



TECHNISCHE UNIVERSITÄT MÜNCHEN  
FAKULTÄT FÜR MATHEMATIK  
LEHRSTUHL FÜR FINANZMATHEMATIK

# Complexity Reduction for Option Pricing

Parametric Problems and Methodological Risk

Mirco Mahlstedt

Vollständiger Abdruck der von der Fakultät für Mathematik der  
Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Naturwissenschaften (Dr. rer. nat.)**

genehmigten Dissertation.

Vorsitzende: Prof. Dr. Barbara Wohlmuth

Prüfer der Dissertation: 1. Prof. Dr. Kathrin Glau  
2. Prof. Dr. Peter Tankov (ENSAE ParisTech, Frankreich)  
3. Prof. Dr. Wim Schoutens (KU Leuven, Belgien)

Die Dissertation wurde am 25.04.2017 bei der Technischen Universität München  
eingereicht und durch die Fakultät für Mathematik am 26.06.2017 angenommen.



## Abstract

For financial institutions, fast and accurate computational methods for parametric asset models are essential. We start with a numerical investigation of a widely applied approach in the financial industry, the de-Americanization methodology. Here, the problem of calibrating to American option prices is reduced to calibrating to European options by translating American option data via binomial tree techniques into European prices. The results of this study identify scenarios in which the de-Americanization methodology performs well and in which de-Americanization leads into pitfalls. Therefore, the need of executing recurrent tasks such as pricing, calibration and risk assessment accurately and in real-time, sets the direction to complexity reduction. Via Chebyshev interpolation the recurrent nature of these tasks is exploited by polynomial interpolation in the parameter space. Identifying criteria for (sub)exponential convergence and deriving explicit error bounds enables to reduce run-times while maintaining accuracy. For the Chebyshev interpolation any option pricing technique can be applied for evaluating the function at the nodal points. With option pricing in mind, a new approach is pursued. The Chebyshev interpolation is combined with dynamic programming concepts. The resulting generality of this framework allows for various applications in mathematical finance and beyond our example of pricing American options.



## Zusammenfassung

Finanzinstitutionen stehen vor der Herausforderung, numerische Methoden zur parametrischen Optionspreisbewertung zu verwenden, die sowohl exakt als auch schnell sind. Zunächst wird eine in der Finanzindustrie verbreitete Methode, die de-Americanization Methode, untersucht. Bei dieser werden vor dem Starten des Kalibrierungsprozesses amerikanische Optionspreise mit Hilfe von Binomialbäumen in pseudo-europäische Optionspreise übersetzt. Damit wird die Kalibrierung an amerikanischen Optionen vereinfacht zu einer Kalibrierung an europäischen Optionen. Im Rahmen der empirischen Analyse wurden sowohl Szenarien identifiziert, in denen die vorgeschlagene de-Americanization Methode zuverlässige Ergebnisse liefert, als auch Szenarien, in denen die Methode zu nicht korrekten Ergebnissen führt. Der Bedarf, immer wiederkehrende, parameterabhängige Aufgaben - Optionspreisbewertung, Kalibrierung und Risikobewertungen - sowohl genau als auch in Echtzeit auszuwerten zu können, motivieren den Schritt zu Vereinfachungstechniken, die die Komplexität genau dieser Aufgaben reduzieren. Die Chebyshev Interpolation löst die wiederkehrende Natur durch eine Polynominterpolation im entsprechenden Parameterraum. Durch einen Kriterienkatalog für exponentielle Konvergenz und durch explizite Fehlerschranken ermöglicht diese Methode eine Reduzierung der Laufzeiten bei gleichzeitigem Erhalt der Genauigkeit. Darüber hinaus verknüpfen wir die Chebyshev Interpolation mit der dynamischen Programmierung, um dynamische Probleme effizient lösen zu können. Das resultierende Grundgerüst ist so allgemein konzipiert, dass es in vielen Anwendungsbereichen der Finanzmathematik verwendet werden kann.



## Acknowledgements

First and foremost, I sincerely thank my supervisor Kathrin Glau. Without her genuine guidance, patience and support, this thesis would not have been possible. Up to the present day, I am each day anew impressed by her passion for research, by her open minded attitude, by her creativity in finding new ideas and by her unconditional support.

Moreover, I'd like to thank my co-authors Olena Burkovska, Marcos Escobar, Maximilan Gaß, Kathrin Glau, Maximilan Mair, Sven Panz, Christian Pötz, Wim Schoutens, Barbara Wohlmuth and Rudi Zagst for the intensive discussions and inputs from different point of views.

My deep gratitude goes to Rudi Zagst and Matthias Scherer, who encouraged and supported me since the start of master studies. I appreciate their support and goodwill for creating a wonderful working atmosphere at the chair of mathematical finance and for taking care of the needs of each individual. I especially thank Rudi Zagst for making my first contacts with research during my master studies to a very enjoyable and positive experience.

I thank the management board of the KPMG Center of Excellence in Risk Management. Their financing created my position and made everything possible. Remarkably, Franz Lorenz, Matthias Mayer and Daniel Sommer not only established a bridge between industry and academia, but also live and breath the exchange between both worlds. Their curiosity, insights and support have been very encouraging for me. I deeply appreciate the freedom regarding research directions and I'm very thankful for the two internships I could do with KPMG.

I'm very grateful and I deeply thank all my colleagues during my time at the chair, namely German Bernhart, Tobias Bienek, David Criens, Susanne Deuke, Lexuri Fernández, Tim Friederich, Maximilan Gaß, Bettina Haas, Peter Hieber, Karl Hofmann, Amelie Hüttner, Miriam Jaser, Asma Khedher, Julia Kraus, Daniel Krause, Mikhail Krayzler, Andreas Lichtenstern, Daniël Linders, Maximilan Mair, Aleksey Min, Daniela Neykova, Christian Pötz, Franz Ramsauer, Oliver Schlick, Steffen Schenk, Lorenz Schneider, Thorsten Schulz, Danilea Selch, Natalia Shenkman, Martin Smaga, Markus Wahl and Bin Zou.

Last but not least, I thank my parents and brother for their steady support during my complete life, and for making my little hometown in the north of Germany to a place I always visit with a big smile.

Mirco Mahlstedt

*April 23, 2017*



# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>Mathematical Preliminaries</b>	<b>17</b>
2.1	Asset Price Models and Option Pricing . . . . .	17
2.2	Three Ways to Derive the Option Price . . . . .	20
2.2.1	Connection to Solutions of Partial Differential Equations . . . . .	20
2.2.2	Fourier pricing . . . . .	23
2.2.3	Monte-Carlo simulation . . . . .	25
2.3	Miscellaneous . . . . .	26
<b>3</b>	<b>Numerical Investigation of the de-Americanization Method</b>	<b>30</b>
3.1	De-Americanization Methodology . . . . .	33
3.2	Pricing Methodology . . . . .	37
3.2.1	Pricing PDE . . . . .	38
3.2.2	Variational Formulation . . . . .	39
3.3	Numerical Study of the effects of de-Americanization . . . . .	42
3.3.1	Discretization . . . . .	42
3.3.2	Effects of de-Americanization on Pricing . . . . .	42
3.3.3	Effects of de-Americanization on Calibration to Synthetic Data . . . . .	46
3.3.4	Effects of de-Americanization on Calibration to Market Data . . . . .	49
3.3.5	Effects of de-Americanization in Pricing Exotic Options . . . . .	51
3.4	Conclusion . . . . .	52
3.5	Outlook: The Reduced Basis Method . . . . .	54
3.6	Excursion: The Regularized Heston Model . . . . .	55
3.6.1	Existence and Strong Solution in the Bounded Domain $I = (\epsilon, \bar{V})$ . . . . .	56
3.6.2	Convergence . . . . .	61
<b>4</b>	<b>Chebyshev Polynomial Interpolation Method</b>	<b>67</b>
4.1	Chebyshev Polynomial Interpolation . . . . .	67
4.1.1	Chebyshev Polynomials . . . . .	69
4.1.2	Chebyshev Polynomial Interpolation . . . . .	74
4.1.3	Multivariate Chebyshev Interpolation . . . . .	76
4.2	Convergence Results of the Chebyshev Interpolation Method . . . . .	77
4.2.1	Convergence Results Including the Derivatives . . . . .	95
4.3	Chebyshev Interpolation Method for Parametric Option Pricing . . . . .	99
4.3.1	Exponential Convergence of Chebyshev Interpolation for POP . . . . .	99

*Contents*

4.4	Numerical Experiments for Parametric Option Pricing . . . . .	103
4.4.1	European Options . . . . .	104
4.4.2	Basket and Path-dependent Options . . . . .	104
4.4.3	Study of the Gain in Efficiency . . . . .	110
4.4.4	Relation to Advanced Monte-Carlo Techniques . . . . .	115
4.5	Conclusion and Outlook . . . . .	122
<b>5</b>	<b>Dynamic Programming Framework with Chebyshev Interpolation</b>	<b>123</b>
5.1	Derivation of Conditional Expectations . . . . .	126
5.2	Dynamic Chebyshev in the Case of Analyticity . . . . .	130
5.2.1	Description of Algorithms . . . . .	131
5.2.2	Error Analysis . . . . .	132
5.3	Solutions for Kinks and Discontinuities . . . . .	137
5.3.1	Splitting of the Domain . . . . .	137
5.3.2	Mollifier to the Function $g(t,x)$ . . . . .	143
5.4	Alternative Approximation of General Moments in the Pre-Computation .	144
5.5	Combination of Empirical Interpolation with Dynamic Chebyshev . . . . .	152
5.6	Numerical Experiments - Example Bermudan and American options . . . . .	162
5.7	Conclusion . . . . .	170
<b>A</b>	<b>Detailed Results for Effects of de-Americanization on Pricing</b>	<b>173</b>
	<b>Bibliography</b>	<b>177</b>

# 1 Introduction

In mathematics the complicated  
things are reduced to simple things.  
So it is in painting.

---

Thomas Eakins

For financial institutions with a strong dedication to trading or assessment of financial derivatives and risk management, numerous financial quantities have to be computed on a daily basis. Here, we focus on option prices, sensitivities and risk measures for products in different models and for varying parameter constellations. Growing market activities and fast-paced trading environments require that these evaluations are done in almost real time. Thus, fast and accurate computational methods for parametric stock price models are essential.

Besides market environments, the model sophistication has risen tremendously since the seminal work of Black and Scholes (1973) and Merton (1973). Stochastic volatility and Lévy models, as well as models based on further classes of stochastic processes, have been developed to deal with shortcomings of the Black&Scholes model and to capture market observations more appropriately, such as non-constant volatilities and jumps. For stock models, see Heston (1993), Eberlein et al. (1998), Duffie et al. (2003) and Cuchiero et al. (2012).

The usefulness of a pricing model critically depends on how well it captures the relevant aspects of market reality in its numerical implementation. Exploiting new ways to deal with the rising computational complexity therefore supports the evolution of pricing models and touches a core concern of present mathematical finance. A large body of computational tasks in finance need to be repeatedly performed in real time for a varying set of parameters. Prominent examples are option pricing and hedging of different option sensitivities, e.g. delta and vega, which also need to be calculated in real time. In particular for optimization routines arising in model calibration, and in the context of risk control and assessment, such as for quantification and monitoring of risk measures.

In a nutshell, trade-offs have to be found between accuracy and computational costs, especially with the generally rising complexity of the problems. Which kind of complexity reduction techniques can be applied? In Chapter 3, we take calibrating American options as an example. For single-stock options, only market data for American options is available and so, American options have to be used to calibrate a stock price model. In contrast

## 1 Introduction

to European options, which give the option-holder the right to exercise the option at maturity, American options allow the option-holder to exercise the option once at any time up to the maturity. Thus, American options are so-called path-dependent options and the pricing, especially under advanced models, relies on computationally-expensive numerical techniques, such as the Monte Carlo simulation or partial (integro) differential methods. Naturally, it is much faster to calibrate a model to European options than to American options. Especially since there exists a variety of (semi-)closed pricing formulas for European options. This is applied in the de-Americanization methodology, as for instance mentioned in Carr and Wu (2010), which we will investigate in the third chapter. Basically, before any calibration is applied, the American options are replaced by European options using binomial tree techniques. Our empirical study of the de-Americanization methodology shows that this method tends to perform well in several scenarios. However, in some scenarios, significant errors occur when compared to a direct calibration to American options. The major drawback of the de-Americanization methodology is that no error control is given.

The problems from calibrating to American options serve as an example and motivate our investigations of complexity reduction methods in finance. Our approach in the following is to systemically exploit the recurrent nature of parametric computational problems in finance in order to gain efficiency in combination with error convergence results. Our main focus here is parametric option pricing. In the literature, parametric option pricing problems have largely been addressed by applying Fourier techniques following Carr and Madan (1999) and Raible (2000). The focus is on adopting fast Fourier transform (FFT) methods and variants for option pricing. For pricing European options with FFT, we refer to Lee (2004). Further developments are, for instance, provided by Lord et al. (2008) for early exercise options and by Feng and Linetsky (2008) and Kudryavtsev and Levendorskiĭ (2009) for barrier options. Another path to efficiently handle large parameter sets is built on solving parametrized partial differential equations, the reduced basis methods. Sachs and Schu (2010), Cont et al. (2011), Pironneau (2011) and Haasdonk et al. (2013) and Burkovska et al. (2015) applied this approach to price European, and American, plain vanilla options and European baskets. Looking at both methods, FFT methods can be advantageous when the prices are required in a large number of Fourier variables, e.g. for a large set of strikes of European plain vanillas. Reduced basis methods, on the other hand, when an accurate PDE solver is readily available. We continue by giving an example of how the reduced basis method is applied to the calibration of American options in the Heston stochastic volatility model, and how the results compare to the results of the de-Americanization methodology. Summarizing with respect to parametric option pricing, the reduced basis method, as well as FFT method, reveal an immense complexity reduction potential by targeting parameter dependence. Both techniques have in common that they are add-ons to the functional architecture of the underlying pricing technique. In Figure 1.1, we visually illustrate this add-on feature.

Our following investigations are driven by the observation that, naturally, in financial institutions a diversity of models, a multitude of option types, and, as a consequence,

## 1 Introduction

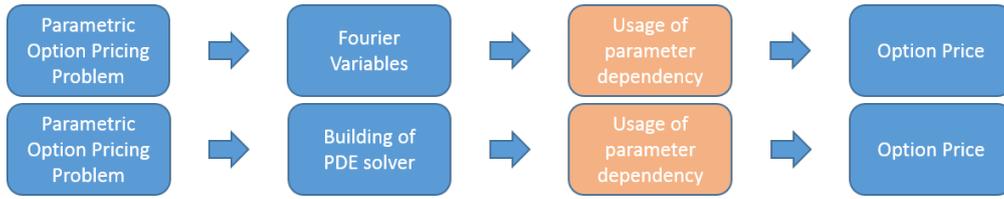


Figure 1.1: Schematic overview: Both option pricing techniques, FFT (add-on to Fourier pricing) and reduced basis (add-on to a PDE technique), exploit the parameter dependency as an add-on to the functional architecture of the underlying pricing technique.

a wide variety of underlying pricing techniques, are used simultaneously to cope with different queries. In contrast to the usage of parameter dependency outlined in Figure 1.1, we introduce polynomial interpolation of option prices in the parameter space as a complexity reduction technique. The resulting procedure splits into two phases: Pre-computation and real-time evaluation. The first one is also called *offline-phase* while the second is also called *online-phase*. In the pre-computation phase, the prices are computed for some fixed parameter configurations, namely the interpolation nodes. Here, any appropriate pricing method, for instance, based on the Fourier, PDE or even Monte-Carlo techniques, can be chosen. Then, the online-phase consists of the evaluation of the interpolation. Provided that the evaluation of the interpolation is faster than the benchmark tool, the scheme permits a gain in efficiency in all cases where accuracy can be maintained. A visualization of this approach is shown in Figure 1.2. We see

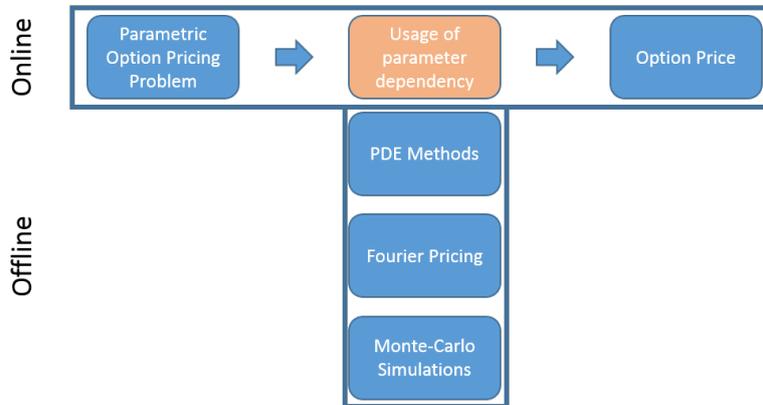


Figure 1.2: Idea of exploiting parameter dependencies independently of the underlying pricing technique. The answer in this thesis will be Chebyshev polynomial interpolation. The pricing techniques of PDE methods, Fourier pricing and Monte-Carlo simulation are only applied during the offline phase.

two use-cases for this approach. Firstly, in comparison to the benchmarking pricing routine, the online evaluation as an evaluation of a polynomial will be rather fast and

## 1 Introduction

can potentially outweigh the expensive pre-computation phase. This may especially be the case in optimization routines in which the same problem for several parameter combinations has to be solved rather frequently. Secondly, even for computing only a few prices, this approach can be beneficial because it allows the application of the computationally-costly pre-computation phase in idle times.

Regarding the choice of polynomial interpolation type, it is well-known that the efficiency depends on the degree of regularity of the approximated function. In Chapter 4, we focus theoretically on the pricing of European (basket) options. In Gaß et al. (2016), we investigate the regularity of the option prices as functions of the parameters and find that these functions are indeed analytic for a large set of option types, models and parameters. We observe that parameters of interest often range within bounded intervals. Chebyshev interpolation has proven to be extremely useful for applications in such diverse fields as physics, engineering, statistics and economics. Nevertheless, for pricing tasks in mathematical finance, Chebyshev interpolation still seems to be rarely used and its potential is yet to be unfolded. In the multivariate case, we choose a tensorized version of Chebyshev interpolation. Pistorius and Stolte (2012) use Chebyshev interpolation of Black&Scholes prices in the volatility as an intermediate step to derive a pricing methodology for a time-changed model. Independently from us, Pachon (2016) recently proposed Chebyshev interpolation as a quadrature rule for the computation of option prices with a Fourier-type representation, which is comparable to the cosine method of Fang and Oosterlee (2008).

The focus in Chapter 4 is on parametric option pricing and on European options. Numerical experiments have shown that the Chebyshev interpolation can also be beneficial for path-dependent options, such as American options. In Chapter 5, we provide a theoretical framework that includes American option pricing, Chebyshev interpolation and error convergence results. As shown in Peskir and Shiryaev (2006), American option pricing is an optimal stopping problem that can be described by a dynamic programming principle. Our approach is the usage of Chebyshev interpolation within the dynamic programming principle to establish a complexity reduction for solving them. Moreover, we derive error convergence results based on the results of the Chebyshev interpolation. Whereas in Chapter 4, the focus is on parametric problems, in the dynamic programming framework in Chapter 5, the Chebyshev interpolation is not applied to the parameters, but solely to the value of the underlying during the backward time stepping scheme. The generality of this dynamic programming framework allows for various applications in the dynamic programming area and therewith for applications in mathematical finance, and is not limited to pricing American options. Additionally, we present ideas to connect the dynamic Chebyshev approach with empirical interpolation techniques to incorporate the parameter dependency, too.

The main contributions of this thesis can be summarized in the following way.

**Chapter 3** In this chapter, we present the de-Americanization methodology and empirically investigate this methodology for the CEV model. To do so, we implement a finite element solver for the CEV model and establish a calibration to synthetic, as well as to market data. We identify scenarios in which the methodology works rather well, but also present scenarios in which the methodology leads to high errors. These results are separately presented in Burkovska et al. (2016), of which I am the leading author, complemented by results for the Heston and the Merton models.

Moreover, we give an outlook on the calibration of American options in the Heston model with the reduced basis method, which is done in Burkovska et al. (2016b). Lastly, we introduce the regularized Heston model as stochastic volatility model with bounded coefficients. These are required by standard Feynman-Kac results to establish the bridge between option price and PDE solution. We conclude by presenting convergence results from the regularized Heston to the Heston model.

**Chapter 4** We present the Chebyshev polynomial interpolation technique and provide a new and improved error bound for analytic functions for the tensorized multivariate extension. We provide accessible sufficient conditions on options and models that guarantee an asymptotic error decay of order  $O(\varrho^{-\frac{D}{\sqrt{N}}})$  in the total number  $N$  of interpolation nodes, where  $\varrho > 1$  is given by the domain of analyticity and  $D$  is the number of varying parameters. In Glau and Mahlstedt (2016), of which I am the leading author, the improved convergence results for the analytic case are presented.

The rest of the chapter is based on Gaß et al. (2016) and I present the parts to which I provided a significant contribution. Empirically, for multivariate basket and path-dependent options, we use Monte-Carlo as a reference method and highlight the quality of the Chebyshev interpolation method beyond the scope of the theoretically-investigated European options. Moreover, we embed the Chebyshev interpolation with Monte-Carlo at the nodal points into the (multilevel) parametric Monte-Carlo framework and show, that for a wide and important range of problems, the Chebyshev method turns out to be more efficient than the parametric multilevel Monte-Carlo.

**Chapter 5** This chapter is based on Glau et al. (2017a) and Glau et al. (2017b) and I present the parts to which I provided a significant contribution. We combine the Chebyshev interpolation with the dynamic programming principle to establish a complexity reduction for solving them. Key idea is a reduction of the occurring conditional expectations to conditional expectations of Chebyshev polynomials. We illustrate the generality of this framework and provide several approaches to derive the conditional expectations of Chebyshev polynomials. In the dynamic programming framework, the Chebyshev interpolation is not applied to the parameters, but to the underlying value itself. To tackle parametric problems here, we combine the framework with empirical interpolation in the parameters.

## *1 Introduction*

## 2 Mathematical Preliminaries

We are servants rather than masters  
in mathematics.

---

Charles Hermite

In this chapter, we present some general mathematical preliminaries on which the thesis relies. As outlined in the introduction, a major part of the thesis is related to option pricing. We will see that within a risk-neutral valuation framework the calculation of an option price is basically the derivation of an expectation, an expectation of a payoff function on a stochastic process. We illustrate the models used in this thesis and then we present three concepts for the derivation of this expectation, the option price. First we show the connection to partial differential equations and present the finite element method in detail. Second, we illustrate the concept of Fourier pricing. As third point, we present the Monte-Carlo method as simulation technique. Lastly, in this chapter, we present some further concepts which will be applied within this thesis.

For basic in probability spaces, stochastic processes and stochastic differential equations, we refer the reader to Musiela and Rutkowski (2006), Øksendal (2003) and Zagst (2002).

### 2.1 Asset Price Models and Option Pricing

We start with the description of asset price models. The asset price dynamics  $(S_\tau)_{\tau \geq 0}$  are governed by a stochastic differential equation (SDE). In this thesis, we introduce the Black&Scholes model, the CEV model, the Heston model and the Merton model. All of these models are described by a SDE of the form

$$dS_\tau = rS_\tau d\tau + \sigma(S, \tau)S_\tau dW_\tau + S_{\tau-} dJ_\tau, \quad S_0 = s \geq 0, \quad (2.1a)$$

$$J_\tau = \sum_{i=0}^{N_\tau} Y_i, \quad (2.1b)$$

with  $W_\tau$  a standard Wiener process,  $r$  the risk-free interest rate and a volatility function  $\sigma(S, \tau)$ . The jump part  $(J_\tau)_{\tau \geq 0}$  is a compound Poisson process with intensity  $\lambda \geq 0$  and

## 2 Mathematical Preliminaries

independent identically distributed jumps  $Y_i$ ,  $i \in \mathbb{N}$ , that are independent of the Poisson process  $(N_\tau)_{\tau \geq 0}$ . The Poisson process and the Wiener process are also independent.

If we let the diffusion coefficient  $\sigma(S, \tau)$  be constant and the jump intensity  $\lambda = 0$ , then we are in the classical Black&Scholes model of Black and Scholes (1973) and Merton (1973).

As an example of a local volatility model, we begin by presenting the CEV model, which was introduced by Cox (1975). Here, the local volatility is assumed to be a deterministic function of the asset price for the process in (2.1),  $\sigma(S, \tau) = \sigma S_\tau^{\zeta-1}$ ,  $0 < \zeta < 1$ ,  $\sigma > 0$  and  $\lambda = 0$ .

As an example of a stochastic volatility model, we use the model proposed by Heston (1993). In contrast to the CEV model, the stochastic volatility is driven by a second Brownian motion  $\widetilde{W}_\tau$  whose correlation with  $W_\tau$  is described by a correlation parameter  $\rho \in [-1, 1]$ , and the model is based on the dynamics of both the stock price (2.1), with jump intensity  $\lambda = 0$ , and the variance  $v_\tau$  (2.2),

$$dv_\tau = \kappa(\gamma - v_\tau)dt + \xi\sqrt{v_\tau}d\widetilde{W}_\tau, \quad (2.2)$$

with  $\sigma(S, \tau) = \sqrt{v_\tau}$ , mean variance  $\gamma > 0$ , rate of mean reversion  $\kappa > 0$  and volatility of volatility  $\xi > 0$ . Jumps are not included in either of the CEV or Heston models.

The Merton model includes jumps. The log-asset price process is not exclusively driven by a Brownian motion, but instead follows a jump-diffusion process. Thus, in the model of Merton (1976), the volatility of the asset process is still assumed to be constant, i.e. for all  $S > 0$  and for all  $\tau > 0$  it holds  $\sigma(S, \tau) \equiv \sigma > 0$ . But being a jump diffusion model, the jump intensity  $\lambda > 0$  is positive and  $N_t \sim \text{Poiss}(\lambda t)$ . The jumps are taken to be independent normally distributed random variables,  $Y_i \sim \mathcal{N}(\alpha, \beta^2)$  with expected jump size  $\alpha \in \mathbb{R}$  and standard deviation  $\beta > 0$ .

After the description of the asset or underlying as a stochastic process, we now focus on option pricing. An option is a derivative whose payoff depends on the performance of the underlying  $S$ . So-called plain vanilla European call or put options have at maturity  $T$ , for a pre-specified strike  $K$ , the payoff  $\max\{S_T - K, 0\}$  (call) or  $\max\{K - S_T, 0\}$  (put). Here, the payoff does only depend on the value of the underlying at maturity  $T$ . American call or put options have the same payoff function than their European counterpart, however the option holder has the right to exercise the option at any time up to maturity  $T$ . In this case, we refer to path-dependent options.

The option is determined by the risk-neutral valuation theory, see Bingham and Kiesel (2004). Here, the basic assumption is that in the determination of the option price the individual risk preferences of a potential investor, may she either be risk-seeking or risk-averse, are not considered. Already implicitly defined by the terminology risk-neutral, for the option price only the expected payoff of the option is important. Furthermore,

## 2 Mathematical Preliminaries

to be consistent with this risk-neutral perspective, the expectation is taken under a measure, under which the underlying process is, in expectation, evolving like the risk-free asset. In other words, the with the risk-free interest rate discounted underlying process is a martingale. Bingham and Kiesel (2004) refer to this measure as strong equivalent martingale measure.

Embedding the risk-neutral valuation theory, in the following the option price at time  $t$ , for an underlying  $S$  described by (2.1), with a payoff function  $g$ , on a filtered probability space  $(\Omega, \mathcal{F}, P, \mathbb{F})$  with filtration  $\mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$ , under a strong equivalent martingale measure  $\mathbb{Q}$  is given by

$$E_{\mathbb{Q}}[e^{-r(T-t)}g(S_T)|\mathcal{F}_t]. \quad (2.3)$$

For notational ease, we use in the following  $E[\cdot]$  for the expectation under the risk-neutral measure,  $E_{\mathbb{Q}}[\cdot]$ .

Before we present three ways to derive this expectation, we introduce the definition of strong solutions based on the following SDE in the one dimensional case

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t, \quad (2.4)$$

where  $b(t, x)$  and  $\sigma(t, x)$  are Borel-measurable functions from  $[0, \infty) \times \mathbb{R} \rightarrow \mathbb{R}$ .

**Definition 2.1.1** (Strong solution). (*Karatzas and Shreve, 1996, Definition 2.1, p. 285*) *A strong solution of the stochastic differential equation (2.4) on the given probability space  $(\Omega, \mathcal{F}, P, \mathbb{F})$  with filtration  $\mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$  and with respect to the fixed Brownian motion  $W$  and initial condition  $\zeta$ , is a process  $X = \{X_t; 0 \leq t < \infty\}$  with continuous sample paths and with the following properties:*

- (i)  $X$  is adapted to the filtration  $\mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$
- (ii)  $P[X_0 = \zeta] = 1$
- (iii)  $P[\int_0^t \{|b(s, X_s)| + \sigma^2(x, X_s)\}ds < \infty] = 1, 0 \leq t < \infty$
- (iv) the integral version of (2.4)

$$X_t = X_0 + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)dW_s; 0 \leq t < \infty,$$

holds almost surely.

**Definition 2.1.2** (Strong uniqueness). (*Karatzas and Shreve, 1996, Definition 5.2.3, p. 286*) *Let the drift vector  $b(t, x)$  and dispersion matrix  $\sigma(t, x)$  be given. Suppose that, whenever  $W$  is a 1-dimensional Brownian motion on some  $(\Omega, \mathcal{F}, P)$ ,  $\zeta$  is an independent, 1-dimensional random vector,  $\{\mathcal{F}_t\}$  is an augmented filtration, and  $X, \tilde{X}$  are two strong solutions of (2.4) relative to  $W$  with initial condition  $\zeta$ , then  $P[X_t = \tilde{X}_t; 0 \leq t < \infty] = 1$ . Under these conditions, we say that strong uniqueness holds for the pair  $(b, \sigma)$ .*

## 2 Mathematical Preliminaries

After the introduction of strong, unique solutions, we present the Proposition of Yamada and Watanabe as illustrated in Karatzas and Shreve (1996):

**Proposition 2.1.3.** *(Karatzas and Shreve, 1996, Proposition 2.13, p. 291) Let us suppose that the coefficients of the one-dimensional equation*

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

*satisfy the conditions*

$$|b(t, x) - b(t, y)| \leq K|x - y|, \quad (2.5)$$

$$|\sigma(t, x) - \sigma(t, y)| \leq h(|x - y|), \quad (2.6)$$

*for every  $0 \leq t < \infty$  and  $x \in \mathbb{R}, y \in \mathbb{R}$ , where  $K$  is a positive constant and  $h : [0, \infty) \rightarrow [0, \infty)$  is a strictly increasing function with  $h(0) = 0$  and for all  $\epsilon > 0$ ,*

$$\int_{(0, \epsilon)} h^{-2}(u)du = \infty. \quad (2.7)$$

*Then strong uniqueness holds for the equation (2.4).*

## 2.2 Three Ways to Derive the Option Price

The derivation of option prices is in the center of this thesis. It is well known that all roads lead to Rome and similarly, there are several ways to derive the option price in (2.3). In this section, we present three ways. First, we show the expectation is connected to the solution of a partial differential equation.

### 2.2.1 Connection to Solutions of Partial Differential Equations

Naturally the question arises of how the stochastic representation can be connected with the solution of a PDE. Karatzas and Shreve (1996) start by considering a solution to the stochastic integral equation

$$X_s^{t,x} = x + \int_t^s b(\theta, X_\theta^{t,x})d\theta + \int_t^s \Sigma(\theta, X_\theta^{t,x})dW_\theta, \quad t \leq s < \infty. \quad (2.8)$$

This representation is connected to our SDE in 2.1 by considering the part  $b(\theta, X_\theta^{t,x})$  as drift coefficient and  $\Sigma(\theta, X_\theta^{t,x})$  as diffusion coefficient. Here, we do not consider jumps and basically set the jump intensity  $\lambda = 0$ . Following Karatzas and Shreve (1996), the connection between the solution of an SDE and the solution of a partial differential equation is stated in Theorem 2.2.5.

In order to provide this theorem, we first define the second-order differential operator.

## 2 Mathematical Preliminaries

**Definition 2.2.1** (Second-order differential operator). (*Karatzas and Shreve, 1996, p. 312*) Suppose  $(X^{(t,x)}, W), (\Omega, \mathcal{F}, \mathcal{P}), \{\mathcal{F}_t\}$  is a weak solution to the stochastic differential equation  $dX_t = b(t, X_t)dt + \Sigma(t, X_t)dW_t$ . For every  $t \geq 0$ , we introduce the second-order differential operator

$$(\mathcal{A}_t f)(x) := \frac{1}{2} \sum_{i=1}^d \sum_{k=1}^d a_{ik}(t, x) \frac{\partial^2 f(x)}{\partial x_i \partial x_k} + \sum_{i=1}^d b_i(t, x) \frac{\partial f(x)}{\partial x_i}, \quad f \in C^2(\mathbb{R}^d), \quad (2.9)$$

where  $a_{ik}(t, x)$  are the components of the diffusion matrix, i.e

$$a_{ik}(t, x) := \sum_{j=1}^r \Sigma_{ij}(t, x) \Sigma_{kj}(t, x).$$

Note that this notation requires a definition of the SDE as follows in component-wise notation  $dX_t^{(i)} = b_i(t, X_t)dt + \sum_{j=1}^r \Sigma_{ij}(t, X_t)dW_t^{(j)}$ .

We will see that the connection between the solution of an SDE and the solution of a partial differential equation is based in the existence of weak solutions and uniqueness in the probability of law. When does a weak solution exist and what does unique in the sense of probability law mean? First, we state the definition of a weak solution.

**Definition 2.2.2** (Weak solution). (*Karatzas and Shreve, 1996, Definition 5.3.1*) A weak solution of equation (2.8) is a triple  $(X^{(t,x)}, W), (\Omega, \mathcal{F}, \mathcal{P}), \{\mathcal{F}_s\}$ , where

- (i)  $(\Omega, \mathcal{F}, \mathcal{P})$  is a probability space, and  $\{\mathcal{F}_s\}$  is a filtration of sub- $\sigma$ -fields of  $\mathcal{F}$  satisfying the usual conditions,
- (ii)  $X = \{X_s, \mathcal{F}_s; 0 \leq s < \infty\}$  is a continuous, adapted  $\mathbb{R}^d$ -valued process,  $W = \{W_s, \mathcal{F}_s; 0 \leq s < \infty\}$  is an  $r$ -dimensional Brownian motion,
- (iii)  $\mathcal{P}[\int_0^s |b_i(t, X_t)| + \sigma_{ij}^2(t, X_t)dt < \infty] = 1$  holds for every  $1 \leq i \leq d, 1 \leq j \leq r$  and  $0 \leq s < \infty$ ,
- (iv) The integral version (2.8) of the SDE (2.1) holds almost surely.

After the definition of a weak solution, we immediately refer to the following theorem of Skorokhod (1965), which provides criteria for the existence of a weak solution. We state the version given in Karatzas and Shreve (1996).

**Theorem 2.2.3.** (*Karatzas and Shreve, 1996, Theorem 5.4.22*) Consider the stochastic differential equation

$$dX_t = b(X_t)dt + \Sigma(X_t)dW_t, \quad (2.10)$$

## 2 Mathematical Preliminaries

where the coefficients  $b_i, \Sigma_{ij} : \mathbb{R}^d \rightarrow \mathbb{R}$  are bounded and continuous functions. Corresponding to every initial distribution  $\mu$  on  $\mathcal{B}(\mathbb{R}^d)$  with

$$\int_{\mathbb{R}^d} \|x\|^{2m} \mu(dx) < \infty, \quad \text{for some } m > 1,$$

there exists a weak solution to (2.10).

Finally, we state the definition of uniqueness in the sense of probability law.

**Definition 2.2.4** (Uniqueness in the sense of probability law). (*Karatzas and Shreve, 1996, Definition 5.3.4*) We say that uniqueness in the sense of probability law holds for (2.8) if, for any two weak solutions  $(X^{(t,x)}, W), (\Omega, \mathcal{F}, \mathcal{P}), \{\mathcal{F}_s\}$  and  $(\tilde{X}^{(t,x)}, \tilde{W}), (\tilde{\Omega}, \tilde{\mathcal{F}}, \tilde{\mathcal{P}}), \{\tilde{\mathcal{F}}_s\}$ , with the same initial distribution, i.e.,

$$\mathcal{P}[X_0 \in \Gamma] = \tilde{\mathcal{P}}[\tilde{X}_0 \in \Gamma], \quad \text{for all } \Gamma \in \mathcal{B}(\mathbb{R}^d),$$

the two processes  $X$  and  $\tilde{X}$  have the same law.

To prove this uniqueness, we refer the reader further to Karatzas and Shreve (1996). Important in this section is the connection established by Theorem 2.2.5 to the solution of partial differential equations.

**Theorem 2.2.5.** (*Karatzas and Shreve, 1996, Theorem 5.7.6*) Under the Assumptions

- the coefficients  $b_i(t, x), \Sigma_{ij}(t, x) : [0, \infty) \times \mathbb{R}^d \rightarrow \mathbb{R}$  of 2.8 are continuous and satisfy the linear growth condition  $\|b(t, x)\|^2 + \|\Sigma(t, x)\|^2 \leq K^2(1 + \|x\|^2)$  for every  $0 \leq t < \infty, x \in \mathbb{R}^d$ , where  $K$  is a positive constant,
- the equation (2.8) has a weak solution  $(X^{(t,x)}, W), (\Omega, \mathcal{F}, \mathcal{P}), \{\mathcal{F}_s\}$  for every pair  $(t, x)$ ,
- this solution is unique in the sense of probability law,
- With an arbitrary but fixed  $T > 0$  and appropriate constants  $L > 0, \lambda \geq 1$  we consider functions  $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ ,  $g(t, x) : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$  and  $k(t, x) : [0, T] \times \mathbb{R}^d \rightarrow [0, \infty)$  which are continuous and satisfy

$$(i) |f(x)| \leq L(1 + \|x\|^{2\lambda}) \quad \text{or} \quad (ii) f(x) \geq 0; \quad \forall x \in \mathbb{R}^d \quad (2.11)$$

as well as

$$(iii) |g(t, x)| \leq L(1 + \|x\|^{2\lambda}) \quad \text{or} \quad (iv) g(t, x) \geq 0; \quad \forall 0 \leq t \leq T, x \in \mathbb{R}^d, \quad (2.12)$$

## 2 Mathematical Preliminaries

suppose that  $v(t, x) : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  is continuous, is of class  $C^{1,2}([0, T] \times \mathbb{R}^d)$  and satisfies the Cauchy problem

$$-\frac{\partial v}{\partial t} + kv = \mathcal{A}_t v + g, \quad \text{in } [0, T] \times \mathbb{R}^d, \quad (2.13)$$

$$v(T, x) = f(x), \quad x \in \mathbb{R}^d, \quad (2.14)$$

as well as the polynomial growth condition

$$\max_{0 \leq t \leq T} |v(t, x)| \leq M(1 + \|x\|^{2\mu}), \quad x \in \mathbb{R}^d, \quad (2.15)$$

for some  $M > 0, \mu \geq 1$ . Then  $v(t, x)$  admits the stochastic representation

$$v(t, x) = E^{t,x} \left[ f(X_T) \exp \left( - \int_t^T k(\theta, X_\theta) d\theta \right) + \int_t^T g(s, X_s) \exp \left( - \int_t^s k(\theta, X_\theta) d\theta \right) ds \right]$$

on  $[0, T] \times \mathbb{R}^d$ , in particular, such a solution is unique.

In Section 3.2, we present a specific technique to solve a partial differential equation for (American) options in the CEV model, namely the finite element method. As an outlook, we refer to the standard result regarding the existence of a weak solution in Theorem 2.2.3. There, boundedness of the coefficients of the SDE is required. For the Heston model, this is not satisfied because the stochastic process describing the stochastic volatility itself is unbounded and therefore, as a coefficient for the underlying price process unbounded, too. This has been our motivation to introduce a regularized Heston model in Section 3.6.2 with bounded coefficients. A second motivation is that the resulting PDE then has nicer properties.

### 2.2.2 Fourier pricing

The conditional expectation in (2.3) can be derived by solving an integral. Here, we introduce the concept of Fourier transforms. We will work here with the following definition of the Fourier transform.

**Definition 2.2.6** (Fourier transform). *Let a function  $f$  be in  $L^1(\mathbb{R})$ . Then, we define the Fourier transform  $\hat{f}$  as follows,*

$$\hat{f}(z) = \int_{-\infty}^{\infty} e^{izx} f(x) dx.$$

As the following lemma shows, the original function  $f$  can be expressed by its Fourier transform.

## 2 Mathematical Preliminaries

**Lemma 2.2.7** (Fourier inversion). (*Rudin, 1987, Theorem 9.11*) Let a function  $f$  be in  $L^1(\mathbb{R})$  and  $\widehat{f}$  be in  $L^1(\mathbb{R})$ . Then,

$$g(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-izx} \widehat{f}(z) dz.$$

and  $g \in C_0(\mathbb{R})$  and  $g(x) = f(x)$  a.e.

The connection between Fourier transform techniques and option pricing follows from the following theorem.

**Theorem 2.2.8** (Parseval's identity). (*Rudin, 1987, Proof of Theorem 9.13*) Let  $f, g \in L^2(\mathbb{R})$ . Then,

$$\int_{-\infty}^{\infty} f(x)g(x)dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(z)\overline{\widehat{g}(z)}dz, \quad (2.16)$$

where  $\bar{\cdot}$  denotes the complex conjugate.

Parseval's identity as in (2.16) can be very useful in determining the option price. If the random variable  $S_T$  has a density function  $f$ , then it holds

$$E[e^{-r(T-t)}g(S_T)|\mathcal{F}_t] = e^{-r(T-t)} \int_{-\infty}^{\infty} g(x)f(x)dx.$$

Basically, we are on the left-hand-side of Parseval's identity. For some stochastic processes, the density function is not known explicitly, like e.g. in the Merton model. However, the characteristic function, the Fourier transform of the probability density function, is known. Heston (1993) describes for his stochastic volatility model the characteristic function and applies Fourier techniques to determine the option price. In a nutshell, Parseval's identity is the link between the option price and Fourier techniques.

**Remark 2.2.9.** Often the payoff function  $g$  in (2.3) is not in  $L^1(\mathbb{R})$ . Then, the Fourier transform does not exist. Here, the idea is the introduction of a dampening factor. Let  $\eta \in \mathbb{R}$  such that  $e^{\eta x}g(x) \in L^1(\mathbb{R})$ . Then, the Fourier transform of  $e^{\eta x}g(x)$  exists. In order to not change the value of the integral on the left-hand-side of Parseval's identity, in this case the function  $f(x)$  is weighted with the function  $e^{-\eta x}$ . In our application,  $f$  is the density function and decaying very rapidly at the limits and, thus, it often holds  $f(x)e^{-\eta x} \in L^1(\mathbb{R})$ . Denoting with  $\widehat{g}_\eta$  the Fourier transform of  $e^{\eta x}g(x)$  and with  $\widehat{f}_{-\eta}$  the Fourier transform of  $e^{-\eta x}f(x)$ , we get

$$\int_{-\infty}^{\infty} [g(x)e^{\eta x}][e^{-\eta x}f(x)]dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{g}_\eta(z)\overline{\widehat{f}_{-\eta}(z)}dz.$$

The dampening factor allows us to use Parseval's identity to switch into the Fourier world, even if the payoff functions are not in  $L^1(\mathbb{R})$ .

### 2.2.3 Monte-Carlo simulation

The idea of the Monte-Carlo simulation is to solve the integral or expectation in (2.3) by repeatedly simulating the underlying SDE in (2.1) independently, determining for each simulation the discounted payoff and, finally, taking the mean,

$$e^{-r(T)} E[g(S_T)] \approx e^{-r(T)} \frac{1}{M} \sum_{k=1}^M g(S_T^k).$$

Following Glasserman (2003), the estimator above is, for  $M \geq 1$ , unbiased in the sense that its expectation is the target quantity and for  $M \rightarrow \infty$ , the estimator is consistent and converging to the true option price. In applications with a finite  $M < \infty$ , the Monte-Carlo simulation makes an approximation error. If we assume that, for the random variable  $S_T$ ,  $E[|g(S_T)|] < \infty$  and  $Var[g(S_T)] = \sigma^2 < \infty$ , then it can easily be shown that  $E[E[g(S_T)] - \frac{1}{M} \sum_{k=1}^M g(S_T^k)] = \frac{\sigma}{\sqrt{M}}$  and that the approximation error is, due to the central limit theorem, asymptotically normal distributed. This yields

$$\lim_{M \rightarrow \infty} \mathcal{P} \left( \frac{\sigma a}{\sqrt{M}} \leq E[g(S_T)] - \frac{1}{M} \sum_{k=1}^M g(S_T^k) \leq \frac{\sigma b}{\sqrt{M}} \right) = \Phi(b) - \Phi(a),$$

where  $\Phi$  is the cumulative distribution function of a standard normal distribution.

Regarding Monte-Carlo simulations, for a given number  $M$  of sample paths it can be beneficial to apply variance reduction techniques to reduce the variance of the Monte-Carlo estimator. Here, we present the idea of antithetic variates. This method uses pairs of samples that are negatively correlated with each other. The motivation is given by the general relation  $Var(X + Y) \leq Var(X) + Var(Y) + 2Cov(X, Y)$ . In our applications, by simulating a Brownian motion, random variables  $Z \sim N(0, \tilde{\sigma})$ , where the volatility  $\tilde{\sigma}$  depends on the explicit application. To apply antithetic variates, we use additionally the random variable  $-Z$  in an additional sample. By denoting with  $S_T^{k+}$  and  $S_T^{k-}$  the two samples, our Monte-Carlo estimator,

$$\frac{2}{M} \sum_{k=1}^{\frac{M}{2}} g(S_T^{k+}) + \frac{2}{M} \sum_{k=1}^{\frac{M}{2}} g(S_T^{k-}),$$

applies the idea of antithetic variates and if  $S_T^{k+}$  and  $S_T^{k-}$ , the variance is reduced. For more details, we refer to Glasserman (2003) and Seydel (2012).

**Remark 2.2.10.** *The name Monte-Carlo traces back to the origins of the Monte-Carlo technique in the 1940s. John von Neumann, contacted by Stanislaw Ulam, came up with the code name Monte-Carlo for a secret project at the Los Alamos National Laboratory, see Anderson (1986) and Andrieu et al. (2003).*

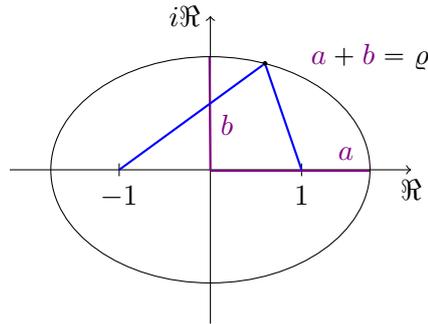


Figure 2.1: Illustration of a Bernstein ellipse with foci at  $\pm 1$ . The sum of the connection of each point on the ellipse with the two foci is exactly  $\varrho$ . We see that semimajor  $a$  and semiminor  $b$  of the ellipse are summing up to the radius of the ellipse  $\varrho$ .

## 2.3 Miscellaneous

In this section, some further concepts are introduced to which we will refer later in the thesis. In the theory later, we require functions defined on  $[-1, 1]$  to be analytically extendable to a Bernstein ellipse with foci at  $\pm 1$  and radius  $\varrho$ . The convergence results for the Chebyshev interpolation are connected to  $\varrho$ . The definition of a Bernstein ellipse traces back to Bernstein (1912). In Figure 2.1, we illustrate a Bernstein ellipse with foci at  $\pm 1$ . The sum of the connection of each point on the ellipse with the two foci is exactly  $\varrho$ . We see that semimajor  $a$  and semiminor  $b$  of the ellipse are summing up to the radius of the ellipse  $\varrho$ . In (4.36), we show how the  $D$ -variate Bernstein ellipse is defined and which transformation has to be applied for arbitrary foci  $\underline{p}$  and  $\bar{p}$ .

Moreover, in Chapter 5, we combine Chebyshev interpolation with the empirical interpolation and present in the following algorithm the basic concept of empirical interpolation. The idea behind empirical interpolation is to approximate a parameter-dependent function  $g(x, \mu)$  by a sum of functions in which parameter dependent part and  $x$  dependent part are separated, e.g.

$$g(x, \mu) \approx \sum_{m=1}^{\mathcal{M}} g(x_m^*, \mu) \Theta_m(x).$$

The points  $x_m^*$ ,  $m = 1, \dots, \mathcal{M}$  are referred to as so-called magic points, see Barrault et al. (2004) and Maday et al. (2009). Especially when applying an integration, a separability of parameter dependent part and space dependent part is beneficial, see Gaß et al. (2016). In the following, we provide in Algorithm 1 the description of the empirical interpolation algorithm from Barrault et al. (2004) as described in Gaß (2016). This version describes the empirical interpolation algorithm for a function  $g : \Omega \times \mathcal{P} \rightarrow \mathbb{R}$  with  $\Omega \subset \mathbb{R}$  and

## 2 Mathematical Preliminaries

$\mathcal{P} \subset \mathbb{R}$ . Thus, the spacial dimension is  $d = 1$  and the dimensionality of the parameter space is  $D$ . Interestingly, (Gaß, 2016, Algorithm 3) is also described in a discrete way, i.e. it reflects that in a numerical implementation,  $\Omega$  as well as  $\mathcal{P}$  are discretized. The idea of the empirical interpolation is that first the parameter  $p^*$  is identified at which the highest error occurs and, then, the space value  $x^*$  for which, given parameter  $p^*$ , the highest error occurs. This value is then determined as magic point and  $g(x^*)$  is incorporated into the empirical interpolator. This is also referred to as greedy search.

---

**Algorithm 1** (Gaß, 2016, Algorithm 3): Discrete EI algorithm,  $d = 1$

---

- 1: Let  $\Omega_{\text{discr.}}$  be a finite, discrete set in  $\mathbb{R}$ ,  $|\Omega_{\text{discr.}}| = N \in \mathbb{N}$ ,  $\Omega = \{\omega_1, \dots, \omega_N\}$
  - 2: Let  $\mathcal{P}_{\text{discr.}}$  be some finite parameter set in  $\mathbb{R}$ ,  $|\mathcal{P}_{\text{discr.}}| = K \in \mathbb{N}$
  - 3: Let further  $\mathcal{U}_{\text{discr.}}$  be a finite set of parametrized vectors on  $\Omega_{\text{discr.}}$ ,  $|\mathcal{U}_{\text{discr.}}| = K \in \mathbb{N}$ ,  
 $\mathcal{U}_{\text{discr.}} = \{\vec{u}_i = (u(p_i)(\omega_1), \dots, u(p_i)(\omega_N)) \mid p_i \in \mathcal{P}_{\text{discr.}}, i \in \{1, \dots, K\}\} \subset \mathbb{R}^N$
  - 4: **function** DISCRETE INTERPOLATION OPERATOR  $I_M^{\text{DISCR}}(\vec{u})$
  - 5:     **return**  $I_M^{\text{discr}}(\vec{u}) = \sum_{i=1}^M \alpha_i(\vec{u}) \vec{q}_i$
  - 6:     with  $\alpha_i \in \mathbb{R}$ ,  $i \in \{1, \dots, M\}$ , depending on  $\vec{u}$  and given by
  - 7:          $Q\vec{\alpha} = (\vec{u}^{(\iota_1)}, \dots, \vec{u}^{(\iota_M)}), \quad Q \in \mathbb{R}^{M \times M}, \quad Q_{ij} = \vec{q}_j^{(\iota_i)}$
  - 8:     where the set of *magic indices*  $\{\iota_1, \dots, \iota_M\} \subset \{1, \dots, N\}$  and the set of *basis vectors*  $\{\vec{q}_1, \dots, \vec{q}_M\}$  are recursively defined by
  - 9:          $\vec{u}_1 = \arg \max_{\vec{u}_i \in \mathcal{U}_{\text{discr.}}, i=1, \dots, K} \max_{j=1, \dots, N} |\vec{u}_i^{(j)}|$
  - 10:          $\iota_1 = \arg \max_{j=1, \dots, N} |\vec{u}_1^{(j)}|$
  - 11:          $\xi_1 = \omega_{\iota_1}$
  - 12:          $\vec{q}_1 = \frac{1}{\vec{u}_1^{(\iota_1)}} \vec{u}_1$
  - 13:     and for  $M > 1$  with  $\vec{r}_i = \vec{u}_i - I_{M-1}^{\text{discr}}(\vec{u}_i)$ ,  $i \in \{1, \dots, N\}$ , by
  - 14:          $\vec{u}_M = \arg \max_{\vec{u}_i \in \mathcal{U}_{\text{discr.}}, i=1, \dots, K} \max_{j \in \{1, \dots, N\}} |\vec{r}_i^{(j)}|$
  - 15:          $\iota_M = \arg \max_{i=1, \dots, N} |\vec{r}_M^{(i)}|$
  - 16:          $\xi_M = \omega_{\iota_M}$
  - 17:          $\vec{q}_M = \frac{1}{\vec{r}_M^{(\iota_M)}} (\vec{u}_M - I_{M-1}^{\text{discr}}(\vec{u}_i))$
- 

The convergence rate of the empirical interpolation is connected to the Kolmogorov n-width. In the following, we state the definition.

**Definition 2.3.1** (Kolmogorov n-width). *Let  $X$  be Banach space of continuous functions defined over a domain  $\bar{\Omega}$  part of  $\mathbb{R}$ ,  $\mathbb{R}^d$ , or  $\mathbb{C}^d$ . The Kolmogorov n-width of  $U$  in  $X$  is defined by*

$$dn(U, X) = \inf_{X_n} \sup_{x \in U} \inf_{y \in X_n} \|x - y\|_X$$

## 2 Mathematical Preliminaries

where  $X_n$  is some (unknown)  $n$ -dimensional subspace of  $X$ . The  $n$ -width of  $U$  thus measures the extent to which  $U$  may be approximated by some finite dimensional space of dimension  $n$ .

## 2 *Mathematical Preliminaries*

### 3 Numerical Investigation of the de-Americanization Method

Mathematics is the cheapest science.  
Unlike physics or chemistry, it does  
not require any expensive equipment.  
All one needs for mathematics is a  
pencil and paper.

---

George Pólya

*This chapter is based on Burkovska et al. (2016) and presents the parts to which I provided a significant contribution*

In the financial industry, this statement of George Pólya does not hold any longer. Regarding derivatives, complex models and product types require extensive pricing techniques and often the fair price of a derivative has to be numerically approximated. Here, pencil and paper are replaced by computers and in addition to accuracy, run-times are essential as well. In this chapter, we focus on calibration to American options. But why American options?

The most frequently traded single stock options are of American type. In general, there exists a variety of (semi-)closed pricing formulas for European options. However, for American options, there hardly exist any closed pricing formulas, and the pricing under advanced models rely on computationally expensive numerical techniques such as the Monte Carlo simulation or partial (integro) differential methods.

Tackling this core problem, in the financial industry, the so-called de-Americanization approach has become market standard: American option prices are transferred into European prices before the calibration process itself is started. This is usually done by applying a relatively simple binomial tree. By replacing American options with European options, the complexity of the calibration problem is reduced and the computational costs are lowered significantly. The striking advantage of this procedure is that it enables to employ the advanced and standard tools for model calibration to European option data which are readily available and typically efficient. Figure 3.1 illustrates the scheme of the de-Americanization methodology.

The de-Americanization methodology enjoys three attractive features,

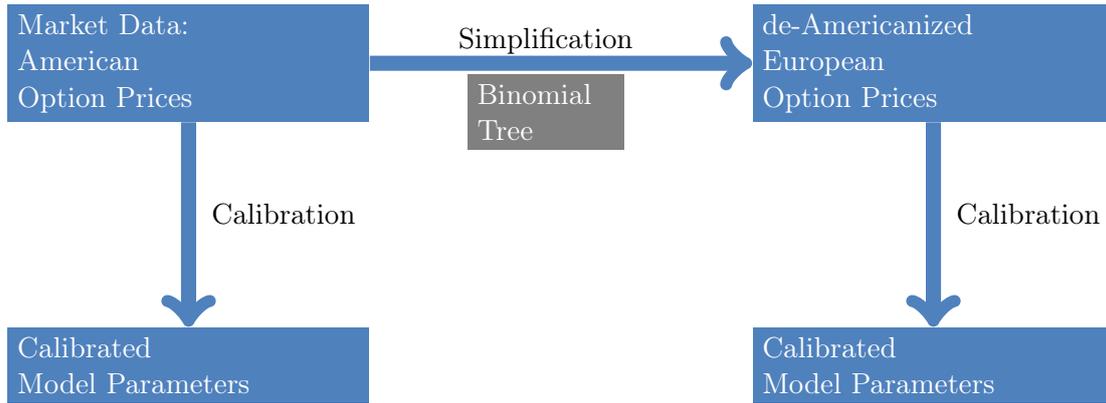


Figure 3.1: De-Americanization scheme: American option prices are transferred into European prices before the calibration process itself is started. We investigate the effects of de-Americanization by comparing the results to directly calibrating American options.

- it delivers fast run-times,
- it is easy to implement,
- it can flexibly be integrated into the pricing and calibration toolbox at hand.

One downside is that no theoretical error control is available. Therefore, it is important to empirically investigate the accuracy, the performance and the resulting methodological risk of the method.

The method is briefly mentioned by Carr and Wu (2010), who describe how their implied volatility data, stemming from the provider OptionMetrics, is obtained by applying exactly this de-Americanization scheme. To the best of our knowledge, the de-Americanization methodology has not been investigated deeply in the literature. We therefore devote the current paper to this task. In order to conduct a thorough investigation, we consider prominent models and identify relevant scenarios in which to perform extensive numerical tests. We focus on options on non-dividend-paying underlyings and explore the CEV model as an example of a local volatility model, the Heston model as a stochastic volatility model and the Merton model as a jump diffusion model. For all of these models, we implemented finite element solvers as benchmark method for pricing American options.

The following questions serve as guidelines to specify decisive parameter settings within our studies.

1. Since American and European puts on non-dividend-paying underlyings coincide for zero interest rates, we analyze in particular the methodology for different interest rates.

### 3 Numerical Investigation of the de-Americanization Method

2. Intuitively, with higher maturities, the early exercise feature of American options becomes more valuable and American and European option prices differ more significantly. Therefore, we investigate the following question: Does the accuracy of the de-Americanization methodology depend on the maturity and do de-Americanization errors increase with increasing maturities?
3. In-the-money and out-of-the-money options play different roles. First, out-of-the-money options are preferred by practitioners for calibration since they are more liquidly traded, see for instance Carr and Wu (2010). Second, in-the-money options are more likely to be exercised. How does the de-Americanization methodology perform for out-of-the-money options and for in-the-money options?

Our investigation is organized as follows. First, we introduce the de-Americanization methodology in Section 3.1. Then, we briefly describe in Section 3.2 the models and the benchmark pricing methodology. Section 3.3 presents the numerical results: The accuracy of the calibration procedure obviously hinges on the accuracy of the underlying pricing routine. We therefore first specify the de-Americanization pricing routine and investigate its accuracy. Afterwards, we present the results of calibration to both synthetic data and market data. To conclude the numerical study, we present the effects of different calibration results on the pricing of exotic options. We summarize our findings in Section 3.4.

#### Short literature overview on American options

For an overview of pricing American options, we refer to Barone-Adesi (2005). The problem of pricing an American put traces back to Samuelson (1965) and McKean (1965). Brennan and Schwartz (1977) were one of the first who provided numerical solutions and also the binomial tree model of Cox et al. (1979) was used to price American options. Broadie and Detemple (1996) approximate the American put price by interpolating between an upper and lower bound of the price. Longstaff and Schwartz (2001) combined American option pricing with Monte-Carlo techniques based on a polynomial interpolation of the continuation value. The American option price problem can also be interpreted as a free boundary problem, see e.g. Kim (1990) or as an optimal stopping problem, see e.g. Peskir and Shiryaev (2006), and be formulated as a dynamic programming principle. Although Barone-Adesi (2005) concludes that the mainstream computational problems have been solved satisfactorily, by switching the focus on calibration, there are rather recent developments for calibrating American options. As examples, we state Haring and Hochreiter (2015), who apply a specific search algorithm, namely a Cuckoo search algorithm, in the calibration process, and Ballestra and Cecere (2016). They provide a method to forecast the parameters of the constant elasticity of variance (CEV) model implied by American options in order to fit the model relatively quickly to market data. To summarize, calibrating American market data is a numerically challenging problem.

The research in the literature puts the focus now on optimizing the calibration procedure to reduce the run-time. At its core, still path-dependent, rather complex, American options are priced.

### 3.1 De-Americanization Methodology

In this section, we give a precise and detailed description of the methodology. The de-Americanization methodology is used to fit models to market data. The core idea of de-Americanization is to transfer the available American option data into pseudo-European option prices prior to calibration. This significantly reduces the computational time as well as the complexity of the required pricing technique. Basically, de-Americanization can be split into three parts. The first part consists in collecting the available market data. The currently observable price of the underlying  $S_0$ , interest rate  $r$  and the available American option prices are collected. In the following, we will denote the American option price of the  $i$ -th observed option by  $V_A^i$ . We interpret the market data as the true option prices, thus we assume that the observed market prices  $V_A^i$  can be interpreted as supremum over all stopping times  $t \in [0, T_i]$ :  $V_A^i = \sup_{t \in [0, T_i]} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t) | \mathcal{F}_0]$ ,  $i = 1, \dots, N$ , where  $t$  is a stopping time,  $\tilde{\mathcal{H}}_i$  is the  $i$ -th payoff function,  $T_i$  the maturity of the  $i$ -th option, and the expectations are taken under a risk-neutral measure,  $\mathcal{F}$  is the natural filtration, and  $N$  denotes the total number of options. Up to this point, no approximation has been used.

The second step is the application of the binomial tree to create pseudo-European – so-called de-Americanized – prices based on the observed American market data. In this step, we look at each American option individually and find the price of the corresponding European option with the same strike and maturity. This European option is found by fitting a binomial tree to the American option. The binomial tree was introduced by Cox et al. (1979) as follows. Starting at  $S_0$ , at each time step and at each node, the underlying can either go up by a factor of  $u$  or down by a factor of  $\frac{1}{u}$  and the risk-neutral probability of an upward movement is given by

$$p = \frac{e^{r\Delta t} - \frac{1}{u}}{u - \frac{1}{u}}. \quad (3.1)$$

Once the tree is set up, options can be valued by going backwards from each final node. Thus, path-dependent options can be evaluated easily. Since for each option  $i$  the American option price  $V_A^i$  is known, as well as  $S_0$  and  $r$ , the only unknown parameter of the tree is the upward factor  $u$ . At this step, the upward factor  $u_i^*$  is determined such that the price of the American option in the binomial tree matches the observed market price. Thus, denoting  $\{0 : \Delta t : T_i\} = \{0, \Delta t, 2\Delta t, \dots, T_i\}$ , we have  $\sup_{t \in \{0 : \Delta t : T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{u_i^*}) | \mathcal{F}_0] = V_A^i$ , where  $t$  is a stopping time,  $S_t^{u_i^*}$  denotes the underlying process described by a binomial tree with upward factor  $u_i^*$ . The early exercise

### 3 Numerical Investigation of the de-Americanization Method

feature of American options is reflected in the fact that the the supremum is taken over all discrete time steps. A detailed description of pricing American options in a binomial tree model is given in Van der Hoek and Elliott (2006). Once  $S_t^{u_i^*}$  is determined, the corresponding European option with the same strike and maturity as the American option is specified,  $V_E^i = E[e^{-rT_i} \tilde{\mathcal{H}}_i(S_{T_i}^{u_i^*}) | \mathcal{F}_0]$ . Note that fixing  $u_i^*$  also implicitly determines the implied volatility.

Then, for each American option  $V_A^i$ , a corresponding European option  $V_E^i$  has been found, and the actual model calibration can start. The goal is to fit a model  $M$ , depending on parameters  $\mu \in \mathbb{R}^d$ , where  $d$  denotes the number of parameters in the model, to the European option prices  $V_E^i$ ,  $i = 1, \dots, N$ . Denote by  $S_{T_i}^{M(\mu)}$  the underlying process in model  $M$  with parameters  $\mu \in \mathbb{R}^d$ . In the calibration, the parameter vector  $\mu$  is determined by minimizing the objective function of the calibration. Algorithm 2 summarizes the de-Americanization methodology in detail.

---

#### Algorithm 2 De-Americanization methodology

---

- 1: **procedure** COLLECTION OF OBSERVABLE DATA
  - 2:      $S_0, r,$
  - 3:      $V_A^i = \sup_{t \in [0, T_i]} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t) | \mathcal{F}_0], \quad i = 1, \dots, N$
  - 4: **procedure** APPLICATION OF THE BINOMIAL TREE TO EACH OPTION INDIVIDUALLY
  - 5:     **for**  $i = 1 : N$
  - 6:         Find  $u_i^*$  such that
  - 7:              $\sup_{t \in \{0: \Delta t: T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{u_i^*}) | \mathcal{F}_0] = V_A^i$  where the supremum is taken over all stopping times  $t$
  - 8:             Derive the corresponding European option price with  $u_i^*$
  - 9:              $V_E^i = E[e^{-rT_i} \tilde{\mathcal{H}}_i(S_{T_i}^{u_i^*}) | \mathcal{F}_0]$
  - 10:     **end**
  - 11: **procedure** CALIBRATION TO EUROPEAN OPTIONS
  - 12:     Find  $\mu$  such that the differences
  - 13:          $E[e^{-rT_i} \tilde{\mathcal{H}}_i(S_{T_i}^{M(\mu)})] - V_E^i, \quad i = 1, \dots, N$
  - 14:     are minimized according to the objective function
- 

Regarding the uniqueness of the factor  $u_i^*$  in the De-Americanization methodology described in Algorithm 2, we will first investigate the case of a European put option. Therefore, we interpret the risk-neutral probability in (3.1) as function of  $u$ ,  $p(u) = \frac{ue^{r\Delta t} - 1}{u^2 - 1}$ . At each node in the binomial tree we have a two-point distribution, that we call Bernoulli distribution  $X \sim QB(u)$ , where the value  $u$  is taken with probability  $p(u)$  and the value  $\frac{1}{u}$  is taken with probability  $(1 - p(u))$ .

