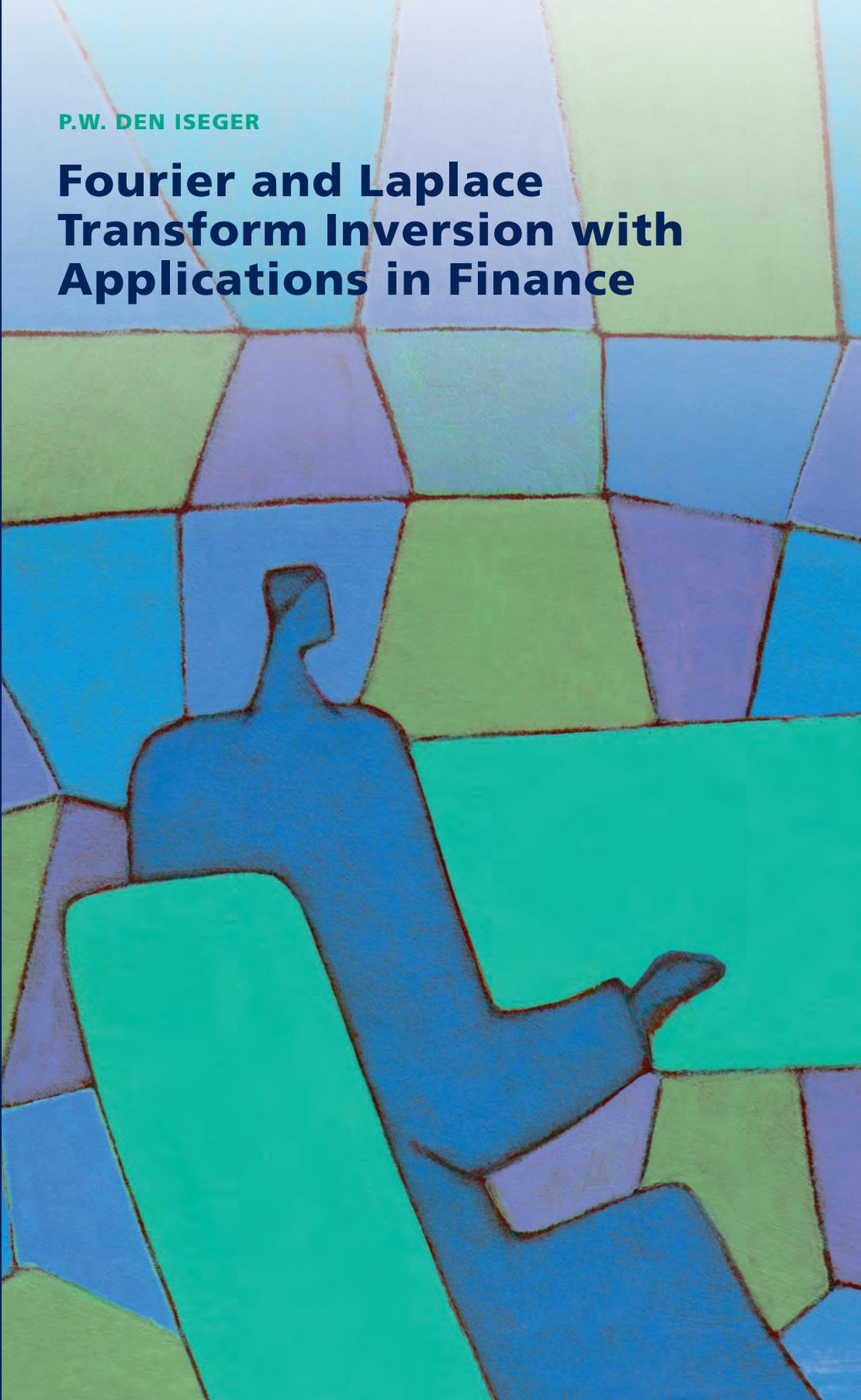


P.W. DEN ISEGER

Fourier and Laplace Transform Inversion with Applications in Finance



**Fourier and Laplace Transform
Inversion with Application in Finance**

Fourier and Laplace Transform Inversion with Application in Finance

Het inverteren van Fourier en Laplace getransformeerden met toepassingen in de
financiering

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de rector magnificus

Prof.dr. H.A.P. Pols

en volgens het besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

donderdag 9 oktober 2014 om 13.30 uur

door

PETER DEN ISEGER
geboren te Delft



Promotiecommissie

Promotor: Prof.dr.ir. R. Dekker

Overige leden: Prof.dr. D.J.C. van Dijk
Prof.dr. M.R.H. Mandjes
Prof.dr. R. Paap

Erasmus Research Institute of Management - ERIM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam
Internet: <http://www.irim.eur.nl>

ERIM Electronic Series Portal: <http://hdl.handle.net/1765/1>

ERIM PhD Series in Research in Management, 322
Reference number ERIM: EPS-2014-322-LIS
ISBN 978-90-5892-375-2
© 2014, Peter den Iseger

Design: B&T Ontwerp en advies www.b-en-t.nl
Print: Havek www.haveka.nl

This book was typeset using the \LaTeX document creation environment. The typeface for the running text is Palatino; Helvetica is used for headings.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the author.

Preface

First of all, I would like to express my gratitude to Rommert Dekker, for being my promotor. I also would like to express my gratitude to Henk Tijms for his confidence in my research. This was always an important motivation to me.

I would like to thank Eموke Oldenkamp for our collaboration during the years. We start many years ago with our research in stochastic operation research and in inventory management. Together we make the switch to quantitative finance, and had a fruitful cooperation at Cardano Risk Management. You co-authored many papers that also results in a number of chapters in this thesis.

I would like to thank Michel Mandjes and Paul Gruntjes for our fruitful cooperation when we applied the techniques in this thesis to the numerical computation of Wiener-Hopf factorizations. I further would like to thank Sander de Groot and Hans van der Weide for our nice collaboration that results in chapter 7 of this thesis. I also would like to express my gratitude to the other members of the committee.

I would like to thank Szabols Gaál for our nice collaboration that results in chapter ten of this thesis. Beside this work I also enjoyed working together at Cardano Risk Management.

Further, I would like express my gratitude to my brothers Willem and Taco den Iseger and to my mother Janny den Iseger for their support during the years.

Peter den Iseger

Contents

List of Figures	xi
List of Tables	xii
1 Introduction	1
2 Fourier transforms inversion and the Poisson summation formula	9
2.1 Fourier transforms and their properties	9
2.2 The Poisson Summation Formula	14
2.3 The Euler Algorithm of Abate en Whitt	15
2.4 Inverting z -Transforms	18
3 Numerical Transform Inversion Using Gaussian Quadrature	21
3.1 Introduction	21
3.2 Preliminaries	23
3.3 Outline of the method	24
3.4 A simple Laplace transform inversion algorithm	27
3.5 Modifications for non-smooth functions	31
3.6 The Fourier Transform	42
3.7 Proof of theorem 3.3.3	44
3.8 The computation of the $\{\mu_k^v\}$ and $\{\alpha_k^v\}$	45
3.9 Error analysis	47
3.10 On the inversion of z -transforms	54
3.11 Quadrature nodes and weights	56

4	Laplace Transform Inversion on the entire line	59
4.1	Introduction	59
4.2	Preliminaries: tools for the algorithm	60
4.3	The backbone of the method	64
4.4	Inner product preserving properties of the method	68
4.5	Implementation issues of the method	71
4.6	Error analysis	78
4.7	Modifications for non-smooth functions	79
4.8	Numerical test results	80
4.9	Computational issues	82
4.10	Proof of theorem 4.4.3	86
4.11	Detailed proof of the error theorem 4.6.1	88
5	Pricing guaranteed return rate products and discretely sampled Asian options	93
5.1	Introduction	93
5.2	Discretely sampled Asian options and guaranteed rate products: Pricing concept	95
5.3	The recursion algorithm for determining G	96
5.4	Greeks	100
5.5	A detailed description of Algorithm 5.3.1	112
5.6	Computing the put option price	120
6	Computing Greeks	129
6.1	Introduction	129
6.2	European style plain – vanilla options	130
6.3	Path dependent options	134
6.4	Computational results	139
7	The Fast Convolution Transform	147
7.1	Introduction	147
7.2	The ladder framework	149
7.3	The fast convolution algorithm	153
7.4	Implementational issues of the ladder framework	158
7.5	Applications	164
7.6	Conclusion	170
7.7	Technical note: The probability density $f_{X_{n+1}}(x)$	171
7.8	Windowed Laplace transform	171

8	Cliquet options	177
8.1	Introduction	177
8.2	Pricing in the constant and stochastic volatility models	178
8.3	Greeks in lognormal and jump diffusion models	184
8.4	Computational details	194
8.5	Preliminary: Plancharel’s identity and derived approximations	194
9	High dimensional transformation algorithms	199
9.1	Introduction	199
9.2	High dimensional transforms in practice: A (geometric) basket option example	200
9.3	The basis: a brief review of the one-dimensional Laplace and inverse Laplace transformation algorithms	202
9.4	Multi-dimensional transformation algorithms	205
9.5	Matrix representations with Kronecker products	209
9.6	Computational issues	212
10	A fast Laplace inversion method	221
10.1	Introduction	221
10.2	Barrier options in the Black-Scholes setting	223
10.3	The Helmholtz equation	226
10.4	Scaling	230
10.5	Implementation issues	236
10.6	The numerical algorithm	239
10.7	Numerical results	241
10.8	Conclusion	242
10.9	Computational issues	243
11	High dimensional Fourier transform inversion with lattice quadrature rules	245
11.1	Introduction	245
11.2	Extending the quadrature rule to higher dimensions	245
11.3	Lattice rules	247
11.4	Reproducing kernels	248
11.5	The inverse Fourier transform	250
11.6	Numerical experiments for the inverse Fourier transform	251
	Bibliography	255

Index	263
Nederlandse Samenvatting	265
About the author	269

List of Figures

4.1	Diagram of the flow of the algorithms; the LT: $(a) \rightarrow (b) \rightarrow (c)$, and the inverse transform: $(1) \rightarrow (2) \rightarrow (3)$	65
4.2	Diagram of the flow of the algorithms; the LT: $(a) \rightarrow (b) \rightarrow (c)$, and the inverse transform: $(1) \rightarrow (2) \rightarrow (3)$	76
5.1	The evolution of the price of a 30-year guarantee as the guaranteed level changes.	98
5.2	The CPU time: The parameters for the figure are $r = 3, \sigma = 20, q = 0, p_0 = \dots = p_{29} = 100$.	111
6.1	CPU times ($r = 3\%, \sigma = 20\%, q = 0, p_0 = \dots = p_{29} = 100$)	141
7.1	Three overlapping windows	175
7.2	Gaussian distribution window on $[0, 1]$ ($\Delta = 1, \delta = 1, \sigma = 0.1$).	176
9.1	Diagram of the flow of the algorithms; the LT: $(a) \rightarrow (b) \rightarrow (c)$, and the inverse transform: $(1) \rightarrow (2) \rightarrow (3)$	203
9.2	The error of approximation with the 3-dimensional inversion algorithm (the eight test functions taken three by three)	216
10.1	The average error on the boundary versus the wavenumber	241
10.2	The contour lines of the quadrupole no touch option for $t = 0.5$	242

List of Tables

3.1	Results for analytical test functions	31
3.2	Results for non-continuous test functions	34
3.3	Results for the waiting time distribution in the M/D/1 queue	35
3.4	Results for continuous non-analytical test functions	38
3.5	Results for continuous non-analytical test functions	42
3.6	Results for non-continuous test functions	43
3.7	Quadrature nodes and weights. Only the numbers λ_j and β_j for $j=1,2,\dots,\frac{n}{2}$ are tabled. The numbers λ_{n+1-j} are given by $-\lambda_j - 2\pi$. The numbers β_{n+1-j} coincide with β_j .	57
4.1	Results for analytical test functions	81
4.2	Numerical values for f with Laplace transform given by (4.8.1)	82
5.1	Price and computation time for discretely sampled Asian options.	99
5.2	Prices of APO's with extrapolations of errors of order one, two and three.	100
5.3	Parameters. For all seven cases $q = 0$ and $K = 2.0$	112
5.4	Price, delta and gamma of APO's	112
5.5	Vega, rho and convexity of APO's	113
6.1	Parameters (for all seven cases $q = 0$ and $K = 2.0$)	140
6.2	Price, delta and gamma of APO's	140
6.3	vega, rho and convexity of APO's	141
7.1	European put option in Black-Scholes model ($S(0) = 50, K = 50, r = 0.05, q = 0.03, T = 1, \sigma = 0.2$). Analytic price: 3.3654588245816521.	165
7.2	Calibrated VG parameters for S&P 500 June 30, 1999	167
7.3	European put option in the variance gamma model ($S(0) = 50, K = 50$)	167
9.1	The test functions and their Laplace transforms	214
9.2	The error made by the 2-dimensional inversion algorithm	215

Chapter 1

Introduction

It is becoming standard for modeling and analysis to include algorithms for computing probability distributions of interest. Therefore, advances in computational probability increase the value of stochastic models in among others quantitative finance, queueing, reliability, and inventory management. Several tools have been developed for this purpose, of which one of the most powerful tools is numerical Fourier transform inversion.

The reason is that probability distributions can be characterized in terms of Laplace transforms, or more generally, Fourier transforms. Many results in among others quantitative finance, queueing and reliability theory, are given in the form of transforms and become amenable for practical computations once fast and accurate methods for numerical Laplace and Fourier transform inversion are available.

This thesis presents advanced numerical inversion techniques of Laplace and Fourier transforms and their innovative application in quantitative finance. Although our applications are mainly focused on quantitative finance, the presented inversion techniques also tackle important problems from logistics, queueing, probability theory.

Fourier transforms inherit many of the very convenient properties of the exponential function. While for exponential functions one has $\exp(s(x + y)) = \exp(sx) \exp(sy)$, this translates in the case of Fourier transforms into the property that *"the transform of the density of the sum of two independent random variables is the product of the transforms of these densities"*. For the derivative of exponential functions holds $\frac{d}{dx} \exp(ax) = a \exp(ax)$; this translates in the property that the transform $\hat{f}'(a)$ of the derivative of a function f is given by $s\hat{f}(s)$ (with $\hat{f}(s)$ being the transform of f). Due to these properties it is possible to determine many quantities of interest in terms of their transform.

Important examples are the prices of financial derivatives such as option prices. The challenge is to invert the Laplace and Fourier transforms in order to compute the actual option prices.

The use of Fourier inversion methods in quantitative finance started with the Carr-Madan method in Carr and Madan (1999). This method applied the well known FFT (Fast Fourier Transform) algorithm to speed up computations, using the trapezoid rule in order to discretize the Fourier inversion integral. The discretization of the Fourier inversion integral is known in the literature as the Fourier series method. The mathematical justification of these methods is based on the Poisson Summation formula. The Poisson Summation Formula relates an infinite sum of Laplace transform values to the z-transform (Fourier series) of the function values $f(k\Delta)$, $k = 0, 1, \dots$. Unfortunately, the infinite sum of Laplace transform values converges in general very slowly. In their seminal paper, Abate and Whitt (1992b) used an acceleration technique called Euler Summation to accelerate the convergence rate of the infinite sum of Laplace transform values. The most successful Fourier series methods known in the literature are Dubner (1968), Abate and Whitt (1992b), Abate and Whitt (1995), Choudhury, Lucantoni and Whitt (1996), O'Conneide (1997) and Sakurai (2004). Other variants of the Fourier series method used in the field of quantitative finance are known as the CONV method Lord e al.(2007) and the COS method Fang and Oosterlee (2008). An excellent overview on Fourier and Laplace transforms and related topics such as the Poisson Summation formula is given by the book of Mallat (2001).

Despite all these papers there are still many open issues with respect to methods published and our main research question is whether we can improve these. Chapter 3 presents a new Gaussian quadrature framework, a method which facilitates replacing that slow-converging infinite series with a finite sum, resulting in an efficient numerical inversion algorithm. This algorithm computes all needed function values at once, in $O(N \log(N))$ time (the function values in the points $k\Delta$, $k = -N, -(N-1), \dots, N$ and Δ a freely chosen parameters). Besides the algorithm being very fast, it is also very accurate. Most transforms used in applications can be transformed with almost machine precision. The results of this chapter are published in the paper den Iseger (2006a).

A second type of numerical inversion method is called Gaver-Stehfest algorithm, which is based on combinations of Gaver functionals (cf. Gaver (1966) and Stehfest (1970)). A third line of research led to the Weeks method, which is based on bilinear transformations and Laguerre expansions (cf. Weeks (1966) and Abate, Choudhury and Whitt (1996) and (1998)). The fourth type of inversion technique is known under the

name of the Talbot method which is based on deforming the contour in the Bromwich inversion integral (cf. Talbot (1979) and Murli and Rizzardi (1990)).

In many applications these methods give excellent results and are popular tools in quantitative finance, queueing, reliability, and inventory management. However, a common feature of these methods is their sensitivity for singularities and discontinuities. One simple discontinuity can ruin the convergence properties of these algorithms. Such unpredictable behavior of an inversion method limits the practical value of numerical transform inversion algorithms. The inversion methods presented in this thesis repair these shortcomings.

Another common feature of most known inversion methods is that the inversion is performed at a single point at a time. However, in practical situations one often needs to compute many function values, or even function values on an entire interval. Examples include calibration of quantitative models to market prices and the pricing of financial derivatives in asset liability management scenario generators. Therefore, when computation of these models is based on such a "single-point" inversion method, computing many function values can get rather time consuming. In chapter three we present a fast and efficient numerical transform inversion, based on a Gaussian quadrature formula, which can perform the inversion on a uniform grid at a time (that is, performs the inversion in several points at a time). This algorithm can be employed for solving many of the problems arising in computational probability with near machine precision. However flexible, fast and precise, this method also has some limitations (i.e. the limitations of a uniform grid vs an arbitrary grid). For this reason, it cannot tackle important problems as iterative algorithms (for instance recursion). Further important applications include partial differential equations and Wiener-Hopf factorizations. Recursion algorithms are widely employed for dynamic programming style problems, a technique employed for instance in pricing callable derivatives like Bermuda options. Partial differential equations find applications among others in the pricing of barrier options and many other financial derivatives. In fact, by the well-known Feynman-Kac theorem we can model many pricing problems in terms of the solution of partial differential equations. Wiener-Hopf factorizations are an important tool in Lévy processes based modeling of credit derivatives and barrier options. As a matter of fact, there are also numerous applications in queueing, among those the famous problem of the computation of the waiting time distribution in a $G/G/1$ queue.

In all these applications it is of great importance that one is able to compute the function not only in one point, not even on a uniform grid, but on an entire interval.

In Chapter 4 a new inversion technique is presented, which overcomes these problems and can be deployed for the earlier mentioned applications. A natural way to do this is to compute a piece wise (Legendre) polynomial expansion for the function in question. The algorithm computes the coefficients of the Legendre expansions of the desired function on all intervals $[k\Delta, (k + 1)\Delta]$ (an arbitrary grid!). This piecewise inversion algorithm maintains the capabilities of the pointwise inversion algorithm. The two key improvements, which enable the new numerical technique to tackle a much wider spectrum of problems than its forebear, can be summarized as follows. First, the original function f (obtained after the numerical inversion) can be evaluated in arbitrary points x_k (instead of only in points $k\Delta, k = 0, 1, \dots$, as previously). Secondly, the new method works both ways: if some numerical values of the original function f are given, then the algorithm generates its Laplace transform $\mathcal{L}f$ (thus no closed-form analytical expression for f is needed!); and the other way around, given some numerical values of the Laplace transform, the original function f can be obtained in arbitrary points. Given this polynomial expansion, one can evaluate derivatives, maxima, integrals and other operations. In combination with an algorithm that can compute the transform of such a piece-wise polynomial yields a powerful framework. The results of this chapter are described in the working paper den Iseger (2006b) *.

In the fifth chapter we apply the techniques of the previous chapter in pricing Asian options and guaranteed rate linked products. An Asian option is an option on the average value of several underlying stock prices. A guaranteed rate linked option is a product that gives a guaranteed rate of return. We derive a recursion scheme for the pricing and the market sensitivities, called the Greeks. In each step of the recursion one needs to compute a transform and the related inverse transform. These computations can be done efficiently with the techniques of the previous chapters. Monte Carlo simulation and Fast Fourier transform methods are much slower when pricing these products, moreover, high accuracy cannot be achieved for the computation of the Greeks. This chapter is based on the paper den Iseger and Oldenkamp (2006a).

The sixth chapter presents a new technique for computing the market sensitivities, called the Greeks. The new method reduces the computation of the Greeks back to a similar problem as of computing the price of the derivative in question; that is, if there is an efficient algorithm pricing the derivative then in order to compute the Greeks we will use the the same efficient algorithm only with a modified payoff function. The results of this chapter are described in the working paper den Iseger and Oldenkamp (2006b).

* In some publication this paper is referred as P. Den Iseger, Numerical evaluation of the Laplace and inverse Laplace transform

In chapter seven we present a fast convolution algorithm and discuss the so called ladder algorithms. This ladder algorithm can compute the solution of dynamic programming problems. An application in quantitative finance is the pricing of Bermuda options. In each step of the ladder algorithm we need to compute a convolution. We develop a new class of fast convolution algorithms based on the inversion algorithms of the previous chapters. The only known fast convolution algorithms known in the literature are for the Gaussian distribution, the so called fast Gauss transform. The new algorithm can handle general probability distributions, including the so called Levy processes. This chapter is based on the working paper de Groot, den Iseger and van der Weide (2006).

In chapter eight the techniques of the previous chapters are applied to meet the challenge of pricing cliquet options, as well as computing its sensitivity to changes in the market, the Greeks. These computations are done in case of (piecewise) constant volatility, also incorporating skews, in case of stochastic (Hestonian) volatility and, lognormal and jumpdiffusion asset price processes. The results of this chapter are described in the working paper den Iseger and Oldenkamp (2006c).

A natural step in extending these inversion algorithms, is the design of an inversion algorithm for two, three or even higher dimensional transforms. In quantitative finance, multidimensional inversion algorithms enable, for instance, pricing of financial derivatives depending on more than one underlying factor, such as stochastic volatility models. The scholastic literature on higher dimensional Laplace (and Fourier) transform algorithms is scarce. As a result, such derivatives are commonly priced by means of Monte Carlo simulation, which is slow and not very accurate. The computation of financial market sensitivities, the so called Greeks, proves to be an especially challenging problem in the literature. The ability of computation of high dimensional piecewise polynomial expansions and an algorithm that can compute the multidimensional transform of such piece-wise polynomial gives us a powerful tool for solving these problems efficiently and accurately.

With the higher dimensional inversion algorithm we can also extend the fast convolution and ladder algorithm to higher dimensions. The results of this chapter are described in the working paper den Iseger and Oldenkamp (2008) .

In the tenth chapter we apply the high dimensional transform inversion algorithms for solving partial differential equations and the heat equation. With the celebrated

Feynman-Kac formula, cf. Shreve (2008), many pricing problems in mathematical finance can be formulated as partial differential equations. We apply these techniques on the pricing of barrier options. These are options where the payoff depends on whether the underlying stock hits a certain barrier. Besides the new algorithm is much faster than the known algorithms in the literature, the algorithm is numerically extremely stable. This facilitates the computation of highly oscillating solutions. This chapter is based on the working paper Gaál and den Iseger (2009)).

In chapter eleven the quadrature framework of chapters 1 and 2 are combined with so the called Lattice integration methods. These are quasi-Monte Carlo methods which are very effective for integrating high dimensional periodic functions. The resulting inversion algorithm can invert ten dimensional transforms with high precision in a few seconds. The algorithm is applied for the pricing of Basket options. Basket options are especially important for pension funds, for they can be used as an insurance instrument, having as underlying (a part of) their investment portfolio. The only alternative computation method for the price of a basket option, known in the literature, is Monte Carlo simulation, which is very time consuming.

Let us conclude this introduction with an answer to the following question. This thesis presents two excellent inversion methods and a couple of highly relevant applications; but could one maybe keep to one of those methods instead of trying to understand them both? Formulating more bluntly, isn't one of these methods redundant?

It is absolutely true that the two methods are very similar, one could even argue the only difference is that while the quadrature rule computes the function in the points $k\Delta$ ($k = 0, 1, \dots$), the entire line algorithm evaluates the function on an entire interval, that is, in any arbitrary point. So indeed, why not just use the latter one and forget about the quadrature method? The quadrature method yields in general somewhat more accurate answers and is a bit faster. It is also easier to implement, that is, it is easier to adapt for handling singularities. Especially in case of singularities around 0, such as for the function \sqrt{x} . Furthermore, as mentioned in the previous paragraph, the quadrature method can be relatively easily extended to much higher dimensions (about 10 dimensions for neat functions) whereas the entire line algorithm can be extended to three or at most four dimensions. Therefore, it depends on the problem at hand which algorithm is advisable to use. For instance, should you need to invert a transform in a single point, then go for the quadrature algorithm; however if you have a recursive problem at hand, you have no other choice than to use the entire line algorithm.

At last but not least we would like to mention that although there are ample relevant applications of the algorithms presented in the present thesis, there are also many applications in different fields, which do not fit in the scope of this thesis, but are published else where:

1. Inventory Control and Regenerative Processes: Computations, Bázsza, Frenk and den Iseger (1999a)
2. Inventory Control and Regenerative Processes: theory, Bázsza, Frenk and den Iseger (1999b)
3. Inventory Modeling of inventory control with regenerative processes, Bázsza, Frenk and den Iseger (2001)
4. Wide Sense One-Dependent Processes with Embedded Harris Chains and their Applications in Inventory Management, Bázsza, and den Iseger (2002a)
5. A Two Level Decentralized Distribution System with Compound Renewal Demand, Bázsza, and den Iseger (2002b)
6. Single Item Inventory Models - A Time- and Event- Averages Approach, Bázsza, and den Iseger (2002c)
7. Optimal Continuous Order Quantity (s, S) Policies - The 45-Degrees Algorithm, Bázsza, and den Iseger (2002d)
8. A Wiener-Hopf based approach to numerical computations in fluctuation theory for Lévy processes, Gruntjes, den Iseger and Mandjes (2013)
9. Numerical techniques in Lévy fluctuation theory, Asghari, den Iseger and Mandjes (2014)

Chapter 2

Fourier transforms inversion and the Poisson summation formula

2.1 Fourier transforms and their properties

Fourier transformations play an important role in many fields of applied mathematics, such as applied probability, statistics, mathematical finance, partial differential equations and others. The Fourier transform of a function f of the space $L^1(-\infty, \infty)$ of Lebesgue integrable functions is defined as

$$\widehat{f}(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt.$$

This transform is also used in the scientific literature of different fields under the names characteristic function, moment generating function and Laplace transform. If f_X is the density of a random variable \mathbf{X} then we can write

$$\widehat{f}_X(s) = \mathbb{E} \exp(-s\mathbf{X}) = \int_{-\infty}^{\infty} e^{-st} f(t) dt.$$

In case \mathbf{X} has no density then

$$\widehat{f}_X(s) = \mathbb{E} \exp(-s\mathbf{X}) = \int_{-\infty}^{\infty} \exp(-sx) dF_X(x).$$

The latter transform is also called the Laplace-Stieltjes transform.

In fact

$$|\widehat{f}(s)| \leq \int_{-\infty}^{\infty} |f(t)| dt = \|f\|_1, \tag{2.1.1}$$

and \widehat{f} is a continuous function. If the support of f is contained in $[0, \infty)$ then we call \widehat{f} the Laplace transform of f . If $\widehat{f} \in L^1(-i\infty, i\infty)$, the space of complex valued Lebesgue integrable functions, satisfying $\frac{1}{2\pi i} \int_{-i\infty}^{i\infty} |x(s)| ds < \infty$, then the inverse Fourier integral is properly defined.

Theorem 2.1.1. If $f \in L^1(-\infty, \infty)$ and $\hat{f} \in L^1(-i\infty, i\infty)$ then

$$f(t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{st} \hat{f}(s) ds,$$

is properly defined and

$$f(t) \leq \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} |\hat{f}(s)| ds = \|\hat{f}\|_1.$$

This theorem can be found in any text about Fourier transforms, we have taken it from Mallat (2001). It follows immediately that $\hat{f} \in L^1(-i\infty, i\infty)$ if and only if f is a continuous function.

2.1.1 Convolution property

It is well known that if \mathbf{X} and \mathbf{Y} are independent random variables with densities f_X and f_Y , then the density f_Z of the random variable $\mathbf{Z} = \mathbf{X} + \mathbf{Y}$, is the convolution of f_X and f_Y , given by

$$f_X * f_Y(z) = \int f_X(z - y) f_Y(y) dy$$

The Fourier transform of this density can be written as

$$\begin{aligned} \hat{f}_Z(s) &= \mathbb{E} \exp(-s\mathbf{Z}) \\ &= \mathbb{E} \exp(-s\mathbf{X}) \mathbb{E} \exp(-s\mathbf{Y}) \\ &= \hat{f}_X(s) \hat{f}_Y(s). \end{aligned}$$

Thus the Fourier transform of the convolution of f_X and f_Y is simply the product of the Fourier transforms $\hat{f}_X(s)$ and $\hat{f}_Y(s)$. In fact this property holds for any function f_X and f_Y .

Example 2.1.2. Suppose that \mathbf{Z} is a compound Poisson process, that is

$$\mathbf{Z} = \sum_{k=1}^{\mathbf{N}(t)} \mathbf{X}_k,$$

with $\mathbf{N}(t)$ a Poisson process with rate λ and with $\{\mathbf{X}_k\}$ random variables with distribution function F_X . Then we obtain for the Fourier transform of the density f_Z

$$\begin{aligned} \hat{f}_Z(s) &= \sum_{k=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} \hat{f}_X^k(s) \\ &= \exp\left(-\lambda t \left(1 - \hat{f}_X(s)\right)\right). \end{aligned}$$

2.1 Fourier transforms and their properties

Example 2.1.3. Suppose that the Fourier transform of log stock price X_T , at time T , is given by \widehat{f}_{X_T} (this is the Fourier transform under the so called forward measure). Suppose that we want to price a plain vanilla put option on this stock price. This price is given by

$$p_t = D(t, T) \mathbb{E} (K - \exp(X_T))^+$$

with $D(t, T)$ the discount factor over the time period $[t, T]$ and $(x)^+ = \max(x, 0)$. Then we obtain for the price as function of the logarithm of the strike $k = \log(K)$

$$\begin{aligned} p_t(k) &= D(t, T) K \mathbb{E} \left(1 - e^{-k} \exp(-(k - X_T)) \right)^+ \\ &= D(t, T) K \int_{-\infty}^k \left(1 - e^{-k} \exp(-(k - x)) \right) dF_X(x) \end{aligned}$$

The price is the convolution of the function

$$V(x) = (1 - e^{-x}) 1_{\{x \geq 0\}}$$

and dF_X . Since the Fourier transform of V is given by

$$\widehat{V}(s) = \frac{1}{s} \frac{1}{1+s},$$

we obtain for the Fourier transform of p_t

$$\widehat{p}_t(s) = D(t, T) K \frac{1}{s} \frac{\widehat{f}_X(s)}{1+s}.$$

2.1.2 Derivative property

Another important property of the Fourier transform is the derivative property. The Fourier transform of the derivative of a function f is given by

$$s\widehat{f}(s) - \sum e^{-sx_k} (f(x_k^+) - f(x_k^-)),$$

where the points x_k are the points where f is discontinuous. This result can easily be obtained by integration by parts. Suppose that the derivative Df of f is continuous on (a, b) , then

$$\begin{aligned} \int_a^b e^{-st} Df(t) dt &= \int_a^b e^{-st} df(t) \\ &= \left(e^{-sb} f(b^-) - e^{-sa} f(a^+) \right) + s \int_a^b e^{-st} f(t) dt. \end{aligned}$$

Applying this to

$$\int_{-\infty}^{\infty} e^{-st} Df(t) dt = \sum_k \int_{x_k}^{x_{k+1}} e^{-st} Df(t) dt,$$

yields exactly the desired result, that is,

$$s\hat{f}(s) - \sum e^{-sx_k} (f(x_k^+) - f(x_k^-)).$$

Example 2.1.4. Suppose that f is the solution of the differential equation

$$Df = \alpha f, f(0^+) = f_0$$

and $f(t) = 0$ for $t \leq 0$. Then we obtain for the Fourier transform \hat{f} the equation

$$s\hat{f}(s) - f_0 = \alpha\hat{f}(s),$$

therefore,

$$\hat{f}(s) = \frac{f_0}{s - \alpha}.$$

Example 2.1.5. Suppose that we want to price a European derivative in the Black-Scholes model. In this model the dynamics of the stock price is modelled with the stochastic differential equation

$$dS(t) = S(t) r dt + \sigma S(t) dW(t),$$

with r the constant riskfree rate and σ the constant volatility. At time T the derivative in question renders a payoff of $\tilde{v}(S_T)$.

It is well known that the price of this derivative is the solution of the Black-Scholes partial differential equation

$$\begin{aligned} \tilde{V}_t + \frac{1}{2}\sigma^2 S^2 \tilde{V}_{SS} + rS\tilde{V}_S - r\tilde{V} &= 0 \\ \tilde{V}(T, S) &= \tilde{v}(S). \end{aligned}$$

Introduce

$$V(t, x) = \tilde{V}(T - t, e^x).$$

Since

$$\begin{aligned} V_t &= -\tilde{V} \\ V_x &= S\tilde{V}_S \\ V_{xx} &= S^2\tilde{V}_{SS} + S\tilde{V}_S \end{aligned}$$

we obtain the equation

$$\begin{aligned} V_t &= \frac{1}{2}\sigma^2 V_{xx} + \left(r - \frac{1}{2}\sigma^2\right) V_x - rV \\ V(T, x) &= v(x) \end{aligned}$$

2.1 Fourier transforms and their properties

with $v(x) = \tilde{v}(\exp(x))$. Taking the Fourier transform with respect to t in the previous equation of $V(t, x)$ yields the differential equation,

$$s_t \widehat{V}(s_t, x) - v(e^x) = \frac{1}{2} \sigma^2 \widehat{V}_{xx}(s_t, x) + \left(r - \frac{1}{2} \sigma^2 \right) \widehat{V}_x(s_t, x) - r \widehat{V}(s_t, x)$$

Taking in turn the Fourier transform of $\widehat{V}(s_t, x)$ with respect to x in the previous equation, yields further

$$s_t \widehat{V}(s_t, s_x) - \widehat{v}(s_x) = \frac{1}{2} \sigma^2 s_x^2 \widehat{V}_{xx}(s_t, s_x) + \left(r - \frac{1}{2} \sigma^2 \right) s_x \widehat{V}_x(s_t, s_x) - r \widehat{V}(s_t, s_x).$$

From this previous equation one can express $\widehat{V}_x(s_t, s_x)$, the two dimensional Fourier transform of the derivative function $V_x(t, x)$ as

$$\widehat{V}_x(s_t, s_x) = \frac{\widehat{v}(s_x)}{s_t + r - \frac{1}{2} \sigma^2 s_x^2 - \left(r - \frac{1}{2} \sigma^2 \right) s_x}.$$

In the above examples we see that many quantities of interest in applied probability, statistics, mathematical finance, partial differential equation and other fields can be modelled with Fourier transforms. The challenge is to invert these Fourier transforms. To be more precise, we aim to compute the function f from the Fourier transform

$$\widehat{f}(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt.$$

Theoretically, there is the so called Bromwich inversion integral

$$f(t) = \frac{1}{2\pi} \int_{-i\infty}^{i\infty} e^{st} \widehat{f}(s) ds.$$

This is a classical result that can be found for instance in Abate and Whitt (1992a). Because in general the integrand is highly oscillating and converges slowly to zero, the practical value of this integral representation is limited. Most modern Laplace and Fourier inversion algorithms are based on the Poisson Summation formula.

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) e^{i2\pi(k+v)t} = \sum_{j=-\infty}^{\infty} f(j+t) e^{-i2j\pi v}.$$

This classical result can also be found in Abate and Whitt (1992a). In the next section we discuss this formula, and show how it can be used for Fourier and Laplace transform algorithms.

2.2 The Poisson Summation Formula

The basis for most Laplace and Fourier inversion algorithms is the Poisson summation formula (PSF). This formula is given by

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) e^{i2\pi(k+v)t} = \sum_{j=-\infty}^{\infty} f(j+t) e^{-i2j\pi v} \quad (2.2.1)$$

in which \widehat{f} is the Laplace (Fourier) transform

$$\widehat{f}(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt$$

and i the imaginary unit, $i^2 = -1$. This formula can relatively easily be derived by introducing a function g as

$$g(t) = \sum_{j=-\infty}^{\infty} f(j+t). \quad (2.2.2)$$

The function g is a periodic function. Under some technical conditions, we can represent g with the Fourier series

$$g = \sum_{k=-\infty}^{\infty} \langle g, w_k \rangle w_k,$$

with $w_k(t) = \exp(i2k\pi t)$ and $\langle \cdot, \cdot \rangle$ the standard inner product on $L^2[0, 1]$. Substitution of equation (2.2.2) in $\langle g, w_k \rangle$ yields

$$\begin{aligned} \langle g, w_k \rangle &= \sum_{j=-\infty}^{\infty} \int_0^1 f(j+t) \exp(-i2k\pi t) dt \\ &= \sum_{j=-\infty}^{\infty} \int_0^1 f(j+t) \exp(-i2k\pi(j+t)) dt \\ &= \int_{-\infty}^{\infty} f(j+t) \exp(-i2k\pi(j+t)) dt \\ &= \widehat{f}(i2k\pi). \end{aligned}$$

Hence we have

$$\sum_{j=-\infty}^{\infty} f(j+t) = g(t) = \sum_{k=-\infty}^{\infty} \widehat{f}(i2k\pi) e^{i2\pi kt}. \quad (2.2.3)$$

The shift operator for a real or complex number α is defined as

$$f_\alpha(t) = e^{-\alpha t} f(t) \quad (2.2.4)$$

$$\widehat{f}_\alpha(s) = \widehat{f}(s + \alpha). \quad (2.2.5)$$

2.3 The Euler Algorithm of Abate en Whitt

This last identity follows from

$$\begin{aligned}\widehat{f}_a(s) &= \int_{-\infty}^{\infty} e^{-st} e^{-at} f(t) dt \\ &= \widehat{f}(a+s).\end{aligned}$$

This yields in combination with equation (2.2.3) the formula

$$e^{-at} \sum_{j=-\infty}^{\infty} e^{-aj} f(j+t) = g(t) = \sum_{k=-\infty}^{\infty} \widehat{f}(i2k\pi + a) e^{i2\pi kt}. \quad (2.2.6)$$

Substituting now simply $a = i2\pi v$ we obtain the desired result (2.2.1).

Remark 2.2.1. *The Laplace transform of the function*

$$f_{\Delta}(t) = f(\Delta t)$$

is given by

$$\begin{aligned}\widehat{f}_{\Delta}(s) &= \int e^{-st} f(\Delta t) dt \\ &= \frac{1}{\Delta} \int e^{-\frac{s}{\Delta}x} f(x) dx \\ &= \frac{1}{\Delta} \widehat{f}\left(\frac{s}{\Delta}\right).\end{aligned}$$

Applying the PSF (2.2.1) to the transform \widehat{f}_{Δ} yields

$$\frac{1}{\Delta} \sum_{k=-\infty}^{\infty} \widehat{f}\left(\frac{i2\pi(k+v)}{\Delta}\right) e^{i2\pi(k+v)t} = \sum_{j=-\infty}^{\infty} f(\Delta(j+t)) e^{-i2j\pi v}.$$

The Gaussian quadrature inversion algorithm that we are about to discuss in this thesis, calculates the function values of f in the points $\{k\Delta; k \in \mathbb{Z}\}$. By applying the algorithm on \widehat{f}_{Δ} we can calculate the function values of f in the points $\{\Delta k; k \in \mathbb{Z}\}$.

2.3 The Euler Algorithm of Abate en Whitt

Abate and Whitt use formula (2.2.1) with $v = 0$ and $t = 0$ for the function f_a , hence

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi k + a) = \sum_{j=-\infty}^{\infty} e^{-aj} f(j). \quad (2.3.1)$$

If f now has support $[0, \infty)$ and if the so-called damping factor a is chosen big enough, then

$$\begin{aligned}\sum_{k=-\infty}^{\infty} \widehat{f}(2\pi i k + a) &= \sum_{j=0}^{\infty} e^{-aj} f(j) \\ &\approx f(0) + e^{-a} f(1)\end{aligned} \quad (2.3.2)$$

Chapter 2. Fourier transforms inversion and the Poisson summation formula

The left summation of formula (2.3.1) converges in general very slowly, for instance, the function $1_{\{t \geq 0\}}$ has Laplace transform s^{-1} . This is caused by the fact that the function $1_{\{t \geq 0\}}$ has a singularity in $t = 0$. It is known that if f is n times continuously differentiable then $|\widehat{f}(is)| \rightarrow O(s^{-(n+1)})$ if $|s| \rightarrow \infty$ (this can simply be checked by partially integrating n times).

If f is a neat function on $(0, \infty)$ and only has singularities on $t = 0$ we can apply formula (2.3.1) to the function

$$f_w(t) = w(t)f(t)$$

with w a so called window function such that $w(1) = 1$ and $\frac{\partial^k w(t)}{(\partial t)^k} |_{t=0} = 0$, for $k = 0, 1, \dots, p-1$. This implies that $|\widehat{f_w}(is)| \rightarrow O(s^{-(p+1)})$, hence for p large enough formula (2.3.1) converges quickly. A good choice for w is

$$w(t) = \sin\left(\frac{\pi t}{2}\right)^{2p} \tag{2.3.3}$$

This yields

$$\begin{aligned} f_w(t) &= \left(\frac{e^{i\frac{\pi}{2}t} - e^{-i\frac{\pi}{2}t}}{2i}\right)^{2p} f(t) \\ &= \left(\frac{1}{2}\right)^{2p} \sum_{j=0}^{2p} (-1)^{p-j} e^{i\pi(p-j)t} f(t), \end{aligned}$$

and thus

$$\widehat{f_w}(s) = \left(\frac{1}{2}\right)^{2p} \sum_{j=0}^{2p} (-1)^{p-j} \widehat{f}(s + i\pi(j-p)).$$

Substituting this expression in formula (2.3.2) gives

$$\sum_{k=-\infty}^{\infty} \left(\frac{1}{2}\right)^{2p} \sum_{j=0}^{2p} \widehat{f}(i\pi(2k+j-p) + a) = e^{-a} f(1). \tag{2.3.4}$$

Here we use the fact that $f_w(0) = 0$ and $f_w(1) = 1$. This yields the well-known Euler algorithm of Abate and Whitt (1992a). The given derivation is an interpretation of the author. The window function (2.3.3) is used in den Iseger (2006a) to smooth singularities in $t = 0$, like $f(t) = t^a$, $-1 < a < 1$.

Remark 2.3.1. The "trick" with the damping factor only works for functions with support $[0, \infty)$. An important extension would be a method that calculates $f(a)$ by means of

$$\widehat{g}(v) = \sum_{j=-\infty}^{\infty} e^{-i2\pi jv} f(j)$$

2.3 The Euler Algorithm of Abate en Whitt

One possibility would be to calculate the harmonic extension of $\widehat{g}(v)$ (i.e. Dirchlet problem). This can be done by solving a PDE or to numerically integrate the Poisson integral. Another possibility would be developing a quadrature formula for the Fourier inversion integrals

$$\int_0^1 e^{-i2\pi v} \widehat{g}(v) dv.$$

In the next section we will discuss inverting the function \widehat{g} (i.e. calculating the function $f(j)$) using the fast Fourier transform (FFT) algorithm).

Remark 2.3.2. The Euler method could be generalized by using other window functions. If $|f(t)| < \epsilon$ for $t \notin [a, b]$ then the Laplace transform of f_w can be determined as follows:

1. Approximate f_w with f_{w_p} and $w_p(t + ja) = w(t)$, for $t \in [a, b]$ and $j \in \mathbb{Z}$.
2. Approximate w_p using the Fourier series

$$w_p^n(t) \approx \sum_{k=-n}^n A_k e^{i2\pi k \frac{t-a}{b-a}},$$

with

$$A_k = \frac{1}{b-a} \int_a^b w(t) e^{-\frac{i2\pi k}{b-a}t} dt.$$

3. Approximate $\widehat{f_w}$ with $\widehat{f_{w_p^n}} = \sum_{k=-n}^n A_k \widehat{f}(s - \frac{i2\pi k}{b-a})$.

The window functions are used in den Iseger (2006a) to accommodate a Laplace transform inversion algorithm that is, they are robust for singularities.

Remark 2.3.3. If $|w(t)| < \epsilon$ for $t \notin [a, b]$ then the Fourier coefficients A_k can be calculated with

$$\begin{aligned} A_k &= \frac{1}{b-a} \int_a^b w(t) e^{-\frac{i2\pi k}{b-a}t} dt \\ &\approx \int_{-\infty}^{\infty} w(t) e^{-\frac{i2\pi k}{b-a}t} dt \\ &= \widehat{w} \left(\frac{i2\pi k}{b-a} \right). \end{aligned}$$

One could also apply formula (2.3.1) directly to $\widehat{f_{w_p^n}}$ or $\widehat{f_w}$.

2.4 Inverting z-Transforms

Given a number of function values of the function

$$\hat{g}(v) = \sum_k g_k e^{-i2k\pi v}, \quad (2.4.1)$$

we aim to calculate the coefficients g_k . We will see that

$$g_k \approx \frac{1}{N} \sum_{j=0}^{N-1} \hat{g}\left(\frac{j}{N}\right) \exp\left(\frac{i2\pi kj}{N}\right) = \sum_{m=-\infty}^{+\infty} g_{mN+k}. \quad (2.4.2)$$

The sums in formula 2.4.1 and 2.4.2 can be calculated efficiently using the FFT algorithm (the first N sums can be calculated in $N \log(N)$ time). Because

$$\sum_{k=0}^{N-1} z^k = \frac{1 - z^N}{1 - z}$$

we obtain for $k \in \mathbb{Z}/\{0\}$

$$\frac{1}{N} \sum_{j=0}^{N-1} \left(e^{-i2\pi \frac{k}{N}}\right)^j = \frac{1 - \exp(-i2\pi k)}{1 - \exp\left(\frac{-i2\pi k}{N}\right)} = 0. \quad (2.4.3)$$

Note that for $k = 0$ the sum in 2.4.3 is equal to 1. Now introduce the inner product

$$\langle \hat{f}, \hat{g} \rangle_N = \frac{1}{N} \sum_{j=0}^{N-1} \hat{f}\left(\frac{j}{N}\right) \hat{g}^*\left(\frac{j}{N}\right), \quad (2.4.4)$$

with \hat{g}^* the complex conjugate of \hat{g} . Let us introduce the function $\bar{w}_k = \exp(-i2\pi kt)$. Notice that

$$\begin{aligned} \langle \bar{w}_k, \bar{w}_j \rangle_N &= \frac{1}{N} \sum_{p=0}^{N-1} \left(\exp(i2\pi p \left(\frac{k-j}{N}\right))\right)^p \\ &= \delta_{kj}, \end{aligned}$$

with the Kronecker delta δ defined by

$$\langle \bar{w}_k, \bar{w}_j \rangle_N = \delta_{kj} = 0, \quad k \neq j \quad (2.4.5)$$

$$= 1, \quad k = j. \quad (2.4.6)$$

This means that $\{\bar{w}_k; k = 0, 1, \dots, N-1\}$ is an orthogonal set with respect to the inner product $\langle \cdot, \cdot \rangle_N$. Hence, if

$$\hat{g} = \sum_{k=0}^{N-1} g_k \bar{w}_k,$$

2.4 Inverting z-Transforms

then

$$\langle \hat{g}, \bar{w}_k \rangle_N = g_k.$$

Generalizing this result, we obtain

$$\hat{g} = \sum_{k=-\infty}^{\infty} g_k \bar{w}_k. \quad (2.4.7)$$

By writing k as $k = pN + q$, with p and q integers, we obtain for g

$$\hat{g} = \sum_p \sum_{q=0}^{N-1} g_{pN+q} \bar{w}_{pN+q}. \quad (2.4.8)$$

Because

$$\begin{aligned} \bar{w}_{pN+q}\left(\frac{j}{N}\right) &= \exp\left(-i2\pi\left(pj + \frac{q}{N}\right)\right) \\ &= \exp\left(-i2\pi\frac{q}{N}\right) \\ &= \bar{w}_q\left(\frac{j}{N}\right), \end{aligned}$$

we get

$$\begin{aligned} \langle \hat{g}, \bar{w}_k \rangle_N &= \sum_p \sum_{q=0}^{N-1} g_{pN+q} \langle \bar{w}_{pN+q}, \bar{w}_k \rangle_N \\ &= \sum_{q=0}^{N-1} \left(\sum_p g_{pN+q} \right) \langle \bar{w}_q, \bar{w}_k \rangle_N \\ &= \sum_p g_{pN+k}. \end{aligned} \quad (2.4.9)$$

If we now use the damping factor,

$$\begin{aligned} \hat{g}(v) &= \sum_{k=-\infty}^{\infty} \hat{f}(i2\pi(k+v)+a) \\ &= \sum_{j=0}^{\infty} e^{-i2\pi v} e^{-aj} f(j) \end{aligned}$$

we then get

$$\begin{aligned} e^{ak} \langle \hat{g}, \bar{w}_k \rangle_N &= e^{ak} \sum_{q=0}^{N-1} \left(\sum_p g_{pN+q} \right) \langle \bar{w}_q, \bar{w}_k \rangle_N \\ &= g_k + \sum_{p=1}^{\infty} e^{-aNp} g_{pN+k}. \end{aligned}$$

Chapter 2. Fourier transforms inversion and the Poisson summation formula

We can make the approximation error

$$\sum_{p=1}^{\infty} e^{-aNp} g_{pN+k}$$

as small as we want by choosing a large enough. Finally, the result are summarized in the following algorithm:

1. Evaluate $\widehat{g}(v) = \sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v) + a)$, for $v = \frac{j}{N}, j = 0, 1, \dots, N-1$
2. Calculate g_j with the FFT algorithm
3. Set $f(j) = e^{aj} g_j$.

Step 1 of the algorithm is the most difficult step. In the next section we will discuss how we can calculate this sum efficiently using a quadrature formula.

Remark 2.4.1. *Choosing the damping factor a is a tradeoff between numerical stability and accuracy of the algorithm. Since we multiply g_j with e^{aj} in step 3, a too large choice of a leads to numerical instability. On the other hand, the approximation error of the FFT algorithm is given by*

$$\sum_{p \neq 0} e^{-apN} g_{pN+j}.$$

This implies that we have to choose a large enough. If we now take an "oversampling" factor m , i.e. we use an mN point FFT formula, then the error is given by

$$\sum_{p \neq 0} e^{-apmN} g_{pmN+j},$$

while the factor in step 3 of the algorithm is smaller than e^{aN} . An oversampling factor of $m = 8$ and a damping factor of $a = \frac{44}{mN}$ yields excellent results (error is of order 10^{-13}).

Remark 2.4.2. *The damping factor can only be used for functions with support $[0, \infty)$. An important extension is inverting \widehat{g} for a function f with support $(-\infty, \infty)$.*

Remark 2.4.3. *This section is based on den Iseger (2006a). The concept of the damping factor is from Abate en Whitt (1992a).*

Chapter 3

Numerical Transform Inversion Using Gaussian Quadrature

3.1 Introduction

The emphasis on computational probability increases the value of stochastic models in queueing, reliability and inventory problems. It is becoming standard for modelling and analysis to include algorithms for computing probability distributions of interest. Several tools have been developed for this purpose. A very powerful tool is numerical Laplace inversion. Probability distributions can often be characterized in terms of Laplace transforms. Many results in queueing and reliability among others are given in the form of transforms and become amenable for practical computations once fast and accurate methods for numerical Laplace inversion are available. Numerical Laplace inversion is much easier to use than it is often made to seem. This paper presents a new and effective algorithm for numerical Laplace inversion. The new algorithm outperforms existing methods, particularly when the function to be inverted involves discontinuities or singularities as is often the case in applications.

The algorithm can compute the desired function values $f(k\Delta)$, $k = 0, 1, \dots, M - 1$ for a much larger class of Laplace transforms than the ones which can be inverted with the known methods in the literature. It can invert Laplace transforms of functions with discontinuities and singularities, even if we do not know the location of these discontinuities and singularities a priori, and also locally non-smooth and unbounded functions. It only needs numerical values of the Laplace transform, the computations cost $M \log(M)$ time and the results are near machine precision. This is especially useful in applications in computational finance, where one needs to compute a large number of function values by transform inversion (cf. Carr and Madan). With the existing Laplace transform inversion methods this is very expensive. With the new method

one can compute the function values $f(k\Delta)$, $k = 0, 1, \dots, M - 1$, at once in $M \log(M)$ time.

There are many known numerical Laplace inversion algorithms. Four widely used methods are: 1) The Fourier series method which is based on the Poisson Summation formula (cf. Dubner and Abate , Abate and Whitt , Choudhury, Lucantoni and Whitt , O'Kinneide and Sakurai), 2) Gaver-Stehfest algorithm which is based on combinations of Gaver functionals (cf. Gaver and Stehfest), 3) Weeks method which is based on bilinear transformations and Laguerre expansions (cf. Weeks and Abate, Choudhury and Whitt ,) and the Talbot method which is based on deforming the contour in the Bromwich inversion integral (cf. Talbot and Murli and Rizzardi).

The new algorithm is based on the well-known Poisson Summation formula and can therefore be seen as a so-called Fourier series method, developed in the sixties of the last century by Dubner and Abate . The Poisson Summation Formula relates an infinite sum of Laplace transform values to the z-transform (Fourier series) of the function values $f(k\Delta)$, $k = 0, 1, \dots$. Unfortunately, the infinite sum of Laplace transform values converges in general very slowly. In their seminal paper, Abate and Whitt used an acceleration technique called Euler Summation to accelerate the convergence rate of the infinite sum of Laplace transform values. Recently, Sakurai extended Euler summation to be effective for a wider class of functions. A disadvantage of all the variants of the Fourier series methods is that unless one has specified information about the location of singularities, the accelerating techniques are not very effective, and the convergence is slow.

We present a Gaussian quadrature rule for the infinite sum of Laplace transform values. The Gaussian quadrature rule approximates accurately the infinite sum with a finite sum. Then, we compute the function values $f(k\Delta)$, $k = 0, 1, \dots, M - 1$, efficiently with the well-known FFT algorithm (cf.). For smooth functions, the results are near machine precision. With a simple modification we can handle known discontinuities in the points $k\Delta$, $k = 0, 1, \dots$, such that the running time of the algorithm is still insensitive to the number of discontinuities and we get results near machine precision. We also extend the Gaussian quadrature formula for multi-dimensional Laplace transform, and present a multi-dimensional Laplace transform inversion algorithm. With this algorithm we can compute the function values $f(k_1\Delta_1, \dots, k_j\Delta_j)$, $k_j = 0, 1, \dots, M - 1$, at once in $M^j \log(M^j)$ time and the results are again of machine precision. We develop a modification that give results near machine precision for functions with singularities and discontinuities on arbitrary locations. We also develop a modification that is effective for almost all kind of discontinuities, singularities and local non-smoothness. With this modification we can obtain results near machine precision in continuity points for almost all functions, even if we do not know the location of these discontinuities and

3.2 Preliminaries

singularities a priori.

The accurateness and robustness of the algorithms is illustrated by examining all the Laplace transforms that are used in the overview article of Martin and Davies (cf.). We also demonstrate the effectiveness of the algorithm with an application in queuing theory by computing the waiting time distribution for a M/D/1 queue.

The results of this chapter are published in the paper den Iseger (2006a).

3.2 Preliminaries

3.2.1 The concept of Gaussian quadrature

Definition 3.2.1. (Gaussian quadrature, cf. Szegő) Let the inner product $\langle \cdot, \cdot \rangle$ be given by

$$\langle f, g \rangle = \int_I f(t) g^*(t) \mu(dt),$$

with μ a given positive measure and I a subinterval of the real axis or a subinterval of the imaginary axis. Let $\{q_k\}$ be an orthogonal set w.r.t. this inner product. Let the inner product $\langle \cdot, \cdot \rangle_n$ given by

$$\langle f, g \rangle_n := \sum_{k=1}^n \alpha_k f(\mu_k) g^*(\mu_k),$$

with $\{\mu_k; k = 0, 1, \dots, n-1\}$ the simple roots of q_n and the strictly positive numbers $\{\alpha_k; k = 0, 1, \dots, n-1\}$, the so called Christphel numbers, are given by

$$\alpha_k = \frac{1}{\sum_{j=1}^n |q_j(\mu_k)|^2} = \frac{A_n}{A_{n-1}} \frac{1}{\partial q_n(\mu_k)} \frac{1}{q_{n-1}^*(\mu_k)}, \quad (3.2.1)$$

with A_n the highest coefficient in q_n . The roots $\{\mu_k; k = 0, 1, \dots, n-1\} \in I$. The inner product $\langle \cdot, \cdot \rangle_n$ is called the Gaussian quadrature rule and is the unique n th order quadrature rule satisfying

$$\langle p, \mathbf{1} \rangle_n = \langle p, \mathbf{1} \rangle, \text{ for all } p \in \pi_{2n-1},$$

where π_n is the space of polynomials with degree not exceeding n , and $\mathbf{1}$ is the constant function $\mathbf{1}(t) = 1$.

3.2.2 Legendre polynomials

The Legendre polynomials $\{\phi_n; n \in N_0\}$ are a complete orthogonal polynomial set in $L^2([0, 1])$. These polynomials $\{\phi_n; n \in N_0\}$, $\phi_n : [0, 1] \rightarrow \mathbb{R}$ are given by

$$\phi_n(t) = \frac{\sqrt{2n+1}}{n!} D^n (t^n (1-t)^n).$$

Consider the polynomials

$$p_n(s) = \sqrt{2n+1} \sum_{k=0}^n \frac{(k+n)! (-s)^k}{(n-k)! k!}.$$

A routine computation shows that the Laplace transform $\widehat{\phi}_n$ of the n^{th} Legendre polynomial ϕ_n is given by

$$\widehat{\phi}_n(s) = \int_0^1 e^{-st} \phi_n(t) dt = \frac{1}{s} \left(p_n\left(\frac{1}{s}\right) - (-1)^n e^{-s} p_n\left(-\frac{1}{s}\right) \right).$$

Introduce

$$q_n^v(s) = \left(p_n(s) - (-1)^n e^{-2\pi i v} p_n(-s) \right),$$

and $\Psi \widehat{f}(s) = s^{-1} \widehat{f}(s^{-1})$. Since $\exp(-s^{-1}) = \exp(-i2\pi v)$ on $\{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}$, it follows that

$$q_n^v = \Psi \widehat{\phi}_n, \tag{3.2.2}$$

on $\{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}$.

3.3 Outline of the method

Let f be a complex valued Lebesgue integrable function f , with $e^{-\alpha t} f(t) \in L^1[0, \infty)$, for all $\alpha > c$. The Laplace transform of f is defined by the integral

$$\widehat{f}(s) := \int_0^\infty e^{-st} f(t) dt, \operatorname{Re}(s) > c. \tag{3.3.1}$$

The Poisson Summation Formula relates an infinite sum of Laplace transform values to the z-transform (damped Fourier series) of the function values $f(k\Delta)$, $k = 0, 1, \dots$

Theorem 3.3.1. *Suppose that $f \in L^1[0, \infty)$ and f is of bounded variation. Then for all $v \in [0, 1)$,*

$$\sum_{k=-\infty}^\infty \widehat{f}(a + 2\pi i(k+v)) = \sum_{k=0}^\infty e^{-ak} e^{-i2\pi kv} f(k), \tag{3.3.2}$$

here $f(t)$ has to be understood as $\frac{f(t^+) + f(t^-)}{2}$, the so-called damping factor a is a given real number and i denotes the imaginary number $\sqrt{-1}$.

In this chapter we present a Gaussian quadrature rule for the left hand side of formula (3.3.2), that is, we approximate this infinite sum with a finite sum,

$$\sum_{k=1}^n \beta_k \widehat{f}(a + i\lambda_k + 2\pi i v),$$

3.3 Outline of the method

with $\{\beta_k\}$ given positive numbers and the $\{\lambda_k\}$ given real numbers. In table 3.7 the reader can find a table with these numbers for various values of n . We can compute the function values $\{f(k); k = 0, 1, \dots, M-1\}$ by

$$\frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi jk}{M_2}\right) \sum_{l=1}^n \beta_l \hat{f}\left(a + i\lambda_l + \frac{2\pi ij}{M_2}\right), \quad (3.3.3)$$

here M_2 is a given power of two. With the well-known FFT algorithm (cf.) we can compute these sums in $M_2 \log(M_2)$ time. We can compute the function values $f(k\Delta)$, $k = 0, 1, \dots$ by applying (3.3.3) to the scaled Laplace transform

$$\hat{f}_\Delta(s) = \frac{1}{\Delta} \hat{f}\left(\frac{s}{\Delta}\right),$$

which is the Laplace transform of the function $f_\Delta(t) = f(\Delta t)$.

3.3.1 A new Gaussian quadrature

We associate with the left-hand side of formula (3.3.2) an inner product, say $\langle \cdot, \cdot \rangle_{Q_v}$.

Definition 3.3.2. Let the inner product $\langle \cdot, \cdot \rangle_{Q_v}$ be given by

$$\langle f, g \rangle_{Q_v} = \sum_k \frac{1}{|2\pi(k+v)|^2} f\left(\frac{1}{2\pi i(k+v)}\right) g^*\left(\frac{1}{2\pi i(k+v)}\right). \quad (3.3.4)$$

Having $\mathbf{1}(s) = 1$ and $\Psi \hat{f}(s) = s^{-1} \hat{f}(s^{-1})$, we can write the left-hand side of formula (3.3.2) as $\langle \Psi \hat{f}, \Psi \mathbf{1} \rangle_{Q_v}$. The idea is to approximate this inner product with a Gaussian quadrature rule. As usual, we associate with the inner product $\langle \cdot, \cdot \rangle_{Q_v}$ the norm $\|\cdot\|_{Q_v}$ which induced the sequence space $L^2(Q_v)$. We say that a function f belongs to $L^2(Q_v)$ iff the sequence $\{f(\frac{1}{2\pi i(k+v)})\}$ belongs to $L^2(Q_v)$. Let the polynomials $\{q_n^v; n \in N_0\}$ be given by

$$q_n^v(s) := p_n(s) - (-1)^n e^{-i2\pi v} p_n(-s),$$

where

$$p_n(s) = \sqrt{2n+1} \sum_{k=0}^n \frac{(k+n)! (-s)^k}{(n-k)! k!}.$$

The following result holds:

Theorem 3.3.3. The set $\{q_j^v; j = 0, 1, \dots\}$ is a complete orthogonal set of polynomials in $L^2(Q_v)$.

The proof is given in Appendix 3.7.

We can now easily obtain the desired Gaussian quadrature rule, cf. definition 3.2.1 in Appendix 3.2.1. We denote this quadrature rule with $\langle \cdot, \cdot \rangle_{Q_v^n}$.

Definition 3.3.4. The inner product $\langle \cdot, \cdot \rangle_{Q_v^n}$ is given by

$$\langle f, g \rangle_{Q_v^n} = \sum_{k=1}^n \alpha_k^v f(\mu_k^v) g^*(\mu_k^v). \quad (3.3.5)$$

The $\{\mu_k^v\}$ are the zeros of the polynomial q_n^v . The so called Christphel numbers $\{\alpha_k^v\}$ are positive numbers given by

$$\alpha_k^v = \frac{1}{\sum_{j=1}^n |q_j^v(\mu_k^v)|^2}. \quad (3.3.6)$$

It is well known, cf. Szegő, that the roots of orthogonal polynomials are all distinct and lie on the support of the inner product, thus the roots $\{\mu_k^v\}$ are all distinct and lie on the imaginary axis. Having $\mathbf{1}(s) = 1$ and $\Psi \hat{f}(s) = s^{-1} \hat{f}(s^{-1})$, we approximate $\langle \Psi \hat{f}, \Psi \mathbf{1} \rangle_{Q_v}$ with our Gaussian quadrature rule and obtain the quadrature formula

$$\sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \hat{f}\left(\frac{1}{\mu_k^v}\right) \approx \sum_{k=-\infty}^{\infty} \hat{f}(2\pi i(k+v)) = \sum_{k=0}^{\infty} \exp(-2\pi i k v) f(k),$$

where the last identity follows from the PSF formula (3.3.2). In appendix 3.8 we explain how we can compute the numbers $\{\mu_k^v\}$ and $\{\alpha_k^v\}$ efficiently. Considering only the real part yields the quadrature formula

$$\sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \operatorname{Re}\left(\hat{f}\left(\frac{1}{\mu_k^v}\right)\right) \approx \sum_{k=0}^{\infty} \cos(2\pi k v) f(k), \quad (3.3.7)$$

we used here that the α_k^v are real. Similarly to the Abate and Whitt algorithm (cf.), we use a damping factor for functions with support contained in $[0, \infty)$, that is we take the Laplace transform $\hat{f}(a+s)$, which is the Laplace transform of $e^{-at} f(t)$. This yields finally the quadrature rule

$$F_a(v) \approx \sum_{k=0}^{\infty} e^{-ak} \cos(2\pi k v) f(k), \quad (3.3.8)$$

where

$$F_a(v) = \sum_{k=1}^n \frac{\alpha_k^v}{|\mu_k^v|^2} \operatorname{Re}\left(\hat{f}\left(a + \frac{1}{\mu_k^v}\right)\right).$$

It remains to compute the function values $f(k)$, $k = 0, 1, \dots, M-1$. We can realize this by the discrete Fourier series inversion formula

$$f(k) \approx \frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi\left(j + \frac{1}{2}\right)k}{M_2}\right) F_a\left(\frac{j + \frac{1}{2}}{M_2}\right), \quad (3.3.9)$$

with M_2 a given power of two. With the well-known FFT algorithm (cf.) we can compute the sums (3.3.9) in $M_2 \log(M_2)$ time. In section 3.10 we discuss the details of this approach.

3.4 A simple Laplace transform inversion algorithm

Remark 3.3.5. The quadrature rule is also valid for a function f with support $(-\infty, \infty)$. The formula (3.3.8) reads then as

$$F_a(v) \approx \sum_{k=-\infty}^{\infty} \cos(2\pi kv) e^{-ak} f(k\Delta).$$

Remark 3.3.6. For smooth functions on $(-\infty, \infty)$ or $[0, \infty)$ the quadrature formula is extremely accurate. In appendix 3.9 and in particular theorem 3.9.3 we prove that for smooth functions the approximation error of formula (3.3.8) is approximately

$$\frac{1}{2} (-1)^n \frac{(n-1)!n!}{(2n-1)!} \sum_{j=1}^{\infty} j e^{-i2\pi jv} \frac{f^{(2n+1)}(j-1+\alpha)}{(2n+1)!}, \quad (3.3.10)$$

with α some number in $(0, 2)$. This implies that if we compute the function values $f(k\Delta), k = 0, 1, \dots$, by inverting the Laplace transform

$$\hat{f}_{\Delta}(s) = \frac{1}{\Delta} \hat{f}\left(\frac{s}{\Delta}\right),$$

with formula (3.3.9), the approximation error is of the order $O(\Delta^{2n+1})$. Formula (3.3.10) shows also that if

$$\frac{f^{(2n+1)}(j-1+\alpha)}{(2n+1)!}$$

is bounded then the quadrature formula converges faster than any power of n

Remark 3.3.7. The convergence of the quadrature rule is insensitive for discontinuity in $t = 0$ (cf. rem 3.9.4 in appendix 3.9).

Remark 3.3.8. Gaussian quadrature is a standard method for the numerical evaluation of integrals (cf. Stoer and Bulirsch). Piessens (cf. and) tried to apply Gaussian quadrature directly on the Bromwich inversion integral. He reports in Piessens for the first Laplace transform of table 3.1 (cf. section 3.4.1) an average approximation error of $1.0e-4$ (he uses a 17 points quadrature rule and compute the function values in the points $t=0,1,\dots,12$). Our method gives an average approximation error of $1.0e-15$ (cf. table 3.1).

3.4 A simple Laplace transform inversion algorithm

In this section we explain how we can approximate the quadrature rule (3.3.7) with a simpler quadrature rule. Besides that this approximation is easier to implement, it is also numerically more stable. Therefore, we strongly recommend to use this quadrature rule.

We start with writing the PSF (cf. formula (3.3.2)) as

$$\sum_{k=-\infty}^{\infty} \widehat{f}_{\rho_v^a}(i(2k+1)\pi) = \sum_{k=0}^{\infty} e^{-ak} e^{-i2\pi kv} f(k), \quad (3.4.1)$$

with

$$\widehat{f}_{\rho_v^a}(s) = \widehat{f}(\rho_v^a + s), \rho_v^a = a + 2\pi i \left(v - \frac{1}{2} \right). \quad (3.4.2)$$

Applying the quadrature rule on the lhs of formula (3.4.1) yields the approximation

$$\sum_{k=1}^n \beta_k \widehat{f}(\lambda_k + a + 2\pi i v) \approx \sum_{k=0}^{\infty} e^{-i2\pi kv} e^{-ak} f(k), \quad (3.4.3)$$

with

$$\lambda_k = \frac{1}{i\mu_k^{0.5}} - \pi \text{ and } \beta_k = \frac{\alpha_k^{0.5}}{|\mu_k^{0.5}|^2}. \quad (3.4.4)$$

Considering only the real part yields the formula

$$F_a(v) \approx \sum_{k=0}^{\infty} \cos(2\pi kv) e^{-ak} f(k), \quad (3.4.5)$$

with

$$F_a(v) = \sum_{k=1}^n \beta_k \operatorname{Re} \left(\widehat{f}(a + \lambda_k + 2\pi i v) \right) \quad (3.4.6)$$

We make F_a periodic by defining $F_a(0)$ and $F_a(1)$ equal to

$$\sum_{k=1}^n \beta_k \frac{1}{2} \operatorname{Re} \left(\widehat{f}(a + \lambda_k) + \widehat{f}(a + \lambda_k + 2\pi i) \right).$$

It remains to compute the function values $f(k)$, $k = 0, 1, \dots, M-1$. We can realize this by the discrete Fourier series inversion formula

$$f(k) \approx \frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi jk}{M_2}\right) F_a\left(\frac{j}{M_2}\right), \quad (3.4.7)$$

with M_2 a given power of two.

Finally, we show how to simplify formula (3.4.7). Since the coefficients of the polynomials $q_k^{0.5}$, $k = 1, 2, \dots$, are real, we can order the roots $\{\mu_k^v\}$ such that the $\mu_k^{0.5}$ and $\mu_{n-k}^{0.5}$ are pairs of conjugate numbers. It follows from formula (3.3.6) that the Christoffel numbers $\alpha_k^{0.5}$ and $\alpha_{n-k}^{0.5}$ are equal. Using this symmetry, formula (3.4.4), and the symmetry $\cos(2\pi v) = \cos(2\pi(1-v))$ yields (for n is even) that formula (3.4.7) can be simplified to

$$f(k) \approx \frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi jk}{M_2}\right) \widetilde{F}_a\left(\frac{j}{M_2}\right), \quad (3.4.8)$$

3.4 A simple Laplace transform inversion algorithm

where

$$\tilde{F}_a(v) = \begin{cases} 2 \sum_{k=1}^{\frac{n}{2}} \beta_k \operatorname{Re} \left(\hat{f}(a + i\lambda_k + i2\pi v) \right), & 0 < v < 1 \\ \sum_{k=1}^{\frac{n}{2}} \beta_k \operatorname{Re} \left(\hat{f}(a + i\lambda_k) + \hat{f}(a + i\lambda_k + i2\pi) \right), & v = 0. \end{cases}$$

In table 3.7 the reader can find a table with these numbers for various values of n . With the well-known FFT algorithm (cf. Cooley and Tukey) we can compute the sums (3.4.8) in $M_2 \log(M_2)$ time. In section 3.10 we discuss the details of the discrete Fourier inversion formula (3.4.8). We can now present our Laplace inversion algorithm.

Algorithm

input: \hat{f} , Δ and M , with M a power of two;

output: $f(\ell\Delta)$, $\ell = 0, 1, \dots, M-1$;

parameters: $M_2 = 8M$ and $a = \frac{44}{M_2}$ and $n = 16$;

Step 1. For $k = 0, 1, \dots, M_2$ and $j = 1, 2, \dots, \frac{n}{2}$ compute the numbers

$$\hat{f}_{jk} = \operatorname{Re} \left[\hat{f} \left(\frac{a + i\lambda_j + \frac{2i\pi k}{M_2}}{\Delta} \right) \right], \hat{f}_k = \frac{2}{\Delta} \sum_{j=1}^{\frac{n}{2}} \beta_j \hat{f}_{jk} \text{ and } \hat{f}_0 = \frac{1}{\Delta} \sum_{j=1}^{\frac{n}{2}} \beta_j (\hat{f}_{j0} + \hat{f}_{jM_2}).$$

Step 2. For $\ell = 0, 1, \dots, M_2 - 1$ compute the numbers

$$f_\ell = \frac{1}{M_2} \sum_{k=0}^{M_2-1} \hat{f}_k \cos \left(\frac{2\pi\ell k}{M_2} \right)$$

with the backwards FFT algorithm;

Step 3. Set $f(\ell\Delta) = e^{a\ell} f_\ell$ for $\ell = 0, 1, \dots, M-1$.

In appendix 3.10 we discuss the details of step 2 and 3 of the above algorithm.

Remark 3.4.1. We can compute the function values $f(k\Delta)$, $k = 0, 1, \dots$, by applying the quadrature rule to the scaled Laplace transform

$$\hat{f}_\Delta(s) = \frac{1}{\Delta} \hat{f} \left(\frac{s}{\Delta} \right).$$

Remark 3.4.2. Extensive numerical experiments show that $n = 16$ gives for all smooth functions results attaining the machine precision. For double precision we choose $a = \frac{44}{M_2}$ and $M_2 = 8M$. For $n = 16$ we need 8 Laplace transform values for the quadrature rule and we use a over sampling factor of $M_2/M = 8$, thus on average we need 64 Laplace transform values for the computation of one function value. The method can be efficiently implemented with modern numerical libraries with FFT routines and matrix/vector multiplication routines (BLAS). If we are satisfied with less precision then we can reduce the oversampling factor and/or the number of quadrature point. For other machine precisions it is recommend to choose a somewhat larger than $\frac{-\log(\epsilon)}{M_2}$, with ϵ the machine precision.

Remark 3.4.3. The choice of the parameter M_2 is a trade off between the running time and the accuracy of the algorithm. The actual running time is $M_2 \log(M_2)$, hence from this perspective we want to chose M_2 as small as possible. However in step 3 of the algorithm we multiply the results with a factor $\exp(a\ell) = \exp(\frac{44}{M_2}\ell)$; to obtain a numerically stable algorithm we want to chose M_2 as big as possible. We recommend to chose $M_2 = 8M$.

Remark 3.4.4. Since the Gaussian quadrature rule is exact on the space of polynomials of degree $2n-1$ or less, we obtain that

$$\sum_{k=1}^n \alpha_k^v = \langle \mathbf{1}, \mathbf{1} \rangle_{Q_v^!} = \langle \mathbf{1}, \mathbf{1} \rangle_{Q_v} = \frac{\langle q_0^v, q_0^v \rangle}{1 - \cos(2\pi v)} = \frac{1}{1 - \cos(2\pi v)}.$$

This is clearly minimal for $v = 0.5$. This shows that formula (3.4.6) is numerically more stable than the original Gaussian quadrature formula.

Remark 3.4.5. If n is odd then one of the $\{\mu_k^{0.5}; k = 1, 2, \dots, n\}$ is equal to 0. The evaluation of formula (3.3.7) can be a problem in this case. To avoid this, we take n even.

Remark 3.4.6. The Gaussian quadrature formula (3.3.8) is extremely accurate for smooth inverse functions (cf. appendix 3.9 and rem 3.3.6). Since the smoothness of the inverse function f implies that the function

$$e^{-s_v^a t} f(t), \text{ with } s_v^a = a + 2\pi v i \tag{3.4.9}$$

is a smooth function too, we can expect that the modified quadrature formula is extremely accurate for smooth functions. All the numerical examples support this statement (cf. section 3.4.1). In fact many numerical experiments show that the modified formula performs even better. The reason for this is that the modified quadrature formula is numerically more stable.

3.4.1 Numerical test results

To test the method we have examined all the Laplace transforms that are used in the overview article of Martin and Davies (cf.). In this section we discuss the results for eight analytical test functions. These results are presented in Table 3.1. The inversions are done in 32 points and so $M = 32$. We have taken $M_2 = 8M = 256$, $a = 44/M_2$ and $n = 16$. The computations are done with double precision. We have taken $\Delta = 1/16$, $\Delta = 1$ and $\Delta = 10$. We see that for smooth functions the approximation error seems to be independent of Δ . The reason for this is that the approximation error is dominated with round-off error. From all the examples we can conclude that the method is very accurate, robust and fast. In section 3.5.2 we discuss the other test functions that are used in the overview article of Martin and Davies (cf.).

3.5 Modifications for non-smooth functions

Table 3.1: Results for analytical test functions

nr	function	Laplace Transform
1	$J_0(x)$	$(s^2 + 1)^{-1/2}$
2	$\exp(-x/2)$	$(s + 1/2)^{-1}$
3	$\exp(-0.2x) \sin(x)$	$((s + 0.2)^2 + 1)^{-1}$
4	1	s^{-1}
5	x	s^{-2}
6	$x \exp(-x)$	$(s + 1)^{-2}$
7	$\sin(x)$	$(s^2 + 1)^{-1}$
8	$x \cos(x)$	$(s^2 - 1)(s^2 + 1)^{-2}$

nr	$\Delta = \frac{1}{16}$	$\Delta = 1$	$\Delta = 10$
1	1e-15	1e-15	5e-13
2	1 e-15	1e-15	3e-16
3	2e-16	1e-15	3e-12
4	1 e-15	1e-15	1e-15
5	3 e-15	5e-15	6e-15
6	2 e-16	3e-16	2e-16
7	1 e-15	1e-15	5e-12
8	4 e-16	6e-15	2e-12

3.5 Modifications for non-smooth functions

In appendix 3.9 we prove that the inversion algorithm is extremely accurate for smooth functions. In the next sections we present a number of modifications to obtain the same accurate results for non-smooth functions. In section 3.5.1 we discuss a modification for piecewise smooth functions. With this modification we can obtain results near machine precision for functions with discontinuities in the points $k\Delta$, $k \in \mathbb{Z}$. In section 3.5.2 we discuss a modification for functions with singularities at arbitrary locations. With this modification we can obtain results near machine precision for functions with discontinuities at arbitrary but a priori known locations. In section 3.5.3 we discuss a modification that is effective for almost all kind of discontinuities, singularities and local non-smoothness. With this modification we can obtain results near machine precision in continuity points for almost all functions, even if we do not know the discontinuities and singularities a priori. The modifications of the next sections can in principle also be used in combination with other Fourier series methods. The effectiveness of the modifications in combination with other Fourier series methods is a subject for future

research.

3.5.1 Piecewise smooth functions

In this section we discuss a modification for piecewise smooth functions. With this modification we can obtain results near machine precision for functions with discontinuities in the points $k\Delta, k \in \mathbb{Z}$. The discontinuities of such a function are represented as exponentials in the Laplace transform. To be more precise, suppose that the function f is a piecewise smooth function with discontinuities in the points $k, k \in \mathbb{Z}$. We can write the Laplace transform \widehat{f} as

$$\widehat{f}(s) = V(s, e^{-s}) = \sum_j e^{-js} \widehat{f}_j(s),$$

with the \widehat{f}_j the Laplace transforms of functions which are smooth on $[0, \infty)$. The function f is then given by

$$f(t) = \sum_j f_j(t - j),$$

we use here the translation property. Since the f_j are smooth functions on $[0, \infty)$, the quadrature rule is accurate. Hence,

$$\begin{aligned} \sum_{k=1}^n \beta_k e^{-j(a+2\pi iv)} \widehat{f}_j(i\lambda_k + s_v^a) &\approx \sum_{k=0}^{\infty} e^{-a(k+j)} e^{-i2\pi(k+j)v} f_j(k) \\ &= \sum_{k=j}^{\infty} e^{-ak} e^{-i2\pi kv} f_j(k - j), \end{aligned}$$

with $s_v^a = a + 2\pi iv$. On the other hand

$$\begin{aligned} \sum_j \sum_{k=1}^n \beta_k e^{-j(a+2\pi iv)} \widehat{f}_j(i\lambda_k + s_v^a) &= \sum_{k=1}^n \sum_j \beta_k e^{-j(a+2\pi iv)} \widehat{f}_j(i\lambda_k + s_v^a) \\ &= \sum_{k=1}^n \beta_k V(i\lambda_k + s_v^a, e^{-a+2\pi iv}). \end{aligned}$$

Combining the above equations yields

$$\sum_{k=0}^{\infty} e^{-ak} e^{-i2\pi kv} f(k) \approx \sum_{k=1}^n \beta_k V(i\lambda_k + s_v^a, e^{-a-2\pi iv}).$$

We now proceed as in section 3.4 and compute the function values $f(k), k = 0, 1, \dots, M-1$, with

$$\frac{e^{ak}}{M_2} \sum_{j=0}^{M_2-1} \cos\left(\frac{2\pi jk}{M_2}\right) F_a\left(\frac{j}{M_2}\right),$$

3.5 Modifications for non-smooth functions

where

$$F_a(v) = \sum_{k=1}^n \beta_k \operatorname{Re} \left(V \left(i\lambda_k + a + 2\pi i v, e^{-a-2\pi i v} \right) \right).$$

Remark 3.5.1. If f is piecewise smooth, with discontinuities in the points $k\alpha$, $k \in \mathbb{Z}$, we can write the Laplace transform \hat{f} as $\hat{f}(s) = V(s, e^{-s\alpha})$. We scale the Laplace transform to

$$\frac{m}{\alpha} \hat{f} \left(\frac{sm}{\alpha} \right) = \frac{m}{\alpha} V \left(\frac{sm}{\alpha}, e^{-sm} \right),$$

and can obtain the values $\{f(\frac{k\alpha}{m})\}$ with high precision.

