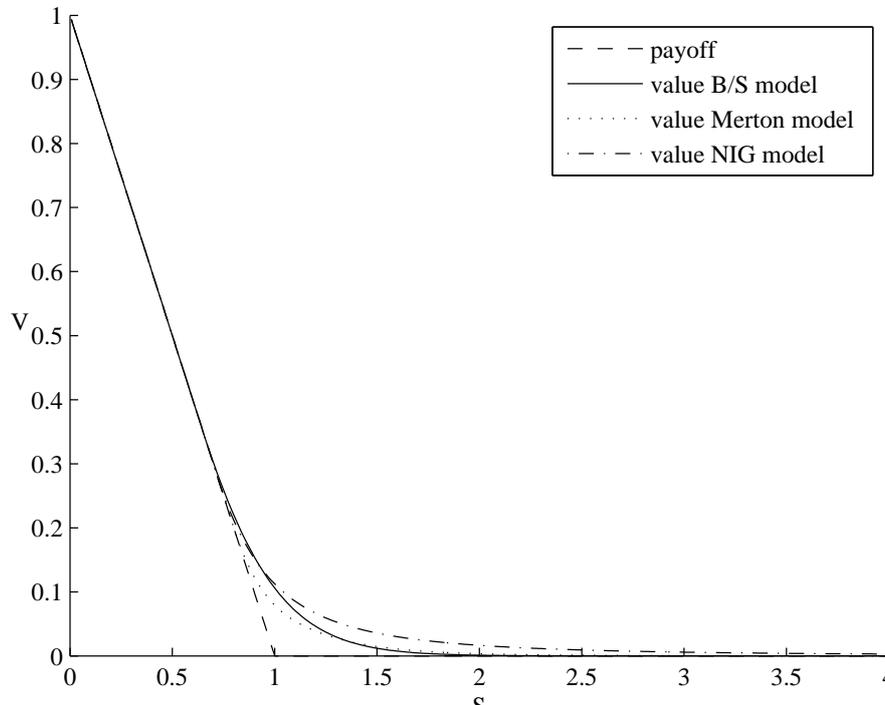
  

GBM model:		Merton model:		NIG model:	
class	GBM	class	Merton	class	NIG
volatility	$\sigma = 0.3$	volatility	$\sigma = 0.14$	volatility	$\sigma = 0.24$
return	$r = 0.03$	return	$r = 0.03$	return	$r = 0.03$
dividend	$q = 0$	dividend	$q = 0$	dividend	$q = 0$
		jump intensity	$\lambda = 0.32$	drift	$\theta = -0.38$
		mean jump size	$\mu = -0.34$	variance of	$\kappa = 0.62$
		std. dev. j. size	$\delta_M = 0.18$	subordinator	

The option values have been computed using the RBF method with a log-equidistant discretization of  $n = 4000$  points on the  $S$ -interval  $[e^{-5}, e^4]$  and  $m = 100$  equidistant exercise times in  $[0, 1]$ . The same test option has been valued by the spline method (using tolerance  $tol = 10^{-5}$  and identical time discretization) and the results are identical for all printed digits in table 6.8. Therefore, an additional illustration for the spline results is not given.

In principle all other examples in this section can be valued as well under the models from chapter 5. However, the computational effort increases by a constant factor as the evaluation of the density functions of the Merton, VG, and NIG model is more expensive than the evaluation of the Black-Scholes density function.<sup>9</sup> Table 6.8 contains

<sup>9</sup>Remember from chapter 5: The density functions of Merton, VG, and NIG involve the approximation of an infinite series, evaluation of the Gamma function, and/or Bessel functions of second kind, respectively. The evaluation of the Black-Scholes density only involves an exponential function.

Figure 6.8.: Option value  $V(S)$  at  $t = 0$  for a test option under different market models.

CPU times to illustrate the additional computational effort. The CPU time for the Black-Scholes model is considerably lower, as for this model the subroutine that builds the model matrix uses an analytical solution of the integrals and thus skips the numerical quadrature.<sup>10</sup>

Table 6.8.: Results and CPU times for different models.

model	$V(1)$	CPU time
B/S	0.10605	3.8s
Merton	0.07924	35.3s
NIG	0.11241	30.5s

Unfortunately, this academic example cannot clearly illustrate the benefits of models with jumps. They would become evident for a calibration to market data, but this is beyond the scope of this work.<sup>11</sup>

<sup>10</sup>Compare section 4.3.2.

<sup>11</sup>For example, a calibration of several Lévy models (including B/S, NIG, and VG) to S&P 500 index option prices is discussed in [Mat05]. (Index options are typically of European style.)

## 6.3. Multi-asset options

This section contains some examples for options on two underlyings. Although it is theoretically possible to value options on three or more underlyings with the RBF method, it turns out to be impracticable for the standard LU decomposition with usual desktop CPU resources.<sup>12</sup>

### 6.3.1. European multi-asset binary option

Option:		Model:	
type	Two-asset binary put	class	GBM with corr.
exercise	European	volatility	$\sigma_1 = 0.2, \sigma_2 = 0.3$
payoff	$g(S_1, S_2) = \mathbf{1}_{\{\max(S_1, S_2) < K\}}$	correlation	$\rho = 0.3$
maturity	$T = 1$	return	$r_1 = r_2 = 0.1$
strike	$K = 5$	dividend	$q_1 = q_2 = 0$

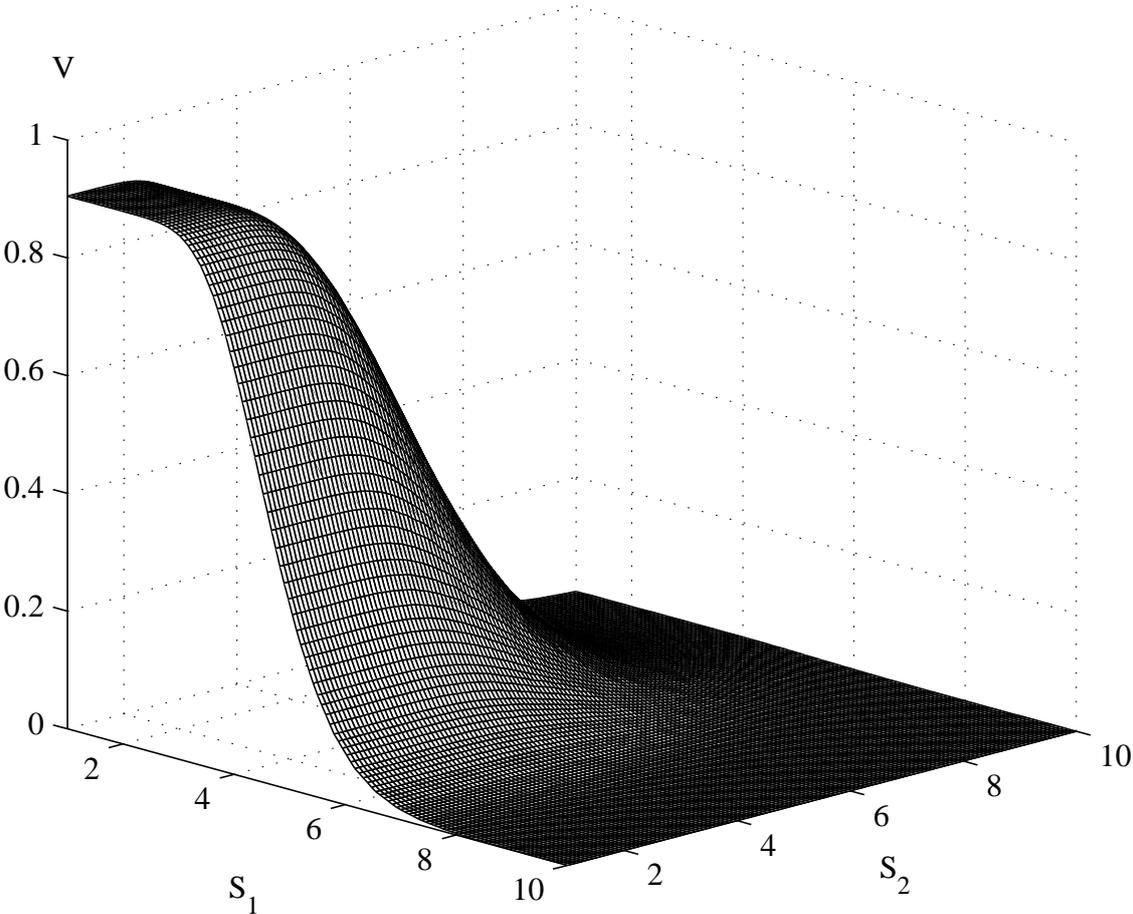
  

Method/discretization:		Results:	
method	RBF	$V(5,5)$	0.1737
space discretization	$n = 50$ , log-eqd.		
time discretization	$m = 1$ (European!)		
computational domain	$[1, 6] \times [1, 6]$		

This example is used for validation purposes only. The result  $V(5, 5) = 0.1737$  of the RBF method can be compared to the analytical solution taken from [Hau97] (cf. figure 6.4):  $V_{Haug}(5, 5) = 0.1734$ . The relative difference is about 0.17%, which corresponds to the expected accuracy for the chosen discretization. Figure 6.9 shows the solution  $V(S_1, S_2)$ .

<sup>12</sup>An alternative could be the use of an incomplete LU decomposition.

Figure 6.9.: Value  $V(S_1, S_2)$  of a European two-asset binary put at  $t = 0$ .



### 6.3.2. American rainbow minimum put

Option:		Model:	
type	Rainbow minimum put	class	GBM with corr.
exercise	American	volatility	$\sigma_1 = 0.2, \sigma_2 = 0.3$
payoff	$g(S_1, S_2) = (K - \min(S_1, S_2))^+$	correlation	$\rho = 0.3$
maturity	$T = 1$	return	$r_1 = r_2 = 0.1$
strike	$K = 5$	dividend	$q_1 = q_2 = 0$

Method/discretization:		Results:	
method	RBF	V(5,5)	0.521123
space discretization	$n = 150, \log\text{-eqd.}$		
time discretization	$m = 150, \text{eqd.}$		
computational domain	$[0.5, 20] \times [0.5, 20]$		

This option can also be found in [IV05], p. 130, where a non-linear programming approach is used to approximate the value of American options. The parameter values are identical.<sup>13</sup> The value  $V(5, 5)$  can be compared to a reference value  $V(5, 5) = 0.522177$  provided by Rainer Int-Veen (personal communication). The relative difference to the value computed by the RBF method is about 0.2%. A second reference value can be obtained by a binomial tree method contained in [Hau97] (cf. figure 6.4):  $V(5, 5) = 0.5218$ . The relative difference to this value is about 0.07%. This corresponds to the expected accuracy for the chosen discretization.

#### Discussion of the error estimate

For an empirical error estimate we use two different discretizations. We assume the results from the fine discretization to be correct. Then we can estimate the error of the results from the coarse discretization. Of course this can only lead to a rough approximation, but it is sufficient to see where the errors are localized and of which magnitude they are. This is done in figure 6.11. The coarse discretization is  $(n_1, m_1) = (50, 50)$ , and the fine discretization is  $(n_2, m_2) = (100, 100)$ . The absolute and relative “errors” are displayed. Apparently the absolute error is localized along the edges of the payoff. Of course, the relative error is also large in the area where the option value is close to zero. The largest error is located at the payoff’s edge, which is parallel to the  $S_2$  axis. This was also expected as  $S_1$  has lower volatility, and therefore this edge is smoothed out more slowly with increasing time to maturity.