**Proposition 3.1.1.** For  $i = 1, \dots, n$ , let  $X_i \sim QB(u)$  and  $Y_i \sim QB(u')$ . If  $u \leq u'$ , and  $u, u' > e^{r\Delta t}$  then for any  $K \in \mathbb{R}$

$$\mathbb{E} \left[ \left( K - \prod_{i=1}^n X_i \right)^+ \right] \leq \mathbb{E} \left[ \left( K - \prod_{i=1}^n Y_i \right)^+ \right].$$

*Proof.* We can reduce this case to the one-dimensional case in the following way. Let  $j \in \{1, \dots, n\}$  and denote by  $P^j(\cdot)(\cdot)$  the conditional probability given  $\sigma(X_i, i \neq j)$ . Then, by definition of the conditional probability

$$E \left[ \left( K - \prod_{i=1}^n X_i \right)^+ \right] = \iint \left( K - X_j(\omega') \prod_{i \neq j} X_i(\omega) \right)^+ P^j(d\omega')(\omega) P(d\omega).$$

For all  $j \in \{1, \dots, n\}$  and  $\omega \in \Omega$  the function  $x \mapsto (K - x \prod_{i \neq j} X_i(\omega))^+$  is convex. By the independence of  $X_j$  and  $\{X_i, i \neq j\}$  a.s.  $P^j(X_j \in \cdot) = P(X_j \in \cdot)$ . This allows us to use the one-dimensional result from Lemma 3.1.2 and from  $X_j \leq_{cx} Y_j$  it follows

$$\int \left( K - X_j(\omega') \prod_{i \neq j} X_i \right)^+ P^j(d\omega') \leq \int \left( K - Y_j(\omega') \prod_{i \neq j} X_i \right)^+ P^j(d\omega') \text{ a.s.}$$

This yields

$$E \left[ \left( K - \prod_{i=1}^n X_i \right)^+ \right] \leq E \left[ \left( K - Y_j \prod_{i \neq j} X_i \right)^+ \right].$$

As a next step, conditioning on  $(\sigma(Y_j, X_i, i \neq j, j_2))$  the same technique is applied to show

$$E \left[ \left( K - X_{j_2} \prod_{i \neq j, j_2} X_i Y_j \right)^+ \right] \leq E \left[ \left( K - Y_{j_2} \prod_{i \neq j, j_2} X_i Y_j \right)^+ \right]$$

and successively, the assertion of the proposition follows.  $\square$

Here, we present an additional lemma which will be used in the proof of Proposition 3.1.1.

**Lemma 3.1.2.** Focusing on one node in the binomial tree, let  $X \sim QB(u)$  and  $Y \sim QB(u')$  with  $u' \geq u$ . Let  $u, u' > e^{r\Delta t}$  be satisfied. Then the random variable  $X$  is smaller than the random variable  $Y$  with respect to the convex order, i.e.  $X \leq_{cx} Y$ .

*Proof.* Following (Müller and Stoyan, 2002, Theorem 1.5.3 and Theorem 1.5.7) it suffices to show

1.  $E[X] = E[Y]$

### 3 Numerical Investigation of the de-Americanization Method

$$2. E[(X - k)^+] \leq E[(Y - k)^+]$$

Since  $p$  as in (3.1) is set up as risk-neutral probability, it holds for any  $u$  that  $E[X] = e^{r\Delta t}$  and thus, the first condition is satisfied. Given a random variable  $X$  with a factor  $u$  and a random variable  $Y$  with factor  $u' > u$ , we distinguish regarding the second condition 5 cases.

$$\text{Case 1: } \frac{1}{u'} \leq \frac{1}{u} \leq u \leq u' \leq k$$

Obviously, in any case both options are out-of-the-money and  $E[(X - k)^+] = 0 = E[(Y - k)^+]$ .

$$\text{Case 2: } \frac{1}{u'} \leq \frac{1}{u} \leq u \leq k \leq u'.$$

Here,  $E[(X - k)^+] = 0$  and hence, the second condition is satisfied.

$$\text{Case 3: } \frac{1}{u'} \leq \frac{1}{u} \leq k \leq u \leq u'.$$

In this case, we have

$$\begin{aligned} E[(Y - k)^+] - E[(X - k)^+] &= p(u')(u' - k) - p(u)(u - k) \\ &= (u')u' - p(u)u - k(p(u') - p(u)). \end{aligned}$$

The function  $p(u) = \frac{ue^{r\Delta t} - 1}{u^2 - 1}$  is a monotonically decreasing function because the derivative  $p'(u) = \frac{-e^{r\Delta t} - u^2 e^{r\Delta t} + 2u}{(u^2 - 1)^2}$  would have the roots  $u_{a,b} = \frac{1 \pm \sqrt{1 - e^{2r\Delta t}}}{e^{r\Delta t}}$ , but due to  $e^{r\Delta t} \geq 1$  either the derivative has no roots or a root at  $u = 1$  in the case  $r = 0$ . Thus, by assuming  $u', u > e^{r\Delta t}$ ,  $E[(Y - k)^+] - E[(X - k)^+]$  is monotone in  $k$  and in Case 4 for  $k = \frac{1}{u}$  we show that  $E[(Y - k)^+] - E[(X - k)^+] = 0$  holds.

$$\text{Case 4: } \frac{1}{u'} \leq k < \frac{1}{u} \leq u \leq u'.$$

Due to  $k \leq \frac{1}{u}$  it follows  $E[(X - k)^+] = E[X - k]$ . Thus,  $E[(Y - k)^+] - E[(X - k)^+] = E[(Y - k)^+] - E[X - k]$ . Obviously, it holds

$$E[(Y - k)^+] \geq E[Y - k]$$

and this leads to  $E[(Y - k)^+] - E[X - k] \geq E[Y - k] - E[X - k] = 0$ .

$$\text{Case 5: } k \leq \frac{1}{u'} \leq \frac{1}{u} \leq u \leq u'.$$

It holds  $E[(Y - k)^+] = E[Y - k]$  and  $E[(X - k)^+] = E[X - k]$ . Thus,  $E[(Y - k)^+] - E[(X - k)^+] = E[Y - k] - E[X - k] = 0$  follows.  $\square$

**Remark 3.1.3.** In the implementation of the tree, we set the time step size  $\Delta t \approx 0.0002$  and we use a simple bi-section approach as suggested by Van der Hoek and Elliott (2006)

### 3 Numerical Investigation of the de-Americanization Method

to find  $u^*$ . Thus, given a market price  $V_A$ , starting with an upper bound  $u_{ub}$  and a lower bound  $u_{lb}$  satisfying the conditions in Proposition 3.1.1 such that,

$$\begin{aligned} \sup_{t \in \{0:\Delta t:T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{u_{ub}}) | \mathcal{F}_0] &> V_A, \\ \sup_{t \in \{0:\Delta t:T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{u_{lb}}) | \mathcal{F}_0] &< V_A, \end{aligned}$$

the bi-section approach is started and the new candidate for  $u^*$  is  $\hat{u} = \frac{u_{ub} + u_{lb}}{2}$ . When  $\sup_{t \in \{0:\Delta t:T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{\hat{u}}) | \mathcal{F}_0] > V_A$ , we set  $u_{ub} = \hat{u}$  for the next iteration, otherwise  $u_{lb} = \hat{u}$ . As stopping criterion, we choose

$$\left| \sup_{t \in \{0:\Delta t:T_i\}} E[e^{-rt} \tilde{\mathcal{H}}_i(S_t^{\hat{u}}) | \mathcal{F}_0] - V_A \right| \leq \varepsilon,$$

and set  $\varepsilon = 10^{-5}$  in our implementation. In Proposition 3.1.1 we have investigated the European put case and can deduce from the convex ordering that the put prices are monotonically increasing in  $u$ . For a strict order, the  $u^*$ -value is thus uniquely determined. In our case, the  $u^*$ -value can be determined uniquely as minimum of all  $u$  values satisfying the stopping criterion. Moreover, this indicates that also the American put price in the binomial tree is increasing with increasing  $u$ . We validated this by numerical tests (not reported). This is in line with the recommendation in Van der Hoek and Elliott (2006). The only observed limitation is that the American put price can not be given by an immediate exercise at the initial time. This is explained in detail in Remark 3.3.1.

## 3.2 Pricing Methodology

In this section, we present model formulation and numerical implementation of the CEV model. To investigate the de-Americanization methodology, we need to price the American and European options. Our market data in the numerical study later on will be based on options on the Google stock (Ticker: GOOG). As Google does not pay dividends, we neglect dividend payments in our pricing methodology. Without dividend payments, for  $r > 0$ , it holds in general that American calls coincide with European calls and only American puts have to be treated differently. The opposite is true for  $r < 0$ , in which case American and European puts coincide and American and European calls have to be treated differently.

In general, for European options, there exists a variety of fast pricing methodologies such as Fast Fourier Transform (Carr and Madan (1999); Raible (2000)) or even closed-form solutions. The common approaches for pricing American options are P(I)DE methods using either the finite difference method (FDM) or a finite element method (FEM). We choose FEM since it is typically more flexible. To solve the resulting variational

inequalities for American options, we use the Projected SOR Algorithm, Achdou and Pironneau (2005), Seydel (2012).

### 3.2.1 Pricing PDE

Denote by  $t = T - \tau$  the time to maturity  $T$ ,  $T < \infty$  and by  $K$  the strike of an option. For the CEV model, we stay with the  $S$  variable,  $S \in (0, \infty)$ . In the following, we will denote an American or European call or put price by  $P_{call/put}^{Am/Eu}$ . For the CEV model we have  $P_{call/put}^{Am/Eu} : (0, T) \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ . The value of an option at  $t = 0$  is given by the payoff function  $\tilde{\mathcal{H}}_{call/put}(\cdot)$ ,  $P_{call/put}(0) = P_0 = \tilde{\mathcal{H}}_{call/put}$  with  $\tilde{\mathcal{H}}_{call}(S) := (S - K)^+$  or  $\tilde{\mathcal{H}}_{put}(S) := (K - S)^+$ .

Then, to find the value of the European option  $P_{call/put}^{Eu}$ , paying  $P_0^{call/put} = \tilde{\mathcal{H}}_{call/put}(S)$  ( $P_0^{call/put} = \tilde{\mathcal{H}}_{call/put}(x)$ ) at  $t = 0$  leads to solve the following initial boundary value problem

$$\frac{\partial P_{call/put}^{Eu}}{\partial t} - \mathcal{L}^{CEV} P_{call/put}^{Eu} = 0, \quad P_{call/put}^{Eu}(0) = P_0^{call/put}, \quad (3.2)$$

where the spatial partial (integro) differential operator  $\mathcal{L}^{CEV}$  is given by

$$\mathcal{L}^{CEV} P_{call/put}^{Am/Eu} := \frac{\sigma S_t^{\zeta-1}}{2} S^2 \frac{\partial^2 P_{call/put}^{Am/Eu}}{\partial S^2} + rS \frac{\partial P_{call/put}^{Am/Eu}}{\partial S} - rP_{call/put}^{Am/Eu}. \quad (3.3a)$$

Due to its early exercise possibility, pricing an American option (e.g., put) results in additional inequality constraints, and leads us to solve the following system of inequalities

$$\frac{\partial P_{call/put}^{Am}}{\partial t} - \mathcal{L}^{CEV} P_{call/put}^{Am} \geq 0, \quad P_{call/put}^{Am} - P_0^{call/put} \geq 0, \quad (3.4a)$$

$$\left( \frac{\partial P_{call/put}^{Am}}{\partial t} - \mathcal{L}^{CEV} P_{call/put}^{Am} \right) \cdot \left( P_{call/put}^{Am} - P_0^{call/put} \right) = 0. \quad (3.4b)$$

We denote the parameter vector by  $\mu := (\sigma, \zeta) \in \mathbb{R}^2$  for the CEV model. Then the problems (3.2), (3.4) are parametrized problems with  $\mu \in \mathcal{P}$ , where  $\mathcal{P} \subset \mathbb{R}^d$  is a parameter space. The solution can be written as  $P = P(\mu)$ . In some cases, for notational convenience, we will omit the parameter-dependence of  $P$  and related quantities.

### 3.2.2 Variational Formulation

As a next step, we pose the problem in the weak form and introduce the variational formulation. We localize the problems to an open bounded domain  $\Omega \subset \mathbb{R}$ , defined as  $\Omega := (S_{\min}, S_{\max}) \subset \mathbb{R}^1$ . The functional space is introduced in (3.5).

$$V := \{v \in L^2(\Omega) : S \frac{\partial v}{\partial S} \in L^2(\Omega)\}. \quad (3.5)$$

Define  $V'$  to be the dual space of  $V$  and denote by  $\langle \cdot, \cdot \rangle$  a duality pairing between  $V$  and  $V'$  and by  $(\cdot, \cdot)$  an inner product on  $L^2(\Omega)$ . Denote a temporal interval  $I := [0, T]$ ,  $T > 0$ .

Following Achdou and Pironneau (2005), we can define  $a^{CEV} : V \times V \rightarrow \mathbb{R}$ ,

$$a^{CEV}(\psi, \phi; \mu) := \int_{\Omega} \frac{\sigma S^{2\zeta}}{2} \frac{\partial \psi}{\partial S} \frac{\partial \phi}{\partial S} dS + r \int_{\Omega} \psi \phi dS + \int_{\Omega} (-rS + \zeta \sigma^2 S^{2\zeta-1}) \frac{\partial \psi}{\partial S} \phi dS. \quad (3.6a)$$

Then a weak form of (3.2) reads as follows, Achdou and Pironneau (2005): Find  $u(\mu) \in L^2(I; V) \cap C^1(I; L^2(\Omega))$  with  $\frac{\partial u(\mu)}{\partial t} \in L^2(I; V')$ ,  $f(\mu) \in L^2(I; V')$ ,  $u_0(\mu) \in L^2(\Omega)$ , such that for a.e.  $t \in I$  holds

$$\left\langle \frac{\partial u}{\partial t}(t; \mu), \phi \right\rangle + a(u(t; \mu), \phi; \mu) = \langle f(t; \mu), \phi \rangle, \quad \phi \in V, \quad (3.7a)$$

$$u(0; \mu) = u_0(\mu), \quad (3.7b)$$

where

$$\langle f(t; \mu), \phi \rangle = -a(u_L(t; \mu), \phi) - \left\langle \frac{\partial u_L}{\partial t}(t; \mu), \phi \right\rangle. \quad (3.8)$$

and  $u(0; \mu) = P(0; \mu) - u_L(0; \mu)$ , the modified payoff is  $\mathcal{H}(t; \mu) := \tilde{\mathcal{H}} - u_L(t; \mu)$ . Here  $u_L(\mu)$  is a Dirichlet lift function,  $u_L(\mu) \in L^2(I; V) \cap C^1(I; L^2(\Omega))$  with  $\frac{\partial u_L}{\partial t}(\mu) \in L^2(I; V')$ , such that  $u_L(t; \mu) = g(t; \mu)$  on  $\partial\Omega_D$ , where the function of the boundary values  $g$  are defined below. Then the price of an option in (3.2) is given as  $P = u(\mu) + u_L(\mu)$ .

With the use of the above notations, we present the variational formulation for American options. Introduce a closed convex subset  $\mathcal{K}(t; \mu) \subset V$ ,

$$\mathcal{K}(t; \mu) := \{\phi \in V : \phi \geq \mathcal{H}(t; \mu), \quad \text{in } \Omega\}, \quad (3.9)$$

where the inequality is understood in a pointwise sense. Then the weak form of (3.4), Achdou and Pironneau (2005), reads as follows: Given  $u_0(\mu) \in \mathcal{K}(\mu)$ , find  $u(\mu) \in L^2(I; V) \cap C^1(I; L^2(\Omega))$  with  $\frac{\partial u}{\partial t}(\mu) \in L^2(I; V')$ , such that  $u(t; \mu) \in \mathcal{K}(t; \mu)$  for a.e.  $t \in I$

### 3 Numerical Investigation of the de-Americanization Method

and the following holds for all  $\phi \in \mathcal{K}(t; \mu)$

$$\left\langle \frac{\partial u}{\partial t}(t; \mu), \phi - u(t; \mu) \right\rangle + a(u(t; \mu), \phi - u(t; \mu); \mu) \geq \langle f(t; \mu), \phi - u(t; \mu) \rangle, \quad (3.10a)$$

$$u(0; \mu) = u_0(\mu), \quad (3.10b)$$

where  $u(0; \mu) = P(0; \mu) - u_L(0; \mu)$  and the price of an American put option (3.4) is determined as  $P(\mu) = u(\mu) + u_L(\mu)$ .

#### Boundary Conditions

We tackle the non-homogeneous truncated Dirichlet boundary conditions by means of the lift function  $u_L(t) = g(t)$  onto the domain. For the CEV model, following Seydel (2012), we applied the boundary conditions

$$\begin{aligned} P_{call}^{Am/Eu}(t, S_{min}) &= 0, & P_{call}^{Am/Eu}(t, S_{max}) &= S_{max} - e^{-rt}K, & \text{for call options,} \\ P_{put}^{Eu}(t, S_{min}) &= e^{-rt}K - S_{min}, & P_{put}^{Eu}(t, S_{max}) &= 0, & \text{for European put options,} \\ P_{put}^{Am}(t, S_{min}) &= K - S_{min}, & P_{put}^{Am}(t, S_{max}) &= 0, & \text{for American put options.} \end{aligned}$$

#### Discrete Approximation in Time and Space

For the discretization of the problem in time and space, we follow Achdou and Pironneau (2005). We introduce an equidistantly-spaced partition of the time interval  $[0, T]$  into  $N$  intervals  $[t_0, t_1], \dots, [t_{N-1}, t_N]$  with  $\Delta t = t_1 - t_0$ . The spacial domain is also split into subintervals  $\omega_i = [S_{i-1}, S_i]$ ,  $1 \leq i \leq N_h + 1$  such that  $S_{min} = S_0 < S_1 < \dots < S_{N_h} < S_{N_h+1} = S_{max}$ . We refer to the size of the interval  $\omega_i$  as  $h_i$ , set  $h = \max_{i=1, \dots, N_h+1} h_i$ , and the grid is chosen in such a way that the strike is a node. Then, we define the discrete space  $V_h$  by

$$V_h = \{v \in V, \forall \omega \in \{\omega_1, \dots, \omega_{N_h+1}\}, v|_{\omega} \in \mathcal{P}_1(\omega)\},$$

where the space of linear functions on  $\omega$  is denoted by  $\mathcal{P}_1(\omega)$ . Then, we can also introduce a discrete version of the closed set  $\mathcal{K}$ ,

$$\mathcal{K}_h(t; \mu) := \{\phi \in V_h : \phi \geq \mathcal{H}(t; \mu), \text{ in } \Omega\}. \quad (3.11)$$

Thus, we formulate (3.10) in semi-discrete form applying an implicit Euler scheme for the time stepping. Find  $u(t_{k+1}; \mu) \in \mathcal{K}_h(t_{k+1}; \mu)$ ,  $t_k \in [0, T]$ , such that for all  $\phi \in \mathcal{K}_h(t_{k+1}; \mu)$

### 3 Numerical Investigation of the de-Americanization Method

holds,

$$\frac{1}{\Delta t} \langle u(t_{k+1}; \mu) - u(t_k; \mu), \phi - u(t_{k+1}; \mu) \rangle + a(u(t_{k+1}; \mu), \phi - u(t_{k+1}; \mu); \mu) \geq \langle f(t_{k+1}; \mu), \phi - u(t_{k+1}; \mu) \rangle, \quad (3.12a)$$

$$u(0; \mu) = u_0(\mu), \quad (3.12b)$$

At this point, we introduce a nodal basis of  $V_h$ . We choose the so-called hat functions  $\varphi_i$ ,  $i = 1, \dots, N_h + 1$ ,

$$\varphi_i(S) = \begin{cases} \frac{S - S_{i-1}}{S_i - S_{i-1}}, & S_{i-1} \leq S < S_i, \\ \frac{S_i - S}{S_i - S_{i-1}}, & S_i \leq S < S_{i+1}, \\ 0, & \text{elsewhere.} \end{cases} \quad (3.13)$$

Figure 3.2 visualizes the concept of hat functions and in the following, we

- express  $u(t_{k+1}; \mu) = \sum_{i=1}^{N_h+1} c_i(t_{k+1})\varphi_i$ ,  $u(t_k; \mu) = \sum_{i=1}^{N_h+1} c_i(t_k)\varphi_i$  and  $f(t_{k+1}; \mu) = \sum_{i=1}^{N_h+1} f_i(t_{k+1})\varphi_i$  with the nodal basis,
- apply the Galerkin method, i.e. we now that (3.12) has to be satisfied for all  $\phi \in \mathcal{K}_h(t_{k+1}; \mu)$  and, hence, especially for each basis function  $\varphi_i$ ,  $i = 1, \dots, N_h + 1$ ,
- introduce the mass matrix  $M = (m_{ij})_{1 \leq i, j \leq N_h+1}$  with  $m_{ij} = \langle \varphi_i, \varphi_j \rangle$ , and the stiffness matrix  $A = (a_{ij})_{1 \leq i, j \leq N_h+1}$  with  $a_{ij} = a(\varphi_j, \varphi_i)$ .

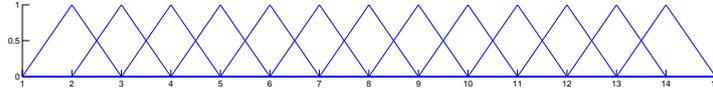


Figure 3.2: Illustration of hat functions  $\varphi_i$ ,  $i = 2, \dots, 14$ , over a node grid with 15 nodes.

These steps allow use to rewrite the inequality from (3.12) in matrix,

$$\frac{1}{\Delta t} M(c(t_{k+1}) - c(t_k)) + A c(t_{k+1}) \geq M \tilde{f}(t_{k+1}).$$

Note that the coefficient vector  $\tilde{f}(t_{k+1})$  is slightly adjusted. To solve the inequality in each time step, we apply the projected successive over-relaxation algorithm as presented in Achdou and Pironneau (2005) and Seydel (2012).

### 3.3 Numerical Study of the effects of de-Americanization

Our main objective is to investigate the de-Americanization methodology with respect to the previously stated questions 1-3 on page 31. But before we look at these questions and the calibration results in detail, we describe the discretization of our FEM pricers followed by an investigation of the effects of de-Americanization on pricing. Then we switch to calibrating to synthetic data and, finally, to market data.

#### 3.3.1 Discretization

We set up mesh sizes and time discretization in all three models such that the errors compared to benchmark solutions are roughly the same. In our test setting, we set  $S_0 = 1$ ,  $r = 0.07$ ,  $T = \{0.5, 0.875, 1.25, 1.625, 2\}$  and  $K$  to 21 equally distributed values in  $[0.5, 1.5]$ . For the discretization, we choose  $[S_{min}, S_{max}] = [0.01, 2]$  for the CEV model and we set  $\mathcal{N} = 1000$ , as well as  $\Delta t = 0.008$  for all models. We choose  $\sigma = 0.15$  and  $\zeta = 0.75$  and as benchmark solution we implement the semi-closed-form solution of the CEV model for European put and call prices as shown in Schroeder (1989). Summarizing the results, we observe that, with the introduced discretization, the absolute error between the benchmark and the FEM solution is in the region of  $10^{-3}$  to  $10^{-4}$ .

#### 3.3.2 Effects of de-Americanization on Pricing

First, we focus on pricing differences caused by de-Americanization. Therefore, we compare the de-Americanized American prices with the derived European option prices in the following way. Starting with a set of model parameters, we price the American and European options. Then, the binomial tree is applied to translate the American option prices into de-Americanized pseudo-European prices. Subsequently, we compare the European and the pseudo-European *so-called* de-Americanized prices to identify the effects of the de-Americanization methodology.

The advantage of this approach is that we can purely focus on de-Americanization, decoupled from calibration issues. In order to do so, we define the following test set for the range of investigated options. Here, we focus on put options due to the fact that American and European calls coincide for non-dividend-paying underlyings.

$$\begin{aligned}
 S_0 &= 1 \\
 K &= 0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10, 1.15, 1.20 \\
 T &= \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{6}{12}, \frac{9}{12}, \frac{12}{12}, \frac{24}{12} \\
 r &= 0, 0.01, 0.02, 0.05, 0.07
 \end{aligned} \tag{3.14}$$

### 3 Numerical Investigation of the de-Americanization Method

Five parameter sets are investigated to cover the parameter range. These are summarized in Table 3.1.

	$\sigma$	$\zeta$
$p_1$	0.2	0.5
$p_2$	0.275	0.6
$p_3$	0.35	0.7
$p_4$	0.425	0.8
$p_5$	0.5	0.9

Table 3.1: Overview of the parameter sets used for the CEV model

#### Motivation of the selected parameters for the CEV model

The main feature of the CEV model is the elasticity of variance parameter  $\zeta$ , which is combined with the level of the underlying to obtain a local volatility, namely  $\sigma(S, t) = \sigma S^{\zeta-1}$ , reflecting the leverage effect. In our example, we investigate American puts and the option-holder benefits from decreasing asset prices. In general, increasing the volatility leads to increasing option prices, but especially compared to the classical Black-Scholes model we are interested in the question of how strongly the incorporated leverage effect influences the put prices and whether the differences between American and European puts can be captured by the binomial tree. Thus, our selection for  $\zeta$  in  $p_1$  is 0.5, which strongly differs from the Black-Scholes model, and then  $\zeta$  is further increased up to 0.9 within the scenarios. Additionally, we increase the values of  $\sigma$ .

**Remark 3.3.1.** *We price the put options in (3.14) for the parameter sets shown in Table 3.1. For some parameters, especially for high interest rates combined with low volatility, it could occur that the price of an American put option equals exactly  $K_i - S_0$ , so that this American put option would be exercised immediately. In the following analysis, we excluded these cases because a unique European option price cannot be determined by applying the binomial tree. As illustrated in the following toy example in Figure 3.3, there are several possible values for  $u$  to replicate the American option price if the price of the American option is determined by immediately exercising it. In the example, a put option with strike  $K = 120$  is priced. Here,  $u = 1.04$  and  $u = 1.11$  are possible solutions. To avoid this, we consequently only consider American put options in our analyses when  $P_{put}^{Am} > (K - S_0)^+ \cdot (1 + \delta)$ . Thus, the American put option price exceeds the immediate exercise price by a factor of  $\delta$ . We set  $\delta = 1\%$ . In theory, we cannot guarantee that in all other cases the application of the binomial tree finds a unique  $u^*$  to replicate the American option price. However, in various empirical tests this has been the case and only when the price of an American put option equals exactly  $K_i - S_0$ , problems have been observed.*

In Tables A.1 - A.3 in the appendix, we show in the appendix the pricing effects for the synthetic prices in (3.14). For each scenario  $p_i, i = 1, \dots, 5$ , we present the average difference between the de-Americanized prices and the European prices for each maturity

3 Numerical Investigation of the de-Americanization Method

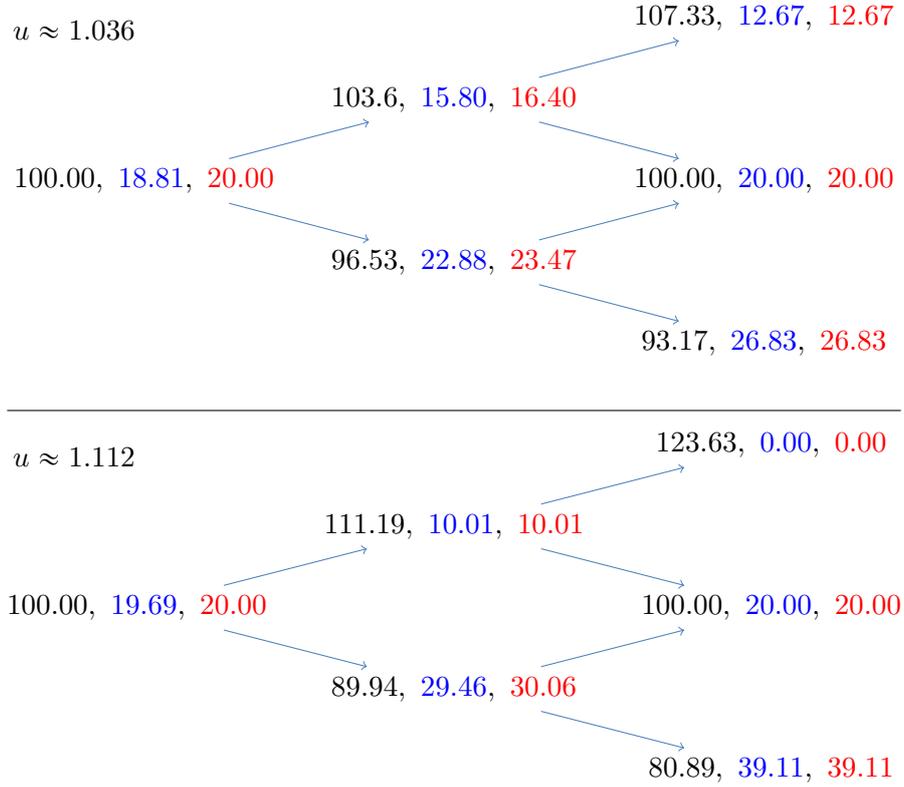


Figure 3.3: Given an American put option price of 20 with  $S_0 = 100$ ,  $K = 120$ ,  $r = 0.01$ , i.e., an American put option in the exercise region, a unique tree cannot be found to replicate this option. In this example, we show two binomial trees for  $u \approx 1.036$  (top) as well as  $u \approx 1.112$  (bottom). In each tree, we show the value of the underlying (black), the European put price (blue) and the American put price (red) at each node. Both trees replicate the American option price of 20.00 but result in different European put prices: 18.81 and 19.69.

### 3 Numerical Investigation of the de-Americanization Method

and each strike and accordingly show the maximal European price in this maturity to reflect the issue stated in Remark 3.3.1. Similar studies have been done for the maximal error at each strike and maturity and confirm the findings based on the average error presented in the following. In Figure 3.4, we highlight the results for scenario  $p_5$  in the CEV model to illustrate the effects of de-Americanization in several interest rate environments for different maturities or different strikes. For a better interpretability, in each of the figures the differences between the corresponding American and European option are shown, too. This figure clearly highlights the case  $r = 0$  as fewer de-Americanization effects.

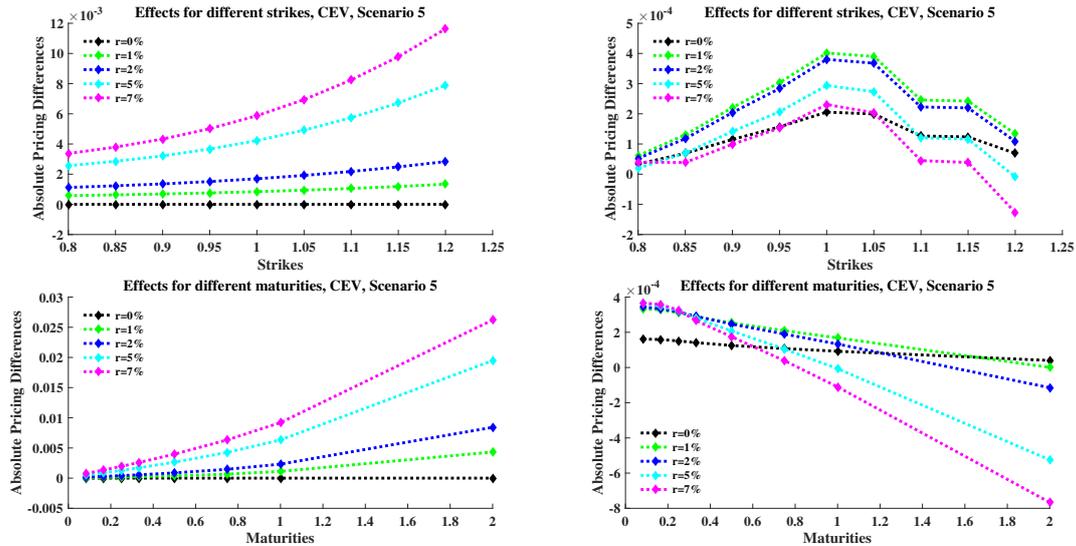


Figure 3.4: De-Americanization effects on pricing put options in the CEV model. As an example, the results are shown for  $p_5$  for the average error between the de-Americanized and the European prices for each strike (top right) and each maturity (bottom right). The average differences of the corresponding American and European prices is shown for each strike (top left) and each maturity (bottom left).

In general, for the CEV model, we observe that for short maturities the de-Americanized prices seem to overprice the European prices, whereas for longer maturities they seem to underprice the European options. We see that with increasing  $\sigma$  and  $\zeta$  parameters the maximal error increases and, overall, all parameter sets behave similarly. Focusing on the interest rate, we observe that for higher interest rates ( $r = 5\%$  and  $r = 7\%$ ) the average errors are higher or at least in a comparable region. Especially for higher interest rates, the maximal price has to be considered, because the higher the interest rate, the higher the probability that we did not consider some in-the-money options due to Remark 3.3.1 and that the options with high prices are neglected in this setting. Thus we deduce that the error increases with increasing interest rates and that at high

maturities the error increases for scenarios with higher volatility. For scenarios  $p_1$ ,  $p_2$  and  $p_3$ , we clearly observe that the effects of de-Americanization increase with increasing strikes. This means that for in-the-money options the de-Americanization effects tend to be stronger than for out-of-the-money options. This is consistent with the statements made by Carr and Wu (2010). However, for higher interest rates, the average error seems to decrease with increasing strikes. The test setting in (3.14) is defined for  $S = 1$ . In the CEV model, the volatility is scaled with  $S^\zeta^{-1}$ . In additional test, we set in (3.14)  $S = 100$  and scaled the strike values  $K$  with a factor of 100, too. Overall, we observe that the errors of the de-Americanization methodology, see Tables A.4 and A.5, give a similar picture to the results for  $S = 1$ . Naturally, by investigating differences of prices, the absolute number of the error is higher by a factor of 10 to 100, which is tolerable when the underlying value is scaled with a factor of 100.

In addition to all of these de-Americanization effects in absolute terms, we checked the magnitude of the relative error for the 1-year at-the-money put option, i.e., the absolute difference between the European and the de-Americanized price divided by the European price. In the CEV model, the average relative error for this option in all scenarios and interest rate settings was 0.1% with a peak of 0.17% at scenario  $p_2$  with  $r = 1\%$ .

Summarizing the results,

- de-Americanization effects are sensitive to interest rate. The higher the interest rates, the higher the observable pricing differences,
- de-Americanization effects increase with increasing volatility and increasing maturities,
- de-Americanization effects tend to be stronger in-the-money,
- de-Americanization effects increase with higher jump intensities as shown by the results for the Merton model in Burkovska et al. (2016).

Overall, in the settings mentioned above, we observe a systematic effect caused by de-Americanization. In the next step, we are interested in finding out whether these effects are also reflected in the calibration results.

### 3.3.3 Effects of de-Americanization on Calibration to Synthetic Data

Here, we study the de-Americanization effect on synthetic American market data. To this effect, in a first step, we generate artificial market data using our FEM implementations of the three considered models. In a second step, we calibrate each model to the previously generated market data. This methodology allows us to disregard the noise affiliated with real market data and thus enables us to study the effect of de-Americanization exclusively.

### 3 Numerical Investigation of the de-Americanization Method

Our artificial market data is specified as follows.

$$\begin{aligned}
 S_0 &= 1 \\
 r &= 7\% \\
 T_1 &= \frac{2}{12}, & K_1 &= \{0.95, 0.975, 1, 1.025, 1.05\}, \\
 T_2 &= \frac{6}{12}, & K_2 &= \{0.9, 0.925, K_1, 1.075, 1.1\}, \\
 T_3 &= \frac{9}{12}, & K_3 &= \{0.85, 0.875, K_2, 1.125, 1.15\}, \\
 T_4 &= 1, & K_4 &= \{0.8, 0.825, K_3, 1.175, 1.2\}, \\
 T_5 &= 2, & K_5 &= \{0.75, 0.775, K_4, 1.225, 1.25\}.
 \end{aligned} \tag{3.15}$$

As the data in (3.15) shows, we consider a high-interest market and a set of maturities ranging from rather short-term American options with 2 months maturity to long-term American products with 2 years maturity. Each maturity  $T_i$  is associated with a set of strikes  $K_i$ ,  $i \in \{1, \dots, 5\}$ . To analyze the effects of de-Americanization on pricing, we price these options for the five parameter scenarios in Table 3.1. Regarding the calibration methodology, we have to make two choices. First, we have to decide which option types to include and, second, we need to determine the objective function.

Regarding the choice of options, we first consider only put options for the whole strike trajectory due to the fact that, in our setting of non-dividend paying underlyings, American and European calls coincide. Thus, we include in-the-money as well as out-of-the-money options. Second, motivated by the fact that the value of out-of-the-money options does not include any intrinsic value and is therefore supposed to better reflect the randomness of the market (as mentioned in Carr and Wu (2010)), we consider as a second approach that only includes out-of-the money puts and out-of-the money calls for the whole set of strikes and maturities. Consequently, in this second study, for each  $i \in \{1, \dots, 5\}$ , we consider call option prices for maturities  $T_i$  and strikes  $k \in K_i$  with  $k > 1$  and put option prices for maturities  $T_i$  and strikes  $k \in K_i$  with  $k < 1$ . At-the-money option data, i.e., options with strike  $K = 1$ , is neglected.

Once the synthetic American market data has been generated, we create associated second synthetic market data by applying the de-Americanization routine using the binomial model.

Remark 3.3.1 and Figure 3.3 describe situations in which the de-Americanization routine yields non-unique results. In the calibration to de-Americanized prices, we exclude options that cannot be de-Americanized uniquely as explained by the following remark.

**Remark 3.3.2** (Disregarding non-unique de-Americanized prices). *As outlined above, we artificially generate American market data for a calibration study on synthetic data. In a first step, we calibrate to the generated American prices directly. In a second step, we de-Americanize the option data and calibrate to the resulting quasi-European options.*

### 3 Numerical Investigation of the de-Americanization Method

Here, we only consider option prices that admit a unique de-Americanized price. Consequently, all American put option prices that violate

$$P_{put}^{Am} > (K - S_0)^+ \cdot (1 + \delta), \quad \text{with } \delta = 1\%, \quad (3.16)$$

are not de-Americanized and thus are neglected in the second step.

The second crucial assumption is the objective function. A variety of objective functions are proposed in the literature, e.g., the root mean square error, the average absolute error as a percentage of the mean price, the average absolute error, the average relative percentage error, absolute price differences, relative price differences, absolute implied volatilities, relative implied volatilities (see for example Detlefsen and Haerdle (2006), Bauer (1991), Fengler (2005), Schoutens et al. (2004)).

We work directly with the observed prices and choose an objective function that considers prices, and due to the fact that the considered out-of-the-money option prices are rather small, we focus on absolute instead of relative differences. In the calibration, we take the absolute average squared error (aase) as the objective function and we minimize,

$$\text{aase} = \frac{1}{\#\text{options}} \sum_{\text{option}_k} |\text{Market price}_k - \text{Model price}_k|^2. \quad (3.17)$$

The results of the calibration to synthetic data are summarized in Table 3.2 for the CEV model for calibrating to put options and calibrating to out-of-the-money options.

Overall, we see that for the CEV model the parameters match well when calibrating to American options. When calibrating to de-Americanized prices however, the volatility parameter  $\sigma$  is underestimated in most cases and this underestimation is counterbalanced by an overestimated  $\zeta$ -value.

Summarizing the results, we observe that when calibrating de-Americanized synthetic data in a high-interest-rate environment for the continuous CEV model, the main parameters driving the volatility of the underlying,  $\zeta$  and  $\sigma$  (CEV), are often not exactly matched. In these cases, the application of the binomial tree is not able to capture the volatility of the underlying exactly. Furthermore, in Burkovska et al. (2016), we see that for the Heston model the parameters  $\xi$  and  $\rho$  are often mismatched and for the jump model (Merton), we observe that due to the de-Americanization the jump intensity is (more strongly) mismatched than when directly calibrating to American options and in these cases the wrongly calibrated jump intensity parameter may be compensated by adjusting the other model parameters accordingly.

### 3 Numerical Investigation of the de-Americanization Method

			CEV		
			$\sigma$	$\zeta$	aase
$p_1$	Put	true	0.2	0.5	—
		Am	0.1977	0.4962	7.74e-6
		DeAm	0.1894	0.4501	8.35e-5
	oom	Am	0.1997	0.4996	4.52e-6
		DeAm	0.1793	0.9609	2.75e-4
$p_2$	Put	true	0.275	0.6	—
		Am	0.2740	0.6004	4.98e-7
		DeAm	0.2607	0.7539	2.04e-6
	oom	Am	0.2736	0.5978	1.91e-6
		DeAm	0.2484	0.5367	1.10e-5
$p_3$	Put	true	0.35	0.7	—
		Am	0.3515	0.7576	4.37e-5
		DeAm	0.3272	0.8528	1.92e-4
	oom	Am	0.3476	0.6984	1.00e-4
		DeAm	0.3141	0.5527	5.99e-4
$p_4$	Put	true	0.425	0.8	—
		Am	0.4258	0.7898	1.53e-6
		DeAm	0.3942	0.8755	7.30e-6
	oom	Am	0.4262	0.7966	3.27e-6
		DeAm	0.3801	0.6009	1.96e-5
$p_5$	Put	true	0.5	0.9	—
		Am	0.4982	0.9036	1.53e-6
		DeAm	0.4570	0.9192	1.02e-5
	oom	Am	0.4986	0.9036	4.02e-6
		DeAm	0.4430	0.6549	2.38e-5

Table 3.2: Calibration results for calibrating to put options only and out-of-the-money options for the CEV model. Due to the effect of non-unique de-Americanization results, for the CEV model, some option prices have been neglected in the calibration to de-Americanized option data, as Remark 3.3.2 explains. In scenarios  $p_1$  to  $p_5$ , 5, 5, 10, 10 and 10 prices were excluded in the calibration to put options only.

#### 3.3.4 Effects of de-Americanization on Calibration to Market Data

In this section, we investigate the effects of de-Americanization by calibrating market data. The single stock of our choice is Google as an example of a non-dividend-paying stock. Table 3.3 gives an overview of the processed data for the calibration procedure. In total, we obtained a data set containing 482 options, with slightly more puts than calls. The risk-free interest rate for maturities of 1 month, 3 months, 6 months, 1 year

### 3 Numerical Investigation of the de-Americanization Method

and 2 years are taken from the U.S. Department of the Treasury<sup>1</sup> and have been linearly interpolated whenever necessary.

	Maturity	T	# of options	r
$T_1$	27.02.2015	0.07	47	0.0001
$T_2$	20.03.2015	0.13	49	0.000129508
$T_3$	17.04.2015	0.20	52	0.00017541
$T_4$	19.06.2015	0.38	87	0.00046087
$T_5$	18.09.2015	0.62	98	0.000955435
$T_6$	15.01.2016	0.95	101	0.001602174
$T_7$	20.01.2017	1.97	48	0.004786339

Table 3.3: Processed Google option data for  $t_0 = 02.02.2015$ ,  $S_0 = 523.76$

In order to structure the available data, we follow the methodology applied for the volatility index (VIX) by the Chicago board of exchange (CBOE (2009)):

- Only out-of-the-money put and call options are used
- The midpoint of the bid-ask spread for each option with strike  $K_i$  is considered
- Only options with non-zero bid prices are considered
- Once two puts with consecutive strike prices are found to have zero bid prices, no puts with lower strikes are considered for inclusion (same for calls)

Basically, by this selection procedure, we only select out-of-the-money options that (due to non-zero bid prices) can be considered as liquid. In general, an option price consists of two components reflecting the time value and the intrinsic value of the option. By focusing on out-of-the-money options, the intrinsic value effects are mostly neglected and the highest option price will be at-the-money. Additionally, the highest market activity is in the at-the-money and slightly out-of-the-money region. The calibration results are summarized in Table 3.4.

		CEV		
		$\sigma$	$\zeta$	aase
Google Data	Am	0.25	0.98	3
	DeAm	0.25	0.97	3.32

Table 3.4: Calibration results for calibrating to out-of-the-money put and call options combined.

Here, we observe hardly any differences in the parameters. This is in line with our observations in Section 3.3.2 for low-interest-rate environments. In these settings, American

<sup>1</sup>[www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield](http://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield)

and European puts almost coincide and, thus, there will hardly be any difference in the prices and it is only natural that we observe very similar calibration results.

### 3.3.5 Effects of de-Americanization in Pricing Exotic Options

Plain vanilla options are traded liquidly in the market and are used to calibrate models. Financial institutions use these calibrated models to price more exotic products such as barrier and lookback options. In this subsection, we analyze which influences different calibration results have on the accuracy of exotic option prices.

We analyze a down-and-out call option and a lookback option and hence translate differences in the calibrated model parameters into quantitative prices. The payoff  $\tilde{\mathcal{H}}_{DOC}(S(T))$  of a down-and-out call option with barrier  $B$  is given by

$$\tilde{\mathcal{H}}_{DOC}(S(T)) = (S(T) - K)^+ \cdot \mathbb{1}_{\min_{t \leq T} S(t) \geq B}. \quad (3.18)$$

In our setting, we set  $S_0 = 100$ , the barrier  $B$  to 90% of the initial underlying value and the strike  $K$  to 105% of the underlying value. For the lookback option, we choose the same strike and the payoff  $\tilde{\mathcal{H}}_{Lookback}(S(T))$  is

$$\tilde{\mathcal{H}}_{Lookback}(S(T)) = (\bar{S}(T) - K)^+, \quad \text{with } \bar{S}(T) = \max_{t \leq T} S(t). \quad (3.19)$$

We price these two exotic options for the calibrated parameters in Tables 3.2 and 3.4 via a standard Monte Carlo method with  $10^6$  sample paths, 400 time steps per year and antithetic variates as variance reduction technique. The results are shown in the following Table 3.5.

In  $p_1$  and  $p_2$  of the CEV model, the scenarios with relatively small volatility, we do not see any differences. Thus, in cases with small volatility and medium elasticity of variance  $\zeta$ , de-Americanization seems to perform well. In the other scenarios, we observe that the calibration of de-Americanized prices leads to higher exotic option prices if we calibrate put options only and lower exotic option prices if we calibrate out-of-the-money options. Thus, the typically lower calibrated  $\sigma$ -value in combination with an increased  $\zeta$ -value obtained by calibrating de-Americanized options has this effect on the pricing of exotic options.

In high-interest-rate environments, the de-Americanization methodology leads to different exotic options prices in the CEV model when the volatility of the underlying is higher. When using only put options, the exotic option prices tend to be higher; when considering out-of-the-money options, the exotic option prices tend to be lower. As additionally illustrated in Burkovska et al. (2016), in the Heston model, we observe a similar picture as in the CEV model, however here no general statement holds between higher and lower exotic option prices. Regarding the Merton model, the differences in the exotic option prices are more visible when considering the down-and-out barrier option.

		CEV		
			barrier	lookback
$p_1$	Put	true	9.93	10.11
		Am	9.93	10.10
		DeAm	9.93	10.01
	oom	Am	9.93	10.11
		DeAm	10.13	11.13
$p_2$	Put	true	10.13	11.14
		Am	10.14	11.14
		DeAm	11.47	14.59
	oom	Am	10.12	11.10
		DeAm	9.95	10.40
$p_3$	Put	true	11.60	14.93
		Am	12.48	17.83
		DeAm	13.53	23.85
	oom	Am	11.56	14.81
		DeAm	10.07	10.91
$p_4$	Put	true	13.56	24.08
		Am	13.54	24.00
		DeAm	14.14	30.87
	oom	Am	13.51	23.81
		DeAm	10.60	12.37
$p_5$	Put	true	14.76	43.40
		Am	14.80	43.99
		DeAm	14.78	43.50
	oom	Am	14.74	42.81
		DeAm	11.68	15.13
Google data	Am	14.21	32.10	
	DeAm	14.12	30.70	

Table 3.5: Overview of prices for barrier and lookback options

### 3.4 Conclusion

We investigated the de-Americanization methodology by performing accuracy studies to compare the empirical results of this approach to those obtained by solving related variational inequalities for local volatility, stochastic volatility and jump diffusion models. On page 31, we pose key questions regarding the robustness of the de-Americanization methodology with regard to changes in the (i) interest rates, (ii) maturities and (iii) in-the-money and out-of-the-money options.

First, focusing on pricing, we observe that de-Americanization causes larger errors (i) for higher interest rates, (ii) for higher maturities and (iii) in the in-the-money region

### 3 Numerical Investigation of the de-Americanization Method

in scenarios with higher volatility and/or correlation. Second, we investigate model calibration to synthetic data for a specified set of maturities and strikes in a high-interest-rate environment. Numerically, we observe noticeable differences in the calibration results of the de-Americanization methodology compared to the benchmark. For continuous models, the main difference lies in the resulting volatility parameters. When calibrating to Google data, hardly any differences occur, which can be explained by the very low-interest-rate environment. This is in line with the results for question (i).

In a final step, we investigate the effects of de-Americanization in the model calibration on pricing exotic options. Here, exotic option prices play the role of a measure of the distance between differently calibrated model parameters. In most cases, we observe that exotic option prices are reasonably close to the benchmark prices. However, we observe severe outliers for all investigated models. We find scenarios in which the exotic option prices differ by roughly 50% in the CEV model ( $p_4$ ), see Table 3.5.

Additionally, the results for the Heston and Merton model in Burkovska et al. (2016) are aligned to these observations. Additionally, as a jump diffusion model, the results for the Merton model also show higher de-Americanization effects for scenarios with higher jump intensities and that the jump intensity is underestimated by the de-Americanization methodology, especially in settings with high jump intensities.

In a nutshell, the methodological risk of de-Americanization critically depends on the interest rate environment.

- For low-interest-environments, the errors caused by de-Americanization are negligibly small and the de-Americanization methodology can be employed when fast run-times are preferred.
- For higher-interest-rate environments, however, de-Americanization leads to uncontrollable outliers.

Intuitively, the higher the interest rate, the higher is the early exercise premium of an American put option and, hence, the higher are the differences between an American and a European put price. At its core, the de-Americanization methodology is replacing American options with European options. Thus, with higher differences between both of them, the potential of making an error increases. As second observation,

- de-Americanization tends to lead to outliers in scenarios with a higher volatility and/or higher jump intensities.

In these scenarios, the model describing the evolution of the underlying differs strongly from the assumptions of the Black&Scholes model, namely a constant volatility coefficient and the absence of jumps. The binomial tree is roughly a discrete version of the Black&Scholes model and, secondly, the model independent approach of the binomial tree leads to higher errors the stronger the underlying process differs from the Black&Scholes model. Therefore, and since the de-Americanization methodology does not provide an

error control, we strongly recommend applying a pricing method in the calibration that is certified by error estimators.

Also for scenarios in which a direct application of the de-Americanization methodology can lead to outliers, the method can be applied usefully.

- It can, for example, be used as first estimator for the model parameters and this estimate can then be used as initial guess in a calibration routine to reduce the run-time.

We leave the inclusion of dividends for future research. The numerical results for the jump diffusion model and the sensitivity to interest rates indicate that discrete and continuous dividends may intensify the errors caused by the de-Americanization method. Furthermore, the application of a binomial tree to translate American into European prices can be challenged in further research and the binomial tree could be replaced by a more sophisticated, maybe not anymore model invariant, technique. Obviously, one has to keep in mind here that this leads to higher computational efforts and one of the advantages of the de-Americanization methodology, like faster run-times and model flexibility could be reduced.

### 3.5 Outlook: The Reduced Basis Method

Previously, we applied the finite element method as PDE solver for calibrating American option prices. The investigations have shown that the de-Americanization methodology does not work efficiently enough for all the parameter scenarios. Based on PDE methods, we propose the reduced basis method here as a complexity reduction technique and will give a quick overview of Burkovska et al. (2016b). Starting with the quote of Thomas Eakins in the introduction, colors can be represented by the RGB color model. In this additive model, each color can be constructed by adding weighted red, blue and green colors. The weights are usually scaled between 0 and 255. Each color can then be represented as a 3-tupel  $(a, b, c)$  with  $0 \leq a, b, c \leq 255$ , with the convention of black being  $(0, 0, 0)$  and white being  $(255, 255, 255)$ . In a fascinating way, all colors can be represented by only a few colors, namely the three colors red, green and blue. This color/painting example illustrates the idea behind the reduced basis method. In order to solve parametric PDEs, compared to the high-dimensional basis in the finite element method, for example, the key idea is to define a problem-adapted lower dimensional basis and thereby significantly reducing the complexity of numerically solving parametric partial differential equations. The problem-adapted basis functions are based on solutions of the PDE for specific parameters. By doing so, the resulting algebraic systems are considerably smaller. The set of few basis functions then basically works like the colors red, blue and green in the example above and each solution for a new parameter is a combination of solutions to these few parameters.

The reduced basis method (RBM) is not a new approach. The method has been studied on a variety of applications. We refer to Patera and Rozza (2006), Quarteroni et al. (2016), Hesthaven et al. (2016) for an overview. For model reduction techniques in finance, we find recent work with a focus on proper orthogonal decomposition (POD), e.g. Sachs and Schu (2008), Sachs and Schneider (2014), Sachs et al. (2014), Pironneau (2012), Peherstorfer et al. (2015) and RBM results, e.g. Cont et al. (2011), Pironneau (2011), Pironneau (2012). Here, the focus has been on the simpler case of European options. For American options, described by parabolic variational inequalities, we refer to Burkovska et al. (2015) and Balajewicz and Toivanen (2016). More generally, variational inequalities are also investigated by Glas and Urban (2014). The basis construction for parabolic inequalities becomes much more challenging than for variational equalities. We refer here to the POD-Angle-Greedy strategy as described in Burkovska et al. (2015). Cont et al. (2011) and Pironneau (2009) study the calibration with the RBM to European option, the extension to American options, to the best of our knowledge, has not yet been addressed.

In Burkovska et al. (2016b), we compare calibration results with the RBM to calibration results of the de-Americanization methodology. In a nutshell, the calibration results using the RBM method are closer to the calibration results applying the FEM solver than the de-Americanization methodology, especially in a high-interest rate setting with high volatility parameters. The de-Americanization methodology is still the fastest method to calibrate to American options. However, the reduced basis method as a complexity reduction technique allows to calibrate relatively fast to American options and tends to work fine in scenarios, for which the de-Americanization methodology runs into pitfalls, too.

### 3.6 Excursion: The Regularized Heston Model

*The work in this section started in 2014 in parallel to Ackerer et al. (2016). This section is an excursion to indicated possible ways to work with regularized models. Our example is a regularized version of the Heston model that falls as a special case into the investigated Jacobi stochastic volatility model of Ackerer et al. (2016).*

At the end of Section 2.2.1, we have indicated that the Heston model, due to an unbounded coefficient in the SDE describing the underlying, does not fit into the classic PDE theory. Moreover, the resulting PDE for bounded coefficients has regularity advantages and does not require a truncation of the domain. This motivates our approach to introduce a regularized Heston model with bounded coefficients and in addition, we show that our regularized Heston model converges to the original Heston model.

Our regularized Heston model is given by the following SDEs

$$dS(t) = \mu S dt + \sqrt{\tilde{v}(t)} S dW_1(t) \quad (3.20)$$

$$d\tilde{v}(t) = \kappa(\theta - \tilde{v}(t))dt + \sigma \sqrt{(\tilde{v}(t) - \epsilon) \cdot \left(1 - \frac{\tilde{v}(t)}{\bar{V}}\right)} dW_2(t), \kappa, \theta, \sigma > 0, \quad (3.21)$$

where  $W_1(t)$  and  $W_2(t)$  are Wiener processes with correlation  $\rho$  and the two constants  $0 < \epsilon < \bar{V} < \infty$  define the space for the volatility process, i.e.  $\tilde{v}(t) \in (\epsilon, \bar{V})$ . Basically, in contrast to the Heston model, the diffusion coefficient of the volatility process is bounded. In the following, we show first that the regularized Heston model, described by SDEs (3.20) and (3.21), has a strong solution and then, we investigate the convergence of the regularized Heston model to the Heston model.

### 3.6.1 Existence and Strong Solution in the Bounded Domain $I = (\epsilon, \bar{V})$

In the following, we show that the regularized volatility process  $\tilde{v}$  has a strong, unique solution within the interval  $(\epsilon, \bar{V})$ . Here, we have the issue that our coefficients are only defined on an open interval  $(\epsilon, \bar{V}) \subset \mathbb{R}$  and we have to prove that our process is well-defined and does not exit this interval. The standard results stated in Chapter 2 have to be adapted to the bounded domain. First, we introduce the concepts of a weak solution up to explosion and then the concept of a solution in an interval up to exit time, i.e. not the time until the process explodes, but the time until the process exits the interval.

**Definition 3.6.1** (Weak solution up to explosion time). (*Karatzas and Shreve, 1996, Definition 5.1, p. 329*) A weak solution up to an explosion time of the equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad (3.22)$$

is a triple  $(X, W), (\Omega, \mathcal{F}, P), \{\mathcal{F}_t\}$ , where

- (i)  $(\Omega, \mathcal{F}, P)$  is a probability space, and  $\{\mathcal{F}_t\}$  is a filtration of sub- $\sigma$ -fields of  $\mathcal{F}$  satisfying the usual conditions
- (ii)  $X = \{X_t, \mathcal{F}_t; 0 \leq t < \infty\}$  is a continuous, adapted  $\mathbb{R} \cup \{\infty\}$ -valued process,  $W = \{W_t, \mathcal{F}_t; 0 \leq t < \infty\}$  is a standard, one-dimensional Brownian motion
- (iii) with

$$S_n := \inf\{t \geq 0; |X_t| \geq n\} \quad (3.23)$$

we have for all  $0 \leq t < \infty$

$$P \left[ \int_0^{t \wedge S_n} \{|b(X_s)| + \sigma^2(X_s)\} ds < \infty \right] = 1, \quad (3.24)$$

valid for every  $n \geq 1$ , and

(iv)

$$P \left[ X_{t \wedge S_n} = X_0 + \int_0^t b(X_s) 1_{s \leq S_n} ds + \int_0^t \sigma(X_s) 1_{s \leq S_n} dW_s; \quad \forall 0 \leq t < \infty \right] = 1 \quad (3.25)$$

valid for every  $n \geq 1$ .

$S := \lim_{n \rightarrow \infty} S_n$  is the explosion time of the process  $X$ .

**Definition 3.6.2** (Non-degeneracy and local integrability). *For the SDE described in (3.22), we define the assumptions of non-degeneracy and local integrability in the following way,*

(Non-degeneracy) for all  $x \in \mathbb{R}$  holds  $\sigma^2 > 0$ ,

(Local integrability) for all  $x \in \mathbb{R}$  exists  $\epsilon > 0$  such that  $\int_{x-\epsilon}^{x+\epsilon} \frac{|b(y)|}{\sigma^2(y)} dy < \infty$ .

**Theorem 3.6.3.** (Karatzas and Shreve, 1996, Theorem 5.15, p.341) *Assume that  $\sigma^{-2}$  is locally integrable at every point in  $\mathbb{R}$ , and non-degeneracy and local integrability assumptions from Definition 3.6.2 hold. Then, for every initial distribution  $\mu$ , the equation (3.22) has a weak solution up to an explosion time, and this solution is unique in the sense of law.*

Of course, with our SDE in (3.21), we do not only care about an explosion in the sense of  $|X_t| = \infty$ , but we want to investigate if our process is leaving the interval of interest, namely  $(\epsilon, \bar{V})$ .

From this point on, we follow Karatzas and Shreve (1996) and do not look at the entire real line, but we do focus on the interval

$$I = (\epsilon, \bar{V}) \subset \mathbb{R}. \quad (3.26)$$

We show in Lemma 3.6.4 that our SDE (3.21) satisfies non-degeneracy and local integrability assumptions and we can introduce the definition of weak solutions in an Interval  $I$  until explosion (exit) time.

**Lemma 3.6.4.** *The SDE (3.21) with coefficients  $\sigma(x) = \sigma \sqrt{(x - \epsilon) \cdot (1 - \frac{x}{\bar{V}})}$  and  $b(x) = \kappa(\theta - x)$  satisfies the following non-degeneracy and local integrability assumptions in the interval  $I$ .*

*Proof.* The function  $x \mapsto \sigma(x)$  is a continuous function on  $I$  and the square-root of a polynomial of degree 2. Thus, there exists a maximum of two roots. Obviously,  $\sigma(\epsilon) = 0$  and  $\sigma(\bar{V}) = 0$  and  $\epsilon, \bar{V} \notin I$ . Hence, for all  $x \in I$  holds  $\Rightarrow \sigma^2(x) > 0$ .

For the local integrability, we define for  $x \in I$ ,  $\delta_x = \min\{\frac{x-\epsilon}{2}, \frac{\bar{V}-x}{2}\}$ . Thus, we achieve

### 3 Numerical Investigation of the de-Americanization Method

that for all  $y \in [x - \delta_x, x + \delta_x] \subset I : \sigma(y) < \infty, |b(y)| < \infty$ . As a direct consequence, the quotient  $\frac{1+|b(x)|}{\sigma^2(x)}$  is continuous on  $[x - \delta_x, x + \delta_x]$ , the local integrability is given.  $\square$

For SDEs of type (3.22) fulfilling non-degeneracy and local integrability assumptions, we can introduce the following definition of weak solutions in intervals.

**Definition 3.6.5** (Weak solution in interval up to exit time). (*Karatzas and Shreve, 1996, Definition 5.20, p. 343*) A weak solution up to exit time in the interval  $I=(l,r)$  of equation (3.22) is a triple  $(X, W), (\Omega, \mathcal{F}, P), \{\mathcal{F}_t\}$ , where

- (i)  $(\Omega, \mathcal{F}, P)$  is a probability space, and  $\{\mathcal{F}_t\}$  is a filtration of sub- $\sigma$ -fields of  $\mathcal{F}$  satisfying the usual conditions
- (ii)  $X = \{X_t, \mathcal{F}_t; 0 \leq t < \infty\}$  is a continuous, adapted  $[l, r]$ -valued process,  $W = \{W_t, \mathcal{F}_t; 0 \leq t < \infty\}$  is a standard, one-dimensional Brownian motion
- (iii) with  $\{l_n\}_{n=1}^\infty$  and  $\{r_n\}_{n=1}^\infty$  strictly monotone sequences satisfying  $l < l_n < r_n < r$ ,  $\lim_{n \rightarrow \infty} l_n = l$ ,  $\lim_{n \rightarrow \infty} r_n = r$ , and

$$S_n := \inf\{t \geq 0 : X_t \notin (l_n, r_n)\}$$

the equations (3.24) and (3.25) hold.

Here,  $S := \inf\{t \geq 0 : X_t \notin (l, r)\} = \lim_{n \rightarrow \infty} S_n$  is the exit time from  $I$ .

**Theorem 3.6.6.** The defined SDE (3.21) has a weak solution in the interval  $(\epsilon, \bar{V})$  up to exit time  $S$ .

*Proof.* (i) and (ii) of Definition 3.6.5 are satisfied. For (iii), we define for  $n \geq 1$ :  $l_n := \epsilon + \frac{1}{n} \frac{\bar{V} - \epsilon}{3}$  and  $r_n := \bar{V} - \frac{1}{n} \frac{\bar{V} - \epsilon}{3}$ . Obviously,  $\lim_{n \rightarrow \infty} l_n = \epsilon$ ,  $\lim_{n \rightarrow \infty} r_n = \bar{V}$ . The equations (3.24) and (3.25) are satisfied due to the fact that for  $t < S_n$  ( $S_n := \inf\{t \geq 0 : X_t \notin (l_n, r_n)\}$ ) the process  $X_t$  is bounded by  $|X_t| < \max\{|l_n|, |r_n|\}$  and the non-degeneracy and local integrability assumptions are satisfied (Lemma 3.6.4). Additionally, we adapt Theorem 3.6.3 for the interval  $(\epsilon, \bar{V})$ . Thus, there exists a weak solution in the interval  $(\epsilon, \bar{V})$  up to exit time  $S$ .  $\square$

Consequently, we want to show that the process  $X$  given by (3.21) does not exit the interval  $(\epsilon, \bar{V})$ , thus we will show  $S := \inf\{t \geq 0 : X_t \notin (\epsilon, \bar{V})\} = \lim_{n \rightarrow \infty} S_n = \infty$ . Here, one way is Feller's test for explosions.