As an example consider the Laplace transform

$$\hat{f}(s) := \frac{e^{-s}}{s}.$$

We then obtain the quadrature formula

$$\sum_{k=1}^n \beta_k \operatorname{Re} \left(\frac{e^{-am} e^{-i2\pi mv}}{a + i\lambda_k + 2\pi i v} \right).$$

Algorithm

input: $V: f(s) = \hat{V}(s, e^{-s\Delta})$, Δ and M , with M a power of two;

output: $f(\ell\Delta)$, $\ell = 0, 1, \dots, M-1$;

parameters: $M_2 = 8M$ and $a = \frac{44}{M_2}$ and $n = 16$;

Step 1. For $k = 0, 1, \dots, M_2$ and $j = 1, 2, \dots, n$ compute the numbers

$$\hat{f}_{jk} = \operatorname{Re} \left[V \left(\frac{a + i\lambda_j + \frac{2i\pi k}{M_2}}{\Delta}, e^{-\left(a + \frac{2i\pi k}{M_2}\right)} \right) \right],$$

and the numbers

$$\hat{f}_k = \frac{1}{\Delta} \sum_{j=1}^n \beta_j \hat{f}_{jk} \text{ and } \hat{f}_0 = \frac{\hat{f}_0 + \hat{f}_{M_2}}{2}.$$

Step 2. For $\ell = 0, 1, \dots, M_2 - 1$ compute the numbers

$$f_\ell = \frac{1}{M_2} \sum_{k=0}^{M_2-1} \hat{f}_k \cos \left(\frac{2\pi \ell k}{M_2} \right)$$

with the backwards FFT algorithm;

Step 3. Set $f(\ell\Delta) = e^{a\ell} f_\ell$ for $\ell = 0, 1, \dots, M-1$.

Table 3.2: Results for non-continuous test functions

nr	function	Laplace Transform	MAE
17	$H(t - 1)$	$s^{-1} \exp(-s)$	2e-15
18	square wave	$(s(1 + \exp(-s)))^{-1}$	8e-15

In Appendix 3.10 we discuss the details of step 2 and 3 of the above algorithm.

We test this modification on two discontinuous test functions. The results are presented in table 3.2. The inversions are done in 32 points and so $M = 32$. We have taken $M_2 = 256 = 8M$ and $\Delta = 1/16$ and we have taken $a = 44/M_2$. Besides the choice of the previous parameters we have taken $n = 16$.

We can conclude that our method is also very accurate, robust and fast for discontinuous functions. In the next section we show the effectiveness of the modification of this section with an application from queuing theory.

An application in queuing theory

Consider the M/G/1 queue arrival rate λ and general service time distribution B with first moment μ_B . It is assumed that $\rho = \lambda\mu_B < 1$. It is well known that the distribution of the stationary waiting time \mathbf{W} is given by (see e.g. Asmussen cf.):

$$W(x) := P\{\mathbf{W} \leq x\} = (1 - \rho) \sum_{n=0}^{\infty} \rho^n B_0^{*n}(x), \quad (3.5.1)$$

with $B_0(x) = \frac{1}{\mu_B} \int_0^x (1 - B(t)) dt$. The Laplace transform \widehat{W} of W is given by

$$\widehat{W}(s) = \frac{1}{s} \frac{1 - \rho}{1 - \rho \left(\frac{1 - \widehat{B}(s)}{s\mu_B} \right)}. \quad (3.5.2)$$

As an example consider the M/D/1 queue, i.e. the service times are deterministic, with mean 1. For this case the Laplace transform \widehat{W} is given by

$$\widehat{W}(s) = \frac{1}{s} \frac{1 - \rho}{1 - \rho(1 - e^{-s})/s}. \quad (3.5.3)$$

Observe that $\widehat{W}(s)$ contains the factor $\exp(-s)$. Therefore, introduce the function

$$V(s, z) = \frac{1}{s} \frac{1 - \rho}{1 - \rho(1 - z)/s}. \quad (3.5.4)$$

For this choice it follows that $\widehat{W}(s) = v(s, \exp(-s))$ and therefore we can apply the modification of section 3.5.1. We have calculated the waiting time distribution for $\rho = 0.7, 0.8, 0.9$ and 0.95 . By comparing our results with results of the algorithm in Tijms [p. 381], it appears that we can calculate the waiting time distribution with almost machine accuracy in a fraction of a second, see table 3.3.

3.5 Modifications for non-smooth functions

Table 3.3: Results for the waiting time distribution in the M/D/1 queue

ρ	MAE
0.7	3e-14
0.8	5e-14
0.9	8e-14
0.95	1e-13

Remark 3.5.2. We can generalize the above approach to a service distribution

$$D(x) = \sum_{k=1}^{\infty} p_k \mathbf{1}_{\{x \leq k\}}.$$

we then obtain

$$V(s, z) = \frac{1}{s} \frac{1 - \rho}{1 - \rho \left(1 - \sum_{k=0}^{\infty} p_k z^k\right) / s}.$$

3.5.2 Functions with singularities

In this section we discuss a modification for functions with singularities at arbitrary locations. With this modification we can obtain results near machine precision for functions with discontinuities at arbitrary but a priori known locations. Suppose that there is a singularity in $t = \alpha$, with $\alpha \in \mathbb{R}$. We then consider the function

$$f_w(x) := w(x)^q f(x), \quad (3.5.5)$$

where the so called window function w is a trigonometric polynomial with period 1, $w(0) = 1$, and $w(\alpha) = 0$, and q is a positive integral number. Hence the function f_w is smooth in $t = \alpha$ and

$$f_w(k) = f(k), \quad k = 0, 1, \dots$$

We can compute the function values $f(k)$ by inverting the Laplace transform \widehat{f}_w with the algorithm of section 3.4. The following class of window functions gives results near machine precision:

$$\begin{aligned} w(x) &= \left(\cos(p\pi x) - \frac{\cos(p\pi\alpha)}{\sin(p\pi\alpha)} \sin(p\pi x) \right)^2 \\ &= \left(Ae^{ip\pi x} + Be^{-ip\pi x} \right)^2, \quad p \in \mathbb{N}, \end{aligned}$$

where the coefficients A and B are given by

$$A = \left(\frac{1}{2} - \frac{1}{2i} \frac{\cos(p\pi\alpha)}{\sin(p\pi\alpha)} \right) \quad \text{and} \quad B = \left(\frac{1}{2} + \frac{1}{2i} \frac{\cos(p\pi\alpha)}{\sin(p\pi\alpha)} \right),$$

Chapter 3. Numerical Transform Inversion Using Gaussian Quadrature
 and p is chosen such that $(p\alpha \bmod 1)$ is close to $\frac{1}{2}$. We obtain by the modulation property that

$$\widehat{f}_w(s) = \sum_{k=0}^{2q} \binom{2q}{k} A^{2q-k} B^k \widehat{f}(s + 2\pi p(q-k)).$$

Remark 3.5.3. Suppose that there are singularities in the points α_j , $j = 1, 2, \dots, m$. Then we can multiply the windows, that is we take

$$w(x) = \prod_{j=1}^m w_j^q(x),$$

where w_j is a trigonometric polynomial with period 1, $w_j(0) = 1$, and $w_j(\alpha_j) = 0$.

Remark 3.5.4. If we want to compute $f(k\Delta)$, we have to replace \widehat{f}_w with $\frac{1}{\Delta}\widehat{f}_w(\frac{s}{\Delta})$.

If there is a singularity in $t = 0$, then A and B are not defined. Consider therefore the function

$$w(x) = \sin^2\left(\frac{\pi x}{2}\right). \quad (3.5.6)$$

The function w has period 2, $w(1) = 1$, $w(\alpha) = 0$ and $\frac{\partial w(\alpha)}{\partial \alpha} = 0$. Hence f_w is smooth in $t = 0$ and

$$f_w(2k+1) = f(2k+1), k = 0, 1, \dots$$

We obtain from

$$\begin{aligned} \left(\sin\left(\frac{\pi x}{2}\right)\right)^{2q} &= \left(\frac{e^{i\frac{\pi}{2}x} - e^{-i\frac{\pi}{2}x}}{2i}\right)^{2q} \\ &= \left(\frac{1}{2}\right)^{2q} \sum_{k=0}^{2q} \binom{2q}{k} (-1)^{q-k} e^{-i\pi(q-k)x}, \end{aligned}$$

and the modulation property that

$$\widehat{f}_w(s) = \left(\frac{1}{2}\right)^{2q} \sum_{k=0}^{2q} \binom{2q}{k} (-1)^{q-k} \widehat{f}(s + i\pi(q-k)).$$

We can compute the function values $f(2k-1)$, with the algorithm of section 3.4. If we want to compute $f((2k-1)\Delta)$, we have to replace \widehat{f}_w with $\frac{1}{\Delta}\widehat{f}_w(\frac{s}{\Delta})$.

Remark 3.5.5. For singularities at $t = 0$ of the order x^α , $0 < \alpha < 1$, we can obtain with $q = 1$ results near machine precision. For singularities at $t = 0$ of the order x^α , $-1 < \alpha < 0$, we can obtain with $q = 2$ results near machine precision. For discontinuities and singularities at arbitrary t we recommend to use $q = 6$.

3.5 Modifications for non-smooth functions

Remark 3.5.6. Since the composite inverse function $f_w(t\Delta)$ is a highly oscillating function, we recommend to choose $n = 32$. With this choice we obtain results near machine precision. If the inversion point is close to a singularity, we can compute $f(t)$ accurately by inverting the scaled Laplace transform

$$f_w(t) = w(t) f(\Delta t),$$

with Δ small enough. The price we have to pay for a small Δ is that M became large ($M \approx \frac{t}{\Delta}$).

Remark 3.5.7. The idea of smoothing through the use of multiplying smoothing functions has been discussed in the context of Euler summation by OCinneide (1997) and Sakurai (2003). OCinneide called the idea product smoothing.

We ended this section with the results for eight test functions with singularities, that are used in the overview article of Martin and Davies (cf.). These results are presented in table 3.4. The inversions are done in 32 points and so $M = 32$. We have taken $M_2 = 256 = 8M$ and for all the examples we have taken $a = 44/M_2$. We have used the window function $w(t) = \sin(\pi t)^2$. Besides the choice of the previous parameters we have taken $n = 32$. The computations are done with double precision. We have taken $\Delta = 1/16$, $\Delta = 1$ and $\Delta = 10$. We see that for smooth functions the approximation error seems to be independent of Δ . The reason for this is that the approximation error is dominated with round-off error and the error caused by the oscillation of the window function. From all the examples we can conclude that the method is very accurate, robust and fast.

From all the examples we can conclude that the method is very accurate, robust and fast for functions with singularities.

3.5.3 A robust Laplace inversion algorithm

In this section we discuss a modification that is effective for almost all kind of discontinuities, singularities and local non-smoothness. With this modification we can obtain results near machine precision in continuity points for almost all functions, even if we do not know the discontinuities and singularities a priori. Similarly as in section 3.5.2 we compute the function value $f(k)$ by inverting the Laplace transform of the function

$$f_{w_k}(t) = w_k(t) f(t),$$

but now the window function depends on the point k . That is, we use for the computation of each function value $f(k)$ a different window function. We choose the window function such that the ϵ -support $\{t; |e^{-\alpha t} f_w(t)| \geq \epsilon\}$ of f_{w_k} is contained in $[k - \delta, k + \delta]$, with δ given positive control parameters, ϵ a given tolerance, and $w_k(k) = 1$. The advantage of this construction is that it is sufficient that the function f is smooth on

Table 3.4: Results for continuous non-analytical test functions

nr	function	Laplace Transform		
9	$(\pi x)^{-1/2} \cos(2x^{1/2})$	$s^{-1/2} \exp(-s^{-1})$		
10	$(\pi x)^{-1/2}$	$s^{-1/2}$		
11	$-\gamma - \ln(x)$	$s^{-1} \ln(s)$		
12	$(e^{-x/4} - e^{-x/2})(4\pi x^3)^{-1/2}$	$(s + 1/2)^{1/2} - (s + 1/4)^{1/2}$		
13	$2e^{-4/t}(\pi x^3)^{-1/2}$	$\exp(-4s^{1/2})$		
14	$\sin(x)/x$	$\arctan(1/s)$		
15	$x^{1/3}$	$\Gamma(\frac{4}{3})s^{-\frac{4}{3}}$		
16	$x^{1/4}$	$\Gamma(\frac{5}{4})s^{-5/4}$		

nr	$\Delta = \frac{1}{16}$	$\Delta = 1$	$\Delta = 10$
9	3e-14	8e-15	3e-15
10	1e-14	4e-15	4e-15
11	2e-15	1e-14	2e-14
12	3e-15	8e-16	4e-16
13	3e-16	4e-16	1e-14
14	1e-14	1e-15	7e-16
15	9e-15	1e-14	2e-14
16	8e-15	1e-14	2e-14

$[k - \delta, k + \delta]$, in order that the function f_{w_k} is smooth on $[0, \infty)$. Thus, we only need that the function f is smooth on $[k - \delta, k + \delta]$ to compute $f(k)$ accurately with the quadrature rule. Hence we can obtain results near machine precision in continuity points for almost all functions.

A second advantage is that we can efficiently invert the resulting z-transform. As the ϵ -support of f_{w_k} is contained in $[k - \delta, k + \delta]$ implies that

$$\begin{aligned} \sum_{j=1}^n \beta_j \operatorname{Re} \left(\widehat{f}_{w_k} (i\lambda_j + 2\pi i v + a) \right) &\approx \sum_{j=0}^{\infty} w_k(j) f(j) e^{-ja} \cos(2\pi j v) \\ &= \sum_{j \in [k-\delta, k+\delta]} w_k(j) f(j) e^{-ja} \cos(2\pi j v), \end{aligned}$$

(the first equation follows from formula (3.4.5)) and the N -points discrete Fourier transform computes the numbers

$$\tilde{f}_k = \sum_j f_{k+jN},$$

(cf. section 3.10), we can compute the function values $f(k)$ efficiently with a N points discrete Fourier transform ($N \geq 1 + 2 \lfloor \delta \rfloor$).

3.5 Modifications for non-smooth functions

We can construct the window functions with the desired properties as follows: Let w be a smooth bounded function satisfying

$$|w(t)| \leq \epsilon, t \in \left[-\frac{P}{2}, -\delta_l\right] \cup \left[\delta_u, \frac{P}{2}\right], w(0) = 1,$$

with P a given positive number. We chose the damping factor α such that

$$|e^{-\alpha t} f(t)| \leq \frac{\epsilon}{\sup_{x \in \left[-\frac{P}{2}, \frac{P}{2}\right]} |w(x)|}, \text{ for } t > P. \quad (3.5.7)$$

We extend w to a periodic function with period P by setting

$$w(t+k) = w(t), t \in \left[-\frac{P}{2}, \frac{P}{2}\right) \text{ and } k \in \mathbb{Z}.$$

The desired window function w_k is defined by

$$w_k(t) = w(t-k).$$

We will now show how we can efficiently evaluate the Fourier series F_k ,

$$F_k(v) = \sum_{j=1}^n \beta_j \operatorname{Re} \left(\widehat{f}_{w_k}(i\lambda_j + 2\pi i v) \right),$$

for $k = 0, 2, \dots, M-1$. Since w has period P and w is smooth, we can expand w in a fast converging Fourier series. Thus we can write

$$w_k(t) = w(t-k) = \sum_{j=-\infty}^{\infty} e^{\frac{2\pi i j k}{P}} A_j e^{-\frac{2\pi i j t}{P}},$$

with

$$A_j = \frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} w(t) e^{\frac{2\pi i j t}{P}} dt.$$

This yields that the Laplace transform of f_{w_k} is given by

$$\widehat{f}_{w_k}(s) = \sum_{j=-\infty}^{\infty} e^{\frac{2\pi i j k}{P}} A_j \widehat{f} \left(s + \frac{2\pi i j}{P} + a \right). \quad (3.5.8)$$

Since w is a smooth periodic function, the $\{A_j\}$ converge rapidly to 0. Hence, we can approximate \widehat{f}_{w_k} accurately with

$$\widehat{f}_{w_k}(s) = \sum_{j=-J}^J e^{\frac{2\pi i j k}{P}} A_j \widehat{f} \left(s + \frac{2\pi i j}{P} + a \right), \quad (3.5.9)$$

for J big enough. Hence,

$$F_k(v) = \operatorname{Re} \left(\sum_{j=-J}^J e^{\frac{2\pi i j k}{P}} A_j G \left(v + \frac{j}{P} \right) \right),$$

with

$$G(v) = \sum_{k=1}^n \beta_k \widehat{f}(a + \lambda_k + 2\pi i v).$$

Moreover, if J is a multiple of P then

$$F_k(v) = \operatorname{Re} \left(\sum_{l=0}^{P-1} e^{\frac{2\pi i l k}{P}} \sum_{m=-L}^{L-1} A_{mP+l} G \left(v + m + \frac{l}{P} \right) \right), \quad (3.5.10)$$

with $L = \frac{J}{P}$. We make F_k periodic by defining $F_k(0)$ and $F_k(1)$ equal to

$$\frac{1}{2} (F_k(0) + F_k(1)).$$

We can compute the sums (3.5.10) in $L \log(L) + J$ time with the FFT algorithm (cf. Cooley and Tukey). It remains to compute the function values $f(k), k = 0, 1, \dots, M-1$. We can realize this by the discrete Fourier series inversion formula

$$f(k) \approx \frac{e^{\alpha k}}{\widehat{m}} \sum_{j=0}^{\widehat{m}-1} \cos \left(\frac{2\pi j l}{\widehat{m}} \right) F_k \left(\frac{j}{\widehat{m}} \right),$$

with $\widehat{m} = 1 + 2 \lfloor \delta \rfloor$ and $l = k \pmod{\widehat{m}}$. If $\delta \leq 1$ then

$$f(k) \approx (-1)^k e^{\alpha k} F_k \left(\frac{1}{2} \right).$$

Let us finally define the window function w . A good choice is the so called Gaussian window (cf. Mallat). The Gaussian window is defined by

$$w(t) = \exp \left(-\frac{1}{2} \left(\frac{t}{\sigma} \right)^2 \right),$$

with σ a given scale parameter. Given a prespecified tolerance ϵ , the scale parameter σ is chosen such that

$$\exp \left(-\frac{1}{2} \left(\frac{\delta}{\sigma} \right)^2 \right) < \epsilon. \quad (3.5.11)$$

We can compute the numbers A_j by

$$\begin{aligned} \frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} \exp \left(-\frac{1}{2} \left(\frac{t}{\sigma} \right)^2 \right) e^{\frac{2\pi i j t}{P}} dt &\approx \frac{1}{P} \int_{-\infty}^{\infty} \exp \left(-\frac{1}{2} \left(\frac{t}{\sigma} \right)^2 \right) e^{\frac{2\pi i j t}{P}} dt \\ &= \frac{\sigma \sqrt{2\pi}}{P} \exp \left(-\frac{1}{2} \left(\frac{2\pi j}{P} \right)^2 \sigma^2 \right). \end{aligned}$$

The truncation parameter L (in equation (3.5.10)) is chosen as

$$L = \arg \min \left\{ L : \frac{\sigma \sqrt{2\pi}}{P} \exp \left(-\frac{1}{2} (2\pi L)^2 \sigma^2 \right) < \epsilon \right\}, \quad (3.5.12)$$

3.5 Modifications for non-smooth functions

and δ is chosen as $\delta = 1$.

Algorithm:

input: \hat{f} , ϵ (desired precision), Δ and M , with M a power of two;

output: $f(\ell\Delta)$, $\ell = 0, 1, \dots, M - 1$;

parameters: $P = 8M$, $\delta_L = \delta_U = 1$, L (cf. formula (3.5.12)), σ (cf. formula (3.5.11)) and $n = 48$.

Step 1. For $k = -PL, \dots, PL - 1$ and $j = 1, 2, \dots, n$ compute the numbers

$$\hat{f}_{jk} = \text{Re} \left[\hat{f} \left(\frac{a + i\lambda_j + i\pi + \frac{2i\pi k}{P}}{\Delta} \right) \right],$$

for $k = -PL, \dots, PL - 1$

$$\hat{f}_k = \frac{1}{\Delta} \sum_{j=1}^n \beta_j \hat{f}_{jk},$$

and for $j = 0, 2, \dots, P - 1$ the numbers

$$\hat{g}_j = \sum_{m=-L}^{L-1} A_{mP+j} \hat{f}_{mP+j},$$

with

$$A_j = \frac{\sigma\sqrt{2\pi}}{P} \exp \left(-\frac{1}{2} \left(\frac{2\pi j}{P} \right)^2 \sigma^2 \right).$$

Step 2. Compute $f_k = \text{Re} \left(\sum_{j=0}^{P-1} e^{\frac{2\pi ijk}{P}} \hat{f}_j \right)$, for $k = 0, 1, \dots, P - 1$, with the FFT algorithm;

Step 3. Compute $f(k\Delta)$ with $f(k\Delta) = (-1)^k e^{ak} f_k$, for $k = 0, 1, \dots, M - 1$;

Remark 3.5.8. The composite inverse functions f_{w_k} are highly peaked, we recommend therefore to choose $n = 48$. With this choice we obtain results near machine precision.

Remark 3.5.9. For an arbitrary window function we can also compute the coefficients A_j efficiently with the fractional FFT, cf. . Since $|w(t)| < \epsilon$ for $[-\frac{P}{2}, -\delta_l] \cup [\delta_u, \frac{P}{2}]$, we can write this discrete Fourier transform as

$$A_j \approx \frac{1}{PN} \sum_{k=-P}^P w \left(\frac{k}{N} \right) e^{\frac{2\pi ijk}{P}} \approx \sum_{-\delta_l \leq \frac{k}{M} \leq \delta_u} w \left(\frac{k}{M} \right) e^{\frac{2\pi ijk}{P}},$$

with M a power of 2. This expression can be efficiently computed with the fractional FFT.

Remark 3.5.10. *If the inversion point is close to a singularity, we can compute $f(t)$ accurately by inverting the scaled Laplace transform*

$$f_{w_k}(t) = w_k(t) f(\Delta t)$$

To be more precise, suppose that f is smooth on the interval $[k - \delta, k + \delta]$ and that the function w_k has ϵ -support contained in $[k - \delta, k + \delta]$. If we chose $\Delta \leq \frac{\delta \epsilon}{\delta}$ then f_{w_k} is a smooth function. The price we have to pay for a small Δ is that the period of w_k is large, and that we need a larger number of terms for an accurate approximation of the Fourier series (3.5.8).

We ended this section with discussing the numerical results for eight test functions with singularities, that are used in the overview article of Martin and Davies (cf.). These results are presented in table 3.5. The inversions are done in 32 points and so $M = 32$. We have taken $P = 256 = 8M$ and for all the examples we have taken $a = 44/M_2$. Besides the choice of the previous parameters we have taken $n = 48$, $\sigma = 6\sqrt{2}$ and $L = 6$. The computations are done with double precision. We have taken $\Delta = 1/16$, $\Delta = 1$ and $\Delta = 10$. We see that for smooth functions the approximation error seems to be independent of Δ . The reason for this is that the approximation error is dominated with round-off error and the error caused by the highly peaked window function. We also test the method on two discontinuously functions. These results are presented in Table 3.6. For this example we have taken $\Delta = 1/16$. From all the examples we can conclude that the method is very accurate, robust and fast.

Table 3.5: Results for continuous non-analytical test functions

nr	$\Delta = \frac{1}{16}$	$\Delta = 1$	$\Delta = 10$
9	3e-14	2e-14	7e-15
10	4e-14	1e-14	4e-15
11	7e-15	2e-14	2e-14
12	5e-15	1e-15	6e-16
13	3e-16	9e-16	9e-17
14	1e-14	3e-15	8e-16
15	1e-14	2e-14	2e-14
16	1e-14	2e-14	2e-14

3.6 The Fourier Transform

We start with introducing the Fourier transform over the space $L^1(-\infty, \infty)$ of Lebesgue integrable functions. For a function $f \in L^1(-\infty, \infty)$ and pure imaginary s the Fourier

3.6 The Fourier Transform

Table 3.6: Results for non-continuous test functions

nr	function	Laplace Transform	MAE
17	$H(t - 1)$	$s^{-1} \exp(-s)$	1e-13
18	square wave	$(s(1 + \exp(-s)))^{-1}$	1e-13

integral

$$\widehat{f}(s) := \int_{-\infty}^{\infty} e^{-st} f(t) dt,$$

is properly defined. In fact

$$|\widehat{f}(s)| \leq \int_{-\infty}^{\infty} |f(t)| dt = \|f\|_1, \tag{3.6.1}$$

and \widehat{f} is a continuous function. If the support of f is contained in $[0, \infty)$ then we call \widehat{f} the Laplace transform of f . If $\widehat{f} \in L^1(-i\infty, i\infty)$, the space of complex valued Lebesgue integrable functions, satisfying $\frac{1}{2\pi i} \int_{-i\infty}^{i\infty} |x(s)| ds < \infty$, then the inverse Fourier integral is properly defined.

Theorem 3.6.1. *If $f \in L^1(-\infty, \infty)$ and $\widehat{f} \in L^1(-i\infty, i\infty)$ then*

$$f(t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{st} \widehat{f}(s) ds,$$

is properly defined and

$$f(t) \leq \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} |\widehat{f}(s)| ds = \|\widehat{f}\|_1.$$

This theorem can be found in any text about Fourier transforms, we have taken it from Mallat . It follows immediately that $\widehat{f} \in L^1(-i\infty, i\infty)$ iff f is a continuous function.

3.7 Proof of theorem 3.3.3

The proof of theorem 3.3.3: Recall that

$$\widehat{\phi}_n(s) = \frac{1}{s} \left(p_n \left(\frac{1}{s} \right) - (-1)^n e^{-s} p_n \left(-\frac{1}{s} \right) \right),$$

and

$$q_n^v(s) = \left(p_n(s) - (-1)^n e^{-2\pi i v} p_n(-s) \right),$$

and $\Psi \widehat{f}(s) = s^{-1} \widehat{f}(s^{-1})$. Since $\exp(-s^{-1}) = \exp(-i2\pi v)$ on $\{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}$, it follows that $q_n^v = \Psi \widehat{\phi}_n$, on $\{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}$. Hence

$$\langle q_m^v, q_n^v \rangle_{Q_v} = \langle \Psi \widehat{\phi}_m, \Psi \widehat{\phi}_n \rangle_{Q_v}.$$

Let

$$\Phi_{mn}(t) = \int_{-\infty}^{\infty} \phi_m(t-u) \phi_n(-u) du.$$

Since

$$\begin{aligned} \langle \Psi \widehat{\phi}_m, \Psi \widehat{\phi}_n \rangle_{Q_v} &= \sum_{k=-\infty}^{\infty} \widehat{\phi}_m(i2\pi(k+v)) \widehat{\phi}_n^*(i2\pi(k+v)) \\ &= \sum_{k=-\infty}^{\infty} \widehat{\Phi}_{mn}(i2\pi(k+v)) \end{aligned}$$

and the function Φ_{mn} is of bounded variation and in L^1 , we can apply the PSF (theorem 3.3.1) and obtain that

$$\langle q_m^v, q_n^v \rangle_{Q_v} = \langle \Psi \widehat{\phi}_m, \Psi \widehat{\phi}_n \rangle_{Q_v} = \langle \phi_m, \phi_n \rangle.$$

This proves that the $\{q_j\}$ are orthogonal.

Define

$$Q_v \Psi \widehat{f}(t) = \sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) e^{-i2\pi vt}.$$

Since

$$\sum_{k=-\infty}^{\infty} |\widehat{f}(i2\pi(k+v))|^2 < \infty,$$

$\text{range}(Q_v \Psi \widehat{f}) \in L^2([0,1])$. Since $\{\phi_k\}$ is a complete orthogonal set in $L^2([0,1])$, we obtain that

$$\sum_{k=-\infty}^{\infty} |\widehat{f}(i2\pi(k+v))|^2 = \sum_{k=0}^{\infty} \left| \langle Q_v \Psi \widehat{f}, \phi_k \rangle \right|^2.$$

Since

$$\langle e^{-2\pi i(v+k)\cdot}, \phi_k \rangle = \widehat{\phi}_k(2\pi i(v+k)),$$

3.8 The computation of the $\{\mu_k^v\}$ and $\{\alpha_k^v\}$

we obtain that

$$\sum_{k=0}^{\infty} \left| \langle Q_v \Psi \hat{f}, \phi_k \rangle \right|^2 = \sum_{k=0}^{\infty} \left| \langle \Psi \hat{f}, \Psi \hat{\phi}_k \rangle_{Q_v} \right|^2 = \sum_{k=0}^{\infty} \left| \langle \Psi \hat{f}, q_k^v \rangle_{Q_v} \right|^2,$$

in the last step we use that $q_n^v = \Psi \hat{\phi}_n$, on $\{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}$. Thus,

$$\sum_{k=0}^{\infty} \left| \langle \Psi \hat{f}, q_k^v \rangle_{Q_v} \right|^2 = \sum_{k=-\infty}^{\infty} |\hat{f}(i2\pi(k+v))|^2.$$

We obtain from Parseval's theorem (cf. Conway [thm 4.13]) $\{q_k^v\}$ is a complete orthogonal set in $L^2(Q_v)$ \square

3.8 The computation of the $\{\mu_k^v\}$ and $\{\alpha_k^v\}$

Let us begin with analyzing the matrix representation of the multiplication operator w.r.t. the basis $\{q_k^v\}$.

Theorem 3.8.1. *Let $M : L_{Q_v}^2 \rightarrow L_{Q_v}^2$ be the multiplication operator defined by $Mf(s) = sf(s)$. The matrix representation of M , \hat{M} , w.r.t. the basis $\{q_k^v\}$ is given by*

$$\begin{bmatrix} \frac{1}{2} \frac{1+\exp(-i2\pi v)}{1-\exp(-i2\pi v)} & -C_1 & & & \\ & C_1 & 0 & -C_2 & \\ & & C_2 & 0 & \ddots \\ & & & \ddots & \ddots \\ & & & & \ddots & \ddots \end{bmatrix},$$

where

$$C_n = \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}}.$$

Let $M_n : L_{Q_v^n}^2 \rightarrow L_{Q_v^n}^2$ be the multiplication operator defined by $M_n f(s) = sf(s)$. The matrix representation of M_n , \hat{M}_n , w.r.t. the basis $\{q_k^v; k = 0, 1, \dots, n-1\}$ is given by

$$\begin{bmatrix} \frac{1}{2} \frac{1+\exp(-i2\pi v)}{1-\exp(-i2\pi v)} & -C_1 & & & \\ & C_1 & 0 & -C_2 & \\ & & C_2 & \ddots & -C_{n-1} \\ & & & \ddots & \\ & & & & C_{n-1} & 0 \end{bmatrix}.$$

Proof. Since,

$$\langle Mf, g \rangle_{Q_v} = \langle sf, g \rangle_{Q_v} = \langle f, s^*g \rangle_{Q_v} = -\langle f, Mg \rangle_{Q_v},$$

M is a skew adjoint operator. Since q_n is orthogonal on π_n , the space of polynomials with degree less or equal $n - 1$, we obtain that

$$\langle Mq_j^v, q_k^v \rangle_{Q_v} = 0, \text{ for } j < k - 1.$$

Since M is skew adjoint we obtain that

$$\langle Mq_j^v, q_k^v \rangle_{Q_v} = 0, \text{ for } j > k + 1.$$

Let A_n be the coefficient of s^n in q_n^v and B_n be the coefficient of s^{n-1} in q_n^v . It can be easily verified that

$$A_n = \sqrt{2n+1} \frac{(2n)!}{n!} (-1)^n (1 - e^{-i2\pi v})$$

$$B_n = \sqrt{2n+1} \frac{(2n-1)!}{(n-1)!} (-1)^{n-1} (1 + e^{-i2\pi v}).$$

Comparing the coefficient of s^n in Mq_{n-1}^v and q_n^v respectively yields that

$$\langle Mq_{n-1}^v, q_n^v \rangle_{Q_v} = \frac{A_{n-1}}{A_n} = -\frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}}.$$

Comparing the coefficients of s^n in $Mq_n^v - \langle Mq_n^v, q_{n+1}^v \rangle_{Q_v} q_{n+1}^v$ and q_n^v respectively yields that

$$\langle Mq_n^v, q_n^v \rangle_{Q_v} = \frac{B_n}{A_n} - \frac{B_{n+1}}{A_{n+1}} = 0, n \geq 1,$$

and

$$\langle Mq_0^v, q_0^v \rangle_{Q_v} = -\frac{B_{n+1}}{A_{n+1}} = \frac{1}{2} \frac{1 + \exp(-i2\pi v)}{1 - \exp(-i2\pi v)}.$$

Expanding Mq_n^v in $\{q_k^v\}$ yields the desired result for the operator M . Since for all $f \in L^2_{Q_v}$,

$$\langle f, q_{n+1}^v \rangle_{Q_v} = 0,$$

we obtain the desired result for the operator M_n . □

Theorem 3.8.2. *The numbers $\{\mu_k^v\}$ are the eigenvalues of M_n . The Christoffel numbers $\{\alpha_k^v\}$ are given by $\frac{|\langle v_k, q_0^v \rangle|^2}{|1 - \exp(-i2\pi v)|^2}$, with the $\{v_k\}$ the normalized eigenfunctions ($\|v_k\| = 1$) of M_n .*

Proof. Let ρ be an arbitrary polynomial in π_{2n-1} . Since the matrix M_n and is skew self-adjoint, we obtain by the spectral theorem that

$$\langle \rho(M_n) \mathbf{1}, \mathbf{1} \rangle_{Q_v} = \sum a_k \rho(\lambda_k) \left| \langle \mathbf{1}, v_k \rangle_{Q_v} \right|^2 = \sum a_k \rho(\lambda_k) \frac{\left| \langle v_k, q_0^v \rangle_{Q_v} \right|^2}{|1 - \exp(-i2\pi v)|^2},$$

3.9 Error analysis

with $\{a_k\}$ the eigenvalues and $\{v_k\}$ the normalized eigenfunctions ($\|v_k\| = 1$) of M_n . On the other hand, since M_n and M are tri-diagonal we obtain that

$$\frac{\langle \rho(\widehat{M}_n) q_0^v, q_0^v \rangle_{Q_v^v}}{|1 - \exp(-i2\pi v)|^2} = \frac{\langle \rho(M) q_0^v, q_0^v \rangle_{Q_v}}{|1 - \exp(-i2\pi v)|^2} = \langle \rho \mathbf{1}, \mathbf{1} \rangle_{Q_v}.$$

Since the Gaussian quadrature rule is the unique n th order quadrature rule satisfying

$$\langle p, \mathbf{1} \rangle_n = \langle p, \mathbf{1} \rangle, \text{ for all } p \in \pi_{2n-1},$$

we obtain the result. \square

Remark 3.8.3. Since the matrix \widehat{M}_n is skew self-adjoint, we can efficiently compute the eigenvalues $\{\mu_k^v\}$ and eigenvectors $\{v_k\}$ with the QR algorithm (cf. Stoer and Bulirsch).

3.9 Error analysis

Let us start with giving an alternative formula for the Christoffel numbers $\{\alpha_j^v\}$.

Lemma 3.9.1. The Christoffel numbers $\{\alpha_j^v\}$ are given by

$$\alpha_j^v = \left(\frac{A_n^v}{A_{n-1}^v} \right)^* \frac{1}{q_{n-1}^v(\mu_j^v)} \frac{-1}{Dq_{n-1}^{-v}(-\mu_k^v)},$$

where A_n^v is the coefficient of s^n in q_n^v .

Proof. Let ℓ_j be the Lagrange polynomial defined by $\ell_j(\mu_k^v) = \delta_{kj}$, with $\delta_{kj} = 1$ if $k = j$ and $\delta_{kj} = 0$ if $k \neq j$. Then, $\langle q_{n-1}^v, \ell_j \rangle = \alpha_j^v q_{n-1}^v(\mu_j^v)$. On the other hand, $\langle \ell_j, q_{n-1}^v \rangle = \frac{B_j}{A_{n-1}^v}$, with A_{n-1}^v and B_j the coefficient of s^{n-1} in q_{n-1}^v and B_j the coefficient of s^{n-1} in ℓ_j . Since $B_j = \frac{A_n^v}{Dq_n^v(\mu_k^v)}$, we obtain the identity

$$\alpha_j^v = \left(\frac{A_n^v}{A_{n-1}^v} \right)^* \frac{1}{q_{n-1}^v(\mu_j^v)} \frac{-1}{Dq_{n-1}^{-v}(-\mu_k^v)}.$$

\square

Before we can give an error estimate for the quadrature rule, we need the following technical lemma:

Lemma 3.9.2. Let the integration operator $I : L^2([0, 1]) \rightarrow L^2([0, 1])$ be given by

$$If(x) = \int_0^x f(t) dt.$$

For $m = 0, 1, \dots, n - 1$ the estimate

$$\|s^m q_{n-1}^v\|_{Q_v^n} \leq \|I^m \phi_{n-1}\|,$$

is valid.

Proof. Since by the PSF (cf. theorem 3.3.1) $\|I^m \phi_{n-1}\| = \|s^m q_{n-1}^v\|_{Q_v^n}$, and $\|s^m q_{n-1}^v\|_{Q_v^n} = \|\widehat{M}_n^m e_n\|$ and $\|s^m q_{n-1}^v\|_{Q_v^n} = \|\widehat{M}_n^m e_n\|$, we only have to prove that $\|\widehat{M}_n^m e_n\| \leq \|\widehat{M}_n^m e_n\|$. Introduce the directed graph G with vertices $\{0, 1, \dots\}$ and edges $V^- = \{V_{j,j+1}\}$ and $V^+ = \{V_{j,j-1}\}$. Let G_n be the sub-graph of G with vertices $\{0, 1, \dots, n - 1\}$ and edges $V^- = \{V_{j,j+1}; j \leq n - 2\}$ and $V^+ = \{V_{j,j-1}; j \leq n - 1\}$. Let the weight of the path $\{j_0, \dots, j_m\}$ be given by $\prod_{p=0}^{m-1} M_{j_p j_{p+1}}$. Let S_k^m and \widehat{S}_k^m be the set of all paths from $n - 1$ to k of length m in G and G_n . The remainder of the proof consists of the following steps:

1. $\langle M_n^m e_n, e_k \rangle = \sum_{x \in S_k^m} \prod_{p=1}^m M_{x_p x_{p+1}}$ and $\langle M_n^m e_n, e_k \rangle = \sum_{x \in \widehat{S}_k^m} \prod_{p=1}^m M_{x_p x_{p+1}}$
2. The weight of each path in S_k^m has the same sign.
3. $|\langle M_n^m e_n, e_k \rangle| \leq |\langle M_n^m e_n, e_k \rangle|$ and $\|M_n^m e_n\| \leq \|\widehat{M}_n^m e_n\|$.

Proof of step 1): We will prove this result with induction. Clearly the result is true for $m = 1$. Since

$$S_{m+1}^k = \left\{ S_m^{k-1}, (k-1, k) \right\} \cup \left\{ S_m^{k+1}, (k+1, k) \right\}$$

we obtain that

$$\begin{aligned} \sum_{j \in S_{m+1}^k} \prod_{p=1}^{m+1} \widehat{M}_{j_p j_{p+1}} &= \widehat{M}_{k-1k} \sum_{j \in S_m^{k-1}} \prod_{p=1}^m \widehat{M}_{j_p j_{p+1}} + \widehat{M}_{k+1k} \sum_{j \in S_m^{k+1}} \prod_{p=1}^m \widehat{M}_{j_p j_{p+1}} \\ &= \widehat{M}_{k-1k} \langle \widehat{M}_n^m e_n, e_{k-1} \rangle + \widehat{M}_{k+1k} \langle \widehat{M}_n^m e_n, e_{k+1} \rangle \\ &= \langle \widehat{M}_n^{m+1} e_n, e_k \rangle, \end{aligned}$$

where the second equation follows from the induction hypotheses, and the last equality follows from the fact that the matrix \widehat{M} is tri-diagonal (cf. theorem 3.8.1) with only the first element on the diagonal non-zero and $m \leq n - 1$.

Proof of step 2): Since each path from $n - 1$ to k contains exactly $\frac{m-(n-1-k)}{2}$ edges of the set V^- and $\frac{m+(n-1-k)}{2}$ edges of the set V^+ , the weight of each path in S_k^m has the same sign.

3.9 Error analysis

Proof of step 3): Since \widehat{S}_k^m is a subset of S_k^m and each path in S_k^m has the same sign we obtain that

$$\left| \left\langle \widehat{M}_n^m e_n, e_k \right\rangle \right| = \sum_{x \in \widehat{S}_k^m} \left| \prod_{p=1}^m \widehat{M}_{x_p, x_{p+1}} \right| \leq \sum_{x \in S_k^m} \left| \prod_{p=1}^m \widehat{M}_{x_p, x_{p+1}} \right| = \left| \left\langle \widehat{M}_n^m e_n, e_k \right\rangle \right|.$$

The second inequality follows from

$$\left\| \widehat{M}_n^m e_n \right\|^2 = \sum_{k=0}^{n-1} \left| \left\langle \widehat{M}_n^m e_n, e_k \right\rangle \right|^2 \leq \sum_{k=0}^{n-1} \left| \left\langle \widehat{M}_n^m e_n, e_k \right\rangle \right|^2 = \left\| \widehat{M}_n^m e_n \right\|^2.$$

□

We can now give an error estimate for the quadrature rule.

Theorem 3.9.3. *The error of the quadrature formula*

$$E_v \widehat{f} = \left\langle \Psi \widehat{f}, \Psi \mathbf{1} \right\rangle_{Q_v} - \left\langle \Psi \widehat{f}, \Psi \mathbf{1} \right\rangle_{Q_v^0},$$

is given by

$$\frac{1}{2} \frac{(-1)^n}{\sqrt{4n^2 - 1}} \sum_{j=1}^{\infty} j e^{-i2\pi j v} W_{j-1} f^{(2)} + \xi \widehat{f},$$

where $f^{(k)}$ denotes the k th order derivative of f , and

$$W_j h = \int_0^1 \int_0^1 h(j+t+y) \phi_n(t) \phi_{n-1}(y) dy dt,$$

with ϕ_n the n th Legendre polynomials given by

$$\phi_n(t) = \frac{\sqrt{2n+1}}{n!} D^n (t^n (1-t)^n),$$

where D denotes the differential operator. Furthermore, the remainder $\xi \widehat{f}$ is bounded by

$$\left| \xi \widehat{f} \right| \leq \frac{1}{4} \min_{0 \leq m \leq (n-1)} \frac{\|I^m \phi_{n-1}\|^2}{4n^2 - 1} \sum_{j=0}^{\infty} (j+1) \left\| R_j f^{(4+2m)} \right\|_1,$$

with

$$R_j h(x) = \begin{cases} 0 & , x < 0 \\ \int_0^1 \int_0^1 h(j+t+y+x) \phi_n(t) \phi_n(y) dy dt & , x \geq 0 \end{cases}.$$

Moreover,

$$W_j h = \sqrt{4n^2 - 1} \frac{(n-1)!}{(2n-1)!} \frac{n!}{(2n+1)!} h^{(2n-1)}(\alpha_j), \alpha_j \in (j, j+2)$$

$$\|R_j h\|_1 \leq (2n+1) \left(\frac{n!}{(2n+1)!} \right)^2 \|h_j^{(2n)}\|_1$$

$$\|I^{n-1} \phi_{n-1}\|^2 = \frac{(2n-1)}{(n-1)!^2} \left(\frac{((2n-2)!)^2}{(4n-3)!} \right),$$

where

$$\|h_j^{(2n)}\|_1 = \int_j^\infty |h_j^{(2n)}(t)| dt.$$

Thus, for smooth functions,

$$E_v \widehat{f} \approx \frac{1}{2} (-1)^n \frac{(n-1)!n!}{(2n-1)!} \sum_{j=1}^{\infty} j e^{-i2\pi jv} \frac{f^{(2n+1)}(j-1+\alpha)}{(2n+1)!},$$

with α some number in $(0,2)$.

Proof. The proof consists of the following steps:

1. The functional E_v can be expanded as

$$E_v \widehat{f} = -\frac{1}{2} \frac{1}{\sqrt{4n^2-1}} \sum_{j=0}^{\infty} e^{-i2\pi jv} \left\langle q_{n-1}^v \Psi \widehat{V_j Df}, \Psi \mathbf{1} \right\rangle_{Q_v^n}, \quad (3.9.1)$$

where V_j is given by

$$V_j h(y) = \begin{cases} 0 & , y < 0 \\ \int_0^1 h(j+t+y) \phi_n(t) dt & , y \geq 0. \end{cases}$$

2. By the definition of the functional E_v

$$E_v \left(q_{n-1}^v (s^{-1}) \widehat{V_j Df} \right) = \left\langle q_{n-1}^v \Psi \widehat{V_j f}, \Psi \mathbf{1} \right\rangle_{Q_v} - \left\langle q_{n-1}^v \Psi \widehat{V_j f}, \Psi \mathbf{1} \right\rangle_{Q_v^n} \quad (3.9.2)$$

The term $\left\langle q_{n-1}^v \Psi \widehat{V_j f}, \Psi \mathbf{1} \right\rangle_{Q_v}$ is equal to

$$(-1)^{n-1} \sum_{k=0}^{\infty} e^{-i2\pi(k+1)v} \int_0^1 \int_0^1 f^{(2)}(k+j+t+u) \phi_n(t) \phi_{n-1}(y) dy dt. \quad (3.9.3)$$

3. The term $\left| \left\langle q_{n-1}^v q_{n-1}^v \Psi \widehat{R_{j+k} D^2 f}, \Psi \mathbf{1} \right\rangle_{Q_v^n} \right|$ is bounded by

$$\|I^m \phi_{n-1}\| \left\| R_{j+k} f^{4+2m} \right\|_1.$$

4. We can estimate $W_j h$ and $\|R_j h\|_1$ by

$$W_j h = \sqrt{(2n)^2 - 1} \frac{(n-1)!}{(2n-1)!} \frac{n!}{(2n+1)!} h^{(2n-1)}(\alpha), \alpha \in (j, j+2)$$

$$\|R_j h\|_1 \leq (2n+1) \left(\frac{n!}{(2n+1)!} \right)^2 \|h_j^{(2n)}\|_1,$$

3.9 Error analysis

Proof of step 1: Let $\widehat{g}(s) = s\widehat{f}(s)$, then

$$\langle \Psi\widehat{f}, \Psi\mathbf{1} \rangle_{Q_v} = -\langle \Psi\widehat{g}, \mathbf{1} \rangle_{Q_v} \text{ and } \langle \Psi\widehat{f}, \Psi\mathbf{1} \rangle_{Q_v^n} = -\langle \Psi\widehat{g}, \mathbf{1} \rangle_{Q_v^n}.$$

Let P_{n-1} be the interpolation polynomial of $\Psi\widehat{g}$ in $\{\mu_k^v\}$, where the $\{\mu_k^v\}$ are the zeros of q_n^v . Since the quadrature rule is exact for polynomials of degree $n-1$, we obtain that

$$\langle P_{n-1}, \mathbf{1} \rangle_{Q_v} = \langle P_{n-1}, \mathbf{1} \rangle_{Q_v^n} = \langle \Psi\widehat{g}, \mathbf{1} \rangle_{Q_v^n}.$$

Hence,

$$E_v\widehat{f} = \langle \Psi\widehat{g}, \mathbf{1} \rangle_{Q_v^n} - \langle \Psi\widehat{g}, \mathbf{1} \rangle_{Q_v} = -\langle \Psi\widehat{h}, \mathbf{1} \rangle_{Q_v},$$

with

$$\widehat{h}(s) = \widehat{g}(s) - \Psi P_{n-1}(s).$$

Since for $s \in \{2\pi i(k+v)\}$ (cf. equation 3.2.2 in appendix 3.2.2)

$$\frac{-s\widehat{\phi}_n(-s)}{q_n^{-v}(-s-1)} = 1,$$

we obtain that

$$-\langle \Psi\widehat{h}, \mathbf{1} \rangle_{Q_v} = \langle \Psi\widehat{k}, \Psi\mathbf{1} \rangle_{Q_v}, \quad (3.9.4)$$

with

$$\widehat{k}(s) = s\widehat{h}(s)\widehat{\phi}_n(-s)\widehat{W}(s),$$

where

$$\widehat{W}(s) = -\frac{1}{s} \frac{1}{q_n^{-v}(-s-1)}.$$

The following observations show that \widehat{k} is a proper Laplace transform, satisfying the conditions for the PSF (cf. theorem 3.3.1):

1. Since P_{n-1} interpolates $\Psi\widehat{g}$ in $\{\mu_k^v\}$, the function \widehat{k} is holomorphic in $\{\frac{1}{\mu_k^v}\}$.
2. Since $\widehat{\phi}_n(-s) = O(s^n)$ and $\frac{1}{q_n^{-v}(-s-1)} = O(s^n)$ in a neighborhood of 0, \widehat{k} is holomorphic in a neighborhood of 0.
3. $\frac{\widehat{k}(s)}{\widehat{f}(s)} \rightarrow \frac{1-(-1)^n e^s}{1-(-1)^n e^{i2\pi v}}$, as $s \rightarrow \pm i\infty$.
4. \widehat{k} is holomorphic in the right half plane $\Pi = \{s; \text{Re}(s) > 0\}$.

Thus,

$$\langle \Psi\widehat{k}, \Psi\mathbf{1} \rangle_{Q_v} = \sum_{j=0}^{\infty} e^{-i2j\pi v} k(j),$$

with

$$k(x) = \frac{1}{2\pi i} \int_{-a-i\infty}^{-a+i\infty} e^{sx} \widehat{k}(s) ds, a > 0.$$

Since \widehat{W} is holomorphic in the plane except in the points $\{\frac{1}{\mu_k^v}\}$, and converges uniformly to 0 as $|s| \rightarrow \infty$, we obtain that

$$W(x) = \begin{cases} 0 & , \text{ for } x \geq 0 \\ \sum_{k=1}^n \text{res} \left(e^{sx} \widehat{W}(s); s = \frac{1}{\mu_k^v} \right) & , \text{ for } x < 0. \end{cases}$$

Hence, we obtain for $x < 0$, that

$$\begin{aligned} W(x) &= - \sum_{j=1}^n \frac{e^{\frac{x}{\mu_j^v}}}{\mu_j^v} \frac{1}{Dq_n^{-v} \left(-\mu_j^v \right)} \\ &= - \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}} \sum_{j=1}^n \alpha_j^v \frac{e^{\frac{x}{\mu_j^v}}}{\mu_j^v} q_{n-1}^v \left(\mu_j^v \right), \end{aligned} \tag{3.9.5}$$

where the last equality follows from lemma 3.9.1. Since the support of W is $(-\infty, 0]$, we obtain that

$$\begin{aligned} k(x) &= \int_0^\infty \int_0^1 h^{(1)}(x+t+y) \phi_n(t) dt W(-y) dy \\ &= \int_0^\infty \int_0^1 f^{(2)}(x+t+y) \phi_n(t) dt W(-y) dy, \end{aligned}$$

in the last equality we use that ϕ_n is orthogonal on each polynomial of degree less than n . This yields by equation (3.9.5) and the definition of the inner product $\langle \cdot, \cdot \rangle_{Q_v^0}$ (cf. definition 3.3.4) that

$$k(j) = - \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}} \left\langle q_{n-1}^v \Psi \widehat{V_j f^{(2)}}, \Psi \mathbf{1} \right\rangle_{Q_v^0},$$

with V_j given by

$$V_j h(y) = \begin{cases} 0 & , y < 0 \\ \int_0^1 h(j+t+y) \phi_n(t) dt & , y \geq 0. \end{cases}$$

Proof of step 2): Since for $s \in \{2\pi i(k+v)\}$,

$$q_{n-1}^v(s^{-1}) = (-1)^{n-1} s e^{-i2\pi v} \widehat{\phi}_{n-1}(-s),$$

the first term of formula (3.9.2) is equal to

$$(-1)^{n-1} e^{-i2\pi v} \left\langle \Psi \left(\widehat{\phi}_{n-1}(-s) \widehat{V_j f^{(2)}} \right), \Psi \mathbf{1} \right\rangle_{Q_v}.$$

3.9 Error analysis

We obtain from the PSF (cf. theorem 3.3.1) that this is equal to

$$(-1)^{n-1} \sum_{k=0}^{\infty} e^{-i2\pi(k+1)v} \int_0^1 \int_0^1 f^{(2)}(k+j+t+u) \phi_n(t) \phi_{n-1}(y) dydt. \quad (3.9.6)$$

We obtain from formula (3.9.1) that the second term of formula (3.9.2) is equal to

$$\frac{1}{2} \frac{1}{\sqrt{4n^2-1}} \sum_{k=0}^{\infty} e^{-i2\pi kv} \left\langle q_{n-1}^v q_{n-1}^v \widehat{\Psi R_{j+k} f^{(2)}, \Psi \mathbf{1}} \right\rangle_{Q_v^n},$$

with

$$R_j h(x) = \begin{cases} 0 & , x < 0 \\ \int_0^1 \int_0^1 h(j+t+y+x) \phi_n(t) \phi_n(y) dydt & , x \geq 0. \end{cases}$$

Proof of step 3): Since ϕ_n is orthogonal on π_{n-1} , we obtain that $R_j p_{2n-1} = 0$, for $p_{2n-1} \in \pi_{2n-1}$. Hence,

$$\left\langle q_{n-1}^v q_{n-1}^v \widehat{\Psi R_{j+k} D^2 f, \Psi \mathbf{1}} \right\rangle_{Q_v^n} = (-1)^{n-1} e^{-i2\pi v} \left\langle R_{j+k} \widehat{f^{(4+2m)}} s^{2m} q_{n-1}^v, q_{n-1}^v \right\rangle_{Q_v^n},$$

for $0 \leq m \leq (n-1)$. Hence,

$$\begin{aligned} \left| \left\langle q_{n-1}^v q_{n-1}^v \widehat{\Psi R_{j+k} f^{(2)}, \Psi \mathbf{1}} \right\rangle_{Q_v^n} \right| &\leq \sup_s \left| R_{j+k} \widehat{f^{(4+2m)}}(s) \right| \left| \left\langle s^{2m} q_{n-1}^v, q_{n-1}^v \right\rangle_{Q_v^n} \right| \\ &\leq \left\| R_{j+k} f^{(4+2m)} \right\|_1 \left| \left\langle s^{2m} q_{n-1}^v, q_{n-1}^v \right\rangle_{Q_v^n} \right|, \end{aligned} \quad (3.9.7)$$

the last inequality follows from inequality (3.6.1). We obtain from lemma 3.9.2 that

$$\left| \left\langle s^{2m} q_{n-1}^v, q_{n-1}^v \right\rangle_{Q_v^n} \right| = \|I^m \phi_{n-1}\|^2.$$

Proof of step 4): Recall that

$$\phi_n(t) = \frac{\sqrt{2n+1}}{n!} D^n (t^n (1-t)^n),$$

Integration by part and using that

$$D^{n-k} (t^n (1-t)^n) \Big|_{t=0}^1 = 0, k = 1, \dots, n,$$

yields that

$$\begin{aligned} W_j h &= \int_0^1 \int_0^1 h(j+t+y) \phi_n(t) \phi_{n-1}(y) dydt \\ &= \sqrt{(2n)^2-1} \frac{(n-1)!}{(2n-1)!} \frac{n!}{(2n+1)!} \int_0^1 \int_0^1 h^{(2n-1)}(j+t+y) B_n(t) B_{n-1}(y) dydt, \end{aligned}$$

where B_n is the density of the Beta distribution, that is

$$B_n(t) = \frac{(2n+1)!}{n!n!} t^n (1-t)^n$$

Since the Beta distribution is a probability distribution we obtain by the mean value theorem that

$$W_j h = \sqrt{(2n)^2 - 1} \frac{(n-1)!}{(2n-1)!} \frac{n!}{(2n+1)!} D^{2n-1} h(\alpha_j), \alpha_j \in (j, j+2).$$

Similarly,

$$\begin{aligned} R_j h(x) &= \int_0^1 \int_0^1 h(j+t+y) \phi_n(t) \phi_n(y) dy dt \\ &= (2n+1) \left(\frac{n!}{(2n+1)!} \right)^2 \int_0^1 \int_0^1 D^{2n} h(j+x+t+y) B_n(t) B_n(y) dy dt. \end{aligned}$$

Hence,

$$\|R_j h\|_1 \leq (2n+1) \left(\frac{n!}{(2n+1)!} \right)^2 \|D^{2n} h_j\|_1.$$

With a similar argument we obtain that

$$\|I^{n-1} \phi_{n-1}\|^2 = \frac{(2n-1)}{(n-1)!^2} \left(\frac{((2n-2)!)^2}{(4n-3)!} \right).$$

□

Remark 3.9.4. It follows from theorem 3.9.3 that the approximation error does not depend on the discontinuity of f in $t = 0$.

3.10 On the inversion of z-transforms

Theorem 3.10.1. Let

$$F(v) = \sum_{k=-\infty}^{\infty} f_k \exp(-2\pi i k v).$$

The following inversion formula holds:

$$\frac{1}{N} \sum_{k=0}^{N-1} F\left(\frac{k}{N}\right) \exp\left(2\pi i j \frac{k}{N}\right) = \sum_{m=-\infty}^{\infty} f_{mN+j}, \quad (3.10.1)$$

for $j = 0, 1, \dots, N-1$.

3.10 On the inversion of z-transforms

Proof. Since for integral k and m , $\exp(-i2\pi km) = 1$, we obtain that

$$\begin{aligned} F\left(\frac{k}{N}\right) &= \sum_{j=0}^{N-1} \sum_{m=-\infty}^{\infty} f_{mN+j} \exp\left(-2\pi ij \frac{k}{N}\right) \exp(-2\pi ikm) \\ &= \sum_{j=0}^{N-1} \exp\left(-2\pi ij \frac{k}{N}\right) \sum_{m=-\infty}^{\infty} f_{mN+j}. \end{aligned}$$

Hence,

$$\frac{1}{N} \sum_{k=0}^{N-1} F\left(\frac{k}{N}\right) \exp\left(2\pi ij \frac{k}{N}\right) = \frac{1}{N} \sum_{k=0}^{N-1} F_N\left(\frac{k}{N}\right) \exp\left(2\pi ij \frac{k}{N}\right),$$

with

$$F_N(v) = \sum_{j=0}^{N-1} \exp(-2\pi ijv) \sum_{m=-\infty}^{\infty} f_{mN+j}.$$

Formula (3.10.1) follows from

$$\frac{1}{N} \sum_{k=0}^{N-1} F_N\left(\frac{k}{N}\right) \exp\left(2\pi ij \frac{k}{N}\right) = \sum_{m=-\infty}^{\infty} f_{mN+j}.$$

□

Remark 3.10.2. *This result is sometimes called the discrete Poisson summation formula (cf. Abate and Whitt). In their paper Abate and Whitt present an efficient inversion algorithm for the inversion of z-transforms (also called generating functions) of discrete probability distributions.*

We proceed similarly as the paper of Abate and Whitt. It follows immediately from theorem 3.10.1 that if the discretization error

$$\tilde{\zeta}_j := \sum_{m \neq 0, m \in \mathbb{Z}} f_{mN+j},$$

is small then

$$f_j \approx \frac{1}{N} \sum_{k=0}^{N-1} F\left(\frac{k}{N}\right) \exp\left(2\pi ij \frac{k}{N}\right).$$

In general it is hard to insure that the discretization error $\tilde{\zeta}_d$ is small. However, for a special class of z-transforms we can easily control the discretization error $\tilde{\zeta}_d$. A z-transform \hat{f} belongs to the Hilbert space $H^2(\mathbb{U})$ if $\hat{f} \in L^2(\mathbb{T})$ and $f_k = 0$, for $k < 0$ (cf. chapter 17 in Rudin). Since $f_k = 0$, for $k < 0$, we obtain that the inverse sequence of the z-transform $\hat{f}^r(z) = \hat{f}(rz)$ is given by $\{r^k f_k; k \in \mathbb{N}\}$. Hence, $r^{-k} f_k^r = f_k$ and

$$\begin{aligned} \tilde{\zeta}_j^r &: = r^{-j} \sum_{m=1}^{\infty} f_{mN+j}^r \\ &= \sum_{m=1}^{\infty} r^{mN} f_{mN+j}^r, \end{aligned}$$

Chapter 3. Numerical Transform Inversion Using Gaussian Quadrature is of order $O(r^N)$. Hence, we can make ξ_j^r arbitrarily small by choosing r small enough. The conclusion is that if $\hat{f} \in H^2(\mathbb{U})$ then we can efficiently compute the numbers f_k with arbitrary precision, with the following algorithm.

Step 1. Set $\hat{f}^r(z_k) = \hat{f}(rz_k)$, $k = 0, 1, \dots, m_2$.

Step 2. Compute $\hat{f}^r(z_k) \rightarrow f_r$ with the m_2 points FFT algorithm and set $f_k = r^{-k} f_r$, $k = 0, 1, \dots, m$.

Remark 3.10.3. On one hand we want to choose r as small as possible, since then the discretization error is as small as possible. On the other hand we multiply f_k^r with the factor r^{-k} , hence a small r makes the algorithm numerically unstable. Suppose that we want to compute the values $\{f_k; k = 0, 1, \dots, m - 1\}$ with a precision of ϵ . We can control both the discretization error and the numerical stability by using an $m_2 = 2^p m$ (with p a positive integral number) discrete Fourier transform (FFT) and choose $r = e^{\frac{1}{m_2}}$. The discretization error is then of the magnitude of ϵ , and the multiplication factor is bounded by $e^{\frac{1}{2^p}}$. For double precision we recommend the parameter values $p = 3$ and $r = \exp(-\frac{44}{2^p m})$.

3.11 Quadrature nodes and weights

In table 3.7 a number of quadrature nodes and weight are tabled.

3.11 Quadrature nodes and weights

λ_j	β_j	λ_j	β_j
n=8		n=48	
0.0	1.0	0.0	1.0
6.28318530717958	1.00000000000004	6.28318530717957	1.0
12.5663706962589	1.00000015116847	12.5663706143592	1.0
18.8502914166954	1.00081841700481	18.8495559215388	1.0
25.2872172156717	1.09580332705189	25.1327412287183	1.0
34.296971663526	2.00687652338724	31.4159265358979	1.0
56.1725527716607	5.94277512934943	37.6991118430775	1.0
170.533131190126	54.9537264520382	43.9822971502571	1.0
n=16		50.2654824574367	1.0
0.0	1.0	56.5486677646182	1.00000000000234
6.28318530717958	1.0	62.8318530747628	1.00000000319553
12.5663706143592	1.0	69.1150398188909	1.00000128757818
18.8495559215388	1.0	75.3984537709689	1.00016604436873
25.1327412287184	1.0	81.6938697567735	1.00682731991922
31.4159265359035	1.00000000000895	88.1889420301504	1.08409730759702
37.6991118820067	1.00000004815464	95.7546784637379	1.3631917322868
43.9823334683971	1.00003440685547	105.767553649199	1.85773538601497
50.2716029125234	1.00420404867308	119.58751936774	2.59022367414073
56.7584358919044	1.09319461846681	139.158762677521	3.73141804564276
64.7269529917882	1.51528642466058	168.156165377339	5.69232680539143
76.7783110023797	2.4132076646714	214.521886792255	9.54600616545647
96.7780294888711	4.16688127092229	298.972429369901	18.8912132110256
133.997553190014	8.3777001312961	497.542914576338	52.7884611477405
222.527562038705	23.6054680083019	1494.71066227687	476.448331869636
669.650134867713	213.824023377988		

Table 3.7: Quadrature nodes and weights.

Only the numbers λ_j and β_j for $j=1,2,\dots,\frac{n}{2}$ are tabled. The numbers λ_{n+1-j} are given by $-\lambda_j - 2\pi$. The numbers β_{n+1-j} coincide with β_j .

Chapter 4

Laplace Transform Inversion on the entire line

4.1 Introduction

The need for advanced numerical techniques for computational probability is rapidly increasing. The literature on numerical Laplace transform inversion algorithms is extensive. There are four widely use methods, namely: 1) The Fourier series method which is based on the Poisson Summation formula (cf. Dubner and Abate (1968), Abate and Whitt (1992, 1995), Choudhury, Lucantoni and Whitt (1997), O’Cinneide (1997) and Sakurai (2004)), 2) Gaver-Stehfest algorithm which is based on combinations of Gaver functionals (cf. Gaver (1966) and Stehfest (1970)), 3) Weeks method which is based on bilinear transformations and Laguerre expansions (cf. Weeks (1966) and Abate, Choudhury and Whitt (1996, 1998)) and 4) the Talbot method which is based on deforming the contour in the Bromwich inversion integral (cf. Talbot (1979) and Murli and Rizzardi (1990)). A common feature of all these methods is that the inversion is performed in a single point at a time. Den Iseger (2006a) has also offered a fast and efficient numerical transform (Laplace transform) inversion, based on a Gaussian quadrature formula, which can perform the inversion on a uniform grid at a time (that is, performs the inversion in several points at a time). This algorithm can be employed for solving many of the problems arising in computational probability with near machine precision. However flexible, fast and precise, this method has some limitations (i.e. the limitations of a uniform grid vs an arbitrary grid). It cannot tackle important problems as iterative algorithms (for instance recursion, widely employed for dynamic programming style problems), Lévy processes based modeling, or problems where the inverse transform as well as the transform itself needs to be computed (a good example is the waiting time distribution in a $G/G/1$ queue and a barrier option with underlying modeled by a general Lévy process).

On the contrary, the numerical inversion method we are about to introduce is a piecewise inversion: the algorithm computes the coefficients of the Legendre expansions of the desired function on all intervals $[k\Delta, (k+1)\Delta]$ (an arbitrary grid). This piecewise inversion algorithm maintains the capabilities of the pointwise inversion algorithm. The two key improvements, which enable the new numerical technique to tackle much wider spectrum of problems than its forebear, can be summarized as follows. First, the original function f (obtained after the numerical inversion) can be evaluated in arbitrary points x_k (instead of only in points $k\Delta$, $k = 0, 1, \dots$, as previously). Secondly, the new method works both ways: if some numerical values of the original function f are given, then the algorithm generates its Laplace transform $\mathcal{L}f$ (thus no closed-form analytical expression for f needed!); and the other way around, given some numerical values of the Laplace transform, the original function f can be obtained in arbitrary points.

The paper is structured as follows. The first section presents an overview of the most important theoretical foundations of the method. Section 4.3 is the core of the paper, it lines up the method and presents the resulting Laplace transform and Laplace transform inversion algorithms. Section 4.5 discusses some important implementation issues, then in section 4.6 a detailed error analysis is presented. Section 4.7 discusses the necessary modifications for an efficient use of the transformation algorithms with non-smooth functions. Finally, the precision and robustness of the algorithms is illustrated by a large number of numerical examples, including all the Laplace transforms used in the overview article of Martin and Davies (1979) and also revisited by den Iseger (2006a).

The results of this chapter are described in the working paper den Iseger (2006b)*.

4.2 Preliminaries: tools for the algorithm

As discussed in the introduction, the algorithm computes the coefficients of the piecewise Legendre polynomial expansion of the original function if specific Laplace transform function values are known, and conversely, obtains Laplace transform function values if the expansion coefficients of the original function are known. To this end, let us introduce these tools needed for the construction of the algorithm.

* In some publication this paper is referred as P. den Iseger, Numerical evaluation of the Laplace and inverse Laplace transform

4.2 Preliminaries: tools for the algorithm

4.2.1 The Fourier transform and the Poisson Summation Formula

Let f be a complex valued Lebesgue integrable function f , with $f \in L^1(-\infty, \infty)$. The Fourier Transform of f is defined by the integral

$$\widehat{f}(s) := \int_{-\infty}^{\infty} e^{-st} f(t) dt. \quad (4.2.1)$$

An operator $F : X \rightarrow Y$ is called unitary if the operator is invertible and has the property that $\langle f, g \rangle_X = \langle Ff, Fg \rangle_Y$, for all f and g in X . The Fourier transform and inverse Fourier transform can be extended to a unitary operator on the the spaces $L^2((-\infty, \infty))$ and $L^2((-i\infty, i\infty))$, known as Plancherel's theorem (see Rudin (1987) theorem 9.13).

Definition 4.2.1. Let $F : L^2((-\infty, \infty)) \rightarrow L^2((-i\infty, i\infty))$ be the unitary operator given by $Ff = \widehat{f}$. Let $F^{-1} : L^2((-i\infty, i\infty)) \rightarrow L^2((-\infty, \infty))$ be the unitary operator given by $F^{-1}\widehat{f} = f$. The operator F is called the Fourier transform and the operator F^{-1} the inverse Fourier transform.

The analytical extension of the Fourier transform to a subplain of the complex plain is often called the Laplace transform. In this paper we will use both terms for the Fourier transform.

The basis for many Laplace and Fourier inversion algorithms is the Poisson summation formula (PSF), see for instance Mallat (2001).

Theorem 4.2.2. Let $f \in L^2((-\infty, \infty))$ and v a given number in $[0, 1)$ then

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) e^{i2\pi(k+v)t} = \sum_{j=-\infty}^{\infty} e^{-i2j\pi v} f(j+t) \quad (4.2.2)$$

with respect to the $L^2[0, 1]$ norm (here t is the argument for the $L^2[0, 1]$ norm). Moreover, let $g \in L^2((-\infty, \infty))$ then

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) \widehat{g}^*(i2\pi(k+v)) = \sum_{j=-\infty}^{\infty} e^{-i2j\pi v} \langle f, g(\cdot - j) \rangle \quad (4.2.3)$$

with respect to the $L^2[0, 1]$ norm (here t is the argument for the $L^2[0, 1]$ norm). We denote with \widehat{g}^* the complex conjugate of \widehat{g} .

The use of the PSF for transform inversion started with the pioneer work of Abate and Dubner (1968). Their method is based on formula (4.2.2) and reads as follows:

$$f(t) \approx \sum_{j=-\infty}^{\infty} f(j+t) = \sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi k) e^{i2\pi kt} \approx \sum_{k=-N}^N \widehat{f}(i2\pi k) e^{i2\pi kt} \quad (4.2.4)$$

In most applications the LHS of formula (4.2.2) is a slowly converging series. For instance if the function f has support $[0, \infty)$ and $f(0)$ doesn't vanish then the sum converges like $O(s^{-1})$ (this can be easily verified by integration by parts). Abate and Whitt (1992, 1995) accelerate the convergence of the last approximation step by using Euler acceleration. For function with support in $[0, \infty)$ the approximation error of the first step of formula (4.2.4) can be reduced by the use of a so called damping factor. The idea of a dampings factor is to apply the PSF on the function $\hat{f}_a(s) = \hat{f}(a + s)$. The function $f(t)$ is then given by $e^{-at}f(t)$ and the we obtain

$$e^{at} \sum_{k=-\infty}^{\infty} \hat{f}_a(i2\pi k) e^{i2\pi kt} = f(t) + R_a(t)$$

with

$$R_a(t) = \sum_{j=1}^{\infty} e^{-aj} f(j+t)$$

By chosing a large enough we can can make the approximation error R_a as small as desired. The price we have to pay for a large value of a is numerical instability due the multiplication of the factor e^{at} .

In den Iseger (2006a) a Gaussian quadrature formula for the infite sum of the RHS of formula (4.2.2) is presented. In that paper it is shown that the approximation error of the Gaussian quadrature rule is almost of the order of the machine precision. In this paper we extend the quadrature framework for formula (4.2.3). This enables us to compute efficiently piece-wise Legendre polynomial expansions for the function f given some functionvalues of \hat{f} . We also show how to evaluate efficiently the Laplace transform of the piece-wise Legendre expansion, and how to compute efficiently the piece-wise Legendre expansion for a function f . This yields an efficient method which evaluate the function f from function values of the Laplace transform \hat{f} , and evaluate the function the Laplace transform \hat{f} from function values of the function f . The method can efficiently be implemented using matrix vector multiplications and (inverse) Fast Fourier Transforms (IFFT,FFT) algorithms.In the next section we discuss the piece-wise Legendre polynomial expansion of a given function f .

4.2.2 Legendre polynomial expansions

The Legendre polynomials $\{\phi_k : k \in \mathbb{N}_0\}$ are a complete orthogonal polynomial set in $L^2([0, 1])$. Therefore every function in $L^2([0, 1])$ can be approximated with an expansion in these polynomials. The algorithm computes the coefficients of this expansion, using as input a number of function values of the Laplace transform. The Legendre

4.2 Preliminaries: tools for the algorithm

polynomials $\{\phi_k : k \in \mathbb{N}_0\}$, $\phi_k : [0, 1] \rightarrow \mathbb{R}$ are given by

$$\phi_k(t) = \frac{\sqrt{2k+1}}{k!} \frac{\partial^k}{(\partial t)^k} (t^k(1-t)^k). \quad (4.2.5)$$

Let $P_n f$ denote the projection of f on the space π_n of polynomials with degree less than or equal to n . The projection $P_n f$ is given by

$$\sum_{k=0}^n \langle f, \phi_k \rangle \phi_k.$$

In order to extend the method to functions in $L^2(-\infty, +\infty)$ and $L^2[0, +\infty)$ we need to consider the shifted version of Legendre polynomials. Since the shifted Legendre polynomials $\{\phi_k(\cdot - j) : k \in \mathbb{N}_0, j \in \mathbb{Z}\}$ are a complete orthogonal set in $L^2((-\infty, \infty))$, any function $f \in L^2((-\infty, \infty))$ can be expanded into

$$f = \sum_{k=0}^{\infty} \sum_{j=-\infty}^{\infty} \langle f, \phi_k(\cdot - t_0 - j) \rangle \phi_k(\cdot - t_0 - j),$$

with t_0 a shift parameter. Moreover, since the support of these polynomials is known to be $[0, 1]$, we obtain that

$$f(t) = \sum_{k=0}^{\infty} \langle f, \phi_k(\cdot - t_0 - j) \rangle \phi_k(t - t_0 - j), t \in [t_0 + j, t_0 + j + 1).$$

Now we can refine the intervals (in order to improve the approximations) by considering the function f_Δ given by $f_\Delta(t) = f(\Delta t)$. By expanding the function f we obtain that for $t \in [t_0 + j\Delta, t_0 + (j+1)\Delta)$,

$$f(t) = f_\Delta\left(\frac{t}{\Delta}\right) = \sum_{k=0}^{\infty} \left\langle f_\Delta, \phi_k\left(\cdot - \frac{t_0}{\Delta} - j\right) \right\rangle \phi_k\left(\frac{t-t_0}{\Delta} - j\right), \quad (4.2.6)$$

and its projection $P_{n,\Delta} f$ on π_n is given by

$$P_{n,\Delta} f(t) = \sum_{k=0}^n \left\langle f_\Delta, \phi_k\left(\cdot - \frac{t_0}{\Delta} - j\right) \right\rangle \phi_k\left(\frac{t-t_0}{\Delta} - j\right), \quad (4.2.7)$$

for $j(t) = \{j : t \in [t_0 + j\Delta, t_0 + (j+1)\Delta)\}$. An important relation for the Laplace transform of the n th Legendre polynomial ϕ_n is given in remark 4.3.1.

Applying formula (4.2.3) with g the m th Legendre polynomial yields the formula

$$\sum_{k=-\infty}^{\infty} \widehat{f}(i2\pi(k+v)) \widehat{\phi}_m^*(i2\pi(k+v)) = \sum_{j=-\infty}^{\infty} \langle f, \phi_m(\cdot - j) \rangle e^{-i2j\pi v} \quad (4.2.8)$$

The RHS is a Fourier series in the coefficients $\langle f, \phi_m(\cdot - j) \rangle$. With the well known IFFT algorithm we can compute these coefficients. The challenge here is the evaluate the LHS for given values of v . In the next section we discuss the concept of Gaussian quadrature. As we will see this gives us the tool to approximate the infinite sum of the LHS accurate with a finite sum.

4.2.3 Gaussian quadrature

Given an inner product $\langle \cdot, \cdot \rangle$ and a set of orthogonal polynomials with respect to that inner product, one can always define a so called Gaussian quadrature rule.

Definition 4.2.3. *Given an inner product $\langle \cdot, \cdot \rangle$ and a set of orthogonal polynomials $\{q_k\}$ the Gaussian quadrature formula (for Gaussian quadrature see the seminal work of Szegő (1975)), is given by*

$$\langle f, q_k \rangle_n = \sum_{k=0}^n \alpha_k f_j(\lambda_k) q_k^*(\lambda_k)$$

$\lambda_k, k = 0, \dots, n$ are the roots of q_{n+1} , and α_k are the so-called Christphel numbers, given by

$$\alpha_k = \frac{1}{\sum_{j=0}^n |q_j(\lambda_k)|^2}$$

In case of the Legendre polynomials, we approximate $\langle f, \phi_k(\cdot - j) \rangle$ by a Gaussian quadrature formula (for Gaussian quadrature see the seminal work of Szegő (1975)), given by

$$\langle f, \phi_k(\cdot - j) \rangle_n = \sum_{k=0}^n \alpha_k f_j(\lambda_k) \phi_k^*(\lambda_k), \quad (4.2.9)$$

where

$$f_j(t) := f(j + t), \quad (4.2.10)$$

$\lambda_k, k = 0, \dots, n$ are the roots of ϕ_{n+1} , and α_k are the so-called Christphel numbers, given by

$$\alpha_k = \frac{1}{\sum_{j=0}^n |\phi_j(\lambda_k)|^2}. \quad (4.2.11)$$

In the next section we develop a quadrature formula for the LHS of formula (4.2.8).

4.3 The backbone of the method

As mentioned in the introduction, the new algorithm works both ways: if some numerical values of the original function are given, then the algorithm generates function values of its Laplace transform (in arbitrary points), and vice versa, given some numerical values of the Laplace transform, the original function can be obtained in arbitrary points. Bearing this to the underlying spaces, we can say that given input from the time domain, the algorithm transfers it to results in the Laplace domain,

4.3 The backbone of the method

and vice versa, from the Laplace domain to the time domain. However, this transfer does not realize directly but through the spaces of the coefficients of the expansions: the coefficients $\langle f, \phi(\cdot - j) \rangle$ of the Legendre expansions, and $\langle \Psi \hat{f}, q_k^v \rangle_v$, defined properly below. The link between the two spaces of expansion coefficients is neither obvious nor complicated, it is realized through a powerful identity, an innovative modification of the so called the Poisson Summation Formula (PSF), which relates the Fourier series of the sequence of Legendre expansion coefficients, namely $\left\{ \sum_j e^{-2\pi i v j} \langle f, \phi(\cdot - j) \rangle \right\}_{k \in \mathbb{N}_0}$, $v \in [0, 1]$, to the coefficients of the expansions of the Laplace transformations, $\langle \Psi \hat{f}, q_k^v \rangle_v$. Diagram 4.1 sketches the way the method works.