We can summarize that the absolute error of  $V_{100}$  is of order  $10^{-3}$  and that the relative error is below 1% in most areas. Qualitatively we have seen that high volatility can reduce the negative impact of non-smooth payoffs on accuracy.

<sup>13</sup>Note: Here  $r_1 = r_2 = 0.1$  is used instead of 0.15 as in [IV05], p. 130. The parameter value 0.1 has been used for the computation of the reference value.

Figure 6.10.: Payoff  $g(S_1, S_2)$  and option value  $V(S_1, S_2)$  at  $t = 0$ .

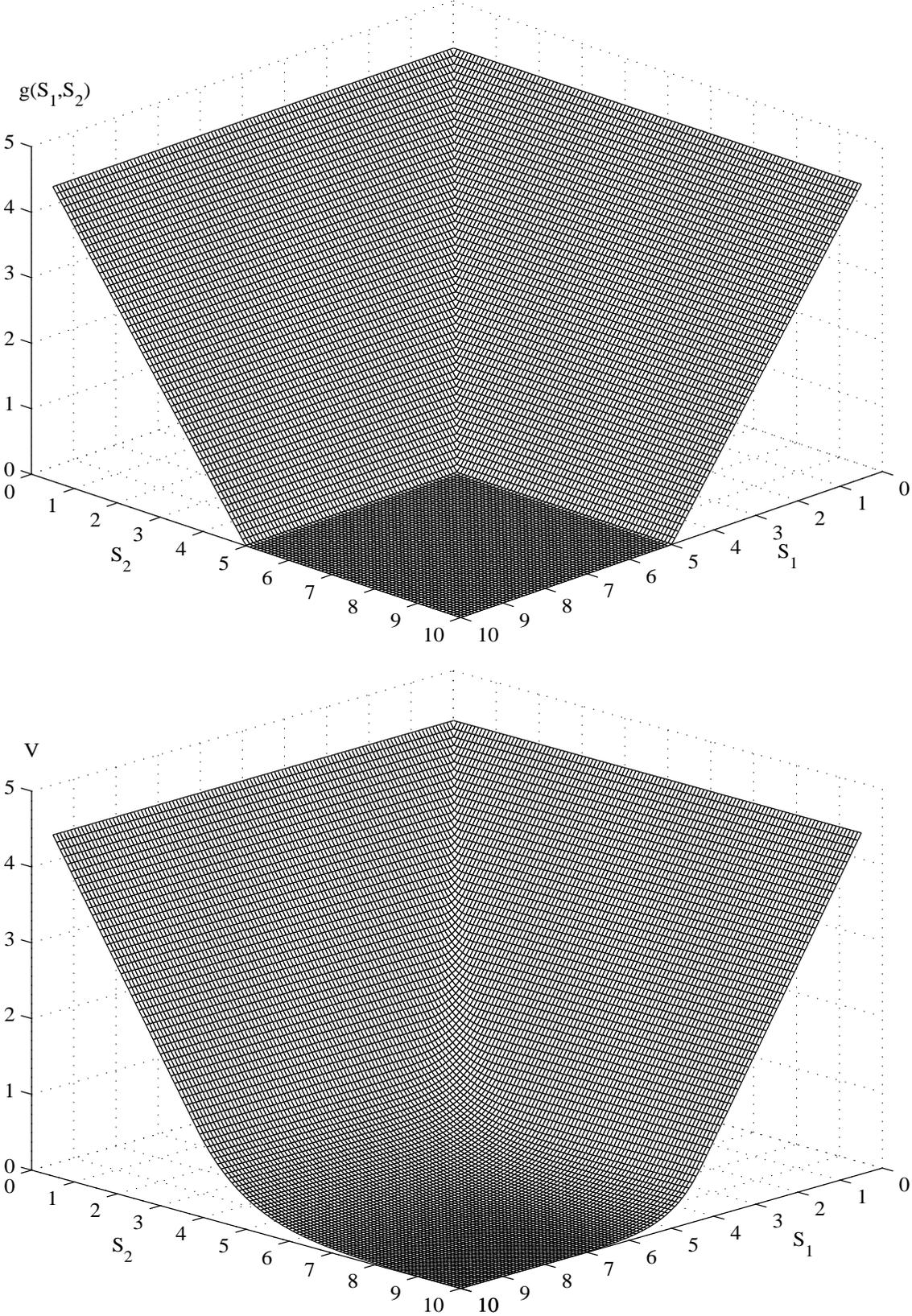
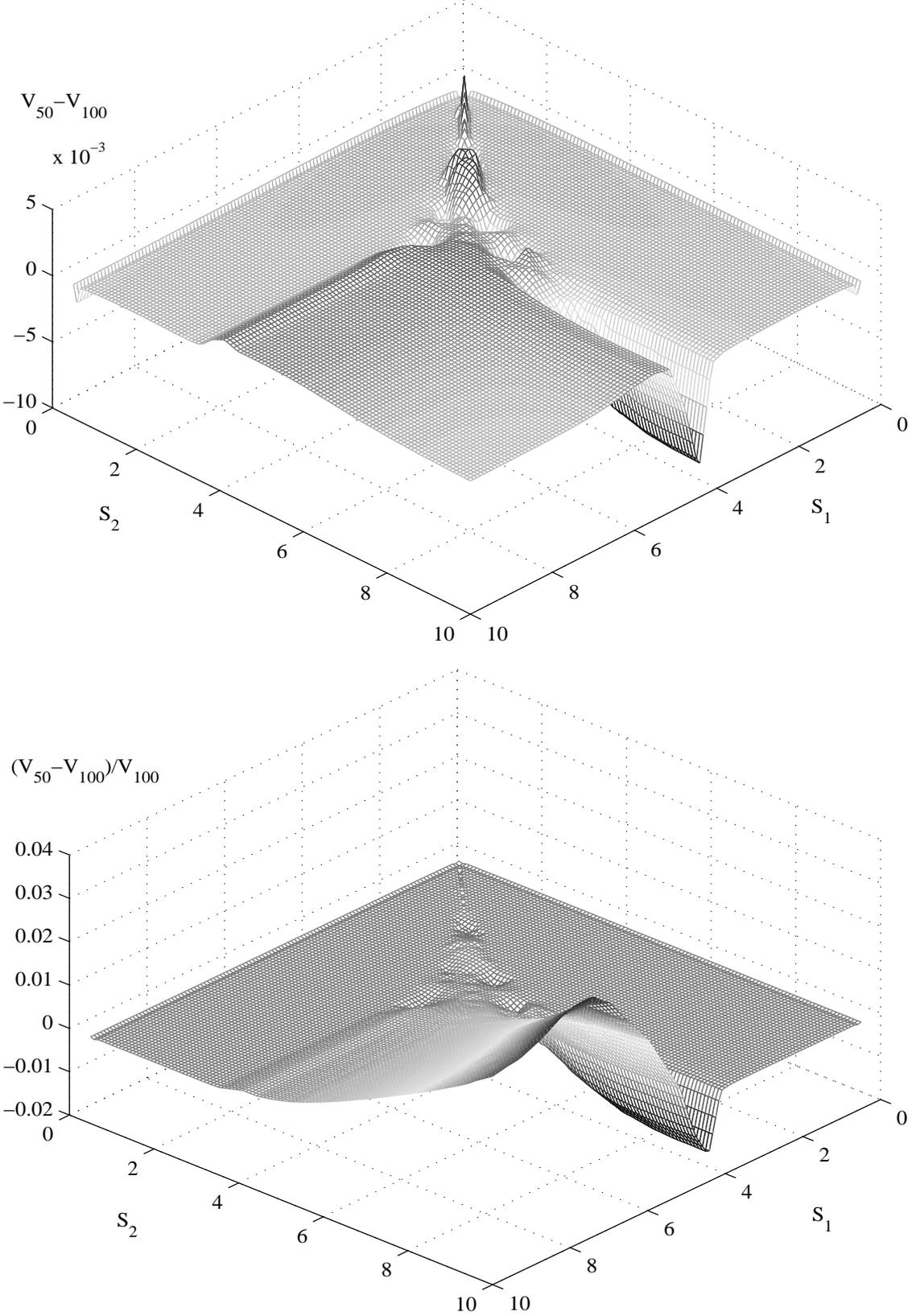


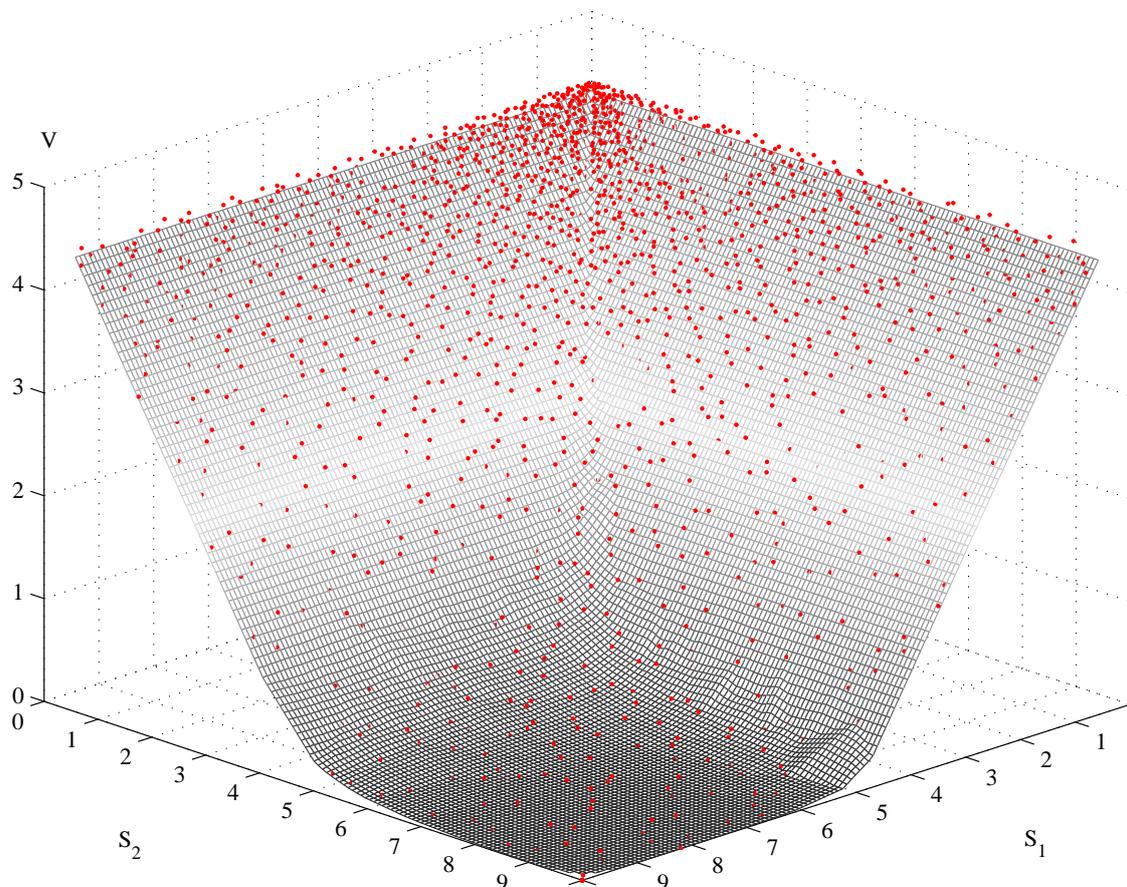
Figure 6.11.: Absolute and relative difference of option values computed with two different discretizations  $(n, m) = (50, 50)$  and  $(n, m) = (100, 100)$ .