**Theorem 3.6.7** (Feller's test for explosion). (*Karatzas and Shreve, 1996, Theorem 5.29, p. 348*) Assume that non-degeneracy and local integrability assumptions hold, and let  $(X, W), (\Omega, \mathcal{F}, P), \{\mathcal{F}_t\}$  be a weak solution in  $I = (l, r)$  of (3.22) with non-random initial condition  $X_0 = x \in I$ . Then  $P[S = \infty] = 1$  or  $P[S = \infty] < 1$ , according to whether

$$v(l+) = v(r-) = \infty$$

### 3 Numerical Investigation of the de-Americanization Method

or not. Here, for all  $x \in I$  and  $c \in I$  arbitrary,

$$\begin{aligned} v(x) &:= \int_c^x (p(x) - p(y))m(dy), \\ p(x) &:= \int_c^x \exp\left(-2 \int_c^\eta \frac{b(\zeta)}{\sigma^2(\zeta)} d\zeta\right) d\eta \text{ (scale function),} \\ m(dx) &:= \frac{2dx}{p'(x)\sigma^2(x)} \text{ (speed measure).} \end{aligned}$$

*Proof.* (Karatzas and Shreve, 1996, p. 348f) □

**Lemma 3.6.8.** *In the setting of Theorem 3.6.7, the following implications hold:*

$$\begin{aligned} p(r-) = \infty &\Rightarrow v(r-) = \infty, \\ p(l+) = -\infty &\Rightarrow v(l+) = \infty \end{aligned}$$

*Proof.* Let  $p(r-) = \infty$ ,

$$\begin{aligned} v(r-) &:= \int_c^x (p(r-) - p(y)) \frac{2dy}{p'(y)\sigma^2(y)} \\ &= \int_c^x (\infty - p(y)) \frac{2dy}{p'(y)\sigma^2(y)} \\ &= \infty. \end{aligned}$$

Let  $p(l+) = -\infty$ , we use  $c > x$  to switch the limits of the integral,

$$\begin{aligned} v(l+) &:= \int_c^x (p(l+) - p(y)) \frac{2dy}{p'(y)\sigma^2(y)} \\ &= \int_c^x (-\infty - p(y)) \frac{2dy}{p'(y)\sigma^2(y)} \\ &= - \int_x^c (-\infty - p(y)) \frac{2dy}{p'(y)\sigma^2(y)} \\ &= \infty. \end{aligned}$$

□

**Lemma 3.6.9.** *Assuming  $\frac{\sigma^2(\bar{V}-\epsilon)}{\sqrt{\bar{V}-\sqrt{\epsilon}}} \leq 2\kappa \min\{\bar{V} - \theta, \theta - \epsilon\}$ , every local weak solution of (3.21) in  $I$  is a global weak solution.*

*Proof.* This statement follows directly from (Ackerer et al., 2016, Theorem 1) which explicitly defines the condition. □

### 3 Numerical Investigation of the de-Americanization Method

So far, we have shown that our SDE (3.21) has a global weak solution existing in the interval  $I = (\epsilon, \bar{V})$ . Now, we are interested in showing uniqueness and stating that we have a global strong solution existing in the interval  $I$ .

**Theorem 3.6.10.** *With the assumptions of Lemma 3.6.9 and  $X_0 = x \in I := (\epsilon, \bar{V})$ , (3.21) has a global unique strong solution in  $I$ .*

*Proof.* We follow (Revuz and Yor, 1999, Theorem 3.5, p. 390) which states that pathwise uniqueness holds for (3.21) if  $|\sigma(x) - \sigma(y)|^2 \leq \rho(|x - y|)$  and  $b$  is Lipschitz continuous, where  $\rho$  is a Borel function on  $(0, \infty)$  with  $\int_{0+} \frac{1}{\rho(\alpha)} d\alpha = \infty$ .  $\rho(x) := \sigma^2 \left( \frac{\bar{V} - \epsilon}{\bar{V}} \right) x$  satisfies these conditions and we have to investigate  $|\sigma(x) - \sigma(y)|^2$  only for  $x, y \in (\epsilon, \bar{V})$ . Let  $x, y \in (\epsilon, \bar{V})$ ,

$$\begin{aligned} |\sigma(x) - \sigma(y)| &= \left| \sigma \sqrt{(x - \epsilon) \cdot \left(1 - \frac{x}{\bar{V}}\right)} - \sigma \sqrt{(y - \epsilon) \cdot \left(1 - \frac{y}{\bar{V}}\right)} \right| \\ &= \sigma \left| \sqrt{(x - \epsilon) \cdot \left(1 - \frac{x}{\bar{V}}\right)} - \sqrt{(y - \epsilon) \cdot \left(1 - \frac{y}{\bar{V}}\right)} \right| \\ &\leq \sigma \sqrt{\left| (x - \epsilon) \cdot \left(1 - \frac{x}{\bar{V}}\right) - (y - \epsilon) \cdot \left(1 - \frac{y}{\bar{V}}\right) \right|}. \end{aligned}$$

Given this estimate for  $|\sigma(x) - \sigma(y)|$ , rearranging yields

$$\begin{aligned} &= \sigma \sqrt{\left| x - \frac{x^2}{\bar{V}} - \epsilon + \frac{\epsilon x}{\bar{V}} - \left( y - \frac{y^2}{\bar{V}} - \epsilon + \frac{\epsilon y}{\bar{V}} \right) \right|} = \sigma \sqrt{\left| x - y - \frac{x^2 - y^2}{\bar{V}} + \frac{\epsilon(x - y)}{\bar{V}} \right|} \\ &= \sigma \sqrt{\left| \frac{\bar{V}(x - y)}{\bar{V}} - \frac{(x - y)(x + y)}{\bar{V}} + \frac{\epsilon(x - y)}{\bar{V}} \right|} = \sigma \sqrt{\left| \frac{\bar{V} - (x + y) + \epsilon}{\bar{V}} (x - y) \right|}. \end{aligned}$$

Finally, using  $x, y \in (\epsilon, \bar{V})$  results in  $|\sigma(x) - \sigma(y)| \leq \sigma \sqrt{\left| \frac{\bar{V} - \epsilon}{\bar{V}} (x - y) \right|}$  and, thus,

$$|\sigma(x) - \sigma(y)|^2 \leq \sigma^2 \left( \frac{\bar{V} - \epsilon}{\bar{V}} \right) |x - y|.$$

For the drift component, we can show, the even stronger condition, that it is globally Lipschitz. Let  $x, y \in (\epsilon, \bar{V})$

$$\begin{aligned} |b(x) - b(y)| &= |\kappa(\theta - x) - \kappa(\theta - y)|, \quad |\kappa| > 0 \\ &= \kappa |(\theta - x) - (\theta - y)| = \kappa |\theta - x - \theta + y| \\ &= \kappa |-x + y| = \kappa |x - y| \end{aligned}$$

Thus, pathwise uniqueness holds. From (Karatzas and Shreve, 1996, Corollary 3.23, p. 310) it follows that weak existence and pathwise uniqueness imply strong existence. From

this point, we can build a bridge back to the Yamada and Watanabe Proposition 2.1.3 and just set  $h(x) := \rho(x)^{\frac{1}{2}}$ .  $\square$

### 3.6.2 Convergence

After the introduction of the regularized Heston model, we embed the regularized Heston model into the Heston model by showing, that for  $\epsilon \rightarrow 0$  and  $\bar{V} \rightarrow \infty$  the regularized Heston model converges to the Heston model in appropriate sense and, further on, we show an upper bound for the error. Let us start with the two SDEs of the Heston

$$\begin{aligned} dv(t) &= b(v(t))dt + \sigma(v(t))dW(t), \kappa, \theta, \sigma > 0, \\ b(v(t)) &= \kappa(\theta - v(t)), \\ \sigma(v(t)) &= \sigma\sqrt{v(t)}, \end{aligned}$$

and the regularized Heston model

$$\begin{aligned} d\tilde{v}(t) &= \tilde{b}(\tilde{v}(t))dt + \tilde{\sigma}(\tilde{v}(t))dW(t), \kappa, \theta, \sigma > 0 \\ \tilde{b}(\tilde{v}(t))dt &= \kappa(\theta - \tilde{v}(t))dt = b(\tilde{v}(t)), \\ \tilde{\sigma}(\tilde{v}(t)) &= \sigma\sqrt{(\tilde{v}(t) - \epsilon) \cdot \left(1 - \frac{\tilde{v}(t)}{\bar{V}}\right)}. \end{aligned}$$

In order to investigate the convergence, we define  $\{v_n(t)\}_{n=1}^{\infty}$

$$\begin{aligned} dv_n(t) &= b_n(v_n(t))dt + \sigma_n(v_n(t))dW(t), \kappa, \theta, \sigma > 0 \\ b_n(v_n(t))dt &= \kappa(\theta - v_n(t))dt = b(v_n(t)), \\ \sigma_n(v_n(t)) &= \sigma\sqrt{(v_n(t) - \epsilon_n) \cdot \left(1 - \frac{v_n(t)}{\bar{V}_n}\right)} \\ \epsilon_n &= \frac{1}{n} \cdot \epsilon \\ \bar{V}_n &= n \cdot \bar{V}. \end{aligned}$$

Obviously,  $\epsilon_n \rightarrow 0$  and  $\bar{V}_n \rightarrow \infty$ . In order to show convergence in law, we apply Theorem IX. 4.8 from Jacod and Shiryaev (2003).

**Theorem 3.6.11.** *The regularized Heston model converges in law to the Heston model, i.e. the process  $\tilde{v}_n(t)$  converges in law to the process  $v(t)$ .*

*Proof.* By applying Theorem 3.6.10, we know that for every  $n \in N$  there exists a unique global solution for the process  $\tilde{v}_n(t)$ . Thus, to apply Theorem IX. 4.8 from Jacod and Shiryaev (2003), we only have to show that the coefficient functions  $\tilde{b}_n(x)$  and  $\tilde{\sigma}_n(x)$  converge locally uniformly to  $b(x)$  resp.  $\sigma(x)$ . Due to  $\tilde{b}_n(x) = b(x)$  for all  $n \in N$  the

### 3 Numerical Investigation of the de-Americanization Method

locally uniformly convergence is naturally given. For the diffusion coefficient, we have to show the locally uniformly convergence. Clearly, for all  $x \in I$  exists  $m \in \mathbb{N}$  such that  $\epsilon_m \leq x \leq \bar{V}_m$  and then for an open domain  $U \subset I$ , we have to show that for  $y \in U$  and  $n \geq m$  the diffusion coefficient converges uniformly. Let  $n \geq m$ ,

$$\begin{aligned} |\sigma_n(y) - \sigma(y)| &= \left| \sigma \sqrt{(y - \epsilon_n) \left(1 - \frac{y}{\bar{V}_n}\right)} - \sigma \sqrt{y} \right| = \sigma \left| \sqrt{(y - \epsilon_n) \left(1 - \frac{y}{\bar{V}_n}\right)} - \sqrt{y} \right| \\ &\leq \sigma \sqrt{\left| (y - \epsilon_n) \left(1 - \frac{y}{\bar{V}_n}\right) - y \right|} = \sigma \sqrt{\left| y - \frac{y^2}{\bar{V}_n} - \epsilon_n + \frac{\epsilon_n y}{\bar{V}_n} \right|} \\ &= \sigma \sqrt{\left| \frac{y^2}{\bar{V}_n} + \epsilon_n - \frac{\epsilon_n y}{\bar{V}_n} \right|} \leq \sigma \sqrt{\left| \frac{y^2}{\bar{V}_n} + \epsilon_n \right|} \leq \sigma \left( \sqrt{\left| \frac{y^2}{\bar{V}_n} \right|} + \sqrt{|\epsilon_n|} \right) \end{aligned}$$

For  $(n \rightarrow \infty)$  it holds  $\sigma \left( \sqrt{\left| \frac{y^2}{\bar{V}_n} \right|} + \sqrt{|\epsilon_n|} \right) \rightarrow 0$ . Thus, the convergence in law follows directly from (Jacod and Shiryaev, 2003, Theorem IX. 4.8).  $\square$

As a last step, we investigate the convergence rate for the processes  $v(t)$  and  $v_n(t)$ .

**Theorem 3.6.12.** *Given a fix interval  $[\epsilon, V] \subset (\epsilon_n, \bar{V}_n)$ , the approximation error between the regularized Heston model and the Heston model (the processes are stopped whenever one of them is leaving  $[\epsilon, V]$ ) with  $C_n = \sqrt{\left| \frac{V^2 + \epsilon_n V}{\bar{V}_n} + \epsilon_n \right|}$  is given by*

$$E[|v^\tau(t) - \tilde{v}_n^\tau(t)|^2] = 4 \frac{C_n^2}{2\kappa^2 + 4 \left( \frac{\sigma}{\sqrt{2\epsilon}} \right)^2} \left( \exp \left( (2\kappa^2 + 4 \left( \frac{\sigma}{\sqrt{2\epsilon}} \right)^2) t \right) - 1 \right) \quad (3.27)$$

Choosing  $\epsilon_n$  and  $\bar{V}_n$  such that  $C_n \rightarrow 0$  ( $n \rightarrow \infty$ ), the approximation error decays to zero,

$$E[|v^\tau(t) - \tilde{v}_n^\tau(t)|^2] \rightarrow 0 \quad (n \rightarrow \infty) \quad (3.28)$$

*Proof.* In order to do an analysis of the convergence rate, we need to have Lipschitz coefficient functions. This is generally not satisfied by the diffusion coefficients. We therefore define the following stopping times.

$$\tau_1 := \inf\{s : \tilde{v}(s) < \epsilon\} \quad (3.29)$$

$$\tau_2 := \inf\{s : v(s) < \epsilon\} \quad (3.30)$$

$$\tau_3 := \inf\{s : \tilde{v}(s) > V\} \quad (3.31)$$

$$\tau_4 := \inf\{s : v(s) > V\} \quad (3.32)$$

$$\tau := \min\{\tau_1, \tau_2, \tau_3, \tau_4\} \quad (3.33)$$

### 3 Numerical Investigation of the de-Americanization Method

Finally, we investigate the convergence rate for the processes  $v^\tau(t)$  and  $v_n^\tau(t)$  stopped at  $t \wedge \tau$  in the interval  $[\epsilon, V] \subset (\epsilon_n, \bar{V}_n)$ .

$$\begin{aligned} E[|v^\tau(t) - v_n^\tau(t)|^2] &\leq 2E\left[\left|\int_0^{t \wedge \tau} b(v(s)) - b(v_n(s)) ds\right|^2\right] \\ &\quad + 2E\left[\left|\int_0^{t \wedge \tau} \sigma(v(s)) - \sigma_n(v_n(s)) dW_s\right|^2\right], \end{aligned}$$

Applying Itô isometry yields

$$\begin{aligned} &\leq 2E\left[\int_0^{t \wedge \tau} |b(v(s)) - b(v_n(s))|^2 ds\right] \\ &\quad + 2E\left[\int_0^{t \wedge \tau} |\sigma(v(s)) - \sigma_n(v_n(s))|^2 ds\right] \end{aligned}$$

Now, we exploit  $(a + b)^2 \leq 2(a^2 + b^2)$ ,

$$\begin{aligned} &= 2E\left[\int_0^{t \wedge \tau} |b(v(s)) - b(v_n(s))|^2 ds\right] \\ &\quad + 2E\left[\int_0^{t \wedge \tau} |\sigma(v(s)) - \sigma_n(v(s)) + \sigma_n(v(s)) - \sigma_n(v_n(s))|^2 ds\right] \\ &\leq 2E\left[\int_0^{t \wedge \tau} |b(v(s)) - b(v_n(s))|^2 ds\right] \\ &\quad + 4E\left[\int_0^{t \wedge \tau} |\sigma(v(s)) - \sigma_n(v(s))|^2 ds\right] \\ &\quad + 4E\left[\int_0^{t \wedge \tau} |\sigma_n(v(s)) - \sigma_n(v_n(s))|^2 ds\right] \end{aligned}$$

As a next step, the stopping time is taken into the expectation via the indicator function,

$$\begin{aligned} &\leq 2E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |b(v(s)) - b(v_n(s))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |\sigma(v(s)) - \sigma_n(v(s))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |\sigma_n(v(s)) - \sigma_n(v_n(s))|^2 ds\right] \\ &\leq 2E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |b(v(s \wedge \tau)) - b(v_n(s \wedge \tau))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t \mathbb{1}_{\{s \leq \tau\}} |\sigma_n(v(s \wedge \tau)) - \sigma_n(v_n(s \wedge \tau))|^2 ds\right] \end{aligned}$$

### 3 Numerical Investigation of the de-Americanization Method

The indicator function is bounded by 1, which yields for the expected error,

$$\begin{aligned} &\leq 2E\left[\int_0^t |b(v(s \wedge \tau)) - b(v_n(s \wedge \tau))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t |\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2 ds\right] \\ &\quad + 4E\left[\int_0^t |\sigma_n(v(s \wedge \tau)) - \sigma_n(v_n(s \wedge \tau))|^2 ds\right] \end{aligned}$$

Applying Fubini then yields,

$$\begin{aligned} &\leq 2 \int_0^t E[|b(v(s \wedge \tau)) - b(v_n(s \wedge \tau))|^2] ds \\ &\quad + 4 \int_0^t E[|\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2] ds \\ &\quad + 4 \int_0^t E[|\sigma_n(v(s \wedge \tau)) - \sigma_n(v_n(s \wedge \tau))|^2] ds. \end{aligned}$$

From the proof of Theorem 3.6.10 we know that the coefficient function  $b$  is Lipschitz. Thus, with Lipschitz constant  $\kappa$ ,  $|b(x) - b(y)| \leq \kappa|x - y| \Rightarrow |b(x) - b(y)|^2 \leq \kappa^2|x - y|^2$ . This leads directly to

$$\begin{aligned} E[|v^\tau(t) - v_n^\tau(t)|^2] &\leq 2\kappa^2 \int_0^t E[|v(s \wedge \tau) - v_n(s \wedge \tau)|^2] ds \\ &\quad + 4 \int_0^t E[|\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2] ds \\ &\quad + 4 \int_0^t E[|\sigma_n(v(s \wedge \tau)) - \sigma_n(v_n(s \wedge \tau))|^2] ds \end{aligned}$$

From Lemma 3.6.13, we know that on  $[\epsilon, V]$  the coefficient function  $\sigma_n$  is Lipschitz with constant  $L := \frac{\sigma}{\sqrt{2\epsilon}}$ .

$$\begin{aligned} E[|v^\tau(t) - v_n^\tau(t)|^2] &\leq (2\kappa^2 + 4L^2) \int_0^t E[|v(s \wedge \tau) - v_n(s \wedge \tau)|^2] ds \\ &\quad + 4 \int_0^t E[|\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2] ds \end{aligned}$$

Lastly, we have to find an upper bound for the term  $E[|\sigma(v(s \wedge \tau)) - \sigma_n(v(s \wedge \tau))|^2]$ ,

### 3 Numerical Investigation of the de-Americanization Method

due to the definition of  $\tau$  only on the interval  $[\epsilon, V]$ . Let  $x \in [\epsilon, V]$

$$\begin{aligned} |\sigma(x) - \tilde{\sigma}_n(x)| &= \left| \sqrt{x} - \sqrt{(x - \epsilon_n)\left(1 - \frac{x}{\bar{V}_n}\right)} \right| \leq \sqrt{\left| x - (x - \epsilon_n)\left(1 - \frac{x}{\bar{V}_n}\right) \right|} \\ &= \sqrt{\left| \frac{x^2}{\bar{V}_n} + \epsilon_n - \frac{\epsilon_n x}{\bar{V}_n} \right|} \leq \sqrt{\left| \frac{x^2}{\bar{V}_n} + \epsilon_n + \frac{\epsilon_n x}{\bar{V}_n} \right|} \\ &\leq \sqrt{\left| \frac{V^2 + \epsilon_n V}{\bar{V}_n} + \epsilon_n \right|} =: C_n \end{aligned}$$

Thus, it follows,

$$E[|v^\tau(t) - v_n^\tau(t)|^2] \leq (2\kappa^2 + 4L^2) \int_0^t E[|v(s \wedge \tau) - v_n(s \wedge \tau)|^2] ds + 4tC_n^2.$$

For notation purposes, we set at this point:  $K_1 = 2\kappa^2 + 4L^2$  and apply Gronwall's inequality, see Gronwall (1919),

$$\begin{aligned} E[|v^\tau(t) - v_n^\tau(t)|^2] &\leq 4tC_n^2 + \int_0^t 4sC_n^2 * K_1 e^{\int_s^t K_1 du} ds \\ &= 4tC_n^2 + \int_0^t 4sC_n^2 * K_1 e^{K_1(t-s)} ds \\ &= 4tC_n^2 + 4C_n^2 * K_1 e^{K_1 t} \int_0^t s e^{-K_1 s} ds \end{aligned}$$

Solving the integral  $\int_0^t s e^{-K_1 s} ds$  yields,

$$\begin{aligned} &= 4tC_n^2 + 4C_n^2 * K_1 e^{K_1 t} \left[ -\frac{e^{-K_1 s}(K_1 s + 1)}{K_1^2} \right]_{s=0}^t = 4tC_n^2 + 4C_n^2 * K_1 e^{K_1 t} \frac{1 - e^{-K_1 t}(K_1 t + 1)}{K_1^2} \\ &= 4C_n^2 \left( t + \frac{1}{K_1} (e^{K_1 t} - e^{K_1 t - K_1 t}(K_1 t + 1)) \right) = 4C_n^2 \left( t + \frac{1}{K_1} (e^{K_1 t} - (K_1 t + 1)) \right) \\ &= 4 \frac{C_n^2}{K_1} (e^{K_1 t} - 1) = 4 \frac{C_n^2}{2\kappa^2 + 4 \left( \frac{\sigma}{\sqrt{2\epsilon}} \right)^2} \left( \exp \left( (2\kappa^2 + 4 \left( \frac{\sigma}{\sqrt{2\epsilon}} \right)^2) t \right) - 1 \right) \end{aligned}$$

From the definition of  $C_n = \sqrt{\left| \frac{V^2 + \epsilon_n V}{\bar{V}_n} + \epsilon_n \right|}$  we can directly conclude (due to  $\epsilon_n \rightarrow 0$  and  $\bar{V}_n \rightarrow \infty$  ( $n \rightarrow \infty$ )) that  $C_n \rightarrow 0$  and thus,  $C_n^2 \rightarrow 0$  for ( $n \rightarrow \infty$ ). We therefore know, for ( $n \rightarrow \infty$ ),

$$E[|v^\tau(t) - v_n^\tau(t)|^2] \rightarrow 0$$

□

### 3 Numerical Investigation of the de-Americanization Method

**Lemma 3.6.13.** *The function  $\sigma_n(v_n(t)) = \sigma\sqrt{(v_n(t) - \epsilon_n) \cdot (1 - \frac{v_n(t)}{V_n})}$  is Lipschitz on the interval  $[\epsilon, V] \subset (\epsilon_n, \bar{V}_n)$ .*

*Proof.* From the proof of Theorem 3.6.10, we can directly jump to the part

$$\begin{aligned} \frac{|\sigma_n(x) - \sigma_n(y)|}{|x - y|} &\leq \frac{\sigma\sqrt{\frac{|\bar{V}_n - (x+y) + \epsilon_n|}{V_n}}\sqrt{|x - y|}}{|x - y|} \\ &= \frac{\sigma\sqrt{\frac{|\bar{V}_n - (x+y) + \epsilon_n|}{V_n}}\sqrt{|x - y|}}{\sqrt{|x - y|}\sqrt{|x + y|}} \\ &= \frac{\sigma\sqrt{\frac{|\bar{V}_n - (x+y) + \epsilon_n|}{V_n}}}{\sqrt{|x + y|}} \end{aligned}$$

At this point, we know that  $x, y \in [\epsilon, V] \subset (\epsilon_n, \bar{V}_n)$  and that the quotient  $\frac{|\bar{V}_n - (x+y) + \epsilon_n|}{V_n}$  is smaller than 1. This leads to

$$\frac{|\sigma_n(x) - \sigma_n(y)|}{|x - y|} \leq \frac{\sigma}{\sqrt{|x + y|}} \quad (3.34)$$

Finally, the quotient takes its maximum value for the minimum values for  $x$  and  $y$ , which is due to  $x, y \in [\epsilon, V] \subset (\epsilon_n, \bar{V}_n)$  given by

$$\frac{|\sigma_n(x) - \sigma_n(y)|}{|x - y|} \leq \frac{\sigma}{\sqrt{2\epsilon}}. \quad (3.35)$$

Thus, the Lipschitz constant is given by  $\frac{\sigma}{\sqrt{2\epsilon}}$ . □

# 4 Chebyshev Polynomial Interpolation Method

I had a polynomial once. My doctor removed it.

---

Michael Grant

*This chapter is based on Gaß et al. (2016) and Glau and Mahlstedt (2016), and presents the parts to which I mainly contributed.*

## 4.1 Chebyshev Polynomial Interpolation

In the previous chapter, we have seen, given an example of calibrating American option prices, that recurrent tasks, such as parametric option pricing, can be a computationally-challenging procedure. Further on, we look for complexity reduction techniques to decrease run-times while maintaining accuracy. In this chapter, we will interpret option prices as functions of the parameters and then apply polynomial interpolation to the parameters. We will see that this complexity reduction technique is decomposed in a so-called online and offline-part. Once the coefficients of the polynomial are determined in the offline-phase, option pricing in the online-phase is reduced to the evaluation of a polynomial which is a task with hardly any computational costs. Referring to Michael Grant's rather amusing quote, a rather skeptical approach towards polynomial interpolation may result from the Runge's phenomenon, Runge (1901). Applying polynomial interpolation of a function  $f$  on a domain, the natural intuition is to fix the interpolating polynomials on equidistantly-spaced interpolation points, so-called nodal points. Runge (1901) shows that this equidistant nodal point approach may lead to a polynomial interpolation with oscillations. He showed that polynomial interpolation on equidistantly-spaced grids may diverge, even for analytic functions. In Figure 4.1, we present the interpolation of the Runge function  $f(x) = \frac{1}{1+25x^2}$  on the domain  $[-1, 1]$  with 5 (top) and 10 (bottom) equidistantly-spaced nodal points.

Obviously, by increasing the number of nodal points to  $N = 10$ , the absolute error increases towards the limits  $-1$  and  $1$  of the domain. Overall, this is an example in which the maximal interpolation error over the domain, i.e.  $\|f(x) - p(x)\|_\infty$  with  $p(x)$  denoting the interpolating polynomial, is not decreasing with increasing  $N$ . This may be a reason

#### 4 Chebyshev Polynomial Interpolation Method

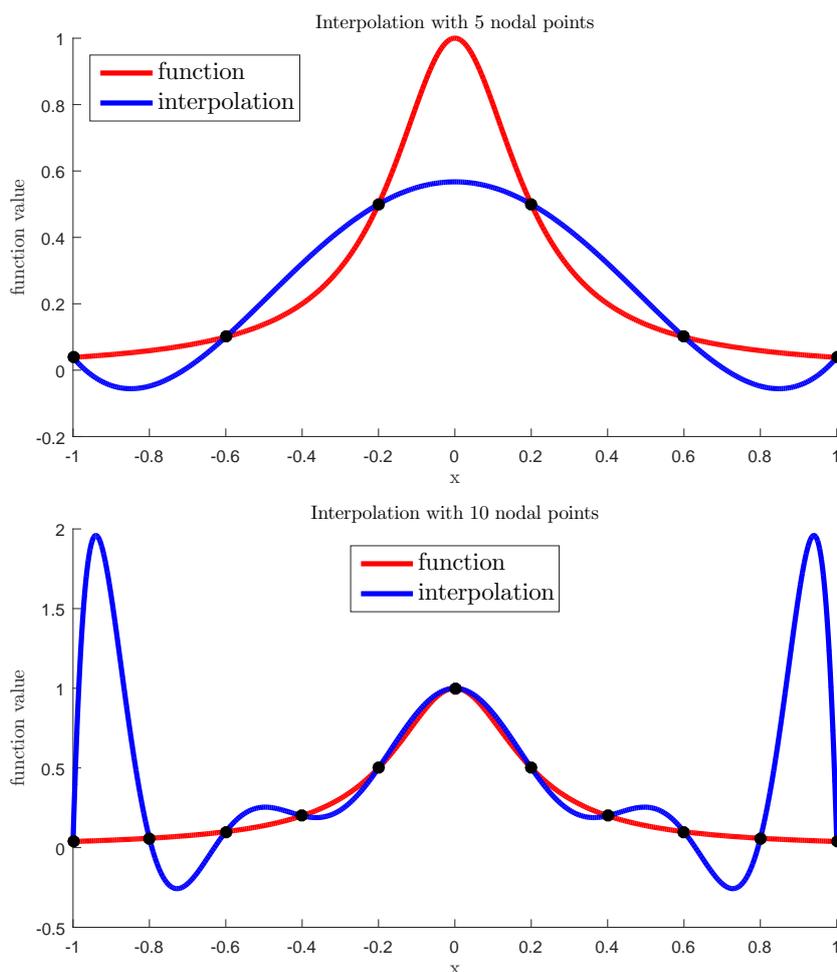


Figure 4.1: Polynomial interpolation of the Runge function  $f(x) = \frac{1}{1+25x^2}$  with equidistantly-spaced nodal points. We use 5 (top) and 10 (bottom) nodal points. These are marked in black.

to call a doctor. He would diagnose that this is connected to the increasing derivatives of the function  $f$  at the bound. In general, a polynomial of degree  $N$  is determined uniquely if the value of the polynomial at  $N + 1$  points is specified, see (Davis, 1975, Theorem 2.1.1). As Runge's phenomenon has shown, equidistantly-chosen nodal points are not a good choice. For functions  $f$  that are  $n$  times continuously differentiable on  $[-1, 1]$ , and for which  $f^{(n+1)}$  exists on  $(-1, 1)$ , (Davis, 1975, Theorem 3.1.1) provides a bound for the interpolation error. If the function  $f$  is approximated by a polynomial  $p_n$ , fixed on nodal points  $x_0, \dots, x_n$ , the error is given by

$$f(x) - p_n(x) = \frac{\prod_{i=0}^n (x - x_i)}{(n+1)!} f^{(n+1)}(\zeta), \quad (4.1)$$

## 4 Chebyshev Polynomial Interpolation Method

where  $\zeta \in (-1, 1)$  depends on the choice of the nodal points and the specific  $x$ . By replacing in (4.1)  $f^{(n+1)}(\zeta)$  with  $\max_{-1 < y < 1} f^{(n+1)}(y)$ , clearly an upper bound for the error is found. The remaining term  $\frac{\prod_{i=0}^n (x-x_i)}{(n+1)!}$  does not depend on the function  $f$  and, therefore, the choice of nodal points effects the error bound. Although we see here the derivative of the function  $f$  influencing the error bound, later on we will work mostly with the assumption that the function  $f$  is analytic. Following Davis (1975), Chebyshev points are the way to minimizing the product  $\prod_{i=0}^N (x-x_i)$  and hence, the interpolation error. As explained in Trefethen (2013), there are two sets of Chebyshev points. One is connected to the zeros of Chebyshev polynomials, the other one is connected to the extrema of a Chebyshev polynomial. The above mentioned minimizing property from Davis (1975) is connected to the zeros of the Chebyshev polynomial based on the fact that  $\prod_{i=0}^N (x-x_i)$  is a monic polynomial of degree  $N+1$  and on choosing the nodal points in such a way that  $\prod_{i=0}^N (x-x_i) = \frac{T_{N+1}(x)}{2^N}$ . Here,  $T_{N+1}(x)$  is the Chebyshev polynomial of degree  $N+1$  and by the scaling factor,  $\frac{T_{N+1}(x)}{2^N}$  is a monic polynomial. For the monic Chebyshev polynomial it holds that  $\max_{x \in [-1, 1]} \frac{T_{N+1}(x)}{2^N} \leq \max_{x \in [-1, 1]} p_{N+1}(x)$ , where  $p_{N+1}$  is an arbitrary monic polynomial of degree  $N+1$ . Hence, by choosing the zeros of  $T_{N+1}(x)$  as nodal points for  $\prod_{i=0}^N (x-x_i)$ , the minimizing property follows. In the following, we will introduce the Chebyshev polynomials and some main properties.

**Remark 4.1.1.** *The Chebyshev polynomials trace back to the Russian mathematician, Pafnuty Lvovich Chebyshev. Interestingly, the Chebyshev polynomials are denoted with the letter  $T$ . This traces back to the works of the French mathematician, Bernstein, who used the letter  $T$  based on French transliterations, such as "Tchebischeff", see Trefethen (2013).*

### 4.1.1 Chebyshev Polynomials

We start with an introduction of Chebyshev polynomials and some basic properties. Here, we follow Rivlin (1990) and Trefethen (2013).

**Definition 4.1.2.** *On  $[-1, 1]$ , the  $n$ -th Chebyshev polynomial ( $n \in \mathbb{N}$ ) is defined as*

$$T_n(x) = \cos(n \cdot \arccos(x)).$$

Trefethen (2013) also gives the interpretation that the  $n$ -th Chebyshev polynomial can be defined as the real part of  $z^n$  on the unit circle,

$$\begin{aligned} x &= \frac{1}{2}(z + z^{-1}) = \cos(\theta), \quad \theta = \arccos(x), \\ T_n(x) &= \frac{1}{2}(z^n + z^{-n}) = \cos(n\theta). \end{aligned}$$

The Chebyshev polynomials are defined on the interval  $x \in [-1, 1]$ . Figure 4.2 shows

#### 4 Chebyshev Polynomial Interpolation Method

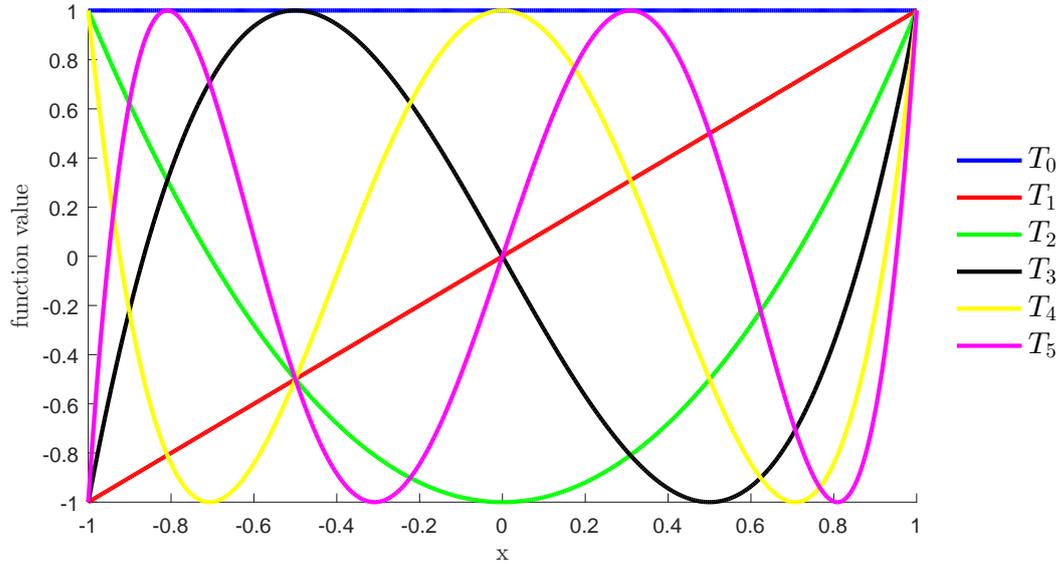


Figure 4.2: The first 6 Chebyshev polynomials  $T_0(x), \dots, T_5(x)$  on the interval  $[-1, 1]$ .

for  $n = 0, \dots, 5$  the Chebyshev polynomials  $T_n$ . The figure indicates graphically that the Chebyshev polynomial  $T_n$  is a polynomial of degree  $n$ . Rivlin (1990) shows explicitly that each Chebyshev polynomial is a polynomial of degree  $n$  and derives the coefficients as follows, denoting with  $[x]$  the greatest integer not exceeding  $x$ ,

$$T_n(x) = \sum_{i=0}^n t_i x^i$$

$$t_{n-(2k+1)} = 0, \quad k = 0, \dots, \left[ \frac{n-1}{2} \right] \quad (4.2)$$

$$t_{n-(2k)} = (-1)^k \sum_{j=k}^{\lfloor n/2 \rfloor} \binom{n}{2j} \binom{j}{k}, \quad k = 0, \dots, \left[ \frac{n}{2} \right].$$

The Chebyshev polynomial  $T_n(x)$  has  $n$  roots  $\hat{x}_1, \dots, \hat{x}_n$ ,

$$\hat{x}_i = \cos \left( \frac{2i-1}{n} \frac{\pi}{2} \right), \quad i = 1, \dots, n, \quad (4.3)$$

and takes at  $n+1$  points  $x_0, \dots, x_n$ ,

$$x_i = \cos \left( \frac{i\pi}{n} \right), \quad i = 0, \dots, n, \quad (4.4)$$

the extrema value  $\pm 1$ .

#### 4 Chebyshev Polynomial Interpolation Method

In addition, the following recursion formula for  $n \geq 1$  for the Chebyshev polynomials holds,

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x). \quad (4.5)$$

With (4.5), we conclude that in Figure 4.2, the Chebyshev polynomials are given as  $T_0(x) = 1$ ,  $T_1(x) = x$ ,  $T_2(x) = 2x^2 - 1$ ,  $T_3(x) = 4x^3 - 3x$ ,  $T_4(x) = 8x^4 - 8x^2 + 1$  and  $T_5(x) = 16x^5 - 20x^3 + 5x$ . The choice of Chebyshev polynomials for polynomial interpolation is connected to the orthogonality properties of the Chebyshev polynomials. First, there is, as shown in Rivlin (1990), the continuous orthogonality relationship with a weight function  $w(x) := \frac{1}{\sqrt{1-x^2}}$ ,

$$\int_{-1}^1 T_j(x)T_i(x) \cdot \frac{1}{\sqrt{1-x^2}} dx = 0, \quad \text{for } i \neq j \quad (4.6)$$

$$\int_{-1}^1 T_j^2(x) \cdot \frac{1}{\sqrt{1-x^2}} dx = \begin{cases} \frac{\pi}{2}, & j \neq 0, \\ \pi, & j = 0. \end{cases} \quad (4.7)$$

Hence, the Chebyshev polynomials  $\{T_n(x)\}_{n=0}^{\infty}$  form a sequence of orthogonal polynomials on  $[-1, 1]$  with the weight function  $w(x)$ . Later, this property will be applied in the derivation of error bounds for the polynomial interpolation with Chebyshev polynomials. By interpolating a given function with a polynomial of degree  $n$ ,  $n + 1$  nodal points have to be fixed. As regards Chebyshev polynomial interpolation, there are two possible ways. Either the  $n + 1$  roots of  $T_{n+1}(x)$ , see (4.3), are used as nodal points, referred to as Chebyshev points of the first kind, or the  $n + 1$  points at which  $T_n(x)$  takes its extreme values, see (4.4), are taken. In this case, the points are also referred to as Chebyshev-Lobatto points, Chebyshev extreme points, or Chebyshev points of the second kind and, following Trefethen (2013), these points are applied more in practice. For this reason, we later focus on this interpolation nodes and refer to them as Chebyshev points or nodal points. By choosing either the roots  $\hat{x}_i$  or the extremal points  $x_i$  as nodal points, orthogonality properties of Chebyshev polynomials can be exploited for the setting of the coefficients. We state both orthogonality properties taken from Rivlin (1990) and, for the first one, provide a proof for an understanding of this orthogonality property and get a feeling for Chebyshev polynomials.

**Proposition 4.1.3.** *Denoting with  $x_k$ ,  $k = 0, \dots, N$  the extremal points of  $T_N(x)$  as in (4.4), the following property holds for  $0 \leq i, j \leq N$ , denoting with " that the first and last summand are halved,*

$$\sum_{k=0}^N {}''T_j(x_k)T_i(x_k) = \begin{cases} 0, & i \neq j, \\ \frac{N}{2}, & i = j \neq 0 \text{ and } i = j \neq N, \\ N, & i = j = 0 \text{ or } i = j = N. \end{cases} \quad (4.8)$$

#### 4 Chebyshev Polynomial Interpolation Method

*Proof.* During the proof, we will apply the following trigonometric calculation rules.

$$\cos(x) \cdot \cos(y) = \frac{1}{2} [\cos(x+y) + \cos(x-y)] \quad (4.9)$$

$$\sum_{k=0}^N \cos(a+k \cdot b) = \frac{\sin\left(\frac{(N+1)b}{2}\right) \cdot \cos\left(a + \frac{Nb}{2}\right)}{\sin\left(\frac{b}{2}\right)} \quad (4.10)$$

$$\sin(x) \cdot \cos(y) = \frac{1}{2} [\sin(x+y) + \sin(x-y)] \quad (4.11)$$

$$\frac{\sin(x+y)}{\sin(x)} = \sin(y) \cot(x) + \cos(y) \quad (4.12)$$

Let  $i \neq j$ , then

$$\begin{aligned} \sum_{k=0}^N T_j(x_k) T_i(x_k) &= \sum_{k=0}^N \cos\left(\frac{ik\pi}{N}\right) \cos\left(\frac{jk\pi}{N}\right) \\ &\stackrel{(4.9)}{=} \frac{1}{2} \sum_{k=0}^N \left[ \cos\left(\frac{k(i+j)\pi}{N}\right) + \cos\left(\frac{k(i-j)\pi}{N}\right) \right] \end{aligned} \quad (4.13)$$

By applying (4.10) with  $a = 0$  and  $b = \frac{(i+j)\pi}{N}$  respectively  $b = \frac{(i-j)\pi}{N}$ , this yields,

$$\begin{aligned} (4.13) &= \frac{1}{2} \left[ \frac{\sin\left(\frac{N+1}{2} \frac{(i+j)\pi}{N}\right) \cos\left(\frac{N}{2} \frac{(i+j)\pi}{N}\right)}{\sin\left(\frac{(i+j)\pi}{2N}\right)} + \frac{\sin\left(\frac{N+1}{2} \frac{(i-j)\pi}{N}\right) \cos\left(\frac{N}{2} \frac{(i-j)\pi}{N}\right)}{\sin\left(\frac{(i-j)\pi}{2N}\right)} \right] \\ &= \frac{1}{4} \left[ \frac{\sin\left(\frac{2N+1}{2} \frac{(i+j)\pi}{N}\right) + \sin\left(\frac{(i+j)\pi}{2N}\right)}{\sin\left(\frac{(i+j)\pi}{2N}\right)} + \frac{\sin\left(\frac{2N+1}{2} \frac{(i-j)\pi}{N}\right) + \sin\left(\frac{(i-j)\pi}{2N}\right)}{\sin\left(\frac{(i-j)\pi}{2N}\right)} \right] \\ &= \frac{1}{4} \left[ 2 + \frac{\sin\left(\frac{2N+1}{2} \frac{(i+j)\pi}{N}\right)}{\sin\left(\frac{(i+j)\pi}{2N}\right)} + \frac{\sin\left(\frac{2N+1}{2} \frac{(i-j)\pi}{N}\right)}{\sin\left(\frac{(i-j)\pi}{2N}\right)} \right] \\ &= \frac{1}{4} \left[ 2 + \frac{\sin\left(\frac{(i+j)\pi}{2N} + (i+j)\pi\right)}{\sin\left(\frac{(i+j)\pi}{2N}\right)} + \frac{\sin\left(\frac{(i-j)\pi}{2N} + (i-j)\pi\right)}{\sin\left(\frac{(i-j)\pi}{2N}\right)} \right] \\ &\stackrel{(4.12)}{=} \frac{1}{4} \left[ 2 + \sin((i+j)\pi) \cot\left(\frac{(i+j)\pi}{2N}\right) + \cos((i+j)\pi) \right] \\ &\quad + \frac{1}{4} \left[ \sin((i-j)\pi) \cot\left(\frac{(i-j)\pi}{2N}\right) + \cos((i-j)\pi) \right] \end{aligned}$$

#### 4 Chebyshev Polynomial Interpolation Method

From  $\sin((i + j)\pi) = 0 = \sin((i - j)\pi)$ , it directly follows,

$$\begin{aligned}
 (4.13) &= \frac{1}{4} [2 + \cos((i + j)\pi) + \cos((i - j)\pi)] \\
 &= \frac{1}{4} [2 + 2 \cos(i\pi) \cos(j\pi)] \\
 &= \frac{1}{2} [1 + \cos(i\pi) \cos(j\pi)] \\
 &= \frac{1}{2} [T_i(x_0)T_j(x_0) + T_i(x_N)T_j(x_N)] \\
 &\Rightarrow \sum_{k=0}^N T_j(x_k)T_i(x_k) = 0.
 \end{aligned}$$

Let  $i = j = 0$ , then

$$\sum_{k=0}^N T_0(x_k)T_0(x_k) = \frac{1}{2} [T_0(x_0)T_0(x_0) + T_0(x_N)T_0(x_N)] + \sum_{k=1}^{N-1} 1 = N. \quad (4.14)$$

Let  $i = j = N$ , then

$$\sum_{k=0}^N T_N(x_k)T_N(x_k) = \sum_{k=0}^N \cos(k\pi) \cos(k\pi) = N.$$

Let  $1 \leq i = j < N$ , then

$$\begin{aligned}
 \sum_{k=0}^N T_i(x_k)T_i(x_k) &= \sum_{k=0}^N \cos\left(\frac{ik\pi}{N}\right) \cos\left(\frac{ik\pi}{N}\right) \\
 &\stackrel{(4.9)}{=} \frac{1}{2} \sum_{k=0}^N \left[ \cos\left(\frac{2ik\pi}{N}\right) + 1 \right] \\
 &= \frac{N+1}{2} + \frac{1}{2} \sum_{k=0}^N \cos\left(\frac{2ik\pi}{N}\right) \\
 &\stackrel{(4.10)}{=} \frac{N+1}{2} + \frac{1}{2} \frac{\sin\left(\frac{(N+1)2\pi i}{2N}\right) \cos(\pi i)}{\sin\left(\frac{2\pi i}{2N}\right)}.
 \end{aligned}$$

#### 4 Chebyshev Polynomial Interpolation Method

Here, we distinguish two cases. First let  $i$  be even, then  $\cos(\pi i) = 1$  and,

$$\begin{aligned} \sum_{k=0}^N T_i(x_k)T_i(x_k) &= \frac{N+1}{2} + \frac{1}{2} \frac{\sin\left(\frac{(N+1)2\pi i}{2N}\right)}{\sin\left(\frac{2\pi i}{2N}\right)} \\ &\stackrel{(4.12)}{=} \frac{N+1}{2} + \frac{1}{2} \left[ \sin(\pi i) \cot\left(\frac{2\pi i}{2N}\right) + \cos(\pi i) \right] \\ &= \frac{N+1}{2} + \frac{1}{2} = \frac{N}{2} + 1. \end{aligned}$$

Let  $i$  be odd, then  $\cos(\pi i) = -1$  and,

$$\begin{aligned} \sum_{k=0}^N T_i(x_k)T_i(x_k) &= \frac{N+1}{2} - \frac{1}{2} \frac{\sin\left(\frac{(N+1)2\pi i}{2N}\right)}{\sin\left(\frac{2\pi i}{2N}\right)} \\ &\stackrel{(4.12)}{=} \frac{N+1}{2} - \frac{1}{2} \left[ \sin(\pi i) \cot\left(\frac{2\pi i}{2N}\right) + \cos(\pi i) \right] \\ &= \frac{N+1}{2} - \frac{1}{2} \cdot (-1) = \frac{N}{2} + 1. \end{aligned}$$

In both cases, it holds  $T_i(x_0)T_i(x_0) = 1$  and  $T_i(x_N)T_i(x_N) = 1$ .

Therefore,  $\sum_{k=0}^N T_i(x_k)T_i(x_k) = \frac{N}{2}$ . □

**Proposition 4.1.4.** Denoting with  $\hat{x}_k$ ,  $k = 1, \dots, N+1$  the roots of  $T_{N+1}(x)$  as in (4.3), the following property holds for  $0 \leq i, j \leq N$

$$\sum_{k=1}^N T_j(\hat{x}_k)T_i(\hat{x}_k) = \begin{cases} 0, & i \neq j, \\ \frac{N}{2}, & i = j \neq 0, \\ N, & i = j = 0. \end{cases} \quad (4.15)$$

#### 4.1.2 Chebyshev Polynomial Interpolation

In the following, we want to approximate a function  $f : [-1, 1] \rightarrow \mathbb{R}$  via Chebyshev polynomial interpolation, i.e. we want to approximate  $f$  by a sum of Chebyshev polynomials from degree 0 until  $N$ ,

$$f(x) \approx \sum_{j=0}^N c_j T_j(x). \quad (4.16)$$

The coefficients  $c_j$ ,  $j = 1, \dots, N$  are defined by evaluating the function  $f$  at specific interpolation points. Recalling the definition and properties of Chebyshev polynomials, the Chebyshev polynomial of degree  $N$  has  $N+1$  extreme points at which either the value 1 or  $-1$  is taken. Exactly these points  $x_k = \cos\left(\frac{k\pi}{N}\right)$ ,  $k = 0, \dots, N$  are taken as

#### 4 Chebyshev Polynomial Interpolation Method

interpolation points. Therefore, the coefficients  $c_j$ ,  $j = 1, \dots, N$  have to be defined in such a way, that

$$f(x_k) = \sum_{j=0}^N c_j T_j(x_k), \quad (4.17)$$

holds. In a first step we assume, that (4.17) holds and both sides of the equation are multiplied by  $T_i(x_k)$ ,  $i = 0, \dots, N$ . This leads to,

$$f(x_k)T_i(x_k) = \sum_{j=0}^N c_j T_j(x_k)T_i(x_k). \quad (4.18)$$

Next, we summarize over all interpolation points  $x_k$  and due the orthogonality properties of the Chebyshev polynomials we take the first and last summand with factor  $\frac{1}{2}$ , which is indicated by " at the sigma sign,

$$\sum_{k=0}^N " f(x_k)T_i(x_k) = \sum_{k=0}^N " \sum_{j=0}^N c_j T_j(x_k)T_i(x_k). \quad (4.19)$$

To finally apply the orthogonality, we rearrange the equation in the following way,

$$\sum_{k=0}^N " f(x_k)T_i(x_k) = \sum_{j=0}^N c_j \left[ \sum_{k=0}^N " T_j(x_k)T_i(x_k) \right]. \quad (4.20)$$

In this case, the orthogonality of the Chebyshev polynomials (4.8) leads to the following coefficients.

$$c_0 = \frac{1}{N} \sum_{k=0}^N " f(x_k)T_0(x_k), \quad (4.21)$$

$$c_j = \frac{2}{N} \sum_{k=0}^N " f(x_k)T_j(x_k), \quad 1 \leq j \leq N-1, \quad (4.22)$$

$$c_N = \frac{1}{N} \sum_{k=0}^N " f(x_k)T_N(x_k). \quad (4.23)$$

We condense the definition of the Chebyshev interpolation method coefficients in the following equation,

$$c_j = \frac{2^{1_{0 < j < N}}}{N} \sum_{k=0}^N " f(x_k)T_j(x_k), \quad 0 \leq j \leq N. \quad (4.24)$$

Here, we briefly discuss the second way, using the zeros of the Chebyshev polynomial.

#### 4 Chebyshev Polynomial Interpolation Method

Obviously, by setting up (4.16) not the zeros of  $T_N(x)$  are taken because there are only  $N$  zeros and additionally, the summand including  $T_N(x)$  would be zero at all interpolation points. Here the zeros of  $T_{N+1}(x)$  are taken. They are given by  $\hat{x}_k = \cos\left(\frac{2k-1}{N+1}\frac{\pi}{2}\right)$ ,  $k = 1, \dots, N+1$ . Taking the  $\hat{x}_k$ ,  $k = 1, \dots, N+1$  as interpolation points we can approach similarly to the previous case with the extreme points as interpolation points and start with

$$f(\hat{x}_k) = \sum_{j=0}^N c_j T_j(\hat{x}_k). \quad (4.25)$$

Again, both sides of the equation are multiplied by  $T_i(x_k)$ ,  $i = 0, \dots, N$  and summarize over all interpolation points  $\hat{x}_k$ ,

$$\sum_{k=1}^{N+1} f(\hat{x}_k) T_i(\hat{x}_k) = \sum_{k=1}^{N+1} \sum_{j=0}^N c_j T_j(\hat{x}_k) T_i(\hat{x}_k). \quad (4.26)$$

Note, that here we do not have to halve the first and the last summand. Rearranging terms leads to,

$$\sum_{k=1}^{N+1} f(\hat{x}_k) T_i(\hat{x}_k) = \sum_{j=0}^N c_j \left[ \sum_{k=1}^{N+1} T_j(\hat{x}_k) T_i(\hat{x}_k) \right]. \quad (4.27)$$

At this point we apply again the orthogonality properties of Chebyshev polynomials, (4.15) and achieve,

$$c_j = \frac{2^{1_{j>0}}}{N+1} \sum_{k=1}^{N+1} f(x_k) T_j(x_k), \quad 0 \leq j \leq N. \quad (4.28)$$

#### 4.1.3 Multivariate Chebyshev Interpolation

The Chebyshev polynomial interpolation has a tensor based extension to the multivariate case. In order to obtain a nice notation, consider interpolation of functions

$$f(x), \quad x \in [-1, 1]^D. \quad (4.29)$$

For a more general hyperrectangular parameter space  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$ , the appropriate linear transformations need to be performed. Let  $\bar{N} := (N_1, \dots, N_D)$  with  $N_i \in \mathbb{N}_0$  for  $i = 1, \dots, D$ . The interpolation with  $\prod_{i=1}^D (N_i + 1)$  summands is given by

$$I_{\bar{N}}(f(\cdot))(x) := \sum_{j \in J} c_j T_j(x), \quad (4.30)$$

#### 4 Chebyshev Polynomial Interpolation Method

where the summation index  $j$  is a multiindex ranging over  $J := \{(j_1, \dots, j_D) \in \mathbb{N}_0^D : j_i \leq N_i \text{ for } i = 1, \dots, D\}$ , i.e.

$$I_{\overline{N}}(f(\cdot))(x) = \sum_{j_1=0}^{N_1} \cdots \sum_{j_D=0}^{N_D} c_{(j_1, \dots, j_D)} T_{(j_1, \dots, j_D)}(x). \quad (4.31)$$

The basis functions  $T_j$  for  $j = (j_1, \dots, j_D) \in J$  are defined by

$$T_j(x_1, \dots, x_D) = \prod_{i=1}^D T_{j_i}(x_i). \quad (4.32)$$

The coefficients  $c_j$  for  $j = (j_1, \dots, j_D) \in J$  are given by

$$c_j = \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right) \sum_{k_1=0}^{N_1} \cdots \sum_{k_D=0}^{N_D} f(x^{(k_1, \dots, k_D)}) \prod_{i=1}^D \cos\left(j_i \pi \frac{k_i}{N_i}\right), \quad (4.33)$$

where  $\sum''$  indicates that the first and last summand are halved and the Chebyshev nodes  $x^k$  for multiindex  $k = (k_1, \dots, k_D) \in J$  are given by

$$x^k = (x_{k_1}, \dots, x_{k_D}) \quad (4.34)$$

with the univariate Chebyshev nodes  $x_{k_i} = \cos\left(\pi \frac{k_i}{N_i}\right)$  for  $k_i = 0, \dots, N_i$  and  $i = 1, \dots, D$ . In Figure 4.3, we show for the one-dimensional, two-dimensional and three-dimensional Chebyshev polynomial interpolation the nodal points by setting  $N = 10$  in each dimensions. This results in one dimension to 11 nodal points, in two dimensions to 121 ( $= 11^2$ ) nodal points and finally to 1331 ( $= 11^3$ ) nodal points in three dimensions. This already indicates that the approach with tensorized Chebyshev polynomials suffers from the curse of dimensionality. Additionally, for the one-dimensional case we also present an interpretation of the Chebyshev nodes as shown in Trefethen (2013).  $N + 1$  equidistantly spaced points on the upper half of the unit circle and the projection to the x-axis.

## 4.2 Convergence Results of the Chebyshev Interpolation Method

In this section, we investigate the error bounds for analytic and differentiable functions. The main motivation in improving the existing error bounds in the literature is that sharper error bounds reduce computational cost, because they allow to use less nodal points to achieve a given accuracy. Especially in combination with the Monte-Carlo technique to evaluate the function at the nodal points, this is beneficial.

In the univariate case, it is well known that the error of approximation with Chebyshev polynomials decays polynomially for differentiable functions and exponentially for ana-

#### 4 Chebyshev Polynomial Interpolation Method

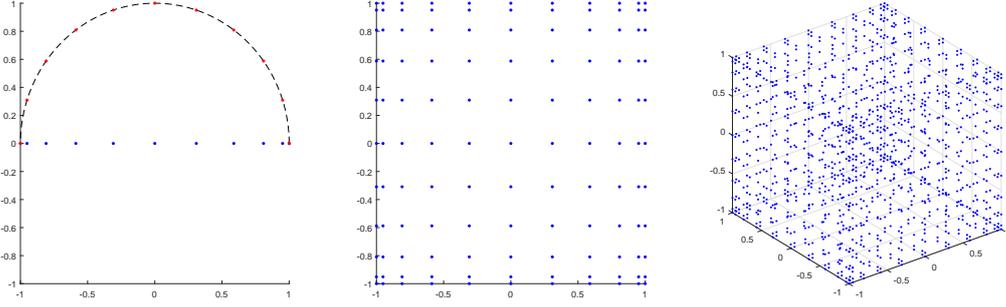


Figure 4.3: Nodal points in blue of the one-dimensional (left), two-dimensional (middle) and three-dimensional (right) Chebyshev polynomial interpolation by setting  $N = 10$  in each dimensions. For the one-dimensional case, we also show in red  $N + 1$  equidistantly spaced points on the upper half of the unit circle.

lytic functions. Let  $f$  be analytic in  $[-1, 1]$ , then it has an analytic extension to some *Bernstein ellipse*  $B([-1, 1], \varrho)$  with parameter  $\varrho > 1$ , defined as the open region in the complex plane bounded by the ellipse with foci  $\pm 1$  and semiminor and semimajor axis lengths summing up to  $\varrho$ . This and the following result traces back to the seminal work of Bernstein (1912).

**Theorem 4.2.1.** (*Trefethen, 2013, Theorem 8.2*) *Let a function  $f$  be analytic in the open Bernstein ellipse  $B([-1, 1], \varrho)$ , with  $\varrho > 1$ , where it satisfies  $|f| \leq V$  for some constant  $V > 0$ . Then for each  $N \geq 0$ ,*

$$\|f - I_N(f)\|_{L^\infty([-1,1])} \leq 4V \frac{\varrho^{-N}}{\varrho - 1}.$$

In the multivariate case we will extend a convergence result from Sauter and Schwab (2004). We consider parametric option prices of form

$$f(x) \quad \text{for } x \in \mathcal{X} \tag{4.35}$$

with  $\mathcal{X} \subset \mathbb{R}^D$  of hyperrectangular structure, i.e.  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$  with real  $\underline{x}_i \leq \bar{x}_i$  for all  $i = 1, \dots, D$ . We define the  $D$ -variate and transformed analogon of a Bernstein ellipse around the hyperrectangle  $\mathcal{X}$  with parameter vector  $\varrho \in (1, \infty)^D$  as

$$B(\mathcal{X}, \varrho) := B([\underline{x}_1, \bar{x}_1], \varrho_1) \times \dots \times B([\underline{x}_D, \bar{x}_D], \varrho_D) \tag{4.36}$$

with  $B([\underline{x}, \bar{x}], \varrho) := \tau_{[\underline{x}, \bar{x}]} \circ B([-1, 1], \varrho)$ , where for  $x \in \mathbb{C}$  we have the transform  $\tau_{[\underline{x}, \bar{x}]}(\Re(x)) := \bar{x} + \frac{\underline{x} - \bar{x}}{2}(1 - \Re(x))$  and  $\tau_{[\underline{x}, \bar{x}]}(\Im(x)) := \frac{\bar{x} - \underline{x}}{2}\Im(x)$ . We call  $B(\mathcal{X}, \varrho)$  *generalized Bernstein ellipse* if the sets  $B([-1, 1], \varrho_i)$  are Bernstein ellipses for  $i = 1, \dots, D$ .

#### 4 Chebyshev Polynomial Interpolation Method

**Theorem 4.2.2.** *Let  $\mathcal{X} \ni x \mapsto f(x)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}, \varrho)} |f(x)| \leq V$ . Then*

$$\max_{x \in \mathcal{X}} |f(x) - I_{\overline{N}}(f(\cdot))(x)| \leq 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}}.$$

The basic structure of the proof is the same as in (Sauter and Schwab, 2004, Proof of Lemma 7.3.3). To provide a complete, understandable proof, we first show the same steps as in (Sauter and Schwab, 2004, Proof of Lemma 7.3.3) and state explicitly at which point the proof changes.

*Proof.* In (Sauter and Schwab, 2004, Proof of Lemma 7.3.3) the proof is given for the following error bound:

$$\max_{p \in \mathcal{P}} |f - I_{\overline{N}}(f)| \leq \sqrt{D} 2^{\frac{D}{2}+1} V \varrho_{\min}^{-N} (1 - \varrho_{\min}^{-2})^{-\frac{D}{2}},$$

where  $N$  is the number of interpolation points in each of the  $D$  dimensions,  $\varrho_{\min} := \min_{i=1}^D \varrho_i$  and  $V$  the bound of  $f$  on  $B(\mathcal{P}, \varrho)$  with  $\mathcal{P} = [-1, 1]^D$ . Here, we extend (Sauter and Schwab, 2004, Proof of Lemma 7.3.3) by incorporating the different values of  $N_i$ ,  $i = 1, \dots, D$ , as well as expressing the error bound with the different  $\varrho_i$ ,  $i = 1, \dots, D$ .

In general we work with a parameter space  $\mathcal{P}$  of hyperrectangular structure,  $\mathcal{P} = [\underline{p}_1, \overline{p}_1] \times \dots \times [\underline{p}_D, \overline{p}_D]$ . With the introduced linear transformation, we have a transformation  $\tau_{\mathcal{P}} : [-1, 1]^D \rightarrow \mathcal{P}$  with

$$\tau_{\mathcal{P}}(p) = \left( \overline{p}_i + \frac{\underline{p}_i - \overline{p}_i}{2} (1 - p) \right)_{i=1}^D. \quad (4.37)$$

Let  $p \mapsto Price^p$  be a function on  $\mathcal{P}$ . We set  $\widehat{Price}^p = Price^p \circ \tau_{\mathcal{P}}(p)$ . Furthermore, let  $\widehat{I}_{\overline{N}}(\widehat{Price}^{(\cdot)})(p)$  be the Chebyshev interpolation of  $\widehat{Price}^p$  on  $[-1, 1]^D$ . Then it holds

$$I_{\overline{N}}(Price^{(\cdot)})(p) = \widehat{I}_{\overline{N}}(\widehat{Price}^{(\cdot)})(\cdot) \circ \tau_{\mathcal{P}}^{-1}(p).$$

Hence, it directly follows

$$Price^p - I_{\overline{N}}(Price^{(\cdot)})(p) = \left( \widehat{Price} - \widehat{I}_{\overline{N}}(\widehat{Price}^{(\cdot)})(\cdot) \right) \circ \tau_{\mathcal{P}}^{-1}(p).$$

#### 4 Chebyshev Polynomial Interpolation Method

Applying the error estimation from (Sauter and Schwab, 2004, Lemma 7.3.3) results

$$\begin{aligned} |Price - I_{\overline{N}}(Price^{(\cdot)})(\cdot)|_{C^0(\mathcal{P})} &= |Price - I_{\overline{N}}(Price^{(\cdot)})(\cdot)|_{C^0([-1,1]^D)} \\ &\leq \sqrt{D} 2^{\frac{D}{2}+1} \widehat{V} \varrho_{\min}^{-N} (1 - \varrho_{\min}^{-2})^{-\frac{D}{2}} \\ &= \sqrt{D} 2^{\frac{D}{2}+1} V \varrho_{\min}^{-N} (1 - \varrho_{\min}^{-2})^{-\frac{D}{2}}, \end{aligned}$$

where  $\widehat{V} = \sup_{p \in B([-1,1]^D, \varrho)} \widehat{Price}^p$ ,  $V = \sup_{p \in B(\mathcal{P}, \varrho)} Price^p$ . Summarizing, the transformation  $\tau_{\mathcal{P}} : [-1, 1]^D \rightarrow \mathcal{P}$  does not affect the error analysis, only by applying the transformation as described in (4.37),

$$B(\mathcal{P}, \varrho) := B([\underline{p}_1, \bar{p}_1], \varrho_1) \times \dots \times B([\underline{p}_D, \bar{p}_D], \varrho_D),$$

with  $B([\underline{p}, \bar{p}], \varrho) := \tau_{[\underline{p}, \bar{p}]} \circ B([-1, 1], \varrho)$ . Note that  $\varrho_i$  is not the radius of the ellipse  $B([\underline{p}_i, \bar{p}_i], \varrho_i)$  but of the normed ellipse  $B([-1, 1], \varrho_i)$ . Therefore, in the following it suffices to show the proof for  $\mathcal{P} = [-1, 1]^D$ .