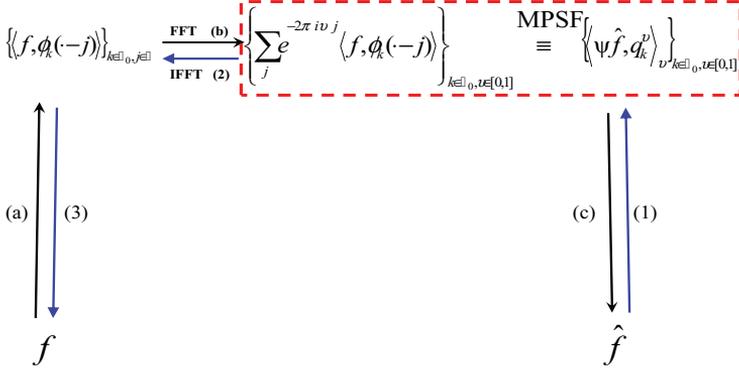


Figure 4.1: Diagram of the flow of the algorithms; the LT: (a) \rightarrow (b) \rightarrow (c), and the inverse transform: (1) \rightarrow (2) \rightarrow (3)

Let $\{\lambda_k^v : k \in \mathbb{Z}, 0 < v < 1\}$, $\{\eta_k^v : k \in \mathbb{Z}, 0 < v < 1\}$ and Ψ be given by

$$\lambda_k^v = i(2\pi(k + v)) \quad w_k^v(t) = \exp(-\lambda_k^v t),$$

and

$$\Psi f(s) = s^{-1} f(s^{-1}), \tag{4.3.1}$$

respectively. We proceed to use the notation $h = \Psi \hat{f}$. Note also that $\Psi h = \hat{f}$, resulting from the definition of Ψ (see relation (4.3.1)). Let us define the inner product $\langle \cdot, \cdot \rangle_v$ needed for the coefficients of expansion for the Laplace transforms

$$\langle h_1, h_2 \rangle_v := \sum_{k=-\infty}^{\infty} \beta_k^v h_1(\eta_k^v) h_2^*(\eta_k^v), \quad \eta_k^v = \frac{1}{\lambda_k^v}, \quad \beta_k^v = |\mu_k^v|^2, \tag{4.3.2}$$

here $h_2^*(\mu_k^v)$ denotes the complex conjugate of $h_2(\mu_k^v)$. We say that $h \in L_v^2$ if h has a finite $\|\cdot\|_v$ norm ($\|\cdot\|_v$ induced by the inner product (4.3.2)). Finally, let us introduce the polynomials $\{q_k^v : k \in \mathbb{N}\}$, given by

$$q_k^v(s) := p_k(s) - (-1)^k \exp(-2\pi i v) p_k(-s), \quad (4.3.3)$$

with

$$p_k(s) = \sqrt{2k+1} \sum_{j=0}^k \frac{j+k!}{k-j!} \frac{(-s)^j}{j!}.$$

Remark 4.3.1. By straightforward computation, the Laplace transform of the n th Legendre polynomial ϕ_n is given by

$$\widehat{\phi}_n(s) := s^{-1} p_n(s^{-1}) - (-1)^n \exp(-s) s^{-1} p_n(-s^{-1}) \quad (4.3.4)$$

Expression (4.3.3) yields a fundamental relation with the Laplace transform of the n th Legendre polynomial ϕ_n :

$$q_n^v = \Psi \widehat{\phi}_n \quad \text{on } \{(i2\pi(k+v))^{-1}, k \in \mathbb{Z}\}. \quad (4.3.5)$$

Theorem 4.3.2. For every $0 < v < 1$ the set $\{q_k^v : k = 0, 1, \dots\}$ is a complete orthogonal set of polynomials in L_v^2 .

For the proof see den Iseger (2006a), Appendix D. The first important consequence of this result is that any $h \in L_v^2$ has the expansion

$$h = \sum_{k=0}^{\infty} \langle h, q_k^v \rangle_v q_k^v, \quad \text{w.r.t. the } \|\cdot\|_v \text{ norm.}$$

A second important consequence of theorem 4.3.2 is that it enables the application of gaussian quadrature. Before proceeding with the quadrature rules, we need another important result. As we mentioned at the beginning of this section, a key result is what we call the Modified Poisson Summation Formula (MPSF). This formula links the expansion coefficients of the Laplace transforms ($\langle \Psi \widehat{f}, q_k^v \rangle_v$) to the expansion coefficients of the L^2 functions ($\langle f, \phi_k(\cdot - j) \rangle$), representing therefore the core of our transform algorithm (see boxed relation in diagram 4.1).

Theorem 4.3.3 (MPSF). The following relation holds true

$$\langle \Psi \widehat{f}, q_k^v \rangle_v = \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle f, \phi_k(\cdot - j) \rangle, \quad (4.3.6)$$

where the RHS is a Fourier series in the coefficients of the Legendre expansion, and $\widehat{f} \in L^2(-i\infty, i\infty)$ denotes the Fourier (Laplace) transform.

4.3 The backbone of the method

Proof. From equations (4.3.5), (4.3.2) and the PSF (4.2.3) the theorem follows. \square

Remark 4.3.4. *The MPSF is in fact a relation between a Fourier series of the coefficients of the Legendre expansion and the coefficients of expansion of $\Psi\hat{f}$, instead of \hat{f} . Since Ψ is invertible and has a simple form (see (4.3.1)) it costs no computational effort to obtain the expansion for \hat{f} once we obtained the expansion for $\Psi\hat{f}$, and vice versa. Furthermore,*

$$\Psi\hat{f} \in L_v^2 \quad \text{if and only if} \quad \{\hat{f}(2\pi i(k+v)) : k \in \mathbb{Z}\} \in l^2. \quad (4.3.7)$$

For this reason, in the further description of the method, we omit the difference between obtaining \hat{f} directly or by inverting Ψ in the expression of $\Psi\hat{f}$.

With the well known IFFT (inverse fast Fourier transform) algorithm (see section 4.5.3) one can compute the sought coefficients $\langle f, \phi_k(\cdot - j) \rangle$. This means, that if one can compute the LHS of (4.3.6) then the algorithm is completed. The problem is, however, that the LHS of (4.3.6) is an infinite summation. We will therefore resort to Gaussian quadrature. Theorem 4.3.2 enables us to approximate $\langle \Psi\hat{f}, q_k^v \rangle_v$ with a Gaussian quadrature formula. That is, $\langle \Psi\hat{f}, q_k^v \rangle_v$ can be approximated with the Gaussian quadrature rule $\langle \Psi\hat{f}, q_k^v \rangle_{vn}$, defined by

$$\langle h_1, h_2 \rangle_{vn} := \sum_{k=0}^n \alpha_k^v h_1(\mu_k^v) h_2^*(\mu_k^v), \quad (4.3.8)$$

with $\mu_k^v, k = 0, \dots, n$ the roots of q_{n+1}^v , and α_k^v the Christoffel numbers related to q_{n+1}^v ,

$$\alpha_k^v = \frac{1}{\sum_{j=0}^n |q_j^v(\mu_k^v)|^2}, \quad (4.3.9)$$

Algorithmically the method can be sketched as follows.

Inverse Laplace Transform:

1. Approximate $\langle \Psi\hat{f}, q_k^v \rangle_v$ with $\langle \Psi\hat{f}, q_k^v \rangle_{vn}$ for $k = 0, 1, \dots, n$.
2. Invert $\sum_{j=0}^{\infty} \exp(-i2\pi vj) \langle f, \phi_k(\cdot - j) \rangle$ with the IFFT algorithm;
3. Approximate $f(j+t)$ with $\sum_{k=0}^n \langle f, \phi_k(\cdot - j) \rangle \phi_k(t)$.

All the operations can be efficiently implemented as matrix products and FFTs (see section 4.5.1 for the matrix representations). The reverse of the algorithm computes the Laplace transform, given specific function values:

Laplace Transform:

- a. Approximate $\langle f, \phi_k(\cdot - j) \rangle$ with $\langle f, \phi_k(\cdot - j) \rangle_n$ for $k = 0, 1, \dots, n$;
- b. Compute $\langle \Psi \hat{f}, q_k^v \rangle_v$ with the FFT;
- c. Approximate $\Psi \hat{f} \approx \sum_{k=0}^n \langle \Psi \hat{f}, q_k^v \rangle_{vn} q_k^v$ with (4.3.8).

An efficient computation method of the quadrature rules of step a, c, and step 1. is given in section 4.5.

Remark 4.3.5. In step 1. of the Inverse Laplace Transform algorithm, through the Poisson Summation Formula, $\langle \Psi \hat{f}, q_k^v \rangle_{vn}$ is interpreted as the Fourier series in step 2. In step b. of the Laplace Transform algorithm one actually computes with the FFT the Fourier series of the sequence of coefficients $\langle f, \phi_k(\cdot - j) \rangle_n$ obtained in step a, which, again through the PSF, can be identified as $\langle \Psi \hat{f}, q_k^v \rangle_v$.

There remains one more issue to discuss. Step (c) computes the expansion of \hat{f} with respect to the $\|\cdot\|_{vn}$ -norm. However, $\|\Psi \hat{f}\|_{vn} = 0$ does not necessarily imply that $\hat{f}(s) \equiv 0$ for all s , it solely implies that $\hat{f}(\mu_k^v) = 0$ (see relation (4.3.2)). In order to be able to compute \hat{f} in arbitrary points, that is, compute an expansion with respect to the L^2 norm, we will need the following result: The inner product

$$\langle \hat{f}, \hat{g} \rangle_{L_n^2} = \int_0^1 \langle \Psi \hat{f}, \Psi \hat{g} \rangle_{vn} dv.$$

coincide with the inner product $\langle \cdot, \cdot \rangle_{L^2}$ on $L^2((-\infty, \infty))$. We will prove this result in the next section.

Remark 4.3.6. It follows from the Plancherel identity for the Fourier transform (see Rudin (1987) theorem 9.13) that $\langle \hat{f}, \hat{g} \rangle_{L^2} = \langle f, g \rangle$. Hence

$$\langle \hat{f}, \hat{g} \rangle_{L_n^2} = \langle \hat{f}, \hat{g} \rangle_{L^2} = \langle f, g \rangle$$

This implies that besides the approximation error due the IFFT in step 2 of the inverse Laplace Transform algorithm, the method is inner product preserving.

4.4 Inner product preserving properties of the method

We start with the following result:

Theorem 4.4.1. Let $\{\mu_k^v; 0 \leq v \leq 1, k = 0, \dots, n\}$ be the roots of q_{n+1}^v ordered such that $Im(\mu_k^v) < Im(\mu_{k+1}^v)$ and suppose that the Christophel numbers (see (4.3.9)) $\{\alpha_k^v; 0 \leq v \leq$

4.4 Inner product preserving properties of the method

$1, k = 0, \dots, n\}$ are ordered in the same way. Let the functions $\mu : [-\frac{n+1}{2}, \frac{n+1}{2}]$ and $\alpha : [-\frac{n+1}{2}, \frac{n+1}{2}]$ be given by

$$\mu \left(v + k - \frac{n+1}{2} \right) = \mu_k^v \quad \text{and} \quad \alpha \left(v + k - \frac{n+1}{2} \right) = \alpha_k^v,$$

for $v \in [0, 1]$ and $k = 0, 1, \dots, n$. The function α and μ are continuous and infinitely differentiable functions and μ is the unique solution of the differential equation

$$\frac{D\mu(v)}{2\pi i} = \alpha(v), \quad \mu(0) = 0. \quad (4.4.1)$$

The function $\mu(v)$ is a strictly increasing function with $\mu\left(-\frac{n+1}{2}\right) = -i\infty$ and $\mu\left(\frac{n+1}{2}\right) = i\infty$. The function μ is an odd function and the function α is an even function, that is, $\mu(t) = -\mu(-t)$ and $\alpha(t) = \alpha(-t)$, for $t \in [-\frac{n+1}{2}, \frac{n+1}{2}]$.

The proof can be found in section 4.9.3. The above result implies that $\|\Psi\hat{f}\|_{vn} = 0$, for all $v \in [0, 1]$ in fact does imply that $\hat{f}(s) \equiv 0$ for all s . We can now proof the claim that the method is inner product preserving.

Theorem 4.4.2. *Let*

$$\langle \hat{f}, \hat{g} \rangle_{L_n^2} = \int_0^1 \langle \Psi\hat{f}, \Psi\hat{g} \rangle_{vn} dv. \quad (4.4.2)$$

The inner products $\langle \cdot, \cdot \rangle_{L^2}$ (of the L^2 -norm) and $\langle \cdot, \cdot \rangle_{L_n^2}$ coincide on $L^2((-\infty, i\infty))$.

Proof. Since

$$\begin{aligned} \langle \hat{f}, \hat{g} \rangle_{L_n^2} &= \int_0^1 \sum_{j=0}^n \Psi\hat{f}(\mu_j^v) (\Psi\hat{g})^*(\mu_j^v) \alpha_j^v dv \\ &= \int_{-\frac{n+1}{2}}^{\frac{n+1}{2}} \Psi\hat{f}(\mu(v)) (\Psi\hat{g})^*(\mu(v)) \alpha(v) dv \\ &= \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} \frac{1}{\mu^2} f(\mu^{-1}) g^*(\mu^{-1}) d\mu \\ &= \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} f(s) g^*(s) ds, \end{aligned}$$

we obtain that $\langle \hat{f}, \hat{g} \rangle_{L_n^2} = \langle \hat{f}, \hat{g} \rangle_{L^2}$. □

Let us now detail on how this result yields an expansion of \hat{f} in L^2 . The quadrature points are the roots of

$$q_n^v(s) = p_n(s) - (-1)^n e^{-i2\pi v} p_n(-s)$$

and thus it holds that on $L^2_{v,n}$ (i.e. the space L^2 induced by $\langle \cdot, \cdot \rangle_{v,n}$)

$$e^{-i2\pi v(s)} = e^{-i2\pi v}$$

with

$$e^{-i2\pi v(s)} = (-1)^n \frac{p_n(s)}{p_n(-s)},$$

The function $v(s)$ is explicitly given by

$$v(s) = \frac{1}{2\pi i} \log \left((-1)^{n+1} \frac{p_{n+1}(-s)}{p_{n+1}(s)} \right).$$

It follows from theorem 4.4.1 that $|e^{-i2\pi v(s)}| = 1$ for $s = it$, $t \in \mathbb{R}$, hence the function

$$\widehat{g}(s) := \sum_{k=0}^n \langle \Psi \widehat{f}, q_k^v \rangle_{v_n} q_k^v(s) \quad \text{with } v = v(s).$$

is properly defined. We aim to show that $\|\widehat{f} - \Psi \widehat{g}\|_{L^2} = 0$ or, equivalently, $\langle \widehat{f} - \Psi \widehat{g}, \widehat{f} - \Psi \widehat{g} \rangle_{L^2} = 0$. Theorem 4.4.2 yields that

$$\langle \widehat{f} - \Psi \widehat{g}, \widehat{f} - \Psi \widehat{g} \rangle_{L^2} = \int_0^1 \left\langle \Psi \widehat{f} - \sum_{k=0}^n \langle \Psi \widehat{f}, q_k^v \rangle_{v_n} q_k^v, \Psi \widehat{f} - \sum_{k=0}^n \langle \Psi \widehat{f}, q_k^v \rangle_{v_n} q_k^v \right\rangle_{v_n} dv = 0,$$

since $\sum_{k=0}^n \langle \Psi \widehat{f}, q_k^v \rangle_{v_n} q_k^v$ is an expansion of $\Psi \widehat{f}$ with respect to the $\|\cdot\|_{v_n}$ -norm (see theorem 4.3.2). This means that the algorithm computes \widehat{f} in arbitrary points (any point needed). In fact we can formalize this discussion with the following: Introduce the functions $\widehat{\psi}_{kj}$, with

$$\widehat{\psi}_j(s) = s^{-1} \left(p_j(s^{-1}) - (-1)^j e^{-i2\pi v(s^{-1})} p_j(-s^{-1}) \right)$$

and

$$\widehat{\psi}_{kj} = e^{-i2k\pi v(s^{-1})} \widehat{\psi}_j(s)$$

Theorem 4.4.3. *The set $\{\psi_{kj}; k \in \mathbb{Z}, j = 0, 1, \dots, n\}$ with $\psi_{kj} = F^{-1} \widehat{\psi}_{kj}$, is a complete orthogonal set in $L^2(-\infty, \infty)$. The identity*

$$\langle \Psi \widehat{f}, \widehat{q}_j^v \rangle_{v,n} = \sum_{k=-\infty}^{\infty} e^{-i2\pi kv} \langle f, \psi_{kj} \rangle \quad (4.4.3)$$

holds. The Laplace transform has the representation

$$\sum_{j=0}^n \langle \Psi \widehat{f}, \widehat{q}_j^v(s) \rangle_{v(s),n} \widehat{\psi}_j(s) \quad (4.4.4)$$

The support of the functions $\{\psi_{kj}; k \in \mathbb{N}, j = 0, 1, \dots, n\}$ is $[0, \infty)$. The functions ψ_{kj} satisfy $\psi_{-(1+k)j}(t) = \psi_{kj}(-t)$.

4.5 Implementation issues of the method

The proof can be found in section 4.10.

Remark 4.4.4. Since the support of the functions $\{\psi_{kj}; k \in \mathbb{N}, j = 0, 1, \dots, n\}$ is $[0, \infty)$, we obtain for a function f with support $[0, \infty)$

$$\langle \Psi \widehat{f}, \widehat{q}_j^v \rangle_{v,n} = \sum_{k=0}^{\infty} e^{-i2\pi kv} \langle f, \psi_{kj} \rangle$$

4.5 Implementation issues of the method

4.5.1 Matrix representations for the quadrature rules in the Laplace domain

Matrix representations facilitate an efficient computer implementation of the method, it enables us to make use of the so called BLAS library (see <http://www.netlib.org/blas/>). Let us first introduce the multiplication operator M induced by the norm $\|\cdot\|_v$, defined by $M(v) : L_v^2 \rightarrow L_v^2$,

$$Mh(s) = sh(s), \text{ w.r.t. the norm } \|\cdot\|_v. \quad (4.5.1)$$

We denote with \widehat{M} the matrix representation of M with respect to the basis $\{q_k^v\}$, that is, $(\widehat{M})_{kj} = \langle Mq_k^v, q_j^v \rangle_v$. In den Iseger (2006a) it is shown that

$$\widehat{M} = \begin{bmatrix} \frac{1}{2} \frac{1+\exp(-i2\pi v)}{1-\exp(-i2\pi v)} & -\gamma_0 & 0 & \cdots & 0 & \cdots \\ \gamma_0 & 0 & -\gamma_1 & \ddots & \vdots & \cdots \\ 0 & \gamma_1 & \ddots & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & 0 & -\gamma_{n-1} & \cdots \\ 0 & \cdots & 0 & \gamma_{n-1} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad (4.5.2)$$

where the numbers γ_k are given by

$$\gamma_k = \frac{1}{2} \frac{1}{\sqrt{2k+1}} \frac{1}{\sqrt{2k+3}}.$$

The finite rank approximation $M_n : \pi_n \rightarrow \pi_n$ is $M_n = P_n M$, with P_n the orthogonal projector on π_n (the space of polynomials with degree less than or equal to n , and the matrix representation of M_n with respect to the basis $\{q_k^v\}$ is given by

$$(\widehat{M}_n)_{kj} = \langle P_n M q_k^v, q_j^v \rangle_v = \langle M q_k^v, q_j^v \rangle_v = (\widehat{M})_{kj}.$$

Since the matrix \widehat{M}_n is skew symmetric, it is diagonalizable and we can write

$$\widehat{M}_n = VD(\mu^v)V^*,$$

where $D(\mu^v)$ denotes the diagonal matrices with the pure imaginary valued eigenvalues $\{\mu_k^v : k = 0, 1, \dots, n\}$ of \widehat{M}_n , and V is a unitary matrix (that is $VV^* = V^*V = I$), the eigenvectors of \widehat{M}_n are the columns of V . For the final step of the matrix representation of the quadrature rules we need the following result (for the full result and its proof see theorem 4.9.5):

Theorem 4.5.1.

$$\left\langle (\Psi \widehat{f}) q_k, q_j \right\rangle_{vn} = \left\langle \Psi \widehat{f}(M_n) q_k, q_j \right\rangle_v = \left\langle \Psi \widehat{f}(M_n) q_k, q_j \right\rangle_{vn}. \quad (4.5.3)$$

This result implies that

$$(\Psi \widehat{f}(\widehat{M}_n) e_0)_k = \left\langle \Psi \widehat{f}, q_k^v \right\rangle_{vn}, \quad (4.5.4)$$

where

$$\Psi \widehat{f}(\widehat{M}_n) := VD(\Psi \widehat{f}(\mu^v))V^*.$$

This leads to the following matrix representation for the quadrature rule.

Theorem 4.5.2. *The identities*

$$\begin{aligned} \left\langle \Psi \widehat{f}, q^v \right\rangle_{vn} &= V_0 \Psi \widehat{f}(\mu^v), \\ \Psi \widehat{f}(\mu^v) &= V_0^{-1} \left\langle \Psi \widehat{f}, q^v \right\rangle_{vn}, \end{aligned}$$

with $V_0(v) = VD(V^* e_0)$ and $V_0^{-1}(v) = D^{-1}(V^*(v) e_0) V^*(v)$.

4.5.2 Matrix representations for the quadrature rules in the time domain

Similarly as for the Laplace domain in section 4.5.1, let us introduce the multiplication operator $N : L^2[0, 1] \rightarrow L^2[0, 1]$, defined by

$$Nf(t) = tf(t) \text{ w.r.t. the norm on } L^2, \text{ i.e., } \|Nf - g\|_{L^2} = 0, \text{ with } g(t) := tf(t). \quad (4.5.5)$$

4.5 Implementation issues of the method

We denote with \widehat{N} the matrix representation of N with respect to the basis $\{\phi_k\}$, that is, $\widehat{N}_{kj} = \langle N\phi_k, \phi_j \rangle$. In the same manner as for M we obtain that

$$\widehat{N} = \begin{bmatrix} \frac{1}{2} & \theta_0 & 0 & \cdots & 0 & \cdots \\ \theta_0 & \frac{1}{2} & \theta_1 & \ddots & \vdots & \cdots \\ 0 & \theta_1 & \ddots & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & \frac{1}{2} & \theta_{n-1} & \cdots \\ 0 & \cdots & 0 & \theta_{n-1} & \frac{1}{2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

with

$$\theta_k = \frac{k+1}{4k+2} \sqrt{\frac{2k+1}{2k+3}}. \quad (4.5.6)$$

The finite rank approximation $N_n : \pi_n \rightarrow \pi_n$ is $N_n = P_n N$, with P_n the orthogonal projector on π_n , and the matrix \widehat{N}_n is given by $(\widehat{N}_n)_{kj} = (\widehat{N})_{kj}$, $k, j = 0, \dots, n$. Since \widehat{N}_n is symmetric, it is a diagonalizable and we can write

$$\widehat{N}_n = UD(\lambda)U^*, \quad (4.5.7)$$

where $D(\lambda)$ denotes the diagonal matrix with the real valued eigenvalues $\{\lambda_k : k = 0, 1, \dots, n\}$ of \widehat{N}_n . The matrix U is unitary and the eigenvectors of \widehat{N}_n are the columns of U . Using again relation (4.5.3) (see again theorem 4.9.5) yields that

$$(f(\widehat{N}_n))_{kj} = \langle f\phi_k, \phi_j \rangle, \quad (4.5.8)$$

where

$$f(\widehat{N}_n) := UD(f(\lambda))U^*.$$

This translates to the matrix representation of the quadrature rules in the time domain:

Theorem 4.5.3. *The following two identities are valid:*

$$\begin{aligned} \langle f, \phi \rangle_n &= U_0 f(\lambda), \\ f(\lambda) &= U_0^{-1} \langle f, \phi \rangle_n. \end{aligned}$$

Theorems 4.5.2 and 4.5.3 represent an efficient computation method for steps a. and 1., respectively, of the transform and inverse transform algorithms (section 4.3). We will now discuss another implementation issue, using damping factors in order to enhance convergence and speed of the algorithms.

4.5.3 Using damping factors with the FFT and IFFT

For our purposes we consider the one-sided Z-transform

$$F(z) = \sum_{k=0}^{+\infty} f_k z^k.$$

The N -points IFFT (step 2.) computes

$$\widehat{f}_k = \sum_{j=-\infty}^{+\infty} f_{k+jN} = \sum_{j=0}^{+\infty} f_{k+jN},$$

see den Iseger (2006a), Theorem H.1. If f_{k+jN} converges slowly to 0, in order to obtain convergence ($\sum_j f_{k+jN} \approx f_k$), one needs to choose N large. However, choosing N too large takes toll of the speed of the algorithm. In order to avoid giving in on speed, one can use the so called damped Z-transform

$$F_r(z) = F(rz), \tag{4.5.9}$$

then the N -point IFFT computes

$$\widehat{f}_k = \sum_{j=0}^{+\infty} r^{k+jN} f_{k+jN} = r^k f_k + \sum_{j=1}^{\infty} r^{k+jN} f_{k+jN}. \tag{4.5.10}$$

In order to find out about the effects of the parameters on our transform algorithm, we rearrange the terms,

$$r^{-k} \widehat{f}_k - f_k = \sum_{j=1}^{\infty} r^{jN} f_{k+jN}, \tag{4.5.11}$$

and we will estimate the error of the approximation

$$\sum_{k=0}^{n-1} \left\| r^{-k} \widehat{f}_k - f_k \right\|^2 = \sum_{k=0}^{n-1} \left(\sum_{j=1}^{\infty} r^{jN} f_{k+jN} \right)^2 \leq \frac{r^{2N}}{1-r^{2N}} \sum_{k=0}^{n-1} \sum_{j=1}^{\infty} |f_{k+jN}|^2 = \frac{r^{2N}}{1-r^{2N}} \|f\|_2^2.$$

In conclusion, the speed and precision of our transform method will depend, among others, on a good balance between the sizes of the damping factor r and the number of inversion points N . A more detailed explanation about the choice of parameter values will be provided later (see Remarks 4.5.9, 4.5.8 and 4.5.10), now we will rather focus on how $F_r(z)$ can be computed efficiently. Relating definition (4.5.9) to Laplace transforms yields an important property

$$\widehat{f}_\alpha(s) = \widehat{f}(s + \alpha) \Leftrightarrow f_\alpha(t) = e^{-\alpha t} f(t). \tag{4.5.12}$$

The coefficients of expansion $\langle f_\alpha, \phi_k(\cdot - j) \rangle$ become

$$\langle e^{-\alpha t} f(t), \phi_k(t - j) \rangle = e^{-\alpha j} \langle e^{-\alpha u} f(j + u), \phi_k(u) \rangle, \quad u \in [0, 1]. \tag{4.5.13}$$

4.5 Implementation issues of the method

This enables us to compute $f(j+x)$ for any $x \in [0, 1]$ as

$$f(j+x) = \sum_{k=0}^{\infty} \langle e^{-\alpha u} f(j+u), \phi_k(u) \rangle e^{\alpha x} \phi_k(x). \quad (4.5.14)$$

Finally, the Fourier series (see step (b)) is given by

$$\sum_{j=0}^{\infty} e^{-\alpha j} \langle e^{-\alpha \cdot} f(j+\cdot), \phi_k \rangle e^{-2\pi i v j} = \sum_{j=0}^{\infty} \langle f_{\alpha}(j+\cdot), \phi_k \rangle e^{-2\pi i v j} = \langle \Psi \widehat{f}_{\alpha}, q_k^v \rangle_v. \quad (4.5.15)$$

That is, the algorithm remains the same when working with damping factors. What we do have to revise is how the matrix representations need to be changed in order to incorporate damping factors.

4.5.4 Back to matrix representations: dealing with damping factors

According to the definition of the multiplication operator N ,

$$(e^{-\alpha N} f)(t) = e^{-\alpha t} f(t), \quad t \in [0, 1].$$

We seek the matrix representation for the damped series in relation (4.5.15):

$$\langle e^{-\alpha N} f(j+\cdot), \phi_k \rangle = \sum_{l=0}^{\infty} \langle f(j+\cdot), \phi_l \rangle \langle e^{-\alpha N} \phi_l, \phi_k \rangle, \quad (4.5.16)$$

according to the Parseval relation. Since one can approximate $\langle e^{-\alpha N} \phi_l, \phi_k \rangle$ by $\langle e^{-\alpha N_n} \phi_l, \phi_k \rangle$, and $\langle f(j+\cdot), \phi_l \rangle, l = 0, \dots, \infty$ by $\langle f(j+\cdot), \phi_l \rangle, l = 0, \dots, n$,

$$\langle e^{-\alpha N} f(j+\cdot), \phi_k \rangle \approx \sum_{l=0}^n \langle f(j+\cdot), \phi_l \rangle \langle e^{-\alpha N_n} \phi_l, \phi_k \rangle =: e^{-\alpha \widehat{N}_n} [\langle f(j+\cdot), \phi \rangle]_n. \quad (4.5.17)$$

Hence we obtained

$$\langle \Psi \widehat{f}_{\alpha}, q_k^v \rangle_v \approx e^{-\alpha \widehat{N}_n} \left[\sum_{j=0}^{\infty} e^{-\alpha j} e^{-2\pi i v j} \langle f, \phi(\cdot - j) \rangle \right]_n. \quad (4.5.18)$$

Now we can use the matrix representations

$$e^{\alpha \widehat{N}_n} = UD(e^{\alpha \lambda})U^* \quad (4.5.19)$$

together with theorem 4.5.2 for the matrix representation of the quadrature formula $\langle \cdot, \cdot \rangle_{vn}$:

Theorem 4.5.4. *The following identities are approximately valid:*

$$\Psi \widehat{f}_\alpha(\mu^v) \approx V_0^{-1} U D(e^{-\alpha\lambda}) U^* \left(L_n f_\alpha(e^{-2\pi i v}) \right), \quad (4.5.20)$$

$$L f_\alpha(e^{-2\pi i v}) \approx U D(e^{\alpha\lambda}) U^* V_0 \Psi \widehat{f}(\mu^v), \quad (4.5.21)$$

where the vectors $L f_\alpha(e^{-2\pi i v})$ and $L_n f_\alpha(e^{-2\pi i v})$ are given by

$$L f_\alpha(e^{-2\pi i v}) = \sum_{j=0}^{\infty} \langle f_\alpha, \phi(\cdot - j) \rangle e^{-2\pi i j v} \quad \text{and} \quad L_n f_\alpha(e^{-2\pi i v}) = \sum_{j=0}^{\infty} \langle f_\alpha, \phi(\cdot - j) \rangle_n e^{-2\pi i j v}.$$

The diagram of the method, presented in figure 4.1 changes to the diagram presented in figure 4.2.

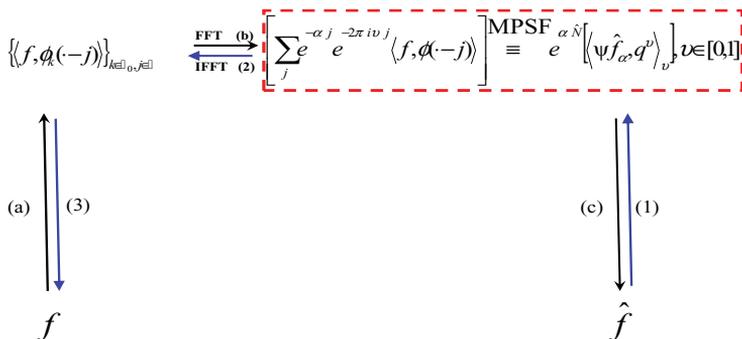


Figure 4.2: Diagram of the flow of the algorithms; the LT: (a) \rightarrow (b) \rightarrow (c), and the inverse transform: (1) \rightarrow (2) \rightarrow (3)

Remark 4.5.5. *It can be shown (see relation (4.9.5)) that the inversion algorithm is the most stable for $v = 0.5$. On the other hand, one aims to be able to compute the quadrature rule for all v in any point (not just specific, predefined points). In order to achieve this goal set first $v = 0.5$ and obtain*

$$e^{\alpha \widehat{N}_n} \left[\left\langle \psi \widehat{f}_\alpha, q_k^{0.5} \right\rangle_{0.5} \right] = \mathcal{F}_{e^{-\alpha}}(-1) = \mathcal{F}(-e^{-\alpha}).$$

Although it seems a restriction to have set $v = 0.5$, setting at the same time $\alpha = c + i2\pi(w - 0.5)$ solves our implementation problem, since $\mathcal{F}(e^{-c} e^{-i2\pi w}) = \mathcal{F}_{e^{-c}}(e^{-i2\pi w})$.

4.5 Implementation issues of the method

Remark 4.5.6. Since the matrix \widehat{N}_n is tri-diagonal, real and symmetric, and \widehat{M}_n is tri-diagonal, real, and skew-symmetric, we can use the well-known QR algorithm for efficient computation of the eigenvalues and eigenvectors.

4.5.5 Detailed Laplace and Laplace inversion algorithms with matrix representations and damping factors

Let us first give the algorithm for the inversion of Laplace transforms followed by the algorithm which determines the Laplace transform given the value of the function in a prespecified set of points.

Algorithm 1: Inverse Laplace transform

parameters: ($m = 8\widehat{m}$ and $c = \frac{44}{m}$ if damping is used) or ($m = \widehat{m}$ and $c = 0$ if no damping is used).

input: n , numerical values of the transform $\Psi \widehat{f}_\alpha(\mu^v)$, for $v = 0.5$, $\alpha = c - i\pi + \frac{2\pi ij}{m}$ (see remark 4.5.5), $j = 0, 1, \dots, m-1$, $i = \sqrt{-1}$

output: $\langle f, \phi_k(\cdot - j) \rangle$, $k = 0, 1, \dots, n-1$, $j = 0, 1, \dots, \widehat{m}-1$.

1. Compute $e^{\alpha \widehat{N}} \left[\left\langle \Psi \widehat{f}_\alpha, q^v \right\rangle_v \right]$ with $UD (e^{\alpha \lambda}) U^* V_0 \Psi \widehat{f}(\mu^v)$. This yields the value for $L_n f_\alpha(e^{-i2\pi v})$ given by (4.5.4).
2. For $k = 0, 1, \dots, n$ approximate $\langle f_\alpha, \phi_k(\cdot - j) \rangle$ with the IFFT algorithm (4.5.11).
3. Compute f with its expansion in the basis $\{\phi_k\}$.

Remark 4.5.7. In order to improve the approximation we can refine the expansion. Compute therefore $P_{n,\Delta} f$ by inverting the Laplace transform $\widehat{f}_\Delta(s) = \frac{1}{\Delta} \widehat{f}\left(\frac{s}{\Delta}\right)$.

Algorithm 2: Laplace transform

parameters: ($m = 8\widehat{m}$ and $c = \frac{44}{m}$ if damping is used) or ($m = \widehat{m}$ and $c = 0$ if no damping is used).

input: n , $\langle f, \phi_k(\cdot - j) \rangle$, $k = 0, 1, \dots, n-1$, $j = 0, 1, \dots, \widehat{m}-1$.

output: numerical values of the Laplace transform $\Psi \widehat{f}_\alpha(\mu^v)$, for $v = 0.5$, $\alpha = c - i\pi + \frac{2\pi ij}{m}$ (see remark 4.5.5), $j = 0, 1, \dots, m-1$.

- a. Compute $\langle f_\alpha, \phi_k(\cdot - j) \rangle$ with the quadrature rule $\langle f_\alpha, \phi_k(\cdot - j) \rangle_{n_i}$
- b. Compute $\left\langle \Psi \widehat{f}_\alpha, q_k^v \right\rangle_v$ with the FFT;
- c. Compute $\Psi \widehat{f}(\mu^{0.5})$ with $V_0^{-1} UD (e^{-\alpha \lambda}) U^* (L_n f_\alpha (e^{-\pi i}))$.

Remark 4.5.8. The choice of the parameter M_2 is a trade-off between the running time and the accuracy of the algorithm. The actual running time is $M_2 \log(M_2)$, hence from this perspective we want to choose M_2 as small as possible. However in (4.5.11) of the algorithm we multiply the results with a factor $\exp(c\ell) = \exp(\frac{44}{M_2}\ell)$. In order to obtain a numerically stable algorithm one needs to choose M_2 as large as possible. We recommend setting $M_2 = 8M$.

Remark 4.5.9. Extensive numerical experiments show that $n = 16$ gives, for all smooth functions, results attaining the machine precision. For double precision we choose $c = \frac{44}{M_2}$ and $M_2 = 8M$. For $n = 16$ we need 16 Laplace transform values for the quadrature rule and we use an oversampling factor of $M_2/M = 8$; thus, on average, we need 128 Laplace transform values for the computation of the function on one interval. The method can be efficiently implemented with modern numerical libraries with FFT routines and matrix/vector multiplication routines (BLAS). If one is satisfied with less precision, then the oversampling factor and/or the number of quadrature points can be reduced. For other machine precisions it is recommended to choose c somewhat larger than $\frac{-\log(\epsilon)}{M_2}$, with ϵ the machine precision.

Remark 4.5.10. Since the Gaussian quadrature rule is exact on the space of polynomials of degree $2n - 1$ or less, π_{2n-1} , we obtain that

$$\sum_{k=1}^n \alpha_k^v = \langle \mathbf{1}, \mathbf{1} \rangle_{vn} = \langle \mathbf{1}, \mathbf{1} \rangle_v = \frac{\langle q_0^v, q_0^v \rangle_v}{1 - \cos(2\pi v)} = \frac{1}{1 - \cos(2\pi v)}. \quad (4.5.22)$$

This is clearly minimal for $v = 0.5$. Therefore the algorithms of this section are numerically more stable than the algorithms given in section 4.3.

Remark 4.5.11. If n is odd then one of the $\{\mu_k^{0.5} : k = 1, 2, \dots, n\}$ is equal to zero. The evaluation of $\Psi \hat{f}(\mu^v)$ can be a problem in this case. To avoid this, take n even.

Remark 4.5.12. We can compute the values $\frac{1}{\Delta s} \hat{f}(\frac{1}{\Delta}(\rho + \frac{1}{s}))$ by using the inner products $\langle f_\Delta, \phi_k(\cdot - j) \rangle$, $k = 0, 1, \dots, n$, $j = 0, 1, \dots, \hat{m} - 1$ as input in the above algorithm (with $f_\Delta(t) = f(\Delta t)$).

4.6 Error analysis

The aim of this section is to estimate the error made by the approximation by the quadrature rule $\langle \cdot, \cdot \rangle_{vn}$. The following theorem gives the estimation for the approximation error.

4.7 Modifications for non-smooth functions

Theorem 4.6.1. For a smooth function f (with $f^{(2)} \in L^1(-\infty, +\infty)$) the error made by the approximation by the quadrature rule $\langle \cdot, \cdot \rangle_{vn}$ is bounded as follows:

$$\sqrt{\sum_{k=0}^n \left| \langle \Psi \hat{f}, q_k^v \rangle_{vn} - \langle \Psi \hat{f}, q_k^v \rangle_v \right|^2} \leq \frac{1}{2\sqrt{4n^2-1}} \sum_{j=-\infty}^{\infty} \left(V_j f^{(1)}(0) + \|V_j f^{(2)}\|_1 \right), \quad (4.6.1)$$

where $f^{(k)}$ denotes the k th derivative of f and the operator V_x is for $x \geq 0$ given by

$$V_x g(y) = \begin{cases} 0 & , y < 0 \\ \int_0^1 g(x+t+y) \phi_{n+1}(t) dt & , y \geq 0. \end{cases} \quad (4.6.2)$$

and the operator V_x is for $x < 0$ given by

$$V_x g(y) = \begin{cases} 0 & , y > 0 \\ \int_0^1 g(x+t+y) \phi_{n+1}(t) dt & , y \leq 0. \end{cases} \quad (4.6.3)$$

In order to show that the presented method is a fast and stable algorithm, one needs to show that the bound in (4.6.1) converges fast to 0. From formula 4.2.5 it follows that the Legendre polynomial ϕ_{n+1} , $\langle f^{(k)}, \phi_{n+1} \rangle$ is bounded by $\frac{\sqrt{2n+3}}{(2n+3)!} \|f^{(k+n+1)}\|_{\infty}$. The above ensure a fast convergence of the r.h.s. in (4.6.1) to 0. The detailed proof of theorem 4.6.1 can be found in 4.11.

4.7 Modifications for non-smooth functions

As shown in the error analysis of section 4.6, the transformation method works very well for smooth functions. In this section we extend the method to functions with local singularities. Let ζ be a sufficiently smooth function with finite support contained in $[k_1, k_2]$. We aim to evaluate inner products of the form $\{\langle f, \zeta(\cdot - m) \rangle\}_{m=-\infty}^{\infty}$ for the computation of approximations such as

$$\sum_{m=-\infty}^{\infty} \langle f, \zeta(\cdot - m) \rangle \zeta(t - m),$$

for suitable basis functions $\{\zeta(\cdot - m)\}_{m=-\infty}^{\infty}$ and $\{\zeta(\cdot - m)\}_{m=-\infty}^{\infty}$. For notional convenience, let $\zeta_m := \zeta(\cdot - m)$. We obtain from the inner product theorem that

$$\begin{aligned} \langle f, \zeta_m \rangle &= \langle \hat{f}, \hat{\zeta}_m \rangle_{L^2} = \int_0^1 \langle \Psi \hat{f}, \Psi \hat{\zeta}_m \rangle_{vn} dv \\ &= \sum_{j=0}^n \int_0^1 \langle \Psi \hat{f}, q_j^v \rangle_{vn} \langle q_j^v, \Psi \hat{\zeta}_m \rangle_{vn} dv. \end{aligned}$$

Let

$$F_{kj} := \int_0^1 e^{i2\pi kv} \langle \Psi \widehat{f}, q_j^v \rangle_{vn} dv,$$

and

$$W_{kmj}^n := \int_0^1 e^{i2\pi kv} \langle q_j^v, \Psi \widehat{\zeta}_m \rangle_{vn} dv,$$

then we obtain through the Plancharel theorem (see Mallat(2001)) that

$$\langle f, \zeta_m \rangle = \sum_{j=0}^n \sum_{k=-\infty}^{\infty} F_{kj} W_{kmj}^n. \quad (4.7.1)$$

Since ζ_m is a smooth function and $\widehat{\zeta}_m(s) = e^{ism} \widehat{\zeta}(s)$, we obtain that

$$W_{kmj}^n \approx \int_0^1 e^{i2\pi kv} \langle q_j^v, \Psi \widehat{\zeta}_m \rangle_v dv = \int_0^1 e^{i2\pi(k+m)v} \langle q_j^v, \Psi \widehat{\zeta} \rangle_v dv = W_{k-m,j}, \quad (4.7.2)$$

with

$$W_{kj} := \int_0^1 e^{i2\pi kv} \langle q_j^v, \Psi \widehat{\zeta} \rangle_v dv. \quad (4.7.3)$$

Applying the PSF (see (4.3.6)) and the inverse Fourier series theorem we obtain that

$$W_{kj} = \langle \phi_j(\cdot - k), \zeta \rangle. \quad (4.7.4)$$

Approximating (4.7.1) by substituting (4.7.2), then using (4.7.4) together with the fact that the support of ζ is contained in $[k_1, k_2]$, finally yields that

$$\langle f, \zeta_m \rangle \approx \sum_{j=0}^n \sum_{k=k_1}^{k_2-1} F_{k+mj} W_{kj}. \quad (4.7.5)$$

Hence for almost every function f we can compute the expansion coefficients $\langle f, \zeta(\cdot - m) \rangle$ accurately. An example of widely used basis functions, the local cosine basis, is given in section 4.8.1.

4.8 Numerical test results

In order to test the method we have examined all the Laplace transforms that are used in the overview article of Martin and Davies (1979). In this section we discuss the results for eight analytical test functions. These results are presented in table 4.1. The inversions are done on 32 intervals and so $M = 32$. Also $M_2 = 256 = 8M$ and $\Delta = 1/16$ and for all the examples we have taken the damping factor $44/M_2$. Besides the choice of the previous parameters we have taken $n = 16$. From all the examples we can conclude that the method is very accurate, robust and fast.

4.8 Numerical test results

Table 4.1: Results for analytical test functions

nr	function	Laplace Transform	MAE
1	$J_0(x)$	$(s^2 + 1)^{-1/2}$	6e-15
2	$\exp(-x/2)$	$(s + 1/2)^{-1}$	4e-15
3	$\exp(-0.2x) \sin(x)$	$((s + 0.2)^2 + 1)^{-1}$	4e-15
4	1	s^{-1}	7e-15
5	x	s^{-2}	3e-15
6	$x \exp(-x)$	$(s + 1)^{-2}$	2e-15
7	$\sin(x)$	$(s^2 + 1)^{-1}$	4e-15
8	$x \cos(x)$	$(s^2 - 1)(s^2 + 1)^{-2}$	5e-15

4.8.1 The local cosine basis and numerical results for non-smooth functions

The local cosine basis is defined by

$$\zeta_k(t) = W(t) \cos\left(\left(\pi\left(k + \frac{1}{2}\right)\right) \frac{t + \delta}{2\delta}\right),$$

for $k = 0, 1, \dots, K$ and

$$W(t) = \begin{cases} 0 & \text{if } t < -2\delta \\ w\left(\frac{t + \delta}{\delta}\right), & \text{if } -2\delta < t \leq 0 \\ 1, & \text{if } 0 < t \leq 1 \\ w\left(1 - \frac{t-1}{\delta}\right), & \text{if } 1 < t \leq 1 + 2\delta \\ 0 & \text{if } t > 1 + 2\delta, \end{cases}$$

with $w(t)$ a so-called window function satisfying $w^2(t) + w^2(-t) = 1$. The window function is recursively defined as

$$w_{n+1}(t) = w_n\left(\sin\left(\frac{\pi t}{2}\right)\right), \text{ and } w_0 = \sin\left(\frac{\pi}{4}(1+t)\right).$$

The resulting window function w_n has $2^n - 1$ vanishing derivatives at $t = \pm 1$, and so the ζ_k are $2^n - 1$ times continuously differentiable.

Consider the Laplace transform

$$\hat{f}(s) = \frac{1 - \exp(-s)}{s}, \tag{4.8.1}$$

this yields that the function f is given by $f(t) = 1, 0 < t \leq 1$. The results for $\delta = 4, N = 36, K = 56$, and $\Delta = 1/16, 1/32$, or $1/128$, are summarized in table 4.2.

Table 4.2: Numerical values for f with Laplace transform given by (4.8.1)

t	error with $\Delta = 1/16$	error with $\Delta = 1/32$	error with $\Delta = 1/128$
0.1	-5.6e-004	-1.0e-005	9.2e-013
0.2	-1.0e-005	9.6e-013	-6.3e-012
0.3	-7.3e-012	-7.8e-013	6.2e-012
0.4	6.8e-013	-5.1e-012	-1.0e-012
0.5	1.9e-011	1.3e-011	1.8e-011
0.6	-1.9e-012	5.5e-012	8.2e-013
0.7	1.8e-012	1.0e-012	-8.1e-012
0.8	-3.8e-005	-6.2e-013	6.6e-012
0.9	-1.5e-003	-3.8e-005	-3.6e-012
1.0	-1.3e-003	-1.3e-003	-1.3e-003
1.1	5.6e-004	1.0e-005	-3.6e-013
1.2	1.0e-005	2.1e-013	5.1e-012
1.3	1.3e-011	-7.6e-014	-1.6e-012
1.4	-3.5e-013	-7.5e-013	3.8e-013
1.5	-1.5e-011	2.3e-012	-4.7e-012
1.6	4.6e-013	3.9e-013	2.9e-014
1.7	-1.9e-012	3.2e-014	1.2e-012
1.8	2.4e-012	-9.0e-015	-4.5e-013
1.9	-8.7e-014	-2.3e-013	1.5e-013

We see that despite the discontinuity in $t = 1$, we can compute the result on the intervals $[\delta\Delta, 1 - \delta\Delta]$ and $[1 + \delta\Delta, \infty)$ with almost machine precision.

4.9 Computational issues

4.9.1 Gaussian quadrature rules

Gaussian quadrature is commonly used for numerical evaluation (approximation) of integrals. Examples are Legendre, Chebyshev, Laguerre, and Hermite Gaussian quadrature rules, which are mostly very accurate. With Gaussian quadrature one can approximate any integral with respect to a positive measure (or equivalently any inner product induced by this measure).

Definition 4.9.1. [Gaussian quadrature, cf. Szegő (1975)] Let the inner product $\langle \cdot, \cdot \rangle$ be given

4.9 Computational issues

by

$$\langle f, g \rangle = \int_I f(t) g^*(t) \mu(dt),$$

with μ a given positive measure and I a subinterval of the real axis or a subinterval of the imaginary axis. Let $\{q_k\}$ be an orthogonal set with respect to this inner product. Let the inner product $\langle \cdot, \cdot \rangle_n$ given by

$$\langle f, g \rangle_n := \sum_{k=0}^n \alpha_k f(\mu_k) g^*(\mu_k),$$

with $\{\mu_k; k = 0, 1, \dots, n\}$ the simple roots of q_{n+1} and the strictly positive numbers $\{\alpha_k; k = 0, 1, \dots, n\}$, the so called Christphel numbers, are given by

$$\alpha_k = \frac{1}{\sum_{j=0}^{n-1} |q_j(\mu_k)|^2} = \frac{A_n}{A_{n-1}} \frac{1}{\partial q_n(\mu_j)} \frac{1}{q_{n-1}^*(\mu_j)}, \quad (4.9.1)$$

with A_n the highest coefficient in q_n . The roots $\{\mu_k; k = 0, 1, \dots, n\}$ are in I . The inner product $\langle \cdot, \cdot \rangle_n$ is called the Gaussian quadrature rule and is the unique $(n+1)$ th order quadrature rule satisfying

$$\langle p, \mathbf{1} \rangle_n = \langle p, \mathbf{1} \rangle, \text{ for all } p \in \pi_{2n+1}, \quad (4.9.2)$$

where π_n is the space of polynomials with degree not exceeding n , and $\mathbf{1}$ is the constant function $\mathbf{1}(t) = 1$.

4.9.2 The multiplication operator

The multiplication operator plays a central role in the matrix representations of the quadrature rules (see sections 4.5.1, 4.5.2 and 4.5.4).

Definition 4.9.2. Let $\langle \cdot, \cdot \rangle_v$ the inner product (similarly as defined by (4.3.2)), with representation

$$\langle f, g \rangle_v = \int_I f(t) g^*(t) d\mu(t),$$

with μ a given positive measure and I a given bounded interval. The multiplication operator $M : L_v^2 \rightarrow L_v^2$ is given by

$$(Mh)(s) = sh(s), \text{ w.r.t. } L_v^2.$$

Lemma 4.9.3. If the interval I is a subset of the real axis axis then the multiplication operator is a self-adjoint operator. If the interval I is a subset of the imaginary axis axis then the multiplication operator is a skew self-adjoint operator.

Proof. Since

$$\begin{aligned}\langle Mf, g \rangle_v &= \int tf(t)g^*(t)d\mu(t) \\ &= \int f(t)(tg(t))^*d\mu(t) \\ &= \langle f, Mg \rangle_v,\end{aligned}$$

we obtain that the multiplication operator is a self-adjoint operator. The case that the support of μ is contained in the imaginary axis can be proved with a similar argument. \square

Definition 4.9.4. Let $\langle \cdot, \cdot \rangle$ be a given inner product and let $\{q_k\}$ be a polynomial orthogonal set with respect to this inner product. The finite rank operator $M_n : \pi_n \rightarrow \pi_n$ is given by $M_n x := P_n M x$, with P_n the orthogonal projector on π_n given by

$$P_n f = \sum_{k=0}^{n-1} \langle f, q_n \rangle q_n.$$

We denote with \widehat{M}_n and M the matrix representations with respect to the basis $\{q_k\}$ of M_n and M . hence, the matrix \widehat{M}_n is the n th order principal matrix of the matrix \widehat{M} . Let the inner product $\langle \cdot, \cdot \rangle_n$ be given by

$$\langle f, g \rangle_{M_n} := \langle f(M_n) \mathbf{1}, g(M_n) \mathbf{1} \rangle_v. \quad (4.9.3)$$

Theorem 4.9.5. The inner product $\langle \cdot, \cdot \rangle_{M_n}$ is the Gaussian quadrature rule induced by the polynomials $\{q_0, q_1, \dots\}$. Moreover,

$$\langle f q_k, q_j \rangle_{M_n} = \langle f(M_n) q_k, q_j \rangle_v = \langle f(M_n) q_k, q_j \rangle_{v_n},$$

with the Gaussian quadrature $\langle \cdot, \cdot \rangle_{v_n}$ defined by (4.3.8).

Proof. It has been proven in den Iseger (2006a) that $\langle f, g \rangle_{M_n} = \langle f, g \rangle_{v_n}$. Since $q_k(M) \mathbf{1} \in \pi_n$, we obtain that

$$q_k(s) = q_k(M) \mathbf{1} = q_k(M_n) \mathbf{1}, \text{ w.r.t. } \|\cdot\|_v.$$

Hence,

$$\begin{aligned}\langle f q_k, q_j \rangle_{v_n} &= \langle f(M_n) q_k(M_n) \mathbf{1}, q_j(M_n) \mathbf{1} \rangle_v \\ &= \langle f(M_n) q_k, q_j \rangle_v \\ &= \langle f(M_n) q_k, q_j \rangle_{v_n}.\end{aligned}$$

\square

We will need the following result in the following section (cf. theorem 3.8.2).

Theorem 4.9.6. The numbers $\{\mu_k^v\}$ are the eigenvalues of M_n . The Christoffel numbers $\{\alpha_k^v\}$ are given by $\frac{|\langle v_k, q_0^v \rangle|^2}{|1 - \exp(-i2\pi v)|^2}$, with the $\{v_k\}$ the normalized eigenfunctions ($\|v_k\| = 1$) of M_n .

4.9 Computational issues

4.9.3 A precise description of the roots

Theorem 4.9.7. Let $\{\mu_k^v; 0 \leq v \leq 1, k = 0, \dots, n\}$ be the roots of q_{n+1}^v ordered such that $Im(\mu_k^v) < Im(\mu_{k+1}^v)$ and suppose that the Christopfel numbers (see (4.3.9) $\{\alpha_k^v; 0 \leq v \leq 1, k = 0, \dots, n\}$ are ordered in the same way. Let the functions $\mu : [-\frac{n+1}{2}, \frac{n+1}{2}]$ and $\alpha : [-\frac{n+1}{2}, \frac{n+1}{2}]$ be given by

$$\mu \left(v + k - \frac{n+1}{2} \right) = \mu_k^v \quad \text{and} \quad \alpha \left(v + k - \frac{n+1}{2} \right) = \alpha_k^v,$$

for $v \in [0, 1]$ and $k = 0, 1, \dots, n$. The function α and μ are continuous and infinitely differentiable functions and μ is the unique solution of the differential equation

$$\frac{D\mu(v)}{2\pi i} = \alpha(v), \quad \mu(0) = 0. \quad (4.9.4)$$

The function $\mu(v)$ is a strictly increasing function with $\mu\left(-\frac{n+1}{2}\right) = -i\infty$ and $\mu\left(\frac{n+1}{2}\right) = i\infty$. The function μ is an odd function and the function α is an even function, that is, $\mu(t) = -\mu(-t)$ and $\alpha(t) = \alpha(-t)$, for $t \in [-\frac{n+1}{2}, \frac{n+1}{2}]$.

Proof. We can decompose $\widehat{M}(v)$, see equation (4.5.2), in

$$\widehat{M}(0.5) + e(v) e_0 e_0^*$$

with $e(v) = \frac{1 + \exp(-i2\pi v)}{2(1 - \exp(-i2\pi v))}$ and e_0 the vector with first element equals 1 and further zeros. Let $\mu_k(v)$ and $V_k(v)$ be an eigenvalue and corresponding -vector of $\widehat{M}(v)$, thus $\widehat{M}(v)V_k(v) = \mu_k(v)V_k(v)$. Then

$$\left(\widehat{M}(0.5) + e(v) e_0 e_0^* - \mu_k(v) \right) V_k(v) = 0.$$

Taking the derivative with respect to v gives

$$\left(\widehat{M}(v) - \mu_k(v) \right) V_k'(v) + (e'(v) e_0 e_0^* - \mu_k'(v)) V_k(v) = 0.$$

Multiplying the l.h.s. with $V_k^*(v)$ and using that $V_k^*(v) \left(\widehat{M}(v) - \mu_k(v) \right)$ vanishes yields that

$$V_k^*(v) (e'(v) e_0 e_0^* - \mu_k'(v)) V_k(v) = 0.$$

Hence

$$\begin{aligned} \mu_k'(v) &= e'(v) \left(\frac{|V_k^*(v) e_0|}{\|V_k(v)\|} \right)^2 \\ &= 2\pi i \alpha_k^v \end{aligned}$$

where we use that, cf. theorem 3.8.2,

$$\alpha_k^v = \left(\frac{|V_k^*(v)e_0|}{\|V_k(v)\|} \right)^2 \left| \frac{1}{1 - \exp(-2\pi vi)} \right|^2$$

$$e'(v) = 2\pi i \left| \frac{1}{1 - \exp(-2\pi vi)} \right|^2$$

Since $q_{n+1}^{-v}(-s) = (-1)^n e^{i2\pi v} q_{n+1}^v(s)$ we obtain that $\mu(v) = -\mu(-v)$ and $\alpha(v) = \alpha(-v)$. This yields in particular that $\mu(0) = 0$. Since the function α is strictly positive we obtain that the function $\mu(v)$ is a increasing function. Moreover, since

$$\sum_{k=0}^n \alpha_k(t) = \frac{1}{2} \frac{1}{1 - \cos(2\pi t)}, \quad (4.9.5)$$

we obtain that $\mu\left(\frac{n+1}{2}\right)$ equals

$$2\pi i \int_0^{\frac{n+1}{2}} \alpha(t) dt = \pi i \int_{-\frac{n+1}{2}}^{\frac{n+1}{2}} \alpha(t) dt = \pi i \int_0^1 \sum_{k=0}^n \alpha_k(t) dt$$

$$= \frac{i\pi}{2} \int_0^1 \frac{1}{1 - \cos(2\pi t)} dt = i\infty.$$

□

4.10 Proof of theorem 4.4.3

Proof. The orthogonality of $\{\psi_{kj}\}$ follows from

$$\begin{aligned} \langle \psi_{kj}, \psi_{nm} \rangle &= \int_0^1 \langle \Psi \hat{\psi}_{kj}, \Psi \hat{\psi}_{nm} \rangle_{v,n} dv \\ &= \int_0^1 e^{-i2\pi(k-n)v} \langle q_j^v, q_m^v \rangle_{v,n} dv \\ &= \delta_{jm} \int_0^1 e^{-i2\pi(k-n)v} dv \\ &= \delta_{jm} \delta_{kn} \end{aligned}$$

with δ_{jm} the Kronecker delta defined by $\delta_{jm} = 1 \iff j = m$. We will now prove identity (4.4.3). It holds that

$$\sum_k e^{-i2\pi kv} \langle f, \psi_{kj} \rangle = \sum_k e^{-i2\pi kv} \int_0^1 \langle \Psi \hat{f}, \hat{q}_j^w \rangle_{w,n} e^{i2\pi kw} dw \quad (4.10.1)$$

Now, because

$$\sum_{j=0}^{n-1} \int_0^1 \left| \langle \Psi \hat{f}, \hat{q}_j^w \rangle_{w,n} \right|^2 dw = \langle f, f \rangle$$

4.10 Proof of theorem 4.4.3

$\langle \Psi \hat{f}, \hat{q}_j^w \rangle_{w,n}$ lies in L^2 and hence the Fourier series (4.10.1) converges (with respect to the L^2 norm)

$$\begin{aligned} \sum_k e^{-i2\pi kv} \langle f, \psi_{kj} \rangle &= \sum_k \int_0^1 e^{-i2\pi kv} \langle \Psi \hat{f}, \hat{q}_j^w \rangle_{w,n} e^{i2\pi kw} dw \\ &= \langle \Psi \hat{f}, \hat{q}_j^v \rangle_{v,n} \end{aligned}$$

and hence identity (4.4.3) holds. To show that the set $\{\psi_{kj}\}$ is complete in L^2 , it is sufficient to show that (Parseval theorem)

$$\sum_{k,j} \langle f, \psi_{kj} \rangle \langle \psi_{jk}, g \rangle = \langle f, g \rangle$$

This follows from

$$\begin{aligned} \sum_{k,j} \langle f, \psi_{kj} \rangle \langle \psi_{jk}, g \rangle &= \sum_{j=0}^{n-1} \int_0^1 \left(\sum_k e^{-i2\pi kv} \langle f, \psi_{kj} \rangle dv \right) \left(\sum_k e^{-i2\pi kv} \langle g, \psi_{kj} \rangle \right)^* dv \\ &= \sum_{j=0}^{n-1} \int_0^1 \langle \Psi \hat{f}, \hat{q}_j^v \rangle_{v,n} \langle \hat{q}_j^v, \Psi \hat{g} \rangle_{v,n} dv \\ &= \int_0^1 \langle \Psi \hat{f}, \Psi \hat{g} \rangle_{v,n} dv \\ &= \langle f, g \rangle. \end{aligned}$$

which means that the set $\{\psi_{kj}\}$ is a complete orthogonal set in L^2 . We will now prove identity (4.4.4). Since the Fourier transform F is a unitary operator, we obtain that

$$\begin{aligned} Ff(s) &= \sum_{j=0}^n \langle f, \psi_{kj} \rangle (F\psi_{kj})(s) \\ &= \sum_{j=0}^n \langle f, \psi_{kj} \rangle e^{-i2\pi kv(s)} \hat{\psi}_j(s) \\ &= \sum_{j=0}^n \langle \Psi \hat{f}, \hat{q}_j^v \rangle_{v(s),n} \hat{\psi}_j(s). \end{aligned}$$

We obtain from identity (4.4.3) that

$$\begin{aligned} \sum_{j=0}^n \langle f, \psi_{kj} \rangle (F\psi_{kj})(s) &= \sum_{j=0}^n \langle f, \psi_{kj} \rangle e^{-i2\pi kv(s)} \hat{\psi}_j(s) \\ &= \sum_{j=0}^n \langle \Psi \hat{f}, \hat{q}_j^v \rangle_{v(s),n} \hat{\psi}_j(s). \end{aligned}$$

Since $e^{-i2\pi v(-s)} = e^{i2\pi v(s)}$, we obtain that the functions ψ_{kj} satisfy $\psi_{-(1+k)j}(t) = \psi_{kj}(-t)$.

We will now prove that the support of the functions $\{\psi_{kj}; k \in \mathbb{N}, j = 0, 1, \dots, n\}$ is $[0, \infty)$.

We know from theorem 4.4.1 that all roots of the equation $e^{-i2\pi v(s)} = e^{-i2\pi v}$, lie on the imaginary axis. Hence $|e^{-i2\pi v(s)}| = 1$ implies that $\Re(s) = 0$. Now, because

$$p_k(s) = \sum_{j=0}^k \frac{(k+j)!}{j!(k-j)!} (-s)^j$$

has alternating coefficients, it is easy to see that $|e^{-i2\pi v(a)}| < 1$, for all positive real a . Since the roots of a polynomial are continuous in the coefficients, $|k_n(a)| < 1$ and $\{|k_n(s)| = 1 \Leftrightarrow \Re(s) = 0\}$, we obtain that $|e^{-i2\pi v(s)}| < 1$ in the right half plane $\{s; \Re(s) > 0\}$. This implies that $e^{-i2\pi v(s)}$ is analytical in the right half plane $\{s; \Re(s) > 0\}$. From this we obtain that for $k \geq 0$, $\hat{\psi}_{kj}$ is analytical in the right half plane $\{s; \Re(s) > 0\}$ and hence ψ_{kj} , $k \geq 0$, has support $[0, \infty)$ (Paley-Wiener theorem, cf. Rudin (1987)). \square

4.11 Detailed proof of the error theorem 4.6.1

Proof. We can split the transform in

$$\hat{f}(s) = \hat{f}_-(s) + \hat{f}_+(s)$$

with

$$\begin{aligned} \hat{f}_-(s) &= \int_{-\infty}^0 e^{-sx} f(x) dx \\ \hat{f}_+(s) &= \int_0^{\infty} e^{-sx} f(x) dx \end{aligned}$$

Below we prove the result for \hat{f}_+ , the proof for \hat{f}_- is similar. The result for \hat{f}_+ reads as:

$$\sqrt{\sum_{k=0}^n \left| \langle \Psi \hat{f}_+, q_k^v \rangle_{vn} - \langle \Psi \hat{f}_+, q_k^v \rangle_v \right|^2} \leq \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}} \sum_{j=0}^{\infty} \left(V_j f^{(1)}(0) + \|V_j f^{(2)}\|_1 \right)$$

with for $x \geq 0$ the $V_j g$ given by

$$V_x g(y) = \begin{cases} 0 & , y < 0 \\ \int_0^1 g(x+y+t) \phi_{n+1}(t) dt & , y \geq 0. \end{cases} \quad (4.11.1)$$

and $f^{(k)}$ denotes the k th derivative of f . We can similar proof that

$$\sqrt{\sum_{k=0}^n \left| \langle \Psi \hat{f}_-, q_k^v \rangle_{vn} - \langle \Psi \hat{f}_-, q_k^v \rangle_v \right|^2} \leq \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}} \sum_{j=-\infty}^1 \left(V_j f^{(1)}(0) + \|V_j f^{(2)}\|_1 \right)$$

4.11 Detailed proof of the error theorem 4.6.1

with for $x < 0$

$$V_x g(y) = \begin{cases} 0 & , y > 0 \\ \int_0^1 g(x+y+t)\phi_{n+1}(t)dt & , y \leq 0. \end{cases}$$

In the remainder of the proof we assume that the support for f is contained in $[0, \infty)$.

Let us first summarize the most important steps needed in the proof, then give the details necessary for the understanding of the individual steps. The proof is structured as follows. **First**, we perform a simple estimation of the quadrature rule, using the properties of the interpolation polynomial P_{n+1} of $\Psi\hat{f}$ in the roots $\{\mu_k^v\}$, which yields

$$\left\langle \Psi\hat{f}, q_k^v \right\rangle_{vn} - \left\langle \Psi\hat{f}, q_k^v \right\rangle_v = - \left\langle \Psi\hat{h}, q_k^v \right\rangle_v, \quad (4.11.2)$$

with the function \hat{h} defined by

$$\hat{h}(s) = \hat{f}(s) - \Psi P_{n+1}(s). \quad (4.11.3)$$

In the **second step** we refine estimate (4.11.2) as

$$- \left\langle \Psi\hat{h}, q_k^v \right\rangle_v = \left\langle \Psi\hat{k}, \Psi\hat{\phi}_{n+1} \right\rangle_v, \quad (4.11.4)$$

with

$$\hat{k}(s) = \hat{h}(s) \widehat{W}_k(s), \quad (4.11.5)$$

where

$$\widehat{W}_k(s) = \frac{q_k^{-v} (-s^{-1})}{\widehat{q}_{n+1}^{-v} (-s^{-1})}. \quad (4.11.6)$$

After showing that \hat{k} is holomorphic in the right half plane Π_+ , the **third step** is to apply the Poisson summation formula (see theorem 4.3.6) to obtain

$$\left\langle \Psi\hat{k}, \Psi\hat{\phi}_{n+1} \right\rangle_v = \sum_{j=0}^{\infty} e^{-i2j\pi v} \langle k(j + \cdot), \phi_{n+1} \rangle, \quad (4.11.7)$$

with

$$k(x) = \frac{1}{2\pi i} \int_{-\varepsilon - i\infty}^{-\varepsilon + i\infty} e^{xs} \widehat{k}(s) ds, \quad (4.11.8)$$

with ε a positive number.

Step 4 is to work out expression (4.11.8) and obtain

$$k(x) = \frac{A_n}{A_{n-1}} \left\langle q_n^v \Psi \widehat{V_x f^{(1)}}, q_k^v \right\rangle_{vn}, \quad (4.11.9)$$

Finally, in **step 5** we work out the error (4.11.2) by substituting expression (4.11.9) into (4.11.7), thus arriving to the final estimate (4.6.1).

Let us now fill in the details needed for the understanding of the individual steps.

1. P_{n+1} is the interpolation polynomial (of degree $n + 1$) of $\Psi \hat{f}$ in the zeros of q_{n+1}^v , the numbers $\{\mu_k^v\}$. Since $P_{n+1} \in \pi_{n+1}$ we obtain that

$$\langle P_{n+1}, q_k^v \rangle_v = \langle P_{n+1}, q_k^v \rangle_{n,v} = \langle \Psi \hat{f}, q_k^v \rangle_{n,v}. \quad (4.11.10)$$

This concludes step 1 of the proof.

2. Let us now refine estimate (4.11.2). For $s \in \{2\pi i(k + v)\}$ we can use relation (4.3.5) (see remark 4.3.1) and obtain

$$\frac{s \hat{\phi}_{n+1}(-s)}{q_{n+1}^{-v}(-s-1)} = -1.$$

Then, by straightforward computation we obtain

$$\begin{aligned} \langle \Psi \hat{h}, q_k^v \rangle_v &= - \left\langle \Psi \left(\frac{s \hat{\phi}_{n+1}(-s)}{q_{n+1}^{-v}(-s-1)} \hat{h} \right), q_k^v \right\rangle_v \\ &= \langle \Psi(\hat{h} \hat{W}_k), \Psi \hat{\phi}_{n+1} \rangle_v, \end{aligned} \quad (4.11.11)$$

with \hat{W}_k defined by (4.11.6). This concludes the second step of the proof.

3. For the third step of the proof we need to show that $\hat{W}_k \hat{h}$ is a proper Laplace transform, satisfying the conditions for the PSF (see theorem 4.3.6).

- (a) Since P_{n+1} interpolates $\Psi \hat{f}$ in $\{\mu_k^v\}$, the function \hat{h} has simple roots in $\{\frac{1}{\mu_k^v}\}$, hence the function \hat{k} is holomorphic in $\{\frac{1}{\mu_k^v}\}$.
- (b) Since and $\frac{1}{q_{n+1}^{-v}(-s-1)} = O(s^{n+1})$ in a neighborhood of 0, $\hat{k}(s) = O(s^{n+1-k})$ in a neighborhood of 0, and hence is holomorphic in a neighborhood of 0.
- (c) Since $\lim_{|s| \rightarrow \infty} |\hat{W}_k(s)| < \infty$ and $\hat{h}(s) = O(\max(s^{-1}, \hat{f}(s)))$ as $|s| \rightarrow \infty$, $\hat{k}(s) = O(\max(s^{-1}, \hat{f}(s)))$ as $|s| \rightarrow \infty$

Therefore, \hat{k} is in $L^2(-i\infty, i\infty)$ and holomorphic in the right half plane $\Pi_+ = \{s : \operatorname{Re}(s) > 0\}$, that is $\hat{k} \in H_{\Pi_+}$. Using the result that a function k is holomorphic in the right half plane, i.e., $\hat{k} \in H_{\Pi_+}$ iff. $k \in \mathcal{L}^2(0, \infty)$ (cf. Rudin (1987), theorem 19.2 of Paley and Wiener), we obtain that the support of k is $[0, \infty)$. We can now apply the PSF and obtain result (4.11.7).

4. We start with the partial fraction expansion of $\hat{W}_k(s^{-1})$,

$$\hat{W}_k(s^{-1}) = \sum_{j=0}^n \frac{q_k^{-v}(-\mu_j^v)}{D q_{n+1}^{-v}(-\mu_j^v)} \frac{1}{\mu_j^v - s}$$

4.11 Detailed proof of the error theorem 4.6.1

From the definition of the Christophel numbers, cf. formula (4.9.1) we obtain

$$\begin{aligned}\widehat{W}_k(s^{-1}) &= c \sum_{j=0}^n \alpha_j^v q_n^v(\mu_j^v) q_k^{-v}(-\mu_j^v) \frac{1}{\mu_j^v - s} \\ &= c \left\langle \frac{q_n^v}{\mu - s}, q_k^v \right\rangle_{v,n}\end{aligned}$$

with

$$c = \frac{A_n}{A_{n-1}} = \frac{1}{2} \frac{1}{\sqrt{4n^2 - 1}}$$

Hence

$$\widehat{W}_k(s) = c \left(\left\langle \frac{1}{\mu} q_n^v, q_k^v \right\rangle_{v,n} + \left\langle \frac{1}{\mu^2 s - \frac{1}{\mu}} q_n^v, q_k^v \right\rangle_{v,n} \right)$$

From this we obtain that

$$\begin{aligned}\langle k(x + \cdot), \phi_{n+1} \rangle &= c \left\langle \frac{1}{\mu} q_n^v, q_k^v \right\rangle_{v,n} \langle h(x + \cdot), \phi_{n+1} \rangle \\ &\quad - c \int_0^\infty \int_0^1 h(x + t + y) \left\langle q_n^v \frac{e^{-\mu^{-1}y}}{u^2}, q_k^v \right\rangle_{v,n} dy \phi_{n+1}(t) dt \\ &= c \left\langle \frac{1}{\mu} q_n^v, q_k^v \right\rangle_{v,n} V_x f(0) - c \int_0^\infty V_x f(y) \left\langle q_n^v \frac{e^{-\mu^{-1}y}}{u^2}, q_k^v \right\rangle_{v,n} dy\end{aligned}$$

with for $x > 0$

$$V_x f(y) = \langle h(x + y + \cdot), \phi_{n+1} \rangle$$

Integration by parts yields

$$\begin{aligned}\langle k(x + \cdot), \phi_{n+1} \rangle &= c \int_0^\infty \frac{\partial}{\partial y} V_x h(y) \left\langle q_n^v \frac{e^{-\mu^{-1}y}}{u}, q_k^v \right\rangle_{v,n} dy \\ &= c \left\langle q_n^v \Psi \widehat{V_x f^{(1)}}, q_k^v \right\rangle_{v,n}\end{aligned}$$

where we use

$$\begin{aligned}\frac{\partial}{\partial y} V_x h(y) &= \left\langle \frac{\partial}{\partial y} (f(x + y + \cdot) - p(x + y + \cdot)), \phi_{n+1} \right\rangle \\ &= \left\langle \frac{\partial}{\partial y} f(x + y + \cdot), \phi_{n+1} \right\rangle\end{aligned}$$

with p the $n+1$ th degree polynomial $F^{-1}(\Psi P_{n+1})$. In the second step we use that $\langle p_n, \phi_{n+1} \rangle = 0$, for all polynomials of degree n . Integration by part again yields

$$\begin{aligned}\langle k(x + \cdot), \phi_{n+1} \rangle &= c V_x f(0) \langle q_n^v, q_k^v \rangle_{v,n} - c \int_0^\infty V_x f^{(2)}(y) \left\langle q_n^v e^{-\mu^{-1}y}, q_k^v \right\rangle_{v,n} dy \\ &= c V_x f^{(1)}(0) \langle q_n^v, q_k^v \rangle_{v,n} - c \left\| V_x f^{(2)}(y) \right\|_1 \sup_y \left\langle q_n^v e^{-\mu^{-1}y}, q_k^v \right\rangle_{v,n}\end{aligned}$$

5. Finally we obtain the desired estimate by the observation that

$$\sqrt{\sum_{k=0}^n \left| \left\langle cV_x f^{(1)}(0) \langle q_n^v, q_k^v \rangle_{v,n} - c \|V_x f^{(2)}\|_1 \sup_y \langle q_n^v e^{-\mu^{-1}y}, q_k^v \rangle_{v,n} \right\rangle_{vn} \right|^2}$$

is bounded by

$$c \left(V_x f^{(1)}(0) + \left\| q_n^v e^{-\mu^{-1}y} \right\|_{vn} \left\| V_x f^{(2)} \right\|_1 \right)$$

and

$$\left\| q_n^v e^{-\mu^{-1}y} \right\|_{vn} = 1$$

□

Chapter 5

Pricing guaranteed return rate products and discretely sampled Asian options

5.1 Introduction

Guaranteed return rate products can be regarded as generalized discretely sampled Asian options. Therefore, pricing of guaranteed return rate products is usually done by some kind of generalized algorithm for valuing Asian options. Besides simulation techniques, there are three main pricing approaches in the literature, approximations (Lévy's, Vorst, Curran), analytic calculations with Laplace transforms, concentrating exclusively on continuous averages, and partial differential equation techniques applicable both to discrete and continuous averages.

There are only a few articles in the literature applying the Laplace transform technique for pricing Asian options. These articles all consider continuous averages, that is, continuous sampling, instead of the in practice prevalent discrete sampling. The first in this line of research is the seminal work of Geman and Yor (1993) who determine the Laplace transform of the Asian option price based on Bessel processes theory. However, these Laplace transforms are available only if the risk-neutral drift is not less than half the squared volatility, which substantially restricts its financial applicability (see Carr and Schröder (2001)). Carr and Schröder (2001) repair this restriction; for the inversion of the Laplace transform they use the algorithm of Schröder (1999). Further, to our knowledge, all the work concentrating on Laplace transform techniques relies on the Laplace transform derived by Geman and Yor, using different inversion techniques (see Dufresne (2000), Fu, Madan and Wang(1998), Schröder (1999)). Although the Laplace transform technique is referred to in the literature as an analytic approach, it is also an approximation technique if numerical inversion of the Laplace transforms is used. In the case of the inversion algorithm of den Iseger (2006a and

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options (2006b), the precision of this approximation is almost the same as the precision of a computer; this property being known as "machine precision". His algorithm computes the coefficients of the piecewise Legendre polynomial expansion of the original function if specific Laplace transform function values are known, and, conversely, obtains Laplace transform function values if the expansion coefficients of the original function are known. It can also compute the expansion coefficients if specific function values are known, and it can obtain function values if the expansion coefficients are known.

Although we place our algorithm in the literature of option pricing with Laplace transform techniques, it deflects from it in an important aspect: our analysis tackles discrete averages instead of continuous averages. In case of discrete averages, the underlying is a sum of random variables. However, the problem with guaranteed return rate products precisely is that the distribution of this sum of random variables is not a convolution because the different terms of this sum are correlated. In the present article we deduce a simple recursive algorithm which calculates the distribution of this sum (a discrete average) as an iterated convolution. The innovation of the present article lies in the simplicity of the recursion algorithm used for determining the price function for a general model (the asset price process is required to have independent returns per period, e.g. Lévy processes). Its high accuracy is due to the newly developed transformation technique in den Iseger (2006a, 2006b). However, the flexibility of the algorithm lies not only in the unrestricted choice of parameters. We can also obtain the sensitivity measures, called the Greeks, of the price w.r.t. changes in the market variables using a Girsanov type drift adjustment. We compute the delta, gamma, vega, rho, and convexity of the product for both lognormal and jump-diffusion models. The algorithm also allows different parameter input for every year, such as varying interest rate or volatility (e.g. volatility skew or smile). Its running time is 0.04 seconds (for 30 periods) on a standard computer. Furthermore, the algorithm computes the option price for all strikes at once.

Finally, the paper is organized as follows. Section 5.2 introduces the pricing concept, Section 5.3 presents the recursion algorithm for the general model with some numerical results, and Section 5.4 shows how the Greeks can be computed first for the case of lognormal underlying (subsection 5.4.1), then for a jump-diffusion model (subsection 5.4.2). In subsection 5.4.3 we discuss the Greeks for discretely sampled Asian options and list some numerical results for the price and Greeks of Asian options.

This chapter is based on the paper den Iseger and Oldenkamp (2006a).

5.2 Discretely sampled Asian options and guaranteed rate products: Pricing concept

In this section we introduce the price functions for discretely sampled Asian options and a guaranteed rate product, pointing out that the latter is what we call a ‘generalized average price option’. Therefore an algorithm which calculates the price of a guaranteed rate product will also yield the price of a discretely sampled Asian option through the necessary modifications. Consider first the case of discretely sampled Asian options. The payoff of Asian options depends on the average of the price of the underlying over the option’s lifetime. The data used in the calculation of the average can be continuously or discretely sampled. In practice, discrete sampling is used, for instance closing prices. With the notation

- S_t the price of a unit of the underlying at time t ,
- t_i $i = 1, \dots, n$ the sampling moments,
- T maturity, $T = t_n$,
- K strike level,
- r risk free rate,

the price of an Asian put (call) option is given by

$$e^{-rT} E_Q \left(K - \frac{1}{n} \sum_{i=1}^n S_{t_i} \right)^+ \quad (\text{or } e^{-rT} E_Q \left(\frac{1}{n} \sum_{i=1}^n S_{t_i} - K \right)^+, \text{ respectively}), \quad (5.2.1)$$

where E_Q stands for the expectation w.r.t. the risk-neutral measure. On the other hand, the price of a guaranteed rate product can be written in the form of a generalized average price option (APO) price, that is, the arithmetic average is weighted with the net investment premiums. In order to determine the price of a guaranteed rate product we proceed as follows (using the notation and model description of Schrage and Pelsser (2003)).

- p_i investment premium paid at time t_i , $i = 0, \dots, n - 1$,
- K is guaranteed amount at expiry, the strike of the option.