### Mesh free test with nodes placed by Halton sequence

The RBF method is mesh free by design. To demonstrate this property we next use nodes generated by a Halton sequence.<sup>14</sup> For the placement of the nodes, a two dimensional Halton sequence has been generated in the  $(x_1, x_2)$ -domain of computation and then transformed into the  $(S_1, S_2)$ -plane. Figure 6.12 illustrates the result. For illustration purposes the RBF nodes have been plotted as points into this figure.

Figure 6.12.: Option price surface computed with Halton nodes.



**Remark 6.5** (Artefacts). *There are some artefacts in figure 6.12. First, the triangulation used to plot the value surface has been generated using the Halton points. Therefore there are coarse triangles in regions where only a few points are located. Second, some points, especially near the diagonal  $S_1 = S_2$ , seem to be hidden by the surface. This is an artefact of the plotting software.*

<sup>14</sup>The Halton sequence is a low-discrepancy sequence. The use of such a sequence has the advantage that the distribution in space is more even. For an introduction of the Halton sequence in the context of option pricing see, e.g., [Sey02], §2.4.

### 6.3.3. American rainbow binary option

Option:		Model:	
type	Rainbow binary option	class	GBM with corr.
exercise	American	volatility	$\sigma_1 = 0.2, \sigma_2 = 0.3$
payoff	$g(S_1, S_2) = \mathbf{1}_{[0, K_u]^2 \setminus [0, K_l]^2}$	correlation	$\rho = 0.3$
maturity	$T = 1$	return	$r_1 = r_2 = 0.1$
strikes	$K_l = 5, K_u = 7$	dividend	$q_1 = q_2 = 0$

Method/discretization:		Results:	
method	RBF	V(3,3)	0.120933
space discretization	$n = 140, \text{log-eqd.}$	V(4,4)	0.615592
time discretization	$m = 140, \text{eqd.}$	V(6,6)	1.0
computational domain	$[0.5, 20] \times [0.5, 20]$	V(8,8)	0.181623

This option on two assets  $S_1, S_2$  can be exercised at any time, paying 1, if both  $5 < S_1, 5 < S_2$  and  $S_1 < 7, S_2 < 7$  (and 0 otherwise). This type of option does not arise in practice so far, but from the numerical view point it is an interesting worst-case example. Not only the payoff is discontinuous, but also the location of the discontinuity is not differentiable.

#### Discussion of the error estimate

Figure 6.14 illustrates an estimate of the absolute and relative error. We see that the error is comparatively large and peaks at about 10% at the inner edge of the payoff profile. The relative error increases for small values  $S_1$  and  $S_2$  to a maximum of about 40%. The reason for these large errors lies in the discontinuous payoff. The discretization contains only a few nodes near the edges of the payoff profile. Better results will be only available through an adaptive placement of nodes.

Figure 6.13.: Payoff  $g(S_1, S_2)$  and option value  $V(S_1, S_2)$  at  $t = 0$ .

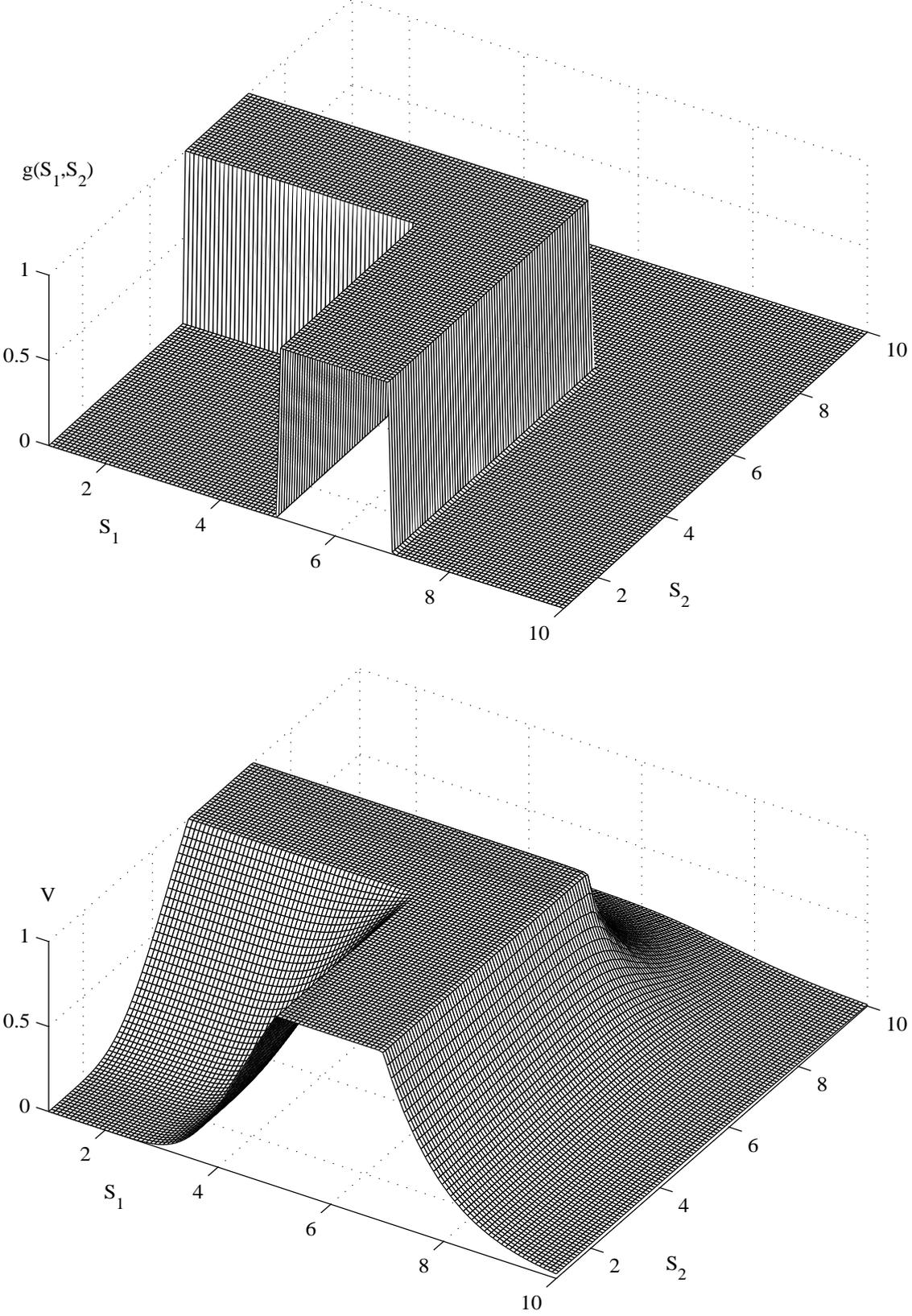
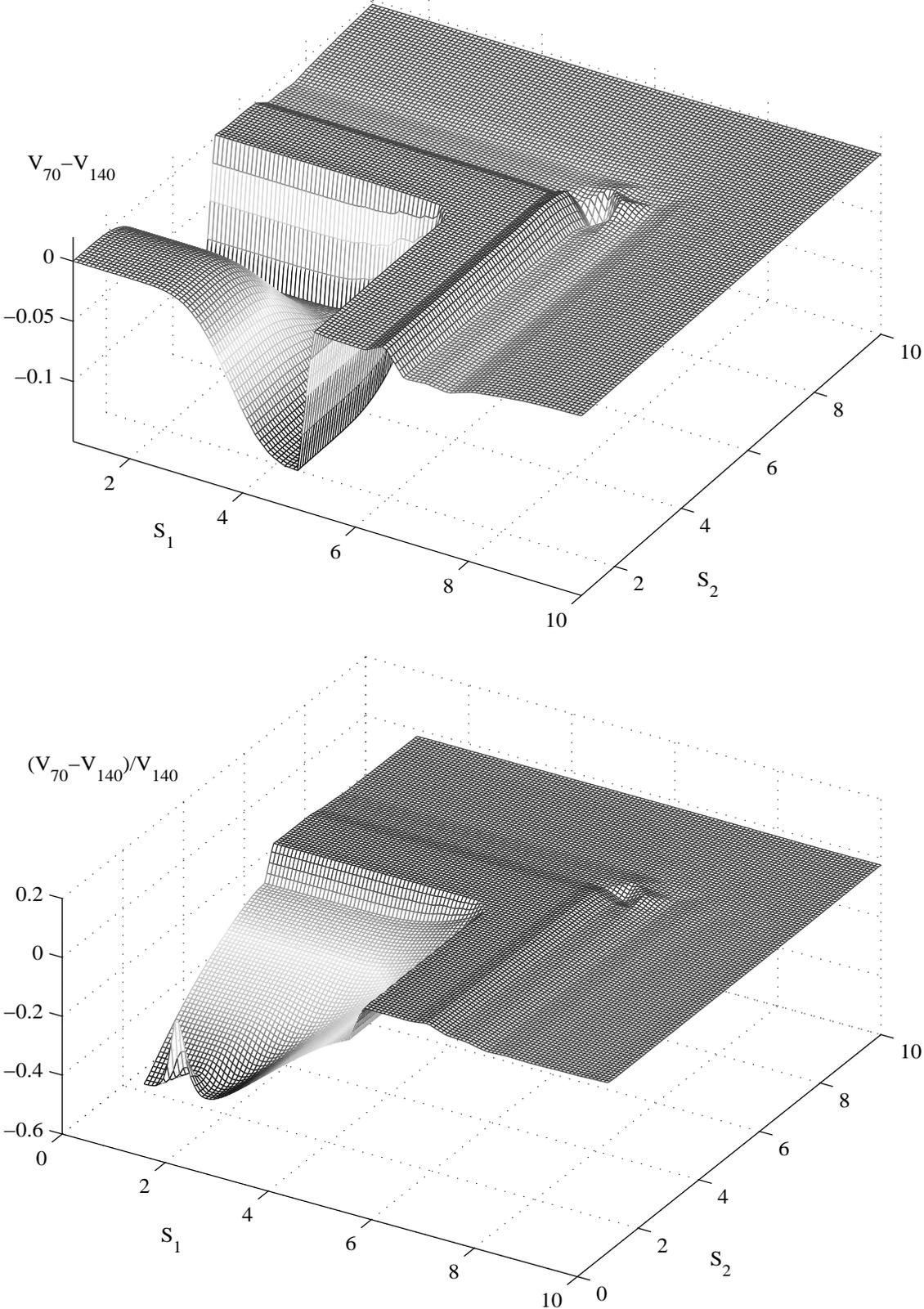


Figure 6.14.: Absolute and relative difference of option values computed with two different discretizations  $(n, m) = (70, 70)$  and  $(n, m) = (140, 140)$ .