As in (Sauter and Schwab, 2004, Proof of Lemma 7.3.3) we introduce the scalar product

$$\langle f, g \rangle_{\varrho} := \int_{B(\mathcal{P}, \varrho)} \frac{f(z) \overline{g(z)}}{\prod_{i=1}^D \sqrt{|1 - z_i^2|}} dz$$

and the Hilbert space

$$L^2(B(\mathcal{P}, \varrho)) := \{f : f \text{ is analytic in } B(\mathcal{P}, \varrho) \text{ and } \|f\|_{\varrho}^2 := \langle f, f \rangle_{\varrho} < \infty\}.$$

Following (Sauter and Schwab, 2004, Proof of Lemma 7.3.3), we define a complete orthonormal system for  $L^2(B(\mathcal{P}, \varrho))$  w.r.t. the scalar product  $\langle \cdot, \cdot \rangle_{\varrho}$  by the scaled Chebyshev polynomials

$$\tilde{T}_{\mu}(z) := c_{\mu} T_{\mu}(z) \text{ with } c_{\mu} := \left(\frac{2}{\pi}\right)^{\frac{D}{2}} \prod_{i=1}^D (\varrho_i^{2\mu_i} + \varrho_i^{-2\mu_i})^{-\frac{1}{2}}, \quad \text{for all } \mu \in \mathbb{N}_0^D.$$

Following Sauter and Schwab (2004), for any arbitrary bounded linear functional  $E$  on  $L^2(B(\mathcal{P}, \varrho))$  we have

$$|E(f)| \leq \|E\|_{\varrho} \|f\|_{\varrho}, \tag{4.38}$$

where  $\|E\|_{\varrho}$  denotes the operator norm. Due to the orthonormality of  $(\tilde{T}_{\mu})_{\mu \in \mathbb{N}_0^D}$  it follows that, see Sauter and Schwab (2004),

$$\|E\|_{\varrho} = \sup_{f \in L^2(B(\mathcal{P}, \varrho)) \setminus \{0\}} \frac{|E(f)|}{\|f\|_{\varrho}} = \sqrt{\sum_{\mu \in \mathbb{N}_0^D} |E(\tilde{T}_{\mu})|^2}.$$

#### 4 Chebyshev Polynomial Interpolation Method

In the following let  $E$  be the error of the Chebyshev polynomial interpolation at a fixed  $p \in \mathcal{P}$ ,

$$E(f) := f(p) - I_{\overline{N}}(f(\cdot))(p).$$

Starting with (4.38), we first focus on  $\|E\|_{\varrho}$ . In a first step, due to the orthonormal system, we get  $\|E\|_{\varrho}^2 = \sum_{\mu \in \mathbb{N}_0^D} |E(\tilde{T}_{\mu})|^2$ . For any  $\mu \in \mathbb{N}_0^D$  it holds that  $|E(\tilde{T}_{\mu})| = |\tilde{T}_{\mu}(x) - I_{\overline{N}}(\tilde{T}_{\mu}(\cdot))(x)| = |c_{\mu}T_{\mu}(x) - c_{\mu}I_{\overline{N}}(T_{\mu}(\cdot))(x)| = |c_{\mu}| \cdot |\mu(x) - I_{\overline{N}}(T_{\mu}(\cdot))(x)| = |c_{\mu}| \cdot |E(T_{\mu})|$ . This leads to,

$$\|E\|_{\varrho}^2 = \sum_{\mu \in \mathbb{N}_0^D} |E(\tilde{T}_{\mu})|^2 = \sum_{\mu \in \mathbb{N}_0^D} c_{\mu}^2 |E(T_{\mu})|^2.$$

From now on the proof differs compared to (Sauter and Schwab, 2004, Proof of Lemma 7.3.3), since we use the values of  $N_i$ ,  $i = 1, \dots, D$  and  $\varrho_i$ ,  $i = 1, \dots, D$ . Due to our choice of the Chebyshev points of the second kind instead of Chebyshev points of the first kind in the Chebyshev interpolation, we cannot apply (Sauter and Schwab, 2004, Corollary 7.3.1), but adjust this result in Lemma 4.2.3 to the Chebyshev points of the second kind. At this step, we analyze the error of interpolating Chebyshev polynomials  $T_{\mu}$  with  $\mu \in \mathbb{N}_0^D$ . Lemma 4.2.3 provides us with the results that if the degree of the Chebyshev polynomial  $T_{\mu}$  in each dimension  $i$  is smaller than the corresponding degree of the interpolation, i.e.  $\mu_i \leq N_i$  for  $i = 1, \dots, D$ , then the error is 0. Thus, we only have to analyze the summands for which at least for one dimension  $i$  the condition  $\mu_i \leq N_i$  does not hold. For these cases we then apply the error bound of Lemma 4.2.3.

$$\sum_{\mu \in \mathbb{N}_0^D} c_{\mu}^2 |E(T_{\mu})|^2 = \sum_{\mu \in \mathbb{N}_0^D, \exists i: \mu_i > N_i} c_{\mu}^2 |E(T_{\mu})|^2 \leq \sum_{\mu \in \mathbb{N}_0^D, \exists i: \mu_i > N_i} 4c_{\mu}^2.$$

Overall, using  $\left(\prod_{j=1}^D \varrho_j^{2\mu_j} + x\right)^{-1} \leq \left(\prod_{j=1}^D \varrho_j^{2\mu_j}\right)^{-1} = \prod_{j=1}^D \varrho_j^{-2\mu_j}$  for  $x > 0$ ,  $\mu_j \in \mathbb{N}_0$  and  $j = 1, \dots, D$  and this leads to

$$\begin{aligned} \|E\|_{\varrho}^2 &\leq 4 \sum_{\mu \in \mathbb{N}_0^D, \exists i: \mu_i > N_i} c_{\mu}^2 \leq 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \left( \sum_{\mu \in \mathbb{N}_0^D, \mu_i > N_i} \prod_{j=1}^D \varrho_j^{-2\mu_j} \right) \\ &\leq 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \varrho_i^{-2N_i} \left( \sum_{\mu \in \mathbb{N}_0^D, \mu_i > N_i} \varrho_i^{-2(\mu_i - N_i)} \prod_{j=1, j \neq i}^D \varrho_j^{-2\mu_j} \right) \\ &\leq 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \varrho_i^{-2N_i} \left( \sum_{\mu \in \mathbb{N}_0^D} \prod_{j=1}^D \varrho_j^{-2\mu_j} \right). \end{aligned}$$

From this point on we use the convergence of the geometric series since  $|\varrho_j^{-2}| < 1$ ,  $j =$

#### 4 Chebyshev Polynomial Interpolation Method

$1, \dots, D,$

$$\begin{aligned} \|E\|_{\varrho}^2 &\leq 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \varrho_i^{-2N_i} \left( \sum_{\mu_1=0}^{\infty} \dots \sum_{\mu_D=0}^{\infty} \prod_{j=1}^D \varrho_j^{-2\mu_j} \right) \\ &= 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}}. \end{aligned}$$

Recalling (4.38), we have to estimate  $\|f\|_{\varrho}$ ,

$$\|f\|_{\varrho}^2 = \int_{B(\mathcal{P}, \varrho)} \frac{f(z)\overline{f(z)}}{\prod_{i=1}^D \sqrt{|1 - z_i^2|}} dz \leq \left( \sup_{z \in B(\mathcal{P}, \varrho)} |f(z)| \right)^2 \|1\|_{\varrho}^2.$$

From  $\pi^{\frac{D}{2}} \tilde{T}_0 = 1$  it directly follows that  $\|1\|_{\varrho}^2 = \left(\pi^{\frac{D}{2}}\right)^2 \|\tilde{T}_0\|_{\varrho}^2 = \pi^D$  and thus

$$\|f\|_{\varrho}^2 \leq \pi^D \cdot V^2.$$

Combining the results leads to

$$\begin{aligned} |E(f)| &= |f(p) - I_{\overline{N}}(f(\cdot))(p)| \leq \left( \pi^D \cdot V^2 \cdot 4 \left(\frac{2}{\pi}\right)^D \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}} \\ &= 2^{\frac{D}{2}+1} V \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}}. \end{aligned}$$

□

The following lemma shows that the Chebyshev interpolation of a polynomial with a degree as most as high as the degree of the interpolating Chebyshev polynomial is exact and furthermore determines an upper bound for interpolating Chebyshev polynomials with a higher degree.

**Lemma 4.2.3.** *For  $x \in [-1, 1]^D$  it holds*

$$|T_{\mu}(x) - I_{\overline{N}}(T_{\mu}(\cdot))(x)| = 0 \quad \text{for all } \mu \in \mathbb{N}_0^D : \mu_i \leq N_i, i = 1, \dots, D, \quad (4.39)$$

$$|T_{\mu}(x) - I_{\overline{N}}(T_{\mu}(\cdot))(x)| \leq 2 \quad \text{for all } \mu \in \mathbb{N}_0^D : \exists i \in \{1, \dots, D\} : \mu_i > N_i. \quad (4.40)$$

*Proof.* Uniqueness properties of the Chebyshev interpolation directly imply (4.39). The proof of (4.40) is similar to (Sauter and Schwab, 2004, Proof of Hilfssatz 7.3.1). They use the zeros of the Chebyshev polynomial as interpolation points, whereas we use the extreme points and therefore, we use a different orthogonality property in this proof. We

#### 4 Chebyshev Polynomial Interpolation Method

first focus on the one-dimensional case. Recalling (4.30), the Chebyshev interpolation of  $T_\mu$ ,  $\mu > N$ , is given as

$$I_N(T_\mu)(x) = \sum_{j=0}^N c_j T_j(x) \quad \text{with} \quad c_j = \frac{2^{\mathbb{1}_{0 < j < N}}}{N} \sum_{k=0}^N {}''T_\mu(x_k) T_j(x_k), \quad j \leq N,$$

where  $x_k$  denotes the  $k$ -th extremum of  $T_N$ . Here, we can apply the following orthogonality (Rivlin, 1990, p.54),

$$\sum_{k=0}^N {}''T_\mu(x_k) T_j(x_k) = \begin{cases} 0, & \mu + j \neq 0 \pmod{2N} \text{ and } |\mu - j| \neq 0 \pmod{2N}, \\ N, & \mu + j = 0 \pmod{2N} \text{ and } |\mu - j| = 0 \pmod{2N}, \\ \frac{N}{2}, & \mu + j = 0 \pmod{2N} \text{ and } |\mu - j| \neq 0 \pmod{2N}, \\ \frac{N}{2}, & \mu + j \neq 0 \pmod{2N} \text{ and } |\mu - j| = 0 \pmod{2N}. \end{cases} \quad (4.41)$$

For  $j \leq N$  and  $\mu > N$  this yields the existence of  $\gamma \leq N$  such that

$$I_N(T_\mu) = T_\gamma. \quad (4.42)$$

(4.42) follows elementarily from the case that for any  $\mu > N$  only for one  $0 \leq j \leq N$  the orthogonality can lead to a coefficient  $c_j > 0$ .

Proving the claim, we distinguish several cases. In all of these cases, we assume that there exists  $0 \leq j \leq N$  such that  $\sum_{k=0}^N {}''T_\mu(x_k) T_j(x_k) \neq 0$ . We will then show that for all other  $0 \leq i \leq N, i \neq j$  it follows  $\sum_{k=0}^N {}''T_\mu(x_k) T_i(x_k) = 0$ .

First, assume there exists  $j$  such that  $\mu + j = 0 \pmod{2N}$  and  $\mu - j = 0 \pmod{2N}$ . Then it directly follows for all  $0 \leq i \leq N, i \neq j$  that  $\mu + i \neq 0 \pmod{2N}$  and  $\mu - i \neq 0 \pmod{2N}$ .

Second, assume there exists  $j$  such that  $\mu + j = 0 \pmod{2N}$  and  $\mu - j \neq 0 \pmod{2N}$ . Analogously, for all  $0 \leq i \leq N, i \neq j$  we have  $\mu + i \neq 0 \pmod{2N}$  and additionally from  $\mu + j = 0 \pmod{2N}$  it follows that  $\mu + j - 2N = 0 \pmod{2N}$  and, thus, for all  $0 \leq i \leq N, i \neq j$  we have  $\mu - i > \mu + j - 2N$  which is equivalent to  $\mu - i \neq 0 \pmod{2N}$ .

A similar argumentation holds for the third case  $\mu + j \neq 0 \pmod{2N}$  and  $|\mu - j| = 0 \pmod{2N}$ .

Therefore, (4.42) holds and it directly follows that  $|T_\mu - I_N(T_\mu)| \leq |T_\mu| + |I_N(T_\mu)| \leq 1 + 1 = 2$ . Thus (4.40) holds in the one-dimensional case. The extension to the  $D$ -dimensional case follows analogously by applying the triangle inequality  $|\prod_{i=1}^D T_{i,\mu_i} - \prod_{i=1}^D I_{N_i}(T_{i,\mu_i})| \leq |\prod_{i=1}^D T_{i,\mu_i}| + |\prod_{i=1}^D I_{N_i}(T_{i,\mu_i})| \leq \prod_{i=1}^D |T_{i,\mu_i}| + \prod_{i=1}^D |I_{N_i}(T_{i,\mu_i})|$  and inserting the one-dimensional result to each tensor component.  $\square$

**Corollary 4.2.4.** *Under the assumptions of Theorem 4.2.2 there exists a constant  $C > 0$*

#### 4 Chebyshev Polynomial Interpolation Method

such that

$$\max_{x \in \mathcal{X}} |f(x) - I_{\underline{N}}(f(\cdot))(x)| \leq C \underline{\varrho}^{-\underline{N}}, \quad (4.43)$$

where  $\underline{\varrho} = \min_{1 \leq i \leq D} \varrho_i$  and  $\underline{N} = \min_{1 \leq i \leq D} N_i$ .

**Remark 4.2.5.** In particular, under the assumptions required by Theorem 4.2.2 with  $N = \prod_{i=1}^D (N_i + 1)$  denoting the total number of nodes, Corollary 4.2.4 shows that the error decay is of (sub)exponential order  $O(\varrho^{-\sqrt[D]{N}})$  for some  $\varrho > 1$ .

So far, we have extended the error bound for the tensorized Chebyshev interpolation of Sauter and Schwab (2004) slightly in Theorem 4.2.2. This proof relies on a method for error estimation for analytic integrands from Davis (1975). The error bound is connected to the radius  $\varrho$  of a Bernstein ellipse and in the one-dimensional case Trefethen (2013) presents a different approach, which goes back to Bernstein (1912). In Börm (2010) error bounds are presented for the case when the derivatives of the function  $f$  are bounded. However, here we assume  $f$  to be analytic. We iteratively extend in the following the one-dimensional result shown in Trefethen (2013) to the multivariate by induction over the dimension. In each iteration step the interpolation in one additional variable is added consecutively. The resulting nested structure of the proof reaches a certain complexity and therefore requires more space than the proof in Sauter and Schwab (2004). Figure 4.4 illustrates this iterative interpolation idea in the proof of the upcoming Theorem 4.2.6.

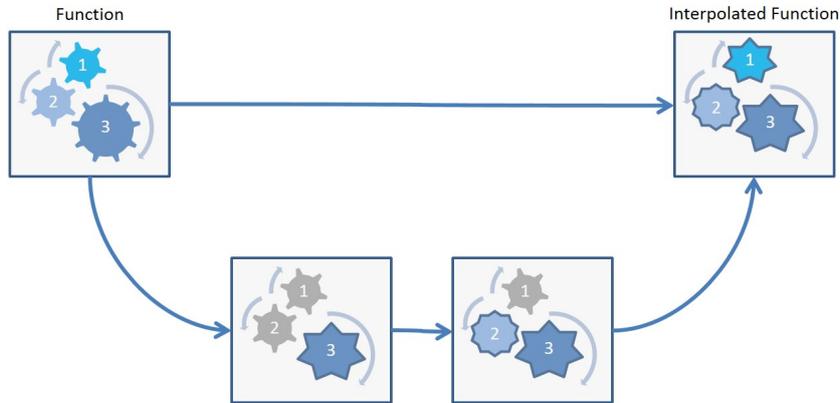


Figure 4.4: Schematic illustration of the iterative proof for Theorem 4.2.6. A function in 3 variables is iteratively interpolated in one dimension at each step. At the first step of the proof, the one-dimensional interpolation results can easily be applied using given properties of the function  $f$ . From the second step onwards however, even if at each step only a one-dimensional interpolation is applied, the according properties of a "new temporarily" function have to be verified.

#### 4 Chebyshev Polynomial Interpolation Method

**Theorem 4.2.6.** *Let  $f : \mathcal{X} \rightarrow \mathbb{R}$  have an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^D$  with  $\max_{x \in B(\mathcal{X}, \varrho)} |f(x)| \leq V < \infty$ . Then*

$$\begin{aligned} & \max_{x \in \mathcal{X}} |f(x) - I_{\bar{N}}(f)(x)| \\ & \leq \min_{\sigma \in S_D} \sum_{i=1}^D 4V \frac{\varrho_{\sigma(i)}^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_{\sigma(k)}^{-N_k}}{\varrho_{\sigma(k)} - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1}{\varrho_{\sigma(j)}})}, \end{aligned}$$

where  $S_D$  denotes the symmetric group on  $D$  elements.

*Proof.* We show the statement for an arbitrary  $\sigma \in S_D$  and for ease of notation we use  $\sigma(i) = i$  for  $i = 1, \dots, D$ . Obviously, we can iteratively interpolate in the parameter in such a way that the error bound is minimized by choosing the corresponding  $\sigma \in S_D$ .

We prove the assertion of the theorem via induction over the dimension  $D$  of the parameter domain. We assume that the function  $f$  is analytic in  $[-1, 1]^D$  and is analytically extendable to the open Bernstein ellipse  $B([-1, 1]^D, \varrho)$ . For  $D = 1$  and  $\mathcal{X} = [-1, 1]$  the proof of the assertion is presented in (Trefethen, 2013, Theorem 8.2). The generalization of the assertion to the case of a general parameter interval  $\mathcal{X} \subset \mathbb{R}$  is elementary and follows from a linear transformation as described in the proof of Theorem 4.2.2.

The key idea of the proof is to use the triangle inequality to estimate the interpolation error in  $D + 1$  components as sum of two parts. First, the interpolation error of the original function in component  $D + 1$ . Second, the error of interpolating a function which has already been interpolated in the first  $D$  dimensions in component  $D + 1$ . Hereby, in both cases the issue is basically reduced to a one-dimensional interpolation and the known theory from (Trefethen, 2013, Theorem 8.2) can be applied. The crucial step is to derive the bound of the function which has already been interpolated in the first  $D$  dimensions on the corresponding Bernstein ellipse.

Let us now assume that the assertion is proven for dimension  $1, \dots, D$ . Let  $\mathcal{X}^{D+1} := [\underline{x}_1, \bar{p}_1] \times \dots \times [\underline{x}_{D+1}, \bar{p}_{D+1}]$  and let  $f : \mathcal{X} \rightarrow \mathbb{R}$  have an analytic extension to the generalized Bernstein ellipse  $B(\mathcal{X}^{D+1}, \varrho^{D+1})$  for some parameter vector  $\varrho^{D+1} \in (1, \infty)^{D+1}$  and let  $\max_{x \in B(\mathcal{X}^{D+1}, \varrho^{D+1})} |f(x)| \leq V$ . To set up notation, we write  $x_1^D = (x_1, \dots, x_D)$  and define in the following the Chebyshev interpolation operators. For interpolation only in the  $i$ -th component with  $N$  Chebyshev points,

$$I_N^i(f)(x_1^{D+1}) := I_N(f(x_1, \dots, x_{i-1}, \cdot, x_{i+1}, \dots, x_{D+1}))(x_i).$$

Analogously, interpolation only in  $j$  components with  $N_{k_1}, \dots, N_{k_j}$  Chebyshev points is denoted by

$$I_{N_{k_1}, \dots, N_{k_j}}^{j_1, \dots, j_j}(f)(x_1^{D+1}) := I_{N_{k_j}}^{j_j} \circ \dots \circ I_{N_{k_1}}^{j_1}(f)(x_1^{D+1}),$$

#### 4 Chebyshev Polynomial Interpolation Method

and finally, the interpolation in all  $D + 1$  components with  $N_1, \dots, N_{D+1}$  Chebyshev points is

$$I_{N_1, \dots, N_{D+1}}(f)(x_1^{D+1}) := I_{N_{D+1}}^{D+1} \circ \dots \circ I_{N_1}^1(f)(x_1^{D+1}).$$

In the following the norm  $|\cdot|$  denotes the  $\infty$ -norm on  $[-1, 1]^{D+1}$ . We are interested in the interpolation error

$$\begin{aligned} & |f(x_1^{D+1}) - I_{N_1, \dots, N_{D+1}}(f)(x_1^{D+1})| \\ & \leq |f(x_1^{D+1}) - I_{N_{D+1}}^{D+1}(f)(x_1^{D+1})| + |I_{N_{D+1}}^{D+1}(f)(x_1^{D+1}) - I_{N_1, \dots, N_{D+1}}(f)(x_1^{D+1})|. \end{aligned}$$

We first show that the first part, as a one dimensional interpolation, is bounded by, (Trefethen, 2013, Theorem 8.2),

$$|f(x_1^{D+1}) - I_{N_{D+1}}^{D+1}(f)(x_1^{D+1})| \leq 4V \frac{\varrho_{D+1}^{-N_{D+1}}}{\varrho_{D+1} - 1}. \quad (4.44)$$

In order to derive (4.44), we have to show that the coefficients of the Chebyshev polynomial interpolation are bounded. Following Trefethen (2013), the on  $x_{D+1}$  depending coefficient  $a_{k_{D+1}}$  is defined as

$$a_{k_{D+1}} := \frac{2^{\mathbb{1}_{k_{D+1} > 0}}}{\pi} \int_{-1}^1 \frac{f(x_1^{D+1}) T_{k_{D+1}}(p_{D+1})}{\sqrt{1 - x_{D+1}^2}} dx_{D+1}.$$

By using the same transformation as in the proof of (Trefethen, 2013, Theorem 8.1), just adapted to the multidimensional setting, i.e.

$$\begin{aligned} x_i &= \frac{z_i + z_i^{-1}}{2}, \quad i = 1, \dots, D + 1, \\ F(z_1, \dots, z_{D+1}) &= F(z_1^{-1}, \dots, z_{D+1}^{-1}) = f(x_1, \dots, x_{D+1}), \end{aligned}$$

we achieve for the estimation of the coefficient  $a_{k_{D+1}}$ ,

$$|a_{k_{D+1}}| = \left| \frac{2^{-\mathbb{1}_{k_{D+1} = 0}}}{\pi i} \int_{|z_{D+1}| = \varrho_{D+1}} z_{D+1}^{-1 - k_{D+1}} F(z_1, \dots, z_{D+1}) dz_{D+1} \right|.$$

Here, we use that  $F$  is bounded by the same constant as  $f$ , which is given by assumption,  $|f(x_1^{D+1})|_{B([-1, 1]^{D+1}, \varrho)} \leq V$ . Therefore, analogously to (Trefethen, 2013, Theorem 8.1), this leads to

$$|a_{k_{D+1}}| \leq 2\varrho_{D+1}^{-k_{D+1}} V. \quad (4.45)$$

This estimate can be used to derive (4.44) by applying (Trefethen, 2013, Theorem 8.2).

#### 4 Chebyshev Polynomial Interpolation Method

For the second part we use

$$|I_{N_{D+1}}^{D+1}(f)(x_1^{D+1}) - I_{N_1, \dots, N_{D+1}}(f)(x_1^{D+1})| = |I_{N_{D+1}}^{D+1}(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))(x_1^{D+1})|.$$

At this point we again apply the triangle inequality and achieve

$$\begin{aligned} & |I_{N_{D+1}}^{D+1}(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))(x_1^{D+1})| \\ & \leq |I_{N_{D+1}}^{D+1}(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))(x_1^{D+1}) - (f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))| \quad (4.46) \\ & \quad + |(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))|. \end{aligned}$$

The term (4.46) is basically an interpolation in the  $D + 1$  component of the function  $(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))$ . An upper bound  $\mathcal{M}(D)$  for this function is given in Lemma 4.2.8. With this bound, we can estimate the interpolation error of interpolating  $(f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))$  in the component  $D+1$ ,

$$\begin{aligned} & |I_{N_{D+1}}^{D+1}(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))(x_1^{D+1}) - (f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))| \\ & \leq 4\mathcal{M}(D) \frac{\varrho_{D+1}^{-N_{D+1}}}{\varrho_{D+1} - 1} \end{aligned}$$

The term  $|(f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))|$  is the interpolation error in  $D$  dimensions, and we assume that this one is by our induction hypothesis bounded, depending on  $D$ , i.e.

$$|(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))| \leq B(D), \quad B(D) > 0. \quad (4.47)$$

Collecting all parts, we achieve for the error of our interpolation in  $D + 1$  components,

$$|I_{N_{D+1}}^{D+1}(f - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1}))(x_1^{D+1})| \leq 4V \frac{\varrho_{D+1}^{-N_{D+1}}}{\varrho_{D+1} - 1} + B(D) + 4\mathcal{M}(D) \frac{\varrho_{D+1}^{-N_{D+1}}}{\varrho_{D+1} - 1}.$$

Finally, if we start with  $D = 1$  and apply the presented procedure step-wise, we get via straightforward induction,

$$B(D) = \sum_{i=1}^D 4V \frac{\varrho_i^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4\mathcal{M}(k-1) \frac{\varrho_k^{-N_k}}{\varrho_k - 1}.$$

Naturally, we can further estimate the error by using  $\frac{s_i}{\varrho_i} < 1$  and resp.  $(1 - \frac{s_i}{\varrho_i}) < 1$  in the numerator,

$$B(D) \leq \sum_{i=1}^D 4V \frac{\varrho_i^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_k^{-N_k}}{\varrho_k - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{s_j}{\varrho_j})}.$$

#### 4 Chebyshev Polynomial Interpolation Method

Recalling the definition of  $s_i = 1 + \epsilon$  with  $\epsilon \in (0, \min_{j=1}^D \varrho_j - 1)$ , the definition holds for any  $\epsilon \in (0, \min_{j=1}^D \varrho_j - 1)$  and therefore also for  $\lim_{\epsilon \rightarrow 0}$

$$\begin{aligned} B(D) &\leq \lim_{\epsilon \rightarrow 0} \sum_{i=1}^D 4V \frac{\varrho_i^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_k^{-N_k}}{\varrho_k - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1+\epsilon}{\varrho_j})} \\ &= \sum_{i=1}^D 4V \frac{\varrho_i^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_k^{-N_k}}{\varrho_k - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1}{\varrho_j})}. \end{aligned}$$

□

In the following lemmata, we use the convention  $\frac{N}{0} = \infty$ ,  $N \in \mathbb{N}^+$  and the notation  $x_1^M = (x_1, \dots, x_M)$ . This notation is to keep the statements shorter and make them more readable.

**Lemma 4.2.7.** *Let  $\mathcal{X} \ni x_1^M \mapsto f(x_1^M)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^M$ . Then the Chebyshev polynomial interpolation  $I_N^1(f)(x_1^M)$  is given by,*

$$I_N^1(f)(x_1^M) = \sum_{k=0}^N a_k(x_2^M) T_k(x_1) + \sum_{k=N+1}^{\infty} a_k(x_2^M) T_{m(k,N)}(x_1), \quad (4.48)$$

where  $m(k, N) = |(k + N - 1) \pmod{2N} - (N - 1)|$  and  $a_k(x_2^M) = \frac{2}{\pi} \int_{-1}^1 f(x_1^M) \frac{T_k(x_1)}{\sqrt{1-x_1^2}} dx_1$

*Proof.* Following (Trefethen, 2013, Equation (4.9)), from aliasing properties of Chebyshev polynomials it results that

$$f(x_1^M) - I_N^1(f)(x_1^M) = \sum_{k=N+1}^{\infty} a_k(x_2^M) (T_k(x_1) - T_{m(k,N)}(x_1)).$$

By writing the Chebyshev series for  $f(x_1^M)$ , see Trefethen (2013), we get,

$$\sum_{k=0}^{\infty} a_k(x_2^M) T_k(x_1) - I_N^1(f)(x_1^M) = \sum_{k=N+1}^{\infty} a_k(x_2^M) (T_k(x_1) - T_{m(k,N)}(x_1)),$$

and rearranging terms yields (4.48). □

As shown previously, the error of interpolating a function  $f(x_1, \dots, x_{D+1})$  in one arbitrary component  $i$ , the value  $V$  bounding the function  $f$  on  $B([-1, 1], \varrho_i)$  is a crucial for the error bound. In the proof of Theorem 4.2.6, we interpolate the already in the first components  $x_1, \dots, x_D$  interpolated function,  $f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})$ , to be precise, in the component  $x_{D+1}$ . Hence, the bound of  $f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})$

#### 4 Chebyshev Polynomial Interpolation Method

on  $B([-1, 1], \varrho_{D+1})$  is needed. Note the convention  $x_1^M = (x_1, \dots, x_M)$  to simplify the notation.

**Lemma 4.2.8.** *Let  $\mathcal{X} \ni x_1^{D+1} \mapsto f(x_1^{D+1})$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^{D+1}$ . Then*

$$\begin{aligned} & \sup_{x_{D+1} \in B([-1, 1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})| \leq \mathcal{M}(D) \\ & := 2^D V \frac{\sum_{i=1}^D \left(\frac{s_i}{\varrho_i}\right)^{N_i+1} + \sum_{\sigma \in \{0, 1\}^D \setminus \{0\}^D} \prod_{\delta: \sigma_\delta=0} \left(1 - \left(\frac{s_\delta}{\varrho_\delta}\right)^{N_\delta+1}\right) \prod_{\delta: \sigma_\delta=1} \left(\frac{s_\delta}{\varrho_\delta}\right)^{N_\delta+1}}{\prod_{j=1}^D \left(1 - \frac{s_j}{\varrho_j}\right)} \end{aligned}$$

*Proof.* Starting with,

$$\sup_{x_{D+1} \in B([-1, 1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})|,$$

we express the interpolation of  $f$  in  $D$  components as in Lemma 4.2.9,

$$\sup_{x_{D+1} \in B([-1, 1], \varrho_{D+1})} \left| f(x_1^{D+1}) - \sum_{\sigma \in \{0, 1\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D, x_1^D) \right|.$$

Following Trefethen (2013) and as used in Lemma 4.2.7, we can express  $f$  in the following way,

$$f(x_1^{D+1}) = \sum_{\delta=1}^D \sum_{k_\delta=0}^{\infty} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D = 0, x_1^D),$$

leading to,

$$\begin{aligned} & \sup_{x_{D+1} \in B([-1, 1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})| \\ & = \sup_{x_{D+1} \in B([-1, 1], \varrho_{D+1})} \left| \sum_{\delta=1}^D \sum_{k_\delta=0}^{\infty} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D = 0, x_1^D) \right. \\ & \quad \left. - \sum_{\sigma \in \{0, 1\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D, x_1^D) \right|. \end{aligned}$$

In the next step, we use from the second summand the part  $\sigma = \{0\}^D$ , subtract it from

#### 4 Chebyshev Polynomial Interpolation Method

the subtrahend and use the triangle inequality.

$$\begin{aligned}
& \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})| \\
&= \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} \left| \sum_{i=1}^D \left( \sum_{k_i=N_i+1}^{\infty} \sum_{j=1, j \neq i}^D \sum_{k_j=0}^{\infty} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D = 0, x_1^D) \right) \right. \\
&\quad \left. - \sum_{\sigma \in \{0,1\}^D \setminus \{0\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D, x_1^D) \right| \\
&\leq \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} \left| \sum_{i=1}^D \left( \sum_{k_i=N_i+1}^{\infty} \sum_{j=1, j \neq i}^D \sum_{k_j=0}^{\infty} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D = 0, x_1^D) \right) \right| \\
&\quad + \left| \sum_{\sigma \in \{0,1\}^D \setminus \{0\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}) \tau(k_1^D, \sigma_1^D, x_1^D) \right|
\end{aligned}$$

To estimate the supremum, we first need estimations for  $|I(k_1^D, x_{D+1})|$  and  $|\tau(k_1^D, \sigma_1^D, x_1^D)|$ .

$$\begin{aligned}
|I(k_1^D, x_{D+1})| &= \left| \prod_{i=1}^D \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \int_{[-1,1]^D} f(x_1^{D+1}) \prod_{j=1}^D \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_1^D) \right| \\
&= \left| \prod_{i=2}^D \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \int_{[-1,1]^{D-1}} \frac{2^{\mathbb{1}_{k_1 > 0}}}{\pi} \int_{-1}^1 f \frac{T_{k_1}(x_1)}{\sqrt{1-x_1^2}} d(x_1) \prod_{j=2}^D \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_2^D) \right|.
\end{aligned}$$

Analogously to deriving the estimation (4.45), we can estimate the integral with respect to  $x_1$  as  $\frac{2^{\mathbb{1}_{k_1 > 0}}}{\pi} \int_{-1}^1 f \frac{T_{k_1}(x_1)}{\sqrt{1-x_1^2}} d(p_1) \leq 2V \varrho_1^{-k_1}$ . The remaining  $D-1$  dimensional integral can in a similar way be estimated as  $D-1$  one-dimensional integrals with  $V = 1$ . Altogether, this results in the following estimation for  $|I(k_1, \dots, k_D)|$ ,

$$|I(k_1^D, x_{D+1})| \leq 2^D V \prod_{i=1}^D \varrho_i^{-k_i}.$$

For  $|\tau(k_1^D, \sigma_1^D = 0, x_1^D)|$ , we make use of Bernstein's inequality, using that the norm of each Chebyshev polynomial is bounded by 1 on  $[-1, 1]$ . For each  $i = 1, \dots, D$  we choose a Bernstein ellipse with radius  $s_i$  such that  $1 < s_i < \varrho_i$ . Here, we define  $s_i = 1 + \epsilon$  and this yields for  $x : x_i \in B([-1, 1], s_i)$ ,  $i = 1, \dots, D$ ,

$$|\tau(k_1^D, \sigma_1^D, x_1^D)| = \prod_{\delta: \sigma_\delta=0} T_{k_\delta}(x_\delta) \prod_{\delta: \sigma_\delta=1} T_{m_\delta(k_\delta)}(x_\delta) \leq \prod_{\delta: \sigma_\delta=0} s_\delta^{k_\delta} \prod_{\delta: \sigma_\delta=1} s_\delta^{m_\delta(k_\delta)}.$$

#### 4 Chebyshev Polynomial Interpolation Method

By definition, it holds  $m_\delta(k_\delta) \leq k_\delta$ . This leads to

$$|\tau(k_1^D, \sigma_1^D = 0, x_1^D)| \leq \prod_{i=1}^D s_i^{k_i}.$$

Using both estimates leads to

$$\begin{aligned} & \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})| \\ & \leq \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} \left| \sum_{i=1}^D \left( \sum_{k_i=N_i+1}^{\infty} \sum_{j=1, j \neq i}^D \sum_{k_j=0}^{\infty} 2^D V \prod_{l=1}^D \left( \frac{s_l}{\varrho_l} \right)^{k_l} \right) \right| \\ & \quad + \left| \sum_{\sigma \in \{0,1\}^D \setminus \{0\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} 2^D V \prod_{l=1}^D \left( \frac{s_l}{\varrho_l} \right)^{k_l} \right|. \end{aligned}$$

Due to  $s_i < \varrho_i$  we can apply the convergence results for the geometric series. This leads to

$$\begin{aligned} & \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} |f(x_1^{D+1}) - I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^{D+1})| \\ & \leq \mathcal{M}(D) := \sup_{x_{D+1} \in B([-1,1], \varrho_{D+1})} \left| 2^D V \sum_{i=1}^D \frac{\left( \frac{s_i}{\varrho_i} \right)^{N_i+1}}{\prod_{j=1}^D \left( 1 - \frac{s_j}{\varrho_j} \right)} \right| \\ & \quad + \left| 2^D V \sum_{\sigma \in \{0,1\}^D \setminus \{0\}^D} \frac{\prod_{\delta: \sigma_\delta=0} \left( 1 - \left( \frac{s_\delta}{\varrho_\delta} \right)^{N_\delta+1} \right) \prod_{\delta: \sigma_\delta=1} \left( \frac{s_\delta}{\varrho_\delta} \right)^{N_\delta+1}}{\prod_{j=1}^D \left( 1 - \frac{s_j}{\varrho_j} \right)} \right| \\ & = 2^D V \frac{\sum_{i=1}^D \left( \frac{s_i}{\varrho_i} \right)^{N_i+1} + \sum_{\sigma \in \{0,1\}^D \setminus \{0\}^D} \prod_{\delta: \sigma_\delta=0} \left( 1 - \left( \frac{s_\delta}{\varrho_\delta} \right)^{N_\delta+1} \right) \prod_{\delta: \sigma_\delta=1} \left( \frac{s_\delta}{\varrho_\delta} \right)^{N_\delta+1}}{\prod_{j=1}^D \left( 1 - \frac{s_j}{\varrho_j} \right)}. \end{aligned}$$

□

**Lemma 4.2.9.** *Let  $\mathcal{X} \ni x_1^M \mapsto f(x_1^M)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^M$ . For  $D \leq M$  let*

$$\begin{aligned} I(k_1^D, x_{D+1}^M) &= \prod_{i=1}^D \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \int_{[-1,1]^D} f(x_1^M) \prod_{j=1}^D \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_1, \dots, x_D), \\ \tau(k_1^D, \sigma_1^D, x_1^D) &= \prod_{\delta: \sigma_\delta=0} T_{k_\delta}(x_\delta) \prod_{\delta: \sigma_\delta=1} T_{m_\delta}(x_\delta), \end{aligned}$$

#### 4 Chebyshev Polynomial Interpolation Method

then the interpolation of  $f(x_1^M)$  in  $D$  components is given by:

$$I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^M) = \sum_{\sigma \in \{0,1\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}^M) \tau(k_1^D, \sigma_1^D, x_1^D).$$

*Proof.* We proof this lemma via induction over the dimension  $D$ . For  $D = 1$  it follows from Lemma 4.2.7,

$$\begin{aligned} I_{N_1}^1(f)(x_1^M) &= \sum_{k_1=0}^{N_1} \frac{2^{\mathbb{1}_{k_1>0}}}{\pi} \int_{[-1,1]} f(x_1^M) \frac{T_{k_1}(x_1)}{\sqrt{1-x_1^2}} dx_1 T_{k_1}(x_1) \\ &\quad + \sum_{k_1=N_1+1}^{\infty} \frac{2^{\mathbb{1}_{k_1>0}}}{\pi} \int_{[-1,1]} f(x_1^M) \frac{T_{k_1}(x_1)}{\sqrt{1-x_1^2}} dx_1 T_{m_1}(x_1). \end{aligned}$$

Embedded in the introduced notation we get for  $D = 1$ ,

$$I_{N_1}^1(f)(x_1^M) = \sum_{\sigma \in \{0,1\}} \sum_{\delta=1}^1 \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^1, x_2^M) \tau(k_1^1, \sigma_1^1, x_1^1).$$

For the induction step from  $D-1$  to  $D$ , we assume the interpolation in  $D-1$  components is given by

$$I_{N_1, \dots, N_{D-1}}^{1, \dots, D-1}(f)(x_1^M) = \sum_{\sigma \in \{0,1\}^{D-1}} \sum_{\delta=1}^{D-1} \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^{D-1}, x_D^M) \tau(k_1^{D-1}, \sigma_1^{D-1}, x_1^{D-1}).$$

For the interpolation in  $D$  components we make use of

$$I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^M) = I_{N_D}^D \circ \dots \circ I_{N_1}^1(f)(x_1^M) = I_{N_D}^D \circ I_{N_1, \dots, N_{D-1}}^{1, \dots, D-1}(f)(x_1^M).$$

As for  $D = 1$  we apply (Trefethen, 2013, p.27) and this leads to

$$\begin{aligned} I_{N_1, \dots, N_D}(f)(x_1^D) &= \sum_{k_D=0}^{N_D} \frac{2^{\mathbb{1}_{k_D>0}}}{\pi} \int_{-1}^1 I_{N_1, \dots, N_{D-1}}^{1, \dots, D-1}(f)(x_1^M) \frac{T_{k_D}(x_D)}{\sqrt{1-x_D^2}} dx_D T_{k_D}(x_D) \\ &\quad + \sum_{k_D=N_D+1}^{\infty} \frac{2^{\mathbb{1}_{k_D>0}}}{\pi} \int_{-1}^1 I_{N_1, \dots, N_{D-1}}^{1, \dots, D-1}(f)(x_1^M) \frac{T_{k_D}(x_D)}{\sqrt{1-x_D^2}} dx_D T_{m_D}(x_D). \end{aligned}$$

By the induction hypothesis and the definitions of  $I(k_1^{D-1}, x_D^M)$  and

#### 4 Chebyshev Polynomial Interpolation Method

$\tau(k_1^{D-1}, \sigma_1^{D-1}, x_1^{D-1})$ , we achieve,

$$\begin{aligned}
& \int_{-1}^1 I_{N_1, \dots, N_{D-1}}^{1, \dots, D-1}(f)(x_1^M) \frac{T_{k_D}(x_D)}{\sqrt{1-x_D^2}} dx_D \\
&= \int_{[-1,1]} \sum_{\sigma \in \{0,1\}^{D-1}} \sum_{\delta=1}^{D-1} \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^{D-1}, x_D^M) \tau(k_1^{D-1}, \sigma_1^{D-1}, x_1^{D-1}) \frac{T_{k_D}(x_D)}{\sqrt{1-x_D^2}} dx_D \\
&= \int_{[-1,1]} \sum_{\sigma \in \{0,1\}^{D-1}} \sum_{\delta=1}^{D-1} \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} \prod_{i=1}^{D-1} \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \\
& \quad \int_{[-1,1]^{D-1}} f(x_1^M) \prod_{j=1}^{D-1} \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_1, \dots, x_{D-1}) \frac{T_{k_D}(x_D)}{\sqrt{1-x_D^2}} dx_D.
\end{aligned}$$

Rearranging terms yields,

$$\begin{aligned}
I_{N_1, \dots, N_D}(f)(x_1^M) &= \sum_{k_D=0}^{N_D} \sum_{\sigma \in \{0,1\}^{D-1}} \sum_{\delta=1}^{D-1} \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} \prod_{i=1}^D \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \int_{[-1,1]^D} f(x_1^M) \\
& \quad \prod_{j=1}^D \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_1^D) \prod_{\delta: \sigma_\delta=0} T_{k_\delta}(x_\delta) \prod_{\delta: \sigma_\delta=1} T_{m_\delta}(x_\delta) T_{k_D}(x_D) \\
& \quad + \sum_{k_D=N_D+1}^{\infty} \sum_{\sigma \in \{0,1\}^{D-1}} \sum_{\delta=1}^{D-1} \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} \prod_{i=1}^D \frac{2^{\mathbb{1}_{k_i > 0}}}{\pi} \int_{[-1,1]^D} f(p_1^M) \\
& \quad \prod_{j=1}^D \frac{T_{k_j}(x_j)}{\sqrt{1-x_j^2}} d(x_1^D) \prod_{\delta: \sigma_\delta=0} T_{k_\delta}(x_\delta) \prod_{\delta: \sigma_\delta=1} T_{m_\delta}(x_\delta) T_{m_D}(x_D).
\end{aligned}$$

This can be expressed as

$$I_{N_1, \dots, N_D}^{1, \dots, D}(f)(x_1^M) = \sum_{\sigma \in \{0,1\}^D} \sum_{\delta=1}^D \sum_{k_\delta=(N_\delta+1) \cdot \sigma_\delta}^{\frac{N_\delta}{1-\sigma_\delta}} I(k_1^D, x_{D+1}^M) \tau(k_1^D, \sigma_1^D, x_1^D). \quad (4.49)$$

□

Finally, in the following we present the new error bound as a combination of this result with the result from Theorems 4.2.2 and 4.2.6. We furthermore discuss examples for which the new error bound yields a significant improvement.

#### 4 Chebyshev Polynomial Interpolation Method

**Theorem 4.2.10.** *Let  $f : \mathcal{X} \rightarrow \mathbb{R}$  have an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho)$  with parameter vector  $\varrho \in (1, \infty)^D$  with  $\max_{x \in B(\mathcal{X}, \varrho)} |f(x)| \leq V < \infty$ . Then*

$$\max_{x \in \mathcal{X}} |f(x) - I_{\overline{N}}(f)(x)| \leq \min\{a(\varrho, N, D), b(\varrho, N, D)\},$$

where, denoting by  $S_D$  the symmetric group on  $D$  elements,

$$a(\varrho, N, D) = \min_{\sigma \in S_D} \sum_{i=1}^D 4V \frac{\varrho_{\sigma(i)}^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_{\sigma(k)}^{-N_k}}{\varrho_{\sigma(k)} - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1}{\varrho_{\sigma(j)}})},$$

$$b(\varrho, N, D) = 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}}.$$

*Proof.* The bound  $\max_{x \in \mathcal{X}} |f(x) - I_{\overline{N}}(f)(x)| \leq b(\varrho, N, D)$  follows from Theorem 4.2.2 as extension of Sauter and Schwab (2004). The second bound  $\max_{x \in \mathcal{X}} |f(x) - I_{\overline{N}}(f)(x)| \leq a(\varrho, N, D)$  follows from Theorem 4.2.6. Combining both results obviously yields the assertion of the theorem.  $\square$

The examples below show that  $\min\{a(\varrho, N, D), b(\varrho, N, D)\}$  improves both error bounds  $a(\varrho, N, D)$  and  $b(\varrho, N, D)$ . Noticing that both bounds are scaled with the factor  $V$ , we set  $V = 1$ , moreover, we choose  $D = 2$ .

**Example 4.2.11.** *For  $\varrho_1 = 2.3$  and  $\varrho_2 = 1.8$ , and  $N_1 = N_2 = 10$ , we have  $b(\varrho, N, D) = 0.0018$  and  $a(\varrho, N, D) = 0.0066$ . Therefore, in this example the error bound  $b(\varrho, N, D)$  is sharper.*

**Example 4.2.12.** *If we change slightly the setting from Example 4.2.11 to  $\varrho_1 = 2.3$  and  $\varrho_2 = 2.5$ , and  $N_1 = N_2 = 10$ , then the resulting error bounds are  $b(\varrho, N, D) = 0.0017$  and  $a(\varrho, N, D) = 0.0011$  and thus, the later is the sharper error bound.*

As shown in Examples 4.2.11 and 4.2.12, slight changes in the domain of analyticity and, thus, the radii of the Bernstein ellipses, may reverse the order of  $a(\varrho, N, D)$  and  $b(\varrho, N, D)$ . Figure 4.5 displays both error bounds  $a(\varrho, N, D)$  and  $b(\varrho, N, D)$  for varying  $\varrho$  with  $\varrho_1 = \varrho_2$ ,  $N_1 = N_2 = 10$ . We observe that both error bounds intersect at  $\varrho_1 = \varrho_2 \approx 2.800882$ . For smaller values of  $\varrho$ , the sharper error bound is  $b(\varrho, N, D)$ , whereas for higher values  $a(\varrho, N, D)$  is sharper. So far, the examples indicate that for a smaller radius of the Bernstein ellipse,  $b(\varrho, N, D)$  tends to be the better error bound and that for higher radii of the Bernstein ellipses or for strongly differing radii,  $a(\varrho, N, D)$  tends to be the sharper error bound. Our last example shows a situation where thanks to Theorem 4.2.10 less nodes are required to guarantee a pre-specified accuracy.

## 4 Chebyshev Polynomial Interpolation Method

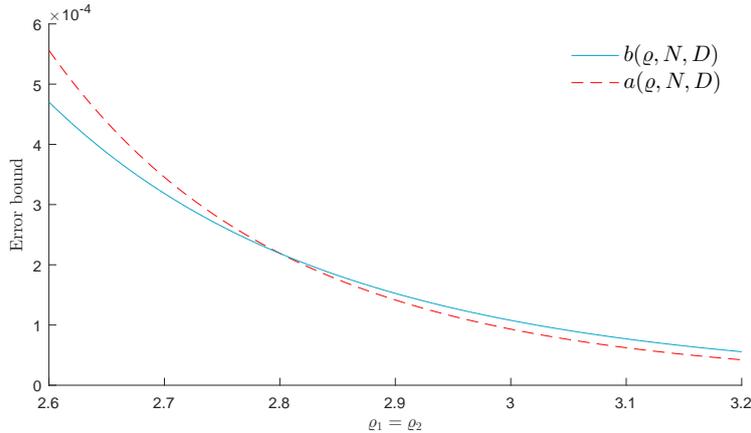


Figure 4.5: Comparison of the error bounds  $b(\varrho, N, D)$  (blue) and  $a(\varrho, N, D)$  (red, dashed) by setting  $\varrho_1 = \varrho_2$  and  $N_1 = N_2 = 10$ . At  $\varrho_1 = \varrho_2 \approx 2.800882$  both error bounds intersect.

**Example 4.2.13.** Let the radii of the Bernstein ellipse be  $\varrho_1 = 2.95$  and  $\varrho_2 = 9.8$ . Assuming  $V = 1$ , we are interested in achieving an accuracy of  $\varepsilon \leq 2 \cdot 10^{-4}$ . To achieve  $b(\varrho, N, D) \leq \varepsilon$ , we have to set  $N_1 = 11$  and  $N_2 = 5$ . For achieving  $a(\varrho, N, D) \leq \varepsilon$ , we have to set  $N_1 = 8$  and  $N_2 = 4$ . An application of error bound  $b(\varrho, N, D)$  would result in an interpolation with  $72 = (11 + 1) \cdot (5 + 1)$  nodes, whereas we only require  $45 = (8 + 1) \cdot (4 + 1)$  nodes when applying the error bound  $a(\varrho, N, D)$ .

Example 4.2.13 highlights the potential of using fewer nodal points to achieve a desired accuracy by comparing both error bounds. Especially when the evaluation of the interpolated function at the nodal points is challenging, this reduces the computational costs noticeably. This particularly arises for Chebyshev interpolation combined with Monte-Carlo simulation for high-dimensional parametric integration.

So far, we have provided an enhanced error bound for tensorized Chebyshev polynomial interpolation in Theorem 4.2.10 and have shown several examples. Example 4.2.13 highlights the effect of the improved error bound. Here, less interpolation nodes are required to guarantee a pre-specified accuracy. This significantly reduces the computational time, especially if the evaluation of function  $f$  at the nodal points is time-consuming.

### 4.2.1 Convergence Results Including the Derivatives

In the setting of Theorem 4.2.2 additionally the derivatives of  $f(x)$  are approximated by the according derivatives of the Chebyshev interpolation. The one-dimensional result is shown in Tadmor (1986) and a multivariate result is derived in Canuto and Quarteroni (1982) for functions in Sobolev spaces. These results allow us to obtain the Chebyshev approximation of derivatives with no additional cost. To state the according convergence

#### 4 Chebyshev Polynomial Interpolation Method

results we follow Canuto and Quarteroni (1982) and introduce the weighted Sobolev spaces for  $\sigma \in \mathbb{N}$  by

$$W_2^{\sigma, \omega}(\mathcal{X}) = \left\{ \phi \in L^2(\mathcal{X}) : \|\phi\|_{W_2^{\sigma, \omega}(\mathcal{X})} < \infty \right\}, \quad (4.50)$$

with norm

$$\|\phi\|_{W_2^{\sigma, \omega}(\mathcal{X})}^2 = \sum_{|\alpha| \leq \sigma} \int_{\mathcal{X}} |\partial^\alpha \phi(p)|^2 \omega(x) \, dx, \quad (4.51)$$

wherein  $\alpha = (\alpha_1, \dots, \alpha_D) \in \mathbb{N}_0^D$  is a multiindex and  $\partial^\alpha = \partial^{\alpha_1} \dots \partial^{\alpha_D}$  and the weight function  $\omega$  on  $\mathcal{X}$  given by

$$\omega(x) := \prod_{j=1}^D \omega(\tau_{[\underline{x}_j, \bar{x}_j]}^{-1}(x_j)), \quad \omega(\tau_{[\underline{x}_j, \bar{x}_j]}^{-1}(x_j)) := (1 - \tau_{[\underline{x}_j, \bar{x}_j]}^{-1}(x_j))^2)^{-\frac{1}{2}} \quad (4.52)$$

with  $\tau_{[\underline{x}_j, \bar{x}_j]}(x) = \bar{x}_j + \frac{x_j - \bar{x}}{2}(1 - x)$ . Then we apply the result of (Canuto and Quarteroni, 1982, Theorem 3.1) in the following corollary.

**Proposition 4.2.14.** *Let  $\mathcal{P} \ni p \mapsto Price^p \in W_2^{\sigma, \omega}(\mathcal{P})$  and set  $N_i = N$ ,  $i = 1, \dots, D$ , i.e. the same number of nodal points in each dimension. Then for any  $\frac{D}{2} < \sigma \in \mathbb{N}$  and any  $\sigma \geq \mu \in \mathbb{N}_0$  there exists a constant  $C > 0$  such that*

$$\|Price^{(\cdot)} - I_{\bar{N}}(Price^{(\cdot)})(\cdot)\|_{W_2^{\mu, \omega}(\mathcal{P})} \leq CN^{2\mu - \sigma} \|Price^{(\cdot)}\|_{W_2^{\sigma, \omega}(\mathcal{P})},$$

*Proof.* Before we apply (Canuto and Quarteroni, 1982, Theorem 3.1), which assumes  $\mathcal{X} = [-1, 1]^D$ , we investigate how the linear transformation  $\tau_{\mathcal{X}}$ , as introduced in the proof of Theorem 4.2.2, influences the derivatives. Let  $x \mapsto f(x)$  be a function on  $\mathcal{X}$ . We set  $\hat{h}(x) = f(x) \circ \tau_{\mathcal{X}}(x)$ . Furthermore, let  $\hat{I}_{\bar{N}}(\hat{h})(x)$  be the Chebyshev interpolation of  $\hat{h}(x)$  on  $[-1, 1]^D$ . Then, it directly follows

$$f(x) - I_{\bar{N}}(f(\cdot))(x) = \left( \hat{h}(\cdot) - \hat{I}_{\bar{N}}(\hat{h})(\cdot) \right) \circ \tau_{\mathcal{X}}^{-1}(x).$$

First, let us assume  $D = 1$ , i.e.  $\mathcal{X} = [\underline{x}, \bar{x}]$ , and let  $\alpha \in \mathbb{N}_0$ . For the partial derivatives it holds

$$\begin{aligned} \partial^\alpha f(x) - \partial^\alpha I_{\bar{N}}(f(\cdot))(x) &= \partial^\alpha (f(x) - I_{\bar{N}}(f(\cdot))(x)) \\ &= \partial^\alpha \left( \left( \hat{h}(\cdot) - \hat{I}_{\bar{N}}(\hat{h})(\cdot) \right) \circ \tau_{\mathcal{X}}^{-1}(x) \right) \\ &= \partial^{\alpha-1} \left( \partial^1 \hat{h}(\tau_{\mathcal{X}}^{-1}(x)) - \partial^1 \hat{I}_{\bar{N}}(\hat{h})(\tau_{\mathcal{X}}^{-1}(x)) \right) \\ &= \partial^{\alpha-1} \frac{2}{\bar{x} - \underline{x}} \left( \left[ \partial^1 \hat{h} \right] (\tau_{\mathcal{X}}^{-1}(x)) - \left[ \partial^1 \hat{I}_{\bar{N}}(\hat{h}) \right] (\tau_{\mathcal{X}}^{-1}(x)) \right). \end{aligned}$$

#### 4 Chebyshev Polynomial Interpolation Method

Repeating this step iteratively yields

$$\partial^\alpha f(x) - \partial^\alpha I_{\bar{N}}(f(\cdot))(x) = \frac{2^\alpha}{(\bar{x} - \underline{x})^\alpha} \left( \left[ \partial^\alpha \hat{h} \right] (\tau_{\mathcal{X}}^{-1}(x)) - \left[ \partial^\alpha \hat{I}_{\bar{N}}(\hat{h}(\cdot)) \right] (\tau_{\mathcal{X}}^{-1}(x)) \right).$$

Hence, the error on  $[-1, 1]$  is scaled with a factor  $\frac{2^\alpha}{(\bar{x} - \underline{x})^\alpha}$ . Extending this to the D-variate case with, this analogously results with  $\alpha = (\alpha_1, \dots, \alpha_D) \in \mathbb{N}_0^D$  is a multi-index and  $\partial^\alpha = \partial^{\alpha_1} \dots \partial^{\alpha_D}$

$$\begin{aligned} \partial^\alpha f(x) - \partial^\alpha I_{\bar{N}}(f(\cdot))(x) = \\ \prod_{i=1}^D \frac{2^{|\alpha_i|}}{(\bar{x}_i - \underline{x}_i)^{|\alpha_i|}} \left( \left[ \partial^\alpha \hat{h} \right] (\tau_{\mathcal{X}}^{-1}(x)) - \left[ \partial^\alpha \hat{I}_{\bar{N}}(\hat{h}(\cdot)) \right] (\tau_{\mathcal{X}}^{-1}(x)) \right). \end{aligned}$$

From Theorem 3.1 in Canuto and Quarteroni (1982) the assertion follows directly for  $\hat{h}(\cdot)$  on  $\mathcal{X} = [-1, 1]^D$ , i.e. for any  $\frac{D}{2} < \sigma \in \mathbb{N}$  and any  $\sigma \geq \mu \in \mathbb{N}_0$  there exists a constant  $\tilde{C} > 0$  such that

$$\|\hat{h}(\cdot) - \hat{I}_{\bar{N}}(\hat{h})(\cdot)\|_{W_2^{\mu, \omega}(\mathcal{X})} \leq \tilde{C} N^{2\mu - \sigma} \|\hat{h}(\cdot)\|_{W_2^{\sigma, \omega}(\mathcal{X})}, \quad (4.53)$$

For arbitrary  $\mathcal{X}$  the constant from 4.53 has to be multiplied with the according factor resulting from the linear transformation  $\tau_{\mathcal{X}}$ .  $\square$

The result in Corollary 4.2.14 is given in terms of weighted Sobolev norms. In the following remark, we connect the approximation error in the weighted Sobolev norm to the  $C^l(\mathcal{X})$  norm, where  $C^l(\mathcal{X})$  is the Banach space of all functions  $u$  in  $\mathcal{X}$  such that  $u$  and  $\partial^\alpha u$  with  $|\alpha| \leq l$  are uniformly continuous in  $\mathcal{X}$  and the norm

$$\|u\|_{C^l(\mathcal{X})} = \max_{|\alpha| \leq l} \max_{x \in \mathcal{X}} |\partial^\alpha u(x)|$$

is finite.

**Proposition 4.2.15.** *Let  $\mathcal{P} \ni p \mapsto Price^p \in W_2^{\sigma, \omega}(\mathcal{P})$  and set  $N_i = N$ ,  $i = 1, \dots, D$ , i.e. the same number of nodal points in each dimension. Then for any  $\frac{D}{2} < \sigma \in \mathbb{N}$  and any  $\sigma \geq \mu \in \mathbb{N}_0$  and  $l \in \mathbb{N}_0$  with  $\mu - l > \frac{D}{2}$ , there exists a constant  $\bar{C}(\sigma) > 0$  depending on  $\sigma$ , such that*

$$\|Price^{(\cdot)} - I_{\bar{N}}(Price^{(\cdot)})(\cdot)\|_{C^l(\mathcal{P})} \leq \bar{C}(\sigma) N^{2\mu - \sigma} \max_{|\alpha| \leq \sigma} \sup_{p \in \mathcal{P}} |\partial^\alpha Price^p|.$$

*Proof.* In the setting of Proposition 4.2.14, we start with the estimation of the approximation error in the weighted Sobolev norms,

$$\|f(\cdot) - I_{\bar{N}}(f(\cdot))(\cdot)\|_{W_2^{\mu, \omega}(\mathcal{X})} \leq C N^{2\mu - \sigma} \|f(\cdot)\|_{W_2^{\sigma, \omega}(\mathcal{X})}. \quad (4.54)$$

#### 4 Chebyshev Polynomial Interpolation Method

On  $\mathcal{X}$  it holds that  $w(x) \geq 1$  and, thus, we can deduce for the Sobolev norm with a constant weight of 1,

$$\|f(\cdot) - I_{\bar{N}}(f(\cdot))(\cdot)\|_{W_2^{\mu,\omega}(\mathcal{X})} \geq \|f(\cdot) - I_{\bar{N}}(f(\cdot))(\cdot)\|_{W_2^{\mu,1}(\mathcal{X})}.$$

With  $W_2^\mu(\mathcal{X})$  the usual Sobolev space,

$$W_2^\mu(\mathcal{X}) = \left\{ \phi \in L^2(\mathcal{X}) : \|\phi\|_{W_2^\mu(\mathcal{X})} < \infty \right\}, \quad \|\phi\|_{W_2^\mu(\mathcal{X})}^2 = \sum_{|\alpha| \leq \mu} \int_{\mathcal{X}} |\partial^\alpha \phi(x)|^2 dx,$$

Corollary 6.2 from Wloka (1987) directly yields that for any  $l$  with  $\mu - l > \frac{D}{2}$  there exists a constant  $\tilde{C}$  such that

$$\|f(\cdot) - I_{\bar{N}}(f(\cdot))(\cdot)\|_{C^l(\mathcal{X})} \leq \tilde{C} \|f(\cdot) - I_{\bar{N}}((\cdot))(\cdot)\|_{W_2^\mu(\mathcal{X})}. \quad (4.55)$$

In formula (4.55) we have derived a lower bound for the left hand side of expression (4.54). Next, we will find an upper bound for the right hand side of (4.54). From the definition of the weighted Sobolev norm, see (4.51), it follows

$$\begin{aligned} \|f(\cdot)\|_{W_2^{\sigma,\omega}(\mathcal{X})} &= \sqrt{\sum_{|\alpha| \leq \sigma} \int_{\mathcal{X}} |\partial^\alpha f(x)|^2 \omega(x) dx} \\ &\leq \sqrt{\sum_{|\alpha| \leq \sigma} \sup_{x \in \mathcal{X}} |\partial^\alpha f(x)|^2 \int_{\mathcal{X}} \omega(x) dx}. \end{aligned}$$

Here, we apply  $\int_{\mathcal{X}} \omega(x) dx = \pi^D$  and that there exists a constant  $\alpha^2(\sigma)$  depending on  $\sigma$  such that

$$\begin{aligned} \|f(x)\|_{W_2^{\sigma,\omega}(\mathcal{X})} &\leq \sqrt{\alpha^2(\sigma) \max_{|\alpha| \leq \sigma} \sup_{x \in \mathcal{X}} |\partial^\alpha f(x)|^2 \pi^D} \\ &= \alpha(\sigma) \max_{|\alpha| \leq \sigma} \sup_{x \in \mathcal{X}} |\partial^\alpha f(x)| \pi^{\frac{D}{2}}. \end{aligned} \quad (4.56)$$

Finally, using (4.55) and (4.56) in (4.54) yields an estimate of the approximation error in the  $C^l(\mathcal{X})$  norm,

$$\frac{1}{\tilde{C}} \|f(\cdot) - I_{\bar{N}}(f(\cdot))(\cdot)\|_{C^l(\mathcal{X})} \leq CN^{2\mu-\sigma} \alpha(\sigma) \max_{|\alpha| \leq \sigma} \sup_{x \in \mathcal{X}} |\partial^\alpha f(x)| \pi^{\frac{D}{2}}.$$

Collecting all constants in  $\bar{C}(\sigma)$ , the assertion follows directly.  $\square$

### 4.3 Chebyshev Interpolation Method for Parametric Option Pricing

Once we have studied the convergence properties of Chebyshev polynomial interpolation, we can now apply the Chebyshev interpolation to parametric option pricing. For parameters  $p \in \mathbb{R}^D$ , where  $D \in \mathbb{N}$  denotes the dimensionality of the parameter space, the price  $Price^p$  is approximated by tensorized Chebyshev polynomials  $T_j$  with pre-computed coefficients  $c_j$ ,  $j \in J$ , as follows,

$$Price^p \approx \sum_{j \in J} c_j T_j(p).$$

#### 4.3.1 Exponential Convergence of Chebyshev Interpolation for POP

In this section we embed the multivariate Chebyshev interpolation into the option pricing framework. We provide sufficient conditions under which option prices depend analytically on the parameters.

Analytic properties of option prices can be conveniently studied in terms of Fourier transforms. First, Fourier representations of option prices are explicitly available for a large class of both option types and asset models. Second, Fourier transformation unveils the analytic properties of both the payoff structure and the distribution of the underlying stochastic quantity in a beautiful way. By contrast, if option prices are represented as expectations, their analyticity in the parameters is hidden. For example the function  $K \mapsto (S_T - K)^+$  is not even differentiable, whereas the Fourier transform of the dampened call payoff function evidently is analytic in the strike, compare (Gaß et al., 2016, Table 1).

#### Conditions for Exponential Convergence

Let us first introduce a general option pricing framework. We consider option prices of the form

$$Price^{p=(p^1, p^2)} = E(f^{p^1}(X^{p^2})) \quad (4.57)$$

where  $f^{p^1}$  is a parametrized family of measurable payoff functions  $f^{p^1} : \mathbb{R}^d \rightarrow \mathbb{R}_+$  with payoff parameters  $p^1 \in \mathcal{P}^1$  and  $X^{p^2}$  is a family of  $\mathbb{R}^d$ -valued random variables with model parameters  $p^2 \in \mathcal{P}^2$ . The parameter set

$$p = (p^1, p^2) \in \mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \subset \mathbb{R}^D \quad (4.58)$$

#### 4 Chebyshev Polynomial Interpolation Method

is again of hyperrectangular structure, i.e.  $\mathcal{P}^1 = [\underline{p}_1, \bar{p}_1] \times \dots \times [\underline{p}_m, \bar{p}_m]$  and  $\mathcal{P}^2 = [\underline{p}_{m+1}, \bar{p}_{m+1}] \times \dots \times [\underline{p}_D, \bar{p}_D]$  for some  $1 \leq m \leq D$  and real  $\underline{p}_i \leq \bar{p}_i$  for all  $i = 1, \dots, D$ .

Typically we are given a parametrized  $\mathbb{R}^d$ -valued driving stochastic process  $H^{p'}$  with  $S^{p'}$  being the vector of asset price processes modeled as an exponential of  $H^{p'}$ , i.e.

$$S_t^{p',i} = S_0^{p',i} \exp(H_t^{p',i}), \quad 0 \leq t \leq T, \quad 1 \leq i \leq d, \quad (4.59)$$

and  $X^{p^2}$  is an  $\mathcal{F}_T$ -measurable  $\mathbb{R}^d$ -valued random variable, possibly depending on the history of the  $d$  driving processes, i.e.  $p^2 = (T, p')$  and

$$X^{p^2} := \Psi(H_t^{p'}, 0 \leq t \leq T),$$

where  $\Psi$  is an  $\mathbb{R}^d$ -valued measurable functional.

We now focus on the case that the price (4.57) is given in terms of Fourier transforms. This enables us to provide sufficient conditions under which the parametrized prices have an analytic extension to an appropriate generalized Bernstein ellipse. For most relevant options, the payoff profile  $f^{p^1}$  is not integrable and its Fourier transform over the real axis is not well defined. Instead, there exists an exponential dampening factor  $\eta \in \mathbb{R}^d$  such that  $e^{\langle \eta, \cdot \rangle} f^{p^1} \in L^1(\mathbb{R}^d)$ . We therefore introduce exponential weights in our set of conditions and denote the Fourier transform of  $g \in L^1(\mathbb{R}^d)$  by

$$\hat{g}(z) := \int_{\mathbb{R}^d} e^{i\langle z, x \rangle} g(x) dx$$

and we denote the Fourier transform of  $e^{\langle \eta, \cdot \rangle} f \in L^1(\mathbb{R}^d)$  by  $\hat{f}(\cdot - i\eta)$ . The exponential weight of the payoff will be compensated by exponentially weighting the distribution of  $X^{p^2}$  and that weight will reappear in the argument of  $\varphi^{p^2}$ , the characteristic function of  $X^{p^2}$ .