In practice, at time t_i , the policyholder purchases $\frac{p_i}{S_{t_i}}$ units, each unit having value S_T at expiry. Hence, the total fund value at T is $\sum_{i=0}^{n-1} p_i \frac{S_T}{S_{t_i}}$. On the other hand, a

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options
 minimum of K is paid out at expiry, yielding a payoff of

$$\max \left(\sum_{i=0}^{n-1} p_i \frac{\mathbf{S}_T}{\mathbf{S}_{t_i}}, K \right) = \sum_{i=0}^{n-1} p_i \frac{\mathbf{S}_T}{\mathbf{S}_{t_i}} + \left(K - \sum_{i=0}^{n-1} p_i \frac{\mathbf{S}_T}{\mathbf{S}_{t_i}} \right)^+. \quad (5.2.2)$$

The first term in relation (5.2.2) is just the payoff from the investment while the second term contains the guarantee. The price of this guarantee is the expected value w.r.t. the risk-neutral measure of this second term, which is precisely the price of a put option on

$$\mathbf{A}_T := \sum_{i=0}^{n-1} p_i \frac{\mathbf{S}_T}{\mathbf{S}_{t_i}}, \quad (5.2.3)$$

with strike price K , formally,

$$G := G_T(K) := e^{-rT} \mathbf{E}_Q \left(K - \sum_{i=0}^{n-1} p_i \frac{\mathbf{S}_T}{\mathbf{S}_{t_i}} \right)^+. \quad (5.2.4)$$

The aim of the next section is to calculate G . In order to achieve this goal we construct a recursion algorithm, making use of the inverse-transformation algorithm of den Iseger (2006a) and (2006b). The advantage of this method is that it is very fast and almost exact up to machine precision. Furthermore, it yields the prices for all strikes at once.

5.3 The recursion algorithm for determining G

The aim of this section is to determine function G of the strike K , the price of the guarantee as a function of the guaranteed level. To that end we will only assume that the returns of the asset process are independent per period, but for the moment we will not assume a specific distribution. The price G is in fact the price of a put option which involves the distribution f_a of $\mathbf{a}_T := \ln(\mathbf{A}_T)$ (see (5.2.3)). An algorithm for the calculation of the price of a put option, given either the density of the underlying or the Laplace transform of the density, is given in appendix 5.6. We will proceed here with determining the distribution f_a . Observe now that the terms of the sum (5.2.3), the random variables $\frac{\mathbf{S}_T}{\mathbf{S}_{t_i}}$, $i = 0, \dots, n-1$ are correlated, hence it is not straightforward to determine the Laplace transform of f_a . We aim to write relation (5.2.3) in a more tractable form, making use of the idea that the value of the portfolio at time $t+1$ equals the value of the portfolio at time t plus the premiums paid at t , times the return in that period. Let us denote the returns on \mathbf{S} in interval $[t_{i-1}, t_i]$ as \mathbf{X}_i , $i = 1, \dots, n$, and assume that they are independent. Let $f_{\mathbf{X}_i}$ denote the distribution of $\ln(\mathbf{X}_i)$ and \mathbf{A}_k be defined

5.3 The recursion algorithm for determining G
by generalizing (5.2.3) such that

$$\mathbf{A}_k := \sum_{i=0}^{k-1} p_i \frac{\mathbf{S}_{t_k}}{\mathbf{S}_{t_i}}, \quad k = 1, \dots, n-1. \quad (5.3.1)$$

In this way, since the value of the portfolio at time t and the return over the period $(t, t+1)$ are independent, we have

$$\mathbf{A}_{k+1} = \mathbf{X}_{k+1}(p_k + \mathbf{A}_k), \quad k = 1, \dots, n-1 \quad \text{with} \quad \mathbf{A}_1 = p_0 \mathbf{X}_1, \quad (5.3.2)$$

such that \mathbf{X}_{k+1} and \mathbf{A}_k are independent. Let us use the notation

$$\mathbf{Z}_k := \ln(\mathbf{A}_k), \quad k = 1, \dots, n, \quad (5.3.3)$$

Relation (5.3.2) can be written in the equivalent form:

$$\mathbf{Z}_{k+1} = \ln(\mathbf{X}_{k+1}) + \ln(p_k + \exp(\mathbf{Z}_k)), \quad \mathbf{Z}_1 = \ln(p_0) + \ln \mathbf{X}_1. \quad (5.3.4)$$

Furthermore,

$$\mathbf{P}\{\ln(p_k + \exp(\mathbf{Z}_k)) \leq z\} = \mathbf{P}\{\exp(\mathbf{Z}_k) \leq \exp(z) - p_k\} = \mathbf{P}\{\mathbf{Z}_k \leq \ln(\exp(z) - p_k)\}.$$

Denote by f_k and g_k the densities of \mathbf{Z}_k and $\ln(p_k + \exp(\mathbf{Z}_k))$, respectively. Then,

$$g_k(z) = f_k(\ln(\exp(z) - p_k)) \frac{\exp(z)}{\exp(z) - p_k}. \quad (5.3.5)$$

Hence, the distribution of \mathbf{Z}_{k+1} from (5.3.4) is given by

$$f_{k+1}(z) = (f_{\mathbf{X}_{k+1}} * g_k)(z). \quad (5.3.6)$$

The calculation of the distribution of $\mathbf{Z}_n = \mathbf{a}_T$ followed by the calculation of G , reduces to the following recursion algorithm. The transformation technique in den Iseger (2006a) and (2006b) can accurately approximate the Laplace transforms of functions, and inversely, it can approximate very accurately a function given its Laplace transform in a set of specific points. Given the piecewise polynomial g_k we can compute a piecewise polynomial approximation for f_{k+1} , next we compute a piecewise polynomial for approximation for g_{k+1} , and so on.

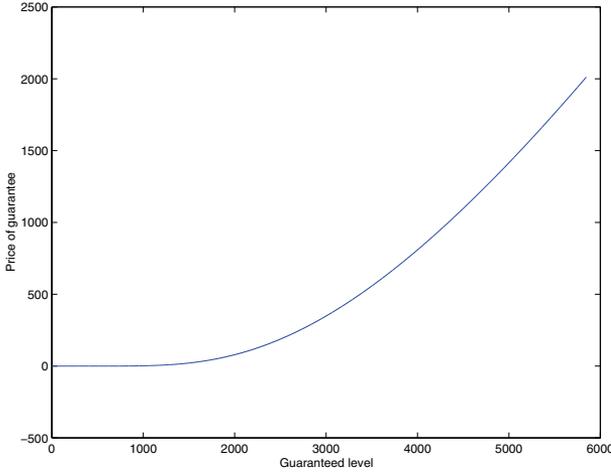


Figure 5.1: The evolution of the price of a 30-year guarantee as the guaranteed level changes.

The parameters for figure 5.1 are $r = 3$, $\sigma = 20$, $q = 0$, $p_0 = \dots = p_{29} = 100$.

Algorithm 5.3.1

- Step 1. $k = 1$; $f_1(x) := f_{X_1}(x + \ln(p_0))$;
- Step 2. $g_k(z) = f_k(\ln(\exp(z) - p_k)) \frac{\exp(z)}{\exp(z) - p_k}$; (see (5.3.5))
- Step 3. $\widehat{g}_k = \mathcal{L}g_k$; (according to algorithm 4.5.5.2)
- Step 4. $\widehat{f}_{k+1}(s) = \widehat{f}_{X_{k+1}}(s)\widehat{g}_k(s)$;
- Step 5. $f_{k+1} = \mathcal{L}^{-1}\widehat{f}_{k+1}$; (according to algorithm 4.5.5.1)
- Step 6. $k = k + 1$, if $k < n$ then go to Step 2;
- Step 7. Compute the option price according to appendix 5.6;

Note that $\mathcal{L}f$ stands for the Laplace transform and $\mathcal{L}^{-1}f$ for the inverse Laplace transform of f . In Appendix 5.5 we discuss the details of this algorithm. The result of this recursion algorithm is a very accurate approximation of the price function not only in one single point but on a whole interval, i.e. a range of strike prices (guarantee levels). An example for the price as a function of the strike is given in Figure 5.1. The running time for 30 periods is 0.04 seconds on a standard computer (pentium 4, 2.4 Ghz, 512 MB memory PC). In order to illustrate the accuracy and stability of the algorithm we demonstrate that our results converge to the price of continuously sampled average options as we increase the number of periods during a 1-year horizon (that is, decreasing the time steps). The numerical example presented in Table 1 is taken from Linetsky

5.3 The recursion algorithm for determining G

(2004), Table 3 (page 865), with parameters $r = 0.05, q = 0, \sigma = 0.5, K = 2, T = 1, S_0 = 2$. As Linetsky also remarks, this set of parameters “has been used as test cases for various numerical methods by Eydeland and Geman (1995), Fu et al. (1998), Shaw (2002), Craddock et al. (2000), Dufresne (2000), and Vecer (2001, 2002)”. Linetsky obtained the continuously sampled Asian call option price 0.2464156905 with eigenfunction expansion, accurate up to 10 decimals. We first computed the put option, then used the put-call parity relationship to obtain the call price. The number of periods are repeatedly increased, then we used an extrapolation technique* and obtained an up to 10 decimals accurate approximation, see Table 2. The error is the difference between the price of continuously sampled and the extrapolated prices of discretely sampled Asian options. The other examples presented by Linetsky (2004) are discussed in section 5.4.3.

# periods	discretely sampled APO	CPU (sec.)
1	0.4358520842573392	0.0006
2	0.3419151899684278	0.0033
4	0.2943433244077809	0.0093
8	0.2704319876815563	0.0132
16	0.2584391354532633	0.0224
32	0.2524316066500627	0.0410
64	0.2494247503198864	0.0742
128	0.2479205029931590	0.1410
256	0.2471681683087853	0.2960
512	0.2467919473931600	0.5470
1024	0.2466038234034073	1.0780
2048	0.2465097579466856	2.1410
4096	0.2464627240910414	4.2810
8192	0.2464392061763754	8.8280
16384	0.2464274450896750	18.8750

Table 5.1: Price and computation time for discretely sampled Asian options.

Observation 5.3.1. *Although the algorithm is set up to determine the value of the guaranteed rate product at t_0 , we can also obtain the value of a contract that has started in the past, by creating an artificial first premium and setting it equal to the realized value of the portfolio up to time t , $p_t := \hat{p}_t$, where*

$$\hat{p}_t := \sum_{k=0}^{N(t)} p_k \frac{S_t}{S_{t_k}} =: \mathbf{A}_t, \tag{5.3.7}$$

* Denoting with $G(2^m, n)$ the n^{th} order extrapolated price with # periods 2^m , then $G(2^m, n+1) = \frac{1}{2^{m-1}}(2^n G(2^m, n) - G(2^{m-1}, n))$.

# periods	difference 1	difference 2	difference 3	price with extrapolation 3
32				
64	2.203494e-006			
128	5.651719e-007	1.906468e-008		
256	1.431238e-007	2.441102e-009	6.630455e-011	0.24641569056630455

Table 5.2: Prices of APO's with extrapolations of errors of order one, two and three.

with $\mathbf{N}(t) = \sum_{k=1}^n 1_{\{t_k \leq t\}}$. Moreover, the value of \mathbf{A}_n given the history up to t is given by

$$\mathbf{A}_{t,n} := \sum_{k=\mathbf{N}(t)+1}^n p_k \frac{\mathbf{S}_{t_n}}{\mathbf{S}_{t_k}} + \hat{p}_t \frac{\mathbf{S}_{t_n}}{\mathbf{S}_t}. \quad (5.3.8)$$

This formula is equivalent to (5.2.3).

Remark 5.3.2. Note that since the algorithm is recursive over the periods $1, \dots, T$ all parameters (namely r, q, σ) can vary from period to period. It is only for the sake of transparency of presentation we used constant parameters. The same observation holds in the next section for the calculation of the Greeks. Furthermore, for the computation of the price it is only required that the underlying asset price process should have independent returns per period. This implies that Lévy processes can be considered (see e.g. Carr et al. (2003)).

5.4 Greeks

The recursion algorithm developed in Section 5.3 need not solely be used for valuation purposes but it can also be used to determine the sensitivity of the price w.r.t. the underlying (delta), the interest rate (rho), the volatility (vega) and the second order sensitivities, gamma and the convexity. We present a Girsanov type drift adjustment technique in order to determine the Greeks. This section demonstrates how these sensitivities can be determined with the recursion algorithm, first for the case when the underlying is modelled by a lognormal process, then by a jump-diffusion process.

5.4.1 Greeks in the lognormal model

The main assumption of this subsection is that \mathbf{S}_t is lognormally distributed, such that

$$\mathbf{S}_t = \mathbf{S}_0 \exp \left((r - q - \frac{1}{2}\sigma^2)t + \sigma \mathbf{W}_t \right), \quad (5.4.1)$$

5.4 Greeks

where \mathbf{W}_t is a Brownian motion at time t , r is the risk free rate, q is the dividend rate, and σ is the volatility. The random variables $\ln(\mathbf{X}_k)$, $k = 1, \dots, n$, are independent and normally distributed with mean $r - q - \frac{\sigma^2}{2}$ and volatility σ , and, as before, we denote their corresponding normal probability density function by f_{X_k} .

Let us first begin with introducing a formula which plays a key role in deducing the delta, gamma, vega, rho and convexity. We use the following notation

$$\mathbf{R}_{kn} := \frac{\mathbf{S}_{t_n}}{\mathbf{S}_{t_k}} = \prod_{i=k+1}^n \mathbf{X}_i. \quad (5.4.2)$$

Then, from the property of lognormal distributions it follows that \mathbf{R}_{kn} is also lognormal

$$\mathbf{R}_{kn} = \exp \left((r - q - \frac{1}{2}\sigma^2)(t_n - t_k) + \sigma \mathbf{W}_{(t_k, t_n)} \right), \quad (5.4.3)$$

and \mathbf{A}_n defined by relation (5.3.1) becomes

$$\mathbf{A}_n(\bar{p}) := \sum_{k=0}^{n-1} p_k \mathbf{R}_{kn}, \quad (5.4.4)$$

with $\bar{p} := (p_0, p_1, \dots, p_{n-1})$. Using the notation

$$\Phi(\mathbf{A}_n(\bar{p})) := \exp(-rt_n)(K - \mathbf{A}_n(\bar{p}))^+,$$

we have that $G(\bar{p}) = E_Q \Phi(\mathbf{A}_n(\bar{p}))$, yielding

$$\frac{\partial G(\bar{p})}{\partial p_k} = E_Q \frac{\partial \Phi(\mathbf{A}_n(\bar{p}))}{\partial \mathbf{A}_n(\bar{p})} \frac{\partial \mathbf{A}_n(\bar{p})}{\partial p_k}. \quad (5.4.5)$$

From (5.4.4) follows directly that $\partial \mathbf{A}_n(\bar{p}) / \partial p_k = \mathbf{R}_{kn}$, hence the above expression equals

$$\begin{aligned} & E_Q \frac{\partial \Phi(\mathbf{A}_n(\bar{p}))}{\partial \mathbf{A}_n(\bar{p})} \exp \left((r - q - \frac{1}{2}\sigma^2)(t_n - t_k) + \sigma \mathbf{W}_{(t_k, t_n)} \right) \\ &= \exp((r - q)(t_n - t_k)) E_Q \frac{\partial \Phi(\mathbf{A}_n(\bar{p}))}{\partial \mathbf{A}_n(\bar{p})} \exp \left(\sigma \mathbf{W}_{(t_k, t_n)} - \frac{1}{2}\sigma^2(t_n - t_k) \right). \end{aligned} \quad (5.4.6)$$

The term $\exp \left(\sigma \mathbf{W}_{(t_k, t_n)} - \frac{1}{2}\sigma^2(t_n - t_k) \right)$ is exactly the Radon-Nikodym derivative, which defines a new measure under which $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t 1_{\{t \geq t_k\}}$ is a Brownian motion. Thus this Radon-Nikodym derivative realizes a drift adjustment of the Brownian motion of size σ . Hence

$$\frac{\partial G(\bar{p})}{\partial p_k} = \exp((r - q)(t_n - t_k)) E \frac{\partial \Phi(\widehat{\mathbf{A}}_n^k(\bar{p}))}{\partial \widehat{\mathbf{A}}_n^k(\bar{p})}, \quad (5.4.7)$$

with

$$\widehat{\mathbf{A}}_n^k(\bar{p}) := \sum_{j=0}^{n-1} p_j \widehat{\mathbf{R}}_{jn}, \quad (5.4.8)$$

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options and $\widehat{\mathbf{R}}_{jn}^k$ is the original (unadjusted) return process up to t_k and adjusted with σ after time t_k , that is,

$$d\widehat{\mathbf{R}}_t = \begin{cases} (r - q + \sigma^2)\widehat{\mathbf{R}}_t dt + \sigma\widehat{\mathbf{R}}_t d\widehat{\mathbf{W}}_t, & \text{if } t \geq t_k, \\ (r - q)\widehat{\mathbf{R}}_t dt + \sigma\widehat{\mathbf{R}}_t d\widehat{\mathbf{W}}_t, & \text{if } t < t_k. \end{cases} \quad (5.4.9)$$

Remark 5.4.1. Various authors, including Broadie and Glasserman (1996), Glasserman and Zhao (1999), Glasserman (2004) and Piterbarg (2003), used a similar result as formula (5.4.5) for the calculation of Greeks by Monte Carlo simulation. The advantage of formula (5.4.7) is, as we will see in the next section, that we can incorporate the computation of the Greeks in the recursion algorithm of Section 5.3.

Delta hedging

The delta of an option is the sensitivity of its price w.r.t. to the underlying, hence we need to evaluate the expression $\partial E_Q \Phi(\mathbf{A}_n(\bar{p})) / \partial \mathbf{S}_0$. At all times t , we can apply Observation 5.3.1 and regard p_t as the realized value of the portfolio up to time t^- . The total number of units bought up to time t is $\frac{\mathbf{A}_t}{\mathbf{S}_t}$, according to definition (5.3.7) of \mathbf{A}_t . Accordingly, the accumulated wealth up to time t is the number of units times their price \mathbf{S}_t at time t , that is, $\mathbf{A}_t = \frac{\mathbf{A}_t}{\mathbf{S}_t} \mathbf{S}_t$. Now, fixing the number of units, the overall sensitivity to changes in the price \mathbf{S}_t is

$$\frac{\partial \mathbf{A}_t}{\partial \mathbf{S}_t} = \frac{\mathbf{A}_t^-}{\mathbf{S}_t^-}. \quad (5.4.10)$$

As we already mentioned in Observation 5.3.1, we relabel the current time to t_0 (that is $\widehat{p}_0 = \mathbf{A}_{t_0}$), hence we have

$$\Delta_0 = \frac{\partial G(\bar{p})}{\partial p_0} \frac{\mathbf{A}_{t_0}^-}{\mathbf{S}_{t_0}^-}. \quad (5.4.11)$$

What remains to be determined is $\frac{\partial G(\bar{p})}{\partial p_0}$, that is, we need to evaluate the second term of the r.h.s. of relation (5.4.7), that is,

$$E \frac{\partial \Phi(\widehat{\mathbf{A}}_n^k)}{\partial \widehat{\mathbf{A}}_n^k} = -\exp(-rt_n) E 1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} = -\exp(-rt_n) P\{\widehat{\mathbf{Z}}_n \leq \ln(K)\},$$

where $\widehat{\mathbf{Z}}_n := \ln(\widehat{\mathbf{A}}_n^k)$. The distribution of $\widehat{\mathbf{Z}}_n$ can be obtained with steps 1 to 5 of the recursion algorithm. In conclusion,

$$\Delta_0 = -\exp(-qt_n) P\{\widehat{\mathbf{Z}}_n \leq \ln(K)\} \frac{\mathbf{A}_{t_0}^-}{\mathbf{S}_{t_0}^-}.$$

5.4 Greeks

Remark 5.4.2. In case of a vanilla option we have $p_0 = S_0$, $\widehat{\mathbf{A}}_n^k = p_0 \widehat{\mathbf{R}}_{0n} = S_0 \widehat{\mathbf{R}}_{0n}$, and $\frac{\mathbf{A}_{t_0^-}}{\mathbf{S}_{t_0^-}} = \frac{p_0}{S_0} = 1$, hence

$$\Delta = -\exp(-qt_n) N\left(\frac{\ln\left(\frac{K}{S_0}\right) - (r - q + \frac{1}{2}\sigma^2)t_n}{\sigma\sqrt{t_n}}\right) = \exp(-qt_n) (N(d_1) - 1),$$

which, with N the cumulative normal distribution function and

$$d_1 := -\frac{\ln\left(\frac{K}{S_0}\right) - (r - q + \frac{1}{2}\sigma^2)t_n}{\sigma\sqrt{t_n}} = \frac{\ln\left(\frac{S_0}{K}\right) + (r - q + \frac{1}{2}\sigma^2)t_n}{\sigma\sqrt{t_n}}, \quad (5.4.12)$$

is indeed the delta of a plain vanilla put option.

Gamma hedging

The gamma of an option is the sensitivity of the delta w.r.t. the underlying, that is,

$$\Gamma = \frac{\partial}{\partial \mathbf{S}_0} \left(\frac{\partial G(\bar{p})}{\partial p_0} \frac{\mathbf{A}_{t_0^-}}{\mathbf{S}_{t_0^-}} \right) = \left(\frac{\mathbf{A}_{t_0^-}}{\mathbf{S}_{t_0^-}} \right)^2 \mathbf{E}_Q \frac{\partial^2 \Phi(\mathbf{A}_n)}{\partial \mathbf{A}_n^2} \mathbf{R}_{0n} \mathbf{R}_{0n}.$$

Since \mathbf{R}_{0n} figures twice in the expression above, we apply twice a Radon-Nikodym derivative as in (5.4.7), each corresponding to a shift σ , that is, $\exp(2\sigma \mathbf{W}_{(t_0, t_n)} - 2\sigma^2(t_n - t_0))$ is the Radon-Nikodym derivative corresponding to the shift 2σ . Then, Γ is given by

$$\exp(2(r - q)(t_n - t_0)) \exp(\sigma^2(t_n - t_0)) \mathbf{E} \frac{\partial^2 \Phi(\widetilde{\mathbf{A}}_n)}{\partial \widetilde{\mathbf{A}}_n^2}, \quad (5.4.13)$$

where

$$\widetilde{\mathbf{A}}_n := \sum_{k=0}^{n-1} p_k \widetilde{\mathbf{R}}_{kn}, \quad (5.4.14)$$

and

$$d\widetilde{\mathbf{R}}_t = (r - q + 2\sigma^2)\widetilde{\mathbf{R}}_t dt + \sigma\widetilde{\mathbf{R}}_t d\widetilde{\mathbf{W}}_t, \text{ if } t \geq t_0. \quad (5.4.15)$$

Furthermore,

$$\mathbf{E} \frac{\partial^2 \Phi(\widetilde{\mathbf{A}}_n)}{\partial \widetilde{\mathbf{A}}_n^2} = \exp(-rt_n) f_{\widetilde{\mathbf{A}}_n}(K),$$

where $f_{\widetilde{\mathbf{A}}_n}(K)$ is the probability density of $\widetilde{\mathbf{A}}_n$ in K , which we can obtain again with the recursion algorithm. Hence,

$$\Gamma = \exp(2(r - q)t_n) \exp(\sigma^2(t_n - t_0)) \exp(-rt_n) f_{\widetilde{\mathbf{A}}_n}(K) \left(\frac{\mathbf{A}_{t_0^-}}{\mathbf{S}_{t_0^-}} \right)^2. \quad (5.4.16)$$

Remark 5.4.3. For a plain vanilla put option we can use the same line of reasoning as in Remark 5.4.2, setting $t_0 = 0$,

$$f_{\tilde{\mathbf{A}}_n}(K) = \frac{1}{K} f_{\tilde{\mathbf{R}}_n} \left(\ln \left(\frac{K}{S_0} \right) \right) = \frac{1}{K} \frac{1}{\sigma \sqrt{t_n}} N' \left(\frac{\ln \left(\frac{K}{S_0} \right) - (r - q + \frac{3}{2} \sigma^2) t_n}{\sigma \sqrt{t_n}} \right). \quad (5.4.17)$$

Furthermore, writing out the density N' we obtain that

$$N' \left(\frac{\ln \left(\frac{K}{S_0} \right) - (r - q + \frac{3}{2} \sigma^2) t_n}{\sigma \sqrt{t_n}} \right) = N'(-d_1) \frac{K}{S_0} \exp(-(r - q) t_n) \exp(-\sigma^2 t_n),$$

with d_1 given by (5.4.12). Hence, substituting into (5.4.16) we obtain that

$$\Gamma_0 = \frac{\exp(-qt_n)}{S_0 \sigma \sqrt{t_n}} N'(-d_1).$$

Rho and convexity hedging

Rho (ρ) is the sensitivity (derivative) w.r.t. the risk free interest rate r . The ‘convexity’ C is the second derivative.

We get

$$\rho = \exp(-rt_n) E_Q \left(-1_{\{\mathbf{A}_n \leq K\}} \frac{\partial \mathbf{A}_n}{\partial r} \right) - t_n G_n(K),$$

and

$$\begin{aligned} C &= \exp(-rt_n) \left(E_Q \left(\delta_K(\mathbf{A}_n) \left(\frac{\partial \mathbf{A}_n}{\partial r} \right)^2 \right) - E_Q \left(1_{\{\mathbf{A}_n \leq K\}} \frac{\partial^2 \mathbf{A}_n}{\partial r^2} \right) \right) \\ &+ \exp(-rt_n) 2t_n E_Q \left(1_{\{\mathbf{A}_n \leq K\}} \frac{\partial \mathbf{A}_n}{\partial r} \right) + t_n^2 G_n(K), \end{aligned}$$

with δ_K the so-called Dirac function in the point K . Since $\mathbf{A}_n = \sum_{k=0}^{n-1} p_k \mathbf{R}_{kn}$ and

$$\frac{\partial \mathbf{R}_{kn}}{\partial r} = (t_n - t_k) \mathbf{R}_{kn}, \quad \text{and} \quad \frac{\partial^2 \mathbf{R}_{kn}}{\partial r^2} = (t_n - t_k)^2 \mathbf{R}_{kn}$$

it follows that $\frac{\partial \mathbf{A}_n}{\partial r} = \sum_{k=0}^{n-1} \hat{p}_k \mathbf{R}_{kn}$ with $\hat{p}_k = p_k(t_n - t_k)$ and $\frac{\partial^2 \mathbf{A}_n}{\partial r^2} = \sum_{k=0}^{n-1} \hat{\hat{p}}_k \mathbf{R}_{kn}$ with $\hat{\hat{p}}_k = p_k(t_n - t_k)^2$. Therefore, we obtain that

$$\rho = -\exp(-rt_n) \sum_{k=0}^{n-1} \hat{p}_k E \left(1_{\{\mathbf{A}_n \leq K\}} \mathbf{R}_{kn} \right) - t_n G_n(K),$$

where $\exp(-(r - q)(t_n - t_k)) \mathbf{R}_{kn}$ is exactly the Radon-Nikodym derivative which makes a drift adjustment with σ of the Brownian motion \mathbf{W}_t , that is

$$\rho = -\exp(-rt_n) \sum_{k=0}^{n-1} \hat{p}_k \exp((r - q)(t_n - t_k)) \mathbf{P}\{\hat{\mathbf{A}}_n^k \leq K\} - t_n G_n(K), \quad (5.4.18)$$

5.4 Greeks

with $\widehat{\mathbf{A}}_n^k$ given by

$$\widehat{\mathbf{A}}_n^k = \sum_{j=0}^{n-1} p_j \widehat{\mathbf{R}}_{jn}^k, \quad (5.4.19)$$

with $\widehat{\mathbf{R}}_i^k$ given by the process (5.4.9) for $k = 0, \dots, n-1$. The derivation of the formulae for the convexity C proceeds along the same lines.

Remark 5.4.4. In order to compute the rho (relation (5.4.18)) one can proceed as follows. Let

$$h_k = \sum_{j=0}^{k-1} \widehat{p}_j \exp((r-q)(t_n - t_j)) f_{\widehat{\mathbf{A}}_k^j}.$$

We aim to compute the density h_n . Let g_k denote the distribution of \mathbf{A}_k . Furthermore, let Th and $\widehat{T}h$ be defined as the density of $\ln(\exp(\mathbf{Z}_h) + p) + \ln \mathbf{R}$ and $\ln(\exp(\mathbf{Z}_h) + p) + \ln \widehat{\mathbf{R}}$, respectively, where \mathbf{Z}_h denotes a random variable with density h . Then the recursion algorithm 3.1 changes such that the following recursion is incorporated:

1. $h_1 = \widehat{p}_0 \exp((r-q)(t_1 - t_0)) f_{\ln \widehat{\mathbf{R}}}$, $g_1 = f_{\ln \mathbf{R}}$,
2. $h_{k+1} = \exp((r-q)(t_{k+1} - t_k)) \widehat{T}(h_k + \widehat{p}_k g_k)$ and $g_{k+1} = Tg_k$, for $k = 1, \dots, n-1$.

Remark 5.4.5. For the convexity we need to compute

$$\mathbf{E}_Q \left(\delta_K(\mathbf{A}_n) \left(\frac{\partial \mathbf{A}_n}{\partial r} \right)^2 \right) = \sum_{i,j=1}^{n-1} \widehat{p}_i \widehat{p}_j e^{(r-q)(2t_n - t_i - t_j)} e^{\sigma^2(t_n - (t_i \vee t_j))} \mathbf{E}_Q \left(\delta_K(\widetilde{\mathbf{A}}_n^{ij}) \right)$$

where $t_i \vee t_j = \max(t_i, t_j)$,

$$\widetilde{\mathbf{A}}_n^{ij} = \sum_{m=1}^{n-1} p_m \widetilde{\mathbf{R}}_{mn}^{ij} \quad (5.4.20)$$

and

$$\frac{d\widetilde{\mathbf{R}}_t^{ij}}{\widetilde{\mathbf{R}}_t^{ij}} = \begin{cases} (r-q)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t \leq \min\{t_i, t_j\}, \\ (r-q + \sigma^2)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t_i \leq t \leq t_j \text{ or } t_j \leq t \leq t_i, \\ (r-q + 2\sigma^2)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t > \max\{t_i, t_j\}. \end{cases} \quad (5.4.21)$$

We aim to compute the density

$$k_m = \sum_{i,j=1}^{m-1} \widehat{p}_i \widehat{p}_j e^{(r-q)(2t_m - t_i - t_j)} e^{\sigma^2(t_m - (t_i \vee t_j))} f_{\widetilde{\mathbf{A}}_m^{ij}}$$

for $m = n$. Let h_k , g_k and the operators T and \widehat{T} be defined as in remark 5.4.4. Let us further define $\widehat{T}h$ as the density of $\ln(\exp(\mathbf{Z}_h) + p) + \ln \widehat{\mathbf{R}}$. Then the recursion algorithm 3.1 changes such that the following recursion is incorporated:

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options

1. $k_1 = \widehat{p}_0 \widehat{p}_0 e^{(r-q+\frac{\sigma^2}{2})(2t_1-2t_0)} f_{\ln \widehat{\mathbf{R}}}$, $h_1 = \widehat{p}_0 e^{(r-q+\frac{\sigma^2}{2})(t_1-t_0)} f_{\ln \widehat{\mathbf{R}}}$ and $g_1 = f_{\ln \mathbf{R}}$.

2. set $k_{k+1} = e^{2(r-q+\frac{\sigma^2}{2})(t_{k+1}-t_k)} \widehat{T} (k_k + 2\widehat{p}_k h_k + \widehat{p}_k \widehat{p}_k f_k)$,
 $h_{k+1} = e^{(r-q)(t_{k+1}-t_k)} \widehat{T} (h_k + \widehat{p}_k f_k)$ and $f_{k+1} = T f_k$, for $k = 1, \dots, n-1$.

Remark 5.4.6. ρ and C can be generalized for the case when the interest rate varies from period to period but it is constant within the periods (bucketing). Let r_j be the interest rate for period $[t_j, t_{j+1})$, then

$$\frac{\partial \mathbf{R}_{kn}}{\partial r_j} = (t_{j+1} - t_j) \mathbf{R}_{kn}, \text{ and } \frac{\partial^2 \mathbf{R}_{kn}}{\partial r_j \partial r_i} = (t_{j+1} - t_j)(t_{i+1} - t_i) \mathbf{R}_{kn}, \quad k \leq j, k \leq i$$

and

$$\frac{\partial \mathbf{A}_n}{\partial r_j} = \sum_{k=0}^j p_k (t_{j+1} - t_j) \mathbf{R}_{kn} = (t_{j+1} - t_j) \sum_{k=0}^j p_k \mathbf{R}_{kn}.$$

Remark 5.4.7. In case of a vanilla put option, similarly to Remark 5.4.2,

$$\rho_0 = -t_n G_n(K) - \exp(-rt_n) p_0 t_n \exp((r-q)t_n) \mathbf{P}\{\widehat{\mathbf{A}}_n^k \leq K\},$$

with $p_0 = S_0$ and $\mathbf{P}\{\widehat{\mathbf{A}}_n^k \leq K\} = N(-d_1)$, where d_1 is given by (5.4.12). Hence,

$$\begin{aligned} \rho_0 &= -t_n (K \exp(-rt_n) N(-d_2) - S_0 \exp(-qt_n) N(-d_1)) - S_0 t_n \exp(-qt_n) N(-d_1) \\ &= -t_n K \exp(-rt_n) N(-d_2), \end{aligned}$$

with $d_2 := d_1 - \sigma \sqrt{t_n}$.

Vega hedging

Vega (ϑ) is the sensitivity of the price w.r.t. the volatility. It is therefore defined by the expression

$$\vartheta = E_Q \left(\frac{\partial \phi(\mathbf{A}_n)}{\partial \mathbf{A}_n} \frac{\partial \mathbf{A}_n}{\partial \sigma} \right), \quad (5.4.22)$$

where

$$\frac{\partial \mathbf{A}_n}{\partial \sigma} = \sum_{k=0}^{n-1} p_k \frac{\partial \mathbf{R}_{kn}}{\partial \sigma} \text{ and } \frac{\partial \mathbf{R}_{kn}}{\partial \sigma} = \left(\mathbf{W}_{(t_k, t_n)} - \sigma(t_n - t_k) \right) \mathbf{R}_{kn}.$$

The term $\exp \left(\sigma \mathbf{W}_{(t_k, t_n)} - \frac{1}{2} \sigma^2 (t_n - t_k) \right)$ is exactly the Radon-Nikodym derivative, which defines a new measure under which $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t 1_{\{t \geq t_k\}}$ is a Brownian motion, yielding

$$- \exp(-rt_n) \sum_{k=0}^{n-1} p_k \exp((r-q)(t_n - t_k)) E_1(1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} \widehat{\mathbf{W}}_{(t_k, t_n)}) \quad (5.4.23)$$

5.4 Greeks

where $\widehat{\mathbf{A}}_n^k$ is defined by (5.4.19) and \widehat{p}_k is defined in section 5.4.1. With the notation $\widehat{\mathbf{W}}_j := \widehat{\mathbf{W}}_{(t_j, t_{j+1})}$ we have

$$E_1(1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} \widehat{\mathbf{W}}_{(t_k, t_n)}) = \sum_{j=k}^{n-1} E_1(1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} \widehat{\mathbf{W}}_j). \quad (5.4.24)$$

The problem which remains now (and which does not occur in the computation of the other Greeks) is to compute the term $E_1(1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} \widehat{\mathbf{W}}_j)$. To this end we observe that $\widehat{\mathbf{W}}_j$ is Gaussian, and for a standard Gaussian density f we have the following result:

$$zf(z) = -\frac{\partial f(z)}{\partial z}. \quad (5.4.25)$$

Hence, if Ψ is a function of $\widehat{\mathbf{W}}_j$, then we obtain by integration by parts (and by $\Psi(w)f(w) \rightarrow 0$ as $|w| \rightarrow \infty$) that

$$E(\widehat{\mathbf{W}}_j \Psi(\widehat{\mathbf{W}}_j)) = (t_{j+1} - t_j) E((D\Psi)(\widehat{\mathbf{W}}_j)), \quad (5.4.26)$$

where D stands for the differentiation operator w.r.t. $\widehat{\mathbf{W}}_j$. We observe that $\widehat{\mathbf{A}}_n^k$ is a function of $\widehat{\mathbf{W}}_j$. Therefore we can apply (5.4.26) to the function $\Psi(\widehat{\mathbf{W}}_j) := 1_{\{\widehat{\mathbf{A}}_n^k(\widehat{\mathbf{W}}_j) \leq K\}}$ and obtain

$$\begin{aligned} E_1(1_{\{\widehat{\mathbf{A}}_n^k \leq K\}} \widehat{\mathbf{W}}_j) &= -\sigma(t_{j+1} - t_j) \sum_{i=0}^j p_i E_1(\delta(K - \widehat{\mathbf{A}}_{in}^k) \widehat{\mathbf{R}}_{in}^k) \\ &= -\sigma(t_{j+1} - t_j) \sum_{i=0}^j e^{(r-q)(t_n - t_i)} e^{\sigma^2(t_n - \{t_i \vee t_k\})} p_i f_{\widehat{\mathbf{A}}_{in}^k}(K), \end{aligned} \quad (5.4.27)$$

with $\widehat{\mathbf{A}}_{in}^k$ defined by (5.4.20). Hence, we obtain for vega, combining (5.4.23), (5.4.24) and (5.4.27)

$$\vartheta = \exp(-rt_n) \sum_{k=0}^{n-1} p_k e^{(r-q)(t_n - t_k)} \sum_{j=k}^{n-1} \sigma(t_{j+1} - t_j) \sum_{i=0}^j e^{(r-q)(t_n - t_i)} e^{\sigma^2(t_n - \{t_i \vee t_k\})} p_i f_{\widehat{\mathbf{A}}_{in}^k}(K). \quad (5.4.28)$$

Switching the summations over j and i finally yields

$$\vartheta = \exp(-rt_n) \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} p_k e^{(r-q)(t_n - t_k)} \sigma(t_n - \{t_i \vee t_k\}) e^{\sigma^2(t_n - \{t_i \vee t_k\})} p_i e^{(r-q)(t_n - t_i)} f_{\widehat{\mathbf{A}}_{in}^k}(K). \quad (5.4.29)$$

Relation (5.4.29) can be efficiently computed with a similar recursion to the one in Remark 5.4.5.

Remark 5.4.8. *The vega can be generalized for the case when the volatility varies from period to period but it is constant within the periods, incorporating the volatility surface (the returns*

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options of the different periods must remain independent). Let σ_k be the volatility for period $[t_k, t_{k+1})$, then

$$\frac{\partial \mathbf{R}_{kn}}{\partial \sigma_j} = \left(-\sigma_j(t_{j+1} - t_j) + \mathbf{W}_{(t_j, t_{j+1})} \right) \mathbf{R}_{kn}, \quad k \leq j$$

which can be substituted into (5.4.22) and the same line of reasoning can be followed as above.

5.4.2 Greeks in a jump-diffusion model

In this section we first show briefly how the recursion algorithm can be applied in case the underlying \mathbf{S} follows a jump-diffusion instead of a lognormal process, then we compute the Greeks. Consider now the process (see for instance Merton(1976), Runggaldier(2003), Glasserman and Merener (2003))

$$\frac{d\mathbf{S}_t}{\mathbf{S}_t} = (r - q)dt + \sigma d\mathbf{W}_t + d \left(\sum_{j=1}^{N_t} (\mathbf{Y}_j - 1) \right), \quad (5.4.30)$$

where \mathbf{W} is a Brownian motion as before, N a Poisson process with rate λ , \mathbf{Y}_j independent and identically distributed random variables with density f_Y , and \mathbf{W} , \mathbf{S} and $\{\mathbf{Y}_1, \mathbf{Y}_2, \dots\}$ mutually independent. \mathbf{S} is right continuous with \mathbf{S}_{t-} the left limit at t . Then \mathbf{S} is a jump diffusion process with jump size $\mathbf{Y}_j, j = 1, \dots$. Given our algorithm, we are actually interested in the dynamics of the process $\ln(\mathbf{S}_t)$. Applying Itô's lemma to the jump diffusion process (5.4.30) (see Runggaldier (2003)) we obtain

$$d \left(\ln \left(\frac{\mathbf{S}_t}{\mathbf{S}_0} \right) \right) = \left(r - q - \frac{\sigma^2}{2} \right) dt + \sigma d\mathbf{W}_t + d \left(\sum_{j=1}^{N_t} \ln(\mathbf{Y}_j) \right).$$

The solution of the above SDE is given by

$$\frac{\mathbf{S}_t}{\mathbf{S}_0} = \exp \left(\left(r - q - \frac{\sigma^2}{2} \right) t + \sigma \mathbf{W}_t \right) \prod_{j=1}^{N_t} \mathbf{Y}_j. \quad (5.4.31)$$

The Laplace transform of $\ln\left(\frac{\mathbf{S}_t}{\mathbf{S}_0}\right)$ is then given by

$$\widehat{f}_{\ln \mathbf{S}}(s) = \exp \left((r - q)ts - \frac{\sigma^2}{2}ts \right) \exp \left(- \int_0^t \lambda(u) du (1 - \widehat{f}_{\ln Y}(s)) \right), \quad (5.4.32)$$

with $\widehat{f}_{\ln \mathbf{S}}$ the Laplace transform of the density of $\ln(\mathbf{S})$, and $\widehat{f}_{\ln Y}$ the Laplace transform of the density of $\ln(\mathbf{Y})$. Now we can use this transformation in the generic algorithm to generate prices associated with an asset described by a jump-diffusion process. More precisely, the Laplace transform of the logarithm of \mathbf{X} , $\widehat{f}_{\mathbf{X}}$ from Steps 1 and 4 will be

5.4 Greeks

given by (5.4.32), with $t = t_{k+1} - t_k$. All the other functions are calculated from \mathbf{X} , with exactly the same methodology as before.

The Greeks can be calculated with exactly the same methodology as for the lognormal model, with only some modifications in the formulae. It follows from (5.4.31) that \mathbf{R}_{kn} , as defined in (5.4.3), now has the form

$$\mathbf{R}_{kn} = \exp \left((r - q - \frac{1}{2}\sigma^2)(t_n - t_k) + \sigma \mathbf{W}_{(t_k, t_n)} \right) \prod_{\mathbf{N}_{t_k+1}}^{\mathbf{N}_{t_n}} \mathbf{Y}_j. \quad (5.4.33)$$

Relations (5.4.4), (5.4.5) and $\partial \mathbf{A}_n(\bar{p}) / \partial p_k = \mathbf{R}_{kn}$ remain valid unchanged, hence, according to (5.4.33), $\frac{\partial G(\bar{p})}{\partial p_k}$ (see relation (5.4.7)) becomes

$$\frac{\partial G(\bar{p})}{\partial p_k} = \exp((r - q)(t_n - t_k)) \mathbf{E} \frac{\partial \Phi(\mathbf{A}_n(\bar{p}))}{\partial \mathbf{A}_n(\bar{p})} \exp \left(\sigma \mathbf{W}_{(t_k, t_n)} - \frac{1}{2}\sigma^2(t_n - t_k) \right) \prod_{\mathbf{N}_{t_k+1}}^{\mathbf{N}_{t_n}} \mathbf{Y}_j.$$

Then we can apply the Girsanov theorem for jump diffusion processes (see Theorem 5.6.3, Appendix 5.6.2). This theorem implies that the expression

$$\exp(\lambda(1 - \mathbf{E}(\mathbf{Y}))) \exp \left(\sigma \mathbf{W}_{(t_k, t_n)} - \frac{1}{2}\sigma^2(t_n - t_k) \right) \prod_{\mathbf{N}_{t_k+1}}^{\mathbf{N}_{t_n}} \mathbf{Y}_j \quad (5.4.34)$$

is the Radon-Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t 1_{\{t \geq t_k\}}$, and changes the intensity of the jump process into $(\lambda_t \mathbf{E}(\mathbf{Y}), \frac{y}{\mathbf{E}(\mathbf{Y})} f_Y(y) dy)$ (see Theorem 5.6.3). Hence, relation (5.4.7) is multiplied by $\exp(-\lambda(1 - \mathbf{E}(\mathbf{Y})))$, and (5.4.8) becomes

$$\widehat{\mathbf{A}}_n(\bar{p}) := \sum_{k=0}^{n-1} p_k \widehat{\mathbf{R}}_{kn}, \quad (5.4.35)$$

with $\widehat{\mathbf{R}}_{kn}$ the original (unadjusted) return process up to t_k and adjusted after time t_k with a new drift $r - q + \sigma^2$ and intensity of jumps $\widehat{\lambda}(t)$, that is,

$$\frac{d\widehat{\mathbf{R}}_t}{\widehat{\mathbf{R}}_t} = \begin{cases} (r - q + \sigma^2)dt + \sigma d\widehat{\mathbf{W}}_t + d \left(\sum_{\mathbf{N}_{t_k+1}^{\widehat{\mathbf{N}}_{t_n}} (\widehat{\mathbf{Y}}_j - 1) \right), & \text{if } t \geq t_k, \\ (r - q)dt + \sigma d\widehat{\mathbf{W}}_t + d \left(\sum_{j=1}^{\mathbf{N}_{t_k}} (\mathbf{Y}_j - 1) \right), & \text{if } t < t_k, \end{cases} \quad (5.4.36)$$

where the process $(\widehat{\mathbf{N}}, \widehat{\mathbf{Y}}_j)$ is the jump process associated with intensity

$$(\lambda_t \mathbf{E}(\mathbf{Y}), \frac{y}{\mathbf{E}(\mathbf{Y})} f_Y(y) dy).$$

Therefore, the expressions for delta and rho remain the same as in subsections 5.4.1, and 5.4.1, but with the processes (5.4.33) instead of (5.4.2), (5.4.35) instead of (5.4.8),

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options and (5.4.36) instead of (5.4.9). Furthermore, the above also holds for the gamma, the convexity and vega having, instead of the processes (5.4.14) and (5.4.15), defined by the 'double shift' $\mathbf{R}_{0n}\mathbf{R}_{0n}$, the following.

$$\begin{aligned}\mathbf{R}_{0n}\mathbf{R}_{0n} &= \exp\left(2(r-q-\frac{1}{2}\sigma^2)(t_n-t_0)+2\sigma\mathbf{W}_{(t_0,t_n)}\right)\prod_{j=1}^{N_{t_n}}\mathbf{Y}_j^2 \\ &= \exp\left((2(r-q)+\sigma^2)(t_n-t_0)\right)\exp\left(-2\sigma^2(t_n-t_0)+2\sigma\mathbf{W}_{(t_0,t_n)}\right)\prod_{j=1}^{N_{t_n}}\mathbf{Y}_j^2.\end{aligned}$$

With an additional normalization factor,

$$\exp\left(-\left(2(r-q)+\sigma^2\right)(t_n-t_0)-\lambda\left(1-\mathbf{E}(\mathbf{Y}^2)\right)\right)\mathbf{R}_{0n}\mathbf{R}_{0n}$$

is exactly the Radon-Nikodym derivative which corresponds to a 2σ shift of the Brownian motion \mathbf{W} , intensity $\lambda\mathbf{E}(\mathbf{Y}^2)$ and distribution of jumps $\frac{y^2}{\mathbf{E}(\mathbf{Y}^2)}f_Y(y)dy$. Hence (5.4.15) is substituted by

$$\frac{d\tilde{\mathbf{R}}_t}{\tilde{\mathbf{R}}_0} = (r-q+2\sigma^2)dt + \sigma d\tilde{\mathbf{W}}_t + d\left(\sum_{j=1}^{\tilde{N}_t}\tilde{\mathbf{Y}}_j - 1\right), \quad (5.4.37)$$

with $(\tilde{\mathbf{N}}, \tilde{\mathbf{Y}}_j)$ the jump process associated with intensity $(\lambda_t\mathbf{E}(\mathbf{Y}^2), \frac{y^2}{\mathbf{E}(\mathbf{Y}^2)}f_Y(y)dy)$.

5.4.3 Discretely sampled Asian options and numerical results

The Greeks for Asian options can be derived in a similar fashion as for guaranteed products. Let ϕ be a cost function (associated with a put or call option), then we obtain

$$\Delta = \frac{e^{-rT}}{n} \sum_{i=1}^n \mathbf{E}_Q(\phi'(\mathbf{A}_n)\mathbf{R}_{t_i}) = \frac{e^{-rT}}{n} \sum_{i=1}^n e^{(r-q)t_i} \mathbf{E}_1(\phi'(\hat{\mathbf{A}}_n^i)), \quad (5.4.38)$$

where

$$\hat{\mathbf{A}}_k^i = \frac{S_0}{k} \sum_{j=1}^k \hat{\mathbf{R}}_{t_j}^i, \quad (5.4.39)$$

and, with $\widehat{\mathbf{W}}_t^i := \mathbf{W}_t - \sigma t 1_{\{t \leq t_i\}}$, a Brownian motion under the new measure, and

$$\frac{d\widehat{\mathbf{R}}_t^i}{\widehat{\mathbf{R}}_t^i} = \begin{cases} (r-q+\sigma^2)dt + \sigma d\widehat{\mathbf{W}}_t^i, & \text{if } t \leq t_i, \\ (r-q)dt + \sigma d\widehat{\mathbf{W}}_t^i, & \text{if } t > t_i. \end{cases} \quad (5.4.40)$$

Relation (5.4.38) can be efficiently computed with remark 5.4.4. Similarly, for the gamma we obtain

$$\Gamma = \frac{e^{-rT}}{n^2} \sum_{i,j=1}^n e^{(r-q)(t_i+t_j)} e^{\sigma^2(t_i \wedge t_j)} \mathbf{E}_2(\phi''(\tilde{\mathbf{A}}_n^{ij})) \quad (5.4.41)$$

5.4 Greeks

where $t_i \wedge t_j = \min(t_i, t_j)$ and

$$\tilde{\mathbf{A}}_k^{ij} = \frac{S_0}{k} \sum_{m=1}^k \hat{\mathbf{R}}_{t_m}^{ij}, \tag{5.4.42}$$

and, with $\tilde{\mathbf{W}}_t^{ij} := \widehat{\mathbf{W}}_t^i - \sigma t 1_{\{t \leq t_j\}}$ a Brownian motion under the new measure,

$$\frac{d\tilde{\mathbf{R}}_t^{ij}}{\tilde{\mathbf{R}}_t^{ij}} = \begin{cases} (r - q + 2\sigma^2)dt + \sigma d\tilde{\mathbf{W}}_t^{ij}, & \text{if } t \leq \min\{t_i, t_j\}, \\ (r - q + \sigma^2)dt + \sigma d\tilde{\mathbf{W}}_t^{ij}, & \text{if } t_i \leq t \leq t_j \text{ or } t_j \leq t \leq t_i, \\ (r - q)dt + \sigma d\tilde{\mathbf{W}}_t^{ij}, & \text{if } t > \max\{t_i, t_j\}. \end{cases} \tag{5.4.43}$$

Relation (5.4.41) can be efficiently computed with remark 5.4.5. As for the case of guaranteed products, the rho, convexity and vega are also functions of the processes (5.4.40) and (5.4.21). As mentioned in the previous sections, the computation of the Greeks can be done by using the recursive pricing Algorithm 5.3.1 and Remarks 5.4.4 and 5.4.5.

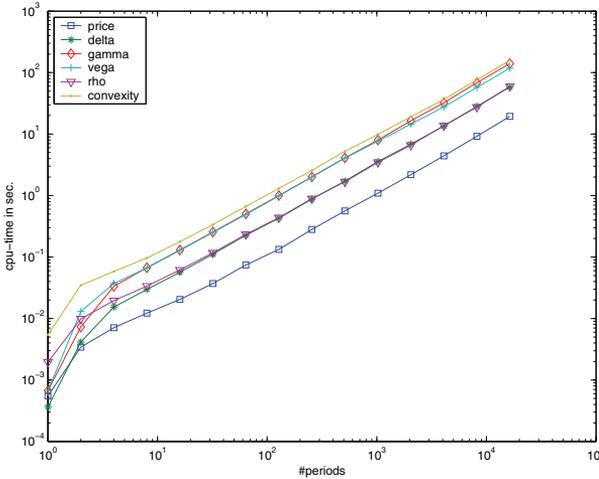


Figure 5.2: The CPU time:

The parameters for the figure are $r = 3, \sigma = 20, q = 0, p_0 = \dots = p_{29} = 100$.

Let us list some numerical results for the price, delta, gamma, vega, rho and convexity of discretely sampled Asian options. We compute the Greeks with a similar extrapolation scheme as we did in section 5.3. The price and the Greeks are computed for seven cases listed in Table 5.3, taken from Linetsky (2004). The prices are compared with the results of Linetsky (2004), however, the computed Greeks are compared with

cases	r	σ	T	S_0
1	0.02	0.10	1.0	2.0
2	0.18	0.30	1.0	2.0
3	0.01	0.25	2.0	2.0
4	0.05	0.50	1.0	1.9
5	0.05	0.50	1.0	2.0
6	0.05	0.50	1.0	2.1
7	0.05	0.50	2.0	2.0

Table 5.3: Parameters. For all seven cases $q=0$ and $K=2.0$

cases	price	delta	gamma
1	0.055986041543 (4e-011)	0.572107791366 (6e-011)	3.353188203114 (2e-010)
2	0.218387546594 (5e-012)	0.661541232179 (1e-010)	0.879916336093(6e-011)
3	0.172268741019 (2e-011)	0.549995593460 (4e-011)	0.953161231928(4e-011)
4	0.193173790285 (1e-011)	0.498093900184 (1e-010)	0.698986962408 (7e-011)
5	0.246415690495 (4e-012)	0.566049429447 (2e-010)	0.657401342083 (1e-010)
6	0.306220364797 (3e-012)	0.629124489589 (4e-011)	0.602319111950(8e-011)
7	0.350095218971 (3e-011)	0.58349 9660355 (3e-010)	0.442251854351 (6e-011)

Table 5.4: Price, delta and gamma of APO's

finite difference approximations of the market sensitivities. We accelerate the convergence of these finite difference approximations with an extrapolation scheme. These computational results are presented in Tables 5.4 and 5.5; the numbers between brackets represent the estimated errors. Figure 5.2 plots the CPU-time for the evaluation of the price, delta, gamma, vega, rho and convexity of discretely sampled Asian options. We can conclude that the CPU time is linear in the number of periods.

5.5 A detailed description of Algorithm 5.3.1

5.5.1 Preliminaries: Legendre polynomial expansions

The Legendre polynomials $\{\phi_k : k \in \mathbb{N}_0\}$ are a complete orthogonal polynomial set in $L^2([0, 1])$. These polynomials $\{\phi_k : k \in \mathbb{N}_0\}$, $\phi_k : [0, 1] \rightarrow \mathbb{R}$ are given by

$$\phi_k(t) = \frac{\sqrt{2k+1}}{k!} \partial^k (t^k (1-t)^k).$$

5.5 A detailed description of Algorithm 5.3.1

cases	vega	rho	convexity
1	0.447398102684 (2e-010)	0.523620411474 (8e-010)	2.642267651044 (9e-010)
2	0.354123779086 (5e-010)	0.476243957180(4e-011)	0.191337829882 (2e-010)
3	0.633744541604 (1e-010)	0.799724838979(4e-010)	1.456815482243 (4e-010)
4	0.426881896070(7e-010)	0.310472050230 (4e-011)	0.171153451784 (1e-010)
5	0.435935577933 (1e-009)	0.351704891791(3e-010)	0.109105610452 (6e-011)
6	0.431764076418 (2e-010)	0.387222661077 (8e-011)	0.032569258057 (2e-010)
7	0.583435612129 (3e-009)	0.559762853266(2e-011)	-0.170596288948 (8e-010)

Table 5.5: Vega, rho and convexity of APO's

Let $P_n f$ denote the projection of f on the space π_n of polynomials with degree less than or equal to n . Hence,

$$P_n f = \sum_{k=0}^n \langle f, \phi_k \rangle \phi_k.$$

Since the shifted Legendre polynomials $\{\phi_k(\cdot - j) : k \in \mathbb{N}_0, j \in \mathbb{Z}\}$ are a complete orthogonal set in $L^2((-\infty, \infty))$, any function $f \in L^2((-\infty, \infty))$ can be expanded into

$$f = \sum_{k=0}^{\infty} \sum_{j=-\infty}^{\infty} \langle f, \phi_k(\cdot - t_0 - j) \rangle \phi_k(\cdot - t_0 - j),$$

with t_0 a shift parameter. Moreover, since the support of these polynomials is known to be $[0, 1]$, we obtain that

$$f(t) = \sum_{k=0}^{\infty} \langle f, \phi_k(\cdot - t_0 - j) \rangle \phi_k(t - t_0 - j), t \in [t_0 + j, t_0 + j + 1].$$

Now we can refine the intervals (in order to improve the approximations) by considering the function f_Δ given by $f_\Delta(t) = f(\Delta t)$. By expanding the function f we obtain that

$$f(t) = f_\Delta \left(\frac{t}{\Delta} \right) = \sum_{k=0}^{\infty} \left\langle f_\Delta, \phi_k \left(\cdot - \frac{t_0}{\Delta} - j \right) \right\rangle \phi_k \left(\frac{t - t_0}{\Delta} - j \right), t \in [t_0 + j\Delta, t_0 + (j+1)\Delta],$$

and its projection $P_{n,\Delta} f$ on π_n is given by

$$P_{n,\Delta} f(t) = \sum_{k=0}^n \left\langle f_\Delta, \phi_k \left(\cdot - \frac{t_0}{\Delta} - j \right) \right\rangle \phi_k \left(\frac{t - t_0}{\Delta} - j \right), j(t) = \{j : t \in [t_0 + j\Delta, t_0 + (j+1)\Delta]\}, \quad (5.5.1)$$

with the shift t_0 . Since $P_{n,\Delta} f$ is determined by the coefficients of the expansion, the stepsize and the shift, we can use the notation $P_{n,\Delta} f = [[\langle f_\Delta, \phi_k(\cdot) \rangle], \Delta, t_0]$. In order to be able to evaluate these Legendre polynomials, we observe that they satisfy the

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options
recurrence relation

$$\phi_0(t) = 1, \phi_1(t) = \frac{\sqrt{3}}{6} \left(t - \frac{1}{2} \right), \quad (5.5.2)$$

$$\left(t - \frac{1}{2} \right) \phi_k = \theta_{k-1} \phi_{k-1}(t) + \theta_k \phi_{k+1}(t),$$

with

$$\theta_k = \frac{k+1}{4k+2} \sqrt{\frac{2k+1}{2k+3}}. \quad (5.5.3)$$

We can now set up the following algorithm.

Algorithm 5.5.1: evaluation of Legendre polynomials

Input: vector $t : t[j] = t_j, j = 0, 1, \dots, m$

Output: values $\phi_k : \phi_k[j] = \phi_k(t_j), k = 0, 1, \dots, n, j = 0, 1, \dots, m$.

1. Set $x : x[j] = \left(t_j - \frac{1}{2} \right), j = 0, 1, \dots, m$.
2. Set $\phi_0[j] = 1, j = 0, 1, \dots, m$ and $\phi_1[j] = \frac{\sqrt{3}}{6} x[j]$.
3. Set $\phi_{k+1}[j] = \frac{\theta_{k-1}}{\theta_k} \phi_{k-1}[j] - \frac{x[j]}{\theta_k} \phi_k[j], k = 1, 2, \dots, n-1, j = 0, 1, \dots, m$.

5.5.2 Error estimation for Legendre polynomials

Theoretically, the precision of the piecewise Legendre polynomial approximation increases, roughly speaking, with the size of the refinements factor Δ . This factor can be chosen arbitrarily, however if we take Δ extremely small, then we need a high number of intervals. This requires a longer computation time and larger computer storage. To control the size of the refinements factor Δ , we need an error estimate. Let

$$P_{n,\Delta} f(t) = \sum_{k=0}^n x_{kj} \phi_k \left(\frac{t-t_0}{\Delta} - j \right), \quad x_{kj} := \langle f_{\Delta}, \phi_k(\cdot - \frac{t_0}{\Delta} - j) \rangle,$$

and its refinement

$$P_{n,\frac{\Delta}{2}} f(t) = \sum_{k=0}^n y_{kj} \phi_k \left(\frac{2(t-t_0)}{\Delta} - j \right), \quad y_{kj} := \langle f_{\frac{\Delta}{2}}, \phi_k(\cdot - \frac{2t_0}{\Delta} - j) \rangle.$$

We denote the coefficients of $P_{n,\frac{\Delta}{2}}(P_{n,\Delta})f(t)$ with $\bar{y}_{kj} := \langle P_{n,\Delta} f(t), \phi_k(\frac{2(t-t_0)}{\Delta} - j) \rangle$, and the following identities hold true

$$\bar{y}_{.2j} = Ax_{.j} \text{ and } \bar{y}_{.2j+1} = Bx_{.j}, \quad (5.5.4)$$

where

$$A_{km} = \int_0^1 \phi_m \left(\frac{u}{2} \right) \phi_k(u) du \text{ and } B_{km} = \int_0^1 \phi_m \left(1 + \frac{u}{2} \right) \phi_k(u) du.$$

5.5 A detailed description of Algorithm 5.3.1

The improvement in the error with the refinement of the interval is given then by

$$\|P_{n,\Delta}f - P_{n,\frac{\Delta}{2}}f\| = \sqrt{\sum_j \|y_{\cdot 2j} - Ax_{\cdot j}\|^2 + \|y_{\cdot 2j+1} - Bx_{\cdot j}\|^2} \sqrt{\Delta} \quad (5.5.5)$$

5.5.3 Gaussian quadrature and matrix representation

We approximate $\langle f, \phi_k(\cdot - j) \rangle$ by a Gaussian quadrature formula (for Gaussian quadrature see the seminal work of Szegő (1975)), given by

$$\langle f, \phi_k(\cdot - j) \rangle_n = \sum_{k=0}^n \alpha_k f_j(\lambda_k) \phi_k(\lambda_k), \quad (5.5.6)$$

where

$$f_j(t) := f(j + t), \quad (5.5.7)$$

$\lambda_k, k = 0, \dots, n$ are the roots of ϕ_{n+1} , and α_k are the so-called Christoffel numbers, given by

$$\alpha_k = \frac{1}{\sum_{j=0}^n |\phi_j(\lambda_k)|^2}.$$

Now we give a matrix representation of (5.5.6). Such a representation facilitates a straightforward evaluation of function operations (summation, multiplication, composite); examples are provided at the end of the subsection. Let us therefore introduce the multiplication operator N , defined by

$$Nf(t) = tf(t) \text{ w.r.t. the norm on } L^2, \|\cdot\|_{L^2}, \text{ that is, } \|Nf - g\|_{L^2} = 0, \text{ with } g(t) := tf(t). \quad (5.5.8)$$

We denote with \widehat{N} the matrix representation of N w.r.t. the basis $\{\phi_k\}$, that is, $\widehat{N}[k, j] = \langle N\phi_k, \phi_j \rangle$. We obtain from (5.5.2) that

$$\widehat{N} = \begin{bmatrix} \frac{1}{2} & \theta_0 & 0 & \cdots & 0 & \cdots \\ \theta_0 & \frac{1}{2} & \theta_1 & \ddots & \vdots & \cdots \\ 0 & \theta_1 & \ddots & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & \frac{1}{2} & \theta_{n-2} & \cdots \\ 0 & \cdots & 0 & \theta_{n-2} & \frac{1}{2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

where the numbers θ_k are given by relation (5.5.3). Since \widehat{N} is a infinite matrix we will approximate it with a finite rank approximation. The finite rank approximation

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options is $N_n = P_n N$, with P_n the orthogonal projector on π_n , and the matrix \widehat{N}_n is given by $\widehat{N}_n[k, j] = \widehat{N}[k, j]$, $k, j = 0, \dots, n$. Since \widehat{N}_n is symmetric, it is a diagonalizable and

$$\widehat{N}_n = UD(\lambda)U^*, \quad (5.5.9)$$

where $D(\lambda)$ denotes the diagonal matrix with the real valued eigenvalues $\{\lambda_k : k = 0, 1, \dots, n\}$ of \widehat{N}_n (and the roots of ϕ_{n+1}), and U^* is the adjungate matrix of U . Furthermore, U is a unitary matrix and the eigenvectors of \widehat{N}_n are the columns of U . Furthermore, we have for all functions f

$$f(\widehat{N}_n) := UD(f(\lambda))U^*, \quad (5.5.10)$$

and

$$\begin{aligned} f_j(\widehat{N}_n)e_0 &= \langle f, \phi(\cdot - j) \rangle_n, \text{ that is,} \\ f_j(\widehat{N}_n)e_0[k] &= \langle f, \phi_k(\cdot - j) \rangle_n, \end{aligned} \quad (5.5.11)$$

where $e_0 = (1, 0, \dots, 0)$. Hence, we can write the product above as

$$f_j(\widehat{N}_n)e_0 = UD(f_j(\lambda))U^*e_0 = U_0f_j(\lambda), \quad (5.5.12)$$

where U_0 is the matrix $U_0 = UD(U^*e_0)$, and $D(U^*e_0)$ denotes the diagonal matrix with the values $\{U^*e_0[k] : k = 0, 1, \dots, n\}$ on the diagonal. The inverse of the matrix U_0 is given by

$$U_0^{-1} = U^{-1}D^{-1}(U^*e_0) = U^*D^{-1}(U^*e_0),$$

where $D^{-1}(U^*e_0)$ is the diagonal matrix with values $\{\frac{1}{U^*e_0[k]} : k = 0, 1, \dots, n\}$ on the diagonal. This implies that we can invert equation (5.5.12), and we obtain the identity

$$f_j(\lambda) = U_0^{-1} \langle f, \phi(\cdot - j) \rangle_n.$$

The following theorem summarizes these results.

Theorem 5.5.1. *The following two identities are valid:*

$$\begin{aligned} \langle f, \phi(\cdot - j) \rangle_n &= U_0f_j(\lambda), \\ f_j(\lambda) &= U_0^{-1} \langle f, \phi(\cdot - j) \rangle_n. \end{aligned}$$

This result shows how one can approximate the expansion coefficients if specific function values are known, and conversely, obtaining function values if the expansion coefficients are known.

Remark 5.5.2. *We can compute the expansion $P_{n,\Delta}f$ by using the functions $f_j(t) = f_\Delta(j + t) = f(\Delta(j + t))$ to obtain $\langle f_\Delta, \phi(\cdot - j) \rangle_n = U_0f_j(\lambda)$, $j = 0, 1, \dots, m - 1$.*

5.5 A detailed description of Algorithm 5.3.1

Let us finally give a few examples of how function operations work with the above matrix expansions. Let $x_f = \langle f, \phi \rangle$. Then

$$\langle f \cdot g, \phi \rangle = g(\widehat{N}_n)x_f = U_0 D(g(\lambda))U_0^{-1}x_f. \quad (5.5.13)$$

In particular, in Step 2 of algorithm 5.3.1 we encounter the expression $f(\ln(\exp(z) - p)) \frac{\exp(z)}{\exp(z) - p}$. First we evaluate, with formula (5.5.1), $h_j(\lambda)$, where $h_j(t) = h(j + t)$ and

$$h(t) = f(\ln(\exp(\lambda) - p)),$$

with t as in Theorem (5.5.1). Next we compute

$$x_g = U_0 (h_j(\lambda)K_j(\lambda)), \text{ with } K(t) = \frac{\exp(t)}{\exp(t) - p}. \quad (5.5.14)$$

This subsection showed how one can approximate functions with orthogonal polynomial expansions. Given specific function values, we can compute these expansions efficiently with a Gaussian quadrature rule. The next subsection shows how one can approximate functions in the so called Q -domain, which makes a transformation between the time domain and Laplace domain possible.

5.5.4 Approximations in the Laplace domain

Let us first introduce an inner product $\langle \cdot, \cdot \rangle_{Q_v}$, induced by the operator Q_v , defined below. Here we only list the necessary definitions, for details and interpretations the interested reader is referred to den Iseger (2006a) and (2006b).

Let $\{\lambda_k^v : k \in \mathbb{Z}, 0 < v < 1\}$ and $\{w_k^v : k \in \mathbb{Z}, 0 < v < 1\}$ be given by

$$\lambda_k^v = i(2\pi(k + v)) \text{ and } w_k^v(t) = \exp(-\lambda_k^v t).$$

Furthermore, the operators $\widehat{S}_v : L^2(\{\lambda_k^v\}) \rightarrow L^2([0, 1])$ and Ψ are given by

$$\widehat{S}_v h := \lim_{N \rightarrow \infty} \sum_{k=-N}^N h(\lambda_k^v) w_k^v$$

and $\Psi h(s) = s^{-1}h(s^{-1})$, respectively. Then the inner product $\langle \cdot, \cdot \rangle_{Q_v}$ is defined by

$$\langle f, g \rangle_{Q_v} := \langle \widehat{S}_v \Psi f, \widehat{S}_v \Psi g \rangle = \sum_{k=-\infty}^{\infty} \alpha_k^v f(\mu_k^v) g^*(\mu_k^v), \quad \mu_k^v = \frac{1}{\lambda_k^v}, \quad \alpha_k^v = |\mu_k^v|^2. \quad (5.5.15)$$

And finally we define the operator $Q_v : L^2(Q_v) \rightarrow L^2([0, 1])$ by $Q_v = \widehat{S}_v \Psi$, having $f \in L^2(Q_v)$ iff $\{\Psi f(2\pi i(k + v)) : k \in \mathbb{Z}\}$ belongs to l^2 . The following important result makes the link between the Q -domain and the time domain.

Theorem 5.5.3. Let Lf be a vector of z -transforms with the k -th element given by

$$Lf[k](\exp(-2\pi iv)) = \sum_{j=-\infty}^{\infty} \langle f, \phi_k(\cdot - j) \rangle \exp(-2\pi i j v)$$

The following relation holds true

$$\langle \Psi \widehat{f}, q_k^v \rangle_{Q_v} = Lf[k](\exp(-2\pi iv)), \quad (5.5.16)$$

where $\widehat{f} \in L^2(-i\infty, i\infty)$ denotes the Fourier (Laplace) transform, and the polynomials $\{q_k^v : k \in \mathbb{N}\}$ are given by

$$q_k^v := p_k(s) - (-1)^k \exp(-2\pi iv) p_k(-s), \quad (5.5.17)$$

with

$$p_k(s) = \sqrt{2k+1} \sum_{j=0}^k \frac{j+k!}{k-j!} \frac{(-s)^j}{j!}.$$

This set of polynomials $\{q_k^v : k \in \mathbb{N}\}$ is orthogonal and complete in $L^2(Q_v)$. Hence any $f \in L^2(Q_v)$ has the expansion

$$f = \sum_{k=0}^{\infty} \langle f, q_k^v \rangle_{Q_v} q_k^v, \text{ w.r.t. the } \|\cdot\|_{Q_v} \text{ norm,}$$

with all the results of the previous subsection holding true for q_k^v instead of $\phi_k(\cdot - j)$. That is, $\langle f, q_k^v \rangle_{Q_v}$ can be approximated with the Gaussian quadrature rule $\langle f, \widehat{q}_k^v \rangle_{Q_v^n}$, defined by

$$\langle f, g \rangle_{Q_v^n} := \sum_{k=0}^{n-1} \beta_k^v f(\eta_k^v) g^*(\eta_k^v), \quad (5.5.18)$$

with $\eta_k^v, k = 0, \dots, n$ the roots of q_{n+1}^v , and β_k^v are the Christoffel numbers related to q_n^v ,

$$\beta_k^v = \frac{1}{\sum_{j=0}^{n-1} |q_j^v(\eta_k^v)|^2}.$$

Similarly as before (see (5.6.5)), we can introduce the matrix representation of (5.5.18), with the multiplication operator $M(v) : L^2(Q_v) \rightarrow L^2(Q_v)$,

$$M(v)f(s) = sf(s), \text{ w.r.t. the norm } \|\cdot\|_{Q_v}, \quad (5.5.19)$$

and

$$f(\widehat{M}(v))e_0[k] = \langle f, q_k^v \rangle_{Q_v}. \quad (5.5.20)$$

5.5 A detailed description of Algorithm 5.3.1

Considering again the finite projection $M_n(v) = P_n M(v)$, the explicit form of matrix $\widehat{M}_n(v)$ is

$$\widehat{M}_n(v) = \begin{bmatrix} \frac{1 + \exp(-i2\pi v)}{2} \frac{1 - \exp(-i2\pi v)}{1 - \exp(-i2\pi v)} & \gamma_0 & 0 & \cdots & 0 \\ -\gamma_0 & 0 & \gamma_1 & \ddots & \vdots \\ 0 & -\gamma_1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & \gamma_{n-1} \\ 0 & \cdots & 0 & -\gamma_{n-1} & 0 \end{bmatrix},$$

where the numbers γ_k are given by

$$\gamma_k = \frac{1}{2} \frac{1}{\sqrt{2k+1}} \frac{1}{\sqrt{2k+3}}.$$

Since the matrix $\widehat{M}_n(v)$ is skew symmetric it is diagonalizable and we can write

$$\widehat{M}_n(v) = V(v) D(\eta^v) V^*(v),$$

where $D(\eta^v)$ denotes the diagonal matrices with the pure imaginary valued eigenvalues $\{\eta_k^v : k = 0, 1, \dots, n-1\}$ of $\widehat{M}_n(v)$, and $V(v)$ is a unitary matrix (that is $V(v)V^*(v) = V^*(v)V(v) = I$), the eigenvectors of $\widehat{M}_n(v)$ consisting the columns of $V(v)$. Furthermore we have

$$f(\widehat{M}_n(v))e_0[k] = \langle f, q_k^v \rangle_{\mathbb{Q}_0^n}, \quad (5.5.21)$$

where

$$f(\widehat{M}_n(v)) := V(v) D(f(\eta^v)) V^*(v).$$

Similarly to the Legendre expansions, we obtain the following result.

Theorem 5.5.4. *The following two identities are valid:*

$$\langle f, q^v \rangle_{\mathbb{Q}_0^n} = V_0(v) f(\eta^v),$$

$$f(\eta^v) = V_0(v)^{-1} \langle f, q^v \rangle_{\mathbb{Q}_0^n},$$

with $V_0(v) = V(v) D(V^*(v)e_0)$ and $V_0^{-1}(v) = D^{-1}(V^*(v)e_0) V^*(v)$.

Together with (5.5.16) the following approximations realize the transformations from the time domain to the Laplace domain and inversely.

Theorem 5.5.5. *Let $L_n f$ be an $(n+1)$ -element row vector of z-transforms, with $L_n f[k] = Lf[k]$, $k = 0, \dots, n$. The following identities are approximately valid:*

$$L_n f(\exp(-2\pi i v)) = V_0(v) (\Psi \widehat{f}(\eta^v)),$$

$$\Psi \widehat{f}(\eta^v) = V_0^{-1}(v) (L_n f(\exp(-2\pi i v))).$$

The following result is a modification of the above theorem, which leads to more efficient computations; we use this step in our recursion algorithm.

Theorem 5.5.6.

$$L_n f(\exp(-\alpha) \exp(-2\pi i v))$$

is equal to

$$\exp\left(\alpha + 2\pi i \left(v - \frac{1}{2}\right) \widehat{N}_n\right) \widehat{V}_0 \Psi \widehat{f}_{\alpha+2\pi i(v-\frac{1}{2})}(\widehat{\eta}) \quad (5.5.22)$$

$$= UD\left(\alpha + 2\pi i \left(v - \frac{1}{2}\right) \lambda\right) W \Psi \widehat{f}_{\alpha+2\pi i(v-\frac{1}{2})}(\widehat{\eta}), \quad (5.5.23)$$

where $\widehat{f}_\alpha(s) = \widehat{f}(\alpha + s)$, $\widehat{V}_0 = V_0(v)$, $W = U^* \widehat{V}_0$ and $\widehat{\eta} := \eta^v$ for $v = \frac{1}{2}$.

$$\Psi \widehat{f}_{\alpha+2\pi i(v-\frac{1}{2})}(\widehat{\eta})$$

is equal to

$$\widehat{V}_0^{-1} \exp\left(-\left(\alpha + 2\pi i \left(v - \frac{1}{2}\right)\right) \widehat{N}_n\right) L_n f(\exp(-\alpha) \exp(-2\pi i v)) \quad (5.5.24)$$

$$= W^{-1} D\left(\exp\left(-\left(\alpha + 2\pi i \left(v - \frac{1}{2}\right)\right) \lambda\right)\right) U^* L_n f(\exp(-\alpha) \exp(-2\pi i v)) \quad (5.5.25)$$

where $\widehat{V}_0^{-1} = V_0^{-1}(v)$, $W^{-1} = \widehat{V}_0^{-1} U$ and $\widehat{\eta} := \eta^v$ for $v = \frac{1}{2}$.

Remark 5.5.7. Since the matrix \widehat{N}_n is tri-diagonal, real and symmetric, and $\widehat{M}_n(v)$ is tri-diagonal, real, and skew-symmetric, we can use the well-known QR algorithm (cf. Stoer and Burlirsch (2000)) for efficient computation of the eigenvalues and eigenvectors.

5.6 Computing the put option price

The integration operator I is given by

$$If(x) = \int_{-\infty}^x f(t) dt.$$

For $f \in \pi_n$ we can approximate this operator by the operator I_n given by $P_n(If)$, where P_n is the projection on π_n .

$$\langle I_n f, \phi(\cdot - j) \rangle = \Delta \widehat{M}_n \left(\frac{1}{2} \right) \langle f_\Delta, \phi(\cdot - j) \rangle + \Delta A_0 e_0,$$

5.6 Computing the put option price

where e_0 is the vector $(1, 0, \dots, 0)^T$ and

$$A_0 = \sum_{l=-\infty}^{j-1} \langle f_{\Delta}, \phi_0(\cdot - l) \rangle + \frac{1}{2} \langle f_{\Delta}, \phi_0(\cdot - j) \rangle.$$

If the set $[L, U]$ is a given ϵ -support set of f then we can accurately approximate A_0 with

$$\sum_{l=L}^{j-1} \langle f, \phi_0(\cdot - l) \rangle + \frac{1}{2} \langle f, \phi_0(\cdot - j) \rangle.$$

The value of a put option with strike e^k is given by

$$P(e^k) = e^{-rT} \int_{-\infty}^k (e^k - e^t) f(t) dt,$$

with f the density of the log of the underlying. The following algorithm computes the above option price.

Algorithm 5.6.1

1. Compute $g(t) = e^t f(t)$ according to (5.5.13);
2. Compute $I_n f$ and $I_n g$;
3. Compute $P(e^k) = e^{-rT} (e^k I_n f(k) - I_n g(k))$ (see again (5.5.13)).

An alternative method is used by Carr and Madan . They write the price of the option as

$$\exp(k) \exp(-rT) (\varphi * f)(k), \tag{5.6.1}$$

with $\varphi(x) = 1 - \exp(-x)$. Convolutions such as (5.6.1) are difficult to calculate directly, but Laplace transforms convert convolutions into products:

$$L_G(s) = \exp(k) \exp(-rT) \frac{1}{s} \frac{L_f(s)}{1+s}, \tag{5.6.2}$$

where L_f denotes the Laplace transform of f . After determining L_f , we invert the Laplace transform of G , L_G , with the transformation algorithm Algorithm 4.5.5.1 and evaluate it in the point k , obtaining the price of the guaranteed product for all strikes k .

Observation 5.6.1. *If we compute the option prices by inverting formula (5.6.2) then in the last iteration of Algorithm 5.3, in Step 3, we have to use the Laplace algorithm with damping, and replace Step 5 by formula (5.6.2) and invert L_G with the Laplace inversion Algorithm 4.5.5.1 with damping.*

5.6.1 Controlling the ϵ -support in the recursion algorithm

Let us now take a closer look at our algorithm. We start with the first step: Suppose that the set $[L_{f_{X_1}}, U_{f_{X_1}}]$ is a given ϵ -support set for the function f_{X_1} . Then the set $[L_{f_1}, U_{f_1}]$ with

$$L_{f_1} = L_{f_{X_1}} - \ln(p_0) \text{ and } U_{f_1} = U_{f_{X_1}} - \ln(p_0) \quad (5.6.3)$$

is an ϵ -support set for the function $f_1(x) = f_{X_1}(x + \ln(p_0))$.

For the second step: Suppose that the set $[L_{f_k}, U_{f_k}]$ is a given ϵ_k -support set for the function f_k . Then the set $[L_{g_k}, U_{g_k}]$ with

$$L_{g_k} = \ln(\exp(L_{f_k}) + p_k) \text{ and } U_{g_k} = \ln(\exp(U_{f_k}) + p_k)$$

is an ϵ_k -support set for the function

$$g_k(z) = f_k(\ln(\exp(z) - p_k)) \frac{\exp(z)}{\exp(z) - p_k}.$$

For step 3,4 and 5: Suppose that the set $[L_{f_{X_k}}, U_{f_{X_k}}]$ is a given ϵ -support set for the function f_{X_k} . Then the set $[L_{f_{k+1}}, U_{f_{k+1}}]$ with

$$L_{f_{k+1}} = L_{g_k} + L_{f_{X_k}} \text{ and } U_{f_{k+1}} = U_{g_k} + U_{f_{X_k}}$$

is a ϵ_{k+1} -support set, with

$$\epsilon_{k+1} = 1 - (1 - \epsilon_k)^2 + \epsilon,$$

for the function f_k .