### 6.3.4. European basket barrier option

Option:		Model:	
type	Basket <sup>15</sup> call with barriers	class	GBM with corr.
exercise	European	volatility	$\sigma_1 = 0.2, \sigma_2 = 0.3$
payoff	$g(S_1, S_2) = (S_1 + S_2 - K)^+$	correlation	$\rho = 0.3$
barriers (cont.)	$S_1 + S_2 > B_u$ : up-and-out $S_1 + S_2 < B_d$ : down-and-out	return	$r_1 = r_2 = 0.1$
maturity	$T = 1$	dividend	$q_1 = q_2 = 0$
strike/barriers	$K = 5, B_d = 5, B_u = 10$		

Method/discretization:		Results:	
method	RBF	V(3,3)	1.27747
space discretization	$n = 140$ , log-eqd.	V(4,4)	1.56239
time discretization	$m = 140$ , eqd.	V(6,2)	1.70626
computational domain	$[0.5, 10] \times [0.5, 10]$	V(4,2)	1.33825

This example is a European basket option with two continuously applied linear barriers. It is similar to that in [PFVS00] (p. 19, figs. 11+12), where the efficiency gains by unstructured meshing for finite element methods are discussed. Figure 6.15 illustrates the payoff at maturity and the current value at  $t = 0$ . The obvious inaccuracy at the barriers is caused by the fact that the barriers are not parallel to the  $S_1/S_2$  axes. The discretization of a line which is not parallel to the axes in a rectangular grid is not smooth.

#### Error estimate

Figure 6.16 shows an estimate for the absolute error. As expected it is seen that the maximal errors occur at the barriers. The mountain-like appearance of the error estimate stems from the fact that the barriers are not reproduced exactly by the (non-adaptive log-equidistant structured rectangular) space discretization. An adaptive (unstructured) discretization would provide much better results. The estimated relative errors  $|V_{140} - V_{70}|/|V_{140}|$  for this example are *not* displayed. Of course they have huge peaks near the barriers as in this region  $V \approx 0$ .

<sup>15</sup>*Basket* options are options on the weighted average of several underlyings.

Figure 6.15.: Payoff  $g(S_1, S_2)$  and option value  $V(S_1, S_2)$  at  $t = 0$ .

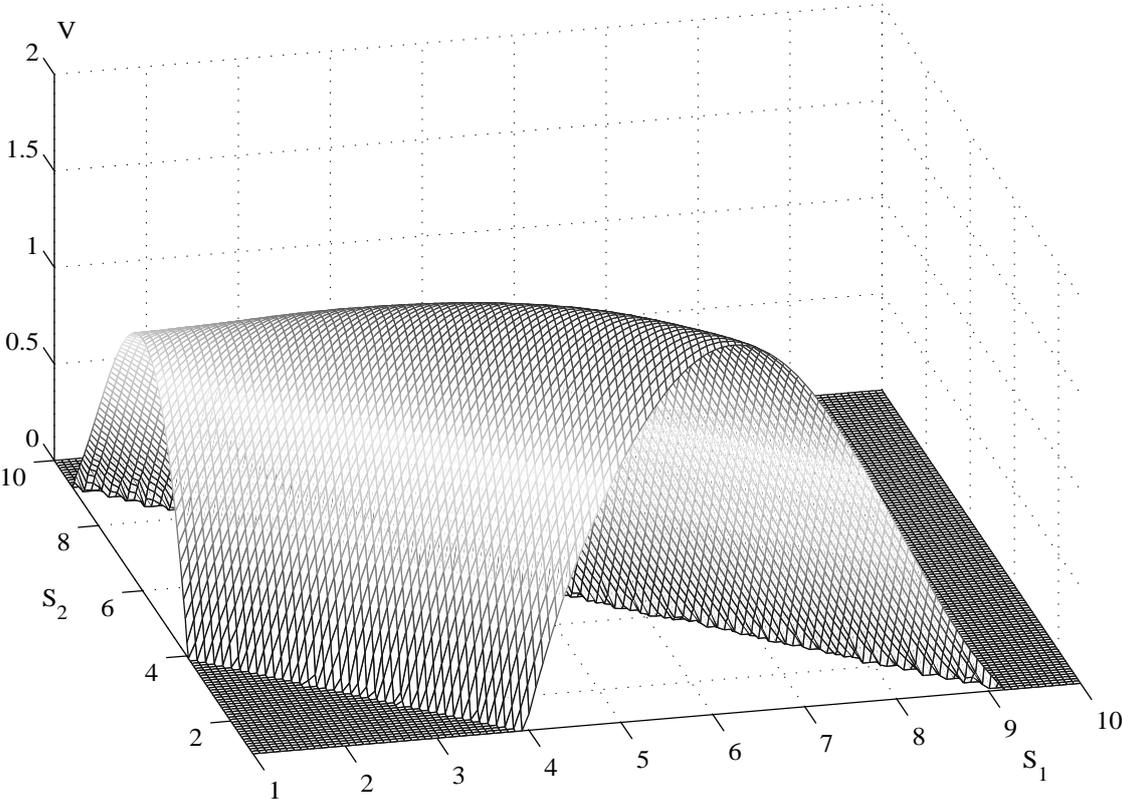
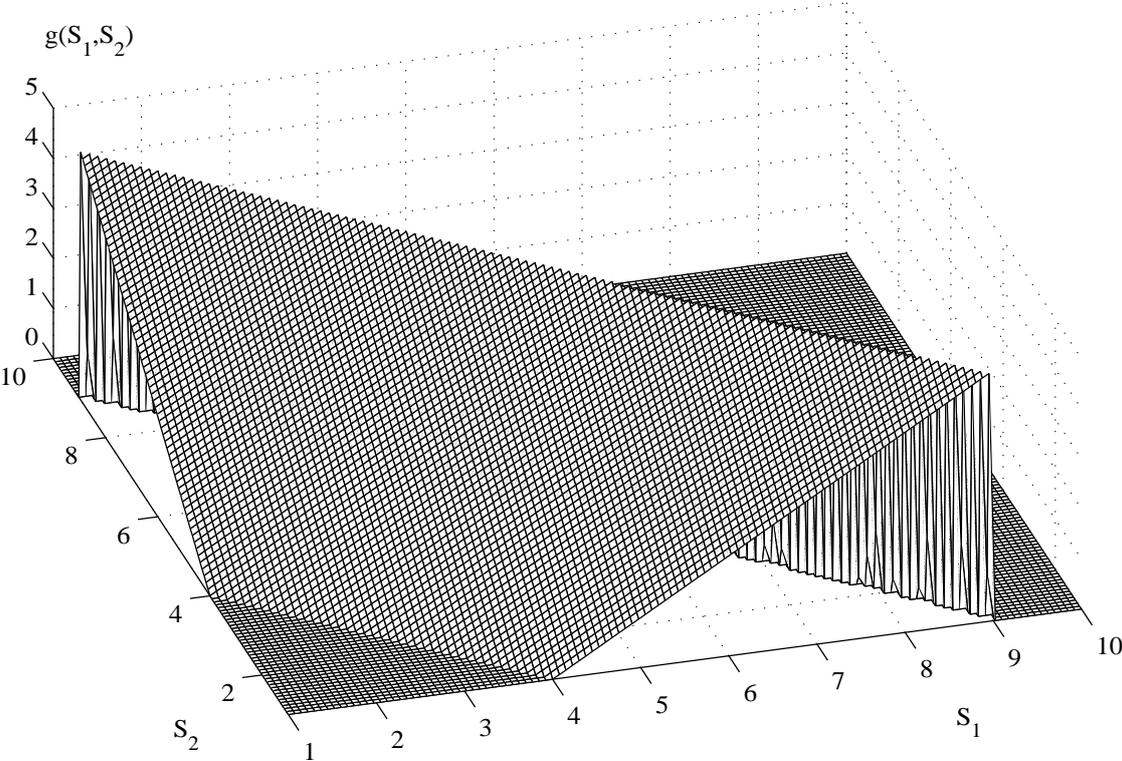
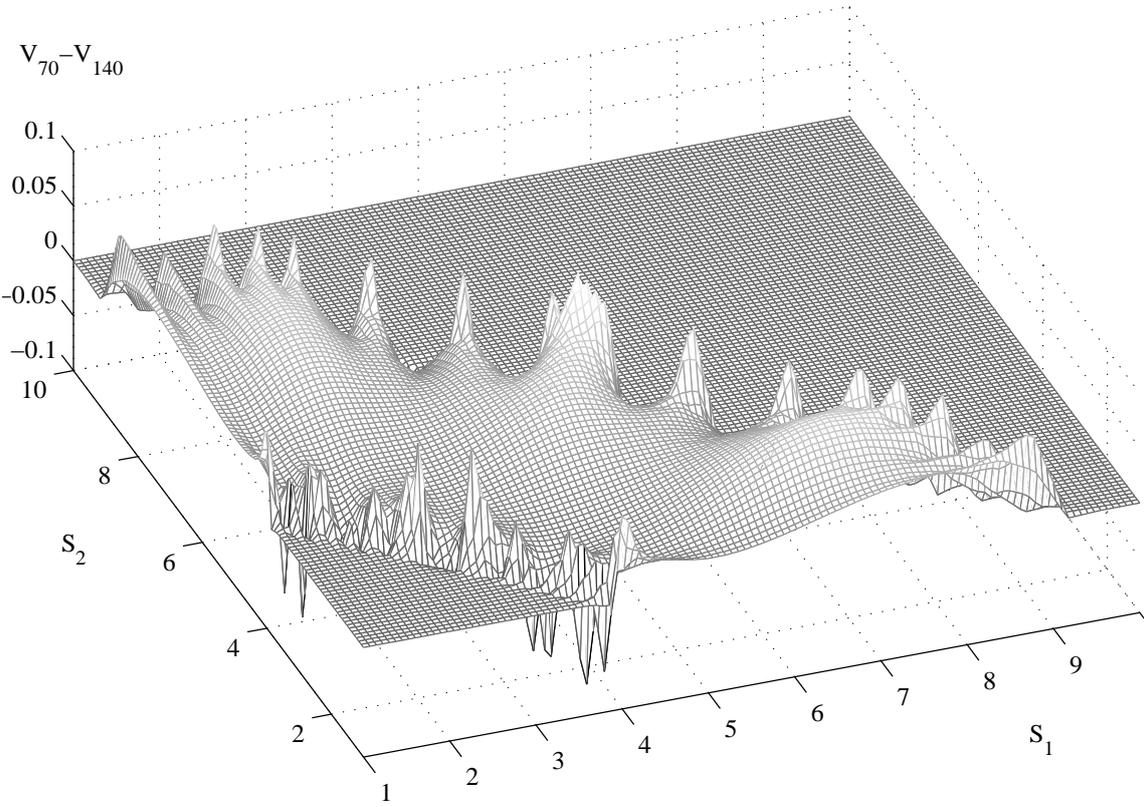


Figure 6.16.: Difference of option values at  $t = 0$  computed with two different discretizations  $(n, m) = (70, 70)$  and  $(n, m) = (140, 140)$ .



### 6.3.5. Three or more underlyings

Examples with three or more underlyings are problematic because for reasonable discretizations these problems cannot be solved with the currently available resources (usual desktop PC).

The reason is the well-known “curse of dimension”: A set of nodes with fill distance  $h$  in dimension  $d$  has  $n = \mathbf{O}(h^{-d})$  elements. A usual discretization for the single-asset case is  $n = 100$  nodes. The corresponding problem in three dimensions requires  $n^3 = 1000000$  nodes and leads in case of the RBF method to a sparse  $10^6 \times 10^6$  matrix. Whether such matrices can be handled with iterative solvers (such as GMRES<sup>16</sup> with preconditioning by incomplete LU decomposition) depends heavily on the number of non-zero elements.