**Conditions 4.3.1.** Let parameter set  $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \subset \mathbb{R}^D$  of hyperrectangular structure as in (4.58). Let  $\varrho \in (1, \infty)^D$  and denote  $\varrho^1 := (\varrho_1, \dots, \varrho_m)$  and  $\varrho^2 := (\varrho_{m+1}, \dots, \varrho_D)$  and let weight  $\eta \in \mathbb{R}^d$ .

- (A1) For every  $p^1 \in \mathcal{P}^1$  the mapping  $x \mapsto e^{\langle \eta, x \rangle} f^{p^1}(x)$  is in  $L^1(\mathbb{R}^d)$ .
- (A2) For every  $z \in \mathbb{R}^d$  the mapping  $p^1 \mapsto \widehat{f^{p^1}}(z - i\eta)$  is analytic in the generalized Bernstein ellipse  $B(\mathcal{P}^1, \varrho^1)$  and there are constants  $c_1, c_2 > 0$  such that  $\sup_{p^1 \in B(\mathcal{P}^1, \varrho^1)} |\widehat{f^{p^1}}(-z - i\eta)| \leq c_1 e^{c_2 |z|}$  for all  $z \in \mathbb{R}^d$ .
- (A3) For every  $p^2 \in \mathcal{P}^2$  the exponential moment condition  $E(e^{-\langle \eta, X^{p^2} \rangle}) < \infty$  holds.
- (A4) For every  $z \in \mathbb{R}^d$  the mapping  $p^2 \mapsto \varphi^{p^2}(z + i\eta)$  is analytic in the generalized Bernstein ellipse  $B(\mathcal{P}^2, \varrho^2)$  and there are constants  $\alpha \in (1, 2]$  and  $c_1, c_2 > 0$  such that  $\sup_{p^2 \in B(\mathcal{P}^2, \varrho^2)} |\varphi^{p^2}(z + i\eta)| \leq c_1 e^{-c_2 |z|^\alpha}$  for all  $z \in \mathbb{R}^d$ .

#### 4 Chebyshev Polynomial Interpolation Method

**Theorem 4.3.2.** *Let  $\varrho \in (1, \infty)^D$  and weight  $\eta \in \mathbb{R}^d$ . Under conditions (A1)–(A4),  $\mathcal{P} \ni p \mapsto Price^p$  has an analytic extension to the generalized Bernstein ellipse  $B(\mathcal{P}, \varrho)$  and  $\sup_{p \in B(\mathcal{P}, \varrho)} |Price^p| \leq V$ , and thus,*

$$\max_{p \in \mathcal{P}} |Price^p - I_{\overline{N}}(Price^{(\cdot)})(p)| \leq \min\{a(\varrho, N, D), b(\varrho, N, D)\},$$

where, denoting by  $S_D$  the symmetric group on  $D$  elements,

$$a(\varrho, N, D) = \min_{\sigma \in S_D} \sum_{i=1}^D 4V \frac{\varrho_{\sigma(i)}^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_{\sigma(k)}^{-N_k}}{\varrho_{\sigma(k)} - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1}{\varrho_{\sigma(j)}})},$$

$$b(\varrho, N, D) = 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}}.$$

*Proof.* Thanks to assumptions (A2) and (A4) the mapping  $z \mapsto \widehat{f}^{p^1}(-z - i\eta)\varphi^{p^2}(z + i\eta)$  belongs to  $L^1(\mathbb{R}^d)$  for every  $p = (p^1, p^2) \in \mathcal{P}$ . Together with conditions (A1) and (A3), we therefore can apply (Eberlein et al., 2010, Theorem 3.2). This gives the following Fourier representation of the option prices,

$$Price^p = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d + i\eta} \widehat{f}^{p^1}(-z)\varphi^{p^2}(z) dz.$$

Due to assumptions (A2) and (A4) the mapping

$$p = (p^1, p^2) \mapsto \widehat{f}^{p^1}(-z)\varphi^{p^2}(z)$$

has an analytic extension to  $B(\mathcal{P}, \varrho)$ .

Let  $\gamma$  be a contour of a compact triangle in the interior of  $B([p_i, \bar{p}_i], \varrho_i)$  for arbitrary  $i = 1, \dots, D$ . Then thanks to assumption (A2) and (A4) we may apply Fubini's theorem and obtain

$$\begin{aligned} \int_{\gamma} Price^{(p_1, \dots, p_D)}(z) dp_i &= \frac{1}{(2\pi)^d} \int_{\gamma} \int_{\mathbb{R}^d + i\eta} \widehat{f}^{p^1}(-z)\varphi^{p^2}(z) dz dp_i \\ &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d + i\eta} \int_{\gamma} \widehat{f}^{p^1}(-z)\varphi^{p^2}(z) dp_i dz = 0. \end{aligned}$$

Moreover, thanks to assumptions (A2) and (A4), dominated convergence shows continuity of  $p \mapsto Price^p$  in  $B(\mathcal{P}, \varrho)$  which yields the analyticity of  $p \mapsto Price^p$  in  $B(\mathcal{P}, \varrho)$  thanks to a version of Morera's theorem provided in (Jänich, 2004, Satz 8).  $\square$

Similar to Corollary 4.2.14 in the setting of Theorem 4.3.2 additionally the according derivatives are approximated as well by the Chebyshev interpolation. A very interesting

#### 4 Chebyshev Polynomial Interpolation Method

application of this result in finance is the computation of sensitivities like delta or vega of an option price for risk assessment purposes. Theorem 4.3.2 together with Corollary 4.2.14 yield the following corollary.

**Corollary 4.3.3.** *Under the assumptions of Theorem 4.3.2, for all  $l \in \mathbb{N}$ ,  $\mu$  and  $\sigma$  with  $\sigma > \frac{D}{2}$ ,  $0 \leq \mu \leq \sigma$  and  $\mu - l > \frac{D}{2}$  there exist a constant  $C$ , such that*

$$\|Price^p - I_{\bar{N}}(Price^{(\cdot)})(p)\|_{C^l(\mathcal{P})} \leq C \bar{N}^{2\mu - \sigma} \|Price^p\|_{W_2^{\sigma}(\mathcal{P})},$$

where the spaces and norms are defined in Section 4.2.1.

In Gaß et al. (2016) it is shown that Conditions (A1)–(A4) are satisfied for a large class of payoff functions and asset models. Here, we focus on basket options in affine models.

Let  $X^{\pi'}$  be a parametric family of affine processes with state space  $\mathfrak{D} \subset \mathbb{R}^d$  for  $\pi' \in \Pi'$  such that for every  $\pi' \in \Pi'$  there exists a complex-valued function  $\nu^{\pi'}$  and a  $\mathbb{C}^d$ -valued function  $\phi^{\pi'}$  such that

$$\varphi^{p^2=(t,x,\pi')}(z) = E\left(e^{i\langle z, X_t^{\pi'} \rangle} \mid X_0^{\pi'} = x\right) = e^{\nu^{\pi'}(t,iz) + \langle \phi^{\pi'}(t,iz), x \rangle}, \quad (4.60)$$

for every  $t \geq 0$ ,  $z \in \mathbb{R}^d$  and  $x \in \mathfrak{D}$ . Under mild regularity conditions, the functions  $\nu^{\pi'}$  and  $\phi^{\pi'}$  are determined as solutions to generalized Riccati equations. We refer to Duffie et al. (2003) for a detailed exposition. The rich class of affine processes comprises the class of Lévy processes, for which  $\nu^{\pi'}(t,iz) = t\psi^{\pi'}(z)$  with  $\psi^{\pi'}$  given as some exponent in the Lévy-Khintchine formula and  $\phi^{\pi'}(t,iz) \equiv 0$ . Moreover, many popular stochastic volatility models such as the Heston model as well as stochastic volatility models with jumps, e.g. the model of Barndorff-Nielsen and Shephard (2001) and time-changed Lévy models, see Carr et al. (2003) and Kallsen (2006), are driven by affine processes.

Consider option prices of the form

$$Price^{(K,T,x,\pi')} = E(f^K(X_T^{\pi'}) \mid X_0^{\pi'} = x) \quad (4.61)$$

where  $f^K$  is a parametrized family of measurable payoff functions  $f^K : \mathbb{R}^d \rightarrow \mathbb{R}_+$  for  $K \in \mathcal{P}^1$ .

**Corollary 4.3.4.** *Under the conditions (A1)–(A3) for weight  $\eta \in \mathbb{R}^d$ ,  $\varrho \in (1, \infty)^D$  and  $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \subset \mathbb{R}^D$  of hyperrectangular structure assume*

(i) *for every parameter  $p^2 = (t, x, \pi') \in \mathcal{P}^2 \subset \mathbb{R}^{D-m}$  that the validity of the affine property (4.60) extends to  $z = \mathbb{R} + i\eta$ , i.e. for every  $z \in \mathbb{R} + i\eta$ ,*

$$\varphi^{p^2=(t,x,\pi')}(z) = E\left(e^{i\langle z, X_t^{\pi'} \rangle} \mid X_0^{\pi'} = x\right) = e^{\nu^{\pi'}(t,iz) + \langle \phi^{\pi'}(t,iz), x \rangle},$$

#### 4 Chebyshev Polynomial Interpolation Method

- (ii) for every  $z \in \mathbb{R}^d$  that the mappings  $(t, \pi') \mapsto \nu^{\pi'}(t, iz - \eta)$  and  $(t, \pi') \mapsto \phi^{\pi'}(t, iz - \eta)$  have an analytic extension to the Bernstein ellipse  $B(\Pi', \varrho')$  for some parameter  $\varrho' \in (1, \infty)^{D-m-1}$ ,
- (iii) there exist  $\alpha \in (1, 2]$  and constants  $C_1, C_2 > 0$  such that uniformly in the parameters  $p^2 = (t, x, \pi') \in B(\mathcal{P}^2, \tilde{\varrho}^2)$  for a generalized Bernstein ellipse with  $\tilde{\varrho}^2 \in (1, \infty)^{D-m}$
- $$\Re(\nu^{\pi'}(t, iz - \eta) + \langle \phi^{\pi'}(t, iz - \eta), x \rangle) \leq C_1 - C_2 |z|^\alpha \quad \text{for all } z \in \mathbb{R}.$$

Then there exist constants  $C > 0$ ,  $\underline{\varrho} > 1$  such that

$$\max_{p \in \mathcal{P}^1 \times \mathcal{P}^2} |Price^p - I_{\bar{N}}(Price^{(\cdot)})(p)| \leq C \underline{\varrho}^{-\bar{N}}.$$

*Proof.* Thanks to Theorem 4.3.2 and Corollary 4.2.4 and in view of the assumed validity of Conditions (A1)–(A3), it suffices to verify Condition (A4). While assumptions (i) and (ii) together yield the analyticity condition in (A4), part (iii) provides the upper bound in (A4).  $\square$

## 4.4 Numerical Experiments for Parametric Option Pricing

We apply the Chebyshev interpolation method to parametric option pricing considering a variety of option types in different well known option pricing models. Moreover, we conduct both an error analysis as well as a convergence study. The first focuses on the accuracy that can be achieved with a reasonable number of Chebyshev interpolation points. The latter confirms the theoretical order of convergence derived in Section 4.3.1, when the number of Chebyshev points increases. Finally, we study the gain in efficiency for selected multivariate options.

We measure the numerical accuracy of the Chebyshev method by comparing derived prices with prices coming from a reference method. We employ the reference method not only for computing reference prices but also for computing prices at Chebyshev nodes  $Price^{p^{(k_1, \dots, k_D)}}$  with  $(k_1, \dots, k_D) \in J$  during the precomputation phase of the Chebyshev coefficients  $c_j$ ,  $j \in J$ , in (4.33). Thereby, a comparability between Chebyshev prices and reference prices is maintained.

We implemented the Chebyshev method for applications with two parameters. To that extent we pick two free parameters  $p_{i_1}, p_{i_2}$  out of (4.58),  $1 \leq i_1 < i_2 \leq D$ , in each model setup and fix all other parameters at reasonable constant values. We then evaluate option prices for different products on a discrete parameter grid  $\bar{\mathcal{P}} \subseteq [\underline{p}_{i_1}, \bar{p}_{i_1}] \times [\underline{p}_{i_2}, \bar{p}_{i_2}]$  defined

by

$$\begin{aligned}\bar{\mathcal{P}} &= \left\{ \left( p_{i_1}^{k_{i_1}}, p_{i_2}^{k_{i_2}} \right), k_{i_1}, k_{i_2} \in \{0, \dots, 100\} \right\}, \\ p_{i_j}^{k_{i_j}} &= \underline{p}_{i_j} + \frac{k_{i_j}}{100} \left( \bar{p}_{i_j} - \underline{p}_{i_j} \right), k_{i_j} \in \{0, \dots, 100\}, j \in \{1, 2\}.\end{aligned}\tag{4.62}$$

Once the prices have been derived on  $\bar{\mathcal{P}}$ , we compute the discrete  $L^\infty(\bar{\mathcal{P}})$  and  $L^2(\bar{\mathcal{P}})$  error measures,

$$\begin{aligned}\varepsilon_{L^\infty}(\bar{N}) &= \max_{p \in \bar{\mathcal{P}}} \left| Price^p - I_{\bar{N}}(Price^{(\cdot)})(p) \right|, \\ \varepsilon_{L^2}(\bar{N}) &= \sqrt{\Delta_{\bar{\mathcal{P}}} \sum_{p \in \bar{\mathcal{P}}} \left| Price^p - I_{\bar{N}}(Price^{(\cdot)})(p) \right|^2},\end{aligned}\tag{4.63}$$

where  $\Delta_{\bar{\mathcal{P}}} = \frac{(\bar{p}_{i_1} - \underline{p}_{i_1})}{100} \frac{(\bar{p}_{i_2} - \underline{p}_{i_2})}{100}$ , to interpret the accuracy of our implementation and of the Chebyshev method as such.

#### 4.4.1 European Options

In Gaß et al. (2016), we first did a consistency study for a plain vanilla European call option on one asset as well as a European digital down&out option. The empirically observed (sub)exponential error decay for increasing  $N$  as well as the error of the Chebyshev interpolation on a test grid, verify the theoretical results from Theorem 4.3.2. The detailed results are in (Gaß et al., 2016, Section 5.1).

#### 4.4.2 Basket and Path-dependent Options

In this section we use the Chebyshev method to price basket and path-dependent options. First, we apply the method to interpolate Monte-Carlo estimates of prices of financial products and check the resulting accuracy. To this aim we exemplarily choose basket, barrier and lookback options in 5-dimensional Black&Scholes, Heston and Merton models. Second, we combine the Chebyshev method with a Crank-Nicolson finite difference solver with Brennan Schwartz approximation, see Brennan and Schwartz (1977), for pricing a univariate American put option in the Black&Scholes model. The finite difference solver and the Monte-Carlo implementation for the Heston and Merton models was provided by Maximilian Mair. In this section, I used these implementation to produce the results. For the efficiency study later on, I modified the provided codes.

In our Monte-Carlo simulation we use  $10^6$  sample paths, antithetic variates as variance reduction technique and 400 time steps per year. The error of the Monte-Carlo method

cannot be computed directly. We thus turn to statistical error analysis and use the well-known 95% confidence bounds to determine the accuracy. These bounds are derived by following the assumption of a normally distributed Monte-Carlo estimator with mean equal to the estimator's value and variance equal to the empirical variance. We pick two free parameters  $p_{i_1}, p_{i_2}$  out of (4.58),  $1 \leq i_1 < i_2 \leq D$ , in each model setup and fix all other parameters at reasonable constant values. In this section we define the discrete parameter grid  $\bar{\mathcal{P}} \subseteq [\underline{p}_{i_1}, \bar{p}_{i_1}] \times [\underline{p}_{i_2}, \bar{p}_{i_2}]$  by

$$\begin{aligned} \bar{\mathcal{P}} &= \left\{ \left( p_{i_1}^{k_{i_1}}, p_{i_2}^{k_{i_2}} \right), k_{i_1}, k_{i_2} \in \{0, \dots, 40\} \right\}, \\ p_{i_j}^{k_{i_j}} &= \underline{p}_{i_j} + \frac{k_{i_j}}{40} \left( \bar{p}_{i_j} - \underline{p}_{i_j} \right), k_{i_j} \in \{0, \dots, 40\}, j \in \{1, 2\}, \end{aligned} \quad (4.64)$$

and call  $\bar{\mathcal{P}}$  test grid. On this test grid the largest confidence bound is 0.025 on average less than 0.013. For the finite difference method, we investigate the error for all parameter tuples in  $\bar{\mathcal{P}}$  by comparing each approximation to the limit of the sequence of finite difference approximations with increasing grid size. Here, the error was below 0.005. In our calculations we work with a grid size in time as well as in space (log-moneyness) of  $50 \cdot \max\{1, T\}$  and compared the result to the resulting prices using grid sizes of  $1000 \cdot \max\{1, T\}$ . This grid size has been determined as sufficient for the limit due to hardly any changes compared to grid sizes of  $500 \cdot \max\{1, T\}$ .

Here, our main concern is the accuracy of the Chebyshev interpolation when we vary for each option the parameters strike and maturity analogously to the previous section. For  $N \in \{5, 10, 30\}$ , we precompute the Chebyshev coefficients as defined in (4.33) with  $D = 2$  where always  $N_1 = N_2 = N$ . An overview of fixed and free parameters in our model selection is given in Table 4.1. For computational simplicity in the Monte-Carlo simulation, we assume uncorrelated underlyings.

Let us briefly define the payoffs of the multivariate basket and path-dependent options. The payoff profile of a basket option for  $d$  underlyings is given as

$$f^K(S_T^1, \dots, S_T^d) = \left( \left( \frac{1}{d} \sum_{j=1}^d S_T^j \right) - K \right)^+.$$

We denote  $S_t = (S_t^1, \dots, S_t^d)$ ,  $\underline{S}_T^j := \min_{0 \leq t \leq T} S_t^j$  and  $\bar{S}_T^j := \max_{0 \leq t \leq T} S_t^j$ . A lookback option for  $d$  underlyings is defined as

$$f^K(\bar{S}_T^1, \dots, \bar{S}_T^d) = \left( \left( \frac{1}{d} \sum_{j=1}^d \bar{S}_T^j \right) - K \right)^+.$$

4 Chebyshev Polynomial Interpolation Method

Model	fixed parameters		free parameters	
	$p^1$	$p^2$	$p^1$	$p^2$
BS	$S_0^j = 100,$ $r = 0.005$	$\sigma_j = 0.2$	$K \in [83.33, 125]$	$T \in [0.5, 2]$
Heston	$S_0^j = 100,$ $r = 0.005$	$\kappa_j = 2,$ $\theta_j = 0.2^2,$ $\sigma_j = 0.3,$ $\rho_j = -0.5,$ $v_{j,0} = 0.2^2$	$K \in [83.33, 125]$	$T \in [0.5, 2]$
Merton	$S_0^j = 100,$ $r = 0.005$	$\sigma_j = 0.2,$ $\alpha_j = -0.1,$ $\beta_j = 0.45,$ $\lambda_j = 0.1$	$K \in [83.33, 125]$	$T \in [0.5, 2]$

Table 4.1: Parametrization of models, basket and path-dependent options. The model parameters are given for  $j = 1, \dots, d$  to reflect the multivariate setting with free parameters strike  $K$  and maturity  $T$ .

As a multivariate barrier option on  $d$  underlyings we define the payoff

$$f^K(\{S(t)\}_{0 \leq t \leq T}) = \left( \left( \frac{1}{d} \sum_{j=1}^d S_T^j \right) - K \right)^+ \cdot \mathbb{1}_{\{\underline{S}_T^j \geq 80, j=1, \dots, d\}}.$$

For an American put option the payoff is the same as for a European put,

$$f^K(S_t) = (K - S_t)^+,$$

but the option holder has the right to exercise the option at any time  $t$  up to maturity  $T$ .

We now turn to the results of our numerical experiments. In order to evaluate the accuracy of the Chebyshev interpolation we look for the worst case error  $\varepsilon_{L^\infty}$ . The absolute error of the Chebyshev interpolation method can be directly computed by comparing the interpolated option prices with those obtained by the reference numerical algorithm i.e. either the Monte-Carlo or the Finite Difference method. Since the Chebyshev interpolation matches the reference method on the Chebyshev nodes, we will use the out-of-sample test grid as in (4.64). Table 4.2 shows the numerical results for the basket and path-dependent options for  $N = 5$ , Table 4.3 for  $N = 10$  and Table 4.4 for  $N = 30$ . In addition to the  $L^\infty$  errors the tables display the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which

4 Chebyshev Polynomial Interpolation Method

Model	Option	$\varepsilon_{L^\infty}$	MC price	MC conf. bound	CI price
BS	Basket	$1.338 \cdot 10^{-1}$	8.6073	$1.171 \cdot 10^{-2}$	8.4735
Heston	Basket	$9.238 \cdot 10^{-2}$	0.0009	$1.036 \cdot 10^{-4}$	0.0933
Merton	Basket	$9.815 \cdot 10^{-2}$	8.8491	$1.552 \cdot 10^{-2}$	8.7510
BS	Lookback	$2.409 \cdot 10^{-1}$	9.4623	$9.861 \cdot 10^{-3}$	9.2213
Heston	Lookback	$5.134 \cdot 10^{-1}$	0.0314	$6.472 \cdot 10^{-4}$	-0.4820
Merton	Lookback	$2.074 \cdot 10^{-1}$	1.0919	$9.568 \cdot 10^{-3}$	0.8844
BS	Barrier	$1.299 \cdot 10^{-1}$	1.0587	$5.092 \cdot 10^{-3}$	1.1887
Heston	Barrier	$1.073 \cdot 10^{-1}$	2.7670	$9.137 \cdot 10^{-3}$	2.6597
Merton	Barrier	$9.916 \cdot 10^{-2}$	1.3810	$1.102 \cdot 10^{-2}$	1.4802

Table 4.2: Interpolation of exotic options with Chebyshev interpolation.  $N = 5$  and  $d = 5$  in all cases. In addition to the  $L^\infty$  errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the  $L^\infty$  error is realized.

Model	Option	$\varepsilon_{L^\infty}$	MC price	MC conf. bound	CI price
BS	Basket	$2.368 \cdot 10^{-3}$	2.4543	$7.493 \cdot 10^{-3}$	2.4566
Heston	Basket	$2.134 \cdot 10^{-3}$	3.1946	$1.073 \cdot 10^{-2}$	3.1925
Merton	Basket	$3.521 \cdot 10^{-3}$	6.1929	$2.231 \cdot 10^{-2}$	6.1894
BS	Lookback	$2.861 \cdot 10^{-2}$	0.9827	$4.197 \cdot 10^{-3}$	0.9541
Heston	Lookback	$1.098 \cdot 10^{-1}$	2.0559	$4.826 \cdot 10^{-3}$	2.1656
Merton	Lookback	$3.221 \cdot 10^{-2}$	4.7072	$1.264 \cdot 10^{-2}$	4.7394
BS	Barrier	$4.414 \cdot 10^{-3}$	5.3173	$1.725 \cdot 10^{-2}$	5.3129
Heston	Barrier	$5.393 \cdot 10^{-3}$	0.7158	$5.879 \cdot 10^{-3}$	0.7212
Merton	Barrier	$3.376 \cdot 10^{-3}$	9.2688	$2.302 \cdot 10^{-2}$	9.2722

Table 4.3: Interpolation of exotic options with Chebyshev interpolation.  $N = 10$  and  $d = 5$  in all cases. In addition to the  $L^\infty$  errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the  $L^\infty$  error is realized.

the  $L^\infty$  error is realized.

The results show that for  $N = 30$  the accuracy is for all selected options at a level of  $10^{-3}$ . We see that the Chebyshev interpolation error is dominated by the Monte-Carlo confidence bounds to a degree which renders it negligible in a comparison between the two. For basket and barrier options the  $L^\infty$  error already reaches satisfying levels of order  $10^{-3}$  at  $N = 10$  already. Again, the Chebyshev approximation falls within the confidence bounds of the Monte-Carlo approximation. Thus, Chebyshev interpolation with only  $121 = (10 + 1)^2$  nodes suffices for mimicking the Monte Carlo pricing results.

4 Chebyshev Polynomial Interpolation Method

Model	Option	$\varepsilon_{L^\infty}$	MC price	MC conf. bound	CI price
BS	Basket	$1.452 \cdot 10^{-3}$	5.1149	$1.200 \cdot 10^{-2}$	5.1163
Heston	Basket	$1.047 \cdot 10^{-3}$	7.6555	$1.371 \cdot 10^{-2}$	7.6545
Merton	Basket	$3.765 \cdot 10^{-3}$	7.2449	$2.359 \cdot 10^{-2}$	7.2412
BS	Lookback	$3.766 \cdot 10^{-3}$	25.9007	$1.032 \cdot 10^{-2}$	25.9045
Heston	Lookback	$1.914 \cdot 10^{-3}$	16.4972	$9.754 \cdot 10^{-3}$	16.4991
Merton	Lookback	$3.646 \cdot 10^{-3}$	27.1018	$1.623 \cdot 10^{-2}$	27.1054
BS	Barrier	$5.331 \cdot 10^{-3}$	5.6029	$1.730 \cdot 10^{-2}$	5.6082
Heston	Barrier	$2.486 \cdot 10^{-3}$	3.6997	$1.353 \cdot 10^{-2}$	3.6972
Merton	Barrier	$4.298 \cdot 10^{-3}$	6.6358	$2.309 \cdot 10^{-2}$	6.6315

Table 4.4: Interpolation of exotic options with Chebyshev interpolation.  $N = 30$  and  $d = 5$  in all cases. In addition to the  $L^\infty$  errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the  $L^\infty$  error is realized.

This statement does not hold for lookback options, where the  $L^\infty$  error still differs noticeably when comparing  $N = 10$  to  $N = 30$ . As can be seen from Table 4.2 Chebyshev interpolation with  $N = 5$  may yield unreliable pricing results. For lookback options in the Heston model we even observe negative prices in individual cases. Chebyshev pricing

$N$	$\varepsilon_{L^\infty}$	FD price	CI price
5	$3.731 \cdot 10^{-3}$	1.9261	1.9224
10	$1.636 \cdot 10^{-3}$	12.0730	12.0746
30	$3.075 \cdot 10^{-3}$	6.3317	6.3286

Table 4.5: Interpolation of one-dimensional American puts with Chebyshev interpolation in the Black&Scholes model. In addition to the  $L^\infty$  errors the table displays the Finite Differences (FD) prices and the Chebyshev Interpolation (CI) prices for those parameters at which the  $L^\infty$  error is realized.

of American options in the Black&Scholes model is even more accurate as illustrated in Table 4.5. Here, already for  $N = 5$  the accuracy of the reference method is achieved. We conclude that the Chebyshev interpolation is highly promising for the evaluation of multivariate basket and path-dependent options. Yet the accuracy of the interpolation critically depends on the accuracy of the reference method at the nodal points which motivates further analysis that we perform in the subsequent subsection.

### Interaction of Approximation Errors at Nodal Points and Interpolation Errors

The Chebyshev method is most promising for use cases, where computationally intensive pricing methods are required. For applying here a Chebyshev interpolation, the issue of distorted prices at the nodes and their consequences rises naturally. The observed noisy prices at the nodal points are

$$Price_{\varepsilon}^{p^{(k_1, \dots, k_D)}} = Price^{p^{(k_1, \dots, k_D)}} + \varepsilon^{p^{(k_1, \dots, k_D)}},$$

where  $\varepsilon^{p^{(k_1, \dots, k_D)}}$  is the approximation error introduced by the underlying numerical technique at the Chebyshev nodes. Due to linearity, the resulting interpolation is of the form

$$I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})(p) = I_{\overline{N}}(Price^{(\cdot)})(p) + I_{\overline{N}}(\varepsilon^{(\cdot)})(\cdot) \quad (4.65)$$

with the error function

$$\varepsilon^{(p)} = \sum_{j_D=0}^{N_D} \dots \sum_{j_1=0}^{N_1} c_{j_1, \dots, j_D}^{\varepsilon} T_{j_1, \dots, j_D}(p), \quad (4.66)$$

with the coefficients  $c_j^{\varepsilon}$  for  $j = (j_1, \dots, j_D) \in J$  given by

$$c_j^{\varepsilon} = \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right) \sum_{k_1=0}^{N_1} \dots \sum_{k_D=0}^{N_D} \varepsilon^{p^{(k_1, \dots, k_D)}} \prod_{i=1}^D \cos \left( j_i \pi \frac{k_i}{N_i} \right). \quad (4.67)$$

If  $\varepsilon^{p^{(k_1, \dots, k_D)}} \leq \bar{\varepsilon}$  for all Chebyshev nodes  $p^{(k_1, \dots, k_D)}$ , we obtain

$$|\varepsilon^{(p)}| \leq 2^D \bar{\varepsilon} \prod_{i=1}^D (N_i + 1), \quad (4.68)$$

since the Chebyshev polynomials are bounded by 1. This yields the following remark.

**Remark 4.4.1.** Let  $\mathcal{P} \ni p \mapsto Price^p$  be given as in Theorem 4.2.10 and assume that  $\varepsilon^{p^{(k_1, \dots, k_D)}} \leq \bar{\varepsilon}$  for all Chebyshev nodes  $p^{(k_1, \dots, k_D)}$ . Then

$$\begin{aligned} & \max_{p \in \mathcal{P}} |Price^p - I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})(p)| \\ & \leq \min\{a(\varrho, N, D), b(\varrho, N, D)\} + 2^D \bar{\varepsilon} \prod_{i=1}^D (N_i + 1). \end{aligned} \quad (4.69)$$

The following example shall illustrate the practical consequences of Remark 4.4.1. In the setting of (Gaß et al., 2016, Corollary 3) we set  $[S_0/\overline{K}, \overline{S}_0/\underline{K}] = [0.8, 1.2]$ ,  $[\underline{T}, \overline{T}] = [0.5, 2]$ . This results in  $\zeta^1 = \frac{2.5}{1.5} = \frac{5}{3}$  and  $\zeta^2 = \frac{2}{0.4} = 5$ . Thus, for  $\varrho_1 = 2.9 \in (1, 3)$  and

#### 4 Chebyshev Polynomial Interpolation Method

$\varrho_2 = 9.8 \in (1, 5 + \sqrt{24})$ , Remark 4.4.1 yields with  $N_1 = N_2 = 6$ ,

$$\max_{p \in \mathcal{P}} |Price^p - I_{\bar{N}}(Price^{(\cdot)})(p)| \leq 0.0072 + 196 \cdot \bar{\epsilon}.$$

In this example, the accuracy of the reference method has to reach a level of  $10^{-5}$  to guarantee an overall error of order  $10^{-3}$ . This demonstrates a trade-off between increasing  $N_1$  and  $N_2$  compared to the accuracy of the reference method. The error bound above is rather conservative. Our experiments from the previous section suggest that this bound highly overestimates the errors empirically observed. However, the presented error bound from Remark 4.4.1 can guarantee a desired accuracy by determining an adequate number of Chebyshev nodes and the corresponding accuracy of the reference method used at the Chebyshev nodes. For practical implementations, we suggest the following procedure. For a prescribed accuracy, without considering any distortion at the nodal points, the  $N_i$ ,  $i = 1, \dots, D$ , can be determined from the first term in (4.69) by choosing  $N_i$ ,  $i = 1, \dots, D$ , as small as possible such that the prescribed accuracy is attained. Accordingly, the accuracy that the reference method needs to achieve is bounded by the second term. A very accurate reference method in combination with small  $N_i$ ,  $i = 1, \dots, D$ , promise best results. With this rule of thumb in mind the experiments of Section 4.4.3 below have been conducted.

#### 4.4.3 Study of the Gain in Efficiency

In the previous section, we investigated the accuracy of the Chebyshev polynomial interpolation method using Monte-Carlo as reference pricing methods. Finally, we investigate the gain in efficiency achieved by the method in comparison to Monte-Carlo pricing. We compute the results on a PC with Intel Xeon CPU with 3.10 GHz with 20 MB SmartCache. All codes are written in Matlab R2014a.

#### Comparison to Monte-Carlo pricing

In this section, we choose a multivariate lookback option in the Heston model, based on 5 underlyings, as an example. For the efficiency study, we first vary one parameter, then we vary two.

*Variation of one model parameter*

For the multivariate lookback option in the Heston model, the following parameters are fixed with  $j = 1, \dots, 5$

$$\begin{aligned} S_0^j &= 100, & r &= 0.005, & K &= 100, & T &= 1, \\ \kappa_j &= 2, & \theta_j &= 0.2^2, & \rho_j &= -0.5, & v_{j,0} &= 0.2^2. \end{aligned} \tag{4.70}$$

#### 4 Chebyshev Polynomial Interpolation Method

As the free parameter in the Chebyshev interpolation, we pick the volatility of the volatility coefficient  $\sigma = \sigma_j$ ,  $j = 1, \dots, 5$ ,

$$\sigma \in [\sigma_{\min}, \sigma_{\max}], \quad \sigma_{\min} = 0.1, \quad \sigma_{\max} = 0.5. \quad (4.71)$$

The benchmark method is standard Monte-Carlo pricing, again with  $10^6$  sample paths, antithetic variates as variance reduction technique, and 400 time steps per year. We refer to this setting as the benchmark setting.

Following the discussion from Section 4.4.2, when we evaluate the prices at the nodal points, we guarantee a small  $\bar{\varepsilon}$  in the Monte-Carlo method by enriching the Monte-Carlo setting to  $5 \cdot 10^6$  sample paths, antithetic variates, and 400 time steps per year. In Table 4.6, we present the accuracy results for the Chebyshev interpolation with  $N_{\text{Cheby}}^{\text{Heston}} = 6$  based on the enriched Monte-Carlo setting. To this end, we compare the absolute differences of the Chebyshev prices and the enriched Monte-Carlo prices on the test grid  $\bar{\mathcal{P}} \subseteq [\underline{p}, \bar{p}]$ ,

$$\begin{aligned} \bar{\mathcal{P}} &= \left\{ \left( \sigma^k \right), k \in \{0, \dots, M\} \right\}, \\ \sigma^k &= \sigma_{\min} + \frac{k}{M} (\sigma_{\max} - \sigma_{\min}), k \in \{0, \dots, M\}. \end{aligned} \quad (4.72)$$

The highest observed error on the test grid with  $M = 20$  is at a level of  $10^{-2}$ . On the same test grid with  $M = 20$ , the benchmark Monte-Carlo setting has a worst-case confidence bound of  $1.644 \cdot 10^{-2}$ , and by comparing the benchmark Monte-Carlo prices to the enriched Monte-Carlo prices on this test grid, the maximal absolute error is  $7.361 \cdot 10^{-3}$ . Therefore, we conclude that the Monte-Carlo benchmark setting and the presented Chebyshev interpolation method have a roughly comparable accuracy. On the basis of this accuracy study, we now turn to the comparison of run-times.

We compare the run-times of the Chebyshev interpolation with  $N_{\text{Cheby}}^{\text{Heston}} = 6$ , where the offline phase is based on the enriched Monte-Carlo setting, to the run-times of the Monte-Carlo benchmark setting described above.

Table 4.7 provides the results in each case. The results for  $M = 1$  were empirically measured, all others were extrapolated from that, since the same amount of computation time would have had to be invested for each parameter set. The table indicates that from  $M = 50$  onwards interpolation by Chebyshev is faster. In Figure 4.6, for each  $M = 1, \dots, 100$ , we additionally present the run-times of the Chebyshev interpolation method, including the offline phase, compared to the Monte-Carlo method. Here, we observe that for  $M = 35$  both lines intersect and for  $M > 35$  the Chebyshev interpolation method is faster. The intersection of the two lines does not occur at  $M = N_{\text{Cheby}}^{\text{Heston}} + 1$ . This reflects the fact that a Monte-Carlo method with more sample paths was used in the offline phase for the Chebyshev interpolation.

#### 4 Chebyshev Polynomial Interpolation Method

Varying	$\varepsilon_{L^\infty}$	MC price	MC conf. bound	CI price
$\sigma$	$9.970 \cdot 10^{-3}$	18.6607	$4.592 \cdot 10^{-3}$	18.6707

Table 4.6: Interpolation of multivariate lookback options with Chebyshev interpolation for  $N = 6$  based on an enriched Monte-Carlo setting with  $5 \cdot 10^6$  sample paths, antithetic variates, and 400 time steps per year. In addition to the  $L^\infty$  error on the test grid, we also report the Monte-Carlo (MC) price, the Monte-Carlo confidence bound, and the Chebyshev Interpolation (CI) price for the parameters at which the  $L^\infty$  error is realized. We observe that the accuracy of the Chebyshev interpolation for  $N = 6$  is roughly in the same range as the accuracy of the benchmark Monte-Carlo setting (worst-case confidence bound of  $1.644 \cdot 10^{-2}$  and worst-case error of  $7.361 \cdot 10^{-3}$ ).

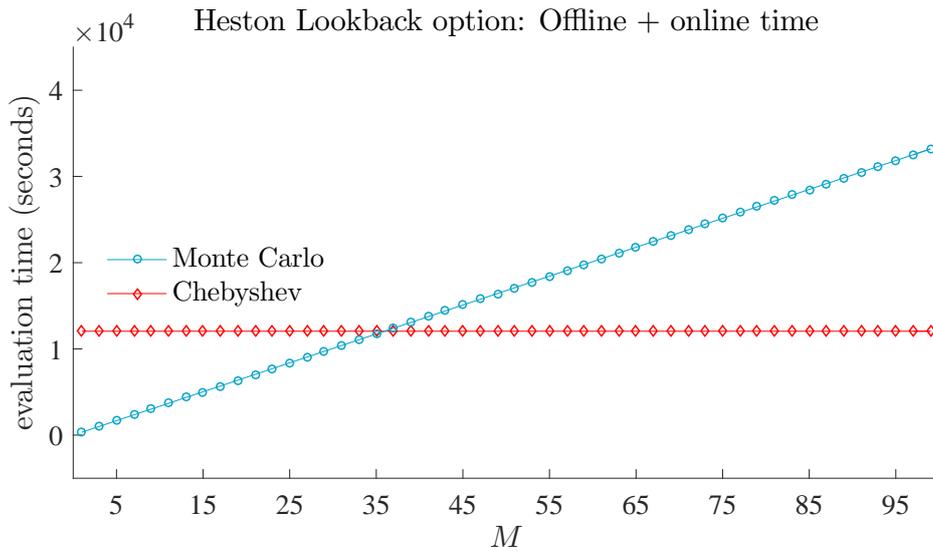


Figure 4.6: Efficiency study for a multidimensional lookback option in the Heston model with 5 underlyings varying one model parameter  $\sigma$ . Comparison of run-times of Monte-Carlo pricing with Chebyshev pricing including the offline phase. Both methods have been set up to deliver comparable accuracies. We observe that both curves intersect at roughly  $M = 35$ .

#### 4 Chebyshev Polynomial Interpolation Method

$M$	1	10	50	100
$T_{\text{online}}^{\text{Cheby}} (s)$	$2.7 \cdot 10^{-5}$	$2.7 \cdot 10^{-4}$	$1.4 \cdot 10^{-3}$	$2.7 \cdot 10^{-3}$
$T_{\text{offline+online}}^{\text{Cheby}} (s)$	$1.2 \cdot 10^4$	$1.2 \cdot 10^4$	$1.2 \cdot 10^4$	$1.2 \cdot 10^4$
$T^{\text{Monte-Carlo}} (s)$	$3.4 \cdot 10^2$	$3.4 \cdot 10^3$	$1.7 \cdot 10^4$	$3.4 \cdot 10^4$
$\frac{T_{\text{offline+online}}^{\text{Cheby}}}{T^{\text{Monte-Carlo}}}$	3473.4%	347.3%	69.5%	34.73%

Table 4.7: Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings. Here, we vary one model parameter and compare the Chebyshev results to Monte-Carlo. Both methods have been set up to deliver comparable accuracies. As the number of computed prices increases, the Chebyshev algorithm increasingly profits from the initial investment of the offline phase.

##### *Variation of two model parameters*

We choose  $\rho_j = \rho$ ,  $j = 1, \dots, 5$ , and vary

$$\begin{aligned} \rho &\in [\rho_{\min}, \rho_{\max}], & \rho_{\min} &= -1, & \rho_{\max} &= 1, \\ \sigma &\in [\sigma_{\min}, \sigma_{\max}], & \sigma_{\min} &= 0.1, & \sigma_{\max} &= 0.5, \end{aligned} \tag{4.73}$$

fixing all other parameters to the values of setting (4.70). In order to guarantee a roughly comparable accuracy between the Chebyshev interpolation method and the benchmark Monte-Carlo pricing, we use the following test grid  $\overline{\mathcal{P}} \subseteq [\sigma_{\min}, \sigma_{\max}] \times [\rho_{\min}, \rho_{\max}]$ ,

$$\begin{aligned} \overline{\mathcal{P}} &= \left\{ \left( \sigma^{k_1}, \rho^{k_2} \right), k_1, k_2 \in \{0, \dots, M\} \right\}, \\ \sigma^{k_1} &= \sigma_{\min} + \frac{k_1}{M} (\sigma_{\max} - \sigma_{\min}), k_1 \in \{0, \dots, M\}, \\ \rho^{k_2} &= \rho_{\min} + \frac{k_2}{M} (\rho_{\max} - \rho_{\min}), k_2 \in \{0, \dots, M\}. \end{aligned}$$

In Table 4.8, we present the accuracy results for the Chebyshev interpolation with  $N_{\text{Cheby}}^{\text{Heston}} = 6$  based on the enriched Monte-Carlo setting by setting  $M = 20$ . Comparing the benchmark Monte-Carlo setting and the enriched Monte-Carlo setting on this test grid with  $M = 20$ , we observe that the maximal absolute error is  $2.791 \cdot 10^{-2}$  and the confidence bounds of the benchmark Monte-Carlo setting do not exceed  $6.783 \cdot 10^{-2}$ .

To compare the run-times, we show the run-times that are required to compute the prices for  $M^2$  parameter tuples for different values of  $M$ . Again, the run-times are measured for  $M = 1$  and extrapolated for other values of  $M$ . Table 4.9 presents the results. In Figure 4.7, for each  $M = 1, \dots, 100$ , the run-times of the Chebyshev interpolation method, including the offline phase, are presented and compared to the Monte-Carlo method. We observe that for  $M = 15$  both lines intersect and for  $M > 15$  the Chebyshev

#### 4 Chebyshev Polynomial Interpolation Method

Varying	$\varepsilon_{L^\infty}$	MC price	MC conf. bound	CI price
$\sigma, \rho$	$5.260 \cdot 10^{-2}$	5.239	$1.428 \cdot 10^{-2}$	5.292

Table 4.8: Interpolation of multivariate lookback options with Chebyshev interpolation for  $N = 6$  based on an enriched Monte-Carlo setting with  $5 \cdot 10^6$  sample paths, antithetic variates, and 400 time steps per year. In addition to the  $L^\infty$  error on the test grid, we also report the Monte-Carlo (MC) price, the Monte-Carlo confidence bound, and the Chebyshev Interpolation (CI) price for the parameters at which the  $L^\infty$  error is realized. We observe that the accuracy of the Chebyshev interpolation  $N = 6$  is roughly in the same range as the accuracy of the benchmark Monte-Carlo setting (worst-case confidence bound of  $6.783 \cdot 10^{-2}$  and worst-case error of  $2.791 \cdot 10^{-2}$ ).

$M$	<b>Heston</b>			
	1	10	50	100
$T_{\text{online}}^{\text{Cheby}} (s)$	$7.1 \cdot 10^{-4}$	$7.1 \cdot 10^{-2}$	1.8	7.1
$T_{\text{offline+online}}^{\text{Cheby}} (s)$	$8.2 \cdot 10^4$	$8.2 \cdot 10^4$	$8.2 \cdot 10^4$	$8.2 \cdot 10^4$
$T^{\text{Monte-Carlo}} (s)$	$3.4 \cdot 10^2$	$3.4 \cdot 10^4$	$8.4 \cdot 10^5$	$3.4 \cdot 10^6$
$\frac{T_{\text{offline+online}}^{\text{Cheby}}}{T^{\text{Monte-Carlo}}}$	24313.9%	243.1%	9.7%	2.4%

Table 4.9: Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings. Here, we vary two model parameters and compare the Chebyshev results to Monte-Carlo. Both methods have been set up to deliver comparable accuracies. As the number of computed prices increases, the Chebyshev algorithm increasingly profits from the initial investment of the offline phase.

method outperforms its benchmark. Contrary to the case where only one parameter is varied, the intersection of both lines occurs at a significantly lower value of  $M$  due to the fact that for each  $M$  pricing must be performed for  $M^2$  parameter tuples.

Additionally, Table 4.9 highlights that, in the case of a total number of  $50^2$  parameter tuples, the Chebyshev method exhibits a significant decrease in (total) pricing run-times. For the maximal number of  $100^2$  parameter tuples that we investigated, pricing in either model resulted in more than 97% of run-time savings in our implementation. While computing  $100^2$  Heston prices using the Monte-Carlo method requires up to 39 days, the Chebyshev method computes the very same prices in 23 hours only. Note that only 7 seconds of this time span are consumed by actual pricing during the online phase.

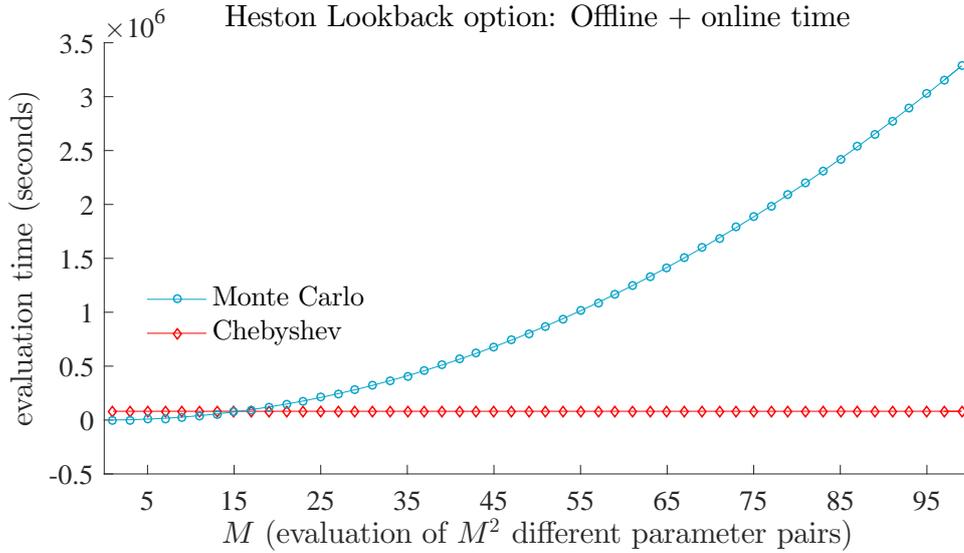


Figure 4.7: Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings, varying the two model parameters  $\sigma$  and  $\rho$ . Comparison of run-times for Monte-Carlo pricing and Chebyshev pricing including the offline phase. Both methods have been set up to deliver comparable accuracies. We observe that the Monte-Carlo and the Chebyshev curves intersect at roughly  $M = 15$ .

#### 4.4.4 Relation to Advanced Monte-Carlo Techniques

Up to this point, we have compared the Chebyshev interpolation method with a standard Monte-Carlo technique. Since the invention of Monte-Carlo methods in the 1940s, see Metropolis (1987), Monte-Carlo techniques have been further developed. In particular, quasi Monte-Carlo and multilevel Monte-Carlo methods have proved to be significantly more efficient in a variety of examples in mathematical finance, L'Ecuyer (2009) and Giles (2015). Thus, by employing these techniques in the offline phase, the Chebyshev interpolation method can be enhanced. In terms of efficiency, we expect Figure 4.7 to change only by rescaling the time axis: The run-time for the computation of the Monte-Carlo prices on the test grid is reduced proportionally. Obviously, the offline phase of the Chebyshev interpolation scales in the same way. As a first improvement of our implementation of the offline phase, in which, for each nodal point we produce a new independent set of samples, one can reuse a once drawn sample set to compute the prices at all nodal points. Furthermore, the run-time of the offline phase can be reduced significantly by parallelisation.

### Relation to parametric multilevel Monte-Carlo

There is an interesting relation between the Chebyshev interpolation approach that we proposed in this section and the parametric multilevel Monte-Carlo method suggested by S. Heinrich in Heinrich (1998) and Heinrich and Sindambiwe (1999). To be more precise, as concisely explained in Section 2.1 in Heinrich (2001), the starting point of Heinrich (1998) is the interpolation of the function

$$p^1 \mapsto E(f^{p^1}(X)) \tag{4.74}$$

and the computation of  $E[f^{p^k}(X)]$  at the nodes  $p_k$  with Monte-Carlo. Note that in this setting, the random variable  $X$  is not parametric. Next, he introduces the multilevel Monte-Carlo method. This is a hierarchical procedure based on nested grids. In each step, the estimator of the coarser grid serves as a control variate. The grids then are chosen optimally to balance cost and accuracy. Heinrich and Sindambiwe (1999) shows that the resulting algorithm is optimal for a certain class of problems. This class of problems is characterized by the regularity of the function  $(p^1, x) \mapsto f^{p^1}(x)$ , namely that it belongs to a Sobolev space. The order  $r$  of the partial derivatives in  $(p^1, x)$  is the determining factor for the efficiency. In particular, the weak partial derivatives in both the parameters  $p^1 \in \mathbb{R}^D$  and in  $x \in \mathbb{R}^d$  need to exist in order to apply the approach.

In contrast, our error analysis is based on the regularity of the mapping

$$(p^1, p^2) \mapsto E(f^{p^1}(X)^{p^2}). \tag{4.75}$$

The resulting problem class is significantly larger than the setting of Heinrich (1998) and Heinrich and Sindambiwe (1999). This is essential for applications in finance, as the examples of a European call and digital option prove: The payoff function of a European call has a kink. The call option prices as a function of the parameters, however, are in many cases even analytic. This is also the case for digital options, whose payoffs are not even weakly differentiable, see Gaß et al. (2016).

We relate the error analysis presented in Section 4.4.2 with the results of Heinrich and Sindambiwe (1999). For given cost, Heinrich and Sindambiwe (1999) presents the expected error in the  $L^2$ -norm. We, on the other hand, work with the expectation of the stronger  $L^\infty$ -norm. This norm is more suitable for quantifying mispricing. Since the Chebyshev interpolation is tailored to minimize the maximum error, this comes without additional cost. In the following lemma, we present the expected error in the  $L^\infty$ -norm. We assume that the error of a Monte-Carlo simulation with cost  $M$  is unbiased and normally distributed with standard deviation  $\sigma_M$ .

**Lemma 4.4.2.** *Let  $\mathcal{P} \ni p \mapsto \text{Price}^p$  be given as in Theorem 4.3.2 and let the errors at all nodal points be independently and identically normally distributed with distribution*

$\mathcal{N}(0, \sigma_M)$ . Then,

$$\begin{aligned} & E \left( \max_{p \in \mathcal{P}} |Price^p - I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})(p)| \right) \\ & \leq 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}} + 2^D \prod_{i=1}^D (N_i + 1) \sigma_M \sqrt{\frac{\pi}{2}}. \end{aligned} \quad (4.76)$$

*Proof.* As in Section 4.4.2, Theorem 4.3.2 yields

$$\begin{aligned} & E \left( \max_{p \in \mathcal{P}} |Price^p - I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})(p)| \right) \\ & \leq E \left( 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}} + 2^D \sum_{k_1=0}^{N_1} \dots \sum_{k_d=0}^{N_D} |\varepsilon^{p^{(k_1, \dots, k_D)}}| \right) \\ & = 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}} + 2^D \sum_{k_1=0}^{N_1} \dots \sum_{k_d=0}^{N_D} E \left( |\varepsilon^{p^{(k_1, \dots, k_D)}}| \right). \end{aligned}$$

The assertion follows from  $E(|\varepsilon^{p^{(k_1, \dots, k_D)}}|) = \sigma_M \sqrt{\frac{\pi}{2}}$ .  $\square$   $\square$

In Heinrich (2001), it is shown that there exist constants  $c_1$  and  $c_2$  such that for each integer  $M$  the cost of the parametric Monte-Carlo method is bounded by  $c_1 M$  and the error is bounded by  $c_2 M^{-\alpha}$ , where  $\alpha$  depends on the regularity of the function  $f$  and  $\alpha \in (0, \frac{1}{2})$ . The index  $\alpha$  depends on the dimension of the parameter space and the Sobolev order of the function space to which  $f$  belongs.

To present an error analysis in the same spirit, we observe that, in terms of costs, if  $\bar{c}_1$  denotes the cost of one Monte-Carlo simulation, it follows directly that the cost of deriving the interpolation  $I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})$  is bounded by  $\bar{c}_1 \mathcal{M}$ . We therefore define the *upper bound of the offline cost* of the Chebyshev method by  $\bar{c}_1 \mathcal{M} = \bar{c}_1 \prod_{i=1}^D (N_i + 1) M$ , where  $N_i$  is the number of nodal points in dimension  $i$  and  $M$  is the number of sample paths at each nodal point. In order to estimate the error, according to the central limit theorem it is reasonable to assume  $\sigma_M = \sigma / \sqrt{M}$  for large  $M$  and  $\sigma_M$  from Lemma 4.4.2.

**Theorem 4.4.3.** *Let the assumptions of Lemma 4.4.2 hold, further let  $\sigma_M = \sigma / \sqrt{M}$  for  $\sigma_M$ . For each  $\beta \in (0, 1/2)$  there exist constants  $\bar{c}_1, \bar{c}_2 > 0$  such that for each integer  $\mathcal{M} > 1$  there is a choice of  $M, N$  such that the offline cost of the Chebyshev method is bounded by  $\bar{c}_1 \mathcal{M}$  and*

$$E \left( \max_{p \in \mathcal{P}} |Price^p - I_{\overline{N}}(Price_{\varepsilon}^{(\cdot)})(p)| \right) \leq \bar{c}_2 \mathcal{M}^{-\beta}.$$

#### 4 Chebyshev Polynomial Interpolation Method

*Proof.* To balance cost and efficiency, we choose the number of sample paths in (4.76) to be  $M$  and the number of nodal points of the Chebyshev interpolation in an appropriate way. Combining (4.76) and Corollary 4.43 results in

$$E \left( \max_{p \in \mathcal{P}} |Price^p - I_{\bar{N}}(Price_{\varepsilon}^{(\cdot)})(p)| \right) \leq C_1 \underline{\varrho}^{-N} + C_2 \prod_{i=1}^D (N_i + 1) \frac{1}{\sqrt{M}},$$

where  $\underline{\varrho} = \min_{1 \leq i \leq D} \varrho_i$ ,  $\underline{N} = \min_{1 \leq i \leq D} N_i$ , and  $C_2 = 2^D \sigma \sqrt{\frac{\pi}{2}}$ .

Let  $\beta \in (0, \frac{1}{2})$ . We show that there exists  $\bar{c}_2$  such that  $C_1 \underline{\varrho}^{-N} + C_2 \prod_{i=1}^D (N_i + 1) \frac{1}{\sqrt{M}} \leq \bar{c}_2 \mathcal{M}^{-\beta}$ . Therefore, we select the number of nodal points in each direction depending on  $M$ . For simplicity, we choose the same number of nodal points in each dimension and set  $(N_i + 1)$  as a function of  $M$ , i.e.  $N_i + 1 = N(M)$ . Furthermore, we set  $N(M) := M^{\frac{\alpha-\beta}{D\beta}}$  for some  $\alpha > \beta$ , which yields  $M^{-\alpha} = \mathcal{M}^{-\beta}$ . Moreover, we set  $\tilde{C}_1 = C_1 \underline{\varrho}$ . Next, we want to find  $\bar{c}_2 > 0$  such that  $\tilde{C}_1 \underline{\varrho}^{-N(M)} \leq \frac{\bar{c}_2}{2} M^{-\alpha}$  and  $C_2 \frac{N(M)^D}{\sqrt{M}} \leq \frac{\bar{c}_2}{2} M^{-\alpha}$ , which implies the statement of the theorem. Both inequalities are equivalent to

$$N(M) \leq \sqrt[D]{\frac{\bar{c}_2 M^{\frac{1}{2}-\alpha}}{2C_2}}, \quad \text{and} \quad N(M) \geq \log_{\underline{\varrho}} \left( \frac{2\tilde{C}_1 M^{\alpha}}{\bar{c}_2} \right).$$

The first inequality becomes

$$M^{\frac{\alpha-\beta}{\beta} + \alpha - \frac{1}{2}} \leq \frac{\bar{c}_2}{2C_2},$$

which holds if  $1/2 > \alpha > \beta > \frac{\frac{\alpha}{\frac{3}{2}-\alpha}}{2}$  and  $\bar{c}_2$  satisfies  $\frac{\bar{c}_2}{2C_2} \geq 1$ . We therefore set  $\alpha := (1 + \epsilon)\beta$  with  $\epsilon := \min(1/(2\beta) - 1, 3/(2(1 + \beta)) - 1)/2$ . The second inequality becomes  $M^{\frac{\alpha-\beta}{D\beta}} \geq \log_{\underline{\varrho}} \left( \frac{2\tilde{C}_1 M^{\alpha}}{\bar{c}_2} \right)$ , which can be satisfied by an appropriate choice of  $\bar{c}_2$ .  $\square$

**Remark 4.4.4.** (i) *The error of the multilevel Monte-Carlo estimate of (Heinrich, 2001, Theorem 1) decays with  $\sqrt{\mathcal{M}}$ , if the function  $f$  is sufficiently regular. This is the only case in which the asymptotic order of convergence in (Heinrich, 2001, Theorem 1) is slightly better than the result of Theorem 4.4.3, which gives an order  $\beta$  arbitrarily close to  $1/2$ , even for the stronger  $L^\infty$ -norm.*

(ii) *In contrast to (Heinrich, 2001, Theorem 1), the payoff function  $(p^1, x) \mapsto f^{p^1}(x)$  is not required to be weakly differentiable to a specific order. Moreover, Theorem 4.4.3 allows a parametrized random variable  $X^{p^2}$ .*

(iii) *In Theorem 4.4.3, the error of the resulting Chebyshev interpolation is put in relation to the cost of the offline phase. This is in the spirit of Heinrich (2001), and, together with (i) and (ii) of this remark, shows that our approach is competitive. From an application point of view, however, the cost of the online phase is crucial.*

#### 4 Chebyshev Polynomial Interpolation Method

*This applies especially to cases where a real-time evaluation in the online phase is required and the offline phase can be executed in idle times.*

- (iv) *The online cost is proportional to the number of nodal points. If the highest priority is given to the efficiency of the online phase, one can proceed as follows to achieve a pre-specified accuracy  $\varepsilon$ : First, choose the number of nodal points such that the first summand of the error bound in (4.76) is smaller than  $\varepsilon/2$ . Then, choose the number of samples  $M$  of the selected Monte-Carlo technique such that also the second summand of the error bound in (4.76) is smaller than  $\varepsilon/2$ . See for instance the example after Remark 4.4.1.*

Referring to part (iv) of Remark 4.4.4, with increasing dimension of the parameter space the second summand in (4.76) grows exponentially. The following lemma shows that by considering the expected weighted  $L^2$ -error for each parameter  $p$  individually instead of the expected  $L^\infty$ -norm, the bound reduces considerably. As weight function we use  $w(p)$  as in (4.52) and we first investigate the case  $\mathcal{P} = [-1, 1]^D$ .

**Lemma 4.4.5.** *Let  $[-1, 1]^D \ni p \mapsto \text{Price}^p$  be given as in Theorem 4.3.2 and let the errors at all nodal points be independently and identically normally distributed with distribution  $\mathcal{N}(0, \sigma_M)$ . Then,*

$$\begin{aligned} & \left( E \left( \int_{[-1,1]^D} (\text{Price}^p - I_{\overline{N}}(\text{Price}_\varepsilon^{(\cdot)})(p))^2 w(p) dp \right) \right)^{\frac{1}{2}} \\ & \leq 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1-\varrho_j^{-2}} \right)^{\frac{1}{2}} + \left( (2\pi)^D \prod_{i=1}^D (N_i + 1) \right)^{\frac{1}{2}} \sigma_M. \end{aligned} \quad (4.77)$$

*Proof.* First, we apply the Minkowski inequality to split the estimate as follows

$$\begin{aligned} & \left( E \left( \int_{[-1,1]^D} (\text{Price}^p - I_{\overline{N}}(\text{Price}_\varepsilon^{(\cdot)})(p))^2 w(p) dp \right) \right)^{\frac{1}{2}} \\ & = \left( E \left( \int_{[-1,1]^D} (\text{Price}^p - I_{\overline{N}}(\text{Price}^{(\cdot)})(p) + I_{\overline{N}}(\varepsilon^{(\cdot)})(\cdot))^2 w(p) dp \right) \right)^{\frac{1}{2}} \\ & \leq \left( E \left( \int_{[-1,1]^D} (\text{Price}^p - I_{\overline{N}}(\text{Price}^{(\cdot)})(p))^2 w(p) dp \right) \right)^{\frac{1}{2}} \end{aligned} \quad (4.78)$$

$$+ \left( E \left( \int_{[-1,1]^D} (I_{\overline{N}}(\varepsilon^{(\cdot)})(\cdot))^2 w(p) dp \right) \right)^{\frac{1}{2}} \quad (4.79)$$

The first summand (4.78) is bounded by  $2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1-\varrho_j^{-2}} \right)^{\frac{1}{2}}$ . To

#### 4 Chebyshev Polynomial Interpolation Method

estimate the second summand (4.79), we make use of the orthogonalities (4.6) and (4.7) of the Chebyshev polynomials.

$$\begin{aligned}
& \left( E \left( \int_{[-1,1]^D} (I_{\overline{N}}(\varepsilon^{(\cdot)})(\cdot))^2 w(p) dp \right) \right)^{\frac{1}{2}} = \left( E \left( \int_{[-1,1]^D} \left( \sum_j c_j T_j(p) \right)^2 w(p) dp \right) \right)^{\frac{1}{2}} \\
& = \left( E \left( \int_{[-1,1]^D} \left( \sum_{j_1=0}^{N_1} \cdots \sum_{j_D=1}^{N_D} c_{(j_1, \dots, j_D)} \prod_{i=1}^D T_{j_i}(p_i) \right)^2 w(p) dp \right) \right)^{\frac{1}{2}} \quad (4.80) \\
& \leq \left( E \left( \pi^D \sum_j c_j^2 \right) \right)^{\frac{1}{2}}
\end{aligned}$$

In (4.80), we applied the orthogonality. Note that we introduce the lower equal relation due to the fact that we just estimated the weighted integral over the product of two Chebyshev polynomials by  $\pi$  and not distinguish cases, in which it actually is  $\frac{\pi}{2}$ .

$$\begin{aligned}
& \left( E \left( \pi^D \sum_j c_j^2 \right) \right)^{\frac{1}{2}} \\
& = \left( \pi^D \sum_j E \left( \left( \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right) \sum_{k_1=0}^{N_1} \cdots \sum_{k_D=0}^{N_D} \varepsilon^{p^{(k_1, \dots, k_D)}} \prod_{i=1}^D \cos \left( j_i \pi \frac{k_i}{N_i} \right) \right)^2 \right) \right)^{\frac{1}{2}}
\end{aligned}$$

The last step has been possible due to the fact that the errors at the nodal points are independent and identically distributed. Knowing that  $\varepsilon$  is normally distributed with normal distribution  $\mathcal{N}(0, \sigma_M)$ , we define a new random variable,

$$Y := \sum_{k_1=0}^{N_1} \cdots \sum_{k_D=0}^{N_D} \varepsilon^{p^{(k_1, \dots, k_D)}} \prod_{i=1}^D \cos \left( j_i \pi \frac{k_i}{N_i} \right)$$

As a sum of normally distributed random variables  $Y$  is normally distributed as well. In our error estimation, we want to find a bound for (4.79) and we apply that  $Y$  is normally distributed with a variance that is lower than the variance of a  $\mathcal{N}(0, \sum_{k_1=0}^{N_1} \cdots \sum_{k_D=0}^{N_D} \sigma_M) = \mathcal{N}(0, \prod_{i=1}^D N_i \sigma_M)$  normal distribution. Let  $\tilde{Y} \sim \mathcal{N}(0, \prod_{i=1}^D N_i \sigma_M)$ . This leads to the

following estimate,

$$\begin{aligned}
 \left( E \left( \pi^D \sum_j c_j^2 \right) \right)^{\frac{1}{2}} &\leq \left( \pi^D \sum_j \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right)^2 E(\tilde{Y}^2) \right)^{\frac{1}{2}} \\
 &\leq \left( \pi^D \sum_j \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right)^2 \prod_{i=1}^D N_i^2 \sigma_M^2 \right)^{\frac{1}{2}} \\
 &\leq \left( (2\pi)^D \sum_j \sigma_M^2 \right)^{\frac{1}{2}} = \left( (2\pi)^D \prod_{i=1}^D (N_i + 1) \right)^{\frac{1}{2}} \sigma_M.
 \end{aligned}$$

Combining the estimates for (4.78) and (4.79) yields the assertion.  $\square$

The complexity considerations from Theorem 4.4.3 can be derived analogously.

Finally, let us mention that the multilevel Monte-Carlo method of Heinrich and Sindambiwe (1999) is described for an arbitrary nodal interpolation and illustrated by a piecewise linear interpolation. The efficiency analysis in (Heinrich, 2001, Theorem 1) and Theorem 4.4.3 focuses only on the complexity of the offline phase and ignores the complexity of the online phase. In this view, the number of nodal points of the interpolation is less important. Whereas this can be appropriate for a one-dimensional parameter space, a simple example makes clear how crucial it can become for multivariate parameter spaces to require as few nodal points as possible to achieve a pre-specified accuracy. For instance, when interpolating piecewise linearly on an equidistant grid in the multilevel Monte-Carlo method of Heinrich (2001) with  $L$  levels,  $2^L$  nodal points in each direction are applied. For a  $D$ -dimensional parameter space, this results in  $2^{LD}$  nodal points. For  $L = 10$  and  $D = 2$ , this results in more than 1 million nodal points. In this case, the online cost is in the range of the cost of a Monte-Carlo simulation, which makes the interpolation redundant. Applying Chebyshev polynomial interpolation, a small number of nodal points such as 7, as shown in Section 4.4.3, suffices for the Chebyshev interpolation method to obtain an appropriate accuracy. In this case, the total number of nodes is 49 for the tensorized Chebyshev interpolation in two dimensions. Thus, the online cost outperforms Monte-Carlo significantly.