Although the above algorithm gives excellent results, the disadvantage is that the intervals $[L_{f_{k+1}}, U_{f_{k+1}}]$ become very large. The following modification controls the growth of these intervals. We show therefore how one needs to determine L^* and U^* of the minimal length interval $[L^*, U^*]$ such that,

$$\int_{L^*}^{U^*} f_{k+1}(x) dx \geq (1 - \epsilon) \int_{L_{f_{k+1}}}^{U_{f_{k+1}}} f_{k+1}(x) dx,$$

with ϵ predefined.

Algorithm 5.6.1.1

$$L := L_f, U := U_f, L^* = L_f, U^* = U_f, \hat{n} = 0, \Delta = \frac{U_f - L_f}{m}, k = 0,$$

$$\eta = \int_{L_f}^{U_f} f(x) dx = \Delta \sum_{j=0}^{m-1} \langle f_\Delta, \phi_0(\cdot - j) \rangle;$$

1. while $(U \leq U_f - 1)$ do steps 2,3 and 4

5.6 Computing the put option price

2. repeat

$$\text{Set } \hat{n} = \hat{n} + \Delta \langle f_{\Delta}, \phi_0(\cdot - U) \rangle;$$

$$\text{Set } U = U + 1;$$

$$\text{until } (\hat{n} \geq (1 - \varepsilon)n) \text{ or } (U \geq U_f);$$

3. repeat

$$\text{Set } \hat{n} = \hat{n} - \Delta \langle f_{\Delta}, \phi_0(\cdot - L) \rangle;$$

$$\text{Set } L = L + 1;$$

$$\text{until } \hat{n} < (1 - \varepsilon)n;$$

4. if $(U - (L - 1) < U^* - L^*)$ and $(\hat{n} \geq (1 - \varepsilon)n)$ {

$$L^* = L - 1;$$

$$U^* = U;}$$

5.6.2 Detailed pseudo code of the main algorithm:

1. **Expansion** (m, L, U, p, f_X) :

output: Pf ;

$$\Delta = (U - L)/m;$$

$f_{kj} := f_X(L + (j + \lambda_k)\Delta + \ln p)$, $k = 0, 1, \dots, n$ and $j = 0, 1, \dots, m - 1$; (λ_k : the roots of ϕ_{n+1})

$$Pf = U_0f; \text{ (cf. theorem 5.5.1)}$$

2. **Step 1** $(m, L, U, p, f_X, \varepsilon)$:

output: m, P_2f ;

This routine computes a piecewise Legendre polynomial expansion for the pdf of Z_1 .

$$P_2f = \mathbf{Expansion}(m, L, U, p, f_X);$$

repeat

$$m = 2m;$$

$$P_1f = P_2f;$$

$$P_2f = \mathbf{Expansion}(m, L, U, p, f_X);$$

$$\text{until } \|P_1f - P_2f\| < \varepsilon \text{ (cf. 5.5.5);}$$

3. $\mathbf{B}_0(m, L, U, Pf, p, \hat{f}_X, L_X, U_X)$

output: Pf_1, L_{f_1}, U_{f_1} ;

$$L_g = \ln(\exp(L) + p);$$

$$U_g = \ln(\exp(U) + p);$$

$$\Delta = \frac{U_g - L_g}{m};$$

$y_{kj} = Pf(\ln(\exp(L_g + (j + \lambda_k)\Delta) - p))$, with shift $Pf.t_0 = L$, $k = 0, 1, \dots, n$ and $j = 0, 1, \dots, m - 1$; (cf. (5.5.1))

$$x_{kj} = \frac{\exp(L_g + (j + \lambda_k)\Delta)}{\exp(L_g + (j + \lambda_k)\Delta) - p}, k = 0, 1, \dots, n \text{ and } j = 0, 1, \dots, m - 1;$$

$$z_{kj} = x_{kj}y_{kj}, k = 0, 1, \dots, n \text{ and } j = 0, 1, \dots, m - 1;$$

Step 2: $Pg = U_0z$ (cf. theorem 5.5.1);

$$m_2 = \min\{2^p; 2^p > m + \frac{U_X - L_X}{\Delta}\}, L_{f_1} = L_g + L_X, U_{f_1} = L_{f_1} + m_2\Delta;$$

Set $(Pg)_{kj} = 0$ $k = 0, 1, \dots, n$ and $j = m, m + 1, \dots, m_2 - 1$;

Step 3: $[\hat{g}(s_{kj})] = \mathcal{L}(Pg)$ (cf. algorithm 4.5.5.2);

Step 4: $\hat{f}_1(s_{kj}) = \frac{1}{\Delta} \hat{f}_X(\frac{s_{kj}}{\Delta}) \hat{g}_k(s_{kj})$, $k = 0, 1, \dots, n$ and $j = 0, 1, \dots, m_2 - 1$;

Step 5: $Pf_1 = [\mathcal{L}^{-1} \hat{f}_1, \Delta, L_{f_1}]$; (cf. algorithm 4.5.5.1 and relation (5.5.1))

4. **The body of the algorithm:** $\mathbf{B}(m, L, U, Pf, p, \hat{f}_X, L_X, U_X, \epsilon)$:

output: $m, P_2f_1, L_{f_1}, U_{f_1}$

$$P_2f_1 = \mathbf{B}_0(m, L, U, Pf, p, L_X, U_X);$$

repeat

$$m = 2m;$$

$$P_1f_1 = P_2f_1;$$

$$[P_2f_1, L_{f_1}, U_{f_1}] = \mathbf{B}_0(m, L, U, Pf, p, L_X, U_X);$$

until $\|P_1f_1 - P_2f_1\| < \epsilon$; (cf. 5.5.5)

$$[L_{f_1}, U_{f_1}] = \text{algorithm 5.6.1.1}(m, L_{f_1}, U_{f_1}, P_2f_1);$$

5. **Integration**($[y; k = 0, 1, \dots, n; j = 0, 1, \dots, m - 1], \Delta$):

output: Iy

This routine computes a Legendre polynomial expansion for $F(x) = \int_{-\infty}^x f(t)dt$ given a Legendre polynomial expansion for f (cf. appendix 5.6).

5.6 Computing the put option price

$$Iy = \Delta \widehat{M}_n \left(\frac{1}{2} \right) y;$$

$$A_0 = \frac{1}{2};$$

for $j = 0$ to $m - 1$ {

$$Iy_{0j} = Iy_{0j} + \Delta A_0;$$

$$A_0 = A_0 + y_{0j};$$

6. **Product function**($[x; k = 0, 1, \dots, n; j = 0, 1, \dots, m - 1], g, L$):

output: x_h

This routine computes a Legendre polynomial expansion for the product function h , $h(x) = f(x)g(x)$ given the function g and a Legendre polynomial expansion for f (cf. (5.5.13)).

$$y_{kj} = U_0^{-1} x_{kj}, k = 0, 1, \dots, n; j = 0, 1, \dots, m - 1;$$

$$z_{kj} = y_{kj} g(L + \Delta(j + \lambda_k)), k = 0, 1, \dots, n, j = 0, 1, \dots, m - 1;$$

$$x_h = U_0 z;$$

7. **Body of algorithm 5.6.1:** $O(x_f, \Delta, L, R)$

output: p

This routine computes a Legendre polynomial expansion for the option price given (as function of the log strike price) a Legendre polynomial expansions for the pdf of the log-price. (cf. appendix 5.6)

$$x_g = \text{Product function}(x_f, e^t, \Delta, L);$$

$$x_{If} = \text{Integration}(x_f, \Delta);$$

$$y = \text{Integration}(x_g, \Delta);$$

$$z = \text{Product function}(x_{If}, e^t, \Delta, L);$$

$$p_{kj} = e^{-R} (z_{kj} - y_{kj});$$

8. **Compute option price**($x, \Delta, L, R, \epsilon$):

output: $P_2 f$

This routine computes a Legendre polynomial expansion for the option price given (as function of the log strike price) a Legendre polynomial expansions for the pdf of the log-price. (cf. appendix 5.6)

Chapter 5. Pricing guaranteed return rate products and discretely sampled Asian options

$$P_2f = [\mathbf{O}(x, \Delta, L, R), \Delta, L]$$

repeat

$$\Delta = \frac{\Delta}{2};$$

$$y_{k2j} = Ax_{kj} \text{ and } y_{k2j+1} = Bx_{kj}, k = 0, 1, \dots, n \text{ (cf. (equation 5.5.4))};$$

$$x = y;$$

$$P_1f = P_2f;$$

$$P_2f = [\mathbf{O}(x, \Delta, L, R), \Delta, L];$$

until $\|P_1f - P_2f\| < \epsilon$; (cf. (5.5.5))

Finally, the algorithm is called as follows:

input: $f_{X_1}, \widehat{f}_{X_k}$, and L_{X_k}, U_{X_k} (defined cf. (5.6.3));

1. $Pf_1 = \mathbf{Step1}(m, L_{f_1}, U_{f_1}, f_{X_1});$

2. $N = \# \text{ of periods};$

3. for $k = 2$ to $N\{$

$$[M_k, Pf_k, L_{f_k}, U_{f_k}] = \mathbf{B}(m, L_{f_{k-1}}, U_{f_{k-1}}, Pf_{k-1}, p_{k-1}, \widehat{f}_{X_k}, L_{X_k}, U_{X_k});\}$$

4. **Compute option price** $(Pf_N, \frac{U_{f_N} - L_{f_N}}{M_N}, L_{f_N}, U_{f_N}).$

Appendix: Girsanov's theorem for lognormal and jump-diffusion processes

For a detailed discussion on a more general level of the following standard results we refer the reader to the overview paper of Runggaldier (2003).

Theorem 5.6.2 (Girsanov's theorem). *Given a filtered probability space $(\Omega, \mathcal{F}_t, \mathbf{P})$, let $t \in [0, T]$ and γ_t be a square integrable predictable process. Define $L = (L_t)$ by*

$$L_t = \exp \left(\int_0^t \gamma_t d\mathbf{W}_t - \frac{1}{2} \int_0^t \gamma_t^2 dt \right) \quad (5.6.4)$$

with $E^{\mathbf{P}}(L_t) = 1$ for all t . Then there exists a probability measure \mathbf{Q} on \mathcal{F} , equivalent to \mathbf{P} , with $d\mathbf{Q} = L_T d\mathbf{P}$ such that

$$d\mathbf{W}_t^{\mathbf{Q}} = d\mathbf{W}_t - \gamma_t dt,$$

where $\mathbf{W}_t^{\mathbf{Q}}$ is a \mathbf{Q} -Wiener process.

Conversely, if $\mathcal{F}_t = \mathcal{F}_t^{\mathbf{W}}$ then every probability measure $\mathbf{Q} \sim \mathbf{P}$, has the above structure.

5.6 Computing the put option price

Now, besides a Wiener process \mathbf{W}_t , we also have a jump process (\mathbf{Y}_j) , occurring randomly over time at $\mathbf{t}_k, k = 0, 1, \dots$ according to the Poisson process

$$\mathbf{N}_t := \sum_{n \geq 1} 1_{\{\mathbf{t}_n \leq t\}} \text{ with intensity } \frac{\partial \mathbf{E}(\mathbf{N}_t)}{\partial t} = \lambda_t, \quad (5.6.5)$$

and the \mathbf{Y}_n are i.i.d., independent of \mathbf{N}_t . The dynamics of such a process are described by

$$d \left(\ln \left(\frac{\mathbf{S}_t}{\mathbf{S}_0} \right) \right) = \left(\mu - \frac{\sigma^2}{2} \right) dt + \sigma d\mathbf{W}_t + d \left(\sum_{j=1}^{\mathbf{N}_t} \ln(\mathbf{Y}_j) \right). \quad (5.6.6)$$

The solution of the above SDE is given by

$$\frac{\mathbf{S}_t}{\mathbf{S}_0} = \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma \mathbf{W}_t \right) \prod_{j=1}^{\mathbf{N}_t} \mathbf{Y}_j. \quad (5.6.7)$$

A Girsanov-type measure transformation allows, in addition to the translation of the Wiener process, to perform also a change in the intensity process of the jump process part. We have

Theorem 5.6.3 (Girsanov's theorem for jump-diffusions). *On the finite time interval $[0, T]$, let \mathbf{S}_t given by (5.6.7) be a jump-diffusion process with \mathbf{P} -intensity of jumps λ and the distribution of jumps $f_Y(dy)$ under \mathbf{P} . Let γ_t be a square integrable predictable process. Define $L = (L_t)$ by*

$$L_t = \exp \left(\int_0^t \gamma_t d\mathbf{W}_t - \frac{1}{2} \int_0^t \gamma_t^2 dt \right) \exp(\lambda(1 - \mathbf{E}(\mathbf{Y}))) \prod_{j=1}^{\mathbf{N}_t} \mathbf{Y}_j, \quad (5.6.8)$$

$\mathbf{E}(L_t) = 1$ for all t , then all the statements of Theorem 5.6.2 hold true in addition to the fact that under \mathbf{Q} the intensity of the jumps becomes $\lambda \mathbf{E}(\mathbf{Y})$, and the distribution of the jumps becomes $\frac{y f_Y^Q(y)}{\mathbf{E}(\mathbf{Y})}$.

Remark 5.6.4. *Theorem 5.6.3 is a special case of Theorem 2.5 of Runggaldier(2003) with $\lambda_t = \lambda$, $h_t(y) = \frac{y}{\mathbf{E}(\mathbf{Y})}$, $m_t(dy) = f_Y(y)dy$, and $\psi_t = \mathbf{E}(\mathbf{Y})$, for all $t \in [0, T]$.*

Chapter 6

Computing Greeks: A drift-adjustment technique for European and Asian style derivatives

6.1 Introduction

The risks associated with derivatives, such as the risk of the stock price moving up or down or implied volatility changing, are measured by the so called Greeks, the sensitivity of the derivative w.r.t. stock price (delta), interest rate (rho), implied volatility (vega). Even the second order sensitivities, such as the gamma and the convexity are of great importance when hedging derivatives. Recently, there have been important achievements in numerical methods for option pricing, such as the fast Gauss transform algorithm of Broadie and Yamamoto (2005) and numerical Laplace transform inversion (see den Iseger (2006a)). However, for the Greeks there has been no such breakthrough without Monte Carlo techniques. There are closed form formulae for plain vanilla options, but for more complex derivative securities less reliable approximation techniques are used. An often used method is, for example, the so called finite difference method in combination with simulation. These methods can become highly unstable in critical points. The present chapter introduces a method which reduces the computation of the Greeks to a similar problem as that of computing the price of the derivative in question, hence enabling the use of the advanced FGT and LT techniques also for the computation of the Greeks. More precisely, the method is based on a Girsanov type drift adjustment technique (see den Iseger and Oldenkamp (2006a)); this technique transforms the expression of the Greeks into an expression (a function) of the price of the derivative under yet another measure; that is, if there is an efficient algorithm pricing the derivative then in order to compute the Greeks we will use the the same efficient algorithm only with a modified payoff function. These

expressions can then be computed using the same numerical techniques (FGT and LT) as for calculating the price of derivative securities, hence does not need Monte-Carlo methods. Peter Carr (2000) derives the n th derivative of a path-independent option, observing that these derivatives can be expressed solely in the terms of the stock price derivative (which observation we also use for the computation of the Greeks).

Our technique can be applied to a large class of path dependent options with lognormal or jump-diffusion underlying and it is uniform for the delta, gamma, rho and convexity, and it is only slightly different for the vega (there we need an extra result, although we use the same drift adjustment technique).

The chapter is structured as follows. First we introduce the method for computing the delta, gamma, rho, convexity and vega for European style options with lognormal underlying. Secondly we demonstrate how the technique can be applied for path dependent options (section 6.3), and sections 6.3.1 and 6.3.3 present specific examples: Greeks for average price options and cliquet options, respectively. In section 6.4 we present numerical results, and appendix 6.4 discusses how the Greeks can be deduced when the underlying follows a jump-diffusion process.

The results of this chapter are described in the working paper den Iseger and Oldenkamp (2006b).

6.2 European style plain – vanilla options

A European style option can only be exercised at the end of its life, T . We assume that the underlying is modelled by the stochastic process \mathbf{S}_t . Let us denote the return on \mathbf{S} by \mathbf{R} , such that $\mathbf{R}_t := \frac{\mathbf{S}_t}{\mathbf{S}_0}$. The payoff is determined by a given function of the underlying $\phi(\mathbf{S}_t)$, yielding for the arbitrage free price of the option

$$e^{-rT} \mathbf{E}_Q(\phi(\mathbf{S}_T)). \quad (6.2.1)$$

This section presents a new technique for the computation of the Greeks for lognormal and jump-diffusion underlying.

6.2.1 Lognormal underlying

The main assumption of this subsection is that \mathbf{S}_t is lognormally distributed, such that

$$\mathbf{S}_t = \mathbf{S}_0 \exp\left(\left(r - q - \frac{1}{2}\sigma^2\right)t + \sigma \mathbf{W}_t\right) = \mathbf{S}_0 \mathbf{R}_t, \quad (6.2.2)$$

where \mathbf{W}_t is a Brownian motion at time t , r is the risk free rate, q is the dividend rate, and σ is the volatility. The risk free rate (as well the dividend rate and the volatility) is deterministic but need not be constant, the method can also handle "buckets". The

6.2 European style plain – vanilla options

computation method remains unchanged for the latter case, and we will point out at the end of the relevant subsections how buckets can be handled. The log return is normally distributed with mean $(r - q - \frac{\sigma^2}{2})t$ and volatility σt .

Delta hedging

The Delta of an option is the sensitivity of its price w.r.t. to the underlying, hence we need to evaluate the expression $e^{-rT} \partial \mathbf{E}_Q \phi(\mathbf{S}_T) / \partial \mathbf{S}_0$. Denoting the derivative of ϕ by ϕ' (that is $\phi'(\mathbf{S}_T) = \partial \phi(\mathbf{S}_T) / \partial \mathbf{S}_T$), one obtains by the chain rule that

$$\Delta = e^{-rT} \mathbf{E}_Q \left(\phi'(\mathbf{S}_T) \mathbf{R}_T \right) = e^{-rT} e^{(r-q)T} \mathbf{E}_Q \left(\phi'(\mathbf{S}_T) e^{\sigma \mathbf{W}_T - \frac{1}{2} \sigma^2 T} \right). \quad (6.2.3)$$

Relation (6.2.3) is well known and often used with Monte Carlo techniques. However, the following transformation is to our best knowledge new. The term $e^{\sigma \mathbf{W}_t - \frac{1}{2} \sigma^2 t}$ is exactly the Radon Nikodym derivative, which, according to the Girsanov theorem, realizes a drift adjustment with σ of the Brownian motion (under a new measure) $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t$. Hence, we obtain for the delta

$$\Delta = e^{-qT} \mathbf{E}_1 \left(\phi'(\widehat{\mathbf{S}}_T) \right), \quad (6.2.4)$$

with

$$\widehat{\mathbf{S}}_t := \mathbf{S}_0 \widehat{\mathbf{R}}_t \quad \text{and} \quad \widehat{\mathbf{R}}_t := \exp \left(\left(r - q + \frac{1}{2} \sigma^2 \right) t + \sigma \widehat{\mathbf{W}}_t \right). \quad (6.2.5)$$

$\phi'(\cdot)$ can be determined once the payoff function is given explicitly. Expression (6.2.4) can also be deduced for complex derivatives (as shown in section 6.3), and to our best knowledge is new.

Remark 6.2.1. *In case of a plain-vanilla call option the price (6.2.1) is $e^{-rT} \mathbf{E}_Q (\mathbf{S}_T - K)^+$, hence the delta is given by $\Delta_{\text{call}} = e^{-qT} \mathbf{E}_1 \left(1_{\{\widehat{\mathbf{S}}_T \geq K\}} \right)$, that is, delta is the probability that the option expires in the money (under the probability measure induced by (6.2.5)). For a plain-vanilla put option $\Delta_{\text{put}} = -e^{-qT} \mathbf{E}_1 \left(1_{\{\widehat{\mathbf{S}}_T \leq K\}} \right)$.*

Gamma hedging

The Gamma of an option is the sensitivity of the Delta w.r.t. the underlying, that is,

$$\Gamma = \frac{\partial}{\partial \mathbf{S}_0} \left(e^{-qT} \mathbf{E}_1 \left(\phi'(\widehat{\mathbf{S}}_T) \right) \right) = e^{-qT} \mathbf{E}_1 \left(\phi''(\widehat{\mathbf{S}}_T) \widehat{\mathbf{R}}_T \right). \quad (6.2.6)$$

Just as in the case of the delta we split $\widehat{\mathbf{R}}_T = e^{(r-q)T} e^{\sigma \widehat{\mathbf{W}}_T + \frac{1}{2} \sigma^2 T}$, where $e^{\sigma \widehat{\mathbf{W}}_T + \frac{1}{2} \sigma^2 T}$ is exactly the Radon Nikodym derivative, which, according to the Girsanov theorem, realizes a drift adjustment with $-\sigma$ of the Brownian motion $\widetilde{\mathbf{W}}$ (under again a different measure) $\widetilde{\mathbf{W}}_t = \widehat{\mathbf{W}}_t + \sigma t$. Hence, we obtain for the gamma

$$\Gamma = e^{-rT} e^{2(r-q)T} \mathbf{E}_2 \left(\phi''(\widetilde{\mathbf{S}}_T) \right), \quad (6.2.7)$$

with

$$\tilde{\mathbf{S}}_t := \mathbf{S}_0 \tilde{\mathbf{R}}_t \text{ and } \tilde{\mathbf{R}}_t := \exp\left(\left(r - q + \frac{3}{2}\sigma^2\right)t + \sigma \tilde{\mathbf{W}}_t\right). \quad (6.2.8)$$

$\phi''(\cdot)$ can be determined once the payoff function is given explicitly.

Remark 6.2.2. *In case of a plain-vanilla put or call option the gamma is $\Gamma = e^{-rT} e^{2(r-q)T} f_{\tilde{\mathbf{S}}_T}(K)$, that is, the Gamma is the probability that the option expires at the money (under the probability measure induced by (6.2.8)).*

Rho and convexity hedging

Rho (ρ) is the sensitivity (derivative) w.r.t. the risk free interest rate r . The 'convexity' C is the second derivative.

$$\rho = -Te^{-rT} \mathbf{E}_Q(\phi(\mathbf{S}_T)) + e^{-rT} \mathbf{E}_Q(\phi'(\mathbf{S}_t) S_0 T \mathbf{R}_T). \quad (6.2.9)$$

Since \mathbf{R}_T figures again in the second term of the expression (the first is just the price times T), we use the fact that $e^{\sigma \mathbf{W}_t - \frac{1}{2}\sigma^2 t}$ is exactly the Radon Nikodym derivative, which, according to the Girsanov theorem, realizes a drift adjustment with σ of the Brownian motion (under a new measure) $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t$, yielding

$$\rho = -Te^{-rT} \mathbf{E}_Q(\phi(\mathbf{S}_T)) + e^{-rT} e^{(r-q)T} S_0 T \mathbf{E}_1(\phi'(\widehat{\mathbf{S}}_T)), \quad (6.2.10)$$

where $\widehat{\mathbf{S}}_t$ is given by (6.2.5). Similarly, we obtain for the convexity $C = \partial\rho/\partial r$, that is,

$$C = -T\rho - Te^{-rT} e^{(r-q)T} \mathbf{E}_1(\phi'(\widehat{\mathbf{S}}_T) S_0 T) + e^{-rT} e^{(r-q)T} \mathbf{E}_1(S_0^2 T^2 \phi''(\widehat{\mathbf{S}}_T) \widehat{\mathbf{R}}_T). \quad (6.2.11)$$

Following the same line of reasoning as for the gamma, we obtain

$$C = -T\rho - e^{-rT} e^{(r-q)T} S_0 T^2 \mathbf{E}_1(\phi'(\widehat{\mathbf{S}}_T)) + e^{-rT} e^{(r-q)T} S_0^2 T^2 \mathbf{E}_2(\phi''(\tilde{\mathbf{S}}_T)), \quad (6.2.12)$$

where $\tilde{\mathbf{S}}_t$ is defined by relation (6.2.8). $\phi'(\cdot)$ and $\phi''(\cdot)$ can be determined once the payoff function is given explicitly.

Remark 6.2.3. ρ and C can be generalized for the case when the interest rate varies from period to period but it is constant within the periods (bucketing). Let r_j be the interest rate for period $[t_j, t_{j+1})$, then

$$\frac{\partial \mathbf{R}_t}{\partial r_j} = (t_{j+1} - t_j) \mathbf{R}_t, \text{ and } \frac{\partial^2 \mathbf{R}_t}{\partial r_j^2} = (t_{j+1} - t_j)^2 \mathbf{R}_t, \quad j \leq N(t).$$

6.2 European style plain – vanilla options

Vega hedging

Vega (ϑ) is the sensitivity of the option w.r.t. the volatility. It is therefore defined by the expression

$$\vartheta = e^{-rT} E_Q \left(\phi'(\mathbf{S}_T) S_0 \frac{\partial \mathbf{R}_T}{\partial \sigma} \right) = e^{-qT} S_0 E_Q \left(\phi'(\mathbf{S}_T) (-\sigma T + \mathbf{W}_T) e^{-\frac{1}{2}\sigma^2 T + \sigma \mathbf{W}_T} \right). \quad (6.2.13)$$

The term $e^{\sigma \mathbf{W}_t - \frac{1}{2}\sigma^2 t}$ is exactly the Radon Nikodym derivative, which, according to the Girsanov theorem, realizes a drift adjustment with σ of the Brownian motion (under a new measure) $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t$, yielding

$$\vartheta = -e^{-qT} S_0 \sigma T E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \right) + e^{-qT} S_0 E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \widehat{\mathbf{W}}_T \right), \quad (6.2.14)$$

where $\widehat{\mathbf{S}}_t$ is defined by relation (6.2.5). The problem which remains now (and which does not occur in the computation of the other Greeks) is to determine $E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \widehat{\mathbf{W}}_T \right)$. To this end we observe that $\widehat{\mathbf{W}}_t$ is Gaussian, and for a standard Gaussian density f we have the following result:

$$w f(w) = -\frac{\partial f(w)}{\partial w}. \quad (6.2.15)$$

Hence, if Ψ is a function of \mathbf{W} then we obtain by integration by parts (and by $\Psi(w)f(w) \rightarrow 0$ as $|w| \rightarrow \infty$) that

$$E(\mathbf{W}\Psi(\mathbf{W})) = E((D\Psi)(\mathbf{W})), \quad (6.2.16)$$

where D stands for the differentiation operator w.r.t. \mathbf{W} . Applying this result to $\Psi(\widehat{\mathbf{W}}_T) := \phi'(S_0 \exp((r - q + \frac{1}{2}\sigma^2)T + \sigma \widehat{\mathbf{W}}_T))$ yields that

$$E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \widehat{\mathbf{W}}_T \right) = S_0 e^{(r-q)T} \sigma E_1 \left(\phi''(\widehat{\mathbf{S}}_T) e^{\sigma \widehat{\mathbf{W}}_T + \frac{1}{2}\sigma^2 T} \right). \quad (6.2.17)$$

The term $e^{\sigma \widehat{\mathbf{W}}_T + \frac{1}{2}\sigma^2 T}$ is again the Radon Nikodym derivative, which, according to the Girsanov theorem, realizes a drift adjustment with σ of the Brownian motion $\widetilde{\mathbf{W}}_T$ (under a new measure) $\widetilde{\mathbf{W}}_t = \widehat{\mathbf{W}}_t + \sigma t$, yielding that (6.2.17) equals

$$S_0 e^{(r-q)T} \sigma E_2 \left(\phi''(\widetilde{\mathbf{S}}_T) \right),$$

with $\widetilde{\mathbf{S}}_t$ defined by (6.2.8). Substituting this into the expression (6.2.14) of the vega, yields

$$\vartheta = -e^{-qT} S_0 \sigma T E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \right) + e^{(r-2q)T} S_0^2 \sigma E_2 \left(\phi''(\widetilde{\mathbf{S}}_T) \right). \quad (6.2.18)$$

$\phi'(\cdot)$ and $\phi''(\cdot)$ can be determined once the payoff function is given explicitly.

Remark 6.2.4. *The vega can be generalized for the case when the volatility varies from period to period but it is constant within the periods, incorporating skews (the returns of the different periods must remain independent). Let σ_j be the volatility for period $[t_j, t_{j+1})$, then*

$$\frac{\partial \mathbf{R}_T}{\partial \sigma_j} = \left(-\sigma_j (t_{j+1} - t_j) + \mathbf{W}_{(t_j, t_{j+1})} \right) \mathbf{R}_T,$$

which can be substituted into (6.2.13) and the same line of reasoning can be followed as above.

By inspecting the Black-Merton-Scholes partial differential equation, instead of using the probabilistic arguments of the present chapter, Carr (2000) also deduced a similar result to (6.2.4) (see Carr(2000), equation (17)). The result Peter Carr derived is solely applicable for path independent European options. On the contrary, our results are also valid for jump-diffusion models (see appendix 6.4), and for path dependent options, including Asian, Bermudan and cliquet options, as demonstrated in the next section.

6.3 Path dependent options

The uniform method detailed in the previous section for European options with log-normal underlying (as well jump-diffusion underlying, see appendix 6.4) can be applied to path dependent options by observing the following structure. Let

$$E_Q \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n))$$

denote the general cost function associated with an asian type path dependent option (for cliquets we specify the cost function apart), \mathbf{S} is the path dependent underlying, and \mathbf{R} is the return, as before. Calculating the Delta, that is differentiating w.r.t. S_0 results in the structure

$$\Delta = \sum_{j=1}^n E_Q \frac{\partial \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n))}{\partial \mathbf{S}(t_j)} \mathbf{R}(t_j), \quad (6.3.1)$$

and

$$\Gamma = \sum_{j=1}^n \sum_{k=1}^n E_Q \frac{\partial^2 \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n))}{\partial \mathbf{S}(t_j) \partial \mathbf{S}(t_k)} \mathbf{R}(t_j) \mathbf{R}(t_k). \quad (6.3.2)$$

The $\mathbf{R}(t_k)$ can be interpreted as the Radon-Nikodym derivative as before (see (6.2.3) - (6.2.5)).

In case of a cliquet option we observe a similar structure as follows. The price is given by $p = E_Q \Phi(\mathbf{R}(t_1), \dots, \mathbf{R}(t_n))$, while the delta and gamma are given by

$$\Delta = \sum_{j=1}^n E_Q \frac{\partial \Phi(\mathbf{R}(t_1), \dots, \mathbf{R}(t_n))}{\partial \mathbf{R}(t_1)} \mathbf{R}(t_1) \quad (6.3.3)$$

$$\Gamma = \sum_{j=1}^n E_Q \frac{\partial^2 \Phi(\mathbf{R}(t_1), \dots, \mathbf{R}(t_n))}{\partial^2 \mathbf{R}(t_1)} \mathbf{R}(t_1)^2. \quad (6.3.4)$$

6.3 Path dependent options

The next two subsections present the Greeks for average price options and cliquet options, deduced by recognizing the structures (6.3.1)- (6.3.4) with the Radon-Nikodym derivative. The change of drift serves to simplify the formulas and reduce them to a similar structure as that of the price of the derivative; this enables us to use the same recursion algorithm as for the calculation of the price, only with a modified payoff function.

6.3.1 Asian options and Asian-style exotic products

The payoff of an asian or average price option (APO) depends on the arithmetic average of the price of the underlying over the maturity. The data used in the calculation of the average can be continuously or discretely sampled. In practice, discrete sampling is used, for instance closing prices. With the notation

- S_t the price of a unit at time t ,
- t_i $i = 1, \dots, n$ the sampling moments,
- T maturity, $T = t_n$,
- K strike level,
- r risk free rate,

the cost of an average price option is given by

$$e^{-rT} E_Q \left(\phi \left(\frac{1}{n} \sum_{i=1}^n S_{t_i} \right) \right), \quad (6.3.5)$$

where E_Q stands for the expectation w.r.t. the risk-neutral measure, and ϕ is a predefined cost function. In den Iseger and Oldenkamp (2006a) the authors present a recursion algorithm which computes the distribution of the sum (discrete average) (6.3.5) as an iterated convolution. Since in the case non-constant modeling (varying interest rates, volatility skews, and so on) the stochastic variables of the different periods (most importantly the returns) will not be independent, the sum (6.3.5) cannot be computed as a simple convolution. To this end den Iseger and Oldenkamp (2006a) have developed an algorithm which calculates this sum recursively, by observing that on only two levels, the value of the portfolio at time t and the return over period $(t, t + 1)$ are independent, hence they do form a convolution. Since a convolution structure can be determined efficiently with numerical advanced Laplace transform techniques (den Iseger (2006a) (2006) and (2006b)), setting up a recursion relation which involves this convolution lead to an efficient and numerically stable algorithm, which calculates the

price of the derivative. The result of the recursion algorithm is a very accurate approximation of the price function not only in one single point but on a whole interval i.e. a range of strike prices. Next to numerical results for the Greeks, in section 6.4 we also present numerical results for the price of an APO.

Using either definition (6.4.2) or (6.2.2), and defining

$$\mathbf{A}_k := \frac{1}{k} \sum_{i=1}^k \mathbf{S}_{t_i} = \frac{S_0}{k} \sum_{i=1}^k \mathbf{R}_{t_i}, \quad (6.3.6)$$

the sensitivity of the price w.r.t. the market variables can be derived in a similar fashion as for European options:

$$\Delta = \frac{e^{-rT}}{n} \sum_{i=1}^n \mathbf{E}_Q (\phi'(\mathbf{A}_n) \mathbf{R}_{t_i}) = \frac{e^{-rT}}{n} \sum_{i=1}^n e^{(r-q)t_i} \mathbf{E}_1 (\phi'(\widehat{\mathbf{A}}_n^i)), \quad (6.3.7)$$

where

$$\widehat{\mathbf{A}}_k^i = \frac{S_0}{k} \sum_{j=1}^k \widehat{\mathbf{R}}_{t_j}^i, \quad (6.3.8)$$

and, with $\widehat{\mathbf{W}}_t^i := \mathbf{W}_t - \sigma t 1_{\{t \leq t_i\}}$, a Brownian motion under the new measure, and

$$\frac{d\widehat{\mathbf{R}}_t^i}{\widehat{\mathbf{R}}_t^i} = \begin{cases} (r - q + \sigma^2)dt + \sigma d\widehat{\mathbf{W}}_t^i, & \text{if } t \leq t_i, \\ (r - q)dt + \sigma d\widehat{\mathbf{W}}_t^i, & \text{if } t > t_i. \end{cases} \quad (6.3.9)$$

Similarly, for the gamma we obtain

$$\Gamma = \frac{e^{-rT}}{n^2} \sum_{i,j=1}^n e^{(r-q+\frac{\sigma^2}{2})(t_i+t_j)} \mathbf{E}_2 (\phi''(\widetilde{\mathbf{A}}_n^{ij})), \quad (6.3.10)$$

$$\widetilde{\mathbf{A}}_k^{ij} = \frac{S_0}{k} \sum_{m=1}^k \widehat{\mathbf{R}}_{t_m}^{ij}, \quad (6.3.11)$$

and, with $\widetilde{\mathbf{W}}_t^{ij} := \widehat{\mathbf{W}}_t^i - \sigma t 1_{\{t \leq t_j\}}$ a Brownian motion under the new measure,

$$\frac{d\widetilde{\mathbf{R}}_t^{ij}}{\widetilde{\mathbf{R}}_t^{ij}} = \begin{cases} (r - q + 2\sigma^2)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t \leq \min\{t_i, t_j\}, \\ (r - q + \sigma^2)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t_i \leq t \leq t_j \text{ or } t_j \leq t \leq t_i, \\ (r - q)dt + \sigma d\widetilde{\mathbf{W}}_t^{ij}, & \text{if } t > \max\{t_i, t_j\}. \end{cases} \quad (6.3.12)$$

As for the case of European options, for the rho, convexity and vega are functions of the processes (6.3.9) and (6.3.12). For jump-diffusion underlying one can use exactly the same line of reasoning as in appendix 6.4. Furthermore the Greeks of other Asian-type exotic products can be handled in a similar fashion. As an example consider unit linked guaranteed products, which is defined as follows. Let p_i denote the investment premium paid at time t_i , $i = 0, \dots, n-1$, and K the guaranteed amount at expiry. At

6.3 Path dependent options

time t_i the policyholder practically purchases $\frac{p_i}{S_{t_i}}$ units, each unit having value S_T at expiry. Hence, the total fund value at T is $\sum_{i=0}^{n-1} p_i \frac{S_T}{S_{t_i}}$. On the other hand, a minimum of K is paid out at expiry. Hence, the price of the guarantee is given by

$$e^{-rT} E_Q \left(K - \sum_{i=0}^{n-1} p_i \frac{S_T}{S_{t_i}} \right)^+ . \quad (6.3.13)$$

The exact expressions for the Greeks can be found in section 4 of den Iseger and Oldenkamp (2006a).

6.3.2 Bermudan options

The technique presented for European and Asian style options also applies to Bermudan style options with some modifications. Since the precise mathematical proofs are somewhat longer than for Asian options, we refer the interested reader to Appendix 6.4.4. The price of a bermudan option is given by

$$p = \sup_{\tau} E_Q (\Phi(S(t_{\tau})) | S(t_0)), \quad (6.3.14)$$

with τ a stopping time. If we define the random variables

$$\mathbf{H}_m := \Phi(S(t_m)) - E_Q(\Phi(S(\tau)) | S(t_m)),$$

then we can define the optimal stopping time τ as

$$\tau = \inf_t \{\mathbf{H}_t > 0\}.$$

The price (6.3.14) can now be written as

$$p = \sum_{m=1}^N E_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \Phi(S(t_m)).$$

Applying here again the general formula (6.3.1) we obtain that the delta is given by

$$\begin{aligned} \Delta &= \sum_{m=1}^N E_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \frac{\partial \Phi(S(t_m))}{\partial S(t_m)} \mathbf{R}(t_m) \\ &+ \sum_{m=1}^N E_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \frac{\partial \mathbf{H}_m}{\partial S_m} \mathbf{G}_m \mathbf{R}(t_m), \end{aligned} \quad (6.3.15)$$

where

$$\mathbf{G}_m = \left(\Phi(S(t_m)) - \sum_{j=m+1}^N \prod_{l=m+1}^{j-1} 1_{\{\mathbf{H}_l \leq 0\}} 1_{\{\mathbf{H}_j > 0\}} \Phi(S(t_j)) \right).$$

As shown in Appendix 6.4.4, the second term of (6.3.15) has 0 value, hence the delta reduces to

$$\Delta = E_Q \left(\sum_{j=1}^{\tau} \frac{\partial \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau)} \mathbf{R}_{\tau} \middle| \mathbf{S}(t_0) \right). \quad (6.3.16)$$

The Gamma of a bermudan option can be deduced in a similar fashion:

$$\Gamma = E_Q \left(\frac{\partial^2 \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau) \partial \mathbf{S}(\tau)} \mathbf{R}_{\tau}^2 \middle| \mathbf{S}(t_0) \right). \quad (6.3.17)$$

6.3.3 Cliquet options

Cliquet options are essentially a series of forward-starting at-the-money options with a single premium determined up front, that lock in any gains on specific dates. The strike price is then reset at the new level of the underlying asset. We will use the following form (see den Iseger and Oldenkamp (2006c) for all details), considering a global cap, global floor and local caps at pre-defined observation times t_i ($i = 1, \dots, n$),

$$\Phi(\mathbf{Z}_n) := \min(\max(\mathbf{Z}_n, F), C), \quad (6.3.18)$$

where C is the global cap, F is the global floor, and with \mathbf{Z}_k defined by

$$\mathbf{Z}_k := \sum_{i=1}^k \mathbf{Y}_i, \text{ where } \mathbf{Y}_i \text{ is given by (6.3.20), and } k = 1, \dots, n. \quad (6.3.19)$$

The locally capped and floored return \mathbf{Y} is defined as

$$\mathbf{Y}_i := \max(F_i, \min(C_i, \mathbf{R}_i - 1)), \quad (6.3.20)$$

$F_i, i = 1, \dots, n$ the local floors, $C_i, i = 1, \dots, n$ are the local caps. The Laplace transform $\mathcal{L}f_{\mathbf{Y}_i}(s)$ of \mathbf{Y}_i is given by relation

$$e^{-sF_i} \left(F_{R_i}(F_i + 1) + \int_0^{C_i - F_i} e^{-sx} f_{R_i}(x + F_i + 1) dx + e^{-s(C_i - F_i)} (1 - F_{R_i}(C_i + 1)) \right). \quad (6.3.21)$$

Assuming that the returns per period are independent, the Laplace transform of the price is just the product of the Laplace transforms (6.3.21). In order to calculate the price one can invert this Laplace transform with advanced numerical techniques (see for instance den Iseger (2006a) and (2006b)). For a detailed computational guide see Den Iseger and Oldenkamp (2006c).

In order to deduce the delta, we observe the same structure as in (6.3.3), resulting in

$$\Delta = \exp(-rt_n) \frac{1}{S_{0^-}} \exp((r - q)t_1) E \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right), \quad (6.3.22)$$

6.4 Computational results

where

$$\widehat{\mathbf{Z}}_n = \widehat{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widehat{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widehat{\mathbf{R}}_1)), \quad (6.3.23)$$

with

$$\widehat{\mathbf{R}}_1 := \exp\left((r - q + \frac{1}{2}\sigma^2)t_1 + \sigma\widehat{\mathbf{W}}_{t_1}\right). \quad (6.3.24)$$

In conclusion, this yields for the Delta of a cliquet option the following relation

$$\Delta = \exp(-rt_n) \exp((r - q)t_1) \frac{1}{S_{0-}} \mathbf{E} \left(1_{\{F \leq \widehat{\mathbf{Z}}_n \leq C\}} 1_{\{F_1 \leq \widehat{\mathbf{R}}_1 \leq C_1\}} \right). \quad (6.3.25)$$

Similarly, for the Gamma of a cliquet option we refer to (6.3.4) and obtain

$$\begin{aligned} \Gamma = & \exp(-rt_n) \frac{1}{S_{0-}^2} \exp(2(r - q)t_1) \exp(\sigma^2 t_1) \cdot \\ & \cdot \mathbf{E} \left((\delta_F(\widetilde{\mathbf{Z}}_n) - \delta_C(\widetilde{\mathbf{Z}}_n)) 1_{\{F_1 \leq \widetilde{\mathbf{R}}_1 \leq C_1\}} + 1_{\{F \leq \widetilde{\mathbf{Z}}_n \leq C\}} (\delta_{F_1}(\widetilde{\mathbf{R}}_1) - \delta_{C_1}(\widetilde{\mathbf{R}}_1)) \right), \end{aligned} \quad (6.3.26)$$

where

$$\widetilde{\mathbf{Z}}_n = \widetilde{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widetilde{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widetilde{\mathbf{R}}_1)),$$

with

$$\widetilde{\mathbf{R}}_1 := \exp\left((r - q + \frac{3}{2}\sigma^2)t_1 + \sigma\widetilde{\mathbf{W}}_{t_1}\right). \quad (6.3.27)$$

The rho, convexity and vega are deduced according to the same template as in sections 6.2.1 - 6.2.1, detailed expressions can also be found in den Iseger and Oldenkamp (2006c).

6.4 Computational results

Be the theory of the previous sections ever so clever, it will only be considered interesting if the formulae can efficiently be computed numerically. As we previously mentioned, the computation of the Greeks reduces to a similar problem as that of computing the price of the derivative in question, enabling the use of the advanced numerical (FGT and LT) techniques. In den Iseger and Oldenkamp (2006) an efficient recursion algorithm is constructed for the computation of the price of an exotic derivative (unit linked guaranteed product and asian option). This algorithm uses a general pay-off function Φ as input. For the computation of the Greeks we will use this very same algorithm, we need only slightly modify it, and use the derivative Φ' (or Φ'') as input instead of the original pay-off function. For the construction of the original recursion algorithm we used exactly the same technique as for the Greeks: we identified the Radon-Nikodym derivative (see section 6.3 and (6.3.1)), performed a change of drift,

cases	r	σ	T	S_0
1	0.02	0.10	1.0	2.0
2	0.18	0.30	1.0	2.0
3	0.01	0.25	2.0	2.0
4	0.05	0.50	1.0	1.9
5	0.05	0.50	1.0	2.0
6	0.05	0.50	1.0	2.1
7	0.05	0.50	2.0	2.0

Table 6.1: Parameters (for all seven cases $q=0$ and $K=2.0$)

cases	price	delta	gamma
1	0.055986041543 (4e-011)	0.572107791366 (6e-011)	3.353188203114 (2e-010)
2	0.218387546594 (5e-012)	0.661541232179 (1e-010)	0.879916336093(6e-011)
3	0.172268741019 (2e-011)	0.549995593460 (4e-011)	0.953161231928(4e-011)
4	0.193173790285 (1e-011)	0.498093900184 (1e-010)	0.698986962408 (7e-011)
5	0.246415690495 (4e-012)	0.566049429447 (2e-010)	0.657401342083 (1e-010)
6	0.306220364797 (3e-012)	0.629124489589 (4e-011)	0.602319111950(8e-011)
7	0.350095218971 (3e-011)	0.583499660355 (3e-010)	0.442251854351 (6e-011)

Table 6.2: Price, delta and gamma of APO's

then used an LT numerical technique for the computation (see den Iseger and Oldenkamp (2006) for all technical details and pseudo codes). Let us list some numerical results for the price, delta, gamma, vega, rho and convexity of average price options. We will use seven cases, i.e. parameter sets, presented in Table 6.1.

These cases were taken from Linetsky (2004), but as Linetsky also remarks, this set of parameters "has been used as test cases for various numerical methods by Eydeland and Geman (1995), Fu et al. (1998), Shaw (2002), Craddock et al. (2000), Dufresne (2000), and Vecer (2001, 2002)". For the price of an asian option our results converge to the price of continuously sampled average options as we increase the number of periods during a 1-year horizon (that is, decreasing the time steps). The number of periods are repeatedly increased, then we used an extrapolation technique* and obtained an up to 10 decimals accurate approximation (we compared our results to those of Linetsky (2004), table 3 (page 865), with parameters $r = 0.05$, $q = 0$, $\sigma = 0.5$, $K = 2$, $T = 1$, $S_0 = 2$ (case 5)). The computed Greeks are compared with results obtained by finite

* Denoting with $G(2^m, n)$ the n^{th} order extrapolated price with # periods 2^m , then $G(2^m, n+1) = \frac{1}{2^m-1}(2^n G(2^m, n) - G(2^{m-1}, n))$.

6.4 Computational results

cases	vega	rho	convexity
1	0.447398102684 (2e-010)	0.523620411474 (8e-010)	2.642267651044 (9e-010)
2	0.354123779086 (5e-010)	0.476243957180(4e-011)	0.191337829882 (2e-010)
3	0.633744541604 (1e-010)	0.799724838979(4e-010)	1.456815482243 (4e-010)
4	0.426881896070(7e-010)	0.310472050230 (4e-011)	0.171153451784 (1e-010)
5	0.435935577933 (1e-009)	0.351704891791(3e-010)	0.109105610452 (6e-011)
6	0.431764076418 (2e-010)	0.387222661077 (8e-011)	0.032569258057 (2e-010)
7	0.583435612129 (3e-009)	0.559762853266(2e-011)	-0.170596288948 (8e-010)

Table 6.3: vega, rho and convexity of APO's

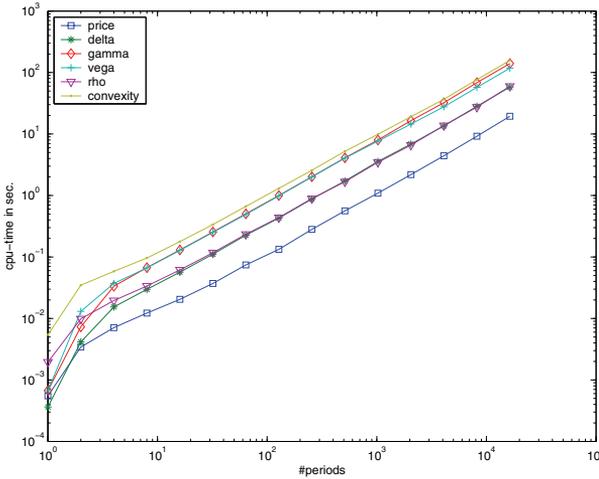


Figure 6.1: CPU times ($r = 3\%$, $\sigma = 20\%$, $q = 0$, $p_0 = \dots = p_{29} = 100$)

difference methods (with extrapolation). These results are presented in Tables 6.2 and 6.3. The estimated errors are the numbers between brackets. Figure 6.4 plots the CPU-time for the evaluation of the price, delta, gamma, vega, rho and convexity of average price options. We can conclude the CPU time is linear in the number of periods.

Appendix A: Jump diffusion underlying

Consider now the process (see for instance Merton(1976), Runggaldier(2003), Glasserman and Merener (2003))

$$\frac{d\mathbf{S}_t}{\mathbf{S}_{t-}} = (r - q)dt + \sigma d\mathbf{W}_t + d\left(\sum_{j=1}^{N_t} (\mathbf{Z}_j - 1)\right), \quad (6.4.1)$$

where \mathbf{W} is a Brownian motion as before, \mathbf{N} a Poisson process with rate λ_t , \mathbf{Z}_j independent and identically lognormally distributed with density f_{Z_j} , and \mathbf{W} , \mathbf{S} and $\{\mathbf{Z}_1, \mathbf{Z}_2, \dots\}$ mutually independent. \mathbf{S} is right continuous with \mathbf{S}_{t-} the left limit at t . Then \mathbf{S} is a jump diffusion process with jump size $\mathbf{Z}_j, j = 1, \dots$. The solution of the above SDE is given by

$$\mathbf{S}_t = S_0 \exp\left(\left(r - q - \frac{\sigma^2}{2}\right)t + \sigma \mathbf{W}_t\right) \prod_{j=1}^{N_t} \mathbf{Z}_j = S_0 \mathbf{R}_t. \quad (6.4.2)$$

The Greeks can be calculated with exactly the same methodology as for the lognormal model, with only some modifications in the formulae. It follows from (6.4.2) that \mathbf{R}_t has now the form

$$\mathbf{R}_t = \exp\left(\left(r - q - \frac{1}{2}\sigma^2\right)t + \sigma \mathbf{W}_t\right) \prod_{j=1}^{N_t} \mathbf{J}_j. \quad (6.4.3)$$

As an equivalent for the lognormal model, the expression

$$\exp(\lambda(1 - \mathbf{E}(\mathbf{J}))) \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma \mathbf{W}_{t_1}\right) \prod_{j=1}^{N_{t_1}} \mathbf{J}_j \quad (6.4.4)$$

is the Radon-Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t$, and changes the intensity of the jump process into

$$(\lambda_t \mathbf{E}(\mathbf{J}), \frac{x}{\mathbf{E}(\mathbf{J})} f_{\mathbf{J}}(x) dx)$$

(see Theorem D.2 in den Iseger and Oldenkamp (2006a)).

6.4.1 Computation of the Delta

Expression (6.2.3) for the Delta does not change in structure, it becomes now

$$\Delta = e^{-rT} E_Q(\phi'(\mathbf{S}_T) \mathbf{R}_T) = e^{-qT} e^{-\lambda(1 - \mathbf{E}(\mathbf{J}))} \mathbf{E}_1\left(\phi'(\widehat{\mathbf{S}}_T)\right), \quad (6.4.5)$$

where

$$\widehat{\mathbf{S}}_t := S_0 \widehat{\mathbf{R}}_t \text{ and } \widehat{\mathbf{R}}_t := \exp\left(\left(r - q + \frac{1}{2}\sigma^2\right)t + \sigma \widehat{\mathbf{W}}_t\right) \prod_{j=1}^{\widehat{N}_t} \widehat{\mathbf{J}}_j, \quad (6.4.6)$$

6.4 Computational results

and the process $(\widehat{\mathbf{N}}, \widehat{\mathbf{J}}_j)$ is the jump process associated with intensity

$$\left(\lambda_t \mathbf{E}(\mathbf{J}), \frac{x}{\mathbf{E}(\mathbf{J})} f_J(x) dx \right).$$

6.4.2 Computation of the Gamma

In the case of the Gamma the adjustment for jump-diffusion processes becomes

$$\Gamma = e^{-rT} \mathbf{E}_Q \left(\phi''(\mathbf{S}_T) \mathbf{R}_T \mathbf{R}_T \right), \quad (6.4.7)$$

with

$$\begin{aligned} \mathbf{R}_T \mathbf{R}_T &= \exp\left(2(r-q) - \frac{1}{2}\sigma^2\right)T + 2\sigma \mathbf{W}_T \prod_{j=1}^{N_T} \mathbf{J}_j^2 \\ &= \exp\left((2(r-q) + \sigma^2)T\right) \exp(-2\sigma^2 T + 2\sigma \mathbf{W}_T) \prod_{j=1}^{N_T} \mathbf{J}_j^2. \end{aligned} \quad (6.4.8)$$

With an additional normalization factor, $\exp(-2((r-q) + \sigma^2)T - \lambda(1 - \mathbf{E}(\mathbf{J}^2))) \mathbf{R}_T \mathbf{R}_T$ is exactly the Radon-Nikodym derivative which corresponds to a 2σ shift of the Brownian motion \mathbf{W} , intensity $\lambda \mathbf{E}(\mathbf{J}^2)$ and distribution of jumps $\frac{x^2}{\mathbf{E}(\mathbf{J}^2)} f_J(x) dx$. Hence Gamma is given by

$$\Gamma = e^{-rT} \exp(-2(r-q) + \sigma^2)T - \lambda(1 - \mathbf{E}(\mathbf{J}^2))) \mathbf{E}_2(\phi''(\widetilde{\mathbf{S}}_T)),$$

where

$$\widetilde{\mathbf{R}}_t = \exp\left(\left(r-q + \frac{3}{2}\sigma^2\right)t + \sigma \widetilde{\mathbf{W}}_t\right) \prod_{j=1}^{\widetilde{N}_t} \widetilde{\mathbf{J}}_j, \quad (6.4.9)$$

instead of (6.2.8), with $(\widetilde{\mathbf{N}}, \widetilde{\mathbf{J}}_j)$ the jump process associated with intensity

$$\left(\lambda_t \mathbf{E}(\mathbf{J}^2), \frac{x^2}{\mathbf{E}(\mathbf{J}^2)} f_J(x) dx \right).$$

Remark 6.4.1. *The statements of remarks 6.2.1 and 6.2.2 remain also valid for the jump-diffusion underlying case.*

6.4.3 Computation of the Rho and convexity

The expressions for rho and the convexity will also have the same structure as in the case of a lognormal underlying, that is,

$$\rho = -T e^{-rT} \mathbf{E}_Q(\phi(\mathbf{S}_T)) + e^{-qT} e^{-\lambda(1-\mathbf{E}(\mathbf{J}))} S_0 T \mathbf{E}_1(\phi'(\widehat{\mathbf{S}}_T)), \quad (6.4.10)$$

where $\widehat{\mathbf{S}}_t$ is given by (6.4.6). Similarly, we obtain for the convexity C

$$-T \rho - e^{-qT - \lambda(1-\mathbf{E}(\mathbf{J}))} S_0 T^2 \mathbf{E}_1(\phi'(\widehat{\mathbf{S}}_T)) + e^{-rT - (2(r-q) + \sigma^2)T - \lambda(1-\mathbf{E}(\mathbf{J}^2))} S_0^2 T^2 \mathbf{E}_2(\phi''(\widetilde{\mathbf{S}}_T)), \quad (6.4.11)$$

where $\widetilde{\mathbf{S}}_t$ is defined by relation (6.4.9).

6.4.4 Computation of the Vega

Similarly to the other Greeks, the expression for lognormal underlying for the vega (6.2.18) remains valid here, after correcting for the normalization factor, hence we obtain

$$\vartheta = -e^{-qT-\lambda(1-E(J))} S_0 \sigma T E_1 \left(\phi'(\widehat{\mathbf{S}}_T) \right) + e^{-rT-(2(r-q)+\sigma^2)T-\lambda(1-E(J^2))} S_0^2 \sigma E_2 \left(\phi''(\widetilde{\mathbf{S}}_T) \right), \quad (6.4.12)$$

where $\widehat{\mathbf{S}}_t$ is defined by relation (6.4.6) and $\widetilde{\mathbf{S}}_t$ is defined by relation (6.4.9).

Remark 6.4.2. *By specifying the function ϕ properly, one can compute with the above methodology the Greeks for more complex, even paths dependent European-style derivatives. Such an example is a cliquet option; for details see den Iseger and Oldenkamp (2006c).*

Appendix B: Bermudan options

The general price function for Asian and Bermudan style options can be formulated as

$$p = E_Q \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n)), \quad (6.4.13)$$

with Φ a corresponding payoff function. Then, the delta and gamma are given by

$$\begin{aligned} \Delta &= \sum_{j=1}^n E_Q \frac{\partial \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n))}{\partial \mathbf{S}(t_j)} \mathbf{R}(t_j) \\ \Gamma &= \sum_{j=1}^n \sum_{k=1}^n E_Q \frac{\partial^2 \Phi(\mathbf{S}(t_1), \dots, \mathbf{S}(t_n))}{\partial \mathbf{S}(t_j) \partial \mathbf{S}(t_k)} \mathbf{R}(t_j) \mathbf{R}(t_k). \end{aligned} \quad (6.4.14)$$

In case of a Bermudan option the general price function (6.4.13) can be formulated as

$$p = \sup_{\tau} E_Q \{ \Phi(\mathbf{S}(t_{\tau})) | \mathbf{S}(t_0) \},$$

with τ a stopping time. If we define the random variables

$$\mathbf{H}_m := \Phi(\mathbf{S}(t_m)) - E_Q \{ \Phi(\mathbf{S}(\tau)) | \mathbf{S}(t_m) \},$$

then the optimal stopping time τ is given by

$$\tau = \inf_t \{ \mathbf{H}_t > 0 \}.$$

With these definitions we can write the price as

$$p = \sum_{m=1}^N E_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \Phi(\mathbf{S}(t_m)).$$

6.4 Computational results

We obtain from formula (6.4.14) that the delta is given by

$$\begin{aligned}\Delta &= \sum_{m=1}^N E_{\mathbb{Q}} \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \frac{\partial \Phi(\mathbf{S}(t_m))}{\partial \mathbf{S}(t_m)} \mathbf{R}(t_m) \\ &\quad + \sum_{m=1}^N E_{\mathbb{Q}} \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} \mathbf{G}_m \mathbf{R}(t_m),\end{aligned}$$

where

$$\mathbf{G}_m = \left(\Phi(\mathbf{S}(t_m)) - \sum_{j=m+1}^N \prod_{l=m+1}^{j-1} 1_{\{\mathbf{H}_l \leq 0\}} 1_{\{\mathbf{H}_j > 0\}} \Phi(\mathbf{S}(t_j)) \right).$$

Observe that since

$$E_{\mathbb{Q}} \{ \mathbf{G}_m | \mathbf{H}_m = 0 \} = E_{\mathbb{Q}} \{ \mathbf{H}_m | \mathbf{H}_m = 0 \} = 0,$$

we obtain by the tower property that

$$\sum_{m=1}^N E_{\mathbb{Q}} \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} \mathbf{G}_m \mathbf{R}(t_m)$$

is equal to

$$\sum_{m=1}^N E_{\mathbb{Q}} \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} E_{\mathbb{Q}} \{ \mathbf{G}_m | \mathbf{H}_m = 0 \} \mathbf{R}(t_m) = 0.$$

Hence, the delta is given by

$$\begin{aligned}\Delta &= \sum_{m=1}^N E_{\mathbb{Q}} \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \frac{\partial \Phi(\mathbf{S}(t_m))}{\partial \mathbf{S}(t_m)} \mathbf{R}(t_m) \\ &= E_{\mathbb{Q}} \left\{ \frac{\partial \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau)} \mathbf{R}_{\tau} \middle| \mathbf{S}(t_0) \right\}.\end{aligned}$$

For the Gamma of a Bermudan option we start again from formula (6.4.14). Let us first introduce the following notation. Let the functions Ψ and random variables Ψ_m be given by

$$\Psi(x) = \frac{\partial \Phi(x)}{\partial \mathbf{S}(x)}, \quad \Psi_m = \Psi(\mathbf{S}_m),$$

and introduce the random variables

$$\begin{aligned}\mathbf{I}_m &= \Psi_m - \sum_{j=m+1}^N \prod_{l=m+1}^{j-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \Psi_j \mathbf{R}(t_m, t_j) \\ &= \Psi_m - \Psi_{\tau} \mathbf{R}(t_m, \tau).\end{aligned}$$

Introduce further the random variables

$$\begin{aligned}\mathbf{J}_m &= E_{\mathbb{Q}} \{ (\mathbf{J}_m) | \mathbf{S}(t_m) \} \\ &= E_{\mathbb{Q}} \{ (\Psi_m - \Psi_{\tau} \mathbf{R}(t_m, \tau)) | \mathbf{S}(t_m) \}.\end{aligned}$$

It is easily seen that

$$\mathbf{J}_m = \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} = \frac{\partial \Phi(\mathbf{S}(t_m))}{\partial \mathbf{S}(t_m)} - \Delta_m,$$

with Δ_m the delta at time t_m that is

$$\Delta_m = \frac{\partial \mathbb{E}_Q \{ \Phi(\mathbf{S}(\tau)) | \mathbf{S}(t_m) \}}{\partial \mathbf{S}(t_m)} = \mathbb{E}_Q \left\{ \frac{\partial \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau)} \mathbf{R}_\tau \middle| \mathbf{S}(t_m) \right\}.$$

We obtain from formula (6.4.14) that the gamma is given by

$$\begin{aligned} \Gamma &= \sum_{m=1}^N \mathbb{E}_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} 1_{\{\mathbf{H}_m > 0\}} \frac{\partial^2 \Phi(\mathbf{S}(t_m))}{\partial \mathbf{S}(t_m) \partial \mathbf{S}(t_m)} \mathbf{R}(t_m) \mathbf{R}(t_m) \\ &\quad + \sum_{m=1}^N \mathbb{E}_Q \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} \mathbf{I}_m \mathbf{R}(t_m). \end{aligned}$$

Observe that by the tower property and since $\mathbb{E}_Q \{ \mathbf{I}_m | \mathbf{S}(t_m) \} = \frac{\partial \mathbf{H}_m}{\partial \mathbf{S}_m} = \Psi_m - \Delta_m$,

$$\begin{aligned} \Gamma &= \mathbb{E}_Q \left\{ \sum_{j=1}^{\tau} \sum_{i=1}^{\tau} \frac{\partial^2 \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau) \partial \mathbf{S}(\tau)} \mathbf{R}_\tau \mathbf{R}_\tau \middle| \mathbf{S}(t_0) \right\} \\ &\quad + \mathbb{E}_Q \left\{ \sum_{m=1}^N \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) (\Psi_m - \Delta_m)^2 \mathbf{R}(t_m)^2 \right\}. \end{aligned}$$

For a Bermuda option we obtain

$$\mathbb{E}_Q \left\{ \sum_{m=1}^N \prod_{k=1}^{m-1} 1_{\{\mathbf{H}_k \leq 0\}} \delta(\mathbf{H}_m) \right\} \mathbb{E}_Q \left\{ (\Psi_m - \Delta_m)^2 \mathbf{R}(t_m)^2 \middle| \mathbf{H}_m = 0, \mathbf{H}_k \leq 0; k = 0, 1, \dots, m-1 \right\}$$

vanishes. In conclusion the Greeks for a Bermudan option

$$\begin{aligned} \Delta &= \mathbb{E}_Q \left\{ \sum_{j=1}^{\tau} \frac{\partial \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau)} \mathbf{R}_\tau \middle| \mathbf{S}(t_0) \right\} \\ \Gamma &= \mathbb{E}_Q \left\{ \frac{\partial^2 \Phi(\mathbf{S}(\tau))}{\partial \mathbf{S}(\tau) \partial \mathbf{S}(\tau)} \mathbf{R}_\tau^2 \middle| \mathbf{S}(t_0) \right\}. \end{aligned} \tag{6.4.15}$$

Chapter 7

The Fast Convolution Transform

7.1 Introduction

The development of the field of mathematical finance has meant a dramatic increase in demand for computationally efficient methods for pricing derivatives and other financial products. Recently, we have seen various numerical methods being employed to price instruments for which no analytical price is available, such as American options and path-dependent derivatives. The main shortcomings of present numerical methods have been lack of accuracy or low convergence speed. The algorithm presented in this chapter, the fast Lévy ladder algorithm, attempts to address and solve these problems by applying a Laplace inversion algorithm to obtain the desired accuracy.

Traditional approaches for valuing American options in the Black-Scholes model have been to calculate the conditional expected value of the payoff (provided the option has not been exercised yet) as a convolution integral with respect to the risk-neutral probability density, but also to solve the associated partial differential equation (PDE) problem with the appropriate boundary conditions using a finite difference method or other PDE techniques. To the former approach belong such chapters as Cox, Ross & Rubinstein (1979), who use binomial trees, and Broadie & Yamamoto (2005), who use the fast Gauss transform to compute the convolution and price barrier and lookback options. Classic chapters on the finite difference method are Brennan & Schwartz (1978), and Hull & White (1990). Seydel (2004) gives an overview of finite difference and finite element methods applied to American options.

One of the limitations of these methods is that they are only applied to models with Gaussian disturbance terms. We would like to be able to extend the applicability of numerical methods to underlying assets based on Lévy processes. We can regard Lévy models as extensions of the jump diffusion model, in that it can be decomposed into a Brownian motion and a quadratic pure jump semimartingale (see Protter (2005)). These models have already been applied to problems in finance, see for instance Eber-

lein & Keller (1995), Barndorff-Nielsen (1998), Geman (2002), and Eberlein & Kluge (2006). The most important reason for this is that Lévy models are capable of reproducing the complex asset dynamics that are observed in the financial markets, and that are not adequately modelled by the geometric Brownian motion of the Black-Scholes model.

The Lévy ladder algorithm we present is capable of dealing with general Lévy processes for the underlying asset, with both the drift and the volatility of the process allowed to be deterministic. Extensions to stochastic drift and volatility could be made, by including an extra state variable, thereby increasing the dimensionality of the problem, but this case is not considered in this chapter. The convolution integrals resulting from the ladder framework involve the density of the underlying process, which may not be explicitly known. We will however have an expression for the characteristic function of the process, which implies that our approach to solving these integral equations will be through a transform inversion algorithm.

The Laplace transform algorithm we use is based on the algorithm presented in den Iseger (2006a) and den Iseger (2006b), deriving a Gaussian quadrature rule to approximate the Legendre expansion of the characteristic function. For other transform-based approaches with a view to finance applications, see for instance Carr & Madan (1999), and Lee (2004) for Fourier inversion methods, and Geman & Yor (1993) and Craddock, Heath, & Platen (2001) for Laplace inversion methods.

To test the accuracy of the Lévy ladder algorithm, we reproduce the price of a vanilla put option in the Black-Scholes model. After all, the Black-Scholes model, with a normally distributed noise term, is a special case of the Lévy models that are admitted in our algorithm. We illustrate the power of our algorithm by pricing a European put option in a variance gamma model, where the disturbance is essentially a time-changed Brownian motion, allowing for better calibration to the market. Even though a closed-form solution for this option price is available (see Madan, Carr & Chang (1998)), this solution is not easily computed. In fact, few computational results with appropriate accuracy are available in the literature. We compute our option price in two ways: once in a forward recursion by computing the risk-neutral density of the underlying process at time of expiry and solving the resulting integral, and once by calculating the option price recursively in the backwards algorithm.

The rest of this chapter is structured as follows. Section 7.2 describes the fast Lévy ladder algorithm we use to price derivative instruments. Section 7.3 then introduces the transformation of convolution integral to a Gaussian quadrature problem. In Section 7.4, we look at practical problems we encounter in the algorithm, and look at ways of further speeding up the algorithm. We briefly review the properties of the variance gamma model, and give numerical results for European call options on this model in

7.2 The ladder framework

Section 7.5. Section 7.6 concludes.

This chapter is based on the working paper de Groot, den Iseger and van der Weide (2006).

7.2 The ladder framework

In this section, we introduce the fast Lévy ladder algorithm in a general framework for the valuation of contingent claims. The underlying will be modelled as a Lévy process, the classical asset price process of geometrical Brownian motion being a special case.

7.2.1 The underlying process

The logarithm of the price $Y = (Y_t)_{t \geq 0}$ of the underlying asset will be modelled as a solution of a stochastic differential equation. A Lévy process $L = (L_t)_{t \geq 0}$ is a stochastic process with stationary and independent increments, which is continuous in probability and satisfies $L_0 = 0$. More details on Lévy processes can be found in Jacod and Shiryaev (1987) and Protter (2005). We assume the risk-neutral dynamics of $\log Y_t$ are given by

$$d \log Y_t = \mu_t dt + \sigma_t dL_t, \quad (7.2.1)$$

where the coefficient μ_t represents the (possibly time dependent) drift component of the underlying process, L_t is a Lévy process, and σ_t is the (possibly time dependent) volatility of the process L_t . We assume here that both the drift and the volatility of the process (7.2.1) are deterministic. The model can be extended to stochastic drifts and volatility, by modelling these factors as extra 'state variables'. This, however, would lead to a multidimensional model, and is therefore not considered at present.

From now on, we shall write $X_t = \log Y_t$ and $X_n = X_{t_n}$. The solution to (7.2.1) is given by

$$X_{n+1} = X_n + \mu_n + \Delta L_{n+1}, \quad (7.2.2)$$

where $\mu_n = \int_{t_n}^{t_{n+1}} \mu_s ds$ and $\Delta L_{n+1} = \int_{t_n}^{t_{n+1}} \sigma_s dL_s$. We assume that the process X_t has a density with respect to the Lebesgue measure. While this assumption does not hold true for general Lévy processes, it does hold for a significant class of processes that are interesting from a finance perspective. An explicit formula for the density may not be available, but we will have a characterization of the process in terms of its characteristic function, which is given by the Lévy-Khintchine formula (see Protter (2005)). This characteristic function is the Fourier (Laplace) transform of the density function; an inversion algorithm will supply us with an approximation of the density, should we need it.

7.2.2 Forward and backward recursion

Now consider the problem of pricing at time $t = 0$ an American (or rather, Bermudan) style contingent claim expiring or maturing at time $t = T$ with payoff function $\Phi(x)$. We will price this claim by determining the value of the underlying X_t at $N + 1$ discrete time points $0 = t_0, \dots, t_N = T$. Starting with the payoff values at t_N , we use transition densities to calculate expected payoffs at t_{N-1} , and compare those with the payoffs that would be received if the claim was exercised at time t_{N-1} . We proceed recursively until the value of the claim at $t = 0$ is found. At each time step, the claim values are calculated for a collection of points $\{y_k\}$, which reflect the possible values the underlying can take. We limit the range of possible values for y_k by first determining the probability density of the process in a forward recursion.

This approach yields a ladder framework comprising two phases, a forward phase in which the density function of the process X_t is calculated, and a backward phase in which the payoff of a contingent claim on X_t is calculated. We can write for the probability density of X_t at time t_{n+1}

$$f_{X_{n+1}}(y) = \int_{\mathbb{R}} \phi_{n+1}(y - x - \mu_n) f_{X_n}(x) dx, \quad (7.2.3)$$

where ϕ_{n+1} denotes the density of the Lévy process ΔL_{n+1} . A derivation of this formula can be found in Appendix 7.7. In other words, we can regard the density of the process X_t at time t_{n+1} as the convolution of the transition density of the process X_t from t_n to t_{n+1} , and its density at time t_n .

Now that we know which values the process X_t can obtain, we determine the payoff $\Phi(X_T)$ at these values, and assign this payoff value to V_N , the value of the option at time t_N . The calculation of the continuation value \tilde{V} of the option is given by

$$\tilde{V}_n(y) = e^{-R_n} \mathbb{E} [V_{n+1}(X_{n+1}) | X_n = y]$$

where $R_n = \int_{t_n}^{t_{n+1}} r_s ds$. This expectation can be written as an integral similar to the one for the probability density

$$\tilde{V}_n(y) = e^{-R_n} \int_{\mathbb{R}} \phi_{n+1}(x - \mu_n - y) V_{n+1}(x) dx, \quad (7.2.4)$$

A derivation of this formula can again be found in Appendix 7.7. Again, the convolution structure in (7.2.4) is evident. Path-dependent properties of the claim can now be incorporated by comparing \tilde{V} with the exercise value $\Phi_n(X_{t_n})$, i.e. setting $V_n(y) = \max(\Phi_n(y), \tilde{V}_n(y))$.

It is clear that both (7.2.3) and (7.2.4) can be discretized in a similar way. The transition densities taper off the farther away from the centre of the density we get. Hence,

7.2 The ladder framework

the integral can easily be truncated. Approximation of this truncated integral through a Gaussian quadrature rule, yields

$$G(y_k) = \sum_j w_j K(y_k - x_j), \tag{7.2.5}$$

where y_k represents the target points and x_j represents the source points in the calculations, denoted by y and x in (7.2.3) and (7.2.4) respectively. The weights w_j incorporate both the weights associated with the chosen quadrature rule, as well as the term $f_{X_n}(x_j)$ or $V_{n+1}(x_j)$. The transition densities in (7.2.3) and (7.2.4) are represented by $K(y_k - x_j)$. In this chapter, we choose the Gauss-Legendre quadrature rule for discretization of the integral. Evaluating this expression through direct summation can prove very time-consuming since the number of points and the associated operations can grow very large, say $O(N^2)$ for N source and target points. The present chapter develops an efficient algorithm for calculating these sums (7.2.5) using functional expansions and Gaussian quadrature. The operations here are in the order of $O(N)$, as we shall see in Chapter 7.3.

7.2.3 Valuing a contingent claim

With the ladder framework outlined above, we now describe the steps needed in valuing a contingent claim. Observe that we don't actually need to know the probability density of the process X in order to calculate the value of a claim (7.2.4). Instead, we use the probability density in the forward recursion phase to determine the optimal integration area to which (7.2.4) can be truncated.

An algorithmic view of the recursion framework now follows. We calculate the probability density in the forward recursion first for points in a set $B_n \subset \mathbb{R}$, and then, if possible, reduce the size of this set to some subset $C_n \subseteq B_n$, which is used as the integration boundary of the backwards recursion.

Forward recursion algorithm

Input: starting value of asset $X_0 = x_0$.

Output: $C_n, n = 0, \dots, N$.

- Step 1. $n = 1, C_0 = x_0, f_{X_1}(y) = \varphi_1(y - x_0)$ for $y \in B_1$.
- Step 2. Reduce integration area to $C_1 \subseteq B_1$ (see Section 7.4.1). Set $n = 2$.
- Step 3. Calculate $f_{X_n}(y)$ using the fast convolution algorithm for $y \in B_n$ (see Section 7.3).
- Step 4. Reduce integration area to $C_n \subseteq B_n$ (see Section 7.4.1).

Step 5. $n = n + 1$. If $n \leq N$, goto Step 3.

The first step in this forward algorithm can also be viewed in the context of (7.2.5), by realizing that all weights w_j are zero except one of unit size.

We now have information about the location of the probability mass of the process, and we have reduced the integration areas B_n accordingly to C_n . We use this information in the backwards recursion phase, as detailed below.

Backward recursion algorithm

Input: $C_n, n = 0, \dots, N$, payoff function $\Phi(x)$.

Output: V_0 .

Step 1. $n = N$, calculate $\Phi_N(x)$, $x \in C_n$. Set $V_N(x) = \Phi_N(x)$.

Step 2. Calculate the discounted continuation value of the claim $V_{n-1}(y)$ for all $y \in C_{n-1}$, using the algorithm from Section 7.3. That is, calculate

$$\tilde{V}_{n-1}(y) = \mathbb{E} \left\{ e^{-R_n} V_n(X_n) | X_{n-1} = y \right\}.$$

Step 3. Calculate the exercise value $\Phi_{n-1}(y)$ of the claim, for all $y \in C_{n-1}$. Set $V_{n-1}(y) = \max(\Phi_{n-1}(y), \tilde{V}_{n-1}(y))$.

Step 4. $n = n - 1$. If $n > 0$, go to Step 2.

Comparing the two algorithms above, we see that the forward recursion takes away a significant amount of computational effort from the backwards recursion by determining the optimal integration bounds and thus reducing the number of evaluation points N . The major advantage of this approach is that, if we want to price multiple claims all based on the same underlying processes, we need to perform the forward recursion only once, while performing the backward recursion for each claim. The same goes for calibrating the model for the underlying asset to market data. Since this implies a procedure where the claim value has to be calculated many times, it is also of great help that the forward recursion does not have to be repeated.

Starting with the next section, we will take a closer look at the implementation of particular steps in the above recursion algorithms. First, we will develop a Gaussian quadrature formula for the convolution integral, which is the core of the Lévy ladder algorithm.

7.3 The fast convolution algorithm

In this section we will discuss the calculation of sums of the form

$$G(y) = \sum_m w_m K(y - x_m) ,$$

which were introduced in Section 7.2. Here, x_m and y are source and target points respectively, and K is a function that represents the transition density of the underlying process. This work is an extension of the work in den Iseger (2006a) and den Iseger (2006b).

In this section, we will often use the definition

$$K_{x_m}(y) := K(y - x_m) . \tag{7.3.1}$$

We shall refer to K as the kernel function. Recall that we may not have an explicit expression for K , but we will have one for its Laplace transform \hat{K} . The weights w_m capture the other components that arise in discretizing the integral.

The important step we will take here is relating values of \hat{K} to the function expansion of K . This will allow us to just use \hat{K} in any calculations. We can expand any $f \in L^2(\mathbb{R})$ using a complete orthonormal base for $L^2(\mathbb{R})$ with respect to the standard inner product. In this chapter, we choose the set $\{\phi_k : k \in \mathbb{N}_0\}$ of Legendre polynomials in $L^2[0,1]$, which can be extended to the set $\{\phi_k(\cdot - j) : k \in \mathbb{N}_0, j \in \mathbb{Z}\}$, an orthonormal base for $L^2(\mathbb{R})$. Each polynomial $\phi_k : [0,1] \rightarrow \mathbb{R}$ is defined as

$$\phi_k(t) = \frac{\sqrt{2k+1}}{k!} \partial^k (t^k (1-t)^k) . \tag{7.3.2}$$

Assuming $G(y) \in L^2(\mathbb{R})$, we use these polynomials to expand $G(y)$ into

$$\begin{aligned} G(y) &= \sum_{k=0}^{\infty} \langle G, \phi_k(\cdot - j) \rangle \phi_k(y - j) \\ &= \sum_m w_m \sum_{k=0}^{\infty} \langle K_{x_m}, \phi_k(\cdot - j) \rangle \phi_k(y - j) , \end{aligned} \tag{7.3.3}$$

where $y \in [j, j+1)$. For ease of notation, we will write K_x instead of K_{x_m} from now on. We will not calculate the inner products in (7.3.3) directly, but we will retrieve them by performing the inverse Fast Fourier Transform (IFFT) algorithm on the Fourier series

$$\hat{H}(v) = \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_x, \phi_k(\cdot - j) \rangle , \tag{7.3.4}$$

with $v \in [0,1)$. We assume the source points x lie in intervals $[m, m+1)$ of unit size, where the points are identically distributed within each interval. We will decompose

the source points x into its integer part and a remainder, i.e. $x = \lfloor x \rfloor + x_r = m + x_r$. Substituting in (7.3.4) yields

$$\begin{aligned} \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_x, \phi_k(\cdot - j) \rangle &= \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_{x_r}, \phi_k(\cdot - (j + m)) \rangle \\ &= \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_{x_r}, \phi_k(\cdot - j) \rangle e^{2\pi i m v}. \end{aligned} \quad (7.3.5)$$

For the moment, we will ignore the constant term $e^{2\pi i m v}$ and just focus on calculating

$$\sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_{x_r}, \phi_k(\cdot - j) \rangle. \quad (7.3.6)$$

We want to be able to rewrite this expression in terms of the Laplace transform \widehat{K} to bypass the practical problem of possibly not having K in explicit form. As we will see later on, we can invert this Laplace transform with a very efficient algorithm. We rewrite (7.3.6) by employing the Poisson Summation Formula (PSF), which in general relates an infinite sum of function values to a sum of values of its Laplace transform. Application of the PSF to (7.3.4) yields

$$\sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_{x_r}, \phi_k(\cdot - j) \rangle = \sum_{m=-\infty}^{\infty} \widehat{K}_{x_r}(2\pi i(m + v)) \widehat{\phi}_k^*(2\pi i(m + v)). \quad (7.3.7)$$

We now introduce the space $L^2(Q_v)$. We say a function f belongs to $L^2(Q_v)$ whenever

$$\sum_{m=-\infty}^{\infty} |f(\mu_m^v)|^2 < \infty$$

holds for the sequence $\{f(\mu_m^v)\}$, with $\mu_m^v = (2\pi i(m + v))^{-1}$. On $L^2(Q_v)$, we can define the inner product $\langle \cdot, \cdot \rangle_{Q_v} : L^2(Q_v) \times L^2(Q_v) \rightarrow \mathbf{C}$ as

$$\langle f, g \rangle_{Q_v} = \sum_{m=-\infty}^{\infty} \alpha_m^v f(\mu_m^v) g^*(\mu_m^v), \quad \alpha_m^v = |\mu_m^v|^2.$$

This inner product allows us to introduce the concept of orthogonality on $L^2(Q_v)$. The sequence of polynomials $\{q_k^v\}$, defined as

$$q_k^v(s) := p_k(s) - (-1)^k e^{-2\pi i v} p_k(-s),$$

with

$$p_k(s) = \sqrt{2k+1} \sum_{j=1}^k \frac{(j+k)!}{(k-j)!} \frac{(-s)^j}{j!},$$

is a complete orthogonal base for $L^2(Q_v)$. Furthermore, we write $(\Psi f)(s) = s^{-1} f(s^{-1})$, and we find that $\Psi \widehat{\phi}_k = q_k^v$ holds with respect to $L^2(Q_v)$, that is, $\langle h \Psi \widehat{\phi}_k, g \rangle_{Q_v} = \langle h q_k^v, g \rangle_{Q_v}$ for all $h, g \in L^2(Q_v)$.