#### Example: Bermudan maximum call on three assets

The following example is a maximum call on three underlyings with a Bermudan exercise structure of 10 equidistant exercise times. The example is similar to those in [BG04], tables 4ff., but here we use only three underlyings, as the computation of options on five underlyings is too expensive for the RBF method.

Option:		Model:	
type	Rainbow maximum call	class	GBM uncorrelated
exercise	Bermudan (10 eqd.)	volatility	$\sigma_1 = \sigma_2 = \sigma_3 = 0.2$
payoff	$g(S_1, S_2, S_3) = (\max(S_i) - K)^+$	return	$r_1 = r_2 = r_3 = 0.05$
maturity	$T = 3$	dividend	$q_1 = q_2 = q_3 = 0$
strike	$K = 100$		

Method/discretization:		Results:	
method	RBF	V(100,100,100)	18.53
space discretization	$n = 18$ , log-eqd.	V(150,100,100)	50.61
time discretization	$m = 10$ , eqd.	V(50,100,100)	13.80
computational domain	$[20, 200]^3$	V(50,150,100)	50.51

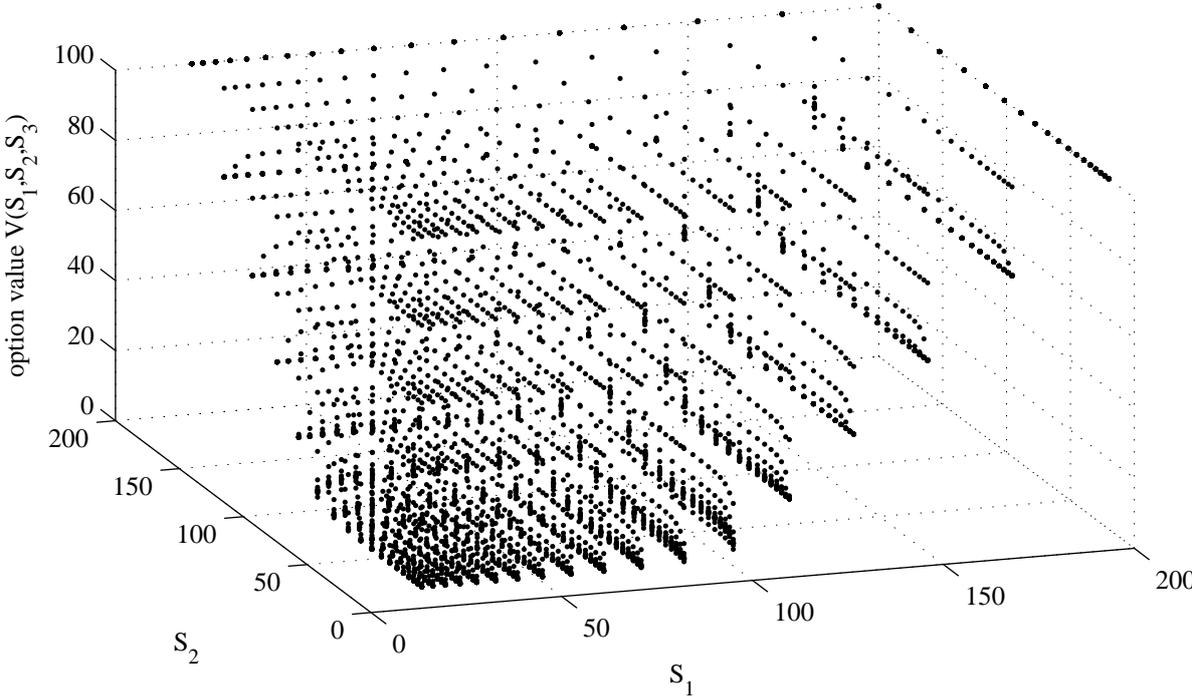
Figure 6.17 illustrates the results of the computation with a coarse discretization. The resulting nodes  $(S_1, S_2, S_3, V)$  have been projected to the hyperplane  $S_3 = 0$  and plotted without any graphical post-processing. For this computation the sparse basis matrix  $B$  is of size  $5832 \times 5832$  and contains  $\text{nnz}(B) = 7334872$  non-zero entries (this is 21.6%). The high density is a result of the coarse discretization. The usual density for a typical discretization is below 1%.

**Remark 6.6** (Accuracy). *Of course the accuracy for this coarse discretization is limited, but this is also the case for confidence intervals from stochastic mesh simulations.*

<sup>16</sup>GMRES: Generalized minimal residual method; cf. [Saa03].

Example for such an interval from [BG04], table 4, first row: The option value  $V$  is in the interval  $[15.804, 16.177]$  (relative error of 1.2%) with a probability of 90%.

Figure 6.17.: Projection of points  $(S_1, S_2, S_3, V)$  to the hyperplane  $S_3 = 0$ , where  $V$  is the value at  $t = 0$  of a Bermudan maximum call option on three underlyings.



## 6.4. Comparisons

Practical comparisons to other methods are difficult. The reason is that a fair comparison has to be performed under uniform conditions (on the same workstation) and, what is even more important, all methods must be implemented efficiently. In this spirit, the comparisons in this section must be treated carefully.

- (i) **Comparison to a finite difference method:** The American option problem from section 6.1.2 is the same as problem 8.3.2 in [IV05], where finite difference methods are used to solve an equivalent linear complementarity problem. The methods in [IV05] require 86s CPU time for an accuracy of  $10^{-4}$  and 300s CPU time for an accuracy of  $10^{-5}$ . (Compare [IV05], figure 8.2 and table 8.1.) For the same problem the spline method requires 2s for an accuracy of  $10^{-4}$  and 56s for an accuracy of  $10^{-5}$ . The RBF method requires 2.5s for an accuracy of  $10^{-5}$  and 65s for an accuracy of  $10^{-6}$ . (Compare table 6.3.) Even if one considers a compensatory factor of 10 for a slower CPU or suboptimal implementation, the RBF method is still by a factor of 10 faster. This indicates that the spline and RBF methods are at least competitive compared to finite difference methods in a Black-Scholes setting. For models with jumps we expect even better results, as the PDE approach then turns into a more expensive PIDE<sup>17</sup> approach.
- (ii) **Comparison to a binomial tree method:** As mentioned in section 6.1.2, a comparison to the binomial tree method is difficult, as each run of this method can only approximate a single option value  $V(S)$ . Indeed, for this task the binomial method is one of the fastest methods available. This can be illustrated by a test for the example from section 6.1.2. An efficiently implemented binomial tree method needs about 2s CPU time to achieve an accuracy of  $10^{-6}$  (thereby using  $n = 25000$  nodes). This is a factor 30 faster than the RBF method (on the same equipment). But the binomial tree does not provide the whole option price surface in time and space, which can, e.g., be used to compute the Greeks<sup>18</sup>. Another drawback of the Binomial method is its restriction to the Black-Scholes model.
- (iii) **Comparison to the Monte Carlo approach:** It is well-known that for option types which admit deterministic valuation methods, Monte Carlo is typically not the best choice. There is no need for numerical tests to confirm this. On the other hand, high-dimensional pricing problems are only accessible to Monte Carlo methods. Originally section 6.3.5 was meant to provide a comparable example for a high-dimensional problem, but options on five underlyings as in [BG04] turned out to be intractable for the RBF method.

<sup>17</sup>PIDE=partial integro-differential equation

<sup>18</sup>The most commonly used derivatives of the option value  $V$  are called *the Greeks*, as they are denoted traditionally by Greek letters, e.g.,  $\Delta := \partial V / \partial S$ ,  $\Gamma := \partial^2 V / \partial S^2$ , and  $\Theta := \partial V / \partial t$ .

- (iv) **Comparison between the spline and RBF method:** The results from section 6.1.2, table 6.3 indicate that the RBF method is better suited for the approximation of American option values than the spline method. Additional experiments confirmed this observation. However, for Bermudan options with only a few exercise times the spline method provides better results.

These short comparisons indicate that the RBF method is more efficient than the spline method. For those options which can be valued with the RBF method this seems a good and flexible choice.

**Remark 6.7.** *A comparison of several tree and analytical methods for American plain-vanilla options in a Black-Scholes setting can be found in [AC97].*

## 6.5. Further range of application

This section outlines the range of application for the proposed valuation methods. The options that can be valued using quadrature methods embrace

- (i) **Single-asset options** with European, Bermudan, or American exercise rights, arbitrary payoffs, knock-out<sup>19</sup> barriers, and continuous or discrete dividends;
- (ii) **Multi-asset options**, where the joint probability density function can be evaluated efficiently; this comprises combinations of the Lévy models in chapter 5 with a dependence structure modeled by copulae (section 5.2) but is not restricted to such models. For the application of quadrature methods the models presented in chapter 5 are not optimal. Models with density functions that can be evaluated more efficiently would be more favorable.
- (iii) **Composed options**, i.e., contracts that can be decomposed into several standard valuation problems. Three examples are:
  - **Option on options** (compound options and multiply compounded options); The corresponding valuation problem can be decomposed into several standard valuation problems that can be solved subsequently backwards in time where the “payoff” for each valuation problem is a linear function of previous results.
  - **Chooser options**, i.e., options that give the purchaser the right to decide at a fixed point  $t^*$  in time whether the derivative will be either a plain-vanilla call or put. The decomposition is as follows. For the interval  $[t^*, T]$  compute both call and put option values. The value of the chooser option at  $t^*$  is the maximum of both. This value can be used as “payoff” for the computation of the chooser option value for the remaining interval  $[0, t^*]$ .

<sup>19</sup>Options with knock-in barriers cannot be valued directly.

- Some **Asian options**, e.g., options whose payoff depends on a discretely sampled arithmetic average of the price of the underlying; The resulting two-dimensional<sup>20</sup> problem can be solved with quadrature methods analogously to the PDE approach (see, e.g., [Sey02], §6.2).

These types of options can also be valued by other methods, e.g., tree or finite difference methods.

- (iv) **Cross options**, i.e., options on an underlying in one currency with a strike denominated in another currency; The pricing of such options is possible but computationally more expensive. The reason is that exchange rate models usually incorporate mean reversion features which violate the space homogeneity assumption 2.5. For models without space homogeneity, lemma 4.11 does not apply, and consequently the model matrix  $M$  does not have a Toeplitz structure.
- (v) **Weather derivatives**. The model of the underlying is described by a conditional probability density function. If for any underlying such a density can be estimated, the quadrature approach can value options on that underlying. The usual approach for stock options is parameter based: A model is specified and the parameters are estimated by a calibration to market data. For quadrature methods, it is possible to use a non-parametric estimation of the density function as well. For example, for weather derivatives (options with a payoff depending on temperatures), a kernel density estimate<sup>21</sup> could be performed based on historic temperature data.

The previous sections cover various combinations from (i) and (ii). Examples for (iii) would be technically involved but little surprising. The remaining cases (iv) and (v) are very interesting but demand additional prerequisites which are not included in this work, namely, exchange rate models for (iv) and kernel estimation techniques for (v).

---

<sup>20</sup>The variables are price  $S$  and average  $A$ .

<sup>21</sup>See e.g., [Sil86] for an introduction to density estimation.