This highlights the fact that the choice of the interpolation method is crucial. One promising idea is to combine the Chebyshev interpolation with multilevel Monte-Carlo. To do so, a hierarchically structured interpolation is essential. This can be achieved by setting the degree of the Chebyshev interpolations to  $2^{l-1}N$ , where  $l = 1, \dots, L$  denotes the level.

## 4.5 Conclusion and Outlook

We have applied the Chebyshev method to parametric option pricing. Within the scope of European options, we defined Conditions 4.3.1 that guarantee (sub-)exponential error convergence. Additionally, also the numerical experiments for American, barrier and lookback options display promising results. A theoretical error analysis for all nonlinear pricing problems is beyond the scope of the thesis, while we are convinced that further investigations in this direction are valuable. Regarding the pricing of American options as optimal stopping problems, we introduce in the next chapter the dynamic Chebyshev interpolation method. The theoretical and experimental results of our case studies show that the method can perform considerably well when few parameters are varied. As a consequence, we recommend the interpolation method for this case and also when solely the strike of a plain vanilla option is varied and fast Fourier methods are available. For calibration purposes for example, strikes are not given in a discrete logarithmic scale, which makes an additional interpolation necessary in order to apply FFT. Here, Chebyshev polynomials offer an attractive alternative. In particular, the maturity can be used as supplementary free variable. Moreover, for models with a low number of parameters, another path could be beneficial: Interpolating the objective function of the parameters directly. Then the optimization would boil down to a minimization of a tensorized polynomial, which could be exploited in further research. As may be seen from Armenti et al. (2015), where the Chebyshev interpolation is applied, this advantage can also be exploited for other optimization procedures in finance for example in risk allocation.

For higher dimensions, the curse of dimensionality occurs by using the tensorized multivariate Chebyshev interpolation. Hence, for multivariate polynomial interpolation, the introduction of sparsity techniques promise higher efficiency, as for instance by compression techniques for tensors as reviewed by Kolda and Bader (2009). The high potential of low-rank tensor methods is illustrated in a numerical example for evaluating spread options in the bivariate Black&Scholes model, which is available online, see Glau et al. (2017). Moreover, Trefethen (2016) presents error analysis for multivariate polynomial approximation.

Additionally, connecting with the idea in Figure 1.1, we refer to Gaß et al. (2016), who replace the Chebyshev interpolation with empirical interpolation for Fourier pricing methods and avoid the curse of dimensionality in the parameters assuming that an only one-dimensional underlying is given.

# 5 Dynamic Programming Framework with Chebyshev Interpolation

The art of doing mathematics is finding that special case that contains all the germs of generality.

---

David Hilbert

*This chapter is based on Glau et al. (2017a) and Glau et al. (2017b), and presents the parts to which I provided a significant contribution.*

Previously, we have seen that the Chebyshev interpolation works rather nicely for parametric option pricing. Focusing on American option pricing, from the theoretical perspective, statements about convergence and error decay are missing in this approach so far. Additionally, by applying the Chebyshev interpolation in the parameter space, we have to consider the curse of dimensionality. In this chapter, we present an idea of applying Chebyshev interpolation to dynamic programming problems. The backbone of this idea is the reduction of dynamic programming problems to the derivation of the conditional expectations of Chebyshev polynomials. The special case is in this case, the Chebyshev polynomials which allow us a strict connection to the previous error analysis. As we will see, this approach is rather general and provides a very broad framework for solving dynamic programming problems. Furthermore, it can be extended to other polynomial interpolation techniques, too. Additionally, the empirical interpolation approach in the parameter space can be combined with solving a dynamic programming problem with Chebyshev interpolation and, thus, avoid the curse of dimensionality in the parameter space.

Now, we will introduce the Bellman-Wald equation as a specific form of dynamic programming, illustrate how the Chebyshev interpolation can be combined with it and then apply it to price American options. In the following, we follow the illustrations in Peskir and Shiryaev (2006). Originating in Wald (1947), sequential analysis has been introduced as a method with the characteristic feature that, at the beginning of the experiment, the number of observations is not pre-specified. Here, at each stage of the experiment, the decision to terminate depends on the results of the observation previously made. In Bellman (1957), the Wald-Bellman equation backward induction is applied as a dynamic programming principle. As introduced before, American options give the option-holder the right to exercise the option at any time up to maturity  $T$ . In this context, the

question arises at which time it is optimal to exercise the option. This lies in the field of optimal stopping. We assume that the process is Markovian and first, we let the time be discrete. We work with the setting as in Peskir and Shiryaev (2006), a time-homogeneous Markov chain  $X = (X_n)_{n \geq 0}$  on a filtered probability space  $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n \geq 0}, \mathcal{P}_x)$  taking values in a measurable space  $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ , where  $\mathcal{B}(\mathcal{X})$  is the Borel  $\sigma$ -algebra on  $\mathcal{X} \subset \mathbb{R}^d$ . It is assumed that the chain  $X$  starts at  $x$  under  $\mathcal{P}_x$  for  $x \in \mathcal{X}$  and that the mapping  $x \rightarrow \mathcal{P}_x(F)$  is measurable for each  $F \in \mathcal{F}$ .

Given a measurable function  $g : \mathcal{X} \rightarrow \mathbb{R}$  fulfilling (with  $g(X_N) = 0$  if  $N = \infty$ ),

$$E_x \left[ \sup_{0 \leq n \leq N} |g(X_n)| \right] < \infty \quad \text{for all } x \in \mathcal{X}, \quad (5.1)$$

the following optimal stopping problem is investigated,

$$V^N(x) = \sup_{0 \leq \tau \leq N} E_x[G(X_\tau)], \quad (5.2)$$

where  $x \in \mathcal{X}$  and the supremum is taken over all stopping times  $\tau$  of  $X$  with respect to  $(\mathcal{F}_n)_{n \geq 0}$ . In our setting, the time horizon is finite. We focus on pricing American options with maturity  $T < \infty$ . We set  $G_n = G(X_n)$  and thus, we can use the method of backward induction which leads to a recursively-defined sequence of random variables  $(S_n^N)_{0 \leq n \leq N}$ ,

$$\begin{aligned} S_n^N &= G_N \quad \text{for } n = N, \\ S_n^N &= V^{N-n} \quad \text{for } n = N-1, \dots, 0. \end{aligned}$$

For  $0 \leq n \leq N$  we consider the following stopping time,

$$\tau_n^N = \inf\{n \leq k \leq N : S_k^N = G_k\}.$$

As shown in (Peskir and Shiryaev, 2006, Proof of Theorem 1.7), the identity  $S_n^N = V^{N-n}(X_n)$  holds and we can introduce a second stopping time

$$\tau_D = \inf\{0 \leq n \leq N : X_n \in D_n\},$$

where we set for  $0 \leq n \leq N$ ,

$$D_n = \{x \in \mathcal{X} : V^{N-n}(x) = G(x)\}.$$

This setting allows us to state the following result from Peskir and Shiryaev (2006).

**Proposition 5.0.1.** *(Peskir and Shiryaev, 2006, Theorem 1.7). Assume condition (5.1) holds and consider the optimal stopping problem (5.2). Then the value function  $V^N(x)$  satisfies the Wald-Bellman equations*

$$V^n(x) = \max\{G(x), E_x[V^{n-1}(X_{n-1}) | \mathcal{F}_{N-n}]\} \quad \text{for all } x \in E,$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

for  $n = 1, \dots, N$  where  $V^0 = G$ . Moreover,

- (i) The stopping time  $\tau_D$  is optimal in (5.2).
- (ii) If  $\tau_*$  is an optimal stopping time in (5.2), then  $\tau_D \leq \tau_*$   $P_X$  - a.s. for every  $x$  in  $\mathcal{X}$ .
- (iii) The sequence  $(V^{N-n}(X_n))_{0 \leq n \leq N}$  is the smallest supermartingale which dominates  $(G(X_n))_{0 \leq n \leq N}$  under  $P_x$  for  $x \in \mathcal{X}$  given and fixed.
- (iv) The stopped sequence  $(V^{N-n \wedge \tau_D}(X_{n \wedge \tau_D}))_{0 \leq n \leq N}$  is a martingale under  $P_x$  for every  $x \in \mathcal{X}$ .

This illustrating introduction is for the time-homogeneous case. In (Peskir and Shiryaev, 2006, Chapter 1.1) also the time-inhomogeneous case is presented. Although our numerical examples in Section 5.6 later on are for the time-homogeneous case, we formulate our Dynamic Programming Principle for both cases. We apply a discrete time stepping  $t = t_1 < \dots < t_{n_T} = T$ . Note that  $n_T$  refers to the number of time steps we apply between  $t$  and  $T$ . For notational convenience, we indicate the value function at each time step with subscript  $t_u$  to directly refer to the time step  $t_u$ . Additionally, we allow for an arbitrary function  $f$ , not necessarily the maximum function at each time step to provide a more general framework. The theoretic convergence results later on will incorporate the Lipschitz constant of  $f$ , namely  $L_f$ .

**Definition 5.0.2.** We consider the following Dynamic Programming Principle (DPP) with  $V_t$  as solution,

$$V_T(x) = g(T, x) \tag{5.3}$$

$$V_{t_u}(x) = f(g(t_u, x), E[V_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = x]), \tag{5.4}$$

where  $t = t_1 < \dots < t_{n_T} = T$  denote the time steps.

Definition 5.0.2 allows us to investigate general dynamic programming principles, in which  $V_t(x)$  represents the value function depending on  $x$ . This can either be the underlying stock price in pricing, e.g. American options, or the wealth of an investor. In the following, we present how Chebyshev interpolation can be applied in this setting to simplify the derivation of the conditional expectations.

## 5.1 Derivation of Conditional Expectations

In this section, we present the combination of solving a DPP and using Chebyshev polynomial interpolation. Numerically, time consuming at each time step  $t_u$ ,  $t = t_1 < \dots < t_{n_T} = T$  is the derivation of the conditional expectation  $E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = x]$ . The key idea is to reduce the derivation of the conditional expectations to derive conditional expectations of Chebyshev polynomials. At the initial time  $T = t_{n_T}$ , we apply tensorized Chebyshev interpolation to the function  $g(T, x)$ , i.e. for  $x \in \mathcal{X}$ ,  $V_T(x) = g(T, x) \approx \sum_j c_j T_j(x)$ . Here,  $j$  is a multi-index reflecting the dimensionality of the underlying space  $\mathcal{X}$  and  $T_j$  is the tensorized Chebyshev polynomial. A detailed notation is given in Section 4.1.2. Note that at the initial time, the value function  $V_T(x)$  is equal to the function  $g(T, x)$  and a conditional expectation has not to be derived. From the first time step backwards in time, however, the conditional expectation has to be derived and, here, the Chebyshev interpolation is applied. At the first time step  $t_{n_T-1}$ , the derivation of  $E[g(X_{t_{n_T}})|X_{t_{n_T-1}} = x]$  is replaced by deriving  $E[\sum_j c_j T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T}}))|X_{t_{n_T-1}} = x]$ . In general, Chebyshev polynomials are defined for the domain  $[-1, 1]$ . Therefore, the transformations  $\tau_{\mathcal{X}}^{-1}$  as in (4.37) have to be applied.

The linearity of the expectations allows us to further simplify the conditional expectation of a sum to a sum of conditional expectations,  $\sum_j c_j E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T}}))|X_{t_{n_T-1}} = x]$ . Up to this point, the value function at  $t_{n_T-1}$  is approximated by

$$V_{t_{n_T-1}}(x) \approx f \left( g(t_{n_T-1}, x), \sum_j c_j E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T}}))|X_{t_{n_T-1}} = x] \right).$$

Moreover, we again apply a Chebyshev interpolation for the value function at time step  $V_{t_{n_T-1}}$ . Therefore,  $V_{t_{n_T-1}}$  only has to be evaluated at the specific Chebyshev nodes. Hence, denoting with  $x^k = (x_{k_1}, \dots, x_{k_D})$  the Chebyshev nodes, only for all nodal points the value function at  $t_{n_T-1}$  has to be evaluated,

$$V_{t_{n_T-1}}(x^k) \approx f \left( g(t_{n_T-1}, \tau_{\mathcal{X}}(x^k)), \sum_j c_j E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T}}))|X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \right).$$

We apply this procedure iteratively at each time step. Therefore, only the conditional expectations of the Chebyshev polynomials,  $E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}}))|X_{t_u} = \tau_{\mathcal{X}}(x^k)]$ , have to be evaluated at each time step  $t_u$  for each Chebyshev node  $x^k$ . This complete iterative procedure is described in detail in Algorithm 3. Moreover, if an equidistant time stepping is applied, the conditional expectations only have to be derived once, see Algorithm 4. Naturally, the question arises how the conditional expectations of the Chebyshev polynomials can be derived. Here, we present four different ways.

### Probability density function

For the derivation of  $E[T_j(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]$ , let the density function of  $X$  for the time  $t_{u+1} - t_u$  be given as  $\text{pdf}(x, \mu_{t_{u+1}-t_u})$ . Then, the conditional expectation can be derived by solving an integral,

$$\begin{aligned} E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}}))|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \\ = \int_{\underline{x}_1 - x_{k_1}}^{\bar{x}_1 - x_{k_1}} \dots \int_{\underline{x}_D - x_{k_D}}^{\bar{x}_D - x_{k_D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i)) \cdot \text{pdf}(x, \mu_{t_{u+1}-t_u}) dx_D \dots dx_1. \end{aligned}$$

This approach is rather intuitive and easy to implement. However, this approach is strictly connected to the availability of the probability density function and, as regards run-times, for all nodal points  $x^k$  a  $D$ -dimensional integral has to be solved.

### Fourier Transformation

An alternative approach, especially for cases in which the probability density function is not given explicitly, e.g. for the Merton model, is the Fourier transform. We illustrate this approach in the one dimensional case and assume first that the transformation  $\tau_{[\underline{x}_1, \bar{x}_1]}^{-1}$  is not required, i.e.  $\mathcal{X} = [-1, 1]$ .

We have,

$$\begin{aligned} \Gamma_{t_u, t_{u+1}}(T_j)(x^k) &= E[T_j(X_{t_{u+1}}) \mathbb{1}_{\{X_{t_{u+1}} \in [-1, 1]\}} | X_{t_u} = x^k] \\ &= \int_{-1-x^k}^{1-x^k} T_j(y + x^k) P^{X_{t_u, t_{u+1}}}(dy), \end{aligned}$$

where  $P^{X_{t_u, t_{u+1}}}$  is the distribution of the underlying process  $X$  over the time horizon  $t_{u+1} - t_u$ . Instead of directly computing these integrals, we decide to use Parseval's identity, see (2.16), and use Fourier transforms. We want to express the Fourier transform of  $T_j(y + x^k)$  with the linear shift of  $x^k$  as the Fourier transform of  $T_j(y)$ . Let the transformation  $H_{x^k} : y \mapsto y + x^k$  denote this linear transformation. Then,

$$\begin{aligned} \widehat{T_j \circ H_{x^k}}(\xi) &= \int_{x_{\min} - x^k}^{x_{\max} - x^k} e^{i\xi y} H_{x^k} T_j(y) dy = \int_{x_{\min} - x^k}^{x_{\max} - x^k} e^{i\xi y} T_j(x^k + y) dy \\ &= \int_{x_{\min}}^{x_{\max}} e^{i\xi(y-x^k)} T_j(y) dy = e^{-i\xi x^k} \widehat{T_j}(\xi). \end{aligned} \quad (5.5)$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

Then, we have,

$$\begin{aligned}\Gamma_{t_i, t_{i+1}}(T_j)(x^k) &= E[T_j(X_{t_{i+1}}) \mathbb{1}_{\{X_{t_{i+1}} \in [x_{\min}, x_{\max}]\}} | X_{t_i} = x^k] \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\xi x^k} \widehat{T_j}(\xi) \overline{P^{X_{t_i, t_{i+1}}}(\xi)} d\xi.\end{aligned}$$

$\widehat{P^{X_{t_i, t_{i+1}}}}(\xi)$  denotes the characteristic function of the underlying process  $X$ . The Fourier transform of the Chebyshev polynomials are presented in Dominguez et al. (2011) and the authors also provide a Matlab implementation.

So far, we have presented the way of applying Fourier transforms to derive the conditional expectations of the Chebyshev polynomials only for the special case  $\mathcal{X} = [-1, 1]$ . In the following, let us assume an arbitrary interval  $\mathcal{X} = [\underline{x}, \bar{x}]$ . In this case, the affine linear transformation  $\tau_{[\underline{x}, \bar{x}]}$  as in (4.37) comes into play. In (5.5), this transformation has to be considered, too. We get,

$$\widehat{T_j \circ H_{x^k}}(\xi) = \int_{\underline{x}-x^k}^{\bar{x}-x^k} e^{i\xi x} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x^k)) dx.$$

First, we set  $y = x + x^k$ . This yields

$$\begin{aligned}\int_{\underline{x}-x^k}^{\bar{x}-x^k} e^{i\xi x} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x^k)) dx &= \int_{\underline{x}}^{\bar{x}} e^{i\xi(y-x^k)} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(y)) dy \\ &= e^{-i\xi x^k} \int_{\underline{x}}^{\bar{x}} e^{i\xi y} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(y)) dy.\end{aligned}$$

Now, by setting  $z = \tau_{[\underline{x}, \bar{x}]}^{-1}(y)$  and using  $\frac{dy}{dz} = \frac{\bar{x}-\underline{x}}{2}$ , we get,

$$e^{-i\xi x^k} \int_{\underline{x}}^{\bar{x}} e^{i\xi y} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(y)) dy = e^{-i\xi x^k} \int_{-1}^1 e^{i\xi \tau_{[\underline{x}, \bar{x}]}(z)} T_j(z) \frac{\bar{x}-\underline{x}}{2} dz.$$

Plugging-in the definition of  $\tau_{[\underline{x}, \bar{x}]}$  results in,

$$\begin{aligned}e^{-i\xi x^k} \int_{-1}^1 e^{i\xi \tau_{[\underline{x}, \bar{x}]}(z)} T_j(z) \frac{\bar{x}-\underline{x}}{2} dz &= e^{-i\xi x^k} \int_{-1}^1 e^{i\xi(\bar{x}-\frac{\bar{x}-\underline{x}}{2})} e^{i\xi \frac{\bar{x}-\underline{x}}{2} z} T_j(z) \frac{\bar{x}-\underline{x}}{2} dz \\ &= e^{-i\xi x^k} e^{i\xi(\bar{x}-\frac{\bar{x}-\underline{x}}{2})} \frac{\bar{x}-\underline{x}}{2} \int_{-1}^1 e^{i\xi \frac{\bar{x}-\underline{x}}{2} z} T_j(z) dz.\end{aligned}$$

Denoting with  $\bar{\xi} = \frac{\bar{x}-\underline{x}}{2} \xi$ , we get,

$$\widehat{H_{x^k} T_j}(\xi) = e^{-i\xi x^k} e^{i\xi(\bar{x}-\frac{\bar{x}-\underline{x}}{2})} \frac{\bar{x}-\underline{x}}{2} \widehat{T_j}(\bar{\xi}).$$

### Truncated moments

In this approach, we use that each one dimensional Chebyshev polynomial can be represented as a sum of monomials, i.e.

$$T_j(x) = \sum_{l=0}^j a_l x^l, \quad j \in \mathbb{N}.$$

The coefficients  $a_l$ ,  $l = 0, \dots, j$ , see (4.2), can easily be derived using the chebfun function `poly()`, Driscoll et al. (2014). Then, if the one dimensional Chebyshev polynomials are given as  $T_{j_i}(x_i) = \sum_{l=0}^{j_i} a_{i,l} x_i^l$ ,  $i = 1, \dots, D$ , we can express

$$\prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i)) = \prod_{i=1}^D \left( \sum_{l=0}^{j_i} a_{i,l} (\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i))^l \right) = \sum_q a_q x^q, \quad (5.6)$$

where  $q = (q_1, \dots, q_D)$ ,  $q_i = 1, \dots, j_i$ ,  $a_q = \prod_{i=1}^D a_{i,q_i}$  and  $x^q = \prod_{i=1}^D (\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i))^{q_i}$ . Therefore, in order to determine

$$E \left[ \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(X_{i,t_{u+1}})) | X_{t_u} = \tau_{\mathcal{X}}(x^k) \right], \quad (5.7)$$

we derive the expectations

$$E \left[ \tau_{[\underline{x}_1, \bar{x}_1]}^{-1}(X_{1,t_{u+1}})^{j_1} \cdot \dots \cdot \tau_{[\underline{x}_D, \bar{x}_D]}^{-1}(X_{D,t_{u+1}})^{j_D} | X_{t_u} = \tau_{\mathcal{X}}(x^k) \right] \quad (5.8)$$

for the products of monomials and use (5.6) to assemble the expectation (5.7). So, the problem of the conditional expectations has been reduced to deriving moments. However, the Chebyshev polynomials do only have support on the compact domain  $\mathcal{X}$ . Therefore, we actually need to derive truncated moments, i.e. an indicator function has to be added,

$$E \left[ \tau_{[\underline{x}_1, \bar{x}_1]}^{-1}(X_{1,t_{u+1}})^{j_1} \cdot \dots \cdot \tau_{[\underline{x}_D, \bar{x}_D]}^{-1}(X_{D,t_{u+1}})^{j_D} \cdot \prod_{i=1}^D \mathbb{1}_{X_{i,t_{u+1}} \in [\underline{x}_i, \bar{x}_i]} | X_{t_u} = \tau_{\mathcal{X}}(x^k) \right].$$

For the normal distribution, Kan and Robotti (2016) present results for the multivariate truncated moments and also provide a Matlab implementation. We will use this result later in the application section to determine option prices in the one dimensional Black&Scholes setting, see Section 5.6.

### Monte-Carlo simulation

Lastly, especially in cases for which neither a probability density function, nor a characteristic function of the underlying process is given, the Monte-Carlo simulation as described in 2.2.3 is possible. The CEV model is one example of a model which has neither a probability density, nor a characteristic function in closed-form.

## 5.2 Dynamic Chebyshev in the Case of Analyticity

In the previous section, we presented several ways to derive the conditional expectations. Now, we focus on the algorithmic structure to solve the DPP (5.3)-(5.4). The key idea is that, in either stochastic control problems or numerical option pricing schemes, an iterative time stepping scheme is necessary and, in this thesis, we make use of the approximation of the specific target function as a Chebyshev polynomial. This significantly simplifies the iterative procedure:

- The derivation of conditional expectations is reduced to the derivation of conditional expectations of Chebyshev polynomials.
- The conditional expectations have to be determined for a relatively small number of nodal points of the interpolation compared to a (fine) space grid.
- In each time step, only the Chebyshev coefficients have to be saved. At  $t = 0$  the pricing of the derivative is reduced to an evaluation of a polynomial. This is especially beneficial when the price has to be determined for a variety of underlying values  $S$ .
- As illustrated in Section 4.2.1, the Chebyshev interpolation also interpolates the partial derivatives. Therefore, by just deriving the derivatives of the Chebyshev polynomial, the Greeks Delta and Gamma are computed with hardly any additional computational effort and especially no computational costs in the iterative scheme.
- In the so-called preparation/precomputation step, the conditional expectations of the Chebyshev polynomials are taken. Thus, in the time stepping scheme several different option types and payoffs can easily be evaluated quickly, because the conditional expectations are not tailored to a specific payoff.

Later on, we will introduce a splitting of the domain  $\mathcal{X}$  at each time step as, clearly, then the conditional expectations cannot be completely derived in the precomputation step.

### 5.2.1 Description of Algorithms

In the following, let the underlying process  $X$  be a stochastic process with state space  $\mathbb{R}^d$  and let two functions be given as  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  and  $g : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ . The function  $g$  will determine the initial condition at time  $T$  and enable the start of the iterative scheme. The function  $f$  determines the value of the value function at each time step  $t_u$ ,  $V_{t_u}(x)$ , see (5.3)-(5.4).

---

#### Algorithm 3 General algorithm

---

- 1: **procedure** PREPARATION STEP
  - 2:     Fix an interval  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$  for the interpolation
  - 3:     Fix  $N_1, \dots, N_D$  as the number of nodal points of the Chebyshev interpolation in each dimension
  - 4:     Determine nodal points  $x^k = (x_{k_1}, \dots, x_{k_D})$  with  $x^{k_i} = \cos\left(\pi \frac{k_i}{N_i}\right)$  for  $k_i = 0, \dots, N_i$  and  $i = 1, \dots, D$ , set  $y^k = \tau_{\mathcal{X}}(x^k)$
  - 5:     Denote with  $T_j$  the Chebyshev polynomial for all  $j \in J$  with  $j = (j_1, \dots, j_D)$ ,  $j_l = 0, \dots, N_l$
  - 6:     Set up time stepping  $0 = t_1, \dots, t_{n_T} = T$
  - 7:     For all  $j \in J$ , for all  $t_u$ ,  $u = 0, \dots, n_T - 1$ , for all  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$
  - 8:     Compute  $\Gamma_{t_u, t_{u+1}}(T_j)(y^k) := E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}})) | X_{t_u} = y^k]$
  - 9: **procedure** INITIAL TIME  $T$
  - 10:      $P_T(y^k) = g(T, y^k)$ ,  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$ , derive
  - 11:      $c_j(T) = \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right) \sum_{k_1=0}^{N_1} \dots \sum_{k_D=0}^{N_D} P_T(y^k) \prod_{i=1}^D \cos\left(j_i \pi \frac{k_i}{N_i}\right)$
  - 12:     Obtain Chebyshev interpolation of  $P_T(x) = \sum_{j \in J} c_j(T) T_j(x)$
  - 13: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 14:     Given Chebyshev interpolation of  $P_{t_{u+1}}(x) = \sum_{j \in J} c_j(t_{u+1}) T_j(\tau_{\mathcal{X}}^{-1}(x))$
  - 15:     Derivation of  $P_{t_u}(y^k)$ ,  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$  at the nodal points
  - 16:     **for**  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$
  - 17:          $P_{t_u}(y^k) = f(g(t_u, y^k), E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y^k])$
  - 18:          $P_{t_u}(y^k) = f(g(t_u, y^k), \sum_{j \in J} c_j(t_{u+1}) E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}})) | X_{t_u} = y^k])$
  - 19:          $P_{t_u}(y^k) = f(g(t_u, y^k), \sum_{j \in J} c_j(t_{u+1}) \Gamma_{t_u, t_{u+1}}(T_j)(y^k))$
  - 20:     **end**
  - 21:     Derive
  - 22:      $c_j(t_u) = \left( \prod_{i=1}^D \frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i} \right) \sum_{k_1=0}^{N_1} \dots \sum_{k_D=0}^{N_D} P_{t_u}(y^k) \prod_{i=1}^D \cos\left(j_i \pi \frac{k_i}{N_i}\right)$
  - 23:     Obtain Chebyshev interpolation of  $P_{t_u}(x) = \sum_{j \in J} c_j(t_u) T_j(\tau_{\mathcal{X}}^{-1}(x))$
  - 24: **procedure** DERIVING THE SOLUTION AT  $t = 0$
  - 25:      $P_0(x) = \sum_{j \in J} c_j(0) T_j(\tau_{\mathcal{X}}^{-1}(x))$
- 

Algorithm 3 can be simplified, if  $X$  is a Markov process with stationary increments and an equidistant time stepping scheme is used. In the preparation step, the derivation of

the conditional expectations has to be done once and not for each time step individually as Algorithm 4 shows.

---

**Algorithm 4** Simplified algorithm for equidistant time stepping and a Markov process with stationary increments

---

- 1: **procedure** PREPARATION STEP
  - 2:   Replace in Algorithm 3 Lines 6-8 with:
  - 3:   Set up equidistant time stepping  $0 = t_1, \dots, t_{n_T} = T$
  - 4:   For all  $j = (j_1, \dots, j_D)$ ,  $j_l = 0, \dots, N_l$ , for all  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$
  - 5:   Compute  $\Gamma_{\Delta t}(T_j)(y^k) := E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T}})) | X_{t_{n_T-1}} = y^k]$
  - 6: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 7:   Replace in Algorithm 3 Line 19 with:
  - 8:    $P_{t_u}(y^k) = f(g(t_u, y^k), \sum_{j \in J} c_j(t_{u+1}) \Gamma_{\Delta t}(T_j)(y^k))$
- 

### 5.2.2 Error Analysis

In this section, we analyze the error of the pricing scheme as described in Algorithm 3. First, we assume that the function only has support on  $\mathcal{X}$ . As illustrated in Algorithm 3,  $P_{t_u}(x)$  denotes the dynamic Chebyshev interpolation of  $V_{t_u}(x)$ .

**Remark 5.2.1.** *The error analysis in the following is connected to the error of the tensorized Chebyshev interpolation. In Theorem 4.2.10, we present our improved error bound. During the iterative time stepping procedure, this error bound will be applied at every time step. Therefore, we introduce a new notation*

$$\alpha(\varrho, N, D, V) := \min\{a(\varrho, N, D, V), b(\varrho, N, D, V)\} \quad (5.9)$$

where, denoting by  $S_D$  the symmetric group on  $D$  elements,

$$a(\varrho, N, D, V) = \min_{\sigma \in S_D} \sum_{i=1}^D 4V \frac{\varrho_{\sigma(i)}^{-N_i}}{\varrho_i - 1} + \sum_{k=2}^D 4V \frac{\varrho_{\sigma(k)}^{-N_k}}{\varrho_{\sigma(k)} - 1} \cdot 2^{k-1} \frac{(k-1) + 2^{k-1} - 1}{\prod_{j=1}^{k-1} (1 - \frac{1}{\varrho_{\sigma(j)}})},$$

$$b(\varrho, N, D, V) = 2^{\frac{D}{2}+1} \cdot V \cdot \left( \sum_{i=1}^D \varrho_i^{-2N_i} \prod_{j=1}^D \frac{1}{1 - \varrho_j^{-2}} \right)^{\frac{1}{2}}.$$

Note that in addition to the statement in Theorem 4.2.10,  $a(\varrho, N, D, V)$  and  $b(\varrho, N, D, V)$  are here also functions of the bound  $V$  of the interpolated function on the corresponding Bernstein ellipse.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

Additionally, for notational ease we introduce

$$C_{D,N} := 2^D \prod_{i=1}^D (N_i + 1). \quad (5.10)$$

**Theorem 5.2.2.** *Let a Dynamic Programming Principle be given as in (5.3) and (5.4). Given a time stepping  $t = t_1 < \dots < t_{n_T} = T$ , let  $\mathcal{X} \ni x \mapsto V_{t_u}(x)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho_{t_u})$  with parameter vector  $\varrho_{t_u} \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}, \varrho_{t_u})} |V_{t_u}(x)| \leq M_{t_u}$  for  $u = 1, \dots, n_T$ . Furthermore, let  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be continuous.*

*Then, by applying Algorithm 3, the resulting solution  $P_{t_u}(x)$  converges to the solution  $V_{t_u}(x)$  for  $\min_{i=1, \dots, D} N_i \rightarrow \infty$ . Furthermore, the approximation error at time  $t_u$  is given by*

$$\max_{x \in \mathcal{X}} |V_{t_u}(x) - P_{t_u}(x)| \leq \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} F_{t_u} =: \varepsilon_{t_u}, \quad (5.11)$$

where  $C_{D,N}$  as in (5.10),  $\alpha(\varrho_{t_u}, N, D, M_{t_u})$  as in (5.9) and

$$F_{t_u} := \max_{j \in J} |V_{t_u}(x_j) - P_{t_u}(x_j)|. \quad (5.12)$$

*Proof.* By constructing the error bound, we follow Algorithm 3 and construct the error bound recursively. At the initial time step,  $t_{n_T} = T$ ,  $P_{t_{n_T}}$  is the Chebyshev interpolation of  $g(T, x) = V_T(x)$ . From Theorem 4.2.10, we see that the interpolation error is bounded by

$$\max_{x \in \mathcal{X}} |V_T(x) - P_{t_{n_T}}(x)| \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}).$$

Now we consider the step from  $t_{n_T} \rightarrow t_{n_T-1}$ . At this step, we interpolate the function  $V_{t_{n_T-1}}$  with  $P_{t_{n_T-1}}$ . Unlike as in the initial time step, here we have to consider distortions at the nodal points of the Chebyshev interpolation. We use the interpolation from the previous time step  $P_{t_{n_T}}$  instead of the true value function  $V_{t_{n_T-1}}$  to evaluate the conditional expectations,

$$\begin{aligned} E[V_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] &\approx E[P_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \\ &= \sum_{j \in J} c_{j, t_{n_T}} E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{n_T-1}})) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)]. \end{aligned}$$

Therefore, a second error source is added. We define the error at the nodal points as

$$\max_k |V_{t_{n_T-1}}(\tau_{\mathcal{X}}(x^k)) - P_{t_{n_T-1}}(\tau_{\mathcal{X}}(x^k))| =: F_{t_{n_T-1}}.$$

Note that  $F_{t_{n_T-1}}$  depends on the error at the previous time step and also on the function

## 5 Dynamic Programming Framework with Chebyshev Interpolation

$f$  from the DPP (5.3)-(5.4). Now, following Remark 4.4.1 yields,

$$\max_{x \in \mathcal{X}} |V_{t_{n_T-1}}(x) - P_{t_{n_T-1}}(x)| \leq \alpha(\varrho_{t_{n_T-1}}, N, D, M_{t_{n_T-1}}) + C_{D,N} F_{t_{n_T-1}},$$

with  $C_{D,N}$  as in (5.10).

We denote the overall error bound at  $t_{n_T-1}$  with

$$\varepsilon_{t_{n_T-1}} = \alpha(\varrho_{t_{n_T-1}}, N, D, M_{t_{n_T-1}}) + C_{D,N} F_{t_{n_T-1}}.$$

This procedure can be applied iteratively through the time stepping of Algorithm 3. At the time step  $t_{u+1} \rightarrow t_u$ , the distortion at the nodal points between the value function and  $P_{t_{u+1}}$ ,  $F_{t_u}(f, \varepsilon_{t_{u+1}})$ , is derived using  $\varepsilon_{t_{u+1}}$ . Then, the overall error bound at  $t_u$  is again a combination of the Chebyshev interpolation error  $\alpha(\varrho_{t_u}, N, D, M_{t_u})$  and an additional error term driven by the distortion at the nodal points, i.e.

$$\varepsilon_{t_u} = \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} F_{t_u}.$$

Thus, the recursive nature of the error is hidden in the distortion term  $F_{t_u}$ . The continuity of the function  $f$  yields  $F_{t_u} \rightarrow 0$  with increasing  $N$ . Due to the convergence of  $P_{t_{u+1}}(x)$  to  $V_{t_{u+1}}(x)$ , the conditional expectation  $E[P_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]$  converges to  $E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]$  by the dominated convergence theorem applying the bound  $\varepsilon_{t_u}$ . The continuity of  $f$  then yields

$$\begin{aligned} & f\left(g(t_u, \tau_{\mathcal{X}}(x^k)), E[P_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]\right) \\ & \rightarrow f\left(g(t_u, \tau_{\mathcal{X}}(x^k)), E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]\right). \end{aligned}$$

The error of the Chebyshev interpolation  $\alpha(\varrho_{t_u}, N, D, M_{t_u})$  decreases exponentially. Concluding, with increasing  $N$ , the overall error bound  $\varepsilon_{t_u} \rightarrow 0$  for all  $u = 1, \dots, n_T$ .  $\square$

**Remark 5.2.3.** Assume that in the setting of Theorem 5.2.2, the conditional expectations  $E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}}))|X_{t_u} = \tau_{\mathcal{X}}(x^k)]$  cannot be derived exactly - due to the used evaluation technique, e.g. Monte-Carlo methods, an additional error is made. Let this error bounded by a constant  $\bar{\delta}$ . We assume that in (5.12), the recursive error reflecting this  $\bar{\delta}$  can be incorporated such that

$$F_{t_u}^{\bar{\delta}} = F_{t_u} + h(\bar{\delta}).$$

Here,  $F_{t_u}$  denotes the error assuming the conditional expectations can be evaluated exactly.

**Corollary 5.2.4.** Let the setting be as in Theorem 5.2.2. Furthermore, let  $f$  be Lipschitz continuous with constant  $L_f$ . The approximation  $P_{t_u}$  from Algorithm 3 converges

## 5 Dynamic Programming Framework with Chebyshev Interpolation

exponentially to the solution  $V_{t_u}$  and the error is bounded by

$$\varepsilon_{t_u} = \sum_{j=u}^{n_T} C_{D,N}^{j-u} L_f^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}). \quad (5.13)$$

*Proof.* The function  $f$  is Lipschitz continuous,

$$|f(x_1, y_1) - f(x_2, y_2)| \leq L_f(|x_1 - x_2| + |y_1 - y_2|).$$

In this case, we can calculate an upper bound for the distortion error in (5.12).

$$\begin{aligned} F_{t_u} &= \max_k |V_{t_u}(\tau_{\mathcal{X}}(x^k)) - P_{t_u}(\tau_{\mathcal{X}}(x^k))| \\ &= \max_k |f(g(t_u, \tau_{\mathcal{X}}(x^k)), E[P_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)]) \\ &\quad - f(g(t_u, \tau_{\mathcal{X}}(x^k)), E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)])| \\ &\leq \max_k L_f \left( \left| g(t_u, \tau_{\mathcal{X}}(x^k)) - g(t_u, \tau_{\mathcal{X}}(x^k)) \right| \right. \\ &\quad \left. + \left| E[P_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] - E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\ &= \max_k L_f \left( \left| E[P_{t_{u+1}}(X_{t_{u+1}}) - V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\ &\leq \max_k L_f \left( \left| E[\varepsilon_{t_{u+1}}|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\ &= L_f \cdot \varepsilon_{t_{u+1}}. \end{aligned}$$

This results in

$$\varepsilon_{t_u} = \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} L_f \varepsilon_{t_{u+1}}.$$

By induction, we now show (5.13). For  $u = n_T$  we have

$$\varepsilon_{n_T} = \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) = \sum_{j=n_T}^{n_T} C_{D,N}^{j-n_T} L_f^{j-n_T} \alpha(\varrho_{t_j}, N, D, M_{t_j}).$$

We assume that for  $n_T, \dots, u+1$  equation (5.13) holds. Then, we obtain for the error  $\varepsilon_{t_u}$

$$\begin{aligned} \varepsilon_{t_u} &= \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} L_f \varepsilon_{t_{u+1}} \\ &= \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} L_f \sum_{j=u+1}^{n_T} C_{D,N}^{j-(u+1)} L_f^{j-(u+1)} \alpha(\varrho_{t_j}, N, D, M_{t_j}) \\ &= \alpha(\varrho_{t_u}, N, D, M_{t_u}) + \sum_{j=u+1}^{n_T} C_{D,N}^{j-u} L_f^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}) \end{aligned}$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

This can be expressed as

$$\varepsilon_{t_u} = \sum_{j=u}^{n_T} C_{D,N}^{j-u} L_f^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}). \quad (5.14)$$

The error bound (5.9) then yields

$$\varepsilon_{t_u} \leq C \bar{N}^D n_T \underline{\varrho}^{-N}, \quad (5.15)$$

where  $\bar{N} = \max_{1 \leq i \leq D} N_i$ ,  $\underline{N} = \min_{1 \leq i \leq D} N_i$  and  $\underline{\varrho} = \min_{1 \leq j \leq n_T} \min_{1 \leq i \leq D} \varrho_{i,t_j}$ . The error bound consists of a polynomial term increasing in  $N$  and an exponentially decaying term in  $N$ . Overall, due to  $\underline{\varrho} > 1$  the exponential decaying behaviour dominates.  $\square$

**Remark 5.2.5.** *Assume that in the framework of Corollary 5.2.4 we have for the solutions  $V_{t_u}$  a constant parameter vector  $1 < \varrho \leq \varrho_{t_u}$  and a constant bound  $M \geq M_{t_u}$  for all  $u = 1, \dots, n_T$ . Furthermore, let  $N = N_i$  for  $i = 1, \dots, D$ . In this case, the error bound (5.14) can be written as*

$$\varepsilon_{t_u} = \alpha(\varrho, N, D, M) \sum_{j=u}^{n_T} (2^D (N+1)^D)^{j-u} L_f^{j-u}.$$

Although the dynamic Chebyshev framework offers a variety of applications, our first motivation has been the pricing of American option. By determining the price of an American option via solving the DPP, a time discretization is applied. Obviously in this case, we would rather have a Bermudan option with exercise dates exactly matching the applied discrete time stepping scheme. Therefore, we are theoretically interested in the error behaviour for  $n_T \rightarrow \infty$ .

**Remark 5.2.6.** *Assume we are in the setup of Corollary 5.2.4 and Remark 5.2.5. If we let  $N$  and  $n_T$  go to infinity, we have to make sure that the error bound goes to zero. The following conditions on the relation between  $n_T$  and  $N$  ensure convergence*

$$n_T < \frac{\log(\varrho)}{D} \cdot \frac{N}{\log(N)}.$$

**Remark 5.2.7.** *In many applications, we need  $f$  of the DPP (5.3) and (5.4) to be the maximum function  $(x, y) \mapsto \max\{x, y\}$ . This function is, of course, Lipschitz continuous with constant 1 and thus, we are in the framework of Corollary 5.2.4.*

The assumption of an analytic value function is relatively strong. So far, our error analysis is based on analytic value functions. In Section 4.2.1, we especially took an additional look at differentiable functions. This can be applied similarly at this point here, too, and will also be shown in Glau et al. (2017b). In the following, we present a different approach, splitting. Often the analyticity assumption is not satisfied on the

complete domain  $\mathcal{X}$  due to a few points. The idea now is to split the domain accordingly at these specific points into sub-domains in which the analyticity assumption holds.

### 5.3 Solutions for Kinks and Discontinuities

In this section, we pose solutions for the case when the mapping  $x \mapsto V_{t_u}(x)$  is not analytic a time step  $t_u$ . First, we present the idea of splitting the domain and second, the concept of applying mollifiers.

#### 5.3.1 Splitting of the Domain

Given a time stepping  $t = t_1 < \dots < t_{n_T} = T$ , in some applications  $\mathcal{X} \ni x \mapsto V_{t_u}(x)$  might not be analytic or differentiable on the complete domain  $\mathcal{X}$ . For instance, the payoff of a call or a put option, in the one dimensional case, at maturity has a kink at the strike  $K$  and is therefore not analytic in the domain  $[\underline{x}, \bar{x}]$  with  $\underline{x} < K < \bar{x}$ . In this section, we present the idea of splitting the hyperrectangular domain  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$  in several sub-domains, such that, on each sub-domain, the function itself is again analytic or differentiable and we can apply the theory presented above. Therefore, we split at each time step  $t_u$ , each one dimensional interval  $[\underline{x}_i, \bar{x}_i]$ ,  $i = 1, \dots, D$  in  $q_{i,t_u}$  intervals such that  $[\underline{x}_i, \bar{x}_i] = \bigcup_{j=1}^{q_{i,t_u}} [\underline{x}_{i,t_u}^j, \bar{x}_{i,t_u}^j]$  with  $\bar{x}_{i,t_u}^j = \underline{x}_{i,t_u}^{j+1}$ ,  $j = 1, \dots, q_i - 1$  and thus,  $\bigcap_{j=1}^{q_{i,t_u}} (\underline{x}_{i,t_u}^j, \bar{x}_{i,t_u}^j) = \emptyset$ . In this way, we can express  $\mathcal{X}$  at each time step  $t_u$  in the following way,

$$\mathcal{X} = \bigcup_{j=1}^{q_{1,t_u}} [\underline{x}_{1,t_u}^j, \bar{x}_{1,t_u}^j] \times \dots \times \bigcup_{j=1}^{q_{D,t_u}} [\underline{x}_{D,t_u}^j, \bar{x}_{D,t_u}^j]. \quad (5.16)$$

Hence, instead of one multivariate Chebyshev interpolation, we apply  $\prod_{i=1}^D q_{i,t_u}$  multivariate Chebyshev interpolations at  $t_u$  on smaller intervals on which the function is of higher regularity and in this sense behaves better. The striking advantage behind this idea is that, in general, less interpolation nodes are required on all of the small intervals together than for one interpolation over the whole interval. For notational ease in the following theorem, we express the space  $\mathcal{X}$  at  $t_u$  from (5.16) with  $Q_{t_u} := \prod_{i=1}^D q_{i,t_u}$  as,

$$\mathcal{X} = \bigcup_{l=1}^{Q_{t_u}} \mathcal{X}_{l,t_u}, \text{ with } \mathcal{X}_{l,t_u} = [\underline{x}_{1,t_u}^{h_{1,t_u}(l)}, \bar{x}_{1,t_u}^{h_{1,t_u}(l)}] \times \dots \times [\underline{x}_{D,t_u}^{h_{D,t_u}(l)}, \bar{x}_{D,t_u}^{h_{D,t_u}(l)}], \quad (5.17)$$

where  $h_{i,t_u}(l) \in \{1, \dots, q_{i,t_u}\}$  is such that  $h_{i,t_u}(1) = 1$  for  $i = 1, \dots, D$  and  $\forall l > 1$  and  $\forall u = 1, \dots, l - 1 \exists d \in \{1, \dots, D\} : h_{d,t_u}(l) \neq h_d(u)$ . First, we introduce Theorem 5.3.1 for the interpolation at one fixed time point, before the extension to the time stepping scheme is applied in Theorem 5.3.4.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

**Theorem 5.3.1.** *Let  $\mathcal{X} \ni X \mapsto g(x)$  be a real valued function and with  $Q := \prod_{i=1}^D q_i$  let*

$$\mathcal{X} = \bigcup_{l=1}^Q \mathcal{X}_l, \text{ with } \mathcal{X}_l = [\underline{x}_1^{h_1(l)}, \bar{x}_1^{h_1(l)}] \times \dots \times [\underline{x}_D^{h_D(l)}, \bar{x}_D^{h_D(l)}].$$

We will denote this partition of  $\mathcal{X}$  as  $\tilde{\mathcal{X}}^g = \{\mathcal{X}_1, \dots, \mathcal{X}_Q\}$ . For  $l = 1, \dots, Q$ , let  $g|_{\mathcal{X}_l}$  have an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}_l, \varrho_l)$  with parameter vector  $\varrho_l \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}_l, \varrho_l)} |g(x)| \leq M_l$ . Assume that the distortion error for all Chebyshev nodes  $x^{(k_{1,l}, \dots, k_{D,l})} \in \mathcal{X}_l$  is bounded by  $\bar{\varepsilon}_l$ . We define the interpolation resulting by the Chebyshev polynomial interpolation on each interval  $\mathcal{X}_i$ ,  $i = 1, \dots, Q$ , in the following way,

$$I_{\bar{N}^*}(g(\cdot))(x) = \sum_{l=1, \dots, Q} I_{N_l}(g|_{\mathcal{X}_l}(\cdot))(x) \mathbf{1}_{\mathcal{X}_l}(x).$$

Note that we indicate the interpolation of the function  $g$  on the complete domain  $\mathcal{X}$  with  $\bar{N}^*$  to depict the splitting concept. Then

$$\begin{aligned} \max_{x \in \mathcal{X}} |g(x) - I_{\bar{N}^*}(g(\cdot))(x)| &\leq \\ &\max_{1 \leq l \leq Q} \alpha(\varrho_l, N_l, D, M_l) + 2^D \bar{\varepsilon}_l \prod_{i=1}^D (N_i + 1). \end{aligned}$$

*Proof.* Let  $x \in \mathcal{X}$ . If  $x \in \mathcal{X}_l$ , then the function  $g_l := g|_{\mathcal{X}_l}$  satisfies the assumptions from Remark 4.4.1 with  $M_l$  and the generalized Bernstein ellipse  $B(\mathcal{X}_l, \varrho_l)$ . This yields

$$\begin{aligned} \max_{x \in \mathcal{X}_l} |g_l(x) - I_{N_l}(g_l(\cdot))(x)| & \\ &\leq \alpha(\varrho_l, N_l, D, M_l) + 2^D \bar{\varepsilon}_l \prod_{i=1}^D (N_{i,l} + 1). \end{aligned}$$

From  $\max_{x \in \mathcal{X}} |g(x) - I_{\bar{N}^*}(g(\cdot))(x)| \leq \max_{1 \leq l \leq Q} \max_{x \in \mathcal{X}_l} |g_l(x) - I_{N_l}(g_l(\cdot))(x)|$  the assertion follows directly.  $\square$

In the following, we use the result from Theorem 5.3.1 to determine the error applying Algorithm 3 considering the splitting of  $\mathcal{X}$  in several sub-domains, i.e. applying on each sub-domain a Chebyshev interpolation. Note that, in this case, we allow the splitting of the domain  $\mathcal{X}$  at each time step  $t_u$  into sub-domains as in (5.17) and that, between different time steps, the number of sub-domains may change. Additionally, we allow the use of different numbers of nodal points in the Chebyshev interpolations  $N_{l,t_u} = (N_{1,l,t_u}, \dots, N_{D,l,t_u})$  at each time step  $t_u$  and on each sub-interval. We introduce the additional notation  $V_{l,t_u} := V_{t_u}|_{\mathcal{X}_{l,t_u}}$ . First, we illustrate the dynamic Chebyshev scheme

with splitting in Algorithm 5 for the one dimensional case and two intervals. Then, the error convergence is investigated in Theorem 5.3.4.

**Remark 5.3.2.** *For the example of pricing American put options, at maturity  $T$ , the kink is obviously given at the money. Thus, we can split the interval into  $[x, \bar{x}] = [\underline{x}, K] \cup [K, \bar{x}]$  and, for the first time step, we preserve analyticity. In all the other time steps, we take the maximum of the payoff function and the conditional expectation as value of the value function. We know that, for instance, the put payoff is analytic on the domain  $[\underline{x}, K]$  and that the conditional expectation is analytic in general, see Gaß et al. (2016). Thus, by taking the maximum at each time step to determine the value of the value function, this maximum has a splitting point  $\tilde{x}$  in the domain  $[\underline{x}, K]$ . In this application, only one splitting point exists at each time  $t_u$  because the function  $g(t_u, x)$  and the conditional expectation are both monotonically decreasing functions in  $x$  and for  $x = \underline{x}$  the function  $g(t_u, \underline{x})$  is greater than the conditional expectation and for  $x = \bar{x}$  vice versa. For the second time step, we therefore split the interval into  $[x, \bar{x}] = [\underline{x}, \tilde{x}] \cup [\tilde{x}, \bar{x}]$ . More iteratively, at each time  $t_u$  we determine a splitting point  $\tilde{x}_{t_u}$  and split the domain  $[x, \bar{x}]$  into  $[\underline{x}, \tilde{x}_{t_u}] \cup [\tilde{x}_{t_u}, \bar{x}]$ . Of course, on both sub-domains Chebyshev interpolations are then applied. In our implementation, we use the Matlab function `fzero` to find at time step  $t_u$  the root of  $y_0 \mapsto g(t_u, y_0) - E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_0]$  in order to determine the splitting point. Even more tailored to the example of a put payoff is the application of a Newton's method as in Fang and Oosterlee (2009). A more general algorithm to find splitting points is presented in Pachon (2016). He implemented a splitting algorithm into the `chebfun` package that finds the splitting points of arbitrary functions and is not limited to one splitting point in a domain, which is already known in advance.*

**Remark 5.3.3.** *Although we can preserve analyticity via splitting, one has to keep in mind that regarding convergence, analyticity is only beneficial when the corresponding radius  $\rho$  of the corresponding Bernstein ellipse is noticeably larger than 1. In numerical test, see Gaß et al. (2016), we observed that the choice of the parameter domain is important. For very short-dated option, the radius of the Bernstein ellipse converged almost to 1 and thus, the error decay only occurred at a very slow rate.*

---

**Algorithm 5** Algorithm 3 with splitting at each time step for an American put option
 

---

- 1: **procedure** PRE-COMPUTATION STEP
  - 2:     Fix an interval  $[\underline{x}, \bar{x}]$  for the space of the underlying
  - 3:     Fix  $N_1$  and  $N_2$  as the number of nodal points of the Chebyshev interpolations on the domains  $[\underline{x}, \tilde{x}]$  and  $[\tilde{x}, \bar{x}]$
  - 4:     Determine nodal points  $x_1^k = \cos\left(\pi \frac{k}{N_1}\right)$  for  $k = 0, \dots, N_1$  and  $x_2^k = \cos\left(\pi \frac{k}{N_2}\right)$  for  $k = 0, \dots, N_2$
  - 5:     Set up time stepping  $0 = t_1, \dots, t_{n_T} = T$
  - 6: **procedure** INITIAL TIME  $T$
  - 7:     Set  $\tilde{x}_T = K$ , split interval  $[\underline{x}, \bar{x}]$  into  $[\underline{x}, \tilde{x}_T]$  and  $[\tilde{x}_T, \bar{x}]$
  - 8:     Set  $y_1^k = \tau_{[\underline{x}, \tilde{x}_T]}(x_1^k)$  and  $y_2^k = \tau_{[\tilde{x}_T, \bar{x}]}(x_2^k)$
  - 9:     Apply Chebyshev interpolation on both intervals
  - 10:      $P_T(y_1^k) = g(T, y_1^k)$ ,  $k = 1 : N_1$ , derive
  - 11:      $c_{1,j_1}(T) = \left(\frac{2^{\mathbb{1}\{0 < j_1 < N_1\}}}{N_1}\right) \sum_{k=0}^{N_1} P_T(y_1^k) \cos\left(j_1 \pi \frac{k}{N_1}\right)$
  - 12:      $P_T(y_2^k) = g(T, y_2^k)$ ,  $k = 1 : N_2$ , derive
  - 13:      $c_{2,j_2}(T) = \left(\frac{2^{\mathbb{1}\{0 < j_2 < N_2\}}}{N_2}\right) \sum_{k=0}^{N_2} P_T(y_2^k) \cos\left(j_2 \pi \frac{k}{N_2}\right)$
  - 14:     Obtain Chebyshev interpolation
  - 15:      $P_T(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(T) T_{j_1}(\tau_{[\underline{x}, \tilde{x}_T]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}_T]}(x) + \sum_{j_2=0}^{N_2} c_{2,j_2}(T) T_{j_2}(\tau_{[\tilde{x}_T, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}_T, \bar{x}]}(x)$
  - 16: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 17:     Define functions in dependence on  $y_0$  to determine the splitting point
  - 18:      $E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_0] =$
  - 19:          $\sum_{j_1=0}^{N_1} c_{1,j_1}(t_{u+1}) E[T_{j_1}(\tau_{[\underline{x}, \tilde{x}_{t_{u+1}}]}^{-1}(X_{t_{u+1}})) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}_{t_{u+1}}]}(X_{t_{u+1}}) | X_{t_u} = y_0]$
  - 20:          $+ \sum_{j_2=0}^{N_2} c_{2,j_2}(t_{u+1}) E[T_{j_2}(\tau_{[\tilde{x}_{t_{u+1}}, \bar{x}]}^{-1}(X_{t_{u+1}})) \cdot \mathbb{1}_{[\tilde{x}_{t_{u+1}}, \bar{x}]}(X_{t_{u+1}}) | X_{t_u} = y_0]$
  - 21:     Determine splitting point
  - 22:     Find  $\tilde{x}_{t_u}$  as root of  $y_0 \mapsto g(t_u, y_0) - E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_0]$
  - 23:     Split interval into  $[\underline{x}, \tilde{x}_{t_u}]$  and  $[\tilde{x}_{t_u}, \bar{x}]$
  - 24:     Set  $y_1^k = \tau_{[\underline{x}, \tilde{x}_{t_u}]}(x_1^k)$  and  $y_2^k = \tau_{[\tilde{x}_{t_u}, \bar{x}]}(x_2^k)$
  - 25:     Apply Chebyshev interpolation on both intervals
  - 26:      $P_{t_u}(y_1^k) = g(t_u, y_1^k)$ ,  $k = 1 : N_1$ , derive
  - 27:      $c_{1,j_1}(t_u) = \left(\frac{2^{\mathbb{1}\{0 < j_1 < N_1\}}}{N_1}\right) \sum_{k=0}^{N_1} P_{t_u}(y_1^k) \cos\left(j_1 \pi \frac{k}{N_1}\right)$
  - 28:      $P_{t_u}(y_2^k) = E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_2^k]$ ,  $k = 1 : N_2$ , derive
  - 29:      $c_{2,j_2}(t_u) = \left(\frac{2^{\mathbb{1}\{0 < j_2 < N_2\}}}{N_2}\right) \sum_{k=0}^{N_2} P_{t_u}(y_2^k) \cos\left(j_2 \pi \frac{k}{N_2}\right)$
  - 30:     Obtain Chebyshev interpolation
  - 31:      $P_{t_u}(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(t_u) T_{j_1}(\tau_{[\underline{x}, \tilde{x}_{t_u}]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}_{t_u}]}(x)$
  - 32:          $+ \sum_{j_2=0}^{N_2} c_{2,j_2}(t_u) T_{j_2}(\tau_{[\tilde{x}_{t_u}, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}_{t_u}, \bar{x}]}(x)$
  - 33: **procedure** DERIVING THE SOLUTION AT  $T=0$
  - 34:      $P_0(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(0) T_{j_1}(\tau_{[\underline{x}, \tilde{x}_1]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}_1]}(x) + \sum_{j_2=0}^{N_2} c_{2,j_2}(0) T_{j_2}(\tau_{[\tilde{x}_1, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}_1, \bar{x}]}(x)$
-

**Theorem 5.3.4.** *Let a Dynamic Programming Principle be given as in (5.3) and (5.4). Given a time stepping  $t = t_1 < \dots < t_{n_T} = T$ , let at each time step  $t_u$   $\mathcal{X}$  be given as in (5.17) such that for each  $l$  and  $t_u$ ,  $\mathcal{X}_{l,t_u} \ni x \mapsto V_{l,t_u}(x)$  is a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}_{l,t_u}, \varrho_{l,t_u})$  with parameter vector  $\varrho_{l,t_u} \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}_{l,t_u}, \varrho_{l,t_u})} |V_{l,t_u}(x)| \leq M_{l,t_u}$  for  $k = 1, \dots, n_T$ . Further, let  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be Lipschitz continuous with constant  $L_f$ .*

*Then, by applying Algorithm 5 on the splitted intervals, the resulting solution  $P_{t_u}(x)$  converges to the solution  $V_{t_u}(x)$  for  $N_{i,l,t_u} \rightarrow \infty$ ,  $i = 1, \dots, D$ . Furthermore, with  $C_{D,N} = 2^D \prod_{i=1}^D (N_i + 1)$  and  $Q_{t_u} = \prod_{i=1}^D q_{i,t_u}$ , the approximation error at time  $t_u$  is given by*

$$\max_{x \in \mathcal{X}} |V_{t_u}(x) - P_{t_u}(x)| \leq \varepsilon_{t_u} := \sum_{j=u}^{n_T} L_f^{j-u} \max_{1 \leq l \leq Q_{t_j}} C_{D,N_{l,t_u}}^{j-u} \alpha(\varrho_{l,t_j}, N_{l,t_j}, D, M_{l,t_j}), \quad (5.18)$$

and decays exponentially.

*Proof.* Similar to the proof of Theorem 5.2.2 at the initial time  $t_{n_T} = T$ , we apply Theorem 5.3.1 with  $\bar{\varepsilon}_l = 0$  resulting in,

$$\max_{x \in \mathcal{X}} |V_T(x) - P_{t_{n_T}}(x)| \leq \max_{l=1, \dots, Q_{t_{n_T}}} \alpha(\varrho_{l,t_{n_T}}, N_{l,t_{n_T}}, D, M_{l,t_{n_T}}).$$

Due to the Lipschitz-continuity of  $f$ , in the setting of Corollary 5.2.4 we know that with  $C_{D,N}$  as in (5.10),

$$\begin{aligned} \varepsilon_{t_{n_T-1}} &= \max_{1 \leq l \leq Q_{t_{n_T-1}}} \alpha(\varrho_{l,t_{n_T-1}}, N_{l,t_{n_T-1}}, D, M_{l,t_{n_T-1}}) \\ &\quad + L_f \max_{1 \leq l \leq Q_{t_{n_T}}} C_{D,N_{l,t_{n_T}}} \alpha(\varrho_{l,t_{n_T}}, N_{l,t_{n_T}}, D, M_{l,t_{n_T}}). \end{aligned}$$

With a similar induction as in the proof of Corollary 5.2.4, it follows for the error at  $t_u$ ,

$$\begin{aligned} \varepsilon_{t_k} &= \sum_{j=k}^{n_T} L_f^{j-k} \max_{1 \leq l \leq Q_{t_j}} C_{D,N_{l,t_j}}^{j-k} \alpha(\varrho_{l,t_j}, N_{l,t_j}, D, M_{l,t_j}) \\ &\leq C \bar{N}^D n_T \underline{\varrho}^{-N}, \end{aligned}$$

where

$$\begin{aligned} \bar{N} &= \max_{1 \leq j \leq n_T} \max_{1 \leq l \leq \prod_{i=1}^D q_{i,t_j}} \max_{1 \leq i \leq D} N_{i,l,t_j}, \quad \underline{N} = \min_{1 \leq j \leq n_T} \min_{1 \leq l \leq \prod_{i=1}^D q_{i,t_j}} \min_{1 \leq i \leq D} N_{i,l,t_j} \\ \underline{\varrho} &= \min_{1 \leq j \leq n_T} \min_{1 \leq l \leq \prod_{i=1}^D q_{i,t_j}} \min_{1 \leq i \leq D} \varrho_{i,l,t_j}. \end{aligned}$$

The error bound consists of a term increasing polynomially in  $\bar{N}$  and a term that decays exponentially in  $\underline{N}$ . Overall, due to  $\underline{\varrho} > 1$ , the exponential decaying behaviour dominates.  $\square$

In Algorithm 6, we present for an equidistant time stepping and a Markov process with stationary increments a simplified version of the splitting applied in Algorithm 5. Here, only at the strike of the option a splitting is applied. Thus, a pre-computation of the conditional expectations is possible.

---

**Algorithm 6** Fixed splitting at the strike  $K$ , simplified version of Algorithm 5 for equidistant time stepping and a Markov process with stationary increments

---

- 1: **procedure** PRE-COMPUTATION STEP
  - 2:     Fix an interval  $[\underline{x}, \bar{x}]$  for the space of the underlying, set  $\tilde{x} = \log(K)$
  - 3:     Fix  $N_1$  and  $N_2$  as number of nodal points of the Chebyshev interpolations on the domains  $[\underline{x}, \tilde{x}]$  and  $[\tilde{x}, \bar{x}]$
  - 4:     Determine nodal points  $x_1^k = \cos\left(\pi \frac{k}{N_1}\right)$  for  $k = 0, \dots, N_1$  and  $x_2^k = \cos\left(\pi \frac{k}{N_2}\right)$  for  $k = 0, \dots, N_2$ , set  $y_1^k = \tau_{[\underline{x}, \tilde{x}]}(x_1^k)$  and  $y_2^k = \tau_{[\tilde{x}, \bar{x}]}(x_2^k)$
  - 5:     Set up equidistant time stepping  $0 = t_1, \dots, t_{n_T} = T$
  - 6:     For  $j_1 = 1, \dots, N_1$  and for  $k = 0, \dots, N_1$ ,
  - 7:     compute  $\Gamma_{\Delta t}^1(T_{j_1})(y_1^k) := E[T_{j_1}(\tau_{[\underline{x}, \tilde{x}]}^{-1}(X_{t_{n_T}})) \mathbb{1}_{[\underline{x}, \tilde{x}]}(X_{t_{n_T}}) | X_{t_{n_T-1}} = y_1^k]$
  - 8:     For  $j_2 = 1, \dots, N_2$  and for  $k = 0, \dots, N_2$ ,
  - 9:     compute  $\Gamma_{\Delta t}^2(T_{j_2})(y_2^k) := E[T_{j_2}(\tau_{[\tilde{x}, \bar{x}]}^{-1}(X_{t_{n_T}})) \mathbb{1}_{[\tilde{x}, \bar{x}]}(X_{t_{n_T}}) | X_{t_{n_T-1}} = y_2^k]$
  - 10: **procedure** INITIAL TIME  $T$
  - 11:     Apply Chebyshev interpolation on both intervals,  $i = 1, 2$
  - 12:      $P_T(y_i^k) = g(T, y_i^k)$ ,  $k = 1 : N_i$ , derive
  - 13:      $c_{i,j_i}(T) = \left(\frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i}\right) \sum_{k=0}^{N_i} P_T(y_i^k) \cos\left(j_i \pi \frac{k}{N_i}\right)$
  - 14:     Obtain Chebyshev interpolation
  - 15:      $P_T(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(T) T_{j_1}(\tau_{[\underline{x}, \tilde{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}]}(x) + \sum_{j_2=0}^{N_2} c_{2,j_2}(T) T_{j_2}(\tau_{[\tilde{x}, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}, \bar{x}]}(x)$
  - 16: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 17:     Apply Chebyshev interpolation on both intervals,  $i = 1, 2$ ,
  - 18:      $P_{t_u}(y_i^k) = f(g(t_u, y_i^k), \sum_{j_i=0}^{N_i} c_{i,j_i}(t_{u+1}) \Gamma_{t_u, t_{u+1}}^1(T_{j_i})(y_i^k))$ ,  $k = 1 : N_i$ , derive
  - 19:      $c_{i,j_i}(t_u) = \left(\frac{2^{\mathbb{1}_{\{0 < j_i < N_i\}}}}{N_i}\right) \sum_{k=0}^{N_i} P_{t_u}(y_i^k) \cos\left(j_i \pi \frac{k}{N_i}\right)$
  - 20:     Obtain Chebyshev interpolation
  - 21:      $P_{t_u}(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(t_u) T_{j_1}(\tau_{[\underline{x}, \tilde{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}]}(x) + \sum_{j_2=0}^{N_2} c_{2,j_2}(t_u) T_{j_2}(\tau_{[\tilde{x}, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}, \bar{x}]}(x)$
  - 22: **procedure** DERIVING THE SOLUTION AT  $T=0$
  - 23:      $P_0(x) = \sum_{j_1=0}^{N_1} c_{1,j_1}(0) T_{j_1}(\tau_{[\underline{x}, \tilde{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}]}(x) + \sum_{j_2=0}^{N_2} c_{2,j_2}(0) T_{j_2}(\tau_{[\tilde{x}, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}, \bar{x}]}(x)$
-

### 5.3.2 Mollifier to the Function $g(t,x)$

In the previous section, we have presented the idea of splitting the domain in several sub-domains, if the mapping  $\mathcal{X} \ni x \mapsto V_{t_u}(x)$  is not analytic on the whole domain. However, if in after each time step, a different splitting of the domain is necessary, the key idea of pre-computing the conditional expectations once as illustrated in Algorithm 4 is not possible. Different splitting of the domain leads to different integration bounds for the conditional expectations. Therefore, at each time step, the conditional expectations have to be re-evaluated. This may reduce the run-time, especially if the derivation of the conditional expectation is rather costly. In this section, we present a different approach. Often, the necessity of splitting results from the fact that the function  $g$  is not analytic. For the DPP (5.3) and (5.4), we illustrate this idea with the example of pricing options. In option pricing, there is usually the kink in the hockey-stick-like payoff function. Our idea is to slightly smooth the payoff function  $g$ . In the case of an American put option, we will replace the payoff function  $(T, x) \mapsto g(T, x) = \max\{K - e^x, 0\}$  with a function  $(T, x) \mapsto \tilde{g}(T, x)$ . This function  $\tilde{g}(T, x)$  a real valued function that has an analytic extension to a generalized Bernstein ellipse with an appropriate parameter vector  $\varrho \in (1, \infty)^D$ . Then, we apply Algorithm 4. Here, our idea is to replace the payoff function  $g(T, x)$  with the Black&Scholes price for a European put with a maturity matching the first time step. In the following theorem, we analyze how the error  $\varepsilon_g := \max_{x \in \mathcal{X}} \{g(T, x) - \tilde{g}(T, x)\}$  propagates.