7.3 The fast convolution algorithm

Returning to (7.3.7), we observe that we can now regard the right-hand side as an inner product on the space $L^2(Q_v)$. Writing $\lambda_m^v = 2\pi i(m+v) = (\mu_m^v)^{-1}$, it follows that

$$\begin{aligned} \sum_{m=-\infty}^{\infty} \widehat{K}_{x_r}(\lambda_m^v) \widehat{\phi}_k^*(\lambda_m^v) &= \sum_{m=-\infty}^{\infty} \alpha_m^v \left(\Psi \widehat{K}_{x_r} \right) (\mu_m^v) (\Psi \widehat{\phi}_k)^* (\mu_m^v) \\ &= \left\langle \Psi \widehat{K}_{x_r}, \Psi \widehat{\phi}_k \right\rangle_{Q_v} = \left\langle \Psi \widehat{K}_{x_r}, q_k^v \right\rangle_{Q_v}. \end{aligned} \quad (7.3.8)$$

Thus we have found a way of extracting the expansion coefficients of the function K_x through calculating an inner product on the space $L^2(Q_v)$ involving only \widehat{K}_{x_r} and known polynomials q_k^v . In the rest of this section, we consider methods of computing this Q_v inner product efficiently. To achieve this, we use a Gaussian quadrature rule $\langle \cdot, \cdot \rangle_{Q_v^n}$ that approximates this inner product $\langle \cdot, \cdot \rangle_{Q_v}$.

We reintroduce the variable x_r explicitly by substituting $\widehat{K}_{x_r}(s) = e^{-sx_r} \widehat{K}(s)$, yielding

$$\begin{aligned} \left\langle \Psi \widehat{K}_{x_r}, q_k^v \right\rangle_{Q_v} &= \left\langle \Psi \left(e^{-x_r \cdot} \widehat{K} \right), q_k^v \right\rangle_{Q_v} \\ &= \left\langle \Psi \widehat{K} e^{-\frac{x_r}{\cdot}}, q_k^v \right\rangle_{Q_v}. \end{aligned} \quad (7.3.9)$$

Expanding the term $e^{-\frac{x_r}{\cdot}}$ in its Legendre expansion, and realizing that $\left\langle e^{-\frac{\cdot}{s}}, \phi_k \right\rangle = \widehat{\phi}_k\left(\frac{1}{s}\right)$, we obtain

$$\begin{aligned} \left\langle \Psi \widehat{K} e^{-\frac{x_r}{\cdot}}, q_k^v \right\rangle_{Q_v} &= \left\langle \Psi \widehat{K} \sum_{j=0}^{\infty} \widehat{\phi}_j\left(\frac{1}{\cdot}\right) \phi_j(x_r), q_k^v \right\rangle_{Q_v} \\ &= \sum_{j=0}^{\infty} \left\langle \frac{1}{\cdot} \widehat{K}\left(\frac{1}{\cdot}\right) \widehat{\phi}_j\left(\frac{1}{\cdot}\right), q_k^v \right\rangle_{Q_v} \phi_j(x_r) \\ &= \sum_{j=0}^{\infty} \left\langle \widehat{K}\left(\frac{1}{\cdot}\right) q_j^v, q_k^v \right\rangle_{Q_v} \phi_j(x_r). \end{aligned} \quad (7.3.10)$$

where we again have used that $s^{-1} \widehat{\phi}(s^{-1}) = q^v(s)$ with respect to the norm $\| \cdot \|_{Q_v}$.

We approximate the above expression by calculating $\left\langle \widehat{K}\left(\frac{1}{\cdot}\right) q_j^v, q_k^v \right\rangle_{Q_v^n}$. The Gaussian quadrature rule $\langle \cdot, \cdot \rangle_{Q_v^n}$ is given by

$$\langle f, g \rangle_{Q_v^n} = \sum_{m=1}^n \alpha_m^v f(\mu_m^v) g^*(\mu_m^v), \quad (7.3.11)$$

where μ_m^v are the zeros of q_m^v , and where the Christoffel numbers α_m^v are given by

$$\alpha_m^v = \frac{1}{\sum_{k=1}^n |q_k^v(\mu_m^v)|^2}. \quad (7.3.12)$$

The matrix representation of (7.3.11) is closely related to the matrix representation of the multiplication operator $M: L^2(Q_v) \rightarrow L^2(Q_v)$, defined as $(Mf)(s) = sf(s)$.

By the completeness of the set of polynomials $\{q_k^v\}_{k=0}^\infty$ in $L^2(Q_v)$, we have, for any $f \in L^2(Q_v)$,

$$\begin{aligned} \langle f q_j^v, q_k^v \rangle_{Q_v^n} &= \langle f(M) q_j^v, q_k^v \rangle_{Q_v^n} \\ &= f(\widehat{M}_n), \end{aligned} \quad (7.3.13)$$

where \widehat{M}_n is the $n \times n$ matrix whose k, j -th element is given by $\langle M q_j^v, q_k^v \rangle_{Q_v}$. This matrix is equal to the $n \times n$ upper block of the infinite dimensional matrix \widehat{M} . For more details on the multiplication operator M and its matrix representation. We approximate (7.3.10) with

$$\sum_{j=0}^n \langle \widehat{K}(\frac{1}{2}) q_j^v, q_k^v \rangle_{Q_v^n} \phi_j(x_r) = \sum_{j=0}^n \langle \widehat{K}(\widehat{M}_n^{-1}) q_j^v, q_k^v \rangle_{Q_v^n} \phi_j(x_r). \quad (7.3.14)$$

Our objective, however, is to compute (7.3.4). Using relation (7.3.5), we obtain the approximation

$$\sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle K_x, \phi_k(\cdot - j) \rangle = \sum_{j=0}^n \langle \widehat{K}(\widehat{M}_n^{-1}) q_j^v, q_k^v \rangle_{Q_v^n} \phi_j(x_r) e^{2\pi i m v} \quad (7.3.15)$$

What remains, at this stage, before we have retrieved $G(y_k)$, is to perform the IFFT on (7.3.15), calculate the expansion (7.3.3) and sum over all source weights w_m . Since the IFFT and the Legendre expansion are linear operations, we can interchange the order of the remaining steps. Therefore, we compute

$$\sum_{j=0}^n \langle \widehat{K}(\widehat{M}_n^{-1}) q_j^v, q_k^v \rangle_{Q_v^n} \sum_m w_m \phi_j(x_{r_m}) e^{2\pi i m v}. \quad (7.3.16)$$

The inner sum is taken over all points m . Grouping those points in the same interval I_p , where p is the interval index, chosen appropriately so as to coincide with the (integer-valued) starting point of the interval, we can rewrite this sum as

$$\sum_m w_m \phi_j(x_{r_m}) e^{2\pi i m v} = \sum_p e^{2\pi i v p} \Phi_p, \quad (7.3.17)$$

with $\Phi_p = \sum_{m \in I_p} \phi_j(x_{r_m}) w_m$. We note that this sum (7.3.17) can be calculated using an FFT operation on $\sum_m \phi_j(x_{r_m}) w_m$, and we shall refer to its result as $\widehat{\Phi}_j^v$. Since the source points x are identically distributed in each interval, the remainder terms x_{r_m} will be identical in each interval, meaning we only have to calculate the Legendre functions ϕ_j for one interval.

7.3 The fast convolution algorithm

Equation (7.3.16) now can be rewritten as a matrix equation, which reads, using the diagonalization of the matrix \widehat{M}_n (\widehat{M} is skew-symmetric),

$$\sum_m w_m \left\langle \Psi \widehat{K}_x, q_k^v \right\rangle_{Q_v^v} = V_v D(\widehat{K}(\frac{1}{\mu^v})) V_v^* \widehat{\Phi}^v \quad (7.3.18)$$

Here, μ^v are the eigenvalues of \widehat{M}_n , V_v the matrix of the associated eigenvectors, and $D(x)$ denotes the diagonal matrix with the only nonzero values the entries of the vector x on its main diagonal. Finally, the value of $G(y_k)$ is found by performing an IFFT operation on (7.3.18), and summing the result over the Legendre expansion (7.3.3).

We now give an pseudo-code description of the fast convolution algorithm. We use the double subscript notation $w_{m,i}$ to denote the m -th weight in interval i . We denote the set of intervals with I , and the number of intervals with $|I|$.

One-dimensional fast convolution algorithm

Input: Source point x_m , source weights w_m , target points y_k , order of quadrature rule n , characteristic function \widehat{K} of the kernel function, set of intervals I .

Output: $G(y_k)$.

- Step 1. Generate the eigenvalues $\mu_j^{v_k}$, $j = 1, \dots, n$, and the matrix of eigenvectors V_{v_k} , $k = 1, \dots, |I|$. Decompose x_m into $x_m = \lfloor x_m \rfloor + x_{r_m}$.
- Step 2. Evaluate the Legendre functions $\phi_j(y_k)$ and $\phi_j(x_{r_m})$ for $j = 0, 1, \dots, n$.
- Step 3. Compute $\Phi_{j,i} = \sum_m \phi_j(x_{r_m}) w_{m,i}$ for $j = 1, \dots, n$, and $i \in I$.
- Step 4. $\widehat{\Phi}_j^{v_k} = \text{FFT}(\Phi_j)$ where $\Phi_j = \{\Phi_{j,i}, i \in I\}$, and $v_k = \frac{2\pi k}{|I|}$, $k = 0, \dots, |I| - 1$.
- Step 5. $\widehat{H}_j^{v_k} = V_{v_k} D(\widehat{K}(\frac{1}{\mu^{v_k}})) V_{v_k}^* \widehat{\Phi}^{v_k}$, where $\widehat{\Phi}^{v_k} = \{\widehat{\Phi}_1^{v_k}, \dots, \widehat{\Phi}_n^{v_k}\}^T$, for each v_k , $k = 0, \dots, |I| - 1$.
- Step 6. $H_j = \text{IFFT}(\widehat{H}_j)$, where $\widehat{H}_j = \{\widehat{H}_j^{v_1}, \dots, \widehat{H}_j^{v_{|I|}}\}$, for each $j = 1, \dots, n$.
- Step 7. $G(y_k) = \phi^T(y_k) H$, where both $\phi(y_k)$ and H have to be interpreted as vectors over j .

Thus, in step 4 the FFT is taken with respect to the intervals with index i . The IFFT in step 6 is taken with respect to the parameters v_k , $k = 1, \dots, |I|$.

Step 6 yields the coefficients of the Legendre expansion of G in the point y_k and is no longer dependent on the points x . Thus we do not have to run the above algorithm every time if only the expansion of G is used.

From the pseudo-code above, it is clear that the computational complexity of the algorithm is $O(N)$ with respect to the source and target points. The FFT operation in step 4 is $O(|I| \log |I|)$, with respect to the number of intervals $|I|$.

Remark. We have assumed in this section that the source points were equally distributed in intervals of unit size. The algorithm also works when the points are generally distributed, since the decomposition into an integer and remainder part can still be made. It will, however, be necessary to calculate the Legendre functions ϕ_j in (7.3.17) for all x_{r_m} . This means our algorithm is suitable for different situations as well, for instance when the source and target points have been generated by Monte Carlo simulation.

7.4 Implementational issues of the ladder framework

In the previous section, we outlined the core of the ladder framework, the fast convolution algorithm. In this section, we will discuss further improvements that can be made to the ladder framework, and address issues that arise during computations.

7.4.1 Integration area restriction

As we have seen in Section 7.2.3, the forward recursion in the ladder framework involves a step where we reduce the area B_n over which the probability density is calculated to the area $C_n \subseteq B_n$. To achieve this, we divide the integration area into smaller intervals, and calculate the probability density for each of these intervals. Recall that we already used this division in intervals when we were decomposing the source points into an integer part and a remainder part. Now, supposing we have a desired level of accuracy ε (i.e. the minimum amount of probability mass we wish to capture is $1 - \varepsilon$), we search for the smallest union of intervals which will yield this precision. We are hereby able to reduce the number of evaluation points and enhance the algorithm performance.

The integration area reduction procedure boils down to the determination of so-called ε -support sets. We say that C_n is an ε -support set for the function f when

$$\int_{C_n} f(t) dt \geq (1 - \varepsilon) \|f\|_1.$$

In this case, since f is a probability density function, we have $\|f\|_1 = 1$. We now determine the amount of probability mass contained in each interval by integrating the probability density function f_{X_n} and sum the contributions for each interval to find the total probability mass in B_n . Observe that

$$\int_{C_n} f(t) dt = \sum_j \langle f, \phi_0(\cdot - j) \rangle,$$

7.4 Implementational issues of the ladder framework

and that we thus already have calculated these inner products, and by extension, these integrals. Denoting the total probability mass in B_n with μ , we determine the smallest union of intervals that will still yield the desired precision $(1 - \varepsilon)\mu$. For practical purposes, we prefer this union of intervals to be all adjacent. For most processes, this poses no problem since the probability mass has a single peak decreasing gradually on either side.

The integration area reduction algorithm, which is taken from den Iseger and Oldenkamp (2006a), now works as follows. We denote by μ_i the probability mass captured by the interval with index i . Starting from the left side of the integration area, we proceed to the right and add intervals until the cumulative probability mass $\hat{\mu}$ reaches $(1 - \varepsilon)\mu$. Then we remove intervals starting from the left edge until we fall below $(1 - \varepsilon)\mu$. Then, we again proceed to the right and add and remove intervals until we have reached the right boundary of the integration area. The desired smallest integration area C_n is now set equal to the union containing the fewest number of intervals.

Below we give a representation of the algorithm in pseudo-code. We assume, as in Section 7.3, that the intervals are of unit size. The variables μ_L and μ_U contain the probability mass of the lower and upper interval of the union respectively.

Integration area reduction algorithm

L : Lower bound of current integration area.

U : Upper bound of current integration area.

L^* : Lower bound smallest integration area.

U^* : Upper bound smallest integration area.

$B_n = [L_f, U_f]$.

Step 1. Initialization: $\hat{\mu} = 0, L = L_f, U = L_f, L^* = L_f, U^* = U_f, \Delta = 1$.

Step 2. while $(\hat{\mu} < (1 - \varepsilon)\mu$ and $U < U_f$), do $(\hat{\mu} = \hat{\mu} + \Delta\mu_U, U = U + 1)$.

Step 3. while $(\hat{\mu} \geq (1 - \varepsilon)\mu)$, do $(\hat{\mu} = \hat{\mu} - \Delta\mu_L, L = L + 1)$.

Step 4. If $(\hat{\mu} \geq (1 - \varepsilon)\mu$ and $U - (L - 1) < U^* - L^*)$, then $(L^* = L - 1$ and $U^* = U)$.

Step 5. If $U \leq U_f - 1$, go to Step 2, else set $C_n = [L^*, U^*]$ and **end**.

7.4.2 Non-smoothness of payoff function

The quadrature integration method we used in Section 7.3 to approximate functions are only accurate for smooth functions. The payoff function one comes across in practice are usually non-smooth through a discontinuity in the function itself or one of its

derivatives, meaning there are intervals containing points where the payoff function is not differentiable. However, if the non-smoothness occurs at the edge of the interval instead of its interior, the quadrature formulas will still be accurate. Therefore, we will adapt the intervals to the discontinuities.

Identification of the non-smooth point will be done by a root finding algorithm. Once found, we can determine the interval in which this point lies. We adapt the intervals to this point by taking this point as the barrier between two intervals. We shift the intervals to either side of the point so as to create new intervals, all of unit size. Effectively, this means the number of intervals increases by one.

7.4.3 Lapped window transforms

As we know, the value of an American option at any time is the maximum of the exercise value and the continuation value of the option. The previous subsection detailed a method that ensures the payoff is calculated accurately by splitting the non-smooth payoff function into subfunctions where the function is smooth. The continuation value of the option at time t_n is determined by calculating the convolution of the kernel function K and the continuation value of the option at time t_{n+1} . The kernel function K is sharply peaked, especially for small time steps. This means that the continuation value of the option at any time is either the convolution of this sharply peaked function with a non-smooth function, or the convolution of such (series of) convolutions. The resulting function varies significantly over a relatively small interval, and remains fairly smooth over a large interval. This poses a computational challenge. In order to integrate this function accurately over the entire range, we would need a large number of points owing to the highly varying behaviour over the small interval. For the smooth part of the function, this number of points is far more than we need to integrate accurately. Therefore, we want to apply some way of varying the number of integration points over the domain of the function.

The method we apply to tackle this problem is the so-called lapped window transform, a technique frequently applied in signal processing (see, for instance, Mallat (2001)). The idea of the lapped window transform is to decompose a function f which we wish to integrate into a sum of subfunctions f_k , where its behaviour is either fairly smooth or highly varying. We achieve this by multiplying f by a series of slightly overlapping windows $\alpha_k(t) := \alpha(t - k)$. Here α is a given function, the so called a profile functions. The integral of f is now the sum of the integrals of the windowed functions $f_k = f\alpha_k^2$, which means we can vary the number of integration points per f_k . The number of different windows we need is then determined by the behaviour of f over its domain.

7.4 Implementational issues of the ladder framework

Figure 7.1 shows a typical case with three overlapping windows α_k . For any point x in this figure, we have

$$\sum_{k=1}^3 \alpha_k^2(x) = 1.$$

The profile function α which generates the window functions in the Figure 7.1 is a smooth function up to the $(2^j - 1)$ -th derivative, where the parameter j indicates the j -th function in the recursion

$$\alpha_0(x) = \sin\left(\frac{\pi}{4}(1+x)\right), \quad \alpha_{j+1}(x) = \alpha_j\left(\sin\left(\frac{\pi x}{2}\right)\right). \quad (7.4.1)$$

and the profile function α is given by $\alpha(x) = \alpha_j(2x)$ for $-0.5 < x < 1.5$ and $\alpha(x) = 0$ else. We can therefore determine the smoothness of the functions f_k by choosing j appropriately. In our chapter, we use the profile function α_5 .

Practically speaking, for a payoff function with one non-smooth point that is convolved with the kernel function, we need three windows: one for the varying part of f , which we shall name α_2 , and one on either side of this window, α_1 and α_3 . The decomposition of f is found by multiplying it by the square of the windows. After this multiplication, we apply the fast convolution transform to each component with different parameters. Summing the components yields the fast convolution transform of the original function.

7.4.4 Peaked distributions

One of the appealing features of the ladder algorithm is that it can handle underlying values modelled by Lévy processes. These processes can exhibit sharper peaks and heavier tails in their distributions than Brownian motion (normal distributions). These Lévy distributions pose a computational challenge since the numerical integration of the convolution integral requires a large number of points for those parts of the integration area where the integrand is not well behaved. We tackle this problem by presenting an adaptive algorithm that is able to vary the size of the intervals of the Legendre expansions according to need.

Suppose that the ladder algorithm yields an integration area of $[a, b]$. That is, the integrand has ε -support on the interval $[a, b]$. We assume the integrand is not well-behaved (i.e. sharply peaked) on a subinterval of $[a, b]$. We start by dividing the interval $[a, b]$ into N subintervals of size Δ . We calculate the projection of the Legendre expansion of the convolution and a refined projection on subintervals of size $\Delta/2$. We then compare these two projections, and, where sufficiently close, we decide that a Legendre expansion with interval size Δ is accurate enough. Where this is not the case, we calculate a refined projection on these subintervals with $\Delta/4$, and compare this

projection with the one for $\Delta/2$. We proceed along these lines, refining the projections for ever smaller intervals until the expansion on the whole of $[a, b]$ can be efficiently computed.

Taken at face value, this approach leads to an ever increasing number of evaluation points for the Legendre expansions. To combat this, we restrict the refined approximations to those subintervals where an improvement is necessary, and stick with our earlier approximations where possible. In the following, we will expand on this refinement method.

Computing the projection of the Legendre expansion amounts to determining the expansion coefficients $\langle K_{x,\Delta}, \phi_m(\frac{\cdot}{\Delta} - j) \rangle$. As we showed in Section 7.3, these coefficients can be retrieved from the Gaussian quadrature rule $\langle \Psi \widehat{K}_{x,\Delta}, q_m^v \rangle_{Q_v^v}$. However, when applying an N -point inverse fast Fourier transform (IFFT) to this Gaussian quadrature rule, we obtain an infinite sum of coefficients, viz

$$\text{IFFT} \left(\langle \Psi \widehat{K}_{x,\Delta}, q_m^v \rangle_{Q_v^v} \right) = \sum_{k=-\infty}^{\infty} \langle K_{x,\Delta}, \phi_m \left(\frac{\cdot}{\Delta} - j - kN \right) \rangle, \quad (7.4.2)$$

while we only need the value of this sum for $k = 0$. We note that, for those values of k which imply K_x is evaluated outside $[a, b]$, the coefficient can be ignored, since K_x is evaluated outside of its ε -support. For projections on the interval $[a, b]$, this means that (7.4.2) produces only our desired coefficient. As soon as we are only looking at a subinterval of $[a, b]$, then we will encounter entries in the sum in (7.4.2) that are outside this subinterval, but inside the ε -support of K_x , and therefore contribute significantly to the sum. We compensate for this by calculating the terms of the sum for $k \neq 0$, and subtracting these from (7.4.2). These terms are easily obtained, since we already have an accurate calculation of them.

Let us introduce some notation. Let N_0 be the original number of subintervals in $[a, b]$, and Δ_0 their width (hence, $b - a = N_0 \Delta_0$). Furthermore, let the subinterval width $2^{-p} \Delta_0$ be denoted as Δ_p , and N_p the related number of subintervals at this level. The infinite sums of coefficients $S_m^0 K_{x,\Delta}$ and $S_m K_{x,\Delta}$ are given by

$$\begin{aligned} S_m^0 K_{x,\Delta_p}(j) &= \sum_{k=-\infty}^{\infty} \left\langle K_x, \phi_m \left(\frac{\cdot}{\Delta_p} - j - kN_p \right) \right\rangle \\ S_m K_{x,\Delta_p}(j) &= \sum_{k \neq 0} \left\langle K_x, \phi_m \left(\frac{\cdot}{\Delta_p} - j - kN_p \right) \right\rangle. \end{aligned}$$

We can look at $S_m K_{x,\Delta_p}(j)$ as a periodic function of period $N_p \Delta_p$, corresponding with the m -th degree Legendre polynomial in the orthogonal expansion. The integer j can be seen as the index of the subinterval considered. Since the sums $S_m^0 K_{x,\Delta_p}(j)$ are calculated through an IFFT operation, it is optimal to choose the initial number of subintervals N_0 in $[a, b]$ as a power of 2. Obviously, the desired coefficient $\langle K_x, \phi_m(\frac{\cdot}{\Delta_p} - j) \rangle$

7.4 Implementational issues of the ladder framework

is given by

$$\left\langle K_x, \phi_m \left(\frac{\cdot}{\Delta_p} - j \right) \right\rangle = S_m^0 K_{x, \Delta_p}(j) - S_m K_{x, \Delta_p}(j) .$$

As we mentioned before, the sum $S_m^0 K_{x, \Delta_p}(j)$ is obtained from the Gaussian quadrature rule $\left\langle \Psi \widehat{K}_{x, \Delta_p}, q_m^v \right\rangle_{Q_m^v}$ by applying the inverse fast Fourier transform to it. We can derive a recursive relation for the sums $S_m K_{x, \Delta_p}(j)$ when we are refining the subinterval size, depending on the properties of the projections. We discriminate between two cases. First, when the initial projection is inaccurate for the whole interval considered, then we refine the whole of the projection, the number of subintervals doubles, and we obtain

$$\begin{aligned} S_m K_{x, \Delta_{p+1}}(2j) &= A_m S_m K_{x, \Delta_p}(j) \\ S_m K_{x, \Delta_{p+1}}(2j+1) &= B_m S_m K_{x, \Delta_p}(j) , \end{aligned} \quad (7.4.3)$$

for $j = 0, \dots, N_p - 1$. Here, A_m and B_m are row vectors taken from the $n \times n$ matrices A and B , which represent the projection of a coefficient on its left half and right half subinterval respectively. The (k, j) -th element of A and B are given by

$$\begin{aligned} A_{k,j} &= \int_0^1 \phi_j\left(\frac{u}{2}\right) \phi_k(u) du \\ B_{k,j} &= \int_0^1 \phi_j\left(\frac{1+u}{2}\right) \phi_k(u) du . \end{aligned} \quad (7.4.4)$$

The second case arises when part of the projection is sufficiently accurate, and part of it needs refinement. In this case we need a second term in the right hand sides of (7.4.3). Suppose that from the $N = 2^\ell$ initial subintervals, $2^q, 0 \leq q \leq \ell$ need refinement. Then the number of refined subintervals that need evaluation is 2^{q+1} . This yields the following recursion.

$$\begin{aligned} S_m K_{x, \Delta_{p+1}}(2j) &= A_m \left(S_m K_{x, \Delta_p}(j) + \sum_{k=1}^{2^q-1} S_m^0 K_{x, \Delta_p} \left(\left(j + \frac{kN_p}{2^q} \right) \bmod N_p \right) \right) \\ S_m K_{x, \Delta_{p+1}}(2j+1) &= B_m \left(S_m K_{x, \Delta_p}(j) + \sum_{k=1}^{2^q-1} S_m^0 K_{x, \Delta_p} \left(\left(j + \frac{kN_p}{2^q} \right) \bmod N_p \right) \right) , \end{aligned}$$

for $j = 0, \dots, 2^q - 1$. It is important to note that the number of subintervals which are further refined is an exponent of 2, to facilitate efficient computation of $S_m K_{x, \Delta_p}(j)$ by means of the IFFT. Furthermore, it is important that these subintervals are connected. In our case, this is no problem since the distributions we consider are unimodal.

Remark. In our case, we apply this adaptive refinement algorithm to a windowed version of the function K_x . The reason for this is that our algorithm requires smoothness of the integrated function on its entire domain in order for approximations to be

locally accurate. By multiplying K_x by a window function w , we obtain a function that is smoother than the original function K_x . See Appendix 7.8 for details.

Remark. We encounter a similar problem in the backward algorithm when computing payoff functions. These functions are generally discontinuous or non-smooth (think for instance the non-differentiability of the payoff function for a vanilla call or put in its strike), which again means that better detailed approximations are necessary. We could solve this by applying the adaptive refinement algorithm to the problem. However, in the case of simple payoff functions such as those occurring with vanilla calls and puts, it is just as good to split the payoff function into two parts where they are completely smooth (thus lending themselves to conventional evaluation).

7.5 Applications

7.5.1 Black-Scholes model

We test the accuracy of our algorithm by calculating the price of a European put option in the Black-Scholes model. The log price dynamics $X(t) = \log S(t)$ of the underlying asset $S(t)$ in this model are given by

$$dX(t) = (r - q)dt + \sigma dW(t), \tag{7.5.1}$$

where r and q are the risk-free interest rate and the continuous dividend yield respectively, σ is the volatility of the asset and $W(t)$ is a Brownian motion. It is clear that this model fits nicely in our general model description (7.2.1). The characteristic function of this asset process, for use in our fast Lévy ladder algorithm, is given by

$$\phi_{X(t)}(s) = \exp \left\{ t \left(\frac{1}{2} \sigma^2 s^2 - (r - q - \frac{1}{2} \sigma^2) s \right) \right\}. \tag{7.5.2}$$

Denoting the asset price at time $t = 0$ as $S(0) = e^{X(0)}$, the value of the put option at time $t = 0$ is

$$P_{BS}(S(0); K, T) = e^{-(r-q)T} \mathbb{E}_Q \left[\left(K - e^{X(T)} \right)^+ \right], \tag{7.5.3}$$

where \mathbb{E}_Q denotes expectation taken with respect to the risk-neutral measure, and the notation x^+ denotes $\max(x, 0)$. The analytical price for this contract is given by

$$\begin{aligned} P_{BS}(S(0); K, T) &= Ke^{-(r-q)T} N(-d_2) - S(0) N(-d_1) \\ d_1 &= \frac{1}{\sigma\sqrt{T}} \left(\log \left(\frac{S(0)}{K} \right) + \left((r - q) + \frac{\sigma^2}{2} \right) T \right) \\ d_2 &= d_1 - \sigma\sqrt{T}, \end{aligned} \tag{7.5.4}$$

where N is the normal cumulative distribution.

7.5 Applications

The numerical results of our algorithm are compared with the analytical Black-Scholes price (7.5.4) in Table 7.1. The number of steps indicate the number of time steps that are used. For an option with a maturity one year from now, we can say 256 time steps roughly corresponds to one evaluation per day, and 16384 to one every eight minutes. The columns for the forward recursion imply the calculation of the risk-neutral probability density at the expiry date, at which point the price is calculated explicitly. In the backwards recursion, the option is priced according to the backwards algorithm detailed in Section 7.2.3. The columns marked error give the difference between the analytic price, and our price.

Table 7.1: European put option in Black-Scholes model ($S(0) = 50, K = 50, r = 0.05, q = 0.03, T = 1, \sigma = 0.2$). Analytic price: 3.3654588245816521.

# Steps	Forward	Error	Backward	Error
64	3.3654588245681962	-1.3456e-011	3.3654588245684458	-1.3206e-011
256	3.3654588245211507	-6.0501e-011	3.3654588245223827	-5.9269e-011
1024	3.3654588242664603	-3.1519e-010	3.3654588242683054	-3.1335e-010
4096	3.3654588210398861	-3.5418e-009	3.3654588210527359	-3.5289e-009
16384	3.3654588068792552	-1.7702e-008	3.3654588068918994	-1.7690e-008

As we can see from Table 7.1, the values for the forward and backward algorithm are almost identical. What’s more, even though there is a slight loss of accuracy with increasing number of time steps, the method remains stable.

7.5.2 Variance gamma model

As we saw above, the classical Black-Scholes model uses a lognormally distributed asset as the underlying for the option. This approach has known drawbacks; for instance, due to the constant implied volatility, the more complex volatility structure observed in practice can not adequately be modelled. One way of addressing this issue is by modelling the underlying asset with a variance gamma (VG) model, which is an example of a Lévy process.

The variance gamma model can be seen as a modified Brownian motion, where the time of evaluation g of the Brownian motion is modulated by a gamma process γ with unit mean. That is, if we write $b(t; \theta, \sigma) = \theta t + \sigma W(t)$ for a Brownian motion with drift, we write for the variance gamma model $L(t; \sigma, \theta, \nu) = b(\gamma(t; 1, \nu), \theta, \sigma)$. Since the gamma process is an increasing process, we can view g as a realization of a time-stretching process. The density of the gamma process $\gamma(t)$ of independent increments

over non-overlapping intervals of size h , with mean μh and variance νh , is given by

$$f_h(g) = \left(\frac{\mu}{\nu}\right)^{\frac{\mu^2 h}{\nu}} \frac{g^{\frac{\mu^2 h}{\nu}-1} \exp\left(-\frac{\mu}{\nu} g\right)}{\Gamma\left(\frac{\mu^2 h}{\nu}\right)}, \quad g > 0. \quad (7.5.5)$$

The density of the VG process $L(t)$ is now found by conditioning the density of the Brownian motion on the random time g ,

$$\begin{aligned} f_{L(t)}(L) &= \int_0^\infty \frac{1}{\sigma\sqrt{2\pi g}} \exp\left\{-\frac{(L-\theta g)^2}{2\sigma^2 g}\right\} f_t(g) dg \\ &= \int_0^\infty \frac{1}{\sigma\sqrt{2\pi g}} \exp\left\{-\frac{(L-\theta g)^2}{2\sigma^2 g}\right\} \frac{g^{\frac{t}{\nu}-1} \exp\left(-\frac{g}{\nu}\right)}{\nu^{\frac{t}{\nu}} \Gamma\left(\frac{t}{\nu}\right)} dg. \end{aligned} \quad (7.5.6)$$

The Fourier transform of the VG process $L(t)$, which we will need for the fast convolution algorithm to work, is

$$\phi_{L(t)}(s) = \left(\frac{1}{1 + \theta \nu s - \frac{\sigma^2 \nu s^2}{2}}\right)^{\frac{t}{\nu}}. \quad (7.5.7)$$

These results can be found in, for instance, Madan, Carr & Chang (1998) and Carr, Geman, Madan & Yor (2002).

The basics of the VG process now known, we can define the asset process $S(t)$. Instead of the classical model of the asset as a Brownian motion with drift, we replace the Brownian motion in the risk-neutral Black-Scholes model with the VG process defined above, obtaining

$$S(t) = S(0) \exp\{(r - q)t + L(t; \sigma, \theta, \nu) + \omega t\}, \quad (7.5.8)$$

where r and q denote the risk-free interest rate (assumed constant here) and the continuous dividend yield respectively. The term ωt is added to ensure the asset process is a martingale. In order to achieve this, we need to set ω as follows (see Madan, Carr & Chang (1998))

$$\omega = \frac{1}{\nu} \log\left(1 - \theta \nu - \frac{1}{2} \sigma^2 \nu\right). \quad (7.5.9)$$

Writing the asset process in its log-form $X(t) = \log S(t)$, we obtain

$$X(t) = X(0) + (r - q + \omega)t + L(t), \quad (7.5.10)$$

and we see that this form corresponds to the one set in (7.2.1).

We can now also interpret the variables present in the model, σ , ν , and θ . The parameter σ still denotes volatility of the asset process, while ν influences kurtosis (which influences the peakedness or heavy-tailedness of the distribution) and θ influences the

7.5 Applications

skewness (the asymmetry in the distribution tails). Madan, Carr & Chang (1998) show that these parameters can replicate the volatility smile and risk-averseness observed in the market.

The next step is to calibrate the model using real market data. Typical values of the different model parameters are given in Table 7.2. These values are taken from Hirsra & Madan (2003), and they were found by calibrating the explicitly known formula for European call options on variance gamma assets to the S&P 500 index on June 30, 1999. The calibration took place over a range of strikes, thus incorporating smile and skew effects. The parameter ν covers quite a wide range of values, implying that this extra degree of freedom is useful.

Table 7.2: Calibrated VG parameters for S&P 500 June 30, 1999

Case	T	r	q	σ	ν	θ
1	0.13972	0.0533	0.011	0.17875	0.13317	-0.30649
2	0.21643	0.0536	0.012	0.18500	0.22460	-0.28837
3	0.46575	0.0549	0.011	0.19071	0.49083	-0.28113
4	0.56164	0.0541	0.012	0.20722	0.50215	-0.22898

We will calculate the value of a European put option. For a European put option, the value at time $t = 0$ is given by

$$P_{VG}(S(0); K, T) = e^{-rT} \mathbb{E}_Q \left[\left(K - e^{X(T)} \right)^+ \right]. \tag{7.5.11}$$

The results of our computation are given in Table 7.3, where we have also given the price of a European put option in the Black-Scholes model with the same parameters (that is, r , q , and σ).

Table 7.3: European put option in the variance gamma model ($S(0) = 50, K = 50$)

Case	Black-Scholes	Forward	Backward	Error
1	1.1846487593467236	1.2791259592630524	1.2791259592577380	5.3144e-012
2	1.4899255446291311	1.6848023555537279	1.6848023555631324	-9.4045e-012
3	2.0829757621733389	2.7414272154190451	2.7414272153759875	4.3058e-011
4	2.4927037738608298	2.8856273710697731	2.8856273710772280	-7.4549e-012

The values for the columns forward and backward are calculated in the same manner as for Table 7.1. The column marked error now contains the difference between these forward and backward values. We see that the values of the forward and the backward algorithm practically coincide. Furthermore, we notice that the variance gamma model prices put options consistently more expensive than the Black-Scholes

model. Therefore, we conclude that the variance gamma model is a useful addition to the finance practice.

7.5.3 Greeks for Lévy processes

A stochastic process is uniquely determined by its joint finite-dimensional densities (Kolmogorov extension theorem, see for instance Billingsley (1995)). That is, given a finite sequence of times $\{t_k\}$ on the interval $[0, T]$, the process X_t is completely determined by the joint densities of all $\{X_{t_k}, t_k \in [0, T]\}$. An equivalent determination of the process X_t is given by the characteristic function of the process

$$\mathbb{E}_{\mathbb{Q}_1} [\exp(-\langle s, \Delta X \rangle)] .$$

Here the inner product in the exponent is given by

$$\langle s, \Delta X \rangle = \sum_{k=1}^K s_k (X_{t_{k+1}} - X_{t_k}) ,$$

a sum of the increments of the process. We want to apply a change of measure that will transform the process X_t with respect to a risk-neutral martingale measure \mathbb{Q}_1 into a process with respect to measure \mathbb{Q}_2 . We use the Radon-Nikodym derivative, which is of the form

$$\exp(\langle \alpha, \Delta X \rangle) C(\alpha) , \tag{7.5.12}$$

where $C(\alpha)$ is a normalisation parameter that ensures the new measure \mathbb{Q}_2 is still a probability measure, i.e.

$$\mathbb{E}_{\mathbb{Q}_1} [\exp(\langle \alpha, \Delta X \rangle) C(\alpha) | \mathcal{F}_0] = 1 , \tag{7.5.13}$$

with $(\mathcal{F}_t)_{t \geq 0}$ the filtration generated by the process X_t . This way, the characteristic function of X_t expressed in terms of the new measure \mathbb{Q}_2 is

$$\mathbb{E}_{\mathbb{Q}_2} [\exp(-\langle s, \Delta X \rangle)] = \mathbb{E}_{\mathbb{Q}_1} [\exp(-\langle s - \alpha, \Delta X \rangle) C(\alpha)] .$$

Let us illustrate the above by finding the delta of a European option on an asset whose price process S_t is given by

$$S_t = \frac{N_t}{N_0} M_t = \frac{N_t}{N_0} S_0 C_t \exp(\langle \sigma, \Delta X \rangle) , \tag{7.5.14}$$

with N_t a numéraire process and C_t an \mathcal{F}_t -adapted drift process. This model is consistent with the log-dynamics described in (7.2.1). The value of the option at time t is given by $V(t)$. The risk-neutral price of this option at $t = 0$ is given by

$$V(0) = \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} V(T) \middle| \mathcal{F}_0 \right] = \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \Phi(S_T) \middle| \mathcal{F}_0 \right] ,$$

7.5 Applications

where Φ is the payoff function associated with the option. The asset return R_t is defined as $R_t = \frac{S_t}{S_0}$, and is independent of S_0 since X_t is a Lévy process. The delta of the option can now be written as

$$\Delta = \frac{\partial V(0)}{\partial S_0} = \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \Phi'(S_T) \frac{\partial S_T}{\partial S_0} \middle| \mathcal{F}_0 \right] = \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \Phi'(S_T) \frac{S_T}{S_0} \middle| \mathcal{F}_0 \right]. \quad (7.5.15)$$

However, since every N_t -numerated asset is a martingale with respect to \mathbb{Q}_1 , we have

$$\mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_t} \frac{S_t}{S_0} \middle| \mathcal{F}_0 \right] = 1 \quad \forall t \geq 0, \quad (7.5.16)$$

and since $\frac{N_0}{N_t} \frac{S_t}{S_0} > 0$ (both the numéraire process and the asset price process are positive by definition), we see that $\frac{N_0}{N_t} \frac{S_t}{S_0} = \frac{N_0}{N_t} R_t$ satisfies the requirements to be a Radon-Nikodym derivative. We now define a new probability measure \mathbb{Q}_2 such that $\frac{d\mathbb{Q}_2}{d\mathbb{Q}_1} = \frac{N_0}{N_t} R_t$ is the Radon-Nikodym derivative. Hence we are able to rewrite (7.5.15) with respect to this new measure

$$\Delta = \mathbb{E}_{\mathbb{Q}_2} [\Phi'(S_T) | \mathcal{F}_0]. \quad (7.5.17)$$

In the context of the characteristic functions above, we write

$$\exp(\langle \alpha, \Delta X \rangle) C(\alpha) = C_t \exp(\langle \sigma, \Delta X \rangle),$$

and thus

$$\mathbb{E}_{\mathbb{Q}_2} [\exp(-\langle s, \Delta X \rangle)] = \mathbb{E}_{\mathbb{Q}_1} [\exp(-\langle s - \alpha, \Delta X \rangle) C_t].$$

The behaviour of the asset process under the new measure now follows from the shifted Laplace transform of the process under the risk-neutral measure, which is already available.

In a similar manner we can derive an expression for the gamma of a European option. Again, we consider the same asset price process (7.5.14) for S_t . The general formula for the gamma of an option is

$$\Gamma = \frac{\partial^2}{\partial S_0^2} V(t) = \frac{\partial^2}{\partial S_0^2} \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \Phi(S_T) \middle| \mathcal{F}_0 \right].$$

Moving the differentiation operator within the expectation and recognising that the Radon-Nikodym derivative that transforms the risk-neutral measure \mathbb{Q}_1 into measure \mathbb{Q}_3 is given by

$$\frac{d\mathbb{Q}_3}{d\mathbb{Q}_1} = \frac{\exp(\langle 2\sigma, \Delta X \rangle)}{\mathbb{E}_{\mathbb{Q}_1} [\exp(\langle 2\sigma, \Delta X \rangle)]},$$

we obtain

$$\begin{aligned}
 \Gamma &= \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \frac{\partial^2 \Phi(S_T)}{\partial S_T^2} \left(\frac{\partial S_T}{\partial S_0} \right)^2 \middle| \mathcal{F}_0 \right] \\
 &= \mathbb{E}_{\mathbb{Q}_1} \left[\frac{N_0}{N_T} \frac{\partial^2 \Phi(S_T)}{\partial S_T^2} S_0^2 C_t^2 \exp(\langle 2\sigma, \Delta X \rangle) \middle| \mathcal{F}_0 \right] \\
 &= \mathbb{E}_{\mathbb{Q}_1} [\exp(\langle 2\sigma, \Delta X \rangle)] \mathbb{E}_{\mathbb{Q}_3} \left[\frac{N_0}{N_T} \frac{\partial^2 \Phi(S_T)}{\partial S_T^2} S_0^2 C_t^2 \middle| \mathcal{F}_0 \right].
 \end{aligned}$$

The \mathbb{Q}_1 expectation is just a shifted version of a Laplace transform we already have.

Consider now, for instance, the variance gamma model introduced in Section 7.5.2. An asset S_t based on the variance gamma model satisfies

$$S_t = S_0 \exp((r - q + \omega)t + L(t; \sigma, \theta, \nu)),$$

where $\omega = \frac{1}{\nu} \log(1 - \theta\nu - \frac{1}{2}\sigma^2\nu)$ guarantees that S_t is a martingale with respect to \mathbb{Q}_1 , and $L(t; \sigma, \theta, \nu)$ is the actual variance gamma process. We have

$$\exp(\langle \alpha, \Delta X \rangle) C(\alpha) = \exp((r - q + \omega)t + L(t; \sigma, \theta, \nu)) \frac{N_0}{N_t},$$

and thus we conclude in this case

$$\exp(\langle \alpha, \Delta X \rangle) = \exp(L(t; \sigma, \theta, \nu))$$

and

$$C(\alpha) = \exp((r - q + \omega)t) \frac{N_0}{N_t}.$$

7.6 Conclusion

In this chapter we presented a recursive ladder algorithm to price derivatives under Lévy processes, leading to a series of convolutions of the transition density of the underlying asset process and a known function. The convolution integral is calculated by using a Laplace inversion algorithm to compute its Legendre expansion. The algorithm uses just the characteristic function of the Lévy process, meaning it is even applicable to processes for which there is no explicit expression for the density. The computational complexity of $O(N)$, where N is the number of evaluation points in each time step, makes the algorithm an attractive alternative to other presently known pricing methods. The extension of the traditional Brownian motion to Lévy models as disturbance term in asset models allows us to price derivatives in a way that is more consistent with the observed market behaviour of assets. The algorithm is shown to be accurate and stable by pricing European options on assets driven by the variance gamma model.

7.7 Technical note: The probability density $f_{X_{n+1}}(x)$

7.7 Technical note: The probability density $f_{X_{n+1}}(x)$

Recall equation (7.2.2)

$$X_{n+1} = X_n + \mu_n + \Delta L_{n+1} .$$

The conditional distribution of X is now given by

$$\begin{aligned} P(X_{n+1} \leq y | X_n) &= P(\Delta L_{n+1} \leq \sigma_n^{-1}(y - X_n - \mu_n) | X_n) \\ &= \Phi_{n+1}(y - X_n - \mu_n) , \end{aligned}$$

where Φ_{n+1} denotes the distribution function of the random variable ΔL_{n+1} , and where we have used the independent increments of L . The distribution of X is found by integrating with respect to the density of X :

$$\begin{aligned} F_{X_{n+1}}(y) &= \int P(X_{n+1} \leq y | X_n = x) f_{X_n}(x) dx \\ &= \int \Phi_{n+1}(y - x - \mu_n) f_{X_n}(x) dx . \end{aligned}$$

Differentiation of $F_{X_{n+1}}(y)$ with respect to y now yields the density of X_{n+1} :

$$f_{X_{n+1}}(y) = \int \phi_{n+1}(y - x - \mu_n) f_{X_n}(x) dx ,$$

where ϕ_{n+1} denotes the density of X_n .

The calculation of the continuation value \tilde{V} of the option is given by

$$\begin{aligned} \tilde{V}_n(y) &= e^{-R_n} \mathbb{E} [V_{n+1}(X_{n+1}) | X_n = y] \\ &= e^{-R_n} \mathbb{E} [V_{n+1}(y + \mu_n + \Delta L_{n+1})] \\ &= e^{-R_n} \int_{\mathbb{R}} \phi_{n+1}(z) V_{n+1}(y + \mu_n + z) dz \end{aligned}$$

where $R_n = \int_{t_n}^{t_{n+1}} r_s ds$. Substitution of $z = x - y - \mu_n$ yields

$$\tilde{V}_n(y) = e^{-R_n} \int_{\mathbb{R}} \phi_{n+1}(x - \mu_n - y) V_{n+1}(x) dx ,$$

7.8 Windowed Laplace transform

In Section 7.4.4 we introduced an adaptive refinement algorithm that enabled us to efficiently evaluate the convolution integral arising from the ladder algorithm, even for distributions that are sharply peaked and heavy-tailed. Multiplying the transition density function with a smooth window function in the spatial domain results in a smoother function in the convolution. This means that the resulting function in the Laplace domain is also smoother than the Laplace transform of the original function,

and thus that expansions and the projections of Section 7.4.4 converge faster. In this appendix we will elaborate on this windowed Laplace transform approach.

In the adaptive algorithm of Section 7.4.4, we divided the integration area of the convolutions into subintervals of size Δ , where we calculated the Legendre expansion coefficients. On those subintervals where the function K_x is not well-behaved, the convergence is slow, and we need smaller subintervals for approximations. Even though we are only interested in K_x for a small subinterval at a time, our fast convolution algorithm takes into account the behaviour of K_x over the entire interval. With an appropriate windowing function that is equal to 1 on the relevant subinterval and tapers off smoothly to 0 outside of this interval, we obtain a function that is better behaved than the original function, and thus leads to faster converging expansion coefficient calculations. The following material is adapted from den Iseger (2006b).

For the moment, we will not specify what form the window function takes. We take w_Δ to be a function that equals 1 on the interval $[0, \Delta]$, and has ε -support on $[-\delta, \Delta + \delta]$ for some $\delta > 0$. We first define w_Δ on $[-\frac{P}{2}, \frac{P}{2}]$, where P is chosen such that $[0 - \delta, \Delta + \delta] \subset [-\frac{P}{2}, \frac{P}{2}]$. We extend w_Δ to a periodic function with period P by setting

$$w_\Delta(t + kP) = w_\Delta(t) \text{ for all } t \in \left[-\frac{P}{2}, \frac{P}{2}\right) \text{ and } k \in \mathbb{Z}.$$

The window function corresponding to the coefficient for the subinterval $[k\Delta, (k+1)\Delta]$ is defined by $w_{k,\Delta}(t) = w_\Delta(t - k\Delta)$.

If we are going to evaluate the windowed kernel function, we need an expression for its Laplace transform. To that end, we first define the windowed kernel function K_{x,Δ,w_k} as

$$K_{x,\Delta,w_k}(t) = w_{k,\Delta}(t)K_x(t).$$

The Laplace transform of the windowed kernel function is given by $W_{k,\Delta}\widehat{K}_x$. We can write this as just a function of \widehat{K}_x by writing the Fourier series expansion of $w_{k,\Delta}(t)$. By the periodicity of w_Δ we have

$$w_{k,\Delta}(t) = w_\Delta(t - k\Delta) = \sum_{j=-\infty}^{\infty} A_j e^{-\frac{2\pi i j(t-k\Delta)}{P}} = \sum_{j=-\infty}^{\infty} e^{\frac{2\pi i j k \Delta}{P}} A_j e^{-\frac{2\pi i j t}{P}},$$

where the Fourier coefficients A_j are given by

$$A_j = \frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} w_\Delta(t) e^{\frac{2\pi i j t}{P}} dt \approx \frac{1}{P} \int_{-\infty}^{\infty} w_\Delta(t) e^{\frac{2\pi i j t}{P}} dt = \frac{1}{P} \widehat{w} \left(\frac{2\pi i j}{P} \right). \quad (7.8.1)$$

This means we can write the Laplace transform $W_{k,\Delta}\widehat{K}_x$ (through its linearity property) as

$$W_{k,\Delta}\widehat{K}_x(s) = \sum_{j=-\infty}^{\infty} e^{\frac{2\pi i j k \Delta}{P}} A_j \widehat{K}_x \left(s + \frac{2\pi i j}{P} \right).$$

7.8 Windowed Laplace transform

By the smoothness of the window function, we know the Fourier series converges and this means we can truncate the sum, say at $2J + 1$ terms, yielding

$$W_{k,\Delta}\widehat{K}_x(s) \approx \sum_{j=-J}^J e^{\frac{2\pi ijk\Delta}{P}} A_j \widehat{K}_x\left(s + \frac{2\pi ij}{P}\right). \quad (7.8.2)$$

Now let us suppose that J is a multiple of P (if it is not already, we can choose J so that it is). Denoting $L = J/P$, we can write

$$\begin{aligned} W_{k,\Delta}\widehat{K}_x(s) &\approx \sum_{q=0}^{P-1} \sum_{j=-L}^{L-1} A_{jP+q} e^{\frac{2\pi ijk\Delta}{P}} \widehat{K}_x\left(s + \frac{2\pi i(jP+q)}{P}\right) \\ &= \sum_{q=0}^{P-1} e^{\frac{2\pi ijk\Delta}{P}} \sum_{j=-L}^{L-1} A_{jP+q} \widehat{K}_x\left(s + \frac{2\pi i(jP+q)}{P}\right), \end{aligned} \quad (7.8.3)$$

where we have omitted the last term from (7.8.2) (for $j = J$).

In order to obtain the expansion coefficients $\langle w_{k,\Delta}K_x(\cdot - k\Delta), \phi_m \rangle = \langle K_x(\cdot - k\Delta), \phi_m \rangle$, we need to invert the Gaussian quadrature rule $F_k(v) = \langle \Psi W_{k,\Delta}\widehat{K}_x, q_m^v \rangle_{Q_v^n}$, which is given by

$$\begin{aligned} F_k(v) &= \left\langle \Psi \sum_{q=0}^{P-1} \sum_{j=-L}^{L-1} A_{jP+q} e^{\frac{2\pi ijk\Delta}{P}} \widehat{K}_x\left(\cdot + \frac{2\pi i(jP+q)}{P}\right), q_m^v \right\rangle_{Q_v^n} \\ &= \sum_{q=0}^{P-1} e^{\frac{2\pi ijk\Delta}{P}} \sum_{j=-L}^{L-1} A_{jP+q} \left\langle \Psi \widehat{K}_x\left(\cdot + \frac{2\pi i(jP+q)}{P}\right), q_m^v \right\rangle_{Q_v^n}. \end{aligned} \quad (7.8.4)$$

The inversion is now realised through an N -point inverse fast Fourier transform

$$\langle K_x(\cdot - k\Delta), \phi_m \rangle \approx \frac{1}{N} \sum_{\ell=0}^{N-1} \exp\left(\frac{2\pi i\ell k\Delta}{N}\right) F_k\left(\frac{\ell}{N}\right).$$

The window function we will use, which satisfies the requirements we have laid on it, is a difference of Gaussian distribution functions

$$w(t) = \Phi\left(\frac{t + \delta/2}{\sigma}\right) - \Phi\left(\frac{t - (1 + \delta/2)}{\sigma}\right),$$

where $\Phi(x)$ is the Gaussian distribution function given by

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{u^2}{2}} du,$$

and σ is a scaling parameter. This scale parameter is chosen in such a way that, given an arbitrarily chosen $\varepsilon > 0$, it satisfies

$$\exp\left(-\frac{1}{2}\left(\frac{\delta}{2\sigma}\right)^2\right) < \varepsilon. \quad (7.8.5)$$

The Laplace transform $\widehat{w}(s)$ of this window function is given by

$$\widehat{w}(s) = \frac{|\sigma|}{s} \exp\left(\frac{s^2\sigma^2}{2}\right) \left(\exp\left(\frac{\delta s\sigma}{2}\right) - \exp\left(-\left(\Delta + \frac{\delta}{2}\right)s\sigma\right) \right).$$

A typical Gaussian window is depicted in Figure 7.2.

Windowed Laplace transform algorithm

Input: $\widehat{K}_x, \widehat{w}, n, N, \Delta, \varepsilon, M$ (a power of 2), J .

Output: $\langle K_x(\cdot - k\Delta), \phi_m \rangle, k = 0, \dots, M - 1$.

Parameters: $P, L = \frac{1}{P}, \delta, \sigma$ (according to (7.8.5)), $v_\ell = \frac{\ell}{N}$.

Step 1. For $m = 0, \dots, n, \ell = 0, \dots, N - 1, j = -L, \dots, L - 1$, and $q = 0, \dots, P - 1$, compute

$$\left\langle \widehat{K}_x \left(\cdot + \frac{2\pi i(jP + q)}{P} \right), q_m^{v_\ell} \right\rangle_{Q_{v_\ell}^n}.$$

Step 2. For $m = 0, \dots, n, \ell = 0, \dots, N - 1$, and $q = 0, \dots, P - 1$, compute

$$\sum_{j=-L}^{L-1} A_{jP+q} \left\langle \widehat{K}_x \left(\cdot + \frac{2\pi i(jP + q)}{P} \right), q_m^{v_\ell} \right\rangle_{Q_{v_\ell}^n},$$

where the Fourier coefficients A_j are given by (7.8.1).

Step 3. For $m = 0, \dots, n$ and $\ell = 0, \dots, N - 1$, compute

$$\sum_{q=0}^{P-1} e^{\frac{2\pi i q k \Delta}{P}} \sum_{j=-L}^{L-1} A_{jP+q} \left\langle \widehat{K}_x \left(\cdot + \frac{2\pi i(jP + q)}{P} \right), q_m^{v_\ell} \right\rangle_{Q_{v_\ell}^n}.$$

Step 4. For $m = 0, \dots, n$, compute

$$\langle K_x(\cdot - k\Delta), \phi_m \rangle = \frac{1}{N} \sum_{\ell=0}^{N-1} \exp\left(\frac{2\pi i \ell k \Delta}{N}\right) F\left(\frac{\ell}{N}\right).$$

Given these coefficients from the window inversion algorithm, we can reconstruct the Legendre expansion and find the desired function K_x .

7.8 Windowed Laplace transform

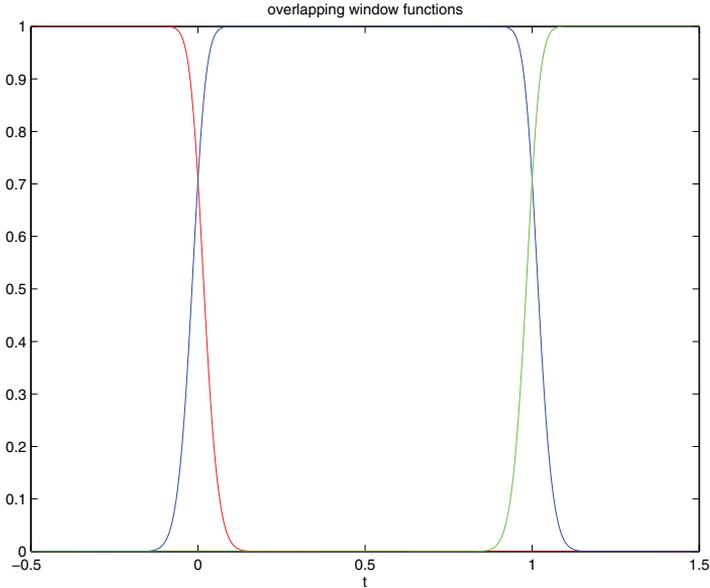


Figure 7.1: Three overlapping windows

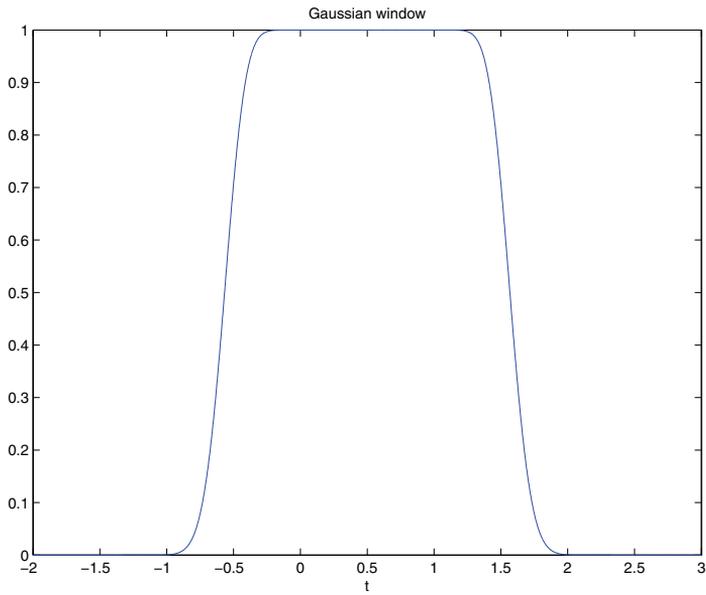


Figure 7.2: Gaussian distribution window on $[0, 1]$ ($\Delta = 1$), $\delta = 1$, $\sigma = 0.1$.

Chapter 8

Cliquet options

8.1 Introduction

Cliquet options are essentially a series of forward-starting at-the-money options with a single premium determined up front, that locks in any gains on specific dates. The strike price is then reset at the new level of the underlying asset. Any declines in the underlying asset reset the strike price at a lower level, but earlier profits are kept. Floors and caps are then added to fix the minimum and maximum returns. Hence, by construction, cliquet options protect against downside risk, yet they are affordably priced, since the payoff is locally capped, hence never too extreme. Cliquets are embedded in several popular retail structures such as Accumulators - capital protected structured notes with embedded options. These products seem appealing to high-net-worth individuals who seek moderate risk with active performance (see Patel (2002)).

Despite being commonly traded, pricing cliquets can be controversial, because it requires the correct pricing of the forward volatility smile (see Patel (2002)). Some dealers price cliques in several different models to have a better idea of the model risk: local volatility, stochastic, jump and ad hoc models, then they calibrate these to vanilla prices, so that they are consistent. (A good overview on exposure to volatility risk with cliquets is Wilmott (2002)). The present chapter discusses three different models: lognormal and jump-diffusion models with piecewise constant volatility (incorporating skews), and stochastic volatility model (Heston's model (1993)). We show how the price of the cliquet option can be computed by developing a (recursion) algorithm and making use of the recently developed numerical Laplace transform inversion algorithm of den Iseger (2006b). The precision of this approximation is almost the same as the precision of a computer. We can also obtain the sensitivity measures, called the Greeks, of the price w.r.t. changes in the market variables using a Girsanov type drift adjustment (see den Iseger and Oldenkamp (2006a)). We compute the Delta, Gamma, Rho and Vega of the cliquet option for all three models. The Vega for the Heston model

consists of the sensitivity of the price w.r.t. the initial volatility v_0 and all the parameters of the Heston model (the latter gives a good insight into the robustness of the parameters of the model).

The chapter is organized as follows. Section 8.2 computes the price for deterministic volatility (section 8.2.1), jump–diffusion (section 8.2.2), and stochastic volatility models (section 8.2.3). Section 8.3 shows how the Greeks can be computed first for the case of lognormal underlying with piecewise constant volatility (section 8.3.1), then for a jump - diffusion model (section 8.3.2), and the hestonian volatility model (section 8.3.3).

The results of this chapter are described in the working paper den Iseger and Oldenkamp (2006c).

8.2 Pricing in the constant and stochastic volatility models

8.2.1 Pricing in the deterministic volatility model

Cliquet options are essentially a series of forward-starting at-the-money options with a single premium determined up front, that lock in any gains on specific dates. The strike price is then reset at the new level of the underlying asset. We will use the following form, considering a global cap, global floor and local caps at pre-defined observation times t_i ($i = 1, \dots, n$),

$$P := \exp(-rt_n) N E_Q \left(\min \left(\max \left(\sum_{i=1}^n \max \left(F_i, \min \left(C_i, \frac{S_i - S_{i-1}}{S_{i-1}} \right) \right), F \right), C \right) \right) \quad (8.2.1)$$

where N is the notional, C is the global cap, F is the global floor, F_i , $i = 1, \dots, n$ the local floors, C_i , $i = 1, \dots, n$ are the local caps, and S is the asset price following either a diffusion with a geometric Brownian motion, or a jump-diffusion process. For the sake of notational simplicity we consider the case $N = 1$. Let

$$Y_i := \max(F_i, \min(C_i, R_i - 1)), \quad (8.2.2)$$

where

$$R_i := \frac{S_i}{S_{i-1}}, \quad (8.2.3)$$

8.2 Pricing in the constant and stochastic volatility models

then we can determine the probability density function of \mathbf{Y}_i and its Laplace transform. The following identities follow from definition (8.2.2)

$$\begin{aligned} f_{Y_i}(x) &= 0, & x < F_i \\ \mathbf{P}\{\mathbf{Y}_i = F_i\} &= F_{R_i}(F_i + 1), \\ f_{Y_i}(x) &= f_{R_i}(x + 1), & F_i < x < C_i \\ \mathbf{P}\{\mathbf{Y}_i = C_i\} &= 1 - F_{R_i}(C_i + 1), \\ f_{Y_i}(x) &= 0, & C_i < x. \end{aligned} \tag{8.2.4}$$

Hence the Laplace transform is given by

$$\mathcal{L}f_{Y_i}(s) = e^{-sF_i} \left(F_{R_i}(F_i + 1) + \int_0^{C_i - F_i} e^{-sx} f_{R_i}(x + F_i + 1) dx + e^{-s(C_i - F_i)} (1 - F_{R_i}(C_i + 1)) \right). \tag{8.2.5}$$

When the distribution of the return \mathbf{R} is given explicitly, then the Laplace transform (8.2.5) can be evaluated either directly as a closed form expression or with the transformation algorithm 4.5.5.2. Assuming that the returns per period are independent, the Laplace transform of the sum $\mathbf{Z} := \sum_{i=1}^n \mathbf{Y}_i$ is the product of the Laplace transforms, that is

$$\mathcal{L}_Z(s) = \prod_{i=1}^n L_{Y_i}(s) = e^{-rt_n} \int_F^C x F_Z(dx) + e^{-sF} F_Z(F) + e^{-sC} (1 - F_Z(C)). \tag{8.2.6}$$

We can invert this Laplace transform with the Laplace inversion algorithm 4.5.5.1, and we obtain the price of the cliquet option by computing f_Z and adjusting it for the global floor F and global cap C .

Remark 8.2.1. *Since the only assumption concerning the asset price process is that the returns per period should be independent, the model can handle skews (smiles), given that the volatility per period remains independent. The same observation holds for the interest rate: it can be defined in "buckets".*

8.2.2 Pricing in the jump–diffusion model

Consider now the process (see for instance Merton(1976), Runggaldier(2003), Glasserman and Merener (2003))

$$\frac{d\mathbf{S}_t}{\mathbf{S}_t} = (r - q)dt + \sigma d\mathbf{W}_t + d \left(\sum_{j=1}^{N_t} (\mathbf{Z}_j - 1) \right), \tag{8.2.7}$$

where r is the risk free interest rate, q the dividend rate, σ the deterministic volatility, \mathbf{W} is a standard Brownian motion, \mathbf{N} a Poisson process with rate λ_t , \mathbf{Z}_j independent

and identically lognormally distributed with density f_Z , and \mathbf{W} , \mathbf{S} and $\{\mathbf{Z}_1, \mathbf{Z}_2, \dots\}$ mutually independent. \mathbf{S} is right continuous with \mathbf{S}_{t-} the left limit at t . Then \mathbf{S} is a jump diffusion process with jump size $\mathbf{Z}_j, j = 1, \dots$. The solution of the above SDE is given by

$$\frac{\mathbf{S}_t}{\mathbf{S}_0} = \exp \left(\left(r - q - \frac{\sigma^2}{2} \right) t + \sigma \mathbf{W}_t \right) \prod_{j=1}^{N_t} \mathbf{Z}_j. \quad (8.2.8)$$

The Laplace transform of $\ln(\mathbf{R}_i)$ is then given by

$$\mathcal{L}f_{\ln R_i}(s) = \exp \left((r - q)ts - \frac{\sigma^2}{2}ts^2 \right) \exp \left(- \int_{t_{i-1}}^{t_i} \lambda(u)du (1 - \widehat{f}_{\ln Z_i}(s)) \right), \quad (8.2.9)$$

with $\mathcal{L}f_{\ln R_i}$ the Laplace transform of the density of $\ln(\mathbf{R}_i)$, and $\mathcal{L}f_{\ln Z_i}$ the Laplace transform of the density of $\ln(\mathbf{Z}_i)$. However, in order to generate prices associated with an asset described by a jump diffusion process with relation (8.2.5), one needs the Laplace transform of \mathbf{R}_i instead of $\ln(\mathbf{R}_i)$. The following abstract algorithm determines the Laplace transform of \mathbf{Y}_i , which we then use in (8.2.5) in order to determine the price of a cliquet contract. For the technical details of the algorithm see Algorithm 8.5.1.1:

Algorithm 8.2.1

1. Invert the Laplace transform $\mathcal{L}f_{\ln R_i}(s)$ with algorithm 4.5.5.1 obtaining $f_{\ln R_i}(x)$;
2. Calculate the density of \mathbf{R}_i by $f_{R_i}(x) = f_{\ln R_i}(\ln(x))\frac{1}{x}$;
3. Compute the Laplace transform $\mathcal{L}f_{Y_i}(s)$ with (8.2.5) and algorithm 4.5.5.2.

Remark 8.2.2. *Similarly as for the previous section, this model can also handle skews (smiles) and 'bucketed' interest rates given that the returns per period remain independent.*

8.2.3 Pricing with Heston's volatility model

Heston's stochastic volatility model allows arbitrary correlation between volatility and spot asset returns (see Heston (1993)). Let $\mathbf{X}_t := \ln \left(\frac{\mathbf{S}_t}{\mathbf{S}_0} \right)$, then

$$d\mathbf{X}_t = \left(\mu - \frac{\mathbf{v}_t}{2} \right) dt + \sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(1)}, \quad (8.2.10)$$

with \mathbf{v}_t the stochastic volatility and $\mathbf{W}_t^{(1)}$ a Brownian motion. The dynamics of the volatility $\sqrt{\mathbf{v}}$ is described by the following Ornstein-Uhlenbeck process

$$d\mathbf{v}_t = \gamma (\theta - \mathbf{v}_t) dt + \sigma \sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(2)}, \quad (8.2.11)$$

8.2 Pricing in the constant and stochastic volatility models

where $\mathbf{W}_t^{(2)}$ has correlation ρ with $\mathbf{W}_t^{(1)}$ (θ is the mean, γ the mean reversion rate, and σ is the volatility of the volatility). The two-dimensional Laplace transform of the joint distribution $f(x, v; v_0)$ of \mathbf{X} and the volatility, given the volatility at the start, v_0 , was computed by Drăgulescu and Yakovenko (2002) as

$$\mathcal{L}_{x,v}f(s_x, s_v; v_0) = \exp\left(-s_x\mu t - p(s_x, s_v)v_0 + \frac{\gamma\theta(\Gamma - \Omega)t}{\sigma^2} - \frac{2\gamma\theta}{\sigma^2} \ln\left(\frac{\zeta - \exp(-\Omega t)}{\zeta - 1}\right)\right), \quad (8.2.12)$$

where

$$\Gamma = \gamma - \rho\sigma s_x,$$

$$\Omega = \sqrt{\Gamma^2 + \sigma^2(s_x - s_x^2)},$$

$$\zeta = 1 + \frac{2\Omega}{\sigma^2 s_v + \Gamma - \Omega}$$

and

$$p(s_x, s_v) = \frac{2\Omega}{\sigma^2} \frac{1}{\zeta \exp(\Omega t) - 1} - \frac{\Gamma - \Omega}{\sigma^2}.$$

Observe that the Laplace transform (8.2.12) can equivalently be written as

$$\mathcal{L}_{x,v}f(s_x, s_v; v_0) = \exp(-p(s_x, s_v)v_0) \mathcal{L}_{x,v}f(s_x, s_v), \quad (8.2.13)$$

where $\mathcal{L}_{x,v}f(s_x, s_v)$ is the Laplace transform of the conditional joint density of \mathbf{X} and \mathbf{v} , given $v_0 = 0$. Because of the stochastic volatility process, the returns of the different periods are no longer independent, hence $\mathbf{Z} = \sum_{i=1}^n \mathbf{Y}_i$ is no longer a convolution, therefore we need to calculate the distribution of \mathbf{Z} recursively. Let us introduce the following definition,

$$\mathbf{Z}_k := \sum_{i=1}^k \mathbf{Y}_i, \text{ where } \mathbf{Y}_i \text{ is given by (8.2.2), and } k = 1, \dots, n; \quad (8.2.14)$$

then $\mathbf{Z} = \mathbf{Z}_n$. Consider now periods k and $k + 1$; if we calculated the Laplace transform $\mathcal{L}_{z,v}\varphi_k(s_z, s_{v_k})$ of the joint distribution of the pair $(\mathbf{Z}_k, \mathbf{v}_k)$, then we could calculate the distribution of $(\mathbf{Z}_{k+1}, \mathbf{v}_{k+1}) = (\mathbf{Z}_k + \mathbf{Y}_{k+1}, \mathbf{v}_{k+1})$. Let us start at $k = 1$, then $(\mathbf{Z}_1, \mathbf{v}_1) = (\mathbf{Y}_1, \mathbf{v}_1)$. While the Laplace transform (8.2.12) refers to the logarithm of the returns, here we need the non-logarithmic form of the returns, capped and floored,

$$\mathcal{L}_{x,v}\varphi_1(s_x, s_{v_1}; v_0) = \mathcal{L}_{x,v}f_Y(s_x, s_{v_1}; v_0), \quad (8.2.15)$$

denoting with $\varphi_k(z, v_k; v_{k-1})$ the density of $(\mathbf{Z}_k, \mathbf{v}_k)$ given \mathbf{v}_{k-1} , and $\varphi_1 = f_{Y_1}$. In order to determine $\mathcal{L}_{x,v}f_Y(s_x, s_{v_1}; v_0)$ we use algorithm 8.2.1 with the Laplace transform (8.2.12) in step 1. Consider now that we calculated the Laplace transform $\mathcal{L}_{z,v}\varphi_k(s_z, s_{v_k})$

of the joint density of $(\mathbf{Z}_k, \mathbf{v}_k)$. We need to calculate now the Laplace transform of the joint density of $(\mathbf{Z}_{k+1}, \mathbf{v}_{k+1}) = (\mathbf{Z}_k + \mathbf{Y}_{k+1}, \mathbf{v}_{k+1})$. Since \mathbf{Z}_k and \mathbf{Y}_{k+1} are independent given \mathbf{v}_k , then $\mathbf{Z}_k + \mathbf{Y}_{k+1}$ is a convolution, hence the related Laplace transform is the product of the Laplace transforms of the individual pdf's. (Observe that \mathbf{Z}_k and \mathbf{v}_{k+1} , given \mathbf{v}_k are also independent.) We have obtained the recursion relation

$$\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}}; v_k) = \mathcal{L}_z\varphi_k(s_z; v_k)\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k). \quad (8.2.16)$$

Note that the ultimate goal is to determine the marginal density $\mathcal{L}_v\varphi_n(z, 0)$. Hence, integrating out over v_k yields

$$\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}}) = \int_0^\infty \mathcal{L}_z\varphi_k(s_z; v_k)\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)dv_k. \quad (8.2.17)$$

Since φ_k is the joint density of $(\mathbf{Z}_k, \mathbf{v}_k)$ then

$$\mathbf{P}\{\mathbf{Z}_k \in A, v_k \geq v_M\} \leq \mathbf{P}\{v_k \geq v_M\} = \mathbf{P}\{|\sqrt{v_k}| \geq \sqrt{v_M}\}.$$

From relations (2)–(4) in Heston (1993) and section 3.5.2 of Lamberton and Lapeyre (1996) we have that $\sqrt{v_k}$ is normally distributed with mean μ_{v_k} , and variance Var_{v_k} . Hence, choosing $v_M > \mu_k + M\sqrt{Var_{v_k}}$

$$\mathbf{P}\{v_k \geq v_M\} \longrightarrow 0, \text{ for } M \text{ large enough.}$$

Since $\mathcal{L}_{z,v}f_Y(s_z, s_{v_{k+1}}; v_k)$ is bounded for a given (finite) v_k , for all imaginary s_z and $s_{v_{k+1}}$ $|\mathcal{L}_{z,v}f_Y(s_z, s_{v_{k+1}})| \leq \|f_Y\|_{L_1}$, and $\|f_Y(\cdot, \cdot; v_k)\|_{L_1} = 1$, we obtain

$$\begin{aligned} \int_{v_M}^\infty \mathcal{L}_z\varphi_k(s_z; v_k)\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)dv_k &\leq \sup_{v_k \geq v_M} \mathcal{L}_{z,v}f_Y(s_z, s_{v_{k+1}}; v_k) \int_{v_M}^\infty \mathcal{L}_z\varphi_k(s_z; v_k)dv_k \\ &\leq \int_{v_M}^\infty \mathcal{L}_z\varphi_k(s_z; v_k)dv_k \leq \mathbf{P}\{v_k \geq v_M\} \longrightarrow 0. \end{aligned}$$

Hence,

$$\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}}) \approx \int_0^\infty \mathcal{L}_z\varphi_k(s_z; v_k)\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)1_{\{v_k \leq v_M\}}dv_k. \quad (8.2.18)$$

An efficient estimation of this integral is given in details in appendix 8.5.2. Here we give an abstract algorithm for the computation of the density of the capped and floored returns $\mathcal{L}_v\varphi_n(z, 0)$.

Algorithm 8.2.2

1. Specify s_z and s_{v_k} as for algorithm 4.5.5.1;
2. $k = 1$;

8.2 Pricing in the constant and stochastic volatility models

3. Compute $\mathcal{L}_{z,v}f_Y(s_z, s_{v_k})$ with algorithm 8.2.1 with input the LT $\mathcal{L}_{z,v}f(s_z, s_{v_k})$ (obtained from (8.2.13));

4. Estimate $\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}})$ with (8.2.18) (details in appendix 8.5.2);

5. If $k < N$ then set $k := k + 1$ and go to step 2.

6. Invert $\mathcal{L}_{z,v}\varphi_N(s_z; 0)$ w.r.t. s_z with algorithm 4.5.5.1.

8.2.4 Numerical results

Windcliff et al.(2003) use PDE methods to value several cliquet options. One of their numerical results calculates the value of a contract with parameters also considered by Wilmott (2002):

Parameter	Value
S_0	100
σ	0.20
r	0.03
periods	5
C_i	0.08
F_i	0.00
C	∞
F	0.16
Notional	1

with $i = 1, \dots, 5$. They obtain the price $P = 0.174060717$ with the Crank-Nicolson algorithm with 640 time steps (refinement). Their result has a precision of 10^{-6} . Our algorithm calculates the price of the cliquet option for all global floors at once in a fraction of a second:

Floor	Value
0.00	0.15243314921649
0.04	0.15371172664930
0.08	0.15593481863880
0.12	0.16349625180472
0.16	0.17406196069385
0.20	0.19322488179335
0.24	0.21591805600381
0.28	0.24530207055173
0.32	0.27650274915383
0.36	0.31022026731203
0.40	0.34428319057002
0.44	0.37871150962703
0.48	0.41313982868405
0.52	0.44756814774106
0.56	0.48199646679809
0.60	0.51642478585513

8.3 Greeks in lognormal and jump diffusion models

In den Iseger and Oldenkamp (2006a) we presented a new technique for calculating Greeks for guaranteed unit linked products which we can also use for the case of cliquet options. We can extend this technique for the stochastic volatility case (Heston model).

8.3.1 Greeks in the lognormal model with (piecewise) constant volatility

The main assumption of this subsection is that \mathbf{S}_t is lognormally distributed, such that

$$\mathbf{S}_t = \mathbf{S}_0 \exp \left((r - q - \frac{1}{2}\sigma^2)t + \sigma \mathbf{W}_t \right), \quad (8.3.1)$$

where \mathbf{W}_t is a Brownian motion at time t , r is the risk free rate, q is the dividend rate, and σ is the volatility. The random variables $\ln(\mathbf{R}_i)$, $i = 1, \dots, n$ (defined by (8.2.3)), are independent and identically normally distributed with mean $r - q - \frac{\sigma^2}{2}$ and variance σ^2 , and, as before, we denote their corresponding normal probability density function by $f_{\ln(\mathbf{R}_i)}$. Let us furthermore use the following notation

$$\Phi(\mathbf{Z}_k) := \min(\max(\mathbf{Z}_k, F), C), \quad (8.3.2)$$

8.3 Greeks in lognormal and jump diffusion models

with \mathbf{Z}_k defined by (8.2.14), then relation (8.2.1) can be written as

$$P = \exp(-rt_n) \mathbf{E}_Q(\Phi(\mathbf{Z}_n)). \quad (8.3.3)$$

Delta hedging

The Delta of an option is the sensitivity of its price w.r.t. to the underlying, hence we need to evaluate the expression $\partial P(S_0)/\partial S_0$, with P defined by (8.2.1). Note that $\partial P(S_i)/\partial S_0 = 0$ $i = 1, \dots, n-1$, since the returns are independent, and so the later periods are independent of changes in the first period. In order to deduce the delta, we observe that the price function P depends on \mathbf{S} through the return \mathbf{R} ; namely we are interested in the effect of an infinitesimal change in the initial value of \mathbf{S}_0 when $\mathbf{R}_1 - 1 = \frac{S_1 - S_0}{S_0}$ with $\mathbf{S}_1 = \mathbf{S}_0 \exp((r - q - \frac{1}{2}\sigma^2)t_1 + \sigma \mathbf{W}_{t_1})$. Using relation (8.3.3) it follows that

$$\begin{aligned} \Delta &:= \exp(-rt_n) \mathbf{E}_Q \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \frac{\partial \mathbf{R}_1}{\partial S_0} \right) \\ &= \exp(-rt_n) \frac{1}{S_{0^-}} \mathbf{E}_Q \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \exp \left((r - q - \frac{1}{2}\sigma^2)t_1 + \sigma \mathbf{W}_{t_1} \right) \right). \end{aligned} \quad (8.3.4)$$

The term $\exp(-\frac{1}{2}\sigma^2 t_1 + \sigma \mathbf{W}_{t_1})$ is the Radon-Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\mathbf{W}}_t := \mathbf{W}_t - \sigma t \mathbf{1}_{\{t \leq t_1\}}$, hence we obtain that the expression above becomes

$$\Delta = \exp(-rt_n) \frac{1}{S_{0^-}} \exp((r - q)t_1) \mathbf{E} \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right), \quad (8.3.5)$$

where

$$\widehat{\mathbf{Z}}_n = \widehat{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widehat{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widehat{\mathbf{R}}_1)), \quad (8.3.6)$$

with

$$\widehat{\mathbf{R}}_1 := \exp \left((r - q + \frac{1}{2}\sigma^2)t_1 + \sigma \widehat{\mathbf{W}}_{t_1} \right). \quad (8.3.7)$$

It remains now to evaluate the last term in Delta given by (8.3.5).

$$\mathbf{E} \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right) = \mathbf{E} \left(\mathbf{1}_{\{F \leq \widehat{\mathbf{Z}}_n \leq C\}} \mathbf{1}_{\{F_1 \leq \widehat{\mathbf{R}}_1 \leq C_1\}} \right).$$

In conclusion, this yields for the Delta (relation (8.3.5)) that

$$\Delta = \exp(-rt_n) \exp((r - q)t_1) \frac{1}{S_{0^-}} \mathbf{E} \left(\mathbf{1}_{\{F \leq \widehat{\mathbf{Z}}_n \leq C\}} \mathbf{1}_{\{F_1 \leq \widehat{\mathbf{R}}_1 \leq C_1\}} \right). \quad (8.3.8)$$

The distribution of $\widehat{\mathbf{Z}}_n$ can be computed similarly to that of \mathbf{Z}_n (see section 8.2.1).