## 7. Conclusion

In this work, two new pricing methods for Bermudan options have been proposed. Already in the univariate Black-Scholes setting, these methods are the first being capable of pricing Bermudan options with a high number of exercise times in an efficient manner. Furthermore, they can be used to price options with arbitrary payoffs under fairly arbitrary market models. However, the main contribution of this work can be seen in an exploration of the new class of quadrature based pricing methods. Potentials and limits of this approach can now be estimated.

The valuation of American options is possible by means of a qualitative convergence result (lemma 2.14, p. 19). Empirical results indicate linear convergence of the values  $V_m$  of Bermudan options with  $m$  equidistant exercise times for  $m \rightarrow \infty$ . In fact, the results even indicate a linear *relation* between  $V_m$  and  $\frac{1}{m}$ . Such a relation would imply that a highly accurate valuation of American options is possible by a highly accurate valuation of Bermudan options with only a moderate number of exercise times. Thus, a thorough analysis of the relation between the value of American and Bermudan options could be useful.

Concerning the space discretization, mixed results have been found. A structural difference between the spline and RBF method is that the spline method interpolates the hold value of an option, while the RBF method interpolates the option value itself. As shown in section 3.4.2, the hold value is smooth under rather general conditions. Thus, for the spline method the interpolated functions are smooth, whereas the integrands have singularities. Contrariwise, for the RBF method all integrands are smooth, whereas the interpolated functions have singularities. As the space discretization error depends on the smoothness of both the integrands and interpolated functions, the challenge in the spline method is primarily to use quadrature methods that respect the structure of the interpolant, while the challenge in the RBF method is primarily to construct an adaptive RBF interpolant that simultaneously provides low interpolation errors and favorable structures of basis and model matrix.

Both problems have been solved satisfactorily for the univariate case. It is shown that the univariate spline method has at least quadratic convergence in the computation time. Empirical results indicate an actually cubic convergence. For the RBF method, empirical results indicate quadratic convergence for both binary and plain-vanilla options. These results are based on equidistant nodes. For adaptively placed nodes, better results can be expected.

In the multivariate case, the efficiency of the spline method suffers from costly evaluations of the interpolant on the one hand and from the inherent difficulty of multivariate

quadrature over nonsmooth integrands on the other hand. An efficient implementation of the multivariate spline method requires an efficient management of adaptive triangulations which is a nontrivial task. For the RBF method, the multivariate implementation is straightforward. However, the costs increase considerably with the dimension of the underlying. There are two reasons. First, in the multivariate case,  $n^d$  centers are required to reach the same accuracy as with  $n$  centers in the univariate case. Second, the quasi-band structure of the basis matrix gets lost.

In a direct comparison, the RBF method seems preferable. All integrals can be approximated in a single setup step. Furthermore, the integrands are for many models globally smooth, which enables the use of high-order quadrature rules. In some cases, the integrals can even be solved analytically or approximated efficiently with adapted quadrature rules. Last but not least, the method is mesh-free, which simplifies the location and management of nodes, as no triangulation structure on the nodes is required.

There are numerous open problems and interesting directions of future research. Particularly important seems to be an efficient extension to *high*-dimensional underlyings. In principle, the RBF approach is suitable for this purpose, but due to vastly increasing costs for higher dimensions it is – in its current form – limited to  $d \leq 3$ . The central issue is that RBF interpolation is, in a formal sense, a *global* method. Thus, an attempt to localize the method in order to avoid huge linear systems seems promising.

# A. Appendix

## A.1. Notation

$(\cdot)^+$	$(\cdot)^+ := \max(\cdot, 0)$
$\ f\ _\infty$	supremum norm of $f$
$\mathbb{E}(X)$	expectation of $X$
$\mathcal{F}(f)$	Fourier transform of function $f$
$\mathcal{L}^1(\mathbb{R}^k)$	space of absolutely integrable functions on $\mathbb{R}^k$
$\mathbf{O}(f)$	Landau symbol: $f \in \mathbf{O}(g) :\Leftrightarrow 0 \leq \limsup_{x \rightarrow \infty}  f(x)/g(x)  < \infty$
$\mathbb{Q}$	equivalent martingale measure to $\mathbb{P}$
$\text{var}(X)$	variance of random variable $X$
<hr/>	
$\alpha$	exponential tail decay rate of probability density functions
$\alpha_i$	shape parameters for interpolation
$B$	base matrix $B \in \mathbb{R}^{n \times n}$ , $B$ is symmetric by definition
$d$	dimension of the model's price space
$f^X$	probability density function of the stochastic variable $X$
$g$	payoff function
$K$	strike price of an option
$\lambda_i$	interpolation coefficients $\lambda_i \in \mathbb{R}$
$L, U$	LU decomposition of $B$
$\mu$	drift parameter (model parameter)
$M$	model matrix $M \in \mathbb{R}^{n \times n}$ (RBF method)
$n$	number of interpolation nodes
$\Omega$	set of possible scenarios which can occur in the market
$\phi$	radial basis function, in this work Gaussian kernel function
$q$	continuous dividend rate paid by an asset
$r$	risk-free interest rate
$\sigma$	volatility (model parameter)
$S$	price of the underlying
$t$	current time to maturity in the valuation process, $t \in [0, T]$
$T$	time to maturity of an option
$\mathcal{T}$	set of exercise times of a Bermudan option
$V^H$	hold value of an option (see p. 14)
$x$	log-price of the underlying ( $x = \log S$ )
$x_i$	interpolation nodes $x_i \in \mathbb{R}^d$ , where $x_{ij} = \log(S_{ij})$

## A.2. Convolution-based methods

This section proposes an alternative valuation method for European options with arbitrary payoffs that can also be used to build valuation methods for Bermudan options. It has not yet appeared in the recent literature. It is included in the present work only to allow the short discussion in section 2.6 (p. 29).

Carr and Madan [CM99] propose a method for pricing European plain-vanilla options by fast Fourier transform. While this method is restricted to plain-vanilla options, a similar idea leads to a method for options with *arbitrary* payoffs and can thus serve as basis for Bermudan pricing methods.

### A method for European options with arbitrary payoffs

Consider a European option. Let  $X_t \in \mathbb{R}$  be the stochastic process describing the log-price of the underlying and  $V$  the price of an option on this underlying. The process  $X_t$  is assumed to be translation invariant (assumption 2.5). While this assumption is *optional* for quadrature methods, it is *mandatory* for convolution-based methods. Assumption 2.5 implies for the conditional probability density function:

$$f^{X_t|X_0=x}(\xi) = f^{X_t|X_0=0}(\xi - x)$$

In this case, the conditional probability can be described by a single univariate function  $f := f^{X_t|X_0=0}$  and for the rest of this section we simply write  $f$ . As before, the option price  $V(x)$  can be represented as a discounted risk-neutral expectation value

$$V(x) = e^{-rT} \int_{-\infty}^{\infty} f(\xi - x)g(\xi) d\xi,$$

where  $g$  is an *arbitrary* terminal payoff. For notational convenience in the following  $\tilde{f}(x) := f(-x)$  is defined. This leads to

$$e^{rT}V(x) = \int_{-\infty}^{\infty} \tilde{f}(x - \xi)g(\xi) d\xi = (\tilde{f} * g)(x),$$

which is the convolution of the two functions. Following the convolution theorem, the convolution corresponds to the product of the Fourier coefficients.<sup>1</sup> Let  $\mathcal{F}(g)$  denote the Fourier transform of a function  $g$ . Applying the Fourier transform to both sides we get

$$\mathcal{F}(e^{rT}V) = \mathcal{F}(\tilde{f} * g) = \mathcal{F}(\tilde{f})\mathcal{F}(g).$$

The pointwise product can be evaluated in  $\mathbf{O}(n)$  for a discretization with  $n$  points. The fast Fourier transformation can be performed in  $\mathbf{O}(n \log n)$  in both directions, leading to total costs of  $\mathbf{O}(n + n \log n) = \mathbf{O}(n \log n)$  for the computation of the European option price.

<sup>1</sup>The convolution theorem can be found in [Bra99], for instance.

## Existence of the Fourier transform

The above motivation ignores that the Fourier transform  $\mathcal{F}(g)$  does not necessarily exist. A sufficient condition for its existence is that  $g$  is absolutely integrable. While the density function  $f$  meets this requirement by definition, it is not guaranteed for the payoff  $g$ . In the following two approaches are introduced which can reduce the pricing problem to one with an absolutely integrable  $g$ .

### Difference pricing

The first idea requires that the option price is known for a payoff function  $g_0$  such that  $(g - g_0)$  is absolutely integrable. Then one can compute the difference price

$$V(x) - V_0(x) = e^{-rT} \int_{-\infty}^{\infty} \tilde{f}(x - \xi)[g(\xi) - g_0(\xi)] d\xi.$$

Using the valuation procedure for  $(g - g_0)$  yields the difference  $V - V_0$  and thus  $V$ . For the pricing of Bermudan options via the reduction principle 2.12 this idea is applicable for plain-vanilla payoffs. In this case,  $V := V^H$  is the hold value of the Bermudan option at the next time step.  $V_0$  can be chosen to be the European plain-vanilla option price, which can be approximated by a valuation method for European options. Then,  $V - V_0$  is absolutely integrable.

### Shift in frequency domain

A more elegant idea is to introduce a shift in the frequency domain, i.e., to compute  $\mathcal{F}(V)(\omega + c)$  instead of  $\mathcal{F}(V)(\omega)$  (with a constant  $c \in \mathbb{C}$ ). A shift in the frequency domain corresponds to a factor  $e^{-ict}$  in the time domain:<sup>2</sup>

$$\mathcal{F}(e^{-ict} f(t))(\omega) = \mathcal{F}(f(t))(\omega + c) \text{ for any function } f \text{ with existing } \mathcal{F}(f)$$

Using a shift by  $c := i\beta$ , where  $\beta \in \mathbb{R}$  is a constant parameter, leads to

$$\begin{aligned} \mathcal{F}(e^{rT} e^{\beta x} V(x))(\omega) &= \mathcal{F}(e^{rT} V(x))(\omega + i\beta) \\ &= \mathcal{F}(f(x))(\omega + i\beta) \mathcal{F}(g(x))(\omega + i\beta) \\ &= \mathcal{F}(e^{\beta x} f(x))(\omega) \mathcal{F}(e^{\beta x} g(x))(\omega) \end{aligned}$$

This allows to “damp” the functions for either positive or negative  $x$ , depending on the sign of  $\beta$ . The following additional assumption ensures integrability.

---

<sup>2</sup>This holds for the unitary version of the Fourier transform.