**Theorem 5.3.5.** *Let the setting be as in Theorem 5.2.2. Further, let  $f$  be Lipschitz continuous with constant  $L_f$ . Let the function  $g(t, x)$  be approximated by a real valued function  $\tilde{g}(t, x)$  that has an analytic extension to a generalized Bernstein ellipse with an appropriate parameter vector  $\varrho \in (1, \infty)^D$  and let  $\varepsilon_g := \max_{x \in \mathcal{X}} \{g(T, x) - \tilde{g}(T, x)\}$ . Then, the approximation  $P_{t_u}$  from Algorithm 3 converges to the solution  $V_{t_u}$ , if  $\varepsilon_g \rightarrow 0$ , and the error is bounded by*

$$\varepsilon_{t_u} = \sum_{j=u}^{n_T} C_{D,N}^{j-u} L_f^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}) + \sum_{j=u}^{n_T} C_{D,N}^{j-u+1} L_f^{j-u} \varepsilon_g. \quad (5.19)$$

*Proof.* As in the proof of Theorem 5.2.2, we first take a look at the error at the first time step. Here, we have additionally the issue that we have a distortion at the nodal points of the Chebyshev interpolation, which is bounded by  $\varepsilon_g$ . Thus, on the initial time step, we observe the following error with  $C_{D,N} = 2^D \prod_{i=1}^D (N_i + 1)$ ,

$$\max_{x \in \mathcal{X}} |V_T(x) - P_{t_{n_T}}(x)| \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) + C_{D,N} \cdot \varepsilon_g.$$

The function  $f$  is Lipschitz continuous, thus we can switch at this point into the proof

of Corollary 5.2.4. For the error at the nodal points at time  $t_u$ ,  $F_{t_u}$ , it holds,

$$\begin{aligned}
 F_{t_u} &= \max_k |V_{t_u}(\tau_{\mathcal{X}}(x^k)) - P_{t_u}(\tau_{\mathcal{X}}(x^k))| \\
 &\leq \max_k L_f \left( \left| g(t_u, \tau_{\mathcal{X}}(x^k)) - \tilde{g}(t_u, \tau_{\mathcal{X}}(x^k)) \right| \right. \\
 &\quad \left. + \left| E[P_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] - E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\
 &\leq \max_k L_f \left( \varepsilon_g + \left| E[P_{t_{u+1}}(X_{t_{u+1}}) - V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\
 &\leq \max_k L_f \left( \varepsilon_g + |E[\varepsilon_{t_{u+1}}|X_{t_u} = \tau_{\mathcal{X}}(x^k)]| \right) \\
 &= L_f \cdot (\varepsilon_g + \varepsilon_{t_{u+1}}).
 \end{aligned}$$

This yields,

$$\varepsilon_{t_u} = \alpha(\varrho_{t_u}, N, D, M_{t_u}) + C_{D,N} L_f (\varepsilon_g + \varepsilon_{t_{u+1}}).$$

A similar induction as in the proof of Corollary 5.2.4, (5.19) follows. Here, the additional part incorporating the  $\varepsilon_g$  terms has to be added accordingly. For  $\varepsilon_g \rightarrow 0$ , the convergence is as in Corollary 5.2.4.  $\square$

## 5.4 Alternative Approximation of General Moments in the Pre-Computation

In this section, we will apply a slightly different approach to the dynamic Chebyshev procedure. Consider a one dimensional setting, let  $\mathcal{X} = [\underline{x}, \bar{x}]$ . Starting with the DPP (5.3) and (5.4), we again apply at time  $t_{u+1}$  a Chebyshev polynomial interpolation to the value function  $V_{t_{u+1}}$ , which then will be used at time  $t_u$  to derive the conditional expectation,

$$\begin{aligned}
 V_{t_u}(x) &= \max \left\{ g(t_u, x), E \left[ \sum_{j=0}^N c_j(t_{u+1}) T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(X_{t_{u+1}})) \mathbb{1}_{X_{t_{u+1}} \in [\underline{x}, \bar{x}]} | X_{t_u} = x \right] \right\} \\
 &= \max \left\{ g(t_u, x), \sum_{j=0}^N c_j(t_{u+1}) E \left[ T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(X_{t_{u+1}})) \mathbb{1}_{X_{t_{u+1}} \in [\underline{x}, \bar{x}]} | X_{t_u} = x \right] \right\}.
 \end{aligned}$$

At this point, we focus on the conditional expectation, which we will evaluate at  $x_0 \in \mathcal{X}$ ,

$$E \left[ T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(X_{t_{u+1}})) \mathbb{1}_{X_{t_{u+1}} \in [\underline{x}, \bar{x}]} | X_{t_u} = x_0 \right] = \int_{\underline{x}-x_0}^{\bar{x}-x_0} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) pdf_{X_{t_{u+1}-t_u}}(x) dx,$$

where  $pdf_{X_{t_{u+1}-t_u}}(x)$  denotes the probability density function of the process  $X$  from  $t_u$  to  $t_{u+1}$ . Here, we apply a second Chebyshev polynomial interpolation to the probability

## 5 Dynamic Programming Framework with Chebyshev Interpolation

density function on  $[\underline{x} - x_0, \bar{x} - x_0]$ , i.e.

$$pdf_{X_{t_{u+1}-t_u}}(x) \approx \sum_{i=0}^M \gamma_i(x_0) T_i(\tau_{[\underline{x}-x_0, \bar{x}-x_0]}^{-1}(x)).$$

Usually, the probability density function has infinite support. However, due to the limits of the integration, we only need to approximate on the bounded domain  $[\underline{x} - x_0, \bar{x} - x_0]$  and therefore, no additional error by truncating the domain from  $(-\infty, \infty)$  to  $[\underline{x} - x_0, \bar{x} - x_0]$  is made.

This yields

$$\begin{aligned} E \left[ T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(X_{t_{u+1}})) \mathbb{1}_{X_{t_{u+1}} \in [\underline{x}, \bar{x}] | X_{t_u} = x_0} \right] \\ \approx \int_{\underline{x}-x_0}^{\bar{x}-x_0} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) \sum_{i=0}^M \gamma_i(x_0) T_i(\tau_{[\underline{x}-x_0, \bar{x}-x_0]}^{-1}(x)) dx \\ = \int_{\underline{x}-x_0}^{\bar{x}-x_0} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) \sum_{i=0}^M \gamma_i(x_0) T_i(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) dx \\ = \sum_{i=0}^M \gamma_i(x_0) \int_{\underline{x}-x_0}^{\bar{x}-x_0} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) T_i(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) dx. \end{aligned}$$

Now, we apply the variable transformation  $z = \tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)$  and this results in,

$$\begin{aligned} \sum_{i=0}^M \gamma_i(x_0) \int_{\underline{x}-x_0}^{\bar{x}-x_0} T_j(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) T_i(\tau_{[\underline{x}, \bar{x}]}^{-1}(x + x_0)) dx \\ = \sum_{i=0}^M \gamma_i(x_0) \int_{-1}^1 T_j(z) T_i(z) \frac{\bar{x} - \underline{x}}{2} dz \\ = \frac{\bar{x} - \underline{x}}{2} \sum_{i=0}^M \gamma_i(x_0) \int_{-1}^1 T_j(z) T_i(z) dz. \end{aligned} \tag{5.20}$$

The integral over a product of Chebyshev polynomials is given by Rivlin (1990),

$$I_{j,i} := \int_{-1}^1 T_j(z) T_i(z) dz = \begin{cases} 0 & \text{for } |j - i| \text{ odd} \\ - \left[ \frac{1}{(i+j)^2 - 1} + \frac{1}{(i-j)^2 - 1} \right] & \text{for } |j - i| \text{ even.} \end{cases} \tag{5.21}$$

Combining the results yields,

$$V_{t_u}(x) = \max \left\{ g(t_u, x), \frac{(\bar{x} - \underline{x})}{2} \sum_{j=0}^N c_j(t_{u+1}) \sum_{i=0}^M \gamma_i(x) I_{j,i} \right\}.$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

Interestingly, the conditional expectations only have to be derived at the nodal points  $x^k$ . This means that for each of the  $N + 1$  nodal points a one dimensional Chebyshev interpolation is applied to  $pdf_{X_{t_{u+1}-t_u}}(x)$  on  $[\underline{x} - x_0, \bar{x} - x_0]$ . Note that the quantities  $\sum_{i=0}^M \gamma_i(x^k) I_{j,i}$  can be derived in the pre-computation step for the nodal points  $x^k$  as long as no splitting procedure is applied in the time stepping.

Let us now switch to the multivariate setting, let  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$ . For  $x \in \mathcal{X}$  consider the DPP,

$$V_T(x) = g(T, x) \tag{5.22}$$

$$V_{t_u}(x) = \max\{g(t_u, x), E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = x]\}, \tag{5.23}$$

we again apply at time  $t_{u+1}$  a  $D$ -dimensional Chebyshev polynomial interpolation to the value function  $V_{t_{u+1}}$ . Then, at  $t_u$  we get for the conditional expectations with  $j = (j_1, \dots, j_D)$ ,

$$\begin{aligned} V_{t_u}(x) &= \max \left\{ g(t_u, x), E \left[ \sum_j c_j(t_{u+1}) \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(X_{t_{u+1}, i})) \mathbf{1}_{X_{t_{u+1}, i} \in [\underline{x}_i, \bar{x}_i]} | X_{t_u} = x \right] \right\} \\ &= \max \left\{ g(t_u, x), \sum_j c_j(t_{u+1}) E \left[ \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(X_{t_{u+1}, i})) \mathbf{1}_{X_{t_{u+1}, i} \in [\underline{x}_i, \bar{x}_i]} | X_{t_u} = x \right] \right\}. \end{aligned}$$

Similar to the one dimensional case, we focus on deriving the conditional expectation for  $x_0 = (x_{01}, \dots, x_{0D}) \in \mathcal{X}$ .

$$\begin{aligned} E \left[ \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(X_{t_{u+1}, i})) \mathbf{1}_{X_{t_{u+1}, i} \in [\underline{x}_i, \bar{x}_i]} | X_{t_u} = x_0 \right] \\ = \int_{\underline{x}_1 - x_{01}}^{\bar{x}_1 - x_{01}} \dots \int_{\underline{x}_D - x_{0D}}^{\bar{x}_D - x_{0D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) pdf_{X_{t_{u+1}-t_u}}(x) dx_D \dots dx_1, \end{aligned}$$

where here  $pdf_{X_{t_{u+1}-t_u}}(x)$  denotes the probability density function of the multivariate process  $X$  from  $t_u$  to  $t_{u+1}$ . As a next step, a second  $D$ -dimensional Chebyshev interpolation is applied to the probability density function on  $[\underline{x}_1 - x_{01}, \bar{x}_1 - x_{01}] \times \dots \times [\underline{x}_D - x_{0D}, \bar{x}_D - x_{0D}]$ , with  $q = (q_1, \dots, q_D)$ ,

$$pdf_{X_{t_{u+1}-t_u}}(x) \approx \sum_q \gamma_q(x_0) \prod_{i=1}^D T_{q_i}(\tau_{[\underline{x}_i - x_{0i}, \bar{x}_i - x_{0i}]}^{-1}(x_i)).$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

Hence, the integral can be approximated in the following way,

$$\begin{aligned}
& \int_{\underline{x}_1-x_{01}}^{\bar{x}_1-x_{01}} \cdots \int_{\underline{x}_D-x_{0D}}^{\bar{x}_D-x_{0D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) pdf_{X_{t_{u+1}-t_u}}(x) dx \\
& \approx \int_{\underline{x}_1-x_{01}}^{\bar{x}_1-x_{01}} \cdots \int_{\underline{x}_D-x_{0D}}^{\bar{x}_D-x_{0D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) \sum_q \gamma_q(x_0) \prod_{i=1}^D T_{q_i}(\tau_{[\underline{x}_i-x_{0i}, \bar{x}_i-x_{0i}]}^{-1}(x_i)) dx \\
& = \sum_q \gamma_q(x_0) \int_{\underline{x}_1-x_{01}}^{\bar{x}_1-x_{01}} \cdots \int_{\underline{x}_D-x_{0D}}^{\bar{x}_D-x_{0D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) \prod_{i=1}^D T_{q_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) dx
\end{aligned}$$

Analogously to the one dimensional case, we apply the variable transformation to  $z = (z_1, \dots, z_D)$  with  $z_i = \tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})$ , for  $i = 1, \dots, D$ . This yields, by reducing the  $D$ -dimensional integral to  $D$  one dimensional, integrals of the form  $\int_{-1}^1 T_{j_i}(z_i) T_{q_i}(z_i) dz_i$

$$\begin{aligned}
& \sum_q \gamma_q(x_0) \int_{\underline{x}_1-x_{01}}^{\bar{x}_1-x_{01}} \cdots \int_{\underline{x}_D-x_{0D}}^{\bar{x}_D-x_{0D}} \prod_{i=1}^D T_{j_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) \prod_{i=1}^D T_{q_i}(\tau_{[\underline{x}_i, \bar{x}_i]}^{-1}(x_i + x_{0i})) dx \\
& = \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D} \sum_q \gamma_q(x_0) \int_{\underline{x}_1-x_{01}}^{\bar{x}_1-x_{01}} \cdots \int_{\underline{x}_D-x_{0D}}^{\bar{x}_D-x_{0D}} \prod_{i=1}^D T_{j_i}(z_i) \prod_{i=1}^D T_{q_i}(z_i) dz_D \cdots dz_1 \\
& = \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D} \sum_q \gamma_q(x_0) \int_{-1}^1 \cdots \int_{-1}^1 \prod_{i=1}^{D-1} T_{j_i}(z_i) \prod_{i=1}^{D-1} T_{q_i}(z_i) \cdot \\
& \quad \left[ \int_{-1}^1 T_{j_D}(z_D) T_{q_D}(z_D) dz_D \right] dz_{D-1} \cdots dz_1 \\
& = \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D} \sum_q \gamma_q(x_0) \cdot \int_{-1}^1 \cdots \int_{-1}^1 \prod_{i=1}^{D-1} T_{j_i}(z_i) \prod_{i=1}^{D-1} T_{q_i}(z_i) [I_{j_D, q_D}] dz_{D-1} \cdots dz_1 \\
& = \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D} \sum_q \gamma_q(x_0) I_{j_D, q_D} \cdot \\
& \quad \int_{-1}^1 \cdots \int_{-1}^1 \prod_{i=1}^{D-2} T_{j_i}(z_i) \prod_{i=1}^{D-2} T_{q_i}(z_i) \left[ \int_{-1}^1 T_{j_{D-1}}(z_{D-1}) T_{q_{D-1}}(z_{D-1}) dz_{D-1} \right] dz_{D-2} \cdots dz_1 \\
& = \cdots = \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D} \sum_q \gamma_q(x_0) \prod_{i=1}^D I_{j_i, q_i}.
\end{aligned}$$

Combining the results yields, with  $K(\mathcal{X}) := \frac{\prod_{i=1}^D \bar{x}_i - \underline{x}_i}{2^D}$ ,

$$V_{t_u}(x) = \max \left\{ g(t_u, x), K(\mathcal{X}) \sum_j c_j(t_{u+1}) \sum_q \gamma_q(x) \prod_{i=1}^D I_{j_i, q_i} \right\}.$$

---

**Algorithm 7** Dynamic Chebyshev with alternative approximation of general moments in the Pre-Computation

---

- 1: **procedure** PRE-COMPUTATION STEP
  - 2: Fix an interval  $\mathcal{X} = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_D, \bar{x}_D]$  for the interpolation
  - 3: Fix  $N_1, \dots, N_D$  as the number of nodal points of the Chebyshev interpolation in each dimension
  - 4: Determine nodal points  $x^k = (x_{k_1}, \dots, x_{k_D})$  with  $x^{k_i} = \cos\left(\pi \frac{k_i}{N_i}\right)$  for  $k_i = 0, \dots, N_i$  and  $i = 1, \dots, D$ , set  $y^k = \tau_{\mathcal{X}}(x^k)$
  - 5: Denote with  $T_j$  the Chebyshev polynomial for all  $j \in J$  with  $j = (j_1, \dots, j_D)$ ,  $j_l = 0, \dots, N_l$
  - 6: Add in the pre-computation step of Algorithm 3 the Chebyshev interpolation of the density function
  - 7:  $pdf_{X_{t_{u+1}-t_u}}(x) \approx \sum_{i=0}^M \gamma_i(x_0) T_i(\tau_{[\underline{x}-x_0, \bar{x}-x_0]}^{-1}(x))$
  - 8: Set up time stepping  $0 = t_1, \dots, t_{n_T} = T$
  - 9: For all  $j \in J$ , for all  $t_u$ ,  $u = 0, \dots, n_T - 1$ , for all  $k = (k_1, \dots, k_D)$ ,  $k_l = 0, \dots, N_l$
  - 10: Compute  $\Gamma_{t_u, t_{u+1}}(T_j)(y^k) := E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}})) | X_{t_u} = y^k]$  as in (5.20), (5.21).
  - 11: Apply the procedures *Initial Time T*, *Iterative Time Stepping* and *Deriving the Solution* as in Algorithm 3.
- 

**Theorem 5.4.1.** *Let a Dynamic Programming Principle be given as in (5.22) and (5.23). Given a time stepping  $t = t_1 < \dots < t_{n_T} = T$ , let  $\mathcal{X} \ni x \mapsto V_{t_u}(x)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho_{t_u})$  with parameter vector  $\varrho_{t_u} \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}, \varrho_{t_u})} |V_{t_u}(x)| \leq M_{t_u}$  for  $k = 1, \dots, n_T$ . Additionally, let for all nodal points  $x^k$  the probability density function  $pdf_{X_{t_{u+1}-t_u}}(x)$  on  $\mathcal{X}(x^k) := [\underline{x}_1 - x_{k_1}, \bar{x}_1 - x_{k_1}] \times \dots \times [\underline{x}_D - x_{k_D}, \bar{x}_D - x_{k_D}]$  have an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}(x^k), \varrho_{pdf})$  with parameter vector  $\varrho_{pdf} \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}(x^k), \varrho_{pdf})} |pdf_{X_{t_{u+1}-t_u}}(x)| \leq M_{pdf}$ . Then, by applying Algorithm 7, the resulting solution  $P_{t_u}(x)$  converges to the solution  $V_{t_u}(x)$  for  $N_i \rightarrow \infty$ ,  $i = 1, \dots, D$  and the error is bounded by*

$$\varepsilon_{t_u} = \sum_{j=u}^{n_T} C_{D,N}^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}) + \sum_{j=u+1}^{n_T} C_{D,N}^{j-u} \left( M_{t_j} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i) \right).$$

where  $\alpha(\varrho_{t_j}, N, D, M_{t_j})$  as in (5.9),  $C_{D,N} = 2^D$  as in (5.10).

*Proof.* Analogously to the proof in Theorem 5.2.2, at the initial time step  $t_{n_T} = T$ ,  $P_{t_{n_T}}$  is the Chebyshev interpolation of  $g(T, x) = V_T(x)$ . From Theorem 4.2.10, we obtain that the interpolation error is bounded by

$$\max_{x \in \mathcal{X}} |V_T(x) - P_{t_{n_T}}(x)| \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}),$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

where  $\alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}})$  as in (5.9). Now we consider the step from  $t_{n_T} \rightarrow t_{n_T-1}$ . At this step, we approximate the function  $V_{t_{n_T-1}}$  with  $P_{t_{n_T-1}}$ . Here, we have to consider the distortion at the nodal points. Unlike as in the proof in Theorem 5.2.2, the distortion at the nodal points does not only result from the previous Chebyshev interpolation, but additionally from the Chebyshev interpolation of the probability density function. This proof is directly shown for  $f(x, y) = \max(x, y)$  and thus, the function is Lipschitz with constant  $L_f = 1$ . This allows us to directly go into the proof of Corollary 5.2.4. At  $t_{n_T-1}$ , we investigate the error at the nodal points. By estimating this error, the additional Chebyshev interpolation of the probability density function has to be considered. For notational ease in the proof, we assume that the true probability density function  $x \mapsto pdf_{X_{t_{u+1}-t_u}}(x)$  is given as,

$$pdf_{X_{t_{u+1}-t_u}}(x) = pdf_{X_{t_{u+1}-t_u}}^I(x) + pdf_{X_{t_{u+1}-t_u}}^\varepsilon(x),$$

where  $pdf_{X_{t_{u+1}-t_u}}^I(x)$  represents the interpolated probability density functions and  $pdf_{X_{t_{u+1}-t_u}}^\varepsilon(x)$  the error term. In the following, we indicate with  $E_{pdf_{X_{t_{u+1}-t_u}}^*}[Y]$  the expectation of a random variable with probability density function  $pdf_{X_{t_{u+1}-t_u}}^*$ , i.e.  $E_{pdf_{X_{t_{u+1}-t_u}}^*}[Y] = \int_{\mathcal{X}} y \cdot pdf_{X_{t_{u+1}-t_u}}^*(y) dy$ . For the error at the nodal points this yields,

$$\begin{aligned} & \max_k |V_{t_{n_T-1}}(\tau_{\mathcal{X}}(x^k)) - P_{t_{n_T-1}}(\tau_{\mathcal{X}}(x^k))| \\ & \leq \max_k \left( \left| E_{pdf_{X_{t_{u+1}-t_u}}^I} [P_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \right. \right. \\ & \quad \left. \left. - E_{pdf_{X_{t_{u+1}-t_u}}} [V_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \right| \right). \end{aligned}$$

Now, with  $x = (x_1, \dots, x_D)$ , we get,

$$\begin{aligned} & \max_k \left( \left| E_{pdf_{X_{t_{u+1}-t_u}}^I} [P_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \right. \right. \\ & \quad \left. \left. - E_{pdf_{X_{t_{u+1}-t_u}}} [V_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T-1}} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\ & = \max_k \left| \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} P_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k)) pdf_{X_{t_{u+1}-t_u}}^I(x) dx \right. \\ & \quad \left. - \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} V_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k)) pdf_{X_{t_{u+1}-t_u}}(x) dx \right| \end{aligned}$$

Plugging-in the approximation of the density function yields,

$$\begin{aligned}
 &= \max_k \left| \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} P_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k)) \left( pdf_{X_{t_{u+1}-t_u}}(x) - pdf_{\tilde{X}_{t_{u+1}-t_u}}^\varepsilon(x) \right) dx \right. \\
 &\quad \left. - \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} V_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k)) pdf_{X_{t_{u+1}-t_u}}(x) dx \right| \\
 &\leq \max_k \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} |V_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k)) - P_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k))| pdf_{X_{t_{u+1}-t_u}}(x) dx \\
 &\quad + \int_{\underline{x}_1 - x_{k1}}^{\bar{x}_1 - x_{k1}} \dots \int_{\underline{x}_D - x_{kD}}^{\bar{x}_D - x_{kD}} |P_{t_{n_T}}(x + \tau_{\mathcal{X}}(x^k))| pdf_{\tilde{X}_{t_{u+1}-t_u}}^\varepsilon(x) dx
 \end{aligned}$$

Now, we know, as shown in the proof of Corollary 5.2.4, that if the error  $\max_{x \in \mathcal{X}} |V_{t_{n_T}}(x) - P_{t_{n_T}}(x)|$  is bounded by  $\alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}})$ , then the expected error using the real probability density function is bounded by exactly  $\alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}})$ . Additionally, we know that  $|P_{t_{n_T}}(x + x_j)| \leq M_{t_{n_T}}$  and that  $|pdf_{\tilde{X}_{t_{u+1}-t_u}}^\varepsilon(x)| \leq \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf})$ . This yields for our estimation,

$$\begin{aligned}
 &\max_k \left( \left| E_{pdf_{X_{t_{u+1}-t_u}}} [P_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T}-1} = \tau_{\mathcal{X}}(x^k)] - \right. \right. \\
 &\quad \left. \left. E_{pdf_{X_{t_{u+1}-t_u}}} [V_{t_{n_T}}(X_{t_{n_T}}) | X_{t_{n_T}-1} = \tau_{\mathcal{X}}(x^k)] \right| \right) \\
 &\quad \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) + M_{t_{n_T}} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i).
 \end{aligned}$$

With  $C_{D,N}$  as in (5.10) this then yields, analogously to the proof of Corollary 5.2.4, for the error at  $t_{n_T-1}$ ,

$$\begin{aligned}
 &\max_{x \in \mathcal{X}} |V_{t_{n_T-1}}(x) - P_{t_{n_T-1}}(x)| \leq \varepsilon_{t_{n_T-1}} := \alpha(\varrho_{t_{n_T-1}}, N, D, M_{t_{n_T-1}}) + \\
 &\quad C_{D,N} \left( \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) + M_{t_{n_T}} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i) \right).
 \end{aligned}$$

Obviously, this directly leads to the recursive scheme for the error bound at  $t_u$ ,

$$\begin{aligned}
 \varepsilon_{t_u} &= \alpha(\varrho_{t_u}, N, D, M_{t_u}) + \\
 &\quad C_{D,N} \left( \varepsilon_{t_{u+1}} + M_{t_{u+1}} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i) \right).
 \end{aligned}$$

Now, applying a similar induction as in the proof of Corollary 5.2.4 directly yields,

$$\varepsilon_{t_u} = \sum_{j=u}^{n_T} C_{D,N}^{j-u} \alpha(\varrho_{t_j}, N, D, M_{t_j}) + \sum_{j=u+1}^{n_T} C_{D,N}^{j-u} \left( M_{t_j} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i) \right).$$

□

**Remark 5.4.2.** In Theorem 5.4.1, we directly assumed  $f(x, y) = \max(x, y)$ . In this case, we could work with Lipschitz constant  $L_f = 1$ . By assuming a general function  $f$  in the DPP (5.3), (5.4), with Lipschitz constant  $L_f$ , the error bound in Theorem 5.4.1 gets adjusted to,

$$\begin{aligned} \varepsilon_{t_k} = & \sum_{j=k}^{n_T} L_f^{j-k} C_{D,N}^{j-k} \alpha(\varrho_{t_j}, N, D, M_{t_j}) \\ & + \sum_{j=k+1}^{n_T} L_f^{j-k} C_{D,N}^{j-k} \left( M_{t_j} \alpha(\varrho_{pdf}, N_{pdf}, D, M_{pdf}) \prod_{i=1}^D (\bar{x}_i - \underline{x}_i) \right). \end{aligned}$$

**Remark 5.4.3.** This approach is related to the idea suggested in Pachon (2016). In Pachon (2016), European options with arbitrary payoffs are investigated. Contrary to the plain vanilla options, which only have a kink at the strike, these arbitrary payoff functions can have several kinks. To derive the European option price, Pachon (2016) first splits the domain in several sub-intervals such that the arbitrary payoff function is smooth on each sub-interval. Second, he interpolates the arbitrary payoff function on each sub-interval with a Chebyshev interpolation. Then he resulting integrals of the form  $\int_{-1}^1 T_j(x) f(x) dx$  are solved by applying Clenshaw-Curtis quadrature. Our approach here, on the contrary, suggests a Chebyshev interpolation of the function  $f$  and then the resulting integral is known explicitly. The error of the Clenshaw-Curtis quadrature of the integral  $\int_{-1}^1 T_j(x) f(x) dx$  is basically connected to the error of approximating the integrated function  $T_j(x) f(x)$  by Chebyshev interpolation. Thus, replacing our approach to compute  $\int_{-1}^1 T_j(x) f(x) dx$  by applying the Clenshaw-Curtis quadrature, we get, see Theorem 4.2.10,

$$\max_{x \in [-1, 1]} |T_j(x) f(x) - I_N(T_j f)(x)| \leq V \min \left\{ \frac{a(\varrho, N, D)}{V}, \frac{b(\varrho, N, D)}{V} \right\},$$

where  $V$  is the surprium of the function  $T_j f$  on the Bernstein ellipse. Let's assume that  $f$  is bounded by  $\bar{V}$  on the Bernstein ellipse. From Bernstein's inequality, see Trefethen (2013), it follows that the Chebyshev polynomial  $T_j(x)$  is bounded by  $\varrho^j$ . Hence, we estimate  $V \leq \varrho^j \bar{V}$  and, thus, the convergence rate in  $\varrho^{-N}$  is reduced to  $\varrho^{-(N-j)}$ . With the scheme introduced in Algorithm 7, we interpolate only the function  $f$  and, hence, we get for the error,

$$\max_{x \in [-1, 1]} |T_j(x)| \cdot |f(x) - I_N(f)(x)| \leq \bar{V} \min \left\{ \frac{a(\varrho, N, D)}{\bar{V}}, \frac{b(\varrho, N, D)}{\bar{V}} \right\}.$$

In this case, the error converges with a rate of  $\varrho^{-N}$ .

A comparison of both approaches is connected to the cost of evaluating the function  $f$ . If this function is given explicitly in closed-form, then the Clenshaw-Curtis quadrature of Pachon (2016) is faster than our three step approach of deriving the coefficients of

the interpolation, assembling the values of (5.21) and combining these derived values to the value of the integral. However, if an evaluation of the function  $f$  is costly, e.g. the function has to be evaluated via numerically-demanding techniques, then our approach, requiring less evaluations of the function  $f$ , becomes beneficial.

## 5.5 Combination of Empirical Interpolation with Dynamic Chebyshev

In the previous sections, we applied Chebyshev interpolation to the dynamic programming principle to price an American option. So far, we interpolate in the space of the underlying, i.e. in  $\mathcal{X}$ . This allows us to derive the American option price for several underlying values, but for a fixed parameter setting. In this case, other pricing techniques tend to be faster. Now, we are going to make use of the parameter dependency to construct the option price at the initial time  $t_0$  in dependence on the option parameters. The key idea is to price the according option not only for different underlying values, but also for different parameter settings. Naturally, deriving at any time the conditional expectation  $E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = x_0]$ , with  $x_0 \in \mathcal{X}$ , is only the derivation of an integral, i.e.

$$E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = x_0] = \int_{\mathcal{X}} V_{t_{u+1}}(x) \cdot \mathbb{1}_{(x-x_0) \in \mathcal{X}}(x) \cdot pdf_{X_{t_{u+1}-t_u}}(x) dx.$$

The conditional probability density function  $pdf_{X_{t_{u+1}-t_u}}$ , capturing  $x_0$ , depends on several parameters here. Although we apply Chebyshev interpolation to the value function, in this section we do not actually apply a Chebyshev interpolation to the probability density function, too. As the concluding remarks of Chapter 4 have indicated, for a potentially higher-dimensional parameter space our tensorized Chebyshev polynomial interpolation is rather slow and requires the evaluation of the density function on a tensorized grid. In this case, we follow a different approach. We use empirical interpolation, see Barrault et al. (2004), for the parameters in the density functions. For a one dimensional underlying, this method, including error analysis, has been presented in Gaß and Glau (2015) for parametric integration and in Gaß et al. (2016) for option pricing. On this basis, we approximate the on the parameter  $\mu \in \mathbb{R}^d$  depending probability density function  $pdf_{X_{t_{u+1}-t_u}}$  in the following way,

$$pdf_{X_{t_{u+1}-t_u}}^{\mu}(x) \approx \sum_{\kappa=1}^{\mathcal{M}} \Theta_{\kappa}(\mu, x_{\kappa}^*) q_{\kappa}(x), \quad (5.24)$$

where the points  $x_{\kappa}^*$  for  $\kappa = 1, \dots, \mathcal{M}$  are the so-called magic points and are determined by a greedy search as shown in Barrault et al. (2004) and depicted in Algorithm 1. The key idea behind this approach is to split the function  $pdf_{X_{t_{u+1}-t_u}}^{\mu}(x)$ , which depends on the parameters  $\mu$  as well as on the space variable  $x$ , into a parameter-dependent

part  $\Theta_\kappa(\mu, x_\kappa^*)$  and a space-dependent part  $q_\kappa(x)$ . In such a way, the computationally demanding tasks like the derivation of integrals with respect to the space variable  $x$ , is reduced to applying these tasks only to the space dependent part  $q_\kappa(x)$ . Evaluating these tasks for several parameters is then reduced to evaluating the linear combinations. The algorithm of Barrault et al. (2004) as depicted in Algorithm 1 does not only provide the magic points  $x_k^*$  for  $k = 1, \dots, \mathcal{M}$ , but also the functions  $q_\kappa(x)$ . For the parameter-dependent part it is known that the functions  $\Theta_\kappa(\mu, x_\kappa^*)$  are given by evaluating the approximated function at the fixed points  $x_k^*$  for the parameters  $\mu$ , i.e. for  $k = 1, \dots, \mathcal{M}$  it holds  $\Theta_\kappa(\mu, x_\kappa^*) = pdf_{X_{t_{u+1}-t_u}}^\mu(x_\kappa^*)$ .

In the following, we now assume

- a one dimensional underlying space,  $\mathcal{X} = [x, \bar{x}] \subset \mathbb{R}$ ,
- the underlying model is a Markov model with density function,
- an equidistant time stepping such that  $\Delta t = t_{u+1} - t_u$  for all  $u = 1, \dots, n_T - 1$  in the dynamic programming scheme.

Hence,  $pdf_{X_{t_{u+1}-t_u}}^\mu(x) = pdf_{X_{\Delta t}}^\mu(x)$ . For instance, in the Black&Scholes model, the probability density function is known in closed-form and thus, with  $\mu = (r, \sigma, \Delta t, x_0)$ ,

$$pdf_{X_{\Delta t}}^\mu(x) = \frac{1}{\sqrt{2\pi\Delta t}\sigma} e^{-\frac{1}{2} \frac{(x-x_0 - [(r-\frac{1}{2}\sigma^2)\Delta t])^2}{\Delta t\sigma^2}}. \quad (5.25)$$

By applying the empirical interpolation of Barrault et al. (2004) as depicted in Algorithm 1, we approximate the density function as

$$\frac{1}{\sqrt{2\pi\Delta t}\sigma} e^{-\frac{1}{2} \frac{(x-x_0 - [(r-\frac{1}{2}\sigma^2)\Delta t])^2}{\Delta t\sigma^2}} \approx \sum_{\kappa=1}^{\mathcal{M}} \Theta_\kappa(r, \sigma, \Delta t, x_0, x_\kappa^*) q_\kappa(x), \quad x \in \mathcal{X}.$$

Here, it is important to understand that we approximate the density function on a bounded parameter domain,

$$\mathcal{P} = [r, \bar{r}] \times [\sigma, \bar{\sigma}] \times [x_0, \bar{x}_0] \times [\underline{\Delta t}, \overline{\Delta t}]. \quad (5.26)$$

This allows the following statement for the approximation error.

**Theorem 5.5.1.** *Let the parametric density function  $pdf_{X_{\Delta t}}^\mu$  be given as in (5.25) and let the empirical interpolation be applied for  $x \in \mathcal{X} = [x, \bar{x}]$  and the bounded parameter domain  $\mathcal{P}$  be given such that the mapping  $p \mapsto pdf_{X_{\Delta t}}^p$  is analytic. Further, let  $\underline{\sigma}, \underline{\Delta t} > 0$ . Then, it exists  $C > 0$  such that*

$$\max_{x \in \mathcal{X}} \max_{\mu \in \mathcal{P}} \left| pdf_{X_{\Delta t}}^\mu - \sum_{\kappa=1}^{\mathcal{M}} \Theta_\kappa(\mu, x_\kappa^*) q_\kappa(x) \right| \leq C \cdot \mathcal{M} e^{-(\alpha - \log(4))\mathcal{M}},$$

where  $\alpha > \log(4)$ .

## 5 Dynamic Programming Framework with Chebyshev Interpolation

*Proof.* From  $\underline{\sigma}, \underline{\Delta t} > 0$  it directly follows that the mapping  $\mathcal{P} \times \mathcal{X} \ni (\mu, x) \mapsto pdf_{X_{\Delta t}}^\mu(x)$  is bounded and  $\mu \mapsto pdf_{X_{\Delta t}}^\mu(\cdot)$  is sequentially continuous. Moreover, the function  $pdf_{X_{\Delta t}}^\mu$  is holomorphic for  $x \in (-\infty, \infty)$ . Therefore, it is possible to stretch the Bernstein ellipse until a corresponding  $\varrho > \log(4)$  is found. Thus, we can apply (Gaß and Glau, 2015, Theorem 3.2) and the assertion follows.  $\square$

In this initial example, we investigated the Black&Scholes model. For several other models, such as the Heston or Merton model, the probability density function is not given in a closed-form. However, the characteristic function often exists for these models. Then, it holds

$$pdf_{X_{\Delta t}}^\mu(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-izx} \varphi(z; \mu) dz,$$

where  $\varphi(z; \mu)$  is the characteristic function. For the empirical interpolation, we follow now the approach of Gaß and Glau (2015) and express the integrand as follows,

$$\begin{aligned} e^{-izx} \varphi(z; \mu) &\approx \sum_{\kappa=1}^{\mathcal{M}} \varphi(z_\kappa^*; \mu, x) q_\kappa(z) \\ &= \sum_{\kappa=1}^{\mathcal{M}} e^{-iz_\kappa^* x} \varphi(z_\kappa^*; \mu) q_\kappa(z). \end{aligned}$$

This leads directly to

$$\begin{aligned} pdf_{X_{\Delta t}}^\mu(x) &\approx \frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{\kappa=1}^{\mathcal{M}} e^{-iz_\kappa^* x} \varphi(z_\kappa^*; \mu) q_\kappa(z) dz \\ &= \frac{1}{2\pi} \sum_{\kappa=1}^{\mathcal{M}} e^{-iz_\kappa^* x} \varphi(z_\kappa^*; \mu) \int_{-\infty}^{\infty} q_\kappa(z) dz. \end{aligned}$$

The integrals can be pre-computed, we denote  $I_\kappa := \int_{-\infty}^{\infty} q_\kappa(z) dz$ ,

$$pdf_{X_{\Delta t}}^\mu(x) \approx \frac{1}{2\pi} \sum_{\kappa=1}^{\mathcal{M}} e^{-iz_\kappa^* x} \varphi(z_\kappa^*; \mu) I_\kappa. \quad (5.27)$$

Given this empirical interpolation at the magic points  $z_\kappa^*$  allows us to adjust Algorithm 4 in the following way. For the derivation of the conditional expectations  $E[V_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = x]$  at any time step  $t_u$ , we still interpolate the value function  $V_{t_{u+1}}$  of the preceding time step. Let  $V_{t_{u+1}}(x) \approx \sum_{j=1}^N c_j(t_{u+1}) T_j(x)$ . The conditional expectation is then given as an integral over the probability density function, which we

replace with the approximation by empirical interpolation, (5.27),

$$E[V_{t_{u+1}}(X_{t_{u+1}})|X_{t_u} = x_0] \approx E\left[\sum_{j=1}^N c_j(t_{u+1})T_j(\tau_{\mathcal{X}}^{-1}(x))|X_{t_u} = x_0\right] \quad (5.28)$$

$$= \sum_{j=1}^N c_j(t_{u+1})E[T_j(\tau_{\mathcal{X}}^{-1}(x))|X_{t_u} = x_0] \quad (5.29)$$

$$= \sum_{j=1}^N c_j(t_{u+1}) \int_{\underline{x}}^{\bar{x}} T_j(\tau_{\mathcal{X}}^{-1}(x)) pdf_{X_{\Delta t}}^{\mu}(x) dx. \quad (5.30)$$

Now, we apply the empirical interpolation for the density function,

$$\approx \sum_{j=1}^N c_j(t_{u+1}) \int_{\underline{x}}^{\bar{x}} T_j(\tau_{\mathcal{X}}^{-1}(x)) \sum_{\kappa=1}^{\mathcal{M}} e^{-iz_{\kappa}^* x} \varphi(z_{\kappa}^*; \mu) I_{\kappa} dx \quad (5.31)$$

$$= \sum_{j=1}^N \sum_{\kappa=1}^{\mathcal{M}} c_j(t_{u+1}) \varphi(z_{\kappa}^*; \mu) I_{\kappa} \int_{\underline{x}}^{\bar{x}} T_j(\tau_{\mathcal{X}}^{-1}(x)) e^{-iz_{\kappa}^* x} dx. \quad (5.32)$$

As shown above, by deriving the conditional expectations we use an approximation at two steps: First, the Chebyshev interpolation in (5.28) and, second, the approximation of the probability density function with the empirical interpolation in (5.31). This requires a modification of Corollary 5.2.4 to incorporate the approximation error from the empirical interpolation, too. Hence, we want to describe the complete dynamic programming procedure as an algorithm. By looking at (5.32), we see an additional interesting feature. For each Chebyshev polynomial  $j$  and each magic point  $z_{\kappa}^*$ , integrals of the form  $\int_{\underline{x}}^{\bar{x}} T_j(x) e^{-iz_{\kappa}^* x} dx$  have to be evaluated, independently of the parameter  $\mu$ . This allows us to evaluate these integrals in a pre-computation step, as long as no splitting is applied in the time stepping scheme. Moreover, in the dynamic Chebyshev procedure, we derive the conditional expectations with respect to each nodal point. The computational efforts can be reduced significantly, when  $x_0$ , the placeholder for the condition in the conditional expectation, is treated as additional parameter in the empirical interpolation. In the following, we assume a specific type of probability density function respectively characteristic function. Let  $pdf_{X_{\Delta t}}^{\mu}$  denote the probability density function and  $\varphi(z; \mu)$  the corresponding characteristic function, then we assume for the conditional density and characteristic functions,

$$\begin{aligned} pdf_{X_{\Delta t}}^{\mu}(x|x_0) &= pdf_{X_{\Delta t}}^{\mu}(x - x_0) \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iz(x-x_0)} \varphi(z; \mu) dz \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-izx} e^{izx_0} \varphi(z; \mu) dz. \end{aligned} \quad (5.33)$$

## 5 Dynamic Programming Framework with Chebyshev Interpolation

For example, this holds for the Black&Scholes model or, more generally, in Lévy models. Before applying the empirical interpolation algorithm, numerical tests have shown that the results of applying empirical interpolation are more stable, if we focus on

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-izx} \varphi(z; \mu) dz = \frac{1}{\pi} \int_0^{\infty} \text{Real} (e^{-izx} \varphi(z; \mu)) dz.$$

Here, we can use twice the value of the integral of the real part over the positive domain instead of the integral of the function over the whole domain. This holds because for the characteristic function  $\overline{\varphi(z; \mu)} = \varphi(-z; \mu)$  holds, thus it also holds also for  $e^{-izx} e^{izx_0} \varphi(z; \mu)$ . Hence, we approximate

$$pdf_{X_{\Delta t}}^{\mu}(x|x_0) \approx \frac{1}{\pi} \int_0^{\bar{z}} \text{Real} (e^{-izx} e^{izx_0} \varphi(z; \mu)) dz.$$

The empirical interpolation is applied to the integrand  $\text{Real} (e^{-izx} e^{izx_0} \varphi(z; \mu))$  and this results in summands of the form  $\text{Real} (e^{-iz^*x} e^{iz^*x_0} \varphi(z^*; \mu))$ . Here, to evaluate the real part, we use,

$$\text{Real} (e^{-iz^*x} e^{iz^*x_0} \varphi(z^*; \mu)) = \frac{e^{-iz^*x} e^{iz^*x_0} \varphi(z^*; \mu) + e^{iz^*x} e^{-iz^*x_0} \varphi(-z^*; \mu)}{2}$$

Due to this step, instead of (5.32), we have to apply this integration twice for both parts of the numerator. However, in the pre-computation step this additional costs are more tolerable. Especially, when the empirical interpolation is numerically more stable in this case. In Algorithm 8, we describe this procedure in detail. Then, in Theorem 5.5.2, we show how the additional approximation via the empirical interpolation effects the error bounds for the dynamic Chebyshev procedure.

---

**Algorithm 8** Dynamic Chebyshev with empirical interpolation
 

---

- 1: **procedure** PRE-COMPUTATION STEP
  - 2:     Fix an interval  $[\underline{x}, \bar{x}]$  for the space of the underlying
  - 3:     Fix  $N$  as number of nodal points of the Chebyshev interpolation
  - 4:     Determine nodal points  $x^k = \cos(\pi \frac{k}{N})$  for  $k = 0, \dots, N$
  - 5:     Set up equidistant time stepping  $0 = t_1, \dots, t_{n_T} = T$  with  $\Delta t = t_2 - t_1$
  - 6:     Define for the parameter  $p \in \mathbb{R}^d$  of the characteristic function a rectangular parameter space  $\mathcal{P} = [\underline{p}_1, \bar{p}_1] \times \dots \times [\underline{p}_d, \bar{p}_d]$
  - 7:     Apply the empirical interpolation algorithm for  $(x, x_0, \mu) \in [\underline{x}, \bar{x}] \times ([\underline{x}, \bar{x}] \times \mathcal{P})$  to  $Real(e^{-izx} e^{izx_0} \varphi(z; \mu))$  with pre-specified tolerance  $\varepsilon_M$  on truncated domain  $[0, \bar{z}]$ .
  - 8:     For  $\kappa = 1 : M$ ,
  - 9:     store  $z_\kappa^*$  of  $Real(e^{-izx} e^{izx_0} \varphi(z; \mu)) \approx \sum_{\kappa=1}^M Real(e^{-iz_\kappa^* x} e^{iz_\kappa^* x_0} \varphi(z_\kappa^*; \mu)) q_\kappa(z)$
  - 10:    derive  $I_\kappa = \int_0^{\bar{z}} q_\kappa(z) dz$
  - 11:    Approximation of density function by  $\frac{1}{\pi} \sum_{\kappa=1}^M Real(e^{-iz_\kappa^* x} e^{iz_\kappa^* x_0} \varphi(z_\kappa^*; \mu)) I_\kappa$
  - 12:    For  $j = 0 : N$  and  $\kappa = 1 : M$ ,
  - 13:    compute integrals  $Int_{j,\kappa}^1 := \int_{\underline{x}}^{\bar{x}} T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x)) e^{-iz_\kappa^* x} dx$
  - 14:    compute integrals  $Int_{j,\kappa}^2 := \int_{\underline{x}}^{\bar{x}} T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x)) e^{iz_\kappa^* x} dx$
  - 15: **procedure** DERIVATION PARAMETER-DEPENDENT PARTS
  - 16:     For  $k = 0 : N$ , compute for fixed  $\mu$  and with  $x_0 = y^k$  for each  $\kappa = 1 : M$ ,  $e^{iz_\kappa^* x_0} \varphi(z_\kappa^*; \mu)$  and  $e^{-iz_\kappa^* x_0} \varphi(-z_\kappa^*; \mu)$
  - 17:     For  $\kappa = 1 : M$ , for  $j = 1 : N$ ,
  - 18:     derive  $S_{j,\kappa}(y^k) = \frac{1}{2} \left( e^{iz_\kappa^* y^k} \varphi(z_\kappa^*; \mu) Int_{j,\kappa}^1 + e^{-iz_\kappa^* y^k} \varphi(-z_\kappa^*; \mu) Int_{j,\kappa}^2 \right)$
  - 19: **procedure** INITIAL TIME  $T$
  - 20:     Set nodal points  $y^k = \tau_{[\underline{x}, \bar{x}]}(x^k)$
  - 21:      $P_T(y^k) = g(T, y^k)$ ,  $k = 1 : N$ , derive
  - 22:      $c_j(T) = \left( \frac{2^{\mathbb{1}_{\{0 < j < N\}}}}{N} \right) \sum_{k=0}^N P_T(y^k) \cos(j\pi \frac{k}{N})$
  - 23:     Obtain Chebyshev interpolation of  $P_T(x) = \sum_{j=0}^N c_j(T) T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x))$
  - 24: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 25:     Given Chebyshev interpolation of  $P_{t_{u+1}}(x) = \sum_{j=0}^N c_j(t_{u+1}) T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x))$
  - 26:     Derivation of  $P_{t_i}(y^k)$  at the nodal points with  $y^k$
  - 27:     **for**  $k = 0, \dots, N$
  - 28:          $P_{t_u}(y^k) = f(g(t_u, y^k), E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y^k])$
  - 29:          $P_{t_u}(y^k) = f(g(t_u, y^k), e^{-r\Delta t} \sum_{j=0}^N c_j(t_{u+1}) E[T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(X_{t_{u+1}}) | X_{t_u} = y^k])$
  - 30:          $P_{t_u}(y^k) = f(g(t_u, y^k), \frac{e^{-r\Delta t}}{\pi} \sum_{j=0}^N \sum_{\kappa=1}^M c_j(t_{u+1}) I_\kappa S_{j,\kappa}(y^k))$
  - 31:     **end**
  - 32:     Derive
  - 33:      $c_j(t_u) = \left( \frac{2^{\mathbb{1}_{\{0 < j < N\}}}}{N} \right) \sum_{k=0}^N P_{t_u}(y^k) \cos(j\pi \frac{k}{N})$
  - 34:     Obtain Chebyshev interpolation of  $P_{t_u}(x) = \sum_{j=0}^N c_j(t_u) T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x))$
  - 35: **procedure** DERIVING THE SOLUTION AT T=0
  - 36:      $P_0(x) = \sum_{j=0}^N c_j(0) T_j(\tau_{[\underline{x}, \bar{x}]^{-1}}^{-1}(x))$
-

**Theorem 5.5.2.** *Let a Dynamic Programming Principle be given as in (5.3) and (5.4). Given a time stepping  $t = t_1 < \dots < t_{n_T} = T$ , let  $\mathcal{X} \ni x \mapsto V_{t_u}(x)$  be a real valued function that has an analytic extension to a generalized Bernstein ellipse  $B(\mathcal{X}, \varrho_{t_u})$  with parameter vector  $\varrho_{t_u} \in (1, \infty)^D$  and  $\sup_{x \in B(\mathcal{X}, \varrho_{t_u})} |V_{t_u}(x)| \leq M_{t_u}$  for  $k = 1, \dots, n_T$ . Furthermore, let  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be continuous. Moreover, let the conditional density and characteristic functions at each time step  $t_u$  satisfy the relation (5.33). Let the mapping  $z \mapsto \text{Real}(e^{-iz(x-x_0)}\varphi(z; \mu))$  be analytic and bounded on  $B([-z, z], \varrho_M)$  with  $\varrho_M > 4$  for any arbitrary parameter combination  $\mu \in [\underline{x}, \bar{x}] \times \mathcal{P}$ , where  $\mathcal{P}$  is a compact parameter domain.*

*Then, by applying Algorithm 8 the resulting solution  $P_{t_u}(x)$  converges to the solution  $V_{t_u}(x)$  for  $N \rightarrow \infty$  and  $\varepsilon_{\bar{z}} \approx 0$ . Furthermore, the approximation error at time  $t_u$  is given by*

$$\begin{aligned} & \max_{x \in \mathcal{X}} |V_{t_k}(x) - P_{t_k}| \\ & \leq \sum_{j=k}^{n_T} C_{D,N}^{j-k} \alpha(\varrho_{t_j}, N, D, M_{t_j}) + \sum_{j=k+1}^{n_T} C_{D,N}^{j-k} \left( M_{t_{n_T}} \left( C_{\bar{z}} \mathcal{M} \left( \frac{\varrho_M}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}} \right) (\bar{x} - \underline{x}) \right) \end{aligned} \quad (5.34)$$

where  $C_{D,N}$  as in (5.10),  $\alpha(\varrho_{t_k}, N, D, M_{t_k})$  as in (5.9) and  $\varepsilon_{\bar{z}}$  denotes the truncation error of the empirical interpolation.

*Proof.* The proof of the recursive backward time stepping will be similar to the proofs of Theorem 5.2.2 and Corollary 5.2.4. For an error estimation, we have to consider the empirical interpolation here in detail. First step is an empirical interpolation on a rectangular domain  $[\underline{x}, \bar{x}] \times \mathcal{P}$ , including the rectangular parameter space  $\mathcal{P} = [\underline{\mu}_1, \bar{\mu}_1] \times \dots \times [\underline{\mu}_d, \bar{\mu}_d]$ . We approximate

$$\text{pdf}_{X_{\Delta t}}^{\mu}(x|x_0) \approx \frac{1}{\pi} \int_0^{\bar{z}} \text{Real}(e^{-izx} e^{izx_0} \varphi(z; \mu)) dz.$$

By applying empirical interpolation, we truncate the domain to  $[0, \bar{z}]$ . Thus, we introduce a truncation error. However, let us first focus on the empirical interpolation on the bounded domain  $[0, \bar{z}]$  and the integration over this bounded domain.

The empirical interpolation is applied to the integrand  $\text{Real}(e^{-izx} e^{izx_0} \varphi(z; \mu))$ , yielding,

$$\text{Real}(e^{-izx} e^{izx_0} \varphi(z; \mu)) \approx \sum_{\kappa=1}^{\mathcal{M}} \text{Real}\left(e^{-iz_{\kappa}^* x} e^{iz_{\kappa}^* x_0} \varphi(z_{\kappa}^*; \mu)\right) q_{\kappa}(z).$$

Then, by choosing a compact parameter domain  $\mathcal{P}$ , similarly to the setting of Theorem 5.5.1, we can apply the error bound of Gaß and Glau (2015). Note that here the mapping is analytic, because  $z \mapsto \text{Real}(e^{-izx} e^{izx_0} \varphi(z; \mu)) = z \mapsto \frac{e^{-izx} e^{izx_0} \varphi(z; \mu) + e^{izx} e^{-izx_0} \varphi(-z; \mu)}{2}$  and  $z \mapsto e^{-izx} e^{izx_0} \varphi(z; \mu)$  is analytic as well as  $z \mapsto e^{izx} e^{-izx_0} \varphi(-z; \mu)$ . Moreover,  $\text{Real}(e^{-izx} e^{izx_0} \varphi(z; \mu))$  is bounded on  $B([-z, z], \varrho_M)$  with  $\varrho_M > 4$ . In this case, Gaß

## 5 Dynamic Programming Framework with Chebyshev Interpolation

and Glau (2015) directly yield that it exists  $C > 0$  such that

$$\begin{aligned} \max_{p \in \mathcal{P}, x \in \mathcal{X}} \left| \frac{1}{\pi} \int_0^{\bar{z}} \text{Real} \left( e^{-izx} e^{izx_0} \varphi(z; \mu) \right) dz - \frac{1}{\pi} \int_0^{\bar{z}} \sum_{\kappa=1}^{\mathcal{M}} \text{Real} \left( e^{-iz_{\kappa}^* x} e^{iz_{\kappa}^* x_0} \varphi(z_{\kappa}^*; \mu) \right) q_{\kappa}(z) dz \right| \\ \leq C \bar{z} \mathcal{M} \left( \frac{\varrho \mathcal{M}}{4} \right)^{-\mathcal{M}}, \end{aligned}$$

for  $\varrho \mathcal{M} > 4$ .

Up to this point, we know that we can approximate the probability density function with an error bounded by  $C \bar{z} \mathcal{M} \left( \frac{\varrho \mathcal{M}}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}}$ . Here, the second part indicates the truncation error of the integration region from  $[0, \infty)$  to  $[0, \bar{z}]$ . With these computations, we are in the setting of Theorem 5.2.2 and Corollary 5.2.4. In a DPP setting like (5.3) and (5.4), we will show how the approximation error evolves in the backward time stepping scheme. At the initial time step  $t_{n_T} = T$ , the error of the Chebyshev interpolation of  $g(T, x) = V_T(x)$  is bounded by (5.9),

$$\max_{x \in \mathcal{X}} |V_T(x) - P_{t_{n_T}}(x)| \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}).$$

Similar to the proof of Theorem 5.4.1, we let the exact value of the conditional probability density function be given by

$$pdf_{X_{\Delta t}}(x) = pdf_{X_{\Delta t}}^{\mathcal{M}}(x) + pdf_{\Delta t}^{\varepsilon}(x),$$

where  $pdf_{X_{t_{u+1}-t_u}}^{\mathcal{M}}(x)$  represents the empirically interpolated probability density functions and  $pdf_{X_{t_{u+1}-t_u}}^{\varepsilon}(x)$  the error term. For the error at the nodal points, this yields analogously,

$$\begin{aligned} \max_{j \in J} |V_{t_{n_T-1}}(x_j) - P_{t_{n_T-1}}(x_j)| \\ \leq \max_{j \in J} \int_{\underline{x}-x_j}^{\bar{x}-x_j} |V_{t_{n_T}}(x+x_j) - P_{t_{n_T}}(x+x_j)| pdf_{X_{\Delta t}}(x) dx \\ + \int_{\underline{x}-x_j}^{\bar{x}-x_j} |P_{t_{n_T}}(x+x_j)| |pdf_{X_{\Delta t}}^{\varepsilon}(x)| dx \end{aligned}$$

Now, as shown in the proof of Corollary 5.2.4, the error  $\max_{x \in \mathcal{X}} |V_{t_{n_T}}(x) - P_{t_{n_T}}(x)|$  is bounded by  $\alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}})$ , we know that the expected error using the real probability density function is bounded by exactly  $\alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}})$ . For the second summand, we use  $|P_{t_{n_T}}(x+x_j)| \leq M_{t_{n_T}}$  and  $|pdf_{X_{\Delta t}}^{\varepsilon}(x)| \leq C \bar{z} \mathcal{M} \left( \frac{\varrho \mathcal{M}}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}}$ . Combining these

yields,

$$\begin{aligned} & \max_{j \in J} |V_{t_{n_T-1}}(x_j) - P_{t_{n_T-1}}(x_j)| \\ & \leq \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) + M_{t_{n_T}} \cdot \left( C_{\bar{z}} \mathcal{M} \left( \frac{\varrho M}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}} \right) \cdot (\bar{x} - \underline{x}). \end{aligned}$$

With  $C_{D,N}$  as in (5.10), this yields, analogously to the proof of Corollary 5.2.4, for the error at  $t_{n_T-1}$ ,

$$\begin{aligned} & \max_{x \in \mathcal{X}} |V_{t_{n_T-1}}(x) - P_{t_{n_T-1}}(x)| \leq \varepsilon_{t_{n_T-1}} := \alpha(\varrho_{t_{n_T-1}}, N, D, M_{t_{n_T-1}}) \\ & + C_{D,N} \left( \alpha(\varrho_{t_{n_T}}, N, D, M_{t_{n_T}}) + M_{t_{n_T}} \cdot \left( C_{\bar{z}} \mathcal{M} \left( \frac{\varrho M}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}} \right) \cdot (\bar{x} - \underline{x}) \right). \end{aligned}$$

This directly leads to the recursive scheme for the error bound at  $t_u$ ,

$$\begin{aligned} \varepsilon_{t_u} & = \alpha(\varrho_{t_u}, N, D, M_{t_u}) + \\ & C_{D,N} \left( \varepsilon_{t_{u+1}} + M_{t_{n_T}} \cdot \left( C_{\bar{z}} \mathcal{M} \left( \frac{\varrho M}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}} \right) \cdot (\bar{x} - \underline{x}) \right). \end{aligned}$$

Now, applying a similar induction, as in the proof of Corollary 5.2.4, directly yields,

$$\varepsilon_{t_k} = \sum_{j=k}^{n_T} C_{D,N}^{j-k} \alpha(\varrho_{t_j}, N, D, M_{t_j}) + \sum_{j=k+1}^{n_T} C_{D,N}^{j-k} \left( M_{t_{n_T}} \left( C_{\bar{z}} \mathcal{M} \left( \frac{\varrho M}{4} \right)^{-\mathcal{M}} + \varepsilon_{\bar{z}} \right) (\bar{x} - \underline{x}) \right).$$

□

**Remark 5.5.3.** *As the error bound in (5.34) indicates, the truncation error of the interval,  $\varepsilon_{\bar{z}}$ , is included as a factor and is not multiplied by any decaying term. Thus, this truncation error only gets scaled with a larger factor. Therefore, in the implementation of Algorithm 8, the empirical interpolation has to be applied carefully. In general, the characteristic functions decay for increasing  $z$  exponentially and therefore, by making  $\bar{z}$  reasonably large, the truncation error  $\varepsilon_{\bar{z}}$  decays exponentially as well. For a variety of Lévy models, this is shown in Glau (2016).*

If we want to combine Algorithm 8 with the theoretic statements from Theorem 5.5.2, analyticity is required. However, due to the kink of the payoff function in the case of a plain vanilla call or put option, this is not given. As in Section 5.3, we will introduce here the concept of splitting. For the example of an American put option, we present in Algorithm 9 a version with splitting.

---

**Algorithm 9** Dynamic Chebyshev with empirical interpolation and splitting for an American put option

---

- 1: **procedure** PRE-COMPUTATION STEP
  - 2:     As in Algorithm 8
  - 3: **procedure** DERIVATION PARAMETER-DEPENDENT PARTS
  - 4:     Fix a parameter  $\mu$
  - 5:     For  $k = 0 : N$ , compute for fixed  $\mu$  and with  $x_0 = x^k$  for each  $\kappa = 1 : M$ ,  
 $e^{iz_\kappa^* x_0} \varphi(z_\kappa^*; \mu)$  and  $e^{-iz_\kappa^* x_0} \varphi(-z_\kappa^*; \mu)$
  - 6: **procedure** INITIAL TIME  $T$
  - 7:     As in Algorithm 5
  - 8: **procedure** ITERATIVE TIME STEPPING FROM  $t_{u+1} \rightarrow t_u$ ,  $u = n_T - 1, \dots, 1$
  - 9:     For  $j_1 = 0 : N_1$ ,  $\kappa = 1 : M$ ,  $\omega = 1, 2$ , compute integrals
  - 10:      $Int_{j_1, \kappa}^\omega := \int_{\underline{x}}^{\tilde{x}_{t_{u+1}}} T_{j_1}(\tau_{[\underline{x}, \tilde{x}_{t_{u+1}}]}^{-1}(x)) e^{(-1)^\omega iz_\kappa^* x} dx$
  - 11:     For  $j_2 = 0 : N_2$ ,  $\kappa = 1 : M$ ,  $\omega = 1, 2$ , compute integrals
  - 12:      $Int_{j_2, \kappa}^\omega := \int_{\tilde{x}_{t_{u+1}}}^{\bar{x}} T_{j_2}(\tau_{[\tilde{x}_{t_{u+1}}, \bar{x}]}^{-1}(x)) e^{(-1)^\omega iz_\kappa^* x} dx$
  - 13:     For  $\kappa = 1 : M$ , for  $j_1 = 1 : N_1$ ,
  - 14:     define  $S_{j_1, \kappa}(y_0) = \frac{1}{2} \left( e^{iz_\kappa^* y_0} \varphi(z_\kappa^*; \mu) Int_{j_1, \kappa}^1 + e^{-iz_\kappa^* y_0} \varphi(-z_\kappa^*; \mu) Int_{j_1, \kappa}^2 \right)$
  - 15:     For  $\kappa = 1 : M$ , for  $j_2 = 1 : N_2$ ,
  - 16:     define  $S_{j_2, \kappa}(y_0) = \frac{1}{2} \left( e^{iz_\kappa^* y_0} \varphi(z_\kappa^*; \mu) Int_{j_2, \kappa}^1 + e^{-iz_\kappa^* y_0} \varphi(-z_\kappa^*; \mu) Int_{j_2, \kappa}^2 \right)$
  - 17:     Define functions in dependence on  $y_0$  to determine the splitting point
  - 18:      $E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_0] =$
  - 19:      $\frac{e^{-r\Delta t}}{\pi} \sum_{j_1=0}^{N_1} \sum_{\kappa=1}^M c_{j_1}(t_{u+1}) I_\kappa S_{j_1, \kappa}(y_0)$
  - 20:      $+ \frac{e^{-r\Delta t}}{\pi} \sum_{j_2=0}^{N_2} \sum_{\kappa=1}^M c_{j_2}(t_{u+1}) I_\kappa S_{j_2, \kappa}(y_0)$
  - 21:     Determine splitting point
  - 22:     Find  $\tilde{x}_{t_u}$  as root of  $y_0 \mapsto g(t_u, y_0) - E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_0]$
  - 23:     Split interval into  $[\underline{x}, \tilde{x}_{t_u}]$  and  $[\tilde{x}_{t_u}, \bar{x}]$
  - 24:     Set  $y_1^k = \tau_{[\underline{x}, \tilde{x}_{t_u}]}(x_1^k)$  and  $y_2^k = \tau_{[\tilde{x}_{t_u}, \bar{x}]}(x_2^k)$
  - 25:     Apply Chebyshev interpolation on both intervals
  - 26:      $P_{t_u}(y_1^k) = g(T, y_1^k)$ ,  $k = 1 : N_1$ , derive
  - 27:      $c_{1, j_1}(t_u) = \left( \frac{2^{\mathbb{1}_{\{0 < j_1 < N_1\}}}}{N_1} \right) \sum_{k=0}^{N_1} P_{t_u}(y_1^k) \cos\left(j_1 \pi \frac{k}{N_1}\right)$
  - 28:      $P_{t_u}(y_2^k) = E[P_{t_{u+1}}(X_{t_{u+1}}) | X_{t_u} = y_2^k]$ ,  $k = 1 : N_2$ , derive
  - 29:      $c_{2, j_2}(t_u) = \left( \frac{2^{\mathbb{1}_{\{0 < j_2 < N_2\}}}}{N_2} \right) \sum_{k=0}^{N_2} P_{t_u}(y_2^k) \cos\left(j_2 \pi \frac{k}{N_2}\right)$
  - 30:     Obtain Chebyshev interpolation
  - 31:      $P_{t_u}(x) = \sum_{j_1=0}^{N_1} c_{1, j_1}(t_u) T_{j_1}(\tau_{[\underline{x}, \tilde{x}_{t_u}]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \tilde{x}_{t_u}]}(x)$
  - 32:      $+ \sum_{j_2=0}^{N_2} c_{2, j_2}(t_u) T_{j_2}(\tau_{[\tilde{x}_{t_u}, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\tilde{x}_{t_u}, \bar{x}]}(x)$
  - 33: **procedure** DERIVING THE SOLUTION AT  $T=0$
  - 34:      $P_0(x) = \sum_{j_1=0}^{N_1} c_{1, j_1}(0) T_{j_1}(\tau_{[\underline{x}, \bar{x}_0]}^{-1}(x)) \cdot \mathbb{1}_{[\underline{x}, \bar{x}_0]}(x)$
  - 35:      $+ \sum_{j_2=0}^{N_2} c_{2, j_2}(0) T_{j_2}(\tau_{[\bar{x}_0, \bar{x}]}^{-1}(x)) \cdot \mathbb{1}_{[\bar{x}_0, \bar{x}]}(x)$
-

## 5.6 Numerical Experiments - Example Bermudan and American options

In finance, there are in general two types of plain vanilla options: European and American options. For a European option, the option-holder only has the right to exercise the option at maturity, whereas, for American options, the option-holder has the right to exercise the option at any time up to the maturity. In the following, we will investigate options which give the option-holder the right to exercise the option at a restricted set of pre-specified possible exercise dates. This option type is called the Bermudan option. Similar to the situation of the Bermuda islands between Europe and America, Bermudan options take an intermediate place between European and American options, see Schweizer (2002).