Gamma hedging

For the computation of the Gamma we follow the same line of reasoning as in the previous section,

$$\Gamma := \exp(-rt_n) E_Q \left(\frac{\partial}{\partial \mathbf{R}_1} \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \right) \left(\frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} \right)^2 \right). \quad (8.3.9)$$

The last term of this expression contains the Radon-Nikodym derivative corresponding to a 2σ shift, $\exp(-2\sigma^2 t_1 + 2\sigma \mathbf{W}_{t_1})$, that is,

$$\frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} = \frac{1}{\mathbf{S}_{0-}^2} \exp(2(r-q)t_1) \exp(\sigma^2 t_1) \exp(-2\sigma^2 t_1 + 2\sigma \mathbf{W}_{t_1}). \quad (8.3.10)$$

Hence, the Gamma is given by

$$\begin{aligned} \Gamma &= \exp(-rt_n) \frac{1}{\mathbf{S}_{0-}^2} \exp(2(r-q)t_1) \exp(\sigma^2 t_1) \cdot \\ &\quad \cdot E \left((\delta_F(\tilde{\mathbf{Z}}_n) - \delta_C(\tilde{\mathbf{Z}}_n)) 1_{\{F_1 \leq \tilde{\mathbf{R}}_1 \leq C_1\}} + 1_{\{F \leq \tilde{\mathbf{Z}}_n \leq C\}} (\delta_{F_1}(\tilde{\mathbf{R}}_1) - \delta_{C_1}(\tilde{\mathbf{R}}_1)) \right), \end{aligned} \quad (8.3.11)$$

where

$$\tilde{\mathbf{Z}}_n = \tilde{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \tilde{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \tilde{\mathbf{R}}_1)),$$

with

$$\tilde{\mathbf{R}}_1 := \exp \left((r-q + \frac{3}{2}\sigma^2)t_1 + \sigma \tilde{\mathbf{W}}_{t_1} \right). \quad (8.3.12)$$

The distribution of $\tilde{\mathbf{Z}}_n$ can be computed similarly to that of \mathbf{Z}_n (see section 8.2.1).

Rho hedging

Rho (ρ) is the sensitivity (derivative) w.r.t. the risk free interest rate r . Similarly as for Δ we get

$$\rho = \exp(-rt_n) E_Q \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \frac{\partial \mathbf{R}_1}{\partial r} \right) - t_n P. \quad (8.3.13)$$

Since

$$\frac{\partial \mathbf{R}_1}{\partial r} = t_1 \mathbf{R}_1,$$

using the same line of reasoning as for Δ , relations (8.3.4) - (8.3.8), we obtain that

$$\rho = \exp(-rt_n) t_1 \exp((r-q)t_1) E \left(1_{\{F \leq \hat{\mathbf{Z}}_n \leq C\}} 1_{\{F_1 \leq \hat{\mathbf{R}}_1 \leq C_1\}} \right) - t_n P, \quad (8.3.14)$$

with $\hat{\mathbf{R}}_1$ given by relation (8.3.7).

8.3 Greeks in lognormal and jump diffusion models

Vega hedging

Vega (ϑ) is the sensitivity of the derivative w.r.t. the volatility. It is therefore defined by the expression

$$\vartheta = \exp(-rt_n) E_Q \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \frac{\partial \mathbf{R}_1}{\partial \sigma} \right). \quad (8.3.15)$$

The last term of this expression can be calculated directly as

$$\frac{\partial \mathbf{R}_1}{\partial \sigma} = (-\sigma t_1 + \sqrt{t_1} \mathbf{W}_{t_1}) \exp((r-q)t_1) \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma \sqrt{t_1} \mathbf{W}_{t_1}\right). \quad (8.3.16)$$

The expression $\mathbf{RN} = \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma \sqrt{t_1} \mathbf{W}_{t_1}\right)$ is again a Radon - Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\mathbf{W}}_t := \mathbf{W}_t - \sigma t 1_{\{t \leq t_1\}}$, hence we obtain for the vega that

$$\vartheta = \exp((r-q)t_1 - rt_n) \left(-\sigma t_1 E \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right) + \sqrt{t_1} E \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \widehat{\mathbf{W}}_{t_1} \right) \right), \quad (8.3.17)$$

where $\widehat{\mathbf{Z}}_n$ and $\widehat{\mathbf{R}}_1$ are define by relations (8.3.6) and (8.3.7), respectively. Observe now that $\widehat{\mathbf{W}}_{t_1}$ is Gaussian, and for a standard Gaussian density f we have the following result:

$$wf(w) = -\frac{\partial f(w)}{\partial w}. \quad (8.3.18)$$

Hence, if Ψ is a function of \mathbf{W} then we obtain by integration by parts (and by $\Psi(w)f(w) \rightarrow 0$ as $|w| \rightarrow \infty$) that

$$E(\mathbf{W}\Psi(\mathbf{W})) = E((D\Psi)(\mathbf{W})), \quad (8.3.19)$$

where D stands for the differentiation operator w.r.t. \mathbf{W} . Applying this result to $\Psi(\widehat{\mathbf{W}}_{t_1}) := \frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1}$ we obtain that

$$E(\widehat{\mathbf{W}}_{t_1} \Psi(\widehat{\mathbf{W}}_{t_1})) = E \left(\frac{\partial^2 \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n^2} \frac{\partial \widehat{\mathbf{Z}}_n^2}{\partial \widehat{\mathbf{R}}_1^2} \frac{\partial \widehat{\mathbf{R}}_1}{\partial \widehat{\mathbf{W}}_{t_1}} \right). \quad (8.3.20)$$

The last term can again be computed directly as

$$\frac{\partial \widehat{\mathbf{R}}_1}{\partial \widehat{\mathbf{W}}_{t_1}} = \sigma t_1 \sqrt{t_1} \exp((r-q)t_1) \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma \sqrt{t_1} \mathbf{W}_{t_1}\right), \quad (8.3.21)$$

where the expression $\mathbf{RN}_2 = \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma \sqrt{t_1} \widehat{\mathbf{W}}_{t_1}\right)$ is again a Radon - Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\widehat{\mathbf{W}}}_t := \widehat{\mathbf{W}}_t - \sigma t 1_{\{t \leq t_1\}} = \mathbf{W}_t - 2\sigma t 1_{\{t \leq t_1\}}$, hence we obtain for the vega that

$$\vartheta = \exp((r-q)t_1 - rt_n) \left(-\sigma t_1 E \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right) + \sqrt{t_1} E \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right) \right), \quad (8.3.22)$$

where

$$\widehat{\mathbf{Z}}_n = \widehat{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widehat{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widehat{\mathbf{R}}_1)), \quad (8.3.23)$$

with

$$\widehat{\mathbf{R}}_1 := \exp\left((r - q + \frac{3}{2}\sigma^2)t_1 + \sigma\widehat{\mathbf{W}}_{t_1}\right). \quad (8.3.24)$$

The expressions $E\left(\frac{\partial\Phi(\widehat{\mathbf{Z}}_n)}{\partial\widehat{\mathbf{Z}}_n} \frac{\partial\widehat{\mathbf{Z}}_n}{\partial\widehat{\mathbf{R}}_1}\right)$ and $E\left(\frac{\partial\Phi(\widehat{\mathbf{Z}}_n)}{\partial\widehat{\mathbf{Z}}_n} \frac{\partial\widehat{\mathbf{Z}}_n}{\partial\widehat{\mathbf{R}}_1}\right)$ can be calculated in exactly the same manner as for the Δ .

8.3.2 Greeks in the jump-diffusion model with constant volatility

The Greeks can be calculated with exactly the same methodology as for the lognormal model, with only some modifications in the formulae. It follows from (8.2.8) that \mathbf{R}_i , as defined by (8.2.3), has now the form

$$\mathbf{R}_i = \exp\left((r - q - \frac{1}{2}\sigma^2)t_i + \sigma\mathbf{W}_{t_i}\right) \prod_{j=\mathbf{N}_{t_{i-1}+1}}^{\mathbf{N}_{t_i}} \mathbf{J}_j. \quad (8.3.25)$$

Relations $\partial\Phi(\mathbf{Z}_n)/\partial\mathbf{Z}_n$ and $\partial\mathbf{Z}_n/\partial\mathbf{R}_1$ remain valid unchanged, and

$$\frac{\partial\mathbf{R}_1}{\partial\mathbf{S}_0} = \frac{1}{\mathbf{S}_0^-} \exp((r - q)t_1) \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma\mathbf{W}_{t_1}\right) \prod_{j=1}^{\mathbf{N}_{t_1}} \mathbf{J}_j.$$

Since

$$\exp(\lambda(1 - E(\mathbf{J}))) \exp\left(-\frac{1}{2}\sigma^2 t_1 + \sigma\mathbf{W}_{t_1}\right) \prod_{j=1}^{\mathbf{N}_{t_1}} \mathbf{J}_j \quad (8.3.26)$$

is the Radon-Nikodym derivative which realizes a drift adjustment with σ of the Brownian motion $\widehat{\mathbf{W}}_t = \mathbf{W}_t - \sigma t 1_{\{t \geq t_1\}}$, and changes the intensity of the jump process into $(\lambda_t E(\mathbf{J}), \frac{x}{E(\mathbf{J})} f_{\mathbf{J}}(x) dx)$ (see Theorem 5.6.3), we obtain that

$$E_Q\left(\frac{\partial\Phi(\mathbf{Z}_n)}{\partial\mathbf{Z}_n} \frac{\partial\mathbf{Z}_n}{\partial\mathbf{R}_1} \frac{\partial\mathbf{R}_1}{\partial\mathbf{S}_0}\right) = \frac{1}{\mathbf{S}_0^-} \exp((r - q)t_1) E\left(\frac{\partial\Phi(\widehat{\mathbf{Z}}_n)}{\partial\widehat{\mathbf{Z}}_n} \frac{\partial\widehat{\mathbf{Z}}_n}{\partial\widehat{\mathbf{R}}_1}\right),$$

where

$$\widehat{\mathbf{Z}}_n = \widehat{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i, \quad \widehat{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widehat{\mathbf{R}}_1)),$$

with

$$\widehat{\mathbf{R}}_1 := \exp(-\lambda(1 - E(\mathbf{J}))) \exp\left((r - q + \frac{1}{2}\sigma^2)t_1 + \sigma\widehat{\mathbf{W}}_{t_1}\right) \prod_{j=1}^{\widehat{\mathbf{N}}_{t_1}} \widehat{\mathbf{J}}_j. \quad (8.3.27)$$

8.3 Greeks in lognormal and jump diffusion models

and the process $(\tilde{\mathbf{N}}, \tilde{\mathbf{J}}_j)$ is the jump process associated with intensity $(\lambda_t \mathbf{E}(\mathbf{J}), \frac{x}{\mathbf{E}(\mathbf{J})} f_J(x) dx)$. Therefore, the expressions for Δ and ρ remain valid unchanged with (8.3.27) instead of (8.3.7). Furthermore, this also holds for the Γ , having in addition, instead of the double shift (8.3.10) the following

$$\begin{aligned} \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} &= \frac{1}{\mathbf{S}_0^2} \exp(2(r - q - \frac{1}{2}\sigma^2)t_1 + 2\sigma \mathbf{W}_{t_1}) \prod_{j=1}^{N_{t_1}} \mathbf{J}_j^2 \\ &= \frac{1}{\mathbf{S}_0^2} \exp((2(r - q) + \sigma^2)t_1) \exp(-2\sigma^2 t_1 + 2\sigma \mathbf{W}_{t_1}) \prod_{j=1}^{N_{t_1}} \mathbf{J}_j^2. \end{aligned} \quad (8.3.28)$$

With an additional normalization factor, $\mathbf{S}_0^2 \exp(-(2(r - q) + \sigma^2)t_1 - \lambda(1 - \mathbf{E}(\mathbf{J}^2))) \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0}$ is exactly the Radon-Nikodym derivative which corresponds to a 2σ shift of the Brownian motion \mathbf{W} , intensity $\lambda \mathbf{E}(\mathbf{J}^2)$ and distribution of jumps $\frac{x^2}{\mathbf{E}(\mathbf{J}^2)} f_J(x) dx$. Hence (8.3.12) is substituted by

$$\tilde{\mathbf{R}}_1 = \exp\left((r - q + \frac{3}{2}\sigma^2)t_1 + \sigma \tilde{\mathbf{W}}_{t_1}\right) \prod_{j=1}^{\tilde{N}_{t_1}} \tilde{\mathbf{J}}_j, \quad (8.3.29)$$

with $(\tilde{\mathbf{N}}, \tilde{\mathbf{J}}_j)$ the jump process associated with intensity $(\lambda_t \mathbf{E}(\mathbf{J}^2), \frac{x^2}{\mathbf{E}(\mathbf{J}^2)} f_J(x) dx)$. For the vega relation (8.3.22) remains valid with (8.3.27) instead of (8.3.7) and (8.3.29) instead of (8.3.24).

8.3.3 Greeks in the lognormal model with Heston's volatility

The deduction of the Greeks for an asset price with stochastic volatility, defined by (8.2.10) and (8.2.11), requires the same sequence of steps as for the asset price with constant volatility, having in addition to deal with the correlation ρ between $d\mathbf{W}^{(1)}$ and $d\mathbf{W}^{(2)}$. From relation (8.2.10) we have that

$$\mathbf{S}_t = \mathbf{S}_0 \exp\left((r - q)t - \frac{1}{2} \int_0^t \mathbf{v}_s ds + \int_0^t \sqrt{\mathbf{v}_s} d\mathbf{W}_s^{(1)}\right), \quad (8.3.30)$$

together with the dynamics (8.2.11) of the volatility \mathbf{v} , driven by the Brownian motion $\mathbf{W}^{(2)}$. Let us use the notation

$$d\mathbf{W} = \begin{bmatrix} d\mathbf{W}^{(1)} \\ d\mathbf{W}^{(2)} \end{bmatrix}, \quad (8.3.31)$$

then

$$\mathbf{E} [d\mathbf{W} d\mathbf{W}^T] = \Omega dt = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} dt.$$

The Choleski decomposition of Ω exists such that $\Omega = LL^T$,

$$d\mathbf{W} = Ld\mathbf{B} = L \begin{bmatrix} d\mathbf{B}^{(1)} \\ d\mathbf{B}^{(2)} \end{bmatrix}, \quad (8.3.32)$$

$\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ standard independent Brownian motions. Moreover, in our case

$$\begin{aligned} d\mathbf{W}^{(1)} &= d\mathbf{B}^{(1)}, \\ d\mathbf{W}^{(2)} &= \varrho d\mathbf{B}^{(1)} + \sqrt{1 - \varrho^2} d\mathbf{B}^{(2)}. \end{aligned} \quad (8.3.33)$$

Delta hedging

The initial expression for Δ in (8.3.4) remains valid, that is,

$$\Delta = \exp(-rt_n) \frac{1}{\mathbf{S}_{0^-}} \mathbf{E} \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \exp \left((r - q)t_1 - \frac{1}{2} \int_0^{t_1} \mathbf{v}_s ds + \int_0^{t_1} \sqrt{\mathbf{v}_s} d\mathbf{W}_s^{(1)} \right) \right). \quad (8.3.34)$$

By substituting the Choleski decomposition (8.3.33) for $d\mathbf{W}^{(1)}$, the exponential under the expectation becomes

$$\mathbf{RN} := \exp \left((r - q)t_1 - \frac{1}{2} \int_0^{t_1} \mathbf{v}_s ds + \int_0^{t_1} \sqrt{\mathbf{v}_s} d\mathbf{B}_s^{(1)} \right).$$

where \mathbf{RN} is the Radon-Nikodym derivative, which realizes a drift adjustment with $\sqrt{\widehat{\mathbf{v}}_t}$ of the Brownian motion $d\mathbf{B}_t^{(1)}$, such that $d\widehat{\mathbf{B}}_t^{(1)} := d\mathbf{B}_t^{(1)} - \sqrt{\widehat{\mathbf{v}}_t} dt$ is a \mathbb{P} -Brownian motion, with

$$d\mathbb{P} := \mathbf{RN} d\mathbf{Q}. \quad (8.3.35)$$

The expression (8.3.34) for Δ becomes

$$\Delta = \exp(-rt_n) \frac{1}{\mathbf{S}_{0^-}} \exp((r - q)t_1) \mathbf{E}_{\mathbb{P}} \left(\frac{\partial \Phi(\widehat{\mathbf{Z}}_n)}{\partial \widehat{\mathbf{Z}}_n} \frac{\partial \widehat{\mathbf{Z}}_n}{\partial \widehat{\mathbf{R}}_1} \right), \quad (8.3.36)$$

where

$$\widehat{\mathbf{Z}}_n = \widehat{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widehat{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widehat{\mathbf{R}}_1)), \quad (8.3.37)$$

and

$$\widehat{\mathbf{R}}_1 := \exp \left((r - q)t_1 + \frac{1}{2} \int_0^{t_1} \widehat{\mathbf{v}}_s ds + \int_0^{t_1} \sqrt{\widehat{\mathbf{v}}_s} d\widehat{\mathbf{W}}_s^{(1)} \right), \quad (8.3.38)$$

$$d\widehat{\mathbf{v}}_t = \alpha(\beta - \widehat{\mathbf{v}}_t)dt + \sigma \sqrt{\widehat{\mathbf{v}}_t} d\widehat{\mathbf{W}}_t^{(2)}, \quad \alpha = \gamma - \sigma\varrho, \quad \beta = \frac{\gamma\theta}{\gamma - \sigma\varrho}, \quad (8.3.39)$$

with

$$d\widehat{\mathbf{W}}^{(1)} := d\widehat{\mathbf{B}}^{(1)}, \quad d\widehat{\mathbf{W}}^{(2)} := \varrho d\widehat{\mathbf{B}}^{(1)} + \sqrt{1 - \varrho^2} d\widehat{\mathbf{B}}^{(2)}, \quad (8.3.40)$$

8.3 Greeks in lognormal and jump diffusion models

and

$$d\widehat{\mathbf{W}} = \begin{bmatrix} d\widehat{\mathbf{W}}^{(1)} \\ d\widehat{\mathbf{W}}^{(2)} \end{bmatrix} = Ld\widehat{\mathbf{B}}. \quad (8.3.41)$$

According to section 8.4.1 the processes $\widehat{\mathbf{R}}_1$ and $\widehat{\mathbf{v}}$ of (8.3.38) and (8.3.39), respectively, can be rewritten again as a Heston model (set $\alpha = -1$ in section 8.4.1). Hence, under the measure \mathbb{P} the model preserves the Hestonian characteristics.

The last term of (8.3.36) can be calculated exactly the same way as in section 8.3.1.

Gamma hedging

For the computation of the Gamma we start again with the initial expression for Gamma in section 8.3.1,

$$\Gamma := \exp(-rt_n) \mathbf{E}_Q \left(\frac{\partial}{\partial \mathbf{R}_1} \left(\frac{\partial \Phi(\mathbf{Z}_n)}{\partial \mathbf{Z}_n} \frac{\partial \mathbf{Z}_n}{\partial \mathbf{R}_1} \right) \left(\frac{\partial \mathbf{R}_1}{\partial \mathbf{S}_0} \right)^2 \right). \quad (8.3.42)$$

In the last term of this expression we can substitute again the Choleski decomposition (8.3.33), yielding

$$\frac{1}{\mathbf{S}_{0-}^2} \exp(2(r-q)t_1) \exp \left(\int_0^{t_1} \mathbf{v}_s ds \right) \mathbf{RN}_2, \quad (8.3.43)$$

where

$$\mathbf{RN}_2 := \exp \left(-2 \int_0^{t_1} \mathbf{v}_s ds + 2 \int_0^{t_1} \sqrt{\mathbf{v}_s} d\mathbf{B}^{(1)} \right) \quad (8.3.44)$$

is the Radon-Nikodym derivative, which realizes a drift adjustment with $2\sqrt{\mathbf{v}_t}$ of the Brownian motion $d\mathbf{B}_t^{(1)}$, such that $d\widetilde{\mathbf{B}}_t^{(1)} := d\mathbf{B}_t^{(1)} - 2\sqrt{\mathbf{v}_t}dt$ is a \mathbb{P}_2 -Brownian motion, with

$$d\mathbb{P}_2 := \mathbf{RN}_2 d\mathbf{Q}. \quad (8.3.45)$$

Hence, the Gamma is given by

$$\begin{aligned} \Gamma &= \exp(-rt_n) \frac{1}{\mathbf{S}_{0-}^2} \exp(2(r-q)t_1) \cdot \\ &\cdot \mathbf{E}_{\mathbb{P}_2} \left(\exp \left(\int_0^{t_1} \widetilde{\mathbf{v}}_s ds \right) \delta_F(\widetilde{\mathbf{Z}}_n) 1_{\{F_1 \leq \widetilde{\mathbf{R}}_1 \leq C_1\}} + 1_{\{\widetilde{\mathbf{Z}}_n \geq F\}} (\delta_{F_1}(\widetilde{\mathbf{R}}_1) - \delta_{C_1}(\widetilde{\mathbf{R}}_1)) \right), \end{aligned}$$

where

$$\widetilde{\mathbf{Z}}_n = \widetilde{\mathbf{Y}}_1 + \sum_{i=2}^n \mathbf{Y}_i \quad \text{and} \quad \widetilde{\mathbf{Y}}_1 = \max(F_1, \min(C_1, \widetilde{\mathbf{R}}_1)),$$

with

$$\widetilde{\mathbf{R}}_1 := \exp \left((r-q)t_1 + \frac{3}{2} \int_0^{t_1} \mathbf{v}_s ds + \int_0^{t_1} \sqrt{\widetilde{\mathbf{v}}_s} d\widetilde{\mathbf{W}}_s^{(1)} \right), \quad (8.3.46)$$

$$d\widetilde{\mathbf{v}}_t = \alpha_2(\beta_2 - \widetilde{\mathbf{v}}_t)dt + \sigma\sqrt{\widetilde{\mathbf{v}}_t}d\widehat{\mathbf{W}}_t^{(2)}, \quad \alpha_2 = \gamma - 2\sigma\varrho, \quad \beta = \frac{\gamma\theta}{\alpha_2}, \quad (8.3.47)$$

with

$$d\widetilde{\mathbf{W}}^{(1)} := d\widetilde{\mathbf{B}}^{(1)}, \quad d\widetilde{\mathbf{W}}^{(2)} := \varrho d\widetilde{\mathbf{B}}^{(1)} + \sqrt{1 - \varrho^2} d\mathbf{B}^{(2)} \quad (8.3.48)$$

and

$$d\widetilde{\mathbf{W}} = \begin{bmatrix} d\widetilde{\mathbf{W}}^{(1)} \\ d\widetilde{\mathbf{W}}^{(2)} \end{bmatrix} = L d\widetilde{\mathbf{B}}. \quad (8.3.49)$$

Applying again section 8.4.1 the processes $\widetilde{\mathbf{R}}_1$ and $\widetilde{\mathbf{v}}$ can be rewritten as a Heston model (set $\alpha = -\frac{3}{2}$ in section 8.4.1). Hence, under the measure \mathbb{P}_2 the model preserves the Hestonian characteristics.

The last term can be calculated exactly the same way as in section 8.3.1.

Rho hedging

The computation of the ρ follows exactly the same steps as section 8.3.1.

Vega hedging

The sensitivity of the price of the cliquet w.r.t. changes in the volatility in this model is represented by the sensitivity of the price w.r.t. changes in v_0 and the parameters of the Heston model, θ , γ , σ and ϱ . Since the volatility process does not have an explicit expression (the SDE (8.2.11) has no explicit solution) we cannot use the drift adjustment - technique for determining the vega. We will therefore proceed in the following manner. Let us rewrite the price P of the cliquet (8.2.1) as

$$P = \exp(-rt_n) \int \Phi(z) f(z, \alpha) dz$$

where $f(z, \alpha)$ is the joint distribution of the return and volatility, with Laplace transform given by (8.2.12). α is the parameter we are interested in, taking the values $v_0, \theta, \gamma, \sigma$ or ϱ . Hence the vega is obtained as

$$\vartheta_\alpha = \frac{\partial}{\partial \alpha} \int \Phi(z) f(z, \alpha) dz = \int \Phi(z) \frac{\partial}{\partial \alpha} f(z, \alpha) dz. \quad (8.3.50)$$

Since taking Laplace transforms is a linear operator we can obtain the vega ϑ_α by

$$\vartheta_\alpha = \int \Phi(z) \mathcal{L}^{-1} \left(\frac{\partial}{\partial \alpha} \mathcal{L}_z f(z, \alpha) \right) dz, \quad (8.3.51)$$

where $\mathcal{L}_z f(s_z, \alpha)$ is given by (8.2.12). We will present now a general method of how ϑ_α can be computed, the detailed calculations for every α is left to the interested reader.

In case of v_0 the result is immediate, since

$$\frac{\partial}{\partial v_0} \mathcal{L}_{x,v} f(s_x, s_v, v_0) = -p(s_x, s_v) \mathcal{L}_{x,v} f(s_x, s_v, v_0).$$

8.3 Greeks in lognormal and jump diffusion models

In case of $\gamma, \theta\sigma$ and ϱ , the computation is somewhat more complicated because $p(s_x, s_v)$ depends on these parameters, hence the term $-\dot{p}(s_x, s_v)v_0\mathcal{L}_{x,v}f(s_x, s_v, \alpha)$ will figure in the expression which needs to be calculated. Let us show this for σ , the computations for γ and ϱ follow the same line of reasoning. Let us differentiate the recursion relation (8.2.17) w.r.t. σ , then we obtain

$$\begin{aligned} \frac{\partial}{\partial\sigma}\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}}) &= \int_0^\infty \frac{\partial}{\partial\sigma}\mathcal{L}_z\varphi_k(s_z; v_k)\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)dv_k \\ &+ \int_0^\infty \mathcal{L}_z\varphi_k(s_z; v_k)\frac{\partial}{\partial\sigma}\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)dv_k. \end{aligned} \quad (8.3.52)$$

This is a recursion relation for $\frac{\partial}{\partial\sigma}\mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}})$, which can be computed with algorithm 8.2.2 if the term $\frac{\partial}{\partial\sigma}\mathcal{L}_{x,v}f_Y(s_z, s_{v_{k+1}}; v_k)dv_k$ can be calculated. Since f_Y is not easy to work with we will use a result proved in appendix 8.5.1: there is a linear operator \widehat{A} for which identity (8.5.14) holds true, that is,

$$\mathcal{L}_{z,v}f_Y(s_z, s_{v_{k+1}}; v_k) = \widehat{A}(\mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}; v_k))(s_z),$$

(see appendix 8.5.1 for the specification and computation of operator \widehat{A}). Since \widehat{A} is linear we obtain that

$$\frac{\partial}{\partial\sigma}\mathcal{L}_{z,v}f_Y(s_z, s_{v_{k+1}}; v_k) = \widehat{A}\left(\frac{\partial}{\partial\sigma}\mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}; v_k)\right)(s_z). \quad (8.3.53)$$

Evaluating the rhs of the expression above yields

$$\widehat{A}\left(p'_\sigma(\cdot, s_{v_{k+1}})v_k\mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}; v_k) + p(\cdot, s_{v_{k+1}})\mathcal{L}_{z,v}\frac{\partial}{\partial\sigma}f(\cdot, s_{v_{k+1}}; v_k)\right)(s_z).$$

Since f has an explicit, closed-form expression given by (8.2.12) $\frac{\partial}{\partial\sigma}f(\cdot, s_{v_{k+1}}; v_k)$ can be determined straightforwardly; the same holds for $p'_\sigma(\cdot, s_{v_{k+1}})$. In order to compute the term $v_k\mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}; v_k)$ we will proceed as in appendix 8.5.2, and therefore we make the following observation

$$-\frac{\partial}{\partial s}(\mathcal{L}f)(s) = (\mathcal{L}g)(s) \quad \text{with } g(x) = xf(x). \quad (8.3.54)$$

This implies that in order to determine $v_k\mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}; v_k)$ we need to compute

$$\frac{\partial}{\partial s_{v_k}}\mathcal{L}_{z,v,v}f(s_z, s_{v_{k+1}}, s_{v_k}),$$

which we will approximate similarly as relation (8.5.15). We can now use algorithm 8.2.2 for the recursion relation (8.3.52) in order to determine $\theta\sigma$.

Remark 8.3.1. *The sensitivities w.r.t. the parameters of a jump diffusion model (e.g. λ , see (8.2.7)) can be determined in a similar fashion.*

8.4 Computational details

8.4.1 Preservation of the Heston model

This section demonstrates how the models $(\widehat{\mathbf{R}}_1, \widehat{\mathbf{v}})$ (given by (8.3.38) and (8.3.39)) and $(\widetilde{\mathbf{R}}_1, \widetilde{\mathbf{v}})$ (given by (8.3.46) and (8.3.47)) can be rewritten (transformed) as Heston models. Consider the following SDE

$$\begin{aligned} dX_t &= \left(\mu - \alpha \frac{\mathbf{v}_t}{2}\right) dt + \sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(1)} \\ d\mathbf{v}_t &= \gamma(\phi - \mathbf{v}_t) dt + \sigma \sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(2)}. \end{aligned}$$

Let $\alpha > 0$ and define

$$\widehat{X}_t = \alpha X_t \text{ and } \widehat{\mathbf{v}}_t = \alpha^2 \mathbf{v}_t.$$

We obtain that

$$\begin{aligned} d\widehat{X}_t &= \left(\alpha\mu - \frac{\widehat{\mathbf{v}}_t}{2}\right) dt + \sqrt{\widehat{\mathbf{v}}_t} d\mathbf{W}_t^{(1)}, \\ d\widehat{\mathbf{v}}_t &= \gamma(\alpha^2\phi - \widehat{\mathbf{v}}_t) dt + |\alpha| \sigma \sqrt{\widehat{\mathbf{v}}_t} d\mathbf{W}_t^{(2)}. \end{aligned}$$

If $\alpha < 0$ then

$$d\widehat{X}_t = \left(\alpha\mu - \frac{\widehat{\mathbf{v}}_t}{2}\right) dt + \sqrt{\widehat{\mathbf{v}}_t} d\widehat{\mathbf{W}}_t^{(1)},$$

with $\widehat{\mathbf{W}}_t^{(1)} = -\mathbf{W}_t^{(1)}$, that is $\widehat{\rho} = -\rho$.

If $h_{\widehat{x}, \widehat{v}}(s_x, s_v)$ is the Laplace transform of the distribution of \widehat{X}_t and $\widehat{\mathbf{v}}_t$ then

$$\begin{aligned} h_{x,v}(s_x, s_v) &= \mathbb{E} \exp(-s_x X_t - s_v \mathbf{v}_t) \\ &= \mathbb{E} \exp\left(-\frac{s_x}{\alpha} \widehat{X}_t - \frac{s_v}{\alpha^2} \widehat{\mathbf{v}}_t\right) \\ &= h_{\widehat{x}, \widehat{v}}\left(\frac{s_x}{\alpha}, \frac{s_v}{\alpha^2}\right). \end{aligned}$$

8.5 Preliminary: Plancharel's identity and derived approximations

Recall, cf. definition 3.3.4, the inner product $\langle \cdot, \cdot \rangle_n$ be defined by

$$\langle \widehat{f}, \widehat{g} \rangle_n := \int_0^1 \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_n^\#} dv.$$

Recall theorem 4.4.2:

8.5 Preliminary: Plancharel's identity and derived approximations

Theorem 8.5.1. *The inner product $\langle \cdot, \cdot \rangle_n$ satisfies Plancharel's identity*

$$\langle \widehat{f}, \widehat{g} \rangle_n = \langle f, g \rangle. \quad (8.5.1)$$

We can approximate the following integral with a finite sum as

$$\int_0^1 \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_v^n} dv \approx \frac{1}{N} \sum_{k=0}^{N-1} \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_{\frac{k}{N}}^n}. \quad (8.5.2)$$

Using Plancherel's identity we obtain that

$$\langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_v^n} = \sum_{j=0}^{n-1} \langle \Psi \widehat{f}, q_j^v \rangle_{Q_v^n} \langle q_j^v, \Psi \widehat{g} \rangle_{Q_v^n}. \quad (8.5.3)$$

Since $\langle \cdot, \cdot \rangle_{Q_v^n}$ defined by (4.3.8) approximates $\langle \cdot, \cdot \rangle_{Q_v}$ in (4.3.6), we obtain that

$$\langle \Psi \widehat{f}, q_j^v \rangle_{Q_v^n} \approx \sum_k e^{-i2\pi kv} \langle f, \phi_j(\cdot - k) \rangle. \quad (8.5.4)$$

Applying theorem 3.10.1 yields that

$$\int_0^1 \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_v^n} dv - \frac{1}{N} \sum_{k=0}^{N-1} \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{Q_{\frac{k}{N}}^n} = - \sum_{m=0}^{n-1} \sum_{j \neq 0} \sum_k \langle f, \phi_m(\cdot - k - jN) \rangle \langle \phi_m(\cdot - k), g \rangle. \quad (8.5.5)$$

If the support of f and g is contained in $[0, \infty)$ then the rhs of identity (8.5.5) is bounded by

$$\sum_{j \neq 0} \langle |f(\cdot + jN)|, |g| \rangle \leq \sum_{j=1}^{\infty} \|f(\cdot + jN)\| \|g\| + \sum_{j=1}^{\infty} \|g(\cdot + jN)\| \|f\| \longrightarrow 0 \text{ for } N \text{ large enough.}$$

Remark 8.5.2. *The Gauss quadrature used in (8.5.4) works only for smooth functions. However in (8.5.16) we need to deal with the non-smooth functions ϕ_k , which are the capped and floored returns per period. Assume that the function g in (8.5.3) is a non-smooth function which can be written in the form*

$$g(x) = \sum_j g_j(x - j), \quad \text{with } g_j \text{ smooth on } [0, \infty). \quad (8.5.6)$$

Define the function z-transform $V(s, z)$ as

$$V(s, z) = \sum_j z^j \widehat{g}_j(s),$$

which is a smooth function by definition; moreover, its inverse is given by

$$F^{-1}(V)(x, z) = \sum_j z^j g_j(x).$$

Then

$$\widehat{g}(s) = \sum_j \exp(-sj) \widehat{g}_j(s) = V(s, \exp(-s)), \quad (8.5.7)$$

which is a smooth function. We can then use $V(s, \exp(-s))$ in Plancharel's identity (8.5.16) instead of the non-smooth \widehat{g} . That is,

$$\langle \Psi V(\cdot, e^{-2\pi v}), \Psi \widehat{f} \rangle_{Q_v} \approx \langle \Psi V(\cdot, e^{-2\pi v}), \Psi \widehat{f} \rangle_{Q_v^n},$$

and on the other hand,

$$\langle \Psi \widehat{g}, \Psi \widehat{f} \rangle_{Q_v} = \langle \Psi V(\cdot, e^{-2\pi v}), \Psi \widehat{f} \rangle_{Q_v}.$$

In our case, we have the smooth function $g(x)1_{\{x \geq 0\}}$ and we define $g_1(x) = g(x + N)1_{\{x \geq 0\}}$. The (non-smooth) cutoff distribution $g(x)1_{\{x \leq N\}}$ can then be written as $g(x)1_{\{x \leq N\}} = g(x) - g_1(x - N)$.

8.5.1 Operator \widehat{A} and its computation

The random variables Y_k , $k = 1, \dots, n$ are defined as the capped and floored returns with density (8.2.4). It can be shown that one can obtain Y_k from the logarithmic returns through a linear operator. To this end let us define two linear operators. The so-called truncation operator T which transforms the pdf of the returns R_i into the pdf of the capped-floored returns Y_i , given by the rule (8.2.4),

$$Tf_{R_i} = f_{Y_i}. \quad (8.5.8)$$

Secondly, let us define the operator B which transforms the pdf of a random variable into the pdf of the exponential of the random variable, or equivalently, it transforms the pdf of the logarithm of a random variable into the pdf of the random variable itself (we will use this last interpretation):

$$Bf(x) = \frac{f(\ln(x))}{x}. \quad (8.5.9)$$

B is also a linear operator, therefore, the composite operator

$$A = TB \quad (8.5.10)$$

is also linear. Since the Laplace transformation operator is also linear,

$$\widehat{A} = \mathcal{L}A\mathcal{L}^{-1}, \quad (8.5.11)$$

8.5 Preliminary: Plancharel's identity and derived approximations
is also linear in the Laplace domain. Operator \widehat{A} realizes

$$\mathcal{L}f_{Y_i} = \widehat{A}\mathcal{L}f_X.$$

For the computation one can use the following algorithm.

Algorithm 8.5.1.1

1. Compute $\mathcal{L}f$ according to (8.5.12);
2. Invert the LT $\mathcal{L}f$, with algorithm 4.5.5.1;
3. Apply operator B according to (8.5.9);
4. Apply operator T according to rule (8.2.4);
5. Calculate its LT with algorithm 4.5.5.2, according to remark 8.5.3.

Remark 8.5.3. Algorithm 4.5.5.2 can not be used for the Dirac function, which does figure in the expression of the distribution of the Y_i . Hence, for the actual computation of the Laplace transform $\mathcal{L}f_{Y_i}$ one needs to take into account that through the truncation there is a probability mass in the points F_i and C_i . For computational purposes we shift the origin to F_i , and we obtain

$$\mathcal{L}f_{Y_i}(s) = \int_0^{C_i-F_i} \exp(-st) f_X(t + F_i) dt + p_0 + p_1 \exp(-s(C_i - F_i)),$$

where $p_0 = F_X(F_i)$ and $p_1 = 1 - F_X(C_i)$. Furthermore, with the parameters of algorithms 4.5.5.1 and 4.5.5.2, the actual computation will be

$$\mathcal{L}f_{Y_i}(s) = \int_0^{C_i-F_i} \exp(-st) f_X(\Delta t + F_i) dt + p_0 + p_1 \exp(-sm) \exp(-2\pi i \nu m), \quad (8.5.12)$$

with $m = (C_i - F_i) / \Delta$.

8.5.2 Computation of the density of the returns for the Heston model

In order to evaluate the integral (8.2.18) let us define the finite support function

$$\widetilde{f}_Y(\cdot, \cdot; v_k) := f_Y(\cdot, \cdot; v_k) \mathbf{1}_{\{v_k \leq v_M\}}. \quad (8.5.13)$$

Since f_Y has no closed-form expression, we will use a result proved in appendix 8.5.1: there is a linear operator \widehat{A} for which

$$\mathcal{L}_{z,v} f_Y(s_z, s_{v_{k+1}}; v_k) = \widehat{A}(\mathcal{L}_{z,v} f(\cdot, s_{v_{k+1}}; v_k))(s_z). \quad (8.5.14)$$

It follows that the Laplace transform of \tilde{f}_Y is given by

$$\mathcal{L}_{z,v,v}\tilde{f}_Y(s_z, s_{v_{k+1}}; s_{v_k}) = \widehat{A} \left(\frac{1 - \exp(-(p(\cdot, s_{v_{k+1}}) + s_{v_k})v_M)}{s_{v_k} + p(\cdot, s_{v_{k+1}})} \mathcal{L}_{z,v}f(\cdot, s_{v_{k+1}}) \right) (s_z), \quad (8.5.15)$$

(see appendix 8.5.1 for the specification and computation of operator \widehat{A}). Applying Plancharel's identity* (see (8.5.1)) to relation (8.2.18) yields the approximation

$$\begin{aligned} \mathcal{L}_{z,v}\varphi_{k+1}(s_z, s_{v_{k+1}}) &\approx \left\langle \mathcal{L}_{z,v}\varphi_k(s_z, s_{v_k}), \mathcal{L}_{z,v,v_k}\tilde{f}_Y(s_z, s_{v_{k+1}}; s_{v_k}) \right\rangle_n \\ &= \frac{1}{N} \sum_{j=0}^{N-1} \left\langle \Psi \mathcal{L}_{z,v}\varphi_k(s_z, s_{v_k}), \Psi \mathcal{L}_{z,v,v_k}\tilde{f}_Y(s_z, s_{v_{k+1}}; s_{v_k}) \right\rangle_{Q_n^{\frac{j}{N}}} \end{aligned} \quad (8.5.16)$$

which inner product can be computed with relation (4.3.8). Relation (8.5.16) can be calculated recursively, according to the following algorithm. Note that we are interested in the marginal distribution of the capped and floored returns, hence in the last iteration we want to obtain $\mathcal{L}_{z,v}\varphi_n(s_z, 0)$.

Algorithm 8.5.2.1

1. Specify s_z and s_{v_k} as for algorithm 4.5.5.1;
2. $k = 1$;
3. Compute $\mathcal{L}_{z,v}f_Y(s_z, s_{v_k})$ with algorithm 8.2.1 with input the LT $\mathcal{L}_{z,v}f(s_z, s_{v_k})$ (obtained from (8.2.13));
4. Compute $\mathcal{L}_{z,v,v}\tilde{f}_Y(s_z, s_{v_k})$ with (8.5.15);
5. Compute the inner product (8.5.16) with (4.3.8);
6. If $k < N$ then set $k := k + 1$ and go to step 2.
7. Invert $\mathcal{L}_{z,v}\varphi_N(s_z; 0)$ w.r.t. s_z with algorithm 4.5.5.1.

* See remark 8.5.2 over using Plancharel's identity for the non-smooth functions φ_k .

Chapter 9

High dimensional transformation algorithms

9.1 Introduction

Efficient algorithms for the computation of d -dimensional Laplace and inverse Laplace transforms know a wide range of end users from financial modeling, through telecommunication and computer systems to petrochemical research. A specific example is a financial model, where the underlying is modeled as stochastic process with stochastic volatility; time varying or stochastic interest rate would add another dimension, and so on.

This chapter is a direct follow-up of the one dimensional transformation algorithms of chapter 4. The one-dimensional algorithm, based on a Gaussian quadrature rule, computes the coefficients of the orthogonal expansions of the desired function (in the time domain, in the case of the inverse Laplace transform algorithm, or in the Laplace domain in the case of the Laplace transform algorithm). This means that the original function (obtained after the numerical inversion) as well as the Laplace transform (generated by the Laplace transform algorithm) can be evaluated in arbitrary points x_k (instead of only in previously determined points). The method is stable and converges fast. However necessary, stability and speed for the one-dimensional problem does not guarantee the same characteristics for the multi-dimensional problem. Fast computation characteristics are guaranteed by recognizing and exploiting the matrix-Kronecker product structure of the matrix representations for the quadrature formulae. This structure facilitates transparent notation and efficient computation; although it might seem that one needs to work with sizable matrices. We also feel the necessity of stressing once again that the d -dimensional method also works both ways: one can obtain the original function (in arbitrary points) by using numerical values of its

Laplace transform as input to the algorithm, and one can also obtain the Laplace transform of a function in arbitrary points, if some numerical values of the original function are given. Also important to realize that closed-form expressions neither of the original function nor of the Laplace transform are needed, solely some of their numerical values as input.

The scholastic literature on multi-dimensional Laplace (Fourier) transform algorithms is scarce. Moorthy (1995) is somewhat related to our method in the sense that it also uses a Fourier series representation, he applies directly the Poisson summation formula. Although our method also uses the PSF, it is based on Gaussian quadrature which is absent in Moorthy's method. Another distinctive inversion algorithm, based on a Laguerre method, is developed by Abate, Choudhury and Whitt (1998). They developed an FFT-based extension of a previously developed variant of their Fourier-series method to calculate the coefficients of the multidimensional Laguerre generating function. They also use methods for scaling to accelerate convergence of infinite series, using Wynn's ε -algorithm and exploiting geometric decay rates of Laguerre coefficients.

The chapter is structured as follows. Section 9.3 presents the cornerstone of the method, the main idea behind the one-dimensional transformation algorithms. Section 9.4 generalizes the method for the one-dimensional case to d -dimensional transformation algorithms. Section 9.5 explains how to structure the matrix representations of the quadrature rules exploiting the properties of Kronecker products, then section 9.6.7 shows how to exploit this structure for the purpose of efficient implementation. Finally, section 9.6.3 presents the numerical results.

The results of this chapter are described in the working paper den Iseger and Oldenkamp (2008).

9.2 High dimensional transforms in practice: A (geometric) basket option example

Geometric basket options are exotic derivatives with payoff determined by the difference between the geometric average value of the underlying d assets (S_t^1, \dots, S_t^d) , and a previously fixed strike price. That is, using the notation

$$\mathbf{A}_t := \left(\prod_{i=1}^d S_t^i \right)^{\frac{1}{d}}, \tag{9.2.1}$$

the payoff upon exercise of this (call) option is given by $V(t, \mathbf{A}_t) := 1_{\{\mathbf{A}_t \geq K\}}$.

9.2 High dimensional transforms in practice:

(geometric) basket option example

Let us first consider the well-known Black-Scholes model. If \mathbf{X}_t^i denotes the logarithm of the asset price \mathbf{S}_t^i , then $\log(\mathbf{A}_t) = \frac{1}{d} \sum_{i=1}^d \mathbf{X}_t^i$, with $\mathbf{X}_t^i \stackrel{d}{\sim} \mathcal{N}\left(\left(r - q - \frac{1}{2}\sigma^2\right)t, t\sigma^2\right)$, \mathcal{N} denoting the normal CDF, and $\stackrel{d}{\sim}$ denoting 'distributed as'. Geometric averaging basket options can be priced via a closed form analytic solution due to the fact that the geometric average of the underlying prices follows a lognormal distribution as well:

$$\log(\mathbf{A}_t) \stackrel{d}{\sim} \mathcal{N}\left(\left(r - q - \frac{1}{2}\sigma^2\right)t, t\frac{\sigma^2}{d}\right).$$

With this analytic solution we can then compare the result using a d -dimensional transformation algorithm (see also Broadie and Glasserman(1997), who compared the analytic solution to check their results from the stochastic mesh method). There are countless applications where there is no analytic solution, then approximations are the only way. The high dimensional transformation algorithms can offer in these cases a reliable and accurate approximation.

A lot more complicated problem we are going to consider now is when the underlying is a Heston-like process. Let $\mathbf{X}_t := \ln\left(\frac{\mathbf{S}_t}{\mathbf{S}_0}\right)$, then

$$d\mathbf{X}_t = \left(\mu - \frac{\mathbf{v}_t}{2}\right) dt + \sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(1)}, \tag{9.2.2}$$

with \mathbf{v}_t the stochastic volatility and $\mathbf{W}_t^{(1)}$ a Brownian motion. The dynamics of the volatility $\sqrt{\mathbf{v}}$ is described by the following Ornstein-Uhlenbeck process

$$d\mathbf{v}_t = \gamma(\theta - \mathbf{v}_t) dt + \sigma\sqrt{\mathbf{v}_t} d\mathbf{W}_t^{(2)}, \tag{9.2.3}$$

where $\mathbf{W}_t^{(2)}$ has correlation ρ with $\mathbf{W}_t^{(1)}$ (θ is the mean, γ the mean reversion rate, and σ is the volatility of the volatility). The two dimensional Laplace transform of the joint distribution of $f(x, v; v_0)$ of \mathbf{X} and the volatility \mathbf{v} , given the initial volatility v_0 , is given in Lord and Kahl (2007). The distribution of $\log(\mathbf{A}_t)$ is no longer obvious, moreover, it is a tedious task to estimate it with traditional methods.

$$\log(\mathbf{A}_t) = \frac{1}{d} \sum_{i=1}^d \mathbf{X}_t^i,$$

is not a convolution, but we do have an explicit expression for the joint distribution, hence we can use a d dimensional inversion algorithm.

9.3 The basis: a brief review of the one-dimensional Laplace and inverse Laplace transformation algorithms

First we introduce the basic ideas for the simple numerical Laplace transform and inverse Laplace transform algorithms, which then will be extended to two dimensional and then generalized to multi-dimensional Laplace transforms. The numerical (inverse) Laplace transform algorithm we are going to generalize is that of Den Iseger (2006b). The algorithm works on the entire line, both ways: if some numerical values of the original function are given, then the algorithm generates function values of its Laplace transform (in arbitrary points), and vice versa, given some numerical values of the Laplace transform, the original function can be obtained in arbitrary points. Bearing this to the underlying spaces, we can say that, given input from the time domain, the algorithm transfers it to results in the Laplace domain, and vice versa, from the Laplace domain to the time domain. However, this transfer does not realize directly but through the spaces of the coefficients of the expansions: the coefficients $\langle f, \phi_k(\cdot - j) \rangle$ of the Legendre expansions, and $\langle \Psi \hat{f}, q_k^v \rangle_v$, defined properly below. The link between the two spaces of expansion coefficients is realized through a powerful identity, a new version of the so called Poisson Summation Formula (PSF), first sketched by Den Iseger (2006b). This identity relates the Fourier series of the sequence of Legendre expansion coefficients, namely $\left\{ \sum_j e^{-2\pi i v} \langle f, \phi_k(\cdot - j) \rangle \right\}_{k \in \mathbb{N}_0}$, $v \in [0, 1]$,

to the coefficients of the expansions of the Laplace transformations, $\langle \Psi \hat{f}, q_k^v \rangle_v$. See Mallat (2001) for the classical PSF, and Theorem 9.3.2 for the modified PSF (MPSF). Diagram 9.1 sketches the way the method works:

The Legendre polynomials $\{\phi_k : k \in \mathbb{N}_0\}$ are a complete orthogonal polynomial set in $L^2([0, 1])$. Therefore every function in L^2 can be approximated with an expansion in these polynomials; they are given by

$$\phi_k(t) = \frac{\sqrt{2k+1}}{k!} \partial^k (t^k (1-t)^k).$$

Remark 9.3.1. A straightforward computation of the Laplace transform of the polynomials $\{\phi_k : k \in \mathbb{N}_0\}$ yields

$$\hat{\phi}_k(s) = \int_0^1 e^{-st} \phi_k(t) dt = s^{-1} \left(p_k(s^{-1}) - (-1)^k e^{-s} p_k(-s^{-1}) \right), \quad (9.3.1)$$

with

$$p_k(s) = \sqrt{2k+1} \sum_{j=0}^k \frac{j+k!}{k-j!} \frac{(-s)^j}{j!}. \quad (9.3.2)$$

9.3 The basis: a brief review of the one-dimensional Laplace and inverse Laplace transformation algorithms

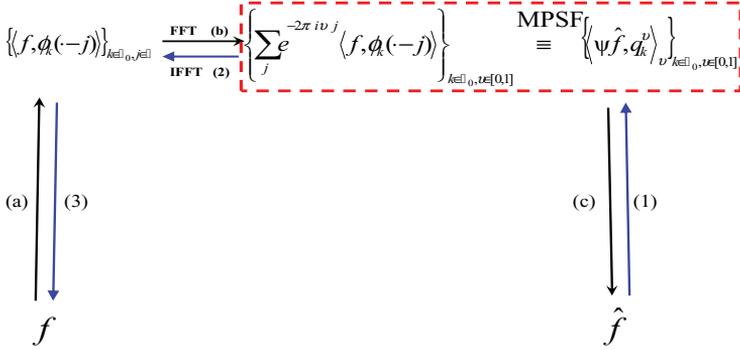


Figure 9.1: Diagram of the flow of the algorithms; the LT: (a) → (b) → (c), and the inverse transform: (1) → (2) → (3)

Since the shifted Legendre polynomials $\{\phi_k(\cdot - j) : k \in \mathbb{N}_0, j \in \mathbb{Z}\}$ are a complete orthogonal set in $L^2((-\infty, \infty))$, any function $f \in L^2((-\infty, \infty))$ can be expanded into

$$f = \sum_{k=0}^{\infty} \sum_{j=-\infty}^{\infty} \langle f, \phi_k(\cdot - j) \rangle \phi_k(\cdot - j). \tag{9.3.3}$$

Given an inner product $\langle \cdot, \cdot \rangle$ and a set of orthogonal polynomials with respect to that inner product, one can always define a so called Gaussian quadrature rule. In case of the Legendre polynomials, we approximate $\langle f, \phi_k(\cdot - j) \rangle$ by a Gaussian quadrature formula (for Gaussian quadrature see the seminal work of Szegő (1975)), given by

$$\langle f, \phi_k(\cdot - j) \rangle_n = \sum_{k=0}^n \alpha_k f_j(\lambda_k) \phi_k^*(\lambda_k), \tag{9.3.4}$$

where $f_j(t) := f(j + t)$, $\lambda_k, k = 0, \dots, n$ are the roots of ϕ_{n+1} , and α_k are the so-called Christophel numbers, given by

$$\alpha_k := \frac{1}{\sum_{j=0}^n |\phi_j(\lambda_k)|^2}. \tag{9.3.5}$$

Recall that in chapters 3 and 4 we introduced the Hilbert space L^2_v with the inner product $\langle \cdot, \cdot \rangle_v$ given by

$$\langle h_1, h_2 \rangle_v := \sum_{k=-\infty}^{\infty} \beta_k^v h_1(\eta_k^v) h_2^*(\eta_k^v), \quad \eta_k^v = \frac{1}{\lambda_k^v}, \beta_k^v = |\mu_k^v|^2, \tag{9.3.6}$$

with $\{\lambda_k^v : k \in \mathbb{Z}, 0 < v < 1\}$, $\{w_k^v : k \in \mathbb{Z}, 0 < v < 1\}$ and Ψ be given by

$$\lambda_k^v = i(2\pi(k+v)) \quad w_k^v(t) = \exp(-\lambda_k^v t),$$

and

$$\Psi h(s) = s^{-1}h(s^{-1}), \tag{9.3.7}$$

respectively. We proceed to use the notation $h = \Psi \hat{f}$. Note also that $\Psi h = \hat{f}$, resulting from the definition of Ψ (see relation (9.3.7)). We identify a set of polynomials $\{q_k^v : k \in \mathbb{N}\}$ in L_v^2 , given by

$$q_k^v(s) := p_k(s) - (-1)^k \exp(-2\pi i v) p_k(-s), \tag{9.3.8}$$

with the polynomials p_k defined by (9.3.2). Since $e^{-s^{-1}} = e^{-2\pi i v}$ on $\{(2\pi i(k+v))^{-1} : k \in \mathbb{Z}\}$ the following fundamental identity fulfills

$$q_k^v = \Psi \hat{\phi}_k \quad \text{on } \{(2\pi i(k+v))^{-1} : k \in \mathbb{Z}\}. \tag{9.3.9}$$

Due to this fundamental identity that we can draw a modification of the classical Poisson Summation Formula, and, consequently, the Laplace transform inversion technique of chapters 3 and 4 are realized:

Theorem 9.3.2 (MPSF). *The following relation holds true*

$$\langle \Psi \hat{f}, q_k^v \rangle_v = \sum_{j=-\infty}^{\infty} e^{-2\pi i j v} \langle f, \phi_k(\cdot - j) \rangle, \tag{9.3.10}$$

where the RHS is a Fourier series in the coefficients of the Legendre expansion, and $\hat{f} \in L^2(-i\infty, i\infty)$ denotes the Fourier (Laplace) transform.

As an important consequence,

$$\langle q_k^v, q_l^v \rangle_v = \langle \Psi \hat{\phi}_k, \Psi \hat{\phi}_l \rangle_v, \tag{9.3.11}$$

which also implies that since $\{\phi_k, k \in \mathbb{N}_0\}$ is a complete orthogonal set in $L^2([0, 1])$, then $\{q_k^v : k \in \mathbb{N}\}$ also forms a complete orthogonal set in L_v^2 . Therefore any function $h \in L_v^2$ can be expanded in these polynomials identically to (9.3.3)

$$h = \sum_{k=0}^{\infty} \langle h, q_k^v \rangle_v q_k^v. \tag{9.3.12}$$

Likewise as in the case of the the inner product defined in the time domain, $\langle \Psi \hat{f}, q_k^v \rangle_v$ can be approximated with the Gaussian quadrature rule $\langle \Psi \hat{f}, q_k^v \rangle_{vn}$, defined by

$$\langle h_1, h_2 \rangle_{vn} := \sum_{k=0}^n \alpha_k^v h_1(\mu_k^v) h_2^*(\mu_k^v), \tag{9.3.13}$$

9.4 Multi-dimensional transformation algorithms

with $\mu_k^v, k = 0, \dots, n$ the roots of q_{n+1}^v , and α_k^v the Christophel numbers related to q_{n+1}^v ,

$$\alpha_k^v := \frac{1}{\sum_{j=0}^n |q_j^v(\mu_k^v)|^2}, \quad (9.3.14)$$

see also (9.3.5). This closes the loop of Diagram 9.1, and the method can be summarized in the following algorithms.

Inverse Laplace Transform (ILT I):

1. Approximate $\langle \Psi \hat{f}, q_k^v \rangle_v$ with $\langle \Psi \hat{f}, q_k^v \rangle_{vn}$ for $k = 0, 1, \dots, n$, according to (9.3.13);
2. Invert $\sum_{j=0}^{\infty} \exp(-i2\pi vj) \langle f, \phi_k(\cdot - j) \rangle$ with the IFFT algorithm;
3. Approximate $f(j+t)$ with $\sum_{k=0}^n \langle f, \phi_k(\cdot - j) \rangle \phi_k(t)$.

The reverse of the algorithm computes the Laplace transform, given specific function values:

Laplace Transform (LT I):

- a. Approximate $\langle f, \phi_k(\cdot - j) \rangle$ with $\langle f, \phi_k(\cdot - j) \rangle_n$ for $k = 0, 1, \dots, n$, according to (9.3.4);
- b. Compute $\langle \Psi \hat{f}, q_k^v \rangle_v$ with the FFT;
- c. Approximate $\Psi \hat{f} \approx \sum_{k=0}^n \langle \Psi \hat{f}, q_k^v \rangle_{vn} q_k^v$ with (9.3.13).

Remark 9.3.3. *If in step c. of Algorithm LT I the computation is performed in the points $\mu_k^v, k = 0, \dots, n$, then we have an identity instead of an approximation. For further details see Remark 3.7 of Den Iseger (2006b), and its precedents. This remark is also valid for all the multi-dimensional algorithms.*

9.4 Multi-dimensional transformation algorithms

Let us first proceed with extending the algorithm described in diagram 9.1 to functions $f(t_1, t_2) \in L^2(-\infty, \infty) \oplus L^2(-\infty, \infty)$. All the results can be then straightforwardly generalized to the n -dimensional case. To this end, all definitions and result from (9.3.3) to (9.3.10) need to be extended. First of all, we need to recall the following result.

Remark 9.4.1. Let X and Y be two Hilbert spaces, then $X \oplus Y$ is also a Hilbert space. Furthermore, if $\{\zeta_1\}$ and $\{\zeta_2\}$ are orthonormal bases for X and Y respectively, then $\{\zeta_1\} \otimes \{\zeta_2\}$ is an orthonormal basis for $X \oplus Y$.

Thus, the shifted Legendre polynomials $\{\phi_{k,l}((x,y) - (j,m)) := \phi_k(x-j)\phi_l(y-m) : k,l \in \mathbb{N}_0, j,m \in \mathbb{Z}\}$ are a complete orthogonal set in $L^2((-\infty, \infty) \oplus L^2(-\infty, \infty))$, any function $f \in L^2(-\infty, \infty) \oplus (-\infty, \infty)$ can be expanded into

$$f = \sum_{k,l=0}^{\infty} \sum_{j,m=-\infty}^{\infty} \langle f, \phi_{k,l}(\cdot - (j,m)) \rangle \phi_{k,l}(\cdot - (j,m)). \quad (9.4.1)$$

Similarly to the one dimensional case, we define a Gaussian quadrature

$$\langle f, \phi_{k,l}(\cdot - (j,m)) \rangle_n = \sum_{k=0}^n \sum_{l=0}^n \alpha_k \alpha_l f_{(j,m)}(\lambda_k, \lambda_l) \phi_{k,l}^*(\lambda_k, \lambda_l), \quad (9.4.2)$$

where $\lambda_k, \lambda_l, k, l = 0, \dots, n$ are the roots of ϕ_{n+1} , and α_k, α_l are the Christphel numbers given by (9.3.5).

In the Laplace domain we do the same, and we also need to redefine the inner product (9.3.6) for two-variable functions $h(s_1, s_2)$. Let L_v^2 and L_w^2 two Hilbert spaces ("Laplace domains") be defined as above, with complete orthogonal bases $\{q_k^v\}$ and $\{q_l^w\}$ respectively. Then remark 9.4.1 yields that $L_v^2 \oplus L_w^2$ is a Hilbert space with complete orthogonal basis $\{q_{k,l}^{v,w}(s_1, s_2) := q_k^v(s_1)q_l^w(s_2) : k,l \in \mathbb{N}, v,w \in [0,1]\}$. Furthermore we define the inner product $\langle \cdot, \cdot \rangle_{v,w}$ as

$$\langle h_1, h_2 \rangle_{v,w} = \sum_{k,l=-\infty}^{\infty} \alpha_k^v \alpha_l^w h_1(\mu_k^v, \mu_l^w) h_2^*(\mu_k^v, \mu_l^w), \quad (9.4.3)$$

with $\mu_k^v, \mu_l^w, \alpha_k^v$ and α_l^w defined as in relation (9.3.6). Therefore the expansion of $h(s_1, s_2) \in L_v^2 \oplus L_w^2$ is given by

$$h = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \langle h, q_{k,l}^{v,w} \rangle_{v,w} q_{k,l}^{v,w}. \quad (9.4.4)$$

Remark 9.4.1 also implies that one can construct a quadrature formula $\langle \cdot, \cdot \rangle_{v,w,n}$ exactly in the same manner as in (9.3.13), that is,

$$\langle h_1, h_2 \rangle_{v,w,n} := \sum_{k=0}^n \sum_{l=0}^n \alpha_k^v \alpha_l^w h_1(\mu_k^v, \mu_l^w) h_2^*(\mu_k^v, \mu_l^w), \quad (9.4.5)$$

with $\mu_k^v, \mu_l^w, k, l = 0, \dots, n$ the roots of q_{n+1}^v and q_{n+1}^w respectively, and α_k^v, α_l^w the Christphel numbers related to q_{n+1}^v and q_{n+1}^w , respectively, as defined in (9.3.14).

Now, the last step in connecting the full circle of diagram 9.1 is the two dimensional Poisson summation formula. This is given by relation

$$\langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w} = \sum_{j=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} e^{-2\pi i(jv+mw)} \langle f, \phi_{k,l}(\cdot - (j,m)) \rangle. \quad (9.4.6)$$

9.4 Multi-dimensional transformation algorithms

In conclusion, the diagram (that is, the algorithms) remains valid for two dimensional Laplace transforms, with $\{\phi_{k,l}(\cdot - (j, m))\}$ and $\{q_{k,l}^{v,w}\}$ instead of $\{\phi_k(\cdot - j)\}$ and $\{q_k^v\}$, respectively. For computational efficiency we are also going to introduce matrix representations, let us first outline the algorithm.

Inverse Laplace Transform II (ILT II):

1. Approximate $\langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w}$ with $\langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w,n}$ for $k, l = 0, 1, \dots, n$ with (9.4.5);
2. Invert $\sum_{j=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} e^{-2\pi i(jv+mw)} \langle f, \phi_{k,l}(\cdot - (j, m)) \rangle$ of (9.4.6) with the IFFT algorithm;
3. Approximate $f(j+t_1, m+t_2)$ with $\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \langle f, \phi_{k,l}(\cdot - (j, m)) \rangle \phi_{k,l}((t_1, t_2) - (j, m))$ as given in (9.4.1).

Further specification of steps 1. - 3. of the algorithm are given after sketching the Laplace transform algorithm, which computes the Laplace transform, given specific function values:

Laplace Transform II (LT II):

- a. Approximate $\langle f, \phi_{k,l}(\cdot - (j, m)) \rangle$ with $\langle f, \phi_{k,l}(\cdot - (j, m)) \rangle_n$ for $k, l = 0, 1, \dots, n$;
- b. Compute $\langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w}$ (according to (9.4.6)) with the FFT;
- c. Approximate $\Psi \hat{f}$ with $\sum_{k=0}^n \sum_{l=0}^n \langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w,n} q_{k,l}^{v,w}$ according to (9.4.4)*.

Step 1. (and a.) can be efficiently implemented with algorithm 9.6.1. For sake of efficient implementation, a matrix representation of the quadrature rules $\langle f, \phi_{k,l}(\cdot - (j, m)) \rangle_n$ and $\langle \Psi \hat{f}, q_{k,l}^{v,w} \rangle_{v,w,n}$ is used (see section 9.5). These matrix representations involve matrix Kronecker products which can efficiently be implemented by exploiting the properties of matrix Kronecker products; the template for efficient Kronecker product implementation is given in sections 9.6.7 and 9.6.2. **Steps 2. and b.** use the well known FFT and IFFT routines (packages available also in Matlab). With a short explanation about IFFT we also introduce the notion of **damping factors**, see section 9.4.1. The function evaluation in **steps 3. and c.** can be computed with algorithm 9.6.2. See also remark 9.3.3 for the function evaluation in step c. Having already deduced the formulae for the two-dimensional Laplace transforms, by straightforward mathematical induction, one can write down the formulae and algorithms for higher dimensional Laplace transforms. The detailed formulae are placed in appendix 9.6.4.

* See also remark 9.3.3

9.4.1 The IFFT and damped Fourier series

Consider the series

$$G(v) := \sum_{j=0}^{\infty} g_j e^{-2\pi i j v} \quad \text{with} \quad g_j := \langle f(j + \cdot), \phi_k \rangle. \quad (9.4.7)$$

According to the MPSF (see Theorem 9.3.2), $G(v) = \langle \Psi \widehat{f}, q_k^v \rangle_{v,n}$. The N -points IFFT yields

$$\widehat{g}_j = g_j + \sum_{k \neq 0} g_{j+Nk}. \quad (9.4.8)$$

If g_{j+Nk} does not converge fast to 0, then $\widehat{g}_j \not\approx g_j$. Standard theory is the use of the so called damped Fourier series:

$$G_r(v) = \sum_{j=0}^{\infty} r^j g_j e^{-2\pi i j v}, \quad (9.4.9)$$

then the N -points IFFT yields

$$\widehat{g}_j = r^j g_j + \sum_{k \neq 0} g_{j+Nk} r^{j+Nk}, \quad \text{that is, } r^{-j} \widehat{g}_j = g_j + \sum_{k \neq 0} g_{j+Nk} r^{Nk}, \quad (9.4.10)$$

since $r < 1$, r^{Nk} does converge fast to 0. Only problem remaining now is the computation of $G_r(v)$.

Let $\alpha := -\log(r)$, and we also have $f_\alpha(x) = e^{-\alpha x} f(x) \Leftrightarrow \widehat{f}_\alpha(s) = \widehat{f}(s + \alpha)$. Intuition suggests that $G_r(v) = \langle \Psi \widehat{f}_\alpha, q_k^v \rangle_v$, but unfortunately this is not entirely true. Let us therefore investigate $\langle \Psi \widehat{f}_\alpha, q_k^v \rangle_v$.

$$\langle \Psi \widehat{f}_\alpha, q_k^v \rangle_v = \sum_{j=0}^{\infty} e^{-2\pi i j v} \langle f_\alpha(j + \cdot), \phi_k \rangle = \sum_{j=0}^{\infty} e^{-2\pi i j v} r^j \langle f(j + \cdot) e^{-\alpha \cdot}, \phi_k \rangle. \quad (9.4.11)$$

That is, $\langle \Psi \widehat{f}_\alpha, q_k^v \rangle_v = \widetilde{G}_r(v)$, with $\widetilde{g}_j := \langle h(j + \cdot), \phi_k \rangle$ and $h(j + x) = e^{-\alpha x} f(j + x)$. This means that multiplying the result with $e^{\alpha x}$ will lead us finally to $G_r(v)$. Let us summarize our findings in an algorithm.

Algorithm ILTD II:

1. Approximate $\langle \Psi \widehat{f}_\alpha, q_{k,l}^{v,w} \rangle_{v,w}$ with $\langle \Psi \widehat{f}_\alpha, q_{k,l}^{v,w} \rangle_{v,w,n}$, for $k, l = 0, 1, \dots, n$ with (9.4.5);
2. Invert $\sum_{j=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} e^{-2\pi i(jv+mw)} \langle f(j + \cdot, m + \cdot) e^{-\alpha \cdot}, \phi_{k,l} \rangle$ of (9.4.6) with the IFFT algorithm;

9.5 Matrix representations with Kronecker products

3. Approximate $f(j + t_1, m + t_2)$ with $e^{\alpha t} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} e^{-\alpha(j,m)} C_{kljm} \phi_{k,l}(t)$ as given in (9.4.1), with $C_{kljm} := \langle f_{(j,m)} e^{-\alpha \cdot}, \phi_{k,l} \rangle (t = (t_1, t_2))$.

The inverse of this algorithm is the adjustment of LT II for damped Fourier series:

Algorithm LTD II:

- Approximate $e^{-\alpha(j,m)} \langle f_{(j,m)} e^{-\alpha \cdot}, \phi_{k,l} \rangle$ with $e^{-\alpha(j,m)} \langle f_{(j,m)} e^{-\alpha \cdot}, \phi_{k,l} \rangle_n$, $k, l = 0, 1, \dots, n$, $\alpha(j, m) = \alpha_1 j + \alpha_2 m$;
- Compute $\langle \Psi \widehat{f}_\alpha, q_{k,l}^{v,w} \rangle_{v,w}$ (according to (9.4.6)) with the FFT;
- Approximate $\Psi \widehat{f}_\alpha$ with $\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \langle \Psi \widehat{f}_\alpha, q_{k,l}^{v,w} \rangle_{v,w,n} q_{k,l}^{v,w}$ according to (9.4.4) and (9.4.5).

The evaluation of $e^{\alpha x} f(x)$ can be very time-consuming for many x , nevertheless, an efficient way of computing it, is by introducing matrix representations and the multiplication operator M .

9.5 Matrix representations with Kronecker products

The matrix representations for the quadrature rules in the one dimensional case (see den Iseger (2006b)) facilitate efficient computer implementation of the method. For the two- and multi-dimensional case matrix representations can be just as efficiently used by exploiting the properties of Kronecker products, hence generalization to multiple dimensions, as well as implementation becomes very efficient. Let us first handle the *matrix representations in the Laplace domain* and introduce the multiplication operator M induced by the norm $\|\cdot\|_v$, defined by $M(v) : L_v^2 \rightarrow L_v^2$,

$$Mh(s) = sh(s), \text{ w.r.t. the norm } \|\cdot\|_v, \text{ i.e. } \|Mh - g\|_v = 0, \text{ with } g(s) := sh(s). \quad (9.5.1)$$

We denote with \widehat{M} the matrix representation of M w.r.t. the basis $\{q_k^v\}$, that is, $(\widehat{M})_{kj} = \langle Mq_k^v, q_j^v \rangle_v$. In den Iseger (2006b) it is shown that

$$\widehat{M} = \begin{bmatrix} \frac{1}{2} \frac{1 + \exp(-i2\pi v)}{1 - \exp(-i2\pi v)} & -\gamma_0 & 0 & \cdots & 0 & \cdots \\ \gamma_0 & 0 & -\gamma_1 & \ddots & \vdots & \cdots \\ 0 & \gamma_1 & \ddots & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & 0 & -\gamma_{n-1} & \cdots \\ 0 & \cdots & 0 & \gamma_{n-1} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

where the numbers γ_k are given by

$$\gamma_k = \frac{1}{2} \frac{1}{\sqrt{2k+1}} \frac{1}{\sqrt{2k+3}}.$$

The finite rank approximation $M_n : \pi_n \rightarrow \pi_n$ is $M_n = P_n M$, with P_n the orthogonal projector on π_n (the space of polynomials with degree less than or equal to n), and the matrix representation of M_n w.r.t. the basis $\{q_k^v\}$ is given by

$$(\widehat{M}_n)_{kj} = \left\langle P_n M q_k^v, q_j^v \right\rangle_v = \left\langle M q_k^v, q_j^v \right\rangle_v = (\widehat{M})_{kj},$$

$k, j = 0, \dots, n$. Since the matrix \widehat{M}_n is skew symmetric, it is diagonalizable and we can write

$$\widehat{M}_n = V D(\mu^v) V^*,$$

where $D(\mu^v)$ denotes the diagonal matrices with the pure imaginary valued eigenvalues $\{\mu_k^v : k = 0, 1, \dots, n\}$ of \widehat{M}_n , and V is a unitary matrix (that is $VV^* = V^*V = I$), the eigenvectors of \widehat{M}_n are the columns of V . This yields the following matrix representation for the quadrature rule:

$$\left[\left\langle \Psi \widehat{f}, q^v \right\rangle_{vn} \right] = V D \left(\Psi \widehat{f}(\mu^v) \right) V_0 = \bar{V} \left(\Psi \widehat{f}(\mu^v) \right), \quad (9.5.2)$$

with $V_0 = V^* e_0$, and the shorthand notation

$$\bar{V}(v) = V(v) D(V_0(v)). \quad (9.5.3)$$

Now we proceed to generalize (9.5.2) for $\left\langle \Psi \widehat{f}(\cdot, \cdot), q_k^v q_j^w \right\rangle_{vwn}$. The detailed derivation of the result can be read in appendix 9.6.5.

Theorem 9.5.1. *The following identity is valid for all $v, w \in [0, 1]$:*

$$\left\langle \Psi \widehat{f}, q^{v,w} \right\rangle_{v,w,n} = (\bar{V}(v) \otimes \bar{V}(w)) \Psi \widehat{f}(\mu(v) \otimes \mu(w)). \quad (9.5.4)$$

Notation $\Psi \widehat{f}(\mu(v) \otimes \mu(w))$ is further explained in section 9.6.5.

Matrix representations in *the time domain* follow exactly the same logic. The multiplication operator in the time domain is given by $N : L^2[0, 1] \rightarrow L^2[0, 1]$, defined by

$$Nf(t) = tf(t) \text{ w.r.t. the norm on } L^2, \text{ i.e., } \|Nf - g\|_{L^2} = 0, \text{ with } g(t) := tf(t). \quad (9.5.5)$$

We denote with \widehat{N} the matrix representation of N w.r.t. the basis $\{\phi_k\}$, that is, $\widehat{N}_{kj} =$

9.5 Matrix representations with Kronecker products

$\langle N\phi_k, \phi_j \rangle$. In the same manner as for M we obtain that

$$\widehat{N} = \begin{bmatrix} \frac{1}{2} & \theta_0 & 0 & \cdots & 0 & \cdots \\ \theta_0 & \frac{1}{2} & \theta_1 & \ddots & \vdots & \cdots \\ 0 & \theta_1 & \ddots & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & \frac{1}{2} & \theta_{n-1} & \cdots \\ 0 & \cdots & 0 & \theta_{n-1} & \frac{1}{2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

with

$$\theta_k = \frac{k+1}{4k+2} \sqrt{\frac{2k+1}{2k+3}}. \quad (9.5.6)$$

The finite rank approximation $N_n : \pi_n \rightarrow \pi_n$ is $N_n = P_n N$, with P_n the orthogonal projector on π_n , and the matrix \widehat{N}_n is given by $(\widehat{N}_n)_{kj} = (\widehat{N})_{kj}$, $k, j = 0, \dots, n$. Since \widehat{N}_n is symmetric, it is a diagonalizable and we can write

$$\widehat{N}_n = UD(\lambda)U^*, \quad (9.5.7)$$

where $D(\lambda)$ denotes the diagonal matrix with the real valued eigenvalues $\{\lambda_k : k = 0, 1, \dots, n\} \subset [0, 1]$ of \widehat{N}_n . The matrix U is unitary and the eigenvectors of \widehat{N}_n are the columns of U . The quadrature rules in the time domain are therefore represented as

$$[\langle f, \phi \rangle_n] = UD(f(\lambda))U_0 = \overline{U}(f(\lambda)), \quad (9.5.8)$$

with $U_0 = U^*e_0$, $\overline{U} := UD(U_0)$, and $[\langle f, \phi \rangle_n]$ denotes the vector with elements $\langle f, \phi_k \rangle_n$, $k = 0, \dots, n$. Exactly in the same manner as for the Laplace domain, (9.5.8) is generalized to multiple dimensions:

$$[\langle f, \phi \rangle_n] = (\overline{U} \otimes \overline{U}) f(\lambda \otimes \mu), \quad (9.5.9)$$

with $[\langle f, \phi \rangle_n]$ denoting in this case the multi-dimensional vector with elements $\langle f, \phi_{k,l} \rangle_n$, $k, l = 0, \dots, n$. As explained in section 9.4.1, in step 2 of the Inverse Laplace Transform algorithm, and step b. of the Laplace transform algorithm, a damped Fourier series is used in the IFFT (and FFT, respectively) in order to enhance convergence. The following remark gives the matrix representations for $G_r(v)$ (see relation (9.4.9)) with $v = 0.5$:

Remark 9.5.2. For the sake of stability of the one dimensional transformation algorithms (see Den Iseger (2006b), Remark 4.5), the value $v = 0.5$ has been chosen together with the damping factor $\alpha(\bar{v}) = c + i2\pi(\bar{v} - 0.5)$. Then the matrix representation for the quadrature rule, that is the coefficients (9.4.11) with $v = 0.5$, is given by (see Den Iseger (2006b), Theorem 4.5)

$$\langle \Psi \widehat{f}_{\alpha(\bar{v})}, q^{0.5} \rangle_{0.5,n} = UD \left(e^{\alpha(\bar{v})\lambda} \right) U^* \overline{V} (\Psi \widehat{f}_{\alpha(\bar{v})}(\mu^{0.5})).$$

For the rest of the chapter we will use the notation

$$M_{\bar{v}} := UD \left(e^{\alpha(\bar{v})\lambda} \right) U^* \bar{V}, \quad (9.5.10)$$

and we omit the index 0.5, letting q stand for $q^{0.5}$, μ for $\mu^{0.5}$, $\langle \cdot, \cdot \rangle_n$ for $\langle \cdot, \cdot \rangle_{0.5,n}$, and so on. Then, for the two dimensional case we obtain the following matrix representation

$$\left[\left\langle \Psi \widehat{f}_{\alpha(\bar{v}), \alpha(\bar{w})}, q \right\rangle_n \right] = (M_{\bar{v}} \otimes M_{\bar{w}}) \Psi \widehat{f}_{\alpha(\bar{v}), \alpha(\bar{w})} (\mu \otimes \mu). \quad (9.5.11)$$

Details of the computation of these coefficients are given in section 9.6.1.

Remark 9.5.3. For the d -dimensional case we obtain the quadrature formula

$$\left[\left\langle \Psi \widehat{f}_{\alpha(\bar{v})}, q \right\rangle_n \right] = (M_{\bar{v}_1} \otimes \dots \otimes M_{\bar{v}_d}) \Psi \widehat{f}_{\alpha(\bar{v})} (\mu \otimes \dots \otimes \mu), \quad (9.5.12)$$

with $\alpha(\bar{v}) := (\alpha(\bar{v}_1), \dots, \alpha(\bar{v}_d))$. Derivation of this formula can be found at the end of appendix 9.6.5.

Remarks 9.5.2 and 9.5.3 can be deduced in exactly the same manner for the time domain, therefore superfluous description is omitted.

9.6 Computational issues

9.6.1 Detailed computation of the coefficients $\left\langle \Psi \widehat{f}_{\alpha(\bar{v})}, q \right\rangle_n$

The inversion coefficients can be computed efficiently by decomposing relation (9.5.11) step by step, according to the associativity of Kronecker products, and using Algorithm 9.6.7 (given in the next section). Accordingly, the following algorithm gives the details of the computation of step 1. of the Inverse Laplace algorithm.

Algorithm 9.6.1

- step 0.** $g := \Psi \widehat{f}_{\alpha(\bar{v}), \alpha(\bar{w})} (\mu \otimes \mu); l(\bar{v}) := e^{\alpha(\bar{v})\lambda};$
- step 1.** Compute $g := (D \otimes D) g$ with $D := \text{diag} \left(1, e^\alpha, e^{2\alpha}, \dots, e^{(N-1)\alpha} \right);$
- step 2.** Compute $g := (\bar{V} \otimes \bar{V}) g$ with algorithm 9.6.7;
- step 3.** Compute $g := (U^* \otimes U^*) g$ with algorithm 9.6.7;
- step 4.** Compute $l := l(\bar{v}) \otimes l(\bar{w});$
- step 5.** Compute $g := l.*g;^*$
- step 5.** Compute $g := (U \otimes U)g$ with algorithm 9.6.7.

* '.*' is an element by element matrix-multiplication command in Matlab®

9.6 Computational issues

Remark 9.6.1. *The hidden clue behind steps 3 and 4 is two-fold. After step 2 expression*

$$[(UD(l(\bar{v}))) \otimes (UD(l(\bar{w})))] g$$

*remains to be computed. Rearranging the terms between the brackets according to the associative property of Kronecker products, the next step will be to compute $(D(l(\bar{v})) \otimes D(l(\bar{w}))) g$. For any matrix Y and vector Z with proper dimensions, $D(Y)Z = Y.*Z$, hence steps 3 and 4.*

9.6.2 Efficient function evaluation

Having computed the coefficients of expansions, we aim to evaluate the desired function in arbitrary $y = (y_1, \dots, y_d)$ points. In fact, for an efficient computation, we aim to perform the implementation in m points at once, $\mathbf{y} := (y^1, \dots, y^m)$, where each y^i is a vector of d elements (y_1^i, \dots, y_d^i) , $i = 1, \dots, m$. This means implementing step 3 of the Inverse Laplace Transform algorithm for \mathbf{y} (see also (9.4.1)), that is,

$$f(\mathbf{y}) \approx \sum_{k_1=0}^{n_1-1} \cdots \sum_{k_d=0}^{n_d-1} C_{\mathbf{kj}} \prod_{i=1}^d \phi_{k_i}(\mathbf{x}_i), \quad (9.6.1)$$

with the multi-indices (see also section 9.6.5) $\mathbf{k} = [k_1, \dots, k_d]$, $\mathbf{j} = [j_1, \dots, j_d]$, where $j_i = \lfloor y_i \rfloor$ and $\mathbf{x}_i = \frac{\mathbf{y}_i - \zeta_i}{\Delta_i} - \mathbf{j}_i$, $i = 1, \dots, d$, and the shift $\zeta = (\zeta_1, \dots, \zeta_d)$. Furthermore

$$C_{\mathbf{kj}} := \left\langle f, \phi_{k_1, \dots, k_d}(\cdot - (j_1, \dots, j_d)) \right\rangle.$$

The following algorithm computes $f := f(\mathbf{y})$:

Algorithm 9.6.2.1:

Step 1. $f_{\mathbf{k}} := \sum_{k_d=0}^{n_d-1} C_{\mathbf{kj}} \phi_{k_d}(\mathbf{x}_d)$;

Step 2. For $i = d - 1$ to 1 do [Step 3];

Step 3. $f = f_{\mathbf{k}} = \sum_{k_i=0}^{n_i-1} f_{\mathbf{k}} \phi_{k_i}(\mathbf{x}_i)$.

However, this algorithm can only be implemented with efficient matrix implementation. To this end, let us define the operator $[\cdot]$, which translates a multi-index into a linear index (see also section 9.6.5):

$$[\mathbf{k}] := \sum_{i=1}^d k_i \prod_{l=1}^{i-1} N_l. \quad (9.6.2)$$

Let us further define the sparse matrices Φ^d and Φ^i , with nonzero elements

$$\Phi_{[k_j]l}^d := \phi_{\mathbf{k}}(\mathbf{x}_d) \quad \text{and} \quad \Phi_{[k_j]l}^i := \phi_{\mathbf{k}}(\mathbf{x}_i),$$

with l the first index of the matrix $\mathbf{y} = y_l p$, $p = 1, \dots, d$ and reformulate algorithm 9.6.2.1 as follows:

Algorithm 9.6.2.2:

Step 1. $f_{\mathbf{k}_d l} := C_{[k][k_d j]} \Phi^d$;

with \mathbf{k}_d the multi-index $(k_1..k_{d-1})$

Step 2. For $i = d - 1$ to 1 do [Step 3];

Step 3. $f = f_{[k_i]l} = f_{[k_i][k_i]} \Phi^i$.

with \mathbf{k}_i the multi-index $(k_1..k_{i-1})$

Efficient function evaluation is a crucial issue since it can cost more memory/computing time than the rest of the whole inversion algorithm altogether.

9.6.3 Numerical results

For one dimensional Laplace transforms den Iseger (2006b) examined eight analytical test functions that are used in the overview article of Martin and Davies (1979). Here we use exactly the same eight test functions two-by-two and then three-by-three in order to test the accuracy and stability of the two- and three dimensional inversion algorithms, respectively. The parameters used are $M = 1$, $M_2 = 16M$ and $\Delta = 1$ and for all the examples we have taken the damping factor $44/M_2$. Besides the choice of the previous parameters we have taken $n = 16$. The test functions and their Laplace transforms are listed in the following table. The next table lists the error made by

Table 9.1: The test functions and their Laplace transforms

nr	function	Laplace Transform
1	$J_0(x)$	$(s^2 + 1)^{-1/2}$
2	$\exp(-x/2)$	$(s + 1/2)^{-1}$
3	$\exp(-0.2x) \sin(x)$	$((s + 0.2)^2 + 1)^{-1}$
4	1	s^{-1}
5	x	s^{-2}
6	$x \exp(-x)$	$(s + 1)^{-2}$
7	$\sin(x)$	$(s^2 + 1)^{-1}$
8	$x \cos(x)$	$(s^2 - 1)(s^2 + 1)^{-2}$

9.6 Computational issues

inverting the two-dimensional Laplace transforms, obtained by pairing the test functions 1 to 8 of table 9.1. Since for the 3-dimensional case a table would be too sizable,

Table 9.2: The error made by the 2-dimensional inversion algorithm

0.79e-15	0.74e-15	0.39e-15	0.79e-15	0.43e-15	0.29e-15	0.42e-15	0.39e-15
0.73e-15	0.69e-15	0.35e-15	0.74e-15	0.40e-15	0.26e-15	0.38e-15	0.35e-15
0.39e-15	0.35e-15	0.17e-15	0.40e-15	0.20e-15	0.12e-15	0.18e-15	0.17e-15
0.80e-15	0.74e-15	0.41e-15	0.79e-15	0.44e-15	0.31e-15	0.42e-15	0.40e-15
0.43e-15	0.40e-15	0.19e-15	0.45e-15	0.22e-15	0.14e-15	0.21e-15	0.19e-15
0.30e-15	0.26e-15	0.12e-15	0.31e-15	0.14e-15	0.08e-15	0.13e-15	0.12e-15
0.41e-15	0.38e-15	0.18e-15	0.43e-15	0.21e-15	0.13e-15	0.20e-15	0.18e-15
0.39e-15	0.35e-15	0.17e-15	0.40e-15	0.19e-15	0.12e-15	0.18e-15	0.16e-15

we summarized the results in the following plot:

9.6.4 Multi-dimensional Laplace and inverse Laplace transforms

Having already deduced the formulae for the two dimensional Laplace transforms, it is straightforward to write down the formulae and algorithms for higher dimensional Laplace transforms by induction. We need to generalize relations (9.4.1) through (9.4.6). Let us indeed start at relation (9.4.1); by induction, the shifted Legendre polynomials $\{\phi_{k_1}(\cdot - j_1) \dots \phi_{k_d}(\cdot - j_d) : k_i \in \mathbb{N}_0, j_i \in \mathbb{Z}, i = 1, \dots, d\}$ are a complete orthogonal set in $L^2((-\infty, \infty) \oplus \dots \oplus (-\infty, \infty))$, any function $f \in L^2(-\infty, \infty) \oplus \dots \oplus (-\infty, \infty)$ can be expanded into

$$f = \sum_{k_1, \dots, k_d=0}^{\infty} \sum_{j_1, \dots, j_d=-\infty}^{\infty} \left\langle f, \phi_{k_1, \dots, k_d}(\cdot - (j_1, \dots, j_d)) \right\rangle \phi_{k_1, \dots, k_d}(\cdot - (j_1, \dots, j_d)). \quad (9.6.3)$$

In the Laplace domain we do the same; $L_{v_1}^2 \oplus \dots \oplus L_{v_d}^2$ is a Hilbert space with complete orthogonal basis $\{q_{k_1, \dots, k_d}^{v_1, \dots, v_d}(s_1, \dots, s_d) := q_{k_1}^{v_1}(s_1) \dots q_{k_d}^{v_d}(s_d) : k_1, \dots, k_d \in \mathbb{N}, v_1, \dots, v_d \in [0, 1]\}$. Furthermore we define the inner product $\langle \cdot, \cdot \rangle_{v_1, \dots, v_d}$ as

$$\langle h_1, h_2 \rangle_{v_1, \dots, v_d} = \sum_{k_1, \dots, k_d=0}^{\infty} \alpha_{k_1}^{v_1} \dots \alpha_{k_d}^{v_d} h_1(\mu_{k_1}^{v_1}, \dots, \mu_{k_d}^{v_d}) h_2^*(\mu_{k_1}^{v_1}, \dots, \mu_{k_d}^{v_d}). \quad (9.6.4)$$

The expansion of $h(s_1, \dots, s_d) \in L_{v_1}^2 \oplus \dots \oplus L_{v_d}^2$ is given by

$$h = \sum_{k_1, \dots, k_d=0}^{\infty} \left\langle h, q_{k_1, \dots, k_d}^{v_1, \dots, v_d} \right\rangle_{v_1, \dots, v_d} q_{k_1, \dots, k_d}^{v_1, \dots, v_d}. \quad (9.6.5)$$

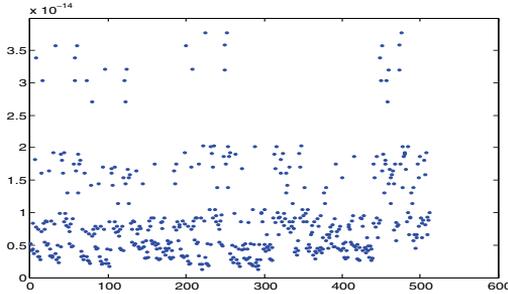


Figure 9.2: The error of approximation with the 3-dimensional inversion algorithm (the eight test functions taken three by three)

The construction of the quadrature formula $\langle \cdot, \cdot \rangle_{v_1, \dots, v_d, n}$ is done in exactly the same manner as for two dimensions, that is,

$$\langle h_1, h_2 \rangle_{v_1, \dots, v_d, n} := \sum_{k_1, \dots, k_d=0}^n \alpha_{k_1}^{v_1} \dots \alpha_{k_d}^{v_d} h_1(\mu_{k_1}^{v_1}, \dots, \mu_{k_d}^{v_d}) h_2^*(\mu_{k_1}^{v_1}, \dots, \mu_{k_d}^{v_d}), \quad (9.6.6)$$

with $\mu_{k_i}^{v_i}, k_i = 0, \dots, n$ the roots of $q_{n+1}^{v_i}, i = 1, \dots, d$ respectively, and $\alpha_{k_i}^{v_i}$ the Christoffel numbers related to $q_{n+1}^{v_i}, i = 1, \dots, d$, respectively. Now, the last step is the d -dimensional Poisson summation formula. This is given by relation

$$\left\langle \Psi \hat{f}, q_{k_1, \dots, k_d}^{v_1, \dots, v_d} \right\rangle_{v_1, \dots, v_d} = \sum_{j_1, \dots, j_d = -\infty}^{\infty} e^{-2\pi i(j_1 v_1 + \dots + j_d v_d)} \left\langle f, \phi_{k_1, \dots, k_d}(\cdot - (j_1, \dots, j_d)) \right\rangle. \quad (9.6.7)$$

9.6 Computational issues

In conclusion, the diagram (that is, the algorithms) remains valid for d -dimensional Laplace transforms.

9.6.5 Matrix representation for the two-dimensional quadrature formulas

Meaning of notation $\Psi \hat{f}(\mu(v) \otimes \mu(w))$ and its construction in d dimensions

Before sketching the matrix denote as $\Psi \hat{f}(\mu^x(v) \otimes \mu^y(w))$, let us specify the form of $\mu^x(v)$.

$$\begin{bmatrix} \mu_0^x(v_0) & \mu_0^x(v_1) & \cdots & \mu_0^x(v_m) \\ \mu_1^x(v_0) & \mu_1^x(v_1) & \cdots & \mu_1^x(v_m) \\ \vdots & \vdots & \cdots & \vdots \\ \mu_n^x(v_0) & \mu_n^x(v_1) & \cdots & \mu_n^x(v_m) \end{bmatrix}.$$

Hence, $\Psi \hat{f}(\mu^x(v) \otimes \mu^y(w))$ is given by the matrix

$$\begin{bmatrix} \Psi \hat{f}(\mu^x(v), \mu_0^y(w_0)) & \Psi \hat{f}(\mu^x(v), \mu_0^y(w_1)) & \cdots & \Psi \hat{f}(\mu^x(v), \mu_0^y(w_m)) \\ \Psi \hat{f}(\mu^x(v), \mu_1^y(w_0)) & \Psi \hat{f}(\mu^x(v), \mu_1^y(w_1)) & \cdots & \Psi \hat{f}(\mu^x(v), \mu_1^y(w_m)) \\ \vdots & \vdots & \cdots & \vdots \\ \Psi \hat{f}(\mu^x(v), \mu_n^y(w_0)) & \Psi \hat{f}(\mu^x(v), \mu_n^y(w_1)) & \cdots & \Psi \hat{f}(\mu^x(v), \mu_n^y(w_m)) \end{bmatrix},$$

with $\Psi \hat{f}(\mu^x(v), \mu_i^y(w_j))$ given by

$$\begin{bmatrix} \Psi \hat{f}(\mu_0^x(v_0), \mu_i^y(w_j)) & \Psi \hat{f}(\mu_0^x(v_1), \mu_i^y(w_j)) & \cdots & \Psi \hat{f}(\mu_0^x(v_m), \mu_i^y(w_j)) \\ \Psi \hat{f}(\mu_1^x(v_0), \mu_i^y(w_j)) & \Psi \hat{f}(\mu_1^x(v_1), \mu_i^y(w_j)) & \cdots & \Psi \hat{f}(\mu_1^x(v_m), \mu_i^y(w_j)) \\ \vdots & \vdots & \cdots & \vdots \\ \Psi \hat{f}(\mu_n^x(v_0), \mu_i^y(w_j)) & \Psi \hat{f}(\mu_n^x(v_1), \mu_i^y(w_j)) & \cdots & \Psi \hat{f}(\mu_n^x(v_m), \mu_i^y(w_j)) \end{bmatrix}.$$

Hence the Kronecker product – notation. Typically for Kronecker products, in d -dimensions the dimensions of the matrix $\Psi \hat{f}(\mu^1(v^1) \otimes \mu^2(v^2) \otimes \dots \otimes \mu^d(v^d))$ increases significantly, therefore one needs a clever method for constructing this matrix. In fact, one only needs to introduce d matrices L_1, \dots, L_d , constructed from $\mu^1(v^1), \dots, \mu^d(v^d)$ such that $\Psi \hat{f}(L_1, L_2) = \Psi \hat{f}(L_1) \otimes \Psi \hat{f}(L_2)$.

Algorithm 9.6.5:

Input: $\mu_1(v^1), \dots, \mu^d(v^d)$;

Output: $L(1), \dots, L(d)$;

Step 0. $L(1) := \mu^1(v^1)$;

Step 1. For $j = 2$ to d

```

E := ones (size(L(j - 1)));
L(j) = μj(vj) ⊗ E;
for k = 1 to j - 1
    E = ones (size(L(k)));
    L(k) = E ⊗ L(k).

```

While writing it out $\Psi\hat{f}(\mu^1(v^1) \otimes \mu^2(v^2) \otimes \dots \otimes \mu^d(v^d))$ is very complicated, it is quite simple to realize with algorithm 9.6.5.

An important (programming) tool: reshape

The vectors and matrices featuring in this chapter are of large dimensions with multi-index. The multi-index

$[\text{index}(1), \text{index}(2), \dots, \text{index}(d)]$, with $\text{index}(k) = 0, \dots, n_k - 1$, notation is interpreted by most programming languages either column-wise (Matlab, Fortran) or row-wise (C). We will adopt here the column-wise data storage interpretation. Each index is interpreted as a linear index, that is index no. $\sum_{k=1}^d \text{index}(k) \prod_{j=1}^{k-1} n_j$ of the single long vector

created from the columns of the multi-index matrix. According to the multi-index usage of programming languages, the reshape command only rearranges the dimensions, it does not actually rearranges the data, which makes it a ‘costless’ operation. As opposed to reshape, a transpose command does rearrange all data, hence it is a time and memory consuming operation. Let now A be a multi-indexed matrix with $[\text{index}]$ rows and $[k_1 k_2 \dots k_m]$ columns; we adopt the notation $A_{[\text{index}][k_1 k_2 \dots k_m]}$. Reshaping A into a matrix with $[\text{index } k_1]$ rows and $[k_2 \dots k_m]$ columns is denoted as $A_{[\text{index } k_1][k_2 \dots k_m]}$.

9.6.6 Detailed deduction of the two-dimensional quadrature formula

This section aims to show that relation (9.5.4) is valid, that is,

$$\left\langle \Psi\hat{f}, q^{v,w} \right\rangle_{v,w,n} = (\bar{V}(v) \otimes \bar{V}(w)) \Psi\hat{f}(\mu(v) \otimes \mu(w)).$$

Let us first suppose that $\Psi\hat{f}(\cdot, \cdot)$ is of the form $\Psi\hat{f}_1(\cdot) \Psi\hat{f}_2(\cdot)$. Then,

$$\Psi\hat{f}(\mu^v \otimes \mu^w) = \Psi\hat{f}_1(\mu^v) \otimes \Psi\hat{f}_2(\mu^w). \tag{9.6.8}$$

Observe that with a separable $\Psi\hat{f}$ we obtain

$$\left\langle \Psi\hat{f}(\cdot, \cdot), q_k^v q_j^w \right\rangle_{v,w,n} \left\langle \Psi\hat{f}_1(\cdot), q_k^v \right\rangle_{vn} \left\langle \Psi\hat{f}_2(\cdot), q_j^w \right\rangle_{wn}, \tag{9.6.9}$$

which together with the matrix representation (9.5.2) yields

$$\left[\left\langle \Psi\hat{f}, q^{v,w} \right\rangle_{v,w,n} \right] = (\bar{V}(v) \Psi\hat{f}_1(\mu^v)) (\bar{V}(w) \Psi\hat{f}_2(\mu^w)) = (\bar{V}(v) \otimes \bar{V}(w)) \Psi\hat{f}(\mu^v \otimes \mu^w), \tag{9.6.10}$$

9.6 Computational issues

using at the last identity the associate property of Kronecker products and relation (9.6.8) for a separable $\Psi\hat{f}$. Substituting $\Psi\hat{f} := q_k^v q_l^w$ yields a result we are going to use later.

$$[\langle q_k^v q_l^w, q^v q^w \rangle_{vwn}] = (\bar{V}(v) \otimes \bar{V}(w)) q_{k,l}^{v,w} (\mu^v \otimes \mu^w). \quad (9.6.11)$$

In order to apply result (9.6.10) to general (i.e. not separable) $\Psi\hat{f}(\cdot, \cdot)$, we observe that through (9.4.4) $\Psi\hat{f}(\cdot, \cdot)$ can be written in the form

$$\Psi\hat{f} = \sum_{k,l=0}^n A_{kl} q_k^v q_l^w, \quad (9.6.12)$$

with $A_{kl} = \langle \Psi\hat{f}, q_k^v q_l^w \rangle_{vwn}$. Combining now (9.6.10), (9.6.12) and (9.6.11) we obtain that

$$\left[\langle \Psi\hat{f}, q^v q^w \rangle_{vwn} \right] = \sum_{k,l=0}^n A_{kl} (\bar{V}(v) \otimes \bar{V}(w)) q_{k,l}^{v,w} (\mu^v \otimes \mu^w). \quad (9.6.13)$$

Since a matrix multiplication (and the Kronecker product) is a linear operator, the above can be written as

$$\left[\langle \Psi\hat{f}, q^v q^w \rangle_{vwn} \right] = (\bar{V}(v) \otimes \bar{V}(w)) \left(\sum_{k,l=0}^n A_{kl} q_{k,l}^{v,w} (\mu^v \otimes \mu^w) \right) = (\bar{V}(v) \otimes \bar{V}(w)) \Psi\hat{f}(\mu^v \otimes \mu^w). \quad (9.6.14)$$

In case of d -dimensional Laplace transformations we obtain by direct generalization that

$$\left\langle \Psi\hat{f}, q^{v_1, \dots, v_d} \right\rangle_{v_1, \dots, v_d, n} = (\bar{V}(v_1) \otimes \dots \otimes \bar{V}(v_d)) \Psi\hat{f}(\mu(v_1) \otimes \dots \otimes \mu(v_d)). \quad (9.6.15)$$

For the implementation of (9.6.15) one can repeatedly apply the efficient implementation method of section 9.6.7. Solely the associative property of Kronecker products needs to be used, that is,

$$[(\bar{V}(v_1) \otimes \dots \otimes \bar{V}(v_{d-1})) \otimes \bar{V}(v_d)] \Psi\hat{f}([\mu(v_1) \otimes \dots \otimes \mu(v_{d-1})] \otimes \mu(v_d)).$$

9.6.7 Efficient implementation with Kronecker products

The matrix representation for the two dimensional quadrature formula (9.5.4), involves matrix- Kronecker products. However short and transparent notation, direct evaluation of these Kronecker products, by sheer size, would cause memory overflow. The aim of this section is therefore to show how one can avoid direct computation by reshaping and rewriting (9.5.4). The expression in (9.5.4) is clearly of the form

$(A_{m_A \times n_A} \otimes B_{m_B \times n_B}) x_{n_A n_B \times 1}$. Reshaping again yields that

$$(A_{m_A \times n_A} \otimes B_{m_B \times n_B}) x_{n_A n_B \times 1} = \begin{bmatrix} a_{11}B & \cdots & a_{1n_A}B \\ \vdots & & \vdots \\ a_{m_A 1}B & \cdots & a_{m_A n_A}B \end{bmatrix} \begin{bmatrix} x_{n_B \times 1}^1 \\ \vdots \\ x_{n_B \times 1}^{n_A} \end{bmatrix}.$$

With simple matrix notation this comes down to

$$(A_{m_A \times n_A} \otimes B_{m_B \times n_B}) x_{n_A n_B \times 1} = \sum_{j=1}^{n_A} a_{kj} \left(B_{m_B \times n_B} x_{n_B \times 1}^j \right). \quad (9.6.16)$$

The following algorithm computes (9.6.16) in Matlab®.

Algorithm 9.6.7:

Input: $A_{m_A \times n_A}$, $B_{m_B \times n_B}$, $x_{n_A n_B \times 1}$

Output: $Z = (A \otimes B)x$

$x = \text{reshape}^*(x, n_B, \text{numel}(x)/n_B);$

$y = Bx;$

$y^T = \text{reshape}(\text{transpose}(y), \text{numel}(y)/n_A);$

$Z = Ay^T.$

Remark 9.6.2. Algorithm 9.6.7 can be used iteratively for the computation of $(A_1 \otimes A_2 \otimes \dots \otimes A_d)x$ for the d -dimensional case, using the associative property of Kronecker products.

* Reshapes vector x into no of rows(x)/ $n_B = n_A$ pieces of size $n_B \times 1$ vectors, see appendix 9.6.5

Chapter 10

A fast Laplace inversion method for the solution of linear partial differential equations

10.1 Introduction

This chapter describes a fast and accurate numerical method to solve linear partial differential equations of elliptic and parabolic types. The partial differential equations being considered are the Helmholtz equation with Dirichlet boundary condition and the Black and Scholes (and Heat) equation in two spatial coordinates. The domains in which these equations are solved include the rectangle, the half plane and the quadrant.

The repository of numerical methods to solve parabolic or elliptic linear partial differential equations includes among others the finite difference methods (see for example Richtmyer (1994)), the variational methods including the Finite Element methods (for basic description see Johnson (2009)), the spectral methods and other meshless methods. The spectral methods express the solution as an integral representation using Green's Functions (Polyanin (2002) or Felsen (1994)) resulting in a Fourier series representation in closed rectangular domains. It is well known, that the Green's Function of the Helmholtz equation in a rectangle involves a slowly convergent double summation (Duffy (2004)). Thus, in order to evaluate the integral representation this slowly converging double sum must be calculated. The contribution of this chapter is to present an accurate method based on the Poisson Summation Formula (PSF) facilitating the fast evaluation of multidimensional Fourier series. The applicability of our method is tested in two dimensions.

There are a handful of numerical ideas described in the literature facilitating the fast

evaluation of this double sum. One of the recently used methods is based on Ewald's method (Beylkin (2008) applied to calculate the Green's Function of a periodic structure. The authors split the Green's Function in two rapidly converging series to be applied in the spatial and spectral domains. The same, splitting-like trick is used by Linton (1999) to calculate the Green's Function of the Laplace equation: integrating the heat equation he splits the integral with respect to two parts to ensure the fast convergence.

The form of the solution

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \hat{h}(k_{n,x}, k_{m,y}) \exp [i(k_{n,x}x + k_{m,y}y)]$$

facilitates the direct application of the PSF. This can be seen as a two dimensional inverse Fourier transform of $\hat{h}(k_{n,x}, k_{m,y})$ where $\hat{h}(k_{n,x}, k_{m,y})$ encompasses the Fourier transforms of the boundary conditions and the Green's Function of the equation with the proper boundary conditions. Indeed cited by, among others report on the direct application of the PSF for periodic structures. To improve the performance of PSF a Kummer's decomposition is used prior to the application of PSF as reported in . Besides Kummer transform one can use Aitken's Δ method for Laplace Zheng (1998) and Gaál (1999) for the Helmholtz equation in rectangular domain. Splitting of the slowly converging sum, together with the application of the PSF and contour integration is developed by Marshall (1999) to determine the Green's Function of the Laplace equation in a rectangular domain.

Another method being applied is the Shrank's Transform Valerio (2007) and Rogier (2008) for periodic structures. By the nonlinear Shrank's transform we eliminate the oscillation of the original series to obtain a series converging faster than the original one.

As it can be seen, the methods mentioned above introduce splitting or variable transformation to improve the convergence properties of the double sum. Our approach, outlined below, facilitates the treatment of the double sum in an uniform manner regardless the spatial coordinates (den Iseger (2006b)).

Den Iseger (2006b) is an extension of den Iseger (2006a). While den Iseger (2006a) inverts the Laplace transform pointwise, den Iseger (2006b) finds the inverse transform on the whole spatial domain by determining the coefficients of the Legendre expansion.

The method presented below relies on the idea of recasting the above Fourier series in a Legendre series. By introducing a complete orthogonal set of functions in the Fourier domain expressions involving the Fourier transforms of the Legendre polynomials can be evaluated extremely quickly and accurately. This results in some very advantageous features of our method. As a most important advantage the effect of the Gibbs phe-

10.2 Barrier options in the Black-Scholes setting

phenomenon can be reduced by several orders of magnitude facilitating the solution of problems with discontinuous (but bounded) boundary conditions. The accuracy of the method is hardly influenced by taking wave numbers in the order of magnitude of $\kappa \sim 10^2$ (note, that in this case the square of the wave number, κ^2 , occurring in the Helmholtz equation is in the order of magnitude of 10^4). As a consequence, for the 2D case an accuracy of 10^{-10} or higher can be attained without needing too much memory resources for the calculations.

The goal of this chapter is to utilize this great accuracy of the Gaussian quadrature method in solving the Helmholtz equation and the Black&Scholes (BS) equation. Since BS equation can be transformed to the Heat equation, its solution can be lead back to solving the Helmholtz equation as described in Section 10.2 below. The BS equation is solved to price a special derivatives contract : the so called barrier options for two underlyings. The barriers are chosen such, that the problem is readily transformed to the solution of the Helmholtz equation in rectangular domain.

The chapter is organized as follows. In Section 10.2 we interrelate the BS and the Heat Equations and pose the problem of pricing barrier options by giving their product description. In Section 10.3 we solve the generalized Dirichlet problem for Helmholtz equation in a rectangle, half plane and quadrant. In Section 10.4 the domain Ω is split in subdomains and the expansion coefficients of each subdomain are determined. Finally in Section 10.5 the pseudo algorithm is given to provide aid for the implementation of the method. The chapter ends with conclusions.

This chapter is based on the working paper Gaál and den Iseger (2009).

10.2 Barrier options in the Black-Scholes setting

The goal of this Section is twofold. Firstly we establish an interlink between the Black-Scholes (BS) partial differential equation to price contingent claims and the Helmholtz equation. Our approach is based on that of : first the BS equation is transformed to the heat equation, then the heat equation is transformed to the Helmholtz equation by means of Laplace transform. The goal of these transforms is that our fast inverse Laplace transform can be applied transparently on a relative simple partial differential equation.

Secondly, we define the family of contingent claims (barrier options) to be tackled in Section 10.2.2 below. Like Lipton (2002) we chose the multifactor setting to price contingent claims.

10.2.1 Transforming the BS equation into Helmholtz equation

We would like to price contingent claims on two underlyings, S_1 and S_2 in a setting with constant instantaneous volatility and constant spot interest rate. The processes are lognormal satisfying the following stochastic differential equations under the risk neutral measure Q :

$$dS_i = r_i S_i dt + \sigma_i S_i dW_i^Q \quad (10.2.1)$$

$$\langle dW_1^Q, dW_2^Q \rangle = 0 \quad (10.2.2)$$

As discussed in many standard textbooks (see eg. Lipton (2002)) the price of the contingent claim denoted by $\phi(S_1, S_2, t)$ satisfies the BS differential equation :

$$\left[\sum_{i=1}^2 \left(\frac{\sigma_i^2}{2} S_i^2 \partial_{S_i S_i}^2 + r_i S_i \partial_{S_i} \right) - r_0 + \partial_t \right] \phi(S_1, S_2, t) = 0 \quad (10.2.3)$$

Following the terminology of Lipton (2002) Eq. (10.2.3) describes the *three country economy*. There terminal and boundary conditions belonging to (10.2.3) are discussed in Section 10.2.2 below.

The BS equation is a linear, second order PDE of parabolic type. By eliminating the convection term it can be transformed to the standard heat equation Lipton (2002) as stated by the following

Lemma 10.2.1. (*interrelation between BS and heat equation*) *The BS equation can be transformed to the heat equation of the following form :*

$$\left(\partial_{x_1}^2 + \partial_{x_2}^2 - \partial_\tau \right) \psi(x_1, x_2, \tau) = 0$$

Proof. Let us define the mapping as follows : $F : (0, \infty) \times (0, \infty) \times R \rightarrow R^2 \times R$

$$F : (S_1, S_2, t) \rightarrow \left(x_1 = \frac{\sqrt{2}}{\sigma_1} \ln S_1, x_2 = \frac{\sqrt{2}}{\sigma_2} \ln S_2, \tau = T - t \right) \quad (10.2.4)$$

and introduce

$$\psi(x_1, x_2, \tau) = \phi(x_1, x_2, \tau) \exp[-\alpha\tau - \beta_1 x_1 - \beta_2 x_2]$$

with

$$\alpha = -r_0 - \frac{(\sigma_1 \gamma_1)^2 + (\sigma_2 \gamma_2)^2}{2} \quad \gamma_1 = -\frac{1}{2} + \frac{r_1}{(\sigma_1)^2} \quad \gamma_2 = -\frac{1}{2} + \frac{r_2}{(\sigma_2)^2}$$

Applying F the BS equation is transformed into

$$0 = (\mathcal{L} - \partial_\tau) \psi(x_1, x_2, \tau) \quad (10.2.5)$$

with

$$\mathcal{L} = \partial_{x_1 x_1}^2 + \partial_{x_2 x_2}^2$$

□

10.2 Barrier options in the Black-Scholes setting

After Laplace transform (10.2.5) takes the following form

$$(\mathcal{L} + \kappa^2) \tilde{\psi}(x, y) = \Phi(x, y, s) \tag{10.2.6}$$

where $\Phi(x, y, s)$ encapsulates the terminal condition of the BS problem and

$$\begin{aligned} \tilde{\psi}(x_1, x_2) &= \int_0^\infty \psi(x_1, x_2, t) e^{-st} dt \\ \kappa^2 &= -s \end{aligned}$$

Remark 10.2.2. Transform (10.2.4) above affects the boundary conditions as well. This is described in the following subsection

10.2.2 Product description of four barriers option

Barrier options are one of the most traded exotic options on the foreign exchange market. Very often the barrier levels are constant in the state space facilitating the direct application of our numerical method. Sometimes one has to calibrate the option model to the prices of barrier options observable on the market : this necessitates the subsequent multiple solution of the pricing problem in the optimization procedure. These arguments motivate the choice of a typical barrier option on foreign exchange : the quadrupole no touch (or "four barriers") option.

Definition 10.2.3. (quadrupole no touch, or "four barriers" option) Suppose we have two currencies, $S_i, i = (1, 2)$ is our portfolio whose price evaluation in time is described by (10.2.1). Find the price $\phi(S_1, S_2, t)$ of a double barrier option, if the option at $t = T$ pays an amount of

$$\phi(S_1, S_2, T) = v(S_1, S_2)$$

and

$$\phi(S_{1,d}, S_2, t) = \phi_1(S_2, t) \tag{10.2.7}$$

$$\phi(S_1, S_{2,d}, t) = \phi_2(S_1, t) \tag{10.2.8}$$

$$\phi(S_{1,u}, S_2, t) = \phi_3(S_2, t) \tag{10.2.9}$$

$$\phi(S_1, S_{2,u}, t) = \phi_4(S_1, t) \tag{10.2.10}$$

$$\phi(S_1, S_{2,u}, 0) = \phi_0(S_1, S_2) \tag{10.2.11}$$

Here $\phi(S_{1,i}, S_2, t)$ and $\phi(S_1, S_{2,i}, t)$, $i = (u, d)$ denote the amount of rebate to be paid if the upper or the lower barrier is touched, respectively.

The transform (10.2.4) above maps the upper and the lower barrier as well. Indeed, the barrier levels (10.2.7) take the following form in the new coordinate system :

$$x_{i,u} = \frac{\sqrt{2}}{\sigma_i} \ln S_{i,u}$$

$$x_{i,d} = \frac{\sqrt{2}}{\sigma_i} \ln S_{i,d}$$

with $i = (1, 2)$. It can be seen, that rectangles are transformed into rectangles, that is the pricing problem in the new coordinate system can be described by means of Cartesian coordinates, as well.

Remark 10.2.4. *The boundary conditions can be interpreted as follows. If S_1 or S_2 crosses a certain value during their random walk then a rebate is paid. The amount of the rebate depends on the state of the other random factor and the time.*

10.3 The Helmholtz equation

As mentioned in the previous Section (10.2.6) is solved both in a rectangular domain and in a (semi) open segment (half plane, quadrant). Given the boundary conditions we obtain closed form solutions for the problems which are evaluated by means of the multidimensional inverse Laplace transform being the most important contribution of this chapter. The solutions are obtained by in integral representation form using standard techniques described in, e.g. Lipton (2002) and Duffy(2002). Since the construction of Green's functions is slightly different for the open and compact domains being discussed in the following subsections.

10.3.1 The solution in the rectangular domain

Problem 10.3.1. *We seek for the solution of the Helmholtz equation (10.2.6) in the following domain*

$$\bar{\Omega} = [0; 1] \times [0; 1]$$

subject to the Dirichlet boundary condition

$$\tilde{\psi}(x, y)|_{\partial\Omega} = \psi_0$$

describing the rebate at knockout explained in Section 10.2.2

The solution of Problem 10.3.1 is obtained in two steps. First we explain the polarization splitting technique recasting the solution as a linear combination of functions with certain symmetry conditions. In the next step we derive the analytical solution with these symmetry conditions.

10.3 The Helmholtz equation

The polarization splitting

Let us define the even and odd functions as follows:

Definition 10.3.1. $f(z)$, $z = (x, y)$ is said to be even in z if

$$f(z) = f(-z)$$

In the similar fashion $f(z)$ is said to be odd in z if

$$f(z) = -f(-z)$$

Let us introduce the projection operators π_{ij} , $i = (e, o)$ and $j = (e, o)$ with (e, o) abbreviating respectively (even, odd) on $\bar{\Omega}$ as given in Appendix (10.9.1). On domain $\bar{\Omega}$ operator π_{eo} projects $\Phi(x, y)$ to the space of functions being even in x and odd in y on the given domain. Since the projection operator commutes with \mathcal{L} defined in (10.2.5) the following lemma holds:

Lemma 10.3.2. Let $(i, j) = (e, o)$.

$$\pi_{i,j}\mathcal{L}\tilde{\psi}(x, y) = \mathcal{L}\pi_{i,j}\tilde{\psi}(x, y) = \pi_{i,j}\Phi(x, y) \quad (10.3.1)$$

Remark 10.3.3. To solve (10.3.1) one has to transform the boundary conditions of (10.2.5) as well. This is outlined in Appendix, as well.

Since the union of the four subspaces with different symmetries is dense, the solution can be obtained as

$$\tilde{\psi}(x, y) = \tilde{\psi}^{(e,e)}(x, y) + \tilde{\psi}^{(e,o)}(x, y) + \tilde{\psi}^{(o,e)}(x, y) + \tilde{\psi}^{(o,o)}(x, y)$$

where $\tilde{\psi}^{(i,j)}(x, y) = \pi_{i,j}\tilde{\psi}(x, y)$. In the following Subsection the projection as well as the analytic solution to (10.3.1) is defined.

Closed form solution of the Dirichlet problem

Definition 10.3.4. Let us define $\pi_{i,j}$ as follows:

$$\pi_{i,j}\Phi(\mathbf{r}') = \langle G_{i,j}(\mathbf{r}|\mathbf{r}'), \Phi(\mathbf{r}') \rangle$$

Here $G_{i,j}(\mathbf{r}|\mathbf{r}')$ is the fundamental solution bearing certain symmetry properties is defined below in (10.3.3).

With the projection operator defined above the solution of the boundary value problem of (10.3.1) on $\partial\Omega'$ is standard (see, for example Polyanin(2002)) and given by the following Theorem :

Theorem 10.3.5. (Closed form solution of Problem 10.3.1) The solution to Problem 10.3.1 can be expressed as :

$$\tilde{\psi}^{(i,j)}(\mathbf{r}) = \oint_{\partial\Omega'} \psi_0^{(i,j)}(\mathbf{r}') \partial_{n'} G_{i,j}(\mathbf{r}|\mathbf{r}') dr + \int_{\Omega} \Phi(\mathbf{r}') G_{i,j}(\mathbf{r}|\mathbf{r}') dA \quad (10.3.2)$$

Here $\mathbf{r} = [x, y]^T$, $dA = dx dy$ and $G_{i,j}(x, y|x', y')$ is the Green's Function of the Helmholtz equation satisfying the following conditions :

$$\begin{aligned} (\Delta + \kappa^2) G_{i,j}(\mathbf{r}|\mathbf{r}') &= -\delta(\mathbf{r} - \mathbf{r}') \\ G_{i,j}|_{\partial\Omega} &= 0 \end{aligned}$$

The fundamental solution $G_{i,j}$ can be expressed as follows :

Lemma 10.3.6. (Fundamental solution) The fundamental solution of the Helmholtz equation can be obtained as an eigenfunction expansion :

$$G_{i,j}(\mathbf{r}) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{\psi_n(x) \psi_m(y)}{\kappa^2 + \lambda_{n,m}^2} \quad (10.3.3)$$

and

$$\begin{aligned} \psi_n(x) &= \exp [ik_{x,n}x] \\ \psi_m(y) &= \exp [ik_{y,m}y] \\ \lambda_{n,m}^2 &= (ik_{x,n})^2 + (ik_{y,m})^2 \end{aligned}$$

At this point we would like to place some remarks.

Remark 10.3.7. (interpretation of \mathbf{k}). The components of the reciprocal lattice vector (which is called wavenumber vector by some authors)

$$\mathbf{k} = [k_{x,n}, k_{y,m}]^T$$

describe the symmetries of the solution $\psi(\mathbf{r})$ in $\overline{\Omega'}$ with the following wavenumbers :

$$k_{x,n}^{(o)} = 2\pi n \quad k_{x,n}^{(e)} = \pi (2n + 1) \quad (10.3.4)$$

$$k_{y,n}^{(o)} = 2\pi m \quad k_{y,n}^{(e)} = \pi (2m + 1) \quad (10.3.5)$$

Indeed, $\psi_{n,m}(\mathbf{r})$ and $\lambda_{n,m}^2$ are respectively the eigenfunctions and eigenvalues of the following problem :

$$\begin{aligned} (\Delta + \lambda_{n,m}^2) \psi_n(x) \psi_m(y) &= 0 \\ \psi_n(x) \psi_m(y)|_{\partial\Omega'} &= 0 \end{aligned}$$

10.3 The Helmholtz equation

Observe, that

$$\tilde{\psi}_{11}^{(i,j)}(\mathbf{r}) = \oint_{\partial\Omega'} \psi_0^{(i,j)}(\mathbf{r}') \partial_{n'} G_{i,j}(x-x', y-y') dr \tag{10.3.6}$$

has the same symmetry property as $G_{i,j}$. By the proper superposition of $\psi_{11}(\mathbf{r})$ in the four quadrants the solution to Problem 10.3.1 can be obtained.

Remark 10.3.8. By proper choice of discrete values of \mathbf{k} it can be achieved, that the Green's Function satisfies the Neumann conditions on $\partial\Omega$.

Note also, that the solution of (10.3.3) is quasi periodic :

$$\begin{aligned} (\Delta + \kappa^2) G_{i,j}(\mathbf{r} - \mathbf{r}') &= -\delta(\mathbf{r} - \mathbf{r}') \\ G_{i,j}(\mathbf{r} + \mathbf{1}|\mathbf{r}') &= G_{i,j}(\mathbf{r}|\mathbf{r}') \end{aligned}$$

where $\mathbf{1} = [1, 1]^T$ are the sides of the rectangle.

To calculate (10.3.2) we have to evaluate the four terms occurring in the line integral. Since the evaluation is identical for all the four terms, let us consider (10.3.6).

This expression splits in four terms corresponding to the four sides of the rectangle. Let us denote the contribution of $\psi_{11}(x, y)$ on side $a = (x, 1)$ by $\psi_{11}^{(a)}(x, y)$. Dropping the indices denoting the symmetry and introducing notations

$$\begin{aligned} \psi(x, y) &= \sum_n \sum_m k_m \exp [i \langle \mathbf{k}, \mathbf{r} \rangle] h(ik_n) \quad \text{and} \\ h_1^{(a)}(ik_{x,n}) &= \int_0^1 \psi_0(x', 1) \exp(-ik_{x,n}x') dx' \end{aligned}$$

we obtain, that $\psi_{11}^{(a)}(x, y)$ takes the following form:

$$\psi_{11}^{(a)}(x, y) = -\psi(x, y) \tag{10.3.7}$$

Substituting $G_{i,j}$ in (10.3.2) we obtain the following expression for $\psi(\mathbf{r})$:

$$\psi(\mathbf{r}) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{\Psi}_0(k_{x,n}, k_{y,m}) \exp [i (k_{x,n}x + k_{y,m}y)] + \int_{\Omega} \Phi(\mathbf{r}') G_{i,j}(\mathbf{r}|\mathbf{r}') dA \tag{10.3.8}$$

where $\hat{\Psi}_0(k_{x,n}, k_{y,m})$ can be interpreted as the Fourier transform of $\psi_0^{(i,j)} \star \partial_{n'} G_{i,j}$ and can be expressed as

$$\hat{\Psi}_0(k_{x,n}, k_{y,m}) = \sum_{j=1}^4 [p_s]_j h_j(ik_{x,n}, ik_{y,m})$$

where $p_s = (1, -1, -1, 1)$.

10.4 Scaling

The fast and accurate evaluation of (10.3.8) is based on Legendre expansion. We split up the rectangular domain $\overline{\Omega}'$ into rectangular subdomains and we apply Legendre expansion in each subdomain.

To do this let us first define gridpoints in the x and y directions as follows :

$$\begin{aligned}\mathbb{X} &= \left(0, \frac{1}{M}, \frac{2}{M}, \dots, \frac{M-1}{M}, 1\right) \\ \mathbb{Y} &= \left(0, \frac{1}{M}, \frac{2}{M}, \dots, \frac{M-1}{M}, 1\right)\end{aligned}$$

From these grid points the corner points of the rectangular subdomains are determined by

$$\mathbb{P} = \mathbb{X} \otimes \mathbb{Y}$$

The solution is expanded in terms of Legendre polynomials in the subdomains [see (10.4.5) below]:

$$\phi_{j_1}(x) \phi_{j_2}(y)$$

with $0 \leq (j_1, j_2) \leq N$.

Our task is to determine the expansion coefficients of the above polynomials in each subdomain. The remainder of this Section is devoted to explain this procedure. To keep the formulae transparent the results in this Section are stated in one dimension. It is, however, straightforward to generalize them to multiple dimensions.

10.4.1 The expansion coefficients in the subdomains

The scaling is based on the Poisson Summation Formula :

Theorem 10.4.1. (Poisson Summation Formula). In the interval $[0, 1] \times [0, 1]$ the following holds:

$$P_{v_1, v_2} \widehat{\Psi}_0(x, y) := P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) = \sum_{\mathbf{l}=-\infty}^{\infty} \exp[-2\pi i \langle \mathbf{l}, \mathbf{v} \rangle] \psi(\mathbf{x} + \mathbf{l})$$

where the following notation is introduced :

$$P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) := \sum_{\mathbf{n}} \exp \left[\langle \boldsymbol{\lambda}^{(\mathbf{v})}, \mathbf{x} \rangle \right] \widehat{\Psi}_0(\boldsymbol{\lambda}^{(\mathbf{v})}) \quad (10.4.1)$$

with

$$\begin{aligned}\psi(\mathbf{x} + \mathbf{l}) &= \psi(x + l_1, y + l_2) \\ \mathbf{l}^T &= [l_1, l_2], \mathbf{v}^T = [v_1, v_2], \mathbf{x}^T = [x, y] \\ [\boldsymbol{\lambda}^{(\mathbf{v})}]^T &= i2\pi [n + v_1, m + v_2] = [\lambda_1^{(\mathbf{v})}, \lambda_2^{(\mathbf{v})}], \text{ and } \mathbf{n} = (n, m) \in \mathbb{Z}^2\end{aligned}$$

10.4 Scaling

Remark 10.4.2. (symmetries) From Remark 10.3.7 it follows, that $\mathbf{v}^T = \left[\frac{1}{2}, \frac{1}{2}\right]$ for even symmetry and $\mathbf{v} = \mathbf{0}$ for odd symmetry, $\mathbf{v}^T = \left[\frac{1}{2}, 0\right]$ for the even-odd symmetry and $\mathbf{v}^T = \left[0, \frac{1}{2}\right]$ for the odd-even symmetry.

In the remainder of this Section odd-odd symmetry is considered, thus \mathbf{v} is taken to be zero. From the Poisson Summation Formula one has the following result:

Corollary 10.4.3. Let $\mathbf{p}^T = [p_1, p_2]$ with $(p_1, p_2) \in \mathbb{N}^2$. The following holds :

$$P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x} - \mathbf{p}) = \exp[-i2\pi \langle \mathbf{p}, \mathbf{v} \rangle] P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) \quad (10.4.2)$$

First suppose, that $N = 1$, that is $[0, 1] \times [0, 1]$ is split up into four subdomains of equal area

$$\left(\left[0, \frac{1}{2}\right], \left[\frac{1}{2}, 1\right] \right) \times \left(\left[0, \frac{1}{2}\right], \left[\frac{1}{2}, 1\right] \right)$$

For the efficient computation of the coefficients in the subintervals we need the following scaling formula :

Lemma 10.4.4. (upsampling) The following scaling relation holds :

$$P_{\mathbf{v}} \widehat{\Psi}_0\left(\frac{\mathbf{x}}{2}\right) = \left(P_{\frac{v_1}{2}, \frac{v_2}{2}} + P_{\frac{v_1+1}{2}, \frac{v_2}{2}} + P_{\frac{v_1}{2}, \frac{v_2+1}{2}} + P_{\frac{v_1+1}{2}, \frac{v_2+1}{2}} \right) \widehat{\Psi}_0(\mathbf{x}) \quad (10.4.3)$$

Proof. Intuitively (10.4.3) can be seen as splitting the spectrum (10.4.1) into four parts. Indeed, introducing

$$\begin{aligned} \mathbf{n}_1 &= (2n, 2m) & \mathbf{v}_1 &= (0, 0) \\ \mathbf{n}_2 &= (2n, 2m + 1) & \mathbf{v}_2 &= \left(0, \frac{1}{2}\right) \\ \mathbf{n}_3 &= (2n + 1, 2m) & \mathbf{v}_3 &= \left(\frac{1}{2}, 0\right) \\ \mathbf{n}_4 &= (2n + 1, 2m + 1) & \mathbf{v}_4 &= \left(\frac{1}{2}, \frac{1}{2}\right) \end{aligned}$$

and making the substitutions

$$\mathbf{n} \rightsquigarrow \mathbf{n}_i, i = (1, \dots, 4)$$

we obtain, that

$$P_{\mathbf{v}} \widehat{\Psi}_0\left(\frac{\mathbf{x}}{2}\right) = \sum_{i=1}^4 \sum_{\mathbf{n}_i=-\infty}^{\infty} \exp\left[i2\pi \left\langle \frac{\mathbf{n}_i}{2} + \mathbf{v}_i, \mathbf{x} \right\rangle\right] \widehat{\Psi}_0\left(i4\pi \left(\frac{\mathbf{n}_i}{2} + \mathbf{v}_i\right)\right)$$

This is what we had to prove. □

Observe, that we scale (10.4.3) the structure of the expression does not change. Thus Theorem 10.4.4 can be generalized with the following

Corollary 10.4.5. (induction argument) The following holds :

$$P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) = \sum_{k_1=0}^{2^M-1} \sum_{k_2=0}^{2^M-1} P_{\frac{v_1+k_1}{2^M}, \frac{v_2+k_2}{2^M}} \widehat{\Psi}_0(2^M x, 2^M y) = \sum_{\mathbf{k}=0}^{2^M-1} P_{\frac{\mathbf{v}+\mathbf{k}}{2^M}} \widehat{\Psi}_0(2^M \mathbf{x})$$

Theorem 10.4.6. This implies that the induction principle can be used to derive the scaling for M from that for $M - 1$. Furthermore, by applying (10.4.2) the following yields:

$$P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) = \sum_{\mathbf{k}=0}^{2^M-1} \exp \left[i2\pi \left\langle \mathbf{p}, \frac{\mathbf{v} + \mathbf{k}}{2^M} \right\rangle \right] P_{\frac{\mathbf{v}+\mathbf{k}}{2^M}} \widehat{\Psi}_0(2^M \mathbf{x} - \mathbf{p}) \quad (10.4.4)$$

Let us define the Legendre polynomial of j th order on

$$\Omega_{sub} = \left[\frac{p_1}{2^M}, \frac{p_1+1}{2^M} \right] \times \left[\frac{p_2}{2^M}, \frac{p_2+1}{2^M} \right] \quad (10.4.5)$$

$(p_1, p_2) = (0, \dots, 2^M - 1) \times (0, \dots, 2^M - 1)$ as follows:

$$\phi_j^{(\mathbf{p})} = \phi_{j_1}(2^M x - p_1) \phi_{j_2}(2^M y - p_2) := \phi_j(2^M \mathbf{x} - \mathbf{p})$$

Since the set $\{ \phi_j(2^M \mathbf{x} - \mathbf{p}) \}$ is a complete set of functions on Ω_{sub} the following holds:

Theorem 10.4.7. For each $\mathbf{x} \in \Omega_{sub}$ the following holds:

$$P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) = \sum_{\mathbf{j}} a_{\mathbf{j}}^{(\mathbf{p})} \phi_{\mathbf{j}}^{(\mathbf{p})}$$

where

$$a_{\mathbf{j}}^{(\mathbf{p})} = \left\langle P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}), \phi_{\mathbf{j}}^{(\mathbf{p})} \right\rangle$$

Theorem 10.4.7 can be transformed into the following form:

Theorem 10.4.8.

$$a_{\mathbf{j}}^{(\mathbf{p})} = \frac{1}{2^M} \sum_{\mathbf{j}=0}^{2^M-1} \exp \left[i2\pi \left\langle \mathbf{p}, \frac{\mathbf{v} + \mathbf{j}}{2^M} \right\rangle \right] \left\langle P_{\frac{\mathbf{v}+\mathbf{j}}{2^M}} \widehat{\Psi}_0(\mathbf{x}), \phi_{\mathbf{j}}^{(\mathbf{p})} \right\rangle \quad (10.4.6)$$

Proof. Applying (10.4.4) the statement can be proven. □

In general, the coefficients in the p -th subdomain ($0 \leq p \leq 2^M$) are obtained by evaluating

$$\int_{\Omega_{sub}} P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x}) \phi_{\mathbf{j}}(2^M \mathbf{x} - \mathbf{p}) d\mathbf{x}$$

Making use of (10.4.4) we have the following generalization:

10.4 Scaling

Remark 10.4.9. If the interval $[0, 1] \times [0, 1]$ is split up into $2^M \times 2^M$ subdomains of equal

length then we obtain, that

$$a_j^{(\mathbf{p})} = \exp \left[i2\pi \left\langle \mathbf{p}, \frac{\mathbf{v}}{2^M} \right\rangle \right] \text{IFFT} \left(\left\langle P_{\frac{\mathbf{v}+\mathbf{j}}{2^M}} \widehat{\Psi}_0(\mathbf{x}), \phi_j^{(\mathbf{p})}(\mathbf{x}) \right\rangle \right) \quad (10.4.7)$$

For an odd-odd solution $\mathbf{v} = \mathbf{0}$ thus the first factor on the RHS of (10.4.7) reduces to 1. For an even-even solution with $\mathbf{v} = \frac{1}{2}$ this reads as

$$\exp \left[i\pi \left(\frac{p_1}{2^M} + \frac{p_2}{2^M} \right) \right]$$

10.4.2 A corollary : elimination of Gibbs phenomenon

An interesting feature of the method is the absence of spurious oscillations due to the Gibbs phenomenon in the neighborhood of discontinuities. The accuracy of the inverse transform is not affected by any discontinuity or singularity at $\mathbf{x} = 0$, if the function is quadratically integrable on the closed domain.

We can utilize this fact as follows. Let us suppose, that the function to be expanded is discontinuous (or exhibits a singularity) at $\mathbf{x} = \mathbf{p}$. As said above, the solution in the neighborhood of $\mathbf{x} = \mathbf{p}$ can be obtained by applying the shift theorem of the Laplace transform. Therefore once the singularity is shifted to $\mathbf{x} = 0$ so that the Gibbs phenomenon is circumvented. To formalize this statement in one dimension we have the following

Theorem 10.4.10. Let

$$g(\mathbf{x}) = \psi(\mathbf{x} + \mathbf{p})$$

and

$$\psi(\mathbf{x}) = P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x})$$

Then the following holds:

$$P_{\mathbf{v}} \widehat{g}(\mathbf{x}) = \exp [i2\pi \langle \mathbf{p}, \mathbf{v} \rangle] P_{\mathbf{v}} \widehat{\Psi}_0(\mathbf{x})$$

Proof. Applying (10.4.2) on $\widehat{\Psi}_0(\mathbf{x} + \mathbf{p})$ the statement is proven. □

The interpretation is this Theorem is, that one can obtain the solution at the singular point $\mathbf{x} = \mathbf{p}$ by evaluating the expression $\exp [-i2\pi \langle \mathbf{p}, \mathbf{v} \rangle] P_{\mathbf{v}} \widehat{g}(\mathbf{x})$ being singular at $\mathbf{x} = 0$.

Example 10.4.11. Let

$$\psi(x) = \mathbb{1}_{\{x \geq 0\}} - \mathbb{1}_{\{x \geq 1\}} = \psi_1(x) - \psi_2(x)$$

thus its Fourier transform takes the following form :

$$\widehat{\Psi}(k_x) = \int_0^1 \psi(x)e^{-k_x x} dx = \widehat{\Psi}_1(k_x) - \widehat{\Psi}_2(k_x)$$

with

$$\begin{aligned} \widehat{\Psi}_1(k_x) &= \frac{1}{k_x} \\ \widehat{\Psi}_2(k_x) &= \frac{e^{-k_x}}{k_x} \end{aligned}$$

When transforming back $\widehat{\Psi}(k_x)$ one has to note, that $\psi(x)$ exhibits discontinuities at $x = 0$ and at $x = 1$ described in the Fourier domain respectively by $\widehat{\Psi}_1(k_x)$ and $\widehat{\Psi}_2(k_x)$. By virtue of the method $\widehat{\Psi}_1(k_x)$ can readily be transformed back. To eliminate the Gibbs phenomenon for $\widehat{\Psi}_2(k_x)$ a shift has to be imposed, thus the expression

$$P_\nu \widehat{\Psi}_2(x) = \sum_k \psi_2(x+k)e^{-\nu k} = e^{-\nu} \sum \psi(x+k-1)e^{-\nu k}$$

must be considered.

10.4.3 Numerical evaluation of the inner product

Our task boils down to evaluating the argument of the $\text{IFFT}(\cdot)$. To do so, we have to introduce the inner product $\langle \cdot, \cdot \rangle_\nu$ and a set of orthogonal functions $\{q_n^\nu\}$ on L^2 . In this section it is explained how to do this in the Gaussian quadrature framework.

Definition 10.4.12. (inner product, see den Iseger (2006b)) Let us define the inner product needed for the coefficients of expansion for the Laplace transforms as follows:

$$\langle h_1, h_2 \rangle_\nu = \sum_{\mathbf{k}} \left| \mu_{k_1}^{(v_1)} \right|^2 \left| \mu_{k_2}^{(v_2)} \right|^2 h_1 \left(\mu_{\mathbf{k}}^{(\mathbf{v})} \right) h_2 \left(\mu_{\mathbf{k}}^{(\mathbf{v})} \right)$$

with $\left[\mu_{\mathbf{k}}^{(\mathbf{v})} \right]^T = \left[\mu_{k_1}^{(v_1)}, \mu_{k_2}^{(v_2)} \right] = \left[\frac{1}{2\pi i(k_1+v_1)}, \frac{1}{2\pi i(k_2+v_2)} \right]$

and $h \left(\mu_{\mathbf{k}}^{(\mathbf{v})} \right) = h \left(\mu_{k_1}^{(v_1)}, \mu_{k_2}^{(v_2)} \right)$

Definition 10.4.13. (set of orthogonal functions) Let us define $\{q_{\mathbf{k}}^\nu(\mathbf{x})\}$ on L^2 as follows. First, introduce $\{q_k^\nu(x)\}$ in one dimension as

$$\begin{aligned} q_k^\nu &= p_k(l) - (-1)^k \exp(-2\pi i \nu) p_k(-l) \\ p_k(l) &= \sqrt{2k+1} \sum_{j=0}^k \frac{j+l!}{l-j!} \frac{(-k)^j}{j!} \end{aligned}$$

10.4 Scaling

In den Iseger (2006b) it is shown, that $\{q_k^v(x)\}$ is orthogonal in L_v^2 . Using this definition we construct $\{q_k^v(\mathbf{x})\}$ as

$$q_{\mathbf{k}}^v(\mathbf{x}) = q_{k_1}^{v_1}(x) q_{k_2}^{v_2}(y)$$

Having defined the norm and the orthogonal set of functions we can state the following theorem:

Theorem 10.4.14. (den Iseger (2006b)).

$$\langle P_v \widehat{\Psi}_0, \phi_j \rangle = \langle \Theta \widehat{\Psi}_0, q_j^v \rangle_v \quad (10.4.8)$$

$$\text{where } \Theta \widehat{\Psi}_0(\lambda^{(v)}) = \frac{1}{\lambda_1^{(v)} \lambda_2^{(v)}} \widehat{\Psi}_0 \left(\frac{1}{\lambda_1^{(v)}}, \frac{1}{\lambda_2^{(v)}} \right)$$

The RHS of (10.4.8) can be evaluated by means of the Gaussian Quadrature Formula described by den Iseger (2006b) as follows :

$$\begin{aligned} \langle \Theta \widehat{\Psi}_0(\lambda^{(v)}), q_j^v(\mathbf{x}) \rangle_v &\approx \langle \Theta \widehat{\Psi}_0(\lambda^{(v)}), q_j^v(\mathbf{x}) \rangle_{\mathbf{vN}} \\ &= \sum_{\mathbf{k}=0}^N c_{k_1 k_2} \widehat{\Psi}_0(\mu_{\mathbf{k}}^{(v)}) \overline{q_{\mathbf{k}}^v(\mu_{\mathbf{k}}^{(v)})} \end{aligned}$$

where the set $\{c_{k_1 k_2}\}$ are the quadrature weights, $\{\mu_{\mathbf{k}}^{(v)}\}$ are the corresponding quadrature points.

Remark 10.4.15. (Two dimensional quadrature formula) The two dimensional quadrature rule to be considered in the following Section takes the following form :

$$\langle P_v \widehat{\Psi}_0(\mathbf{x}), \phi_j(\mathbf{x}) \rangle = \sum_{k_1=0}^N \sum_{k_2=0}^N c_{k_1 k_2} \widehat{\Psi}_0(\mu_{k_1}^{(v_1)}, \mu_{k_2}^{(v_2)}) \times \quad (10.4.9)$$

$$q_{k_1}^{v_1}(\mu_{k_1}^{(v_1)}) q_{k_2}^{v_2}(\mu_{k_2}^{(v_2)}) \quad (10.4.10)$$

where $\phi_{j_1 j_2}$ is defined as follows on a compact support $\left[\frac{p_1-1}{2^M}, \frac{p_1}{2^M} \right] \times \left[\frac{p_2-1}{2^M}, \frac{p_2}{2^M} \right]$

$$\phi_{j_1 j_2} := \phi_{j_1}(2^M x - p_1) \phi_{j_2}(2^M y - p_2)$$

In the following Section we provide numerical algorithm to evaluate (10.4.9)

10.5 Implementation issues

In this Section we discuss how to evaluate the two dimensional inner product (10.4.9) efficiently. Our approach is to recast (10.4.9) in terms of matrix-vector operations. To do this we introduce multiplication operators for operations in the state space and the spectral domain. We start with the Damping Theorem of Fourier transforms taking the form:

$$\widehat{F}_\alpha(\mathbf{s}) = \widehat{F}(\mathbf{s} + \alpha) \Leftrightarrow f_\alpha(\mathbf{x}) = \exp(\langle \alpha, \mathbf{x} \rangle) f(\mathbf{x}) \quad (10.5.1)$$

From this and the PSF of we obtain the identity :

$$\exp(\alpha x) P_\nu \widehat{\Psi}_0(\alpha + \cdot)(x) = P_{\nu+\alpha} \widehat{\Psi}_0(x) \quad (10.5.2)$$

In den Iseger (2006a) it is shown that the quadrature rule is the most stable for the choice of $\nu = \frac{1}{2}$, we use

$$\left(P_w \widehat{\Psi}_0 \right)(x) = \exp\left(2\pi i \left(w - \frac{1}{2}\right) x\right) P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right)(x) \quad (10.5.3)$$

and apply the quadrature rule on $P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right)$. In what follows we recast formula (10.5.2) in terms of a multiplication operator. Next we approximate this multiplication operator with a finite rank operator. This finite rank operator can be efficiently computed with matrix-vector operations. Next we show that we can also recast the operator P_ν in terms of a multiplication operator, and approximate again the multiplication operator with a finite rank operator. This finite rank operator can again be efficiently computed with matrix-vector operations. Since this is a quick recapitulation of the material of den Iseger (2006b) the theorems are presented in one dimension.

10.5.1 Multiplication operators

Multiplication operators in the state space

In this subsection the factor $\exp(\alpha x)$ of (10.5.2) is presented in terms of a multiplication operator. First let us introduce the multiplication operator N with respect to the $L^2([0, 1])$ defined in the state space as :

$$Nf(x) = xf(x)$$

With the system of Legendre polynomials $\{\phi_k\}$ being orthogonal with respect to this norm let us define matrix representation $\{\widehat{N}_{kj}\}$ as

$$\widehat{N}_{jk} = \langle N\phi_k, \phi_j \rangle$$

10.5 Implementation issues

From the fact that $\{\phi_k\}$ is complete orthogonal set, one can infer that

$$\langle Nf, \boldsymbol{\phi} \rangle = \widehat{N} \langle f, \boldsymbol{\phi} \rangle$$

with $\boldsymbol{\phi}$ denoting a vector of ϕ_n 's so that:

$$\langle h, \boldsymbol{\phi} \rangle : \langle h, \boldsymbol{\phi} \rangle_k = \langle h, \phi_k \rangle$$

Observe that since

$$\widehat{N}_{jk} = \langle N\phi_k, \phi_j \rangle = \langle \phi_k, N\phi_j \rangle = \widehat{N}_{kj}$$

\widehat{N} is symmetric and diagonalizable. We now approximate \widehat{N} with $\widehat{N}_n = \{\widehat{N}_{kj}\}_{k,j=1}^n$.

Since \widehat{N}_n is also symmetric we obtain

$$Nf = \widehat{N}_n \langle f, \boldsymbol{\phi} \rangle_n = \mathbf{U}_n \mathbf{D}(\lambda_n) \mathbf{U}_n^T \langle f, \boldsymbol{\phi} \rangle_n$$

with \mathbf{U} denoting the eigenvectors of \widehat{N}_n and $\mathbf{D}(\lambda_n)$ denoting a diagonal matrix with the eigenvalues λ_n of \widehat{N}_n in its diagonal. With this we arrive at the following:

$$\langle h(N)f, \boldsymbol{\phi} \rangle \approx \mathbf{U}_n \mathbf{D}(h(\lambda_n)) \mathbf{U}_n^T \langle f, \boldsymbol{\phi} \rangle_n \quad (10.5.4)$$

with $\mathbf{D}(h(\lambda_n))$ the diagonal matrix with $h(\lambda_n)$ in its diagonal. Applying this result on formula (10.5.3) yields in particular

$$\langle P_w \ominus \widehat{\Psi}_0, \boldsymbol{\phi} \rangle = \left\langle \exp\left(\left(w - \frac{1}{2}\right) \cdot\right) P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right), \boldsymbol{\phi} \right\rangle \quad (10.5.5)$$

$$= \left\langle \exp\left(2\pi i \left(w - \frac{1}{2}\right) N\right) P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right), \boldsymbol{\phi} \right\rangle \quad (10.5.6)$$

$$\approx \mathbf{U}_n \mathbf{D}\left(\exp\left(2\pi i \left(w - \frac{1}{2}\right) \lambda_n\right)\right) \mathbf{U}_n^T \left\langle P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right), \boldsymbol{\phi} \right\rangle_n$$

In what follows we show that we can also recast the operator $\left\langle P_{\frac{1}{2}} \widehat{\Psi}_0\left(w - \frac{1}{2} + \cdot\right), \boldsymbol{\phi} \right\rangle_n$ in terms of a multiplication operator, and approximate again the multiplication operator with a finite rank operator. This finite rank operator can again be efficiently computed with matrix-vector operations. Since this is a quick recapitulation of the material of den Iseger (2006b) the theorems are presented in one dimension.

Multiplication operators in the spectral domain

Introduce the multiplication operator M with respect to the norm in the Fourier space induced by the inner product:

$$M\widehat{f}(k) = k\widehat{f}(k)$$

Denote

$$\widehat{M}_{jk} = \langle Mq_k^v, q_j^v \rangle$$

From the fact that $\{q_k^v\}$ is complete orthogonal set, one can infer that

$$\langle Nf, \mathbf{q}^v \rangle = \widehat{N} \langle f, \mathbf{q}^v \rangle$$

Observe that since

$$\widehat{M}_{jk} = \langle Mq_k^v, \phi_j \rangle = \langle q_k^v, Mq_j^v \rangle = -\widehat{M}_{kj}$$

(we use here that the support of the innerproduct is pure imaginary) \widehat{M} is skew-symmetric and diagonalizable. By similar arguments presented in 10.5.1 \widehat{M} is diagonalizable:

$$\widehat{M}_{kj} = \mathbf{V} \mathbf{D} (\boldsymbol{\mu}_n^v) \mathbf{V}^*$$

with $\mathbf{D} (\boldsymbol{\mu}_n^v)$ denoting a diagonal matrix with the eigenvalues $\boldsymbol{\mu}_n^v$ of \widehat{M}_n in its diagonal. The inner product can be calculated as

$$\langle \widehat{h}, \mathbf{q}^v \rangle = \langle \widehat{h} (M) \mathbf{1}, \mathbf{q}^v \rangle \quad (10.5.7)$$

$$= \widehat{h} (M_n) \mathbf{e}_0^v \quad (10.5.8)$$

$$= \mathbf{V} \mathbf{D} (h (\boldsymbol{\mu}_n^v)) \mathbf{V}^T \mathbf{e}_0^v \quad (10.5.9)$$

$$= \mathbf{V} \mathbf{D} (\mathbf{V}^T \mathbf{e}_0^v) h (\boldsymbol{\mu}_n^v) \quad (10.5.10)$$

with

$$\mathbf{e}_0^v = \langle \mathbf{1}, \mathbf{q}^v \rangle = \frac{\langle q_0^v, \mathbf{q}^v \rangle}{1 - \exp(-2\pi i v)} = \frac{1}{1 - \exp(-2\pi i v)} [1, 0, \dots, 0]^T$$

Combing formula (10.5.5) with formula (10.4.8) and formula (10.5.7) results in

$$\begin{aligned} \langle P_w \widehat{\Psi}_0, \phi \rangle &= \langle \Theta \widehat{\Psi}_0, q^v \rangle_v \\ &\approx \mathbf{U}_n \mathbf{D} \left(\exp \left(2\pi i \left(w - \frac{1}{2} \right) \lambda_n \right) \right) \mathbf{U}_n^T \mathbf{V} \mathbf{D} \left(\mathbf{V}^T \mathbf{e}_0^{\frac{1}{2}} \right) \Theta \widehat{\Psi}_0 \left(\boldsymbol{\mu}_n^{\frac{1}{2}} + \left(w - \frac{1}{2} \right) \right) \end{aligned}$$

10.5.2 Evaluation of the inner product for higher dimensions

In den Iseger and Oldenkamp (2008) the results of the last section are generalized to higher dimension. The resulting formula is given by

$$\begin{aligned} \langle P_w \widehat{\Psi}_0, \phi \rangle &= \langle \Theta \widehat{\Psi}_0, \mathbf{q}^v \rangle_v \\ &\approx \otimes_{j=1}^2 \left(\mathbf{U}_n \mathbf{D} \left(e^{2\pi i (w_j - \frac{1}{2}) \lambda_n} \right) \mathbf{U}_n^T \mathbf{V} \mathbf{D} \left(\mathbf{V}^T \mathbf{e}_0^{\frac{1}{2}} \right) \right) \Theta \widehat{\Psi}_0 \left(\boldsymbol{\mu}_n^{\frac{1}{2}} + \left(\mathbf{w} - \frac{1}{2} \right) \right) \end{aligned}$$

10.6 The numerical algorithm

here

$$\left\langle P_{\mathbf{w}} \widehat{\Psi}_0, \boldsymbol{\phi} \right\rangle_{k_1+n(k_2-1)} = \left\langle P_{\mathbf{w}} \widehat{\Psi}_0, \phi_{k_1} \phi_{k_2} \right\rangle$$

$$\left\langle \Theta \widehat{\Psi}_0, \mathbf{q}^v \right\rangle_v = \left\langle \Theta \widehat{\Psi}_0, q_{k_1}^{v_1} q_{k_2}^{v_2} \right\rangle_v$$

$$\Theta \widehat{\Psi}_0 \left(\boldsymbol{\mu}_n^{\frac{1}{2}} + \left(\mathbf{w} - \frac{1}{2} \right) \right)_{k_1+n(k_2-1)} = \Theta \widehat{\Psi}_0 \left(\boldsymbol{\mu}_{k_1}^{\frac{1}{2}} + \left(w_1 - \frac{1}{2} \right), \boldsymbol{\mu}_{k_2}^{\frac{1}{2}} + \left(w_2 - \frac{1}{2} \right) \right)$$

for $k, j = 1, 2, \dots, n$. In that chapter it is also shown how to compute the above formula efficiently by exploiting the Kronecker product structure.

10.6 The numerical algorithm

As discussed, we applied a Fourier transform with respect to t to reduce the Heat equation to Helmholtz equation. In the subsequent Section we first discuss the solution algorithm of the Helmholtz equation. Knowing the solution of the Helmholtz equation the algorithm is provided to solve the Heat equation.

10.6.1 Solution algorithm of the Helmholtz equation

Formula (10.4.7) can be evaluated as follows :

Step 1.: Calculate the quadrature points $[\mu]$ in each direction in the frequency domain as follows :

1.1. Define $\widehat{M}_i, i = (k_x, k_y)$ as follows :

$$\widehat{M}_i = \frac{1}{2} \begin{bmatrix} \frac{1+\exp[-i2\pi v^{(i)}]}{1-\exp[-i2\pi v^{(i)}]} & -C_1 & & & \\ & C_1 & 0 & -C_2 & \\ & & C_2 & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}$$

with $C_i = \frac{1}{\sqrt{4n^2-1}}$. Let us denote the eigenvectors and eigenvalues of \widehat{M}_i respectively by \mathbf{V}_i and $\boldsymbol{\lambda}_i$. Here vector $\boldsymbol{\lambda}_i$ contains the eigenvalues in descending order sorted by the imaginary part. Let us introduce

$$\mathbf{B} = M \begin{bmatrix} \frac{1}{\lambda_1} & \frac{1}{\lambda_1} & \frac{1}{\lambda_1} & \cdots & \frac{1}{\lambda_1} \\ \frac{1}{\lambda_2} & \frac{1}{\lambda_2} & \frac{1}{\lambda_2} & \cdots & \frac{1}{\lambda_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\lambda_N} & \frac{1}{\lambda_N} & \frac{1}{\lambda_N} & \cdots & \frac{1}{\lambda_N} \end{bmatrix}$$

where \mathbf{B} is an $N \times M$ matrix with N being the number of expansion coefficients and M being the number of subdomains in the i direction

1.2.a. The quadrature points for the Internal Problem 10.3.1 take the following form :

$$\mathbf{L}_i = \mathbf{A} + \mathbf{B}$$

where \mathbf{A} is an $N \times M$ matrix taking the following form:

$$\mathbf{A} = \begin{bmatrix} i\pi M & i\pi(M+2) & i\pi(M+4) & \cdots & i\pi(3M-2) \\ i\pi M & i\pi(M+2) & i\pi(M+4) & \cdots & i\pi(3M-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ i\pi M & i\pi(M+2) & i\pi(M+4) & \cdots & i\pi(3M-2) \end{bmatrix}$$

1.2.b. To calculate discontinuities (see Example 10.4.11 above) one has to take

$$\mathbf{L}\mathbf{v}_i = \begin{bmatrix} i\pi(M+1) & i\pi(M+3) & i\pi(M+5) & \cdots & i\pi(3M-1) \\ i\pi(M+1) & i\pi(M+3) & i\pi(M+5) & \cdots & i\pi(3M-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ i\pi(M+1) & i\pi(M+3) & i\pi(M+5) & \cdots & i\pi(3M-1) \end{bmatrix}$$

Step 2.: Let

$$\mathbf{L} = \otimes_{i=1}^2 \mathbf{L}_i$$

and

$$\mathbf{L}\mathbf{v} = \otimes_{i=1}^2 \mathbf{L}\mathbf{v}_i$$

Calculate $\widehat{\Psi}_0(\mathbf{L}, \mathbf{L}\mathbf{v})$

Step 3.: Calculate (10.4.7).

Step 4.: Evaluate the IFFT on the RHS of (10.4.7) to obtain an $N \times M^2$ matrix containing the coefficients of the Legendre expansions in each subdomain

Step 5.: Knowing the expansion coefficients obtain the solution in each subdomain with the help of the Legendre expansion.

10.6.2 Solution algorithm of the Heat Equation

The algorithm to calculate the solution of the Heat Equation is as follows :

Step 1. Generate the set of quadrature points $\mathbf{L}^{(t)}$ with the algorithm 1.2.b above.

Step 2. For each quadrature point $\mu_i^{(t)}$ solve the equation

$$\left(\Delta \pm \kappa_0^2 + \mu_i^{(t)} \right) \tilde{\psi}(x, y) = \Phi(x, y)$$

with $+\kappa_0^2$ corresponding to the internal problem and $-\kappa_0^2$ corresponding to the external problem.

Step 3. Given the set of solutions evaluate the IFFT in each $\mathbf{r} \in \Omega$ to get the time dependence of the solution in \mathbf{r} .

10.7 Numerical results

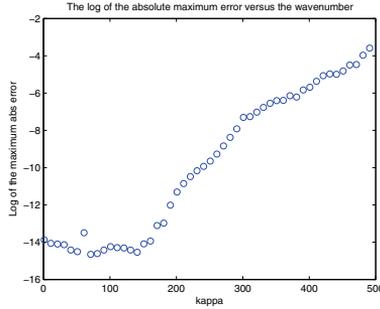


Figure 10.1: The average error on the boundary versus the wavenumber

10.7 Numerical results

In this Section we furnish some numerical results.

First the Helmholtz solver was tested as follows. It is well known, that the Bessel function $\psi_B(x, y) = J_0(\kappa r)$ satisfies (10.2.6) above. The accuracy decreases with an increasing κ^2 . Given $\psi_B(\partial\Omega)$ on the sides for different κ values Eq. (10.2.6) was solved and the absolute error

$$\varepsilon = \int_{\partial\Omega} \left| \psi_B^{(a)} - \psi_B^{(n)} \right| dr$$

was calculated with $\psi_B^{(i)}, i = (a, n)$ denoting respectively the analytical and the numerical solutions of (10.2.6) on the boundary. Our results are displayed on Figure 10.1 :

It can be seen, that the error is very small for small to medium wave numbers and the method remains applicable even for $\kappa^2 \sim 10^3$. This facilitates the scaling of our method enabling its application as a basic PDE solver for problems discretized with rather dense mesh.

As a next numerical example we priced the four barriers option with the following payoff:

$$P(S_1, S_2) = (S_1 + S_2 - 2, 0)^+$$

in $[0.8, 1.2] \times [0.8, 1.2]$ and $\sigma_1 = \sigma_2 = 0.2, r_1 = r_2 = 0.02$. To check the absence of the Gibbs phenomenon, numerical stability and the proper diffusion of option prices the contour plots is presented for $t = 0.5$.

As we see, the Gibbs phenomenon is circumvented.

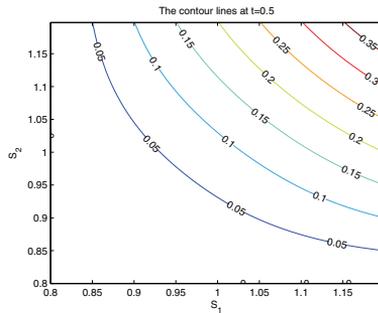


Figure 10.2: The contour lines of the quadrupole no touch option for $t = 0.5$

10.8 Conclusion

In this chapter an accurate and fast inverse Laplace transform was presented to solve linear partial differential equations of the elliptic and parabolic type in a closed domain. The solution in the domain was sought as a Legendre series. The coefficients of the series expansion were determined by means of a Gaussian quadrature based inverse Laplace transform. The great advantage of the method is its accuracy, its speed and the ability to circumvent the Gibbs phenomenon. In this manner problems with bounded but not continuous boundary or initial conditions can be solved in a numerically stable manner. The method is scalable, i.e., it is accurate for a broad range of wave numbers. The favorable properties of the method were demonstrated with numerical examples.

10.9 Computational issues

In this section the projection operators for the four polarizations as well as the transformed boundary conditions are given.

The projection operators take the following form:

$$\begin{aligned}
 \pi_{e,e}\Phi(x,y) &= \Phi(x,y) + \Phi(1-x,y) + \