**Assumption A.1** (Weighted integrability). *There exists a constant  $\beta \in \mathbb{R}$ , such that both the risk-neutral density function  $f$  and the payoff function  $g$  are absolutely integrable with respect to the weighting function  $e^{\beta x}$ :*

$$\exists \beta \in \mathbb{R} : \int_{-\infty}^{\infty} e^{\beta x} f(x) dx < \infty \quad \text{and} \quad \int_{-\infty}^{\infty} e^{\beta x} g(x) dx < \infty$$

**Example A.2.** *Whether assumption A.1 holds or not depends on both the density  $f$  and the payoff  $g$ . For example, consider a plain-vanilla put. In this case it is  $g(x) = 0$  for  $x \gg 0$  and  $g(x) = \mathbf{O}(1)$  for  $x \rightarrow -\infty$ . It suffices to choose  $\beta < 0$ . More critical is the case of a plain-vanilla call, where  $g(x) = 0$  for  $x \ll 0$  and  $g(x) = \mathbf{O}(e^x)$  for  $x \rightarrow \infty$ . In this case it is necessary to choose  $\beta < -1$ . Now consider the integrability of  $e^{\beta x} f(x)$  for the Black-Scholes model. Its density function has the form<sup>3</sup>*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi t}} \exp\left(-\frac{(x - (\mu - \frac{\sigma^2}{2})t)^2}{2\sigma^2 t}\right),$$

which is in  $\mathbf{O}(e^{-\alpha|x|})$  for  $|x| \rightarrow \infty$  for every  $\alpha > 1$  (lemma 5.10) – particularly for  $\alpha := 1 - \beta$ . Then  $e^{\beta x} f(x) = \mathbf{O}(e^{-x})$  for  $x \rightarrow -\infty$  and consequently  $e^{\beta x} f(x)$  is absolutely integrable. Thus, for the Black-Scholes model it is possible to find a suitable damping parameter  $\beta$ .

For an arbitrary model, especially for one with “heavy tails”, assumption A.1 is not guaranteed. Later, section 5.1.7 shows that the right tail of the log-return PDF of any reasonable model decays at least with exponential decay rate 1, i.e.,  $f(x) \in \mathbf{O}(e^{-x})$  for  $x \rightarrow \infty$ . Unfortunately, a similar restriction cannot be derived for the left tail, and although all models considered in chapter 5 have exponentially decaying left tails, the behavior  $f(x) \notin \mathbf{O}(e^{-|x|})$  for  $x \rightarrow -\infty$  is possible. The weighted integrability assumption fails for plain-vanilla calls under such models. Consequently the convolution approach with frequency shifting cannot be used in this case.

### Comparison to quadrature based valuation methods

A short discussion of differences and similarities between convolution-based methods and quadrature methods can be found in section 2.6 (p. 29).

<sup>3</sup>  $\sigma$  and  $\mu$  are model parameters (volatility and drift), and  $t$  denotes the size of the time steps of the Bermudan exercise structure.

## A.3. Implementation notes

This section contains technical remarks on some details of the implementation.

### Choice of suitable programming environments

The algorithms have been implemented in parts using Microsoft Visual C/C++ (version 8.0) and in parts using MATLAB (version 7.2.0). In addition to that, high order quadrature code written in Fortran 77 has been used.

All parts that involve many elementary operations have been implemented in C/C++. This applies to the spline method and to the setup of the basis and model matrices in the RBF method. This kind of operations cannot be implemented efficiently by interpreted MATLAB code and benefits a lot from optimized compilation to native code.

The remaining parts of the algorithms involve only a few large-scale operations (typically linear algebra for large matrices). They have been implemented using MATLAB. The interaction between both parts of the program has been implemented following the “MATLAB Applications Program Interface” (MATLAB API).<sup>4</sup>

The Fortran 77 code for the quadrature methods has been linked against the C/C++ code. It is used in the setup of the model matrix for the RBF method.

### Creating sparse matrices for MATLAB

MATLAB stores matrices in a column wise manner. Sparse matrices use an array *sr* where the (real) non-zero elements are stored. Two additional arrays *irs* and *jcs* contain information about the location of the non-zero elements in the matrix. Non-zero elements are arranged column-wise from left to right. *irs*[*k*] contains the row number of the *k*-th non-zero element, *jcs*[*j*] contains the total number of all non-zero elements in columns with index less than *j*.

As the parts of the program that are written in C create large sparse matrices (e.g.  $100000 \times 100000$ -matrices), it is important to take this structure into account when generating these matrices. As the number of non-zero elements is not known a priori, the following allocation algorithm is used: For an  $n \times n$ -matrix we first assume a density of 1% (i.e.  $n^2/100$  non-zero elements). While the elements are computed and the matrix is filled, the amount of memory allocated is monitored. Whenever the number of non-zero elements exceeds the amount of allocated memory, a reallocation takes place, using the estimation  $nnz * n / j$ , where *j* is the number of columns that are already computed, *nnz* is the number of non-zero elements encountered so far, and *n* is the total number of columns. This allocation algorithm reduces the amount of necessary reallocations.

---

<sup>4</sup>This API is documented in [Mat98].

## Possible parallelization

The setup of the model matrix for the RBF method can be perfectly parallelized for several concurrent processors, as the computation of its entries are mutually independent. For each processor, a single thread is spawned and each thread is assigned to fill a certain sub matrix of the model matrix. When all threads have finished, the sub matrices are assembled into the result matrix. This multi-threading approach can be implemented in the C++ part, such that it is transparent for MATLAB.

## Structure of the code

The methods proposed in this work have been implemented by the author. The implementation consists of three independent program packages, namely

- (i) the spline method – univariate case –,
- (ii) the spline method – multivariate case –, and
- (iii) the RBF method.

The codes (i) and (ii) are written purely in C++, and (iii) is written partly in C++ and partly in MATLAB. While the first two programs are compiled to stand-alone executables, the third “program” consists of a collection of subroutines that can be called from the MATLAB environment. In the following, the structure and some key elements of these codes are briefly described.

### Spline method, univariate case

For the spline method, an abstract class FN is used to represent a function. Classes for special functions, namely spline interpolants, payoff functions, European option value functions, etc. are derived from FN. This structure allows for a convenient handling of the spline functions and avoids long lists of parameters describing a single interpolant. Furthermore, every subroutine that takes an parameter of type FN\* can be invoked with any kind of function.

Listing A.1: Fn.h

```

class Fn : public ::Object { // a general function.
public:
    virtual double operator()( double x ) const = 0;
    virtual double integrate( double x0,
        double x1, double tol ) const;
    virtual void plot( const char* filename ) const;
    virtual void support( double& x0, double& x1 ) const;
    virtual bool getNextSing( double *x, int* hint ) const;
};

```

An important feature of the class `FN` are the member functions `SUPPORT` and `GETNEXTSING`, which can be used to get the support and singularities of the function. This allows a convenient way of handling singularities – especially in the quadrature routine, which has to respect the spline nodes as singularities.

Of course, a key role in this program plays the class `FNSTOREDSPLINE`, which represents a spline interpolant. Its most important member function is the constructor, which adaptively places the nodes as described in section 3.3.1 for the function parameter `FN`. (The type `SP<FN>` is essentially a “smart” pointer with garbage collection.)

Listing A.2: `FnStoredSpline.h`

```
class FnStoredSpline : public Fn {  
public :  
    FnStoredSpline( const SP<Fn>& fn , double tol );  
    [...] ;  
};
```

The main routine of the univariate spline program first constructs the payoff function, then a European option value function with this payoff, which enters the constructor of the first spline instance. The spline can then be used with for building the maximum function with the payoff, and finally enters again a European option value function, etc. The model of the underlying occurs in the implementation of the class `FNEUROPEAN`. A `FOR` loop iterates over all exercise times, and several outputs are written in files for graphical post processing.

**Spline method, multivariate case**

For the multivariate spline method, essentially the same function framework is used as in the univariate case. However, in its current version, singularities are not respected. A specific feature that does not occur in the program for the univariate case is the management of the triangulation by the class MESH.

Listing A.3: Mesh.h

```

class Mesh {
public:
    Mesh();
    bool refineSimplex( int idxSimplex ,
        Function* f, Interpolant* interpolant ,
        double tol );
    void findSimplexIgnore( Simplex* in ,
        const double* x, Simplex** simplex ,
        Simplex* ignore );
    void findSimplex( const double* x, Simplex** simplex );
    bool isInSimplex( const double* x, Simplex* simplex );
    void resizeData( int m );
    void resizeVertices( int m );
    void resizeSimplices( int m );

    double* data;
    int* vertices;
    Simplex* simplices;
    int n, nData, szData;
    int nVertices, szVertices;
    int nSimplices, szSimplices;
};

```

This class cannot be discussed in detail on a few pages. The most important function is `REFINESIMPLEX`, which refines the simplex specified by parameter `IDXSIMPLEX` as described in section 3.5.3. Although mathematically simple, an efficient implementation is quite difficult, as the triangles cannot be represented by instances of classes (as one would expect in C++). That was the first implementation approach and it turned out to slow down the computation due to the memory management routines of C++. Classes are not suitable for a huge number of tiny objects, which are created dynamically. To overcome this problem, the memory allocation for the triangulation objects (vertices, simplices, and their relation) had to be implemented manually.

**RBF method**

The C++ based part of the RBF implementation consists of two C++ modules BASISMATRIX.C and MODELMATRIX.C, which construct the basis matrix  $B$  and model matrix  $M$  from chapter 4. The MATLAB based part of the RBF implementation consists of several auxiliary functions, such as PAYOFF.M, and a main function for each number of dimensions  $d = 1, 2, 3$ . There are different implementations, as the dimension is not handled in a generic way. Probably it would be possible to write a single main function for all dimensions. In the following listing, some lines from the main function for  $d = 2$  are printed. The listing is not complete, but it should suffice to indicate how the different modules work together.

Listing A.4: main2d.m (first part)

```

function [M B xi lambda alpha]=main2d( n, m )
[... ]
% generate n*n equidistant nodes on the square [-a, a]x[-a, a]
for i=1:n
    for j=1:n
        xi((i-1)*n+j, 1) = a0 + (a1-a0)*(i-1)/(n-1);
        xi((i-1)*n+j, 2) = a0 + (a1-a0)*(j-1)/(n-1);
    end
end
[... ]
% construct basis matrix via C++ subroutine:
B=basismatrix(2, length(xi), xi, alpha);
[... ]
% LU decomposition of basis matrix:
if use_sparse==0
    [L,U,P,Q]=lu(B);
else
    [L1,U1]=luinc(B, tol);
end
[... ]
% specify market model by vector Theta:
Theta = [0, dt, mu, delta, sigma]; % B/S
% construct model matrix:
M=modelmatrix(2, length(xi), xi, alpha, Theta)';
[... ]
% evaluate payoff at maturity (parameters K1,K2):
y=payoff_2d(xi, K1, K2);
[... continued on next page ...]

```

Listing A.5: main2d.m (second part)

```
[... continued from previous page ...]
% Iterate backwards through time:
for k=1:(m-1)
    v2=exp(-dt*r)*(M*lambda);
    v2=max(y_pay , v2);
    if use_sparse==0
        lambda = (Q*(U\((L\((P*(v2))))));
    else
        lambda = gmres(B,v2 , restart , tol , maxit , L1 , U1 , lambda_old);
        lambda_old=lambda;
    end
end
[... here plotting commands ...]
```

# Bibliography

- [AC97] F. AitSahlia and P. Carr. American options: A comparison of numerical methods. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*. Cambridge University Press, 1997.
- [ADNW06] A. D. Andricopoulos, P. W. Duck, D. P. Newton, and M. Widdicks. Extending quadrature methods to value multi-asset and complex path dependent options. *Working paper*, 2006.
- [Alt85] H. W. Alt. *Lineare Funktionalanalysis*. Springer, 1985.
- [AS65] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions*. Dover, 1965.
- [AWDN03] A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton. Universal option valuation using quadrature methods. *Journal of Financial Economics*, 67(3):447–471, 2003.
- [BD72] Å. Björk and G. Dahlquist. *Numerische Methoden*. Oldenbourg Verlag, 1972.
- [BG97] M. Broadie and P. Glassermann. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8-9):1323–1352, 1997.
- [BG04] M. Broadie and P. Glassermann. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7(4):35–72, 2004.
- [BL84] A. Bensoussan and J. L. Lions. *Impulse control and quasi-variational inequalities*. Gauthier-Villars, Paris, 1984.
- [BN97] O. E. Barndorff-Nielsen. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scandinavian Journal of Statistics*, 24:1–13, 1997.
- [Bra99] R. Bracewell. Convolution theorem. In *The Fourier Transform and Its Applications*, pages 108–112. McGraw-Hill, New York, 3rd edition, 1999.
- [BRS63] F. L. Bauer, H. Rutishauser, and E. Stiefel. New aspects in numerical quadrature. *Proc. of Symposia in Applied Mathematics*, 15:199–218, 1963.

- [BS73] F. Black and M. Scholes. The price of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.
- [Buh03] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003.
- [CCS01] C.-C. Chang, S.-L. Chung, and R. C. Stapleton. *Richardson Extrapolation Techniques for the Pricing of American-style Options*. Draft, Strathclyde University, United Kingdom, 2001.
- [CF05] S. S. Clift and P. A. Forsyth. Numerical solution of two asset jump diffusion models for option valuation. *Working paper*, 2005. Electronically available: <http://tinyurl.com/2cfyhy>.
- [CGMY02] P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: an empirical investigation. *The Journal of Business*, 75:305–332, 2002.
- [CIR85] J. C. Cox, J. E. Ingersoll, and S. A. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407, 1985.
- [CLM97] J. Y. Campbell, A. W. Lo, and A. C. MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997.
- [CM99] P. Carr and D. B. Madan. Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2(4):61–73, 1999.
- [CRR79] J. Cox, S. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:87–106, 1979.
- [CT03] R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman and Hall / CRC Press, 2003.
- [CW03] P. Carr and L. Wu. The finite moment logstable process and option pricing. *Journal of Finance*, 58(2):753–778, 2003.
- [DK94] E. Derman and I. Kani. The volatility smile and its implied tree. *Risk*, 7(2):139–145, 32–39, 1994.
- [DR75] P. J. Davis and P. Rabinowitz. *Methods of numerical Integration*. Academic Press, New York, 1975.
- [DS94] F. Delbaen and W. Schachermayer. A general version of the fundamental theorem of asset pricing. *Mathematische Annalen*, 300:463–520, 1994.
- [DS98] F. Delbaen and W. Schachermayer. The fundamental theorem of asset pricing for unbounded stochastic processes. *Math. Annalen*, 312:215–250, 1998.

- [DY02] A. A. Dragulescu and V. M. Yakovenko. Probability distribution of returns in the Heston model with stochastic volatility. *cond-mat 0203046v3*, 2002.
- [Eks04] E. Ekström. Properties of American option prices. *Stochastic Processes and their Applications*, 114(2):265–278, 2004.
- [Els02] J. Elstrodt. *Maß- und Integrationstheorie*. Springer, 3rd edition, 2002.
- [FM03] T. Fujiwara and Y. Miyahara. The minimal entropy martingale measures for geometric Lévy processes. *Finance and Stochastics*, 7(4):509–531, 2003.
- [Gen86] A. Genz. Fully symmetric interpolatory rules for multiple integrals. *SIAM Journal on Numerical Analysis*, 23(6):1273–1283, 1986.
- [GG98] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18(3-4):209–232, 1998.
- [GJ84] R. Geske and H. E. Johnson. The American put option valued analytically. *The Journal of Finance*, XXXIX(5):1511–1524, 1984.
- [GK96] A. Genz and B. D. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight. *Journal of Computational & Applied Mathematics*, 71:299–309, 1996.
- [Gre69] T. N. E. Greville. Introduction to spline functions. In T. N. E. Greville, editor, *Theory and Applications of Spline Functions*. Academic Press, New York, 1969.
- [GVL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 1996.
- [Hab70] S. Haber. Numerical evaluation of multiple integrals. *SIAM Review*, 12:481–526, 1970.
- [Hau97] E. G. Haug. *The Complete Guide To Option Pricing Formulas*. McGraw-Hill, 1997.
- [Hes93] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993.
- [HK79] J. Harrison and D. Kreps. Martingales and arbitrage in multiperiod security markets. *Journal of Economic Theory*, 2:381–408, 1979.
- [HM76] C. A. Hall and W. W. Meyer. Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16:105–122, 1976.

- [HP81] J. M. Harrison and S. R. Pliska. Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and their Applications*, 11:215–260, 1981.
- [HP83] J. M. Harrison and S. R. Pliska. A stochastic calculus model of continuous trading: Complete markets. *Stochastic Processes and their Applications*, 15:313–316, 1983.
- [HSY96] J. Huang, M. Subrahmanyam, and G. Yu. Pricing and hedging American options: A recursive integration method. *Review of Financial Studies*, 9:277–300, 1996.
- [Hul00] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall International Editions, Upper Saddle River, fourth edition, 2000.
- [IV05] R. Int-Veen. *Transformation zur Vermeidung numerischer Dispersion und Verfahren zur Steuerung der Zeitschrittweite für die numerische Optionsbewertung*. PhD thesis, Universität zu Köln, 2005.
- [Kim90] I. J. Kim. The analytic valuation of American options. *Review of Financial Studies*, 3:547–572, 1990.
- [Kou02] S. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.
- [Kra96] D. O. Kramkov. Optional decomposition of supermartingales and hedging contingent claims in incomplete security markets. *Probability Theory and Related Fields*, 105:459–479, 1996. cited in CoTa03 p.324f.
- [KVY] A. Q. M. Khaliq, D. A. Voss, and M. Yousuf. Pricing exotic options with L-stable Padé schemes. *to appear in the Journal of Banking and Finance*.
- [Kwo98] Y.-K. Kwok. *Mathematical Models of Financial Derivatives*. Springer, New York, 1998.
- [LS01] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.
- [Mat98] MathWorks. *MATLAB Application Program Interface Guide*. The MathWorks, Inc., 1998.
- [Mat05] K. Matsuda. Calibration of Lévy option pricing models: Application to S&P 500 futures option. Technical report, 2005. Electronically available: <http://tinyurl.com/22vczf>.

- [MCC98] D. B. Madan, P. P. Carr, and E. C. Chang. The variance gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.
- [Mer73] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics*, 4(1):141–183, 1973.
- [Mer76] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976.
- [MN92] W. R. Madych and S. A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70:94–114, 1992.
- [MNS05] A.-M. Matache, P.-A. Nitsche, and C. Schwab. Wavelet Galerkin pricing of American options on Lévy driven assets. *Quantitative Finance*, 5(4):403–424, 2005.
- [MS90] D. B. Madan and E. Seneta. The variance gamma (v.g.) model for share market returns. *Journal of Business*, 63(4):511–524, 1990.
- [Nel98] R. B. Nelsen. *An introduction to copulas*. Lecture Notes in Statistics. Springer, 1998.
- [NR97] R. Novak and K. Ritter. The curse of dimension and a universal method for numerical integration. In G. Nürnberger, J. W. Schmidt, and G. Walz, editors, *Multivariate Approximation and Splines*, pages 177–188. Birkhäuser, 1997.
- [O’S05] C. O’Sullivan. Path dependant option pricing under Lévy processes. *EFA 2005 Moscow Meetings Paper*, 2005. Electronically available: <http://ssrn.com/abstract=673424>.
- [Pag96] A. Pagan. The econometrics of financial markets. *Journal of Empirical Finance*, 3(1):15–102, 1996.
- [Par62] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.
- [Par77] M. Parkinson. Option pricing: the American put. *Journal of Business*, 50:21–36, 1977.
- [PFVS00] D. Pooley, P. A. Forsyth, K. Vetzal, and R. B. Simpson. Unstructured meshing for two asset barrier options. *Applied Mathematical Finance*, 7:33–60, 2000.

- [PS77] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximation on triangles. *ACM Transactions on Mathematical Software*, 3:316–325, 1977.
- [PTVF02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, 2nd edition, 2002.
- [Rud87] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 3rd edition, 1987.
- [Ryd97] T. H. Rydberg. The normal inverse Gaussian Lévy process: simulation and approximation. *Comm. Statist. Stochastic Models*, 13:887–910, 1997.
- [Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM Society for Industrial & Applied Mathematics, 2nd edition, 2003.
- [SB00] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer, 2000.
- [Sch96] R. Schaback. Approximation by radial basis functions with finitely many centers. *Constructive Approximation*, 12:331–340, 1996.
- [Sey02] R. Seydel. *Tools for Computational Finance*. Springer, 2nd edition, 2002.
- [Shr04] S. E. Shreve. *Stochastic Calculus for Finance I+II*. Springer, 2004.
- [Sil86] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman/Hall, 1986.
- [Smo63] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk. SSSR*, 4:240–243, 1963.
- [Sto99] J. Stoer. *Numerische Mathematik 1*. Springer, 8th edition, 1999.
- [Sul00] M. A. Sullivan. Valuing American put options using Gaussian quadrature. *The Review of Financial Studies*, 13(1):75–94, 2000.
- [Vas77] O. Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5:177–188, 1977.
- [Wen96] H. Wendland. *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger*. PhD thesis, University of Göttingen, 1996.
- [Wen05] H. Wendland. *Scattered data approximation*. Cambridge University Press, 2005.
- [Wil98] P. Willmott. *Derivatives*. John Wiley and Sons, Chichester, 1998.
- [Yoo01] J. Yoon. Interpolation by radial basis functions on sobolev space. *Journal of Approximation Theory*, 112(1):1–15, 2001.
- [ZJ96] S. Zhang and J. Jin. *Computation of special functions*. Wiley & Sons, 1996.

# Index

- American option, 1
- Asian options, 144
- B/S model, 91
- Bermudan option, 1
- binomial tree method, 112
- Black-Scholes model, 91
- characteristic exponent, 94
- chooser options, 143
- complete market, 3
- convolution methods, 29
- copula, 107
- cross options, 144
- cubic splines, 30
- density estimation, 108
- EMM, 5, 92
- equivalent martingale measure, 5
- European option, 1
- Gauss quadrature, 81
- Gaussian basis function, 68
- GBM, 89
- Halton sequence, 133
- hedging, 3
- Heston model, 91, 103
- hold value, 14
- incomplete market, 3
- Lévy models, 89
- Lévy process, 90
- log-price, 91
- log-returns, 91
- Merton model, 93
- NIG model, 100
- no-arbitrage principle, 3
- Normal-Inverse-Gaussian model, 100
- option, 1
- PDF, 10
- plain-vanilla option, 1
- pricing rule, 4
- put-call parity, 116
- RBF interpolation, 68
- RBF method, 65
- reduction principle, 14
- risk-neutral pricing theorem, 6
- risk-neutrality condition, 92
- Romberg quadrature, 33
- spline method, 30
- subordinator, 96
- tail index, 105
- thin plate splines, 59
- thin tails, 92
- validation, 110
- Variance-Gamma model, 95
- Vasicek model, 91
- VG model, 95, 96
- volatility smile, 92

# Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie – abgesehen von unten angegebenen Teilpublikationen – noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Herrn Prof. Dr. R. Seydel betreut worden.

Köln, im Februar 2007  
Sebastian Quecke