**Remark 5.6.1.** *So far, in the theoretic results, we have assumed that the function only has support on the bounded domain  $\mathcal{X}$ . The examples in the following are option prices. For computational purposes, we work in the state space with log variables and thus, technically,  $\mathcal{X} = (-\infty, \infty)$  in the one dimensional case being an example. By applying Algorithm 3 on a compact domain  $\tilde{\mathcal{X}} = [\underline{x}, \bar{x}]$ , a truncation error is made at both sides. As we are considering Bermudan and American put options, we neglect the error made by cutting the domain on the right side by setting  $\bar{x}$  reasonably high. We consider the truncation error on the left side by adding to the conditional expectations  $E[T_j(\tau_{\mathcal{X}}^{-1}(X_{t_{u+1}})) \cdot \mathbf{1}_{\{X_{t_{u+1}} \in \mathcal{X}\}} | X_{t_u} = x^k]$ , the expected value of the payoff function on  $(-\infty, \underline{x})$ ,  $E[g(X_{t_{u+1}}) | X_{t_u} = x^k]$ . Here, the assumption is that  $\underline{x}$  is chosen small enough that below  $\underline{x}$ , we are, in any case, in the exercise region and would exercise the Bermudan or American put option.*

In the following, we present numerical results of the introduced dynamic programming framework for the Black&Scholes model. Therefore, we define the test setting,

$$\begin{aligned} S_0 &= 100, & \underline{S} &= 0.02, & \bar{S} &= 250 \\ \sigma &= 0.15, & r &= 0.03, & n_T &= 32 \\ T &= 1, & K &= 100. \end{aligned} \tag{5.35}$$

Our reference option type is an at-the-money American (Bermudan) put option with all the parameters specified in (5.35). To investigate the accuracy of our proposed method, we use the cosine method of Fang and Oosterlee (2009) as the benchmark method. In von Sydow et al. (2015), the benchmark code is provided and reported with a relative accuracy of  $10^{-4}$ .

### Application of Algorithm 6

First, we apply Algorithm 6 and due to the kink of the put option payoff at maturity, we apply a splitting of the domain as  $[\underline{S}, \bar{S}] = [\underline{S}, K] \cup [K, \bar{S}]$ . By applying a fixed

## 5 Dynamic Programming Framework with Chebyshev Interpolation

splitting of the domain at  $K$ , for this numerical study, we set the number of Chebyshev interpolation points equal on both sub-intervals. In Figure 5.1, we present the empirical error decay for increasing the number of nodal points  $N$  for integrating over the density function, in Figure 5.2 for the truncated moment method and in Figure 5.3 for using Fourier techniques to derive the conditional expectations.

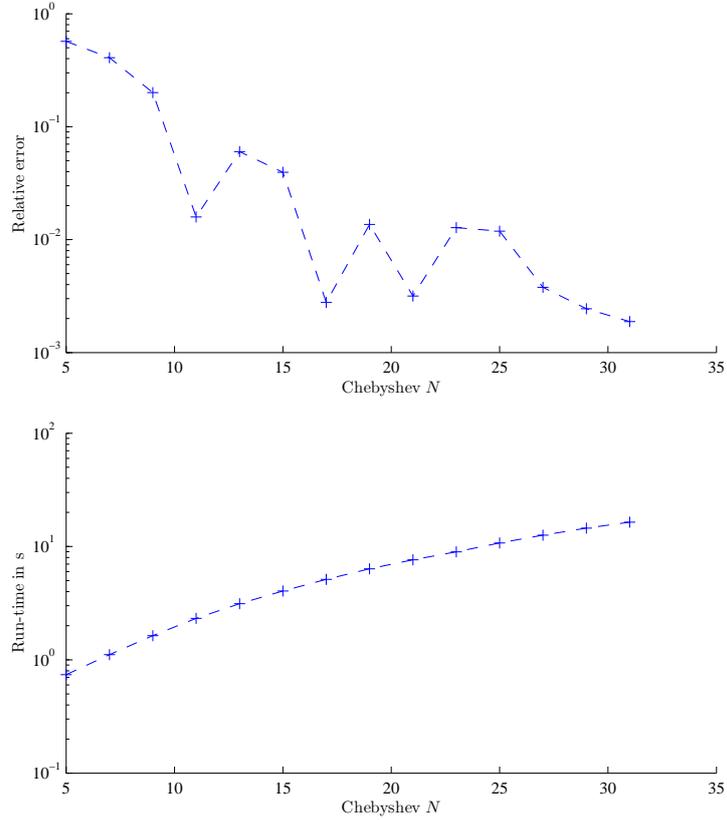


Figure 5.1: Convergence study of Algorithm 6 for the integration over the density function (top). For increasing  $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

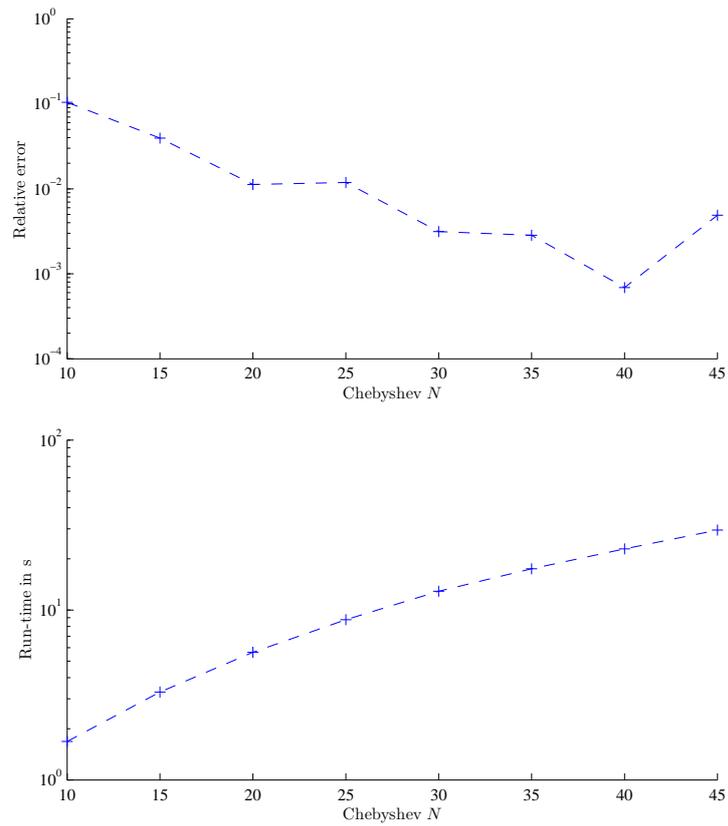


Figure 5.2: Convergence study of Algorithm 6 for applying the truncated moment method (top). For increasing  $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

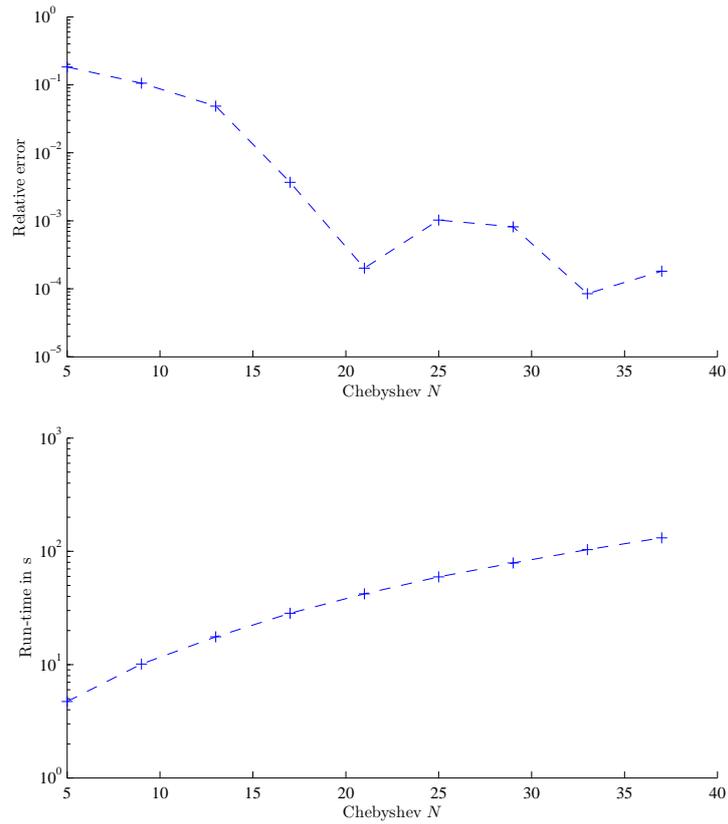


Figure 5.3: Convergence study of Algorithm 6 for applying Fourier techniques (top). For increasing  $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom.

We observe that, while integrating the density function, the error decreases with increasing  $N$ . Similarly, this holds for applying Fourier techniques to derive the conditional expectations. However, the truncated moments method does not appear to be numerically stable. Our implementation is based on the truncated moments given by Kan and Robotti (2016). With increasing  $N$ , the power of  $x^N$  gets too high and the moment is very large. Although the results for truncated moments in Figure 5.1 look rather promising, by increasing  $N$  to 50 the option price literally explodes, explicitly to a value in the region of  $10^{27}$ .

By applying the Chebyshev interpolation, the run-times are directly connected to the number of nodal points. In Figures 5.1, 5.2 and 5.3, we show how the run-times of each method evolve with increasing number of nodal points  $N$ . So far, we conclude

- The dynamic programming framework with Chebyshev interpolation provides reasonable results when the probability density function or Fourier techniques are applied
- numerical stability issues occur by applying the truncated moments method.

### **Application of Algorithm 5**

Now, we apply Algorithm 5, in which the splitting point is derived at each time step. For the Black&Scholes model with configuration (5.35), we use the density function to derive the conditional expectations and Figure 5.4 illustrates the error convergence for an increasing number of nodal points  $N$  and the run-times. In comparison with Figure 5.1, we observe that by applying splitting at each time point, a lower number of nodal points is required to achieve the same accuracy. However, as the run-times in Figure 5.4 illustrate, by applying splitting at each time step, the conditional expectations have to be re-evaluated over different domains at each time step, too. Thus, a pre-computation of the conditional expectations is no longer possible. Therefore, it becomes absolutely crucial to have a fast evaluation technique for the conditional expectation. Additionally, identifying the splitting point at each time step also increases the run-time. In our implementation, we use here the function *fzero* in Matlab.

For this experiment, we conclude,

- with splitting at each time step, we ensure that we are in our theoretical observed framework with respect to the error convergence results
- empirically, the exponential error decay is observed
- the computational cost increases due to the necessity to derive both the conditional expectations and the splitting point at each time step.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

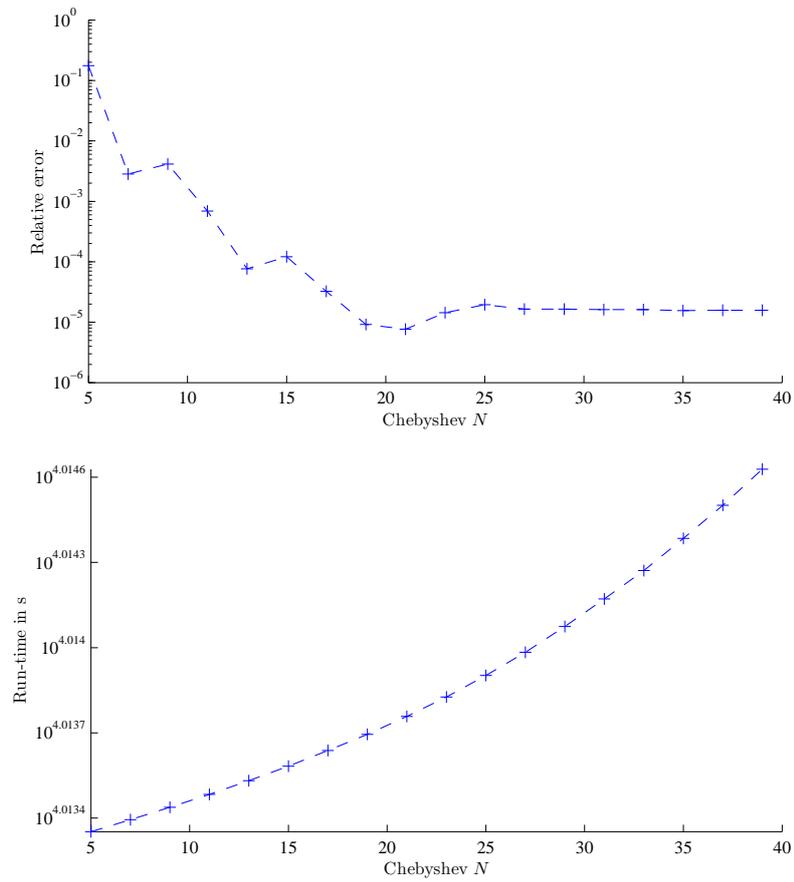


Figure 5.4: Application of Algorithm 5 by applying integration over the density function in the Black&Scholes model. For increasing  $N$  the relative error and the run-time is reported. The cosine method of Fang and Oosterlee (2009) is used as benchmark method.

**Application of Algorithm 9**

As a first step in the experiments here, the integrand of

$$\begin{aligned} \int_{-\infty}^{\infty} e^{-izx} e^{izx_0} \varphi(z) dz &= 2 \int_0^{\infty} \text{Real} (e^{-izx} e^{izx_0} \varphi(z)) dz \\ &\approx 2 \int_0^{1000} \text{Real} (e^{-izx} e^{izx_0} \varphi(z)) dz, \end{aligned}$$

where  $\varphi(z)$  is the characteristic function of the Black&Scholes model, is empirically interpolated in the following parameters,

$$\begin{aligned} \Delta t &\in [0.01, 1], & x, x_0 &\in [\log(0.02), \log(250)], \\ r &\in [0, 0.1], & \sigma &\in [0.1, 0.5]. \end{aligned}$$

The empirical interpolation on  $z = [0, 1000]$  has been stopped for  $M = 1000$  when an empirical accuracy of  $10^{-12}$  has been observed. For a comparison with the cosine benchmark method, we now define a test grid of 121 equidistantly-spaced points on  $S \in [40, 160]$ . Moreover, for parametric option pricing, we define 27 scenarios consisting of all combinations of  $\sigma \in \{0.15, 0.3, 0.45\}$ ,  $r \in \{0.01, 0.03, 0.05\}$  and  $n_T \in \{16, 32, 64\}$ . On the test grid, put option prices can become relatively small. Here, we therefore investigate the absolute error and no longer the relative error. In Figure 5.5, we present the error for increasing  $N$  in the Chebyshev interpolation and the online run-times of the Chebyshev interpolation. We observe that with about  $N = 250$ , we achieve an acceptable accuracy for an online run-time of roughly 17 seconds. The cosine method required a run-time of about 1.13 seconds and is still the faster method in this setting. However, contrary to the application of Algorithm 5, the combination of empirical interpolation and the dynamic Chebyshev interpolation allows us to treat the nodal points as an additional parameters. This, together with the additional interpolation of the characteristic function in the parameters, significantly speeds up the method. Moreover, we can directly make use of the decomposition in offline and online-phase for the parameter dependent 27 scenarios. By applying Algorithm 5, a pre-computation is not possible and the run-times reported in Figure 5.4 have to multiplied by a factor of 27.

## 5 Dynamic Programming Framework with Chebyshev Interpolation

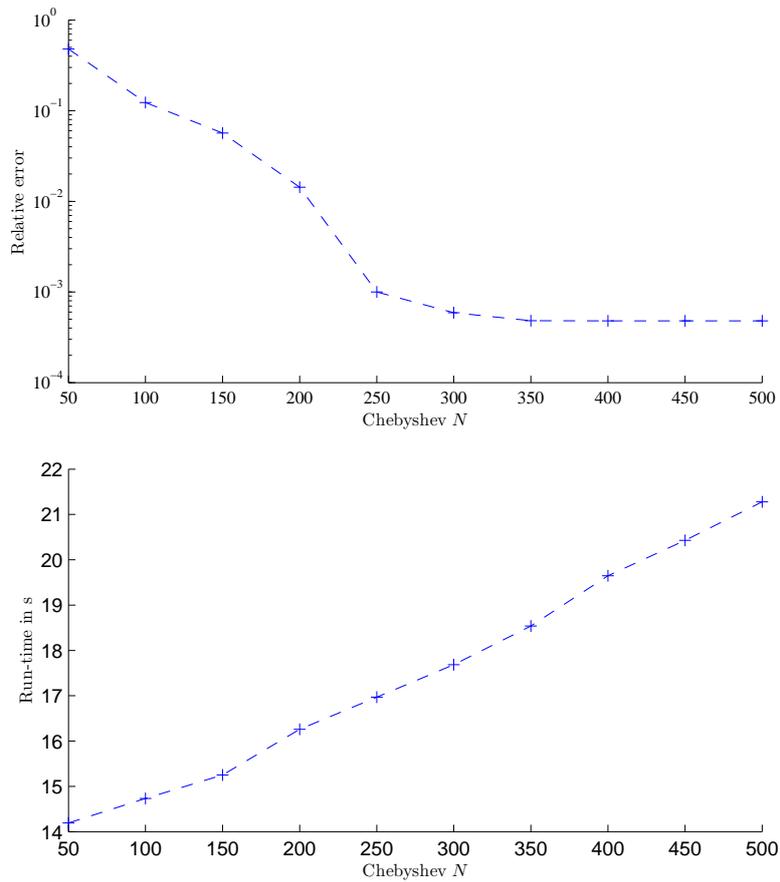


Figure 5.5: Absolute error (top) of applying Algorithm 9 over the test grid and 27 parameter settings for increasing  $N$ . The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-times for the online phase are presented on the bottom.

## 5.7 Conclusion

The combination of dynamic programming and Chebyshev interpolation provides a very general framework for solving DPPs. The driving motivation to investigate this combining idea is the option pricing of American/Bermudan options. However, this approach can be easily extended to further applications in mathematical finance. More complex options, such as swing options, see e.g. Bardou et al. (2009) and Carmona and Touzi (2008), can also be included. Portfolio optimization problems with non-concave utility functions become of special interest, as the problems stated in Carassus and Rásonyi (2015), because due to the non-concave utility functions standard approaches no longer work and a DPP has to be solved. Especially in cases with a non-concave utility function that is analytic, our approach should be suitable because no splitting would be required and, therefore, it is sufficient to derive the conditional expectations once in the pre-computation phase. Moreover, asset liability management problems in discrete time markets as, e.g. in Van Binsbergen and Brandt (2007), can be tackled by solving a DPP. Here, our proposed methodology can be applied.

For the numerical examples, we have chosen a Bermudan option in the Black&Scholes model. So far, the numerical results serve as proof of concept and highlight the generality of the dynamic programming framework with respect to the choice of the technique for solving the conditional expectations. We have tested the integration over the density function, the truncated moment method and Fourier techniques. Only for the truncated moments, we observed some numerical stability issues that have to be investigated in the future.

Comparing Algorithm 6 and Algorithm 5, the dynamic Chebyshev approach in Algorithm 6 is, as regards run-times, the fastest. Thus, this method allows a complete pre-computation of the conditional expectations, however, some more nodal points are required to achieve a desired accuracy. The two concepts of splitting and mollifying for kinks and discontinuities have different advantages and disadvantages. If splitting is required, the splitting point is often time-dependent and changes during the time stepping scheme. Thus, a pre-computation of the conditional expectations is not possible and, at each time step, these have to be derived. This significantly increases the computational costs. The more expensive the derivation of the conditional expectation is, the more significant the run-times increase. Mollifying, on the other side, allows the pre-computation of the conditional expectations. However, the error of replacing the payoff function by a mollified function, for example, evolves through the time stepping scheme.

By focusing on a specific use-case, here the one dimensional American/Bermudan option pricing in the Black&Scholes model, other methods are available that can serve as benchmark method. After the cosine method of Fang and Oosterlee (2009) provided in a benchmark study, see von Sydow et al. (2015), the fastest run-times while maintaining a pre-specified accuracy, we selected this method as the benchmark method. Although we can achieve results with comparable accuracy, all our dynamic Chebyshev algorithms

are, with respect to run-times, slower. Otherwise, the cosine approach is based on an approximation of the characteristic function and thereby, limited to applications, in which the characteristic function can be exploited. To bring the two philosophies from the introduction to full-circle, the dynamic Chebyshev approach is in the spirit of Figure 1.2. Any pricing technique for the derivation of the conditional expectations is sufficient. The cosine approach is a Fourier-based approach and thus in the spirit of Figure 1.1.

So far, in the numerical implementation, we have used a one dimensional underlying. Theoretically, the error convergence results do also hold for higher dimensions. However, analogously to the conclusion of Chapter 4, by increasing the dimension of the tensorized Chebyshev interpolation, the run-times suffer significantly under the curse of dimensionality. Here, a low-rank approximation can be advantageous. Regarding run-times, the time to find the low-rank approximation becomes a crucial factor, because at each time step coefficients for a new Chebyshev interpolation have to be assembled. The coefficients for the tensorized Chebyshev interpolation, contrarily, are known a priori.

Finally, the results shown in Figure 5.5 underline the potential of combining the dynamic Chebyshev interpolation with usage of the parameter dependency via empirical interpolation. Interestingly, the results from Gaß et al. (2016) allow for an arbitrary dimensionality in the parameter space. Thus, the dynamic Chebyshev framework with empirical interpolation can be applied to problems with a high-dimensional parameter space and a lower-dimensional underlying space. The `chebfun` package, see e.g. Driscoll et al. (2014), provides Chebyshev interpolations (including low-rank approximations) for one, two and three dimensions.

An obviously following extension from Algorithm 9, is the application of the empirical interpolation to the integrand resulting from deriving the conditional expectations, i.e. resulting from

$$E \left[ T_j(\tau_{\mathcal{X}}^{-1}(X_{u+1}) | X_u = \tau_{\mathcal{X}}(x^k)) \right].$$

By doing so, additionally the degree  $j$  of the Chebyshev polynomial is incorporated in the empirical interpolation. Thus, the conditional expectation does not have to be computed for each Chebyshev polynomial  $T_0, \dots, T_N$  individually and the run-time is reduced further. This will be investigated in Glau et al. (2017b).

5 *Dynamic Programming Framework with Chebyshev Interpolation*

# A Detailed Results for Effects of de-Americanization on Pricing

The tables in this section present the results to the study of the effects of the de-Americanization methodology on pricing in Section 3.3.2. For the test setting defined in (3.14) and several parameter scenarios, see Table 3.1, American and European put prices are derived and then, the de-Americanized prices are compared to the European prices.

		$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$
$p_1$	$r = 0\%$	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	9.E-5	6.E-5
	$r = 1\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	2.E-4	1.E-4
	$r = 2\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	2.E-4	8.E-5
	$r = 5\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	-3.E-5
	$r = 7\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	-9.E-5
$p_2$	$r = 0\%$	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	1.E-4	1.E-4	7.E-5
	$r = 1\%$	3.E-4	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4
	$r = 2\%$	3.E-4	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	5.E-5
	$r = 5\%$	3.E-4	4.E-4	3.E-4	3.E-4	2.E-4	1.E-4	6.E-5	3.E-4
	$r = 7\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	7.E-5	7.E-5	5.E-4
$p_3$	$r = 0\%$	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	1.E-4	1.E-4	6.E-5
	$r = 1\%$	3.E-4	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4
	$r = 2\%$	3.E-4	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	6.E-5
	$r = 5\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	1.E-4	8.E-5	-2.E-4
	$r = 7\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	1.E-4	-1.E-7	-3.E-4
$p_4$	$r = 0\%$	2.E-4	1.E-4	1.E-4	1.E-4	1.E-4	8.E-5	8.E-5	2.E-4
	$r = 1\%$	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	2.E-4	2.E-4	3.E-4
	$r = 2\%$	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	1.E-4	3.E-4
	$r = 5\%$	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	8.E-5	1.E-4
	$r = 7\%$	4.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	7.E-5	2.E-4
$p_5$	$r = 0\%$	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	1.E-4	9.E-5	4.E-5
	$r = 1\%$	3.E-4	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	6.E-7
	$r = 2\%$	3.E-4	3.E-4	3.E-4	3.E-4	2.E-4	2.E-4	1.E-4	-1.E-4
	$r = 5\%$	4.E-4	3.E-4	3.E-4	3.E-4	2.E-4	1.E-4	-7.E-6	-5.E-4
	$r = 7\%$	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4	4.E-5	-1.E-4	-8.E-4

Table A.1: De-Americanization effects on pricing put options in the CEV model - average error between the de-Americanized and European prices for each maturity.

A Detailed Results for Effects of de-Americanization on Pricing

		0.80	0.85	0.90	0.95	1.00	1.05	1.10	1.15	1.20
$p_1$	$r = 0\%$	4.E-5	7.E-5	1.E-4	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	8.E-5
	$r = 1\%$	8.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	2.E-4	2.E-4
	$r = 2\%$	8.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	2.E-4	2.E-4
	$r = 5\%$	7.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	1.E-4
	$r = 7\%$	7.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	1.E-4	2.E-4	8.E-5
$p_2$	$r = 0\%$	5.E-5	9.E-5	1.E-4	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	8.E-5
	$r = 1\%$	9.E-5	2.E-4	3.E-4	3.E-4	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4
	$r = 2\%$	8.E-5	2.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	2.E-4	2.E-4
	$r = 5\%$	6.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4
	$r = 7\%$	6.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4
$p_3$	$r = 0\%$	4.E-5	7.E-5	1.E-4	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	8.E-5
	$r = 1\%$	7.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4
	$r = 2\%$	7.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	3.E-4	2.E-4
	$r = 5\%$	5.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	3.E-4	2.E-4	6.E-5
	$r = 7\%$	3.E-5	9.E-5	2.E-4	2.E-4	3.E-4	3.E-4	2.E-4	1.E-4	-2.E-5
$p_4$	$r = 0\%$	4.E-5	7.E-5	1.E-4	1.E-4	2.E-4	2.E-4	1.E-4	1.E-4	9.E-5
	$r = 1\%$	9.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	3.E-4	2.E-4
	$r = 2\%$	8.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	1.E-4
	$r = 5\%$	9.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	-3.E-5
	$r = 7\%$	9.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	-1.E-4
$p_5$	$r = 0\%$	3.E-5	7.E-5	1.E-4	2.E-4	2.E-4	2.E-4	1.E-4	1.E-4	7.E-5
	$r = 1\%$	6.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	1.E-4
	$r = 2\%$	5.E-5	1.E-4	2.E-4	3.E-4	4.E-4	4.E-4	2.E-4	2.E-4	1.E-4
	$r = 5\%$	2.E-5	7.E-5	1.E-4	2.E-4	3.E-4	3.E-4	1.E-4	1.E-4	-9.E-6
	$r = 7\%$	4.E-5	4.E-5	1.E-4	2.E-4	2.E-4	2.E-4	4.E-5	4.E-5	-1.E-4

Table A.2: De-Americanization effects on pricing put options in the CEV model - average error between the de-Americanized and European prices for each strike.

		$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$
$p_1$	$r = 0\%$	0.055	0.104	0.108	0.154	0.204	0.211	0.218	0.244
	$r = 1\%$	0.055	0.102	0.106	0.151	0.155	0.203	0.208	0.226
	$r = 2\%$	0.054	0.101	0.104	0.107	0.151	0.196	0.198	0.208
	$r = 5\%$	0.052	0.057	0.097	0.098	0.100	0.136	0.136	0.161
	$r = 7\%$	0.051	0.054	0.057	0.093	0.093	0.093	0.093	0.109
$p_2$	$r = 0\%$	0.103	0.154	0.204	0.208	0.217	0.230	0.242	0.283
	$r = 1\%$	0.103	0.152	0.201	0.204	0.212	0.223	0.233	0.265
	$r = 2\%$	0.102	0.109	0.154	0.201	0.207	0.216	0.224	0.248
	$r = 5\%$	0.099	0.104	0.147	0.150	0.192	0.195	0.198	0.203
	$r = 7\%$	0.058	0.101	0.105	0.144	0.146	0.182	0.181	0.176
$p_3$	$r = 0\%$	0.152	0.205	0.212	0.219	0.233	0.252	0.269	0.320
	$r = 1\%$	0.108	0.203	0.209	0.216	0.228	0.245	0.260	0.303
	$r = 2\%$	0.107	0.201	0.207	0.213	0.224	0.239	0.251	0.287
	$r = 5\%$	0.105	0.153	0.199	0.203	0.210	0.219	0.226	0.241
	$r = 7\%$	0.103	0.150	0.194	0.196	0.201	0.206	0.210	0.214
$p_4$	$r = 0\%$	0.156	0.212	0.223	0.233	0.252	0.276	0.297	0.352
	$r = 1\%$	0.155	0.210	0.220	0.230	0.247	0.270	0.288	0.336
	$r = 2\%$	0.154	0.208	0.218	0.227	0.243	0.263	0.279	0.319
	$r = 5\%$	0.152	0.203	0.210	0.217	0.229	0.244	0.255	0.275
	$r = 7\%$	0.150	0.200	0.206	0.211	0.221	0.232	0.239	0.247
$p_5$	$r = 0\%$	0.205	0.220	0.235	0.248	0.272	0.300	0.323	0.377
	$r = 1\%$	0.205	0.219	0.232	0.245	0.267	0.294	0.314	0.360
	$r = 2\%$	0.204	0.217	0.230	0.242	0.263	0.287	0.306	0.345
	$r = 5\%$	0.201	0.212	0.223	0.233	0.250	0.268	0.281	0.300
	$r = 7\%$	0.156	0.209	0.218	0.227	0.241	0.256	0.266	0.273

Table A.3: De-Americanization effects on pricing put options in the CEV model - maximal European put prices.

A Detailed Results for Effects of de-Americanization on Pricing

		$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$
$p_1$	$r = 0\%$	4.E-5	2.E-5	2.E-5	-3.E-6	9.E-6	4.E-5	-3.E-5	3.E-5
	$r = 1\%$	-2.E-4	-3.E-4	-4.E-4	-4.E-4	-5.E-4	-8.E-4	-1.E-3	-2.E-3
	$r = 2\%$	-4.E-4	-7.E-4	-9.E-4	-9.E-4	-1.E-3	-2.E-3	-2.E-3	-4.E-3
	$r = 5\%$	-2.E-3	-2.E-3	-3.E-3	-2.E-3	-3.E-3	-4.E-3	-5.E-3	-6.E-3
	$r = 7\%$	-3.E-3	-4.E-3	-4.E-3	-3.E-3	-4.E-3	-4.E-3	-5.E-3	-4.E-3
$p_2$	$r = 0\%$	3.E-5	8.E-5	5.E-5	2.E-5	-4.E-5	-3.E-4	-2.E-5	5.E-5
	$r = 1\%$	-6.E-5	-3.E-3	-2.E-3	-2.E-3	-1.E-2	-6.E-3	-6.E-3	-2.E-2
	$r = 2\%$	-2.E-4	-8.E-3	-5.E-3	-4.E-3	-4.E-3	-3.E-2	-2.E-2	-2.E-2
	$r = 5\%$	-7.E-4	-1.E-3	-1.E-3	-2.E-2	-2.E-2	-2.E-2	-2.E-2	-2.E-2
	$r = 7\%$	-1.E-3	-1.E-3	-2.E-3	-2.E-3	-3.E-3	-3.E-2	-3.E-2	-3.E-2
$p_3$	$r = 0\%$	3.E-5	1.E-4	2.E-5	3.E-4	-2.E-4	2.E-4	-1.E-4	2.E-6
	$r = 1\%$	-5.E-4	-3.E-3	-2.E-3	-2.E-3	-2.E-2	-8.E-3	-7.E-3	-9.E-3
	$r = 2\%$	-1.E-3	-1.E-2	-5.E-3	-4.E-3	-2.E-2	-3.E-2	-2.E-2	-2.E-2
	$r = 5\%$	-5.E-3	-4.E-3	-4.E-3	-2.E-2	-1.E-2	-1.E-2	-2.E-2	-6.E-2
	$r = 7\%$	-1.E-2	-6.E-3	-6.E-3	-6.E-3	-3.E-2	-2.E-2	-3.E-2	-4.E-2
$p_4$	$r = 0\%$	-2.E-5	-2.E-4	3.E-5	2.E-4	2.E-4	6.E-5	-3.E-07	1.E-4
	$r = 1\%$	-7.E-4	-7.E-3	-3.E-3	-2.E-3	-2.E-3	-3.E-3	-3.E-3	-3.E-3
	$r = 2\%$	-2.E-3	-7.E-3	-8.E-3	-6.E-3	-5.E-3	-6.E-3	-7.E-3	-9.E-3
	$r = 5\%$	-7.E-3	-4.E-3	-2.E-2	-3.E-2	-2.E-2	-2.E-2	-2.E-2	-3.E-2
	$r = 7\%$	-1.E-2	-7.E-3	-4.E-2	-2.E-2	-4.E-2	-3.E-2	-3.E-2	-5.E-2
$p_5$	$r = 0\%$	2.E-4	-1.E-4	-4.E-4	4.E-4	3.E-4	-1.E-4	-6.E-4	-7.E-4
	$r = 1\%$	-2.E-3	-7.E-4	-1.E-3	-3.E-4	-6.E-4	-5.E-4	2.E-3	8.E-2
	$r = 2\%$	-4.E-3	-2.E-3	-2.E-3	-2.E-3	-2.E-3	-2.E-3	2.E-3	1.E-1
	$r = 5\%$	-7.E-3	-6.E-3	-5.E-3	-6.E-3	-6.E-3	-8.E-3	-6.E-3	1.E-1
	$r = 7\%$	-1.E-2	-1.E-2	-8.E-3	-9.E-3	-9.E-3	-1.E-2	-1.E-2	1.E-1

Table A.4: Additional test for  $S = 100$ . De-Americanization effects on pricing put options in the CEV model - average error between De-Americanized and European prices for each maturity.

		$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$
$p_1$	$r = 0\%$	19.88	19.88	19.88	19.88	19.88	19.88	19.88	19.94
	$r = 1\%$	19.78	19.68	19.58	19.48	19.28	18.98	18.69	17.62
	$r = 2\%$	19.68	19.48	19.28	19.08	18.68	18.09	17.51	15.38
	$r = 5\%$	19.38	18.88	18.38	17.89	16.91	15.46	14.06	9.33
	$r = 7\%$	19.18	18.48	17.79	17.11	15.75	13.75	11.86	6.05
$p_2$	$r = 0\%$	19.88	19.88	19.88	19.88	19.90	19.98	20.10	20.82
	$r = 1\%$	19.78	19.68	19.58	19.48	19.31	19.10	18.96	18.72
	$r = 2\%$	19.68	19.48	19.28	19.08	18.72	18.24	17.84	16.72
	$r = 5\%$	19.38	18.88	18.39	17.90	16.97	15.72	14.64	11.43
	$r = 7\%$	19.18	18.48	17.80	17.12	15.83	14.12	12.65	8.52
$p_3$	$r = 0\%$	19.88	19.89	19.94	20.03	20.29	20.78	21.32	23.44
	$r = 1\%$	19.78	19.69	19.64	19.64	19.73	19.98	20.28	21.55
	$r = 2\%$	19.68	19.49	19.35	19.26	19.17	19.19	19.28	19.76
	$r = 5\%$	19.38	18.90	18.48	18.12	17.55	16.91	16.41	14.94
	$r = 7\%$	19.18	18.50	17.90	17.37	16.50	15.47	14.64	12.19
$p_4$	$r = 0\%$	19.90	20.11	20.45	20.86	21.71	22.97	24.15	28.11
	$r = 1\%$	19.80	19.92	20.18	20.50	21.20	22.24	23.21	26.35
	$r = 2\%$	19.70	19.73	19.90	20.14	20.69	21.52	22.29	24.67
	$r = 5\%$	19.41	19.16	19.09	19.10	19.22	19.46	19.67	20.07
	$r = 7\%$	19.21	18.78	18.55	18.42	18.27	18.15	18.04	17.37
$p_5$	$r = 0\%$	20.17	21.03	21.98	22.91	24.63	26.88	28.81	34.23
	$r = 1\%$	20.08	20.86	21.73	22.58	24.16	26.21	27.93	32.55
	$r = 2\%$	19.99	20.68	21.48	22.26	23.70	25.54	27.07	30.93
	$r = 5\%$	19.70	20.16	20.73	21.30	22.34	23.61	24.59	26.42
	$r = 7\%$	19.51	19.81	20.25	20.68	21.47	22.38	23.02	23.70

Table A.5: Additional test for  $S = 100$ . De-Americanization effects on pricing put options in the CEV model - maximal European put prices.

*A Detailed Results for Effects of de-Americanization on Pricing*

# Bibliography

- Achdou, Y. and O. Pironneau (2005). *Computational methods for option pricing*, Volume 30. SIAM.
- Ackerer, D., D. Filipović, and S. Pulido (2016). The jacobi stochastic volatility model. Preprint, SSRN: 2782486, [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2782486](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2782486).
- Anderson, H. (1986). Metropolis, Monte Carlo and the maniac. *Los alamos Science* 14, 96–108.
- Andrieu, C., N. De Freitas, A. Doucet, and M. I. Jordan (2003). An introduction to MCMC for machine learning. *Machine learning* 50(1), 5–43.
- Armenti, Y., S. Crépey, S. Drapeau, and A. Papapantoleon (2015). Multivariate Shortfall Risk Allocation. Preprint, ArXiv: 1507.05351, <https://arxiv.org/abs/1507.05351>.
- Balajewicz, M. and J. Toivanen (2016). Reduced order models for pricing American options under stochastic volatility and jump-diffusion models. *Procedia Computer Science* 80, 734–743. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA.
- Ballestra, L. V. and L. Cecere (2016). A numerical method to estimate the parameters of the CEV model implied by American option prices: Evidence from NYSE. *Chaos, Solitons & Fractals* 88, 100–106.
- Bardou, O., S. Bouthemy, and G. Pages (2009). Optimal quantization for the pricing of swing options. *Applied Mathematical Finance* 16(2), 183–217.
- Barndorff-Nielsen, O. E. and N. Shephard (2001). Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society, Series B* 63, 167–241.
- Barone-Adesi, G. (2005). The saga of the American put. *Journal of Banking & Finance* 29(11), 2909–2918.
- Barrault, M., Y. Maday, N. C. Nguyen, and A. T. Patera (2004). An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique* 339(9), 667–672.
- Bauer, H. (1991). *Wahrscheinlichkeitstheorie*. de Gruyter.

## Bibliography

- Bellman, R. E. (1957). Dynamic programming. *Princeton University Press*, 151.
- Bernstein, S. N. (1912). Sur l'ordre de la meilleure approximation des fonctions continues par des polynomes de degré donné, mmoires acad. *Académie Royale de Belgique. Classe des Sciences. Mémoires 4*.
- Bingham, N. H. and R. Kiesel (2004). *Risk-neutral valuation: pricing and hedging of financial derivatives, second edition*. Springer Science & Business Media.
- Black, F. and M. Scholes (1973). The pricing of options and other liabilities. *Journal of Political Economy* 81, 637–654.
- Börm, S. (2010). *Efficient numerical methods for non-local operators: H2-matrix compression, algorithms and analysis*, Volume 14. European Mathematical Society.
- Brennan, M. J. and E. S. Schwartz (1977). The valuation of American put options. *The Journal of Finance* 2(32), 449–462.
- Broadie, M. and J. Detemple (1996). American option valuation: new bounds, approximations, and a comparison of existing methods. *Review of Financial Studies* 9(4), 1211–1250.
- Burkovska, O., M. Gaß, K. Glau, M. Mahlstedt, W. Schoutens, and B. Wohlmuth (2016). Calibration to American options: numerical investigation of the de-Americanization methodology. Preprint, ArXiv: 1611.06181, <https://arxiv.org/abs/1611.06181>.
- Burkovska, O., K. Glau, M. Mahlstedt, and B. Wohlmuth (2016b). Model reduction for calibration of American options. Preprint, ArXiv: 1611.06452, <https://arxiv.org/abs/1611.06452>.
- Burkovska, O., B. Haasdonk, J. Salomon, and B. Wohlmuth (2015). Reduced basis methods for pricing options with the Black-Scholes and Heston models. *SIAM Journal on Financial Mathematics* 6(1), 685–712.
- Canuto, C. and A. Quarteroni (1982). Approximation results for orthogonal polynomials in Sobolev spaces. *Mathematics of Computation* 38(157), 67–86.
- Carassus, L. and M. Rásonyi (2015). Maximization of nonconcave utility functions in discrete-time financial market models. *Mathematics of Operations Research* 41(1), 146–173.
- Carmona, R. and N. Touzi (2008). Optimal multiple stopping and valuation of swing options. *Mathematical Finance* 18(2), 239–268.
- Carr, P., H. Geman, D. B. Madan, and M. Yor (2003). Stochastic volatility for Lévy processes. *Mathematical Finance* 13, 345–382.
- Carr, P. and D. B. Madan (1999). Option valuation and the fast Fourier transform. *Journal of Computational Finance* 2(4), 61–73.

## Bibliography

- Carr, P. and L. Wu (2010). Stock options and credit default swaps: a joint framework for valuation and estimation. *Journal of Financial Econometrics* 8(4), 409–449.
- CBOE (2009). The CBOE volatility index - VIX. *CBOE*.
- Cont, R., N. Lantos, and O. Pironneau (2011). A reduced basis for option pricing. *SIAM Journal on Financial Mathematics* 2(1), 287–316.
- Cox, J. C. (1975). Notes on option pricing i: constant elasticity of diffusions. Unpublished draft, Stanford University.
- Cox, J. C., S. A. Ross, and M. Rubinstein (1979). Option pricing: a simplified approach. *Journal of Financial Economics* 7(3), 229–263.
- Cuchiero, C., M. Keller-Ressel, and J. Teichmann (2012). Polynomial processes and their applications to mathematical Finance. *Finance and Stochastics* 4(16).
- Davis, P. J. (1975). *Interpolation and approximation*. Courier Corporation.
- Detlefsen, K. and W. Haerdle (2006). Calibration risk for exotic options. *SFB 649 Discussion Paper 2006-001*.
- Dominguez, V., I. Graham, and V. Smyshlyaev (2011). Stability and error estimates for Filon–Clenshaw–Curtis rules for highly oscillatory integrals. *IMA Journal of Numerical Analysis* 31(4), 1253–1280.
- Driscoll, T. A., N. Hale, and L. N. Trefethen (2014). Chebfun guide. [http://www.chebfun.org/docs/guide/chebfun\\_guide.pdf](http://www.chebfun.org/docs/guide/chebfun_guide.pdf).
- Duffie, D., D. Filipović, and W. Schachermayer (2003). Affine processes and applications in finance. *Annals of Applied Probability* 13(3), 984–1053.
- Eberlein, E., K. Glau, and A. Papapantoleon (2010). Analysis of Fourier transform valuation formulas and applications. *Applied Mathematical Finance* 17(3), 211–240.
- Eberlein, E., U. Keller, and K. Prause (1998). New insights into smile, mispricing and value at risk: the hyperbolic model. *Journal of Business* 71(3), 371–405.
- Fang, F. and C. W. Oosterlee (2008). A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing* 31(2), 826–848.
- Fang, F. and C. W. Oosterlee (2009). Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik* 114(1), 27.
- Feng, L. and V. Linetsky (2008). Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: A fast Hilbert transform approach. *Mathematical Finance* 18(3), 337–384.

## Bibliography

- Fengler, M. R. (2005). Arbitrage-free smoothing of the implied volatility surface. *SFB 649 Discussion Paper 2005-019*.
- Gaß, M. (2016). *PIDE methods and concepts for parametric option pricing*. Dissertation, Technische Universität München.
- Gaß, M. and K. Glau (2015). Parametric integration by magic point empirical interpolation. Preprint, ArXiv: 1511.08510, <https://arxiv.org/abs/1511.08510>.
- Gaß, M., K. Glau, M. Mahlstedt, and M. Mair (2016). Chebyshev Interpolation for Parametric Option Pricing. Preprint (first version 2015), ArXiv:1505.04648v2, <https://arxiv.org/abs/1505.04648v2>.
- Gaß, M., K. Glau, and M. Mair (2016). Magic Points Fourier Transform Option Pricing – Polynomial and Empirical Interpolation. Preprint (first version 2015), ArXiv: 1511.00884, <https://arxiv.org/abs/1511.00884>.
- Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica 24*, 259.
- Glas, S. and K. Urban (2014). On noncoercive variational inequalities. *SIAM Journal on Numerical Analysis 52*(5), 2250–2271.
- Glasserman, P. (2003). *Monte Carlo methods in financial engineering*, Volume 53. Springer Science & Business Media.
- Glau, K. (2016). Classification of Lévy processes with parabolic Kolmogorov backward equations. *Theory of Probability & Its Applications 60*(3), 383–406.
- Glau, K., B. Hashemi, M. Mahlstedt, and C. Pötz (2017). Spread option in 2D Black-Scholes. Example for Chebfun3 in the Chebfun toolbox, <http://www.chebfun.org/examples/applics/BlackScholes2D.html>.
- Glau, K. and M. Mahlstedt (2016). Improved error bound for multivariate Chebyshev polynomial interpolation. Preprint, ArXiv: 1611.08706, <https://arxiv.org/abs/1611.08706>.
- Glau, K., M. Mahlstedt, and C. Pötz (2017a). Dynamic Chebyshev interpolation for non-analytic value functions. Work in progress.
- Glau, K., M. Mahlstedt, and C. Pötz (2017b). Dynamic programming principle with Chebyshev interpolation. Work in progress.
- Gronwall, T. H. (1919). Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics 20*(4), 292–296.
- Haasdonk, B., J. Salomon, and B. Wohlmuth (2013). A reduced basis method for the simulation of American options. In *Numerical Mathematics and Advanced Applications 2011*, pp. 821–829. Springer.

## Bibliography

- Haring, S. and R. Hochreiter (2015). Efficient and robust calibration of the Heston option pricing model for American options using an improved Cuckoo Search Algorithm. Preprint, ArXiv: 1507.08937, <https://arxiv.org/abs/1507.08937>.
- Heinrich, S. (1998). Monte Carlo complexity of global solution of integral equations. *Journal of Complexity* 14(2), 151–175.
- Heinrich, S. (2001). Multilevel Monte Carlo methods. In *International Conference on Large-Scale Scientific Computing*, pp. 58–67. Springer.
- Heinrich, S. and E. Sindambiwe (1999). Monte Carlo complexity of parametric integration. *Journal of Complexity* 15(3), 317–341.
- Hesthaven, J. S., G. Rozza, and B. Stamm (2016). *Certified reduced basis methods for parametrized partial differential equations*. SpringerBriefs in Mathematics. Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao. BCAM SpringerBriefs.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* 6(2), 327–343.
- Jacod, J. and A. N. Shiryaev (2003). *Limit theorems for stochastic processes, second edition*. Berlin, Heidelberg, Germany: Springer.
- Jänich, K. (2004). *Funktionentheorie: eine Einführung*. Springer-Lehrbuch. Springer.
- Kallsen, J. (2006). A didactic note on affine stochastic volatility models. In Y. Kabanov, R. Lipster, and J. Stoyanov (Eds.), *From Stochastic Calculus to Mathematical Finance: The Shiryaev Festschrift*, pp. 343–368. Springer.
- Kan, R. and C. Robotti (2016). On moments of folded and truncated multivariate normal distributions. *Working Paper*.
- Karatzas, I. and S. E. Shreve (1996). *Brownian motion and stochastic calculus, second edition*. New York, USA: Springer.
- Kim, I. J. (1990). The analytic valuation of American options. *Review of financial studies* 3(4), 547–572.
- Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review* 51(3), 455–500.
- Kudryavtsev, O. and S. Z. Levendorskiĭ (2009). Fast and accurate pricing of barrier options under Lévy processes. *Finance and Stochastics* 13(4), 531–562.
- L’Ecuyer, P. (2009). Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics* 13(3), 307–349.
- Lee, R. W. (2004). Option pricing by transform methods: extensions, unification, and error control. *Journal of Computational Finance* 7(3), 51–86.

## Bibliography

- Longstaff, F. A. and E. S. Schwartz (2001). Valuing American options by simulation: a simple least-squares approach. *Review of Financial studies* 14(1), 113–147.
- Lord, R., F. Fang, F. Bervoets, and C. W. Oosterlee (2008). A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing* 30(4), 1678–1705.
- Maday, Y., C. Nguyen, P. A.T., and G. Pau (2009). A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis* 8(1), 383–404.
- McKean, H. P. (1965). Appendix: A free boundary problem for the heat equation arising from a problem in mathematical economics. *Sloan Management Review* 6(2), 32.
- Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science* 4(1), 141–183.
- Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125–144.
- Metropolis, N. (1987). The beginning of the Monte Carlo method. pp. 125–130. Los Alamos Science. Special Issue dedicated to Stanislaw Ulam.
- Müller, A. and D. Stoyan (2002). *Comparison methods for stochastic models and risks*, Volume 389. Wiley.
- Musiela, M. and M. Rutkowski (2006). *Martingale methods in financial modelling*, Volume 36. Springer.
- Øksendal, B. (2003). *Stochastic differential equations, sixth edition*. Springer.
- Pachon, R. (2016). Numerical pricing of European options with arbitrary payoffs. Preprint Available at SSRN: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2712402](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2712402).
- Patera, A. T. and G. Rozza (2006). Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Technical Report Version 1.0, MIT 2006–2007, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering, Massachusetts Institute of Technology.
- Peherstorfer, B., P. Gómez, and H.-J. Bungartz (2015). Reduced models for sparse grid discretizations of the multi-asset Black-Scholes equation. *Advances in Computational Mathematics* 41(5), 1365–1389.
- Peskir, G. and A. Shiryaev (2006). *Optimal stopping and free-boundary problems*. Springer.
- Pironneau, O. (2009). Calibration of options on a reduced basis. *Journal of Computational and Applied Mathematics* 232(1), 139–147.

## Bibliography

- Pironneau, O. (2011). Reduced basis for vanilla and basket options. *Risk and Decision Analysis* 2(4), 185–194.
- Pironneau, O. (2012). Proper orthogonal decomposition for pricing options. *Journal of Computational Finance* 16(1), 33–46.
- Pistorius, M. and J. Stolte (2012). Fast computation of vanilla prices in time-changed models and implied volatilities. *International Journal of Theoretical and Applied Finance* (15), 1250031.
- Quarteroni, A., A. Manzoni, and F. Negri (2016). *Reduced basis methods for partial differential equations*, Volume 92 of *Unitext*. Springer, Cham. An introduction, *La Matematica per il 3+2*.
- Raible, S. (2000). *Lévy processes in finance: theory, numerics, and empirical facts*. Dissertation, Universität Freiburg.
- Revuz, D. and M. Yor (1999). *Continuous martingales and brownian motion, third edition*. Berlin, Heidelberg, Germany: Springer.
- Rivlin, T.-J. (1990). *Chebyshev polynomials*. John Wiley & Sons, Inc., copublished in the United States with John Wiley & Sons.
- Rudin, W. (1987). *Real and complex analysis*. Tata McGraw-Hill Education.
- Runge, C. (1901). Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik* 46(224-243), 20.
- Sachs, E. W. and M. Schneider (2014). Reduced-order models for the implied variance under local volatility. *International Journal of Theoretical and Applied Finance* 17(8), 1450053, 23.
- Sachs, E. W., M. Schneider, and M. Schu (2014). Adaptive trust-region POD methods in PIDE-constrained optimization. In *Trends in PDE constrained optimization*, Volume 165 of *International Series of Numerical Mathematics*, pp. 327–342. Birkhäuser/Springer, Cham.
- Sachs, E. W. and M. Schu (2008). Reduced order models (POD) for calibration problems in finance. In K. Kunisch, G. Of, and O. Steinbach (Eds.), *Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2007, the 7th European Conference on Numerical Mathematics and Advanced Applications, Graz, Austria, September 2007*, pp. 735–742. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Sachs, E. W. and M. Schu (2010). Reduced order models in PIDE constrained optimization. *Control and Cybernetics* 39(3), 661–675.
- Samuelson, P. A. (1965). Rational theory of warrant pricing. *IMR; Industrial Management Review (pre-1986)* 6(2), 13.

## Bibliography

- Sauter, S. and C. Schwab (2004). *Randelementmethoden: Analyse, Numerik und Implementierung schneller Algorithmen*. Vieweg+Teubner Verlag.
- Schoutens, W., E. Simons, and J. Tistaert (2004). A perfect calibration! Now what? *Wilmott Magazine*.
- Schroeder, M. (1989). Computing the constant elasticity of variance option pricing formula. *Journal of Finance* 44(1), 211–219.
- Schweizer, M. (2002). On bermudan options. In *Advances in Finance and Stochastics*, pp. 257–270. Springer.
- Seydel, R. (2012). *Tools for computational finance* (5 ed.). Springer Science & Business Media.
- Skorokhod, A. (1965). *Studies in the theory of random processes*. Reading, Massachusetts, USA: Addison-Wesley Publishing Company, Inc.
- Tadmor, E. (1986). The exponential accuracy of Fourier and Chebyshev differencing methods. *SIAM Journal on Numerical Analysis* 23(1), 1–10.
- Trefethen, L. N. (2013). *Approximation theory and approximation practice*. SIAM books.
- Trefethen, L. N. (2016). Multivariate polynomial approximation in the hypercube. Preprint, ArXiv: 1608.02216, <https://arxiv.org/abs/1608.02216>.
- Van Binsbergen, J. H. and M. W. Brandt (2007). Optimal asset allocation in asset liability management. Technical report, National Bureau of Economic Research.
- Van der Hoek, J. and R. J. Elliott (2006). *Binomial models in finance*. Springer Science & Business Media.
- von Sydow, L., L. Josef Höök, E. Larsson, E. Lindström, S. Milovanović, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Sirén, J. Toivanen, et al. (2015). BENCHOP—The BENCHmarking project in option pricing. *International Journal of Computer Mathematics* 92(12), 2361–2379.
- Wald, A. (1947). *Sequential analysis. 1947*. New York: Wiley.
- Wloka, J. (1987). *Partial differential equations*. Cambridge University Press.
- Zagst, R. (2002). *Interest rate management, first edition*. Berlin: Springer-Verlag.

# List of Tables

3.1	Overview of the parameter sets used for the CEV model . . . . .	43
3.2	Calibration results for calibrating to put options only and out-of-the-money options for the CEV model. Due to the effect of non-unique de-Americanization results, for the CEV model, some option prices have been neglected in the calibration to de-Americanized option data, as Remark 3.3.2 explains. In scenarios $p_1$ to $p_5$ , 5, 5, 10, 10 and 10 prices were excluded in the calibration to put options only. . . . .	49
3.3	Processed Google option data for $t_0 = 02.02.2015$ , $S_0 = 523.76$ . . . . .	50
3.4	Calibration results for calibrating to out-of-the-money put and call options combined. . . . .	50
3.5	Overview of prices for barrier and lookback options . . . . .	52
4.1	Parametrization of models, basket and path-dependent options. The model parameters are given for $j = 1, \dots, d$ to reflect the multivariate setting with free parameters strike $K$ and maturity $T$ . . . . .	106
4.2	Interpolation of exotic options with Chebyshev interpolation. $N = 5$ and $d = 5$ in all cases. In addition to the $L^\infty$ errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the $L^\infty$ error is realized. . . . .	107
4.3	Interpolation of exotic options with Chebyshev interpolation. $N = 10$ and $d = 5$ in all cases. In addition to the $L^\infty$ errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the $L^\infty$ error is realized. . . . .	107
4.4	Interpolation of exotic options with Chebyshev interpolation. $N = 30$ and $d = 5$ in all cases. In addition to the $L^\infty$ errors the table displays the Monte-Carlo (MC) prices, the Monte-Carlo confidence bounds and the Chebyshev Interpolation (CI) prices for those parameters at which the $L^\infty$ error is realized. . . . .	108
4.5	Interpolation of one-dimensional American puts with Chebyshev interpolation in the Black&Scholes model. In addition to the $L^\infty$ errors the table displays the Finite Differences (FD) prices and the Chebyshev Interpolation (CI) prices for those parameters at which the $L^\infty$ error is realized. . .	108

*List of Tables*

4.6	Interpolation of multivariate lookback options with Chebyshev interpolation for $N = 6$ based on an enriched Monte-Carlo setting with $5 \cdot 10^6$ sample paths, antithetic variates, and 400 time steps per year. In addition to the $L^\infty$ error on the test grid, we also report the Monte-Carlo (MC) price, the Monte-Carlo confidence bound, and the Chebyshev Interpolation (CI) price for the parameters at which the $L^\infty$ error is realized. We observe that the accuracy of the Chebyshev interpolation for $N = 6$ is roughly in the same range as the accuracy of the benchmark Monte-Carlo setting (worst-case confidence bound of $1.644 \cdot 10^{-2}$ and worst-case error of $7.361 \cdot 10^{-3}$ ). . . . .	112
4.7	Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings. Here, we vary one model parameter and compare the Chebyshev results to Monte-Carlo. Both methods have been set up to deliver comparable accuracies. As the number of computed prices increases, the Chebyshev algorithm increasingly profits from the initial investment of the offline phase. . . . .	113
4.8	Interpolation of multivariate lookback options with Chebyshev interpolation for $N = 6$ based on an enriched Monte-Carlo setting with $5 \cdot 10^6$ sample paths, antithetic variates, and 400 time steps per year. In addition to the $L^\infty$ error on the test grid, we also report the Monte-Carlo (MC) price, the Monte-Carlo confidence bound, and the Chebyshev Interpolation (CI) price for the parameters at which the $L^\infty$ error is realized. We observe that the accuracy of the Chebyshev interpolation $N = 6$ is roughly in the same range as the accuracy of the benchmark Monte-Carlo setting (worst-case confidence bound of $6.783 \cdot 10^{-2}$ and worst-case error of $2.791 \cdot 10^{-2}$ ). . . . .	114
4.9	Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings. Here, we vary two model parameters and compare the Chebyshev results to Monte-Carlo. Both methods have been set up to deliver comparable accuracies. As the number of computed prices increases, the Chebyshev algorithm increasingly profits from the initial investment of the offline phase. . . . .	114
A.1	De-Americanization effects on pricing put options in the CEV model - average error between the de-Americanized and European prices for each maturity. . . . .	173
A.2	De-Americanization effects on pricing put options in the CEV model - average error between the de-Americanized and European prices for each strike. . . . .	174
A.3	De-Americanization effects on pricing put options in the CEV model - maximal European put prices. . . . .	174

*List of Tables*

A.4	Additional test for $S = 100$ . De-Americanization effects on pricing put options in the CEV model - average error between De-Americanized and European prices for each maturity. . . . .	175
A.5	Additional test for $S = 100$ . De-Americanization effects on pricing put options in the CEV model - maximal European put prices. . . . .	175

# List of Figures

1.1	Schematic overview: Both option pricing techniques, FFT (add-on to Fourier pricing) and reduced basis (add-on to a PDE technique), exploit the parameter dependency as an add-on to the functional architecture of the underlying pricing technique. . . . .	13
1.2	Idea of exploiting parameter dependencies independently of the underlying pricing technique. The answer in this thesis will be Chebyshev polynomial interpolation. The pricing techniques of PDE methods, Fourier pricing and Monte-Carlo simulation are only applied during the offline phase. . . . .	13
2.1	Illustration of a Bernstein ellipse with foci at $\pm 1$ . The sum of the connection of each point on the ellipse with the two foci is exactly $\varrho$ . We see that semimajor $a$ and semiminor $b$ of the ellipse are summing up to the radius of the ellipse $\varrho$ . . . . .	26
3.1	De-Americanization scheme: American option prices are transferred into European prices before the calibration process itself is started. We investigate the effects of de-Americanization by comparing the results to directly calibrating American options. . . . .	31
3.2	Illustration of hat functions $\varphi_i, i = 2, \dots, 14$ , over a node grid with 15 nodes.	41
3.3	Given an American put option price of 20 with $S_0 = 100$ , $K = 120$ , $r = 0.01$ , i.e., an American put option in the exercise region, a unique tree cannot be found to replicate this option. In this example, we show two binomial trees for $u \approx 1.036$ (top) as well as $u \approx 1.112$ (bottom). In each tree, we show the value of the underlying (black), the European put price (blue) and the American put price (red) at each node. Both trees replicate the American option price of 20.00 but result in different European put prices: 18.81 and 19.69. . . . .	44
3.4	De-Americanization effects on pricing put options in the CEV model. As an example, the results are shown for $p_5$ for the average error between the de-Americanized and the European prices for each strike (top right) and each maturity (bottom right). The average differences of the corresponding American and European prices is shown for each strike (top left) and each maturity (bottom left). . . . .	45

*List of Figures*

4.1	Polynomial interpolation of the Runge function $f(x) = \frac{1}{1+25x^2}$ with equidistantly-spaced nodal points. We use 5 (top) and 10 (bottom) nodal points. These are marked in black. . . . .	68
4.2	The first 6 Chebyshev polynomials $T_0(x), \dots, T_5(x)$ on the interval $[-1, 1]$ .	70
4.3	Nodal points in blue of the one-dimensional (left), two-dimensional (middle) and three-dimensional (right) Chebyshev polynomial interpolation by setting $N = 10$ in each dimensions. For the one-dimensional case, we also show in red $N + 1$ equidistantly spaced points on the upper half of the unit circle. . . . .	78
4.4	Schematic illustration of the iterative proof for Theorem 4.2.6. A function in 3 variables is iteratively interpolated in one dimension at each step. At the first step of the proof, the one-dimensional interpolation results can easily be applied using given properties of the function $f$ . From the second step onwards however, even if at each step only a one-dimensional interpolation is applied, the according properties of a "new temporarily" function have to be verified. . . . .	84
4.5	Comparison of the error bounds $b(\varrho, N, D)$ (blue) and $a(\varrho, N, D)$ (red, dashed) by setting $\varrho_1 = \varrho_2$ and $N_1 = N_2 = 10$ . At $\varrho_1 = \varrho_2 \approx 2.800882$ both error bounds intersect. . . . .	95
4.6	Efficiency study for a multidimensional lookback option in the Heston model with 5 underlyings varying one model parameter $\sigma$ . Comparison of run-times of Monte-Carlo pricing with Chebyshev pricing including the offline phase. Both methods have been set up to deliver comparable accuracies. We observe that both curves intersect at roughly $M = 35$ . . . . .	112
4.7	Efficiency study for a multivariate lookback option in the Heston model based on 5 underlyings, varying the two model parameters $\sigma$ and $\rho$ . Comparison of run-times for Monte-Carlo pricing and Chebyshev pricing including the offline phase. Both methods have been set up to deliver comparable accuracies. We observe that the Monte-Carlo and the Chebyshev curves intersect at roughly $M = 15$ . . . . .	115
5.1	Convergence study of Algorithm 6 for the integration over the density function (top). For increasing $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom.	163
5.2	Convergence study of Algorithm 6 for applying the truncated moment method (top). For increasing $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom.	164
5.3	Convergence study of Algorithm 6 for applying Fourier techniques (top). For increasing $N$ , the relative error of pricing the reference option type is reported. The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-time is reported on the bottom. . . . .	165

*List of Figures*

5.4	Application of Algorithm 5 by applying integration over the density function in the Black&Scholes model. For increasing $N$ the relative error and the run-time is reported. The cosine method of Fang and Oosterlee (2009) is used as benchmark method. . . . .	167
5.5	Absolute error (top) of applying Algorithm 9 over the test grid and 27 parameter settings for increasing $N$ . The cosine method of Fang and Oosterlee (2009) is used as the benchmark method. The run-times for the online phase are presented on the bottom. . . . .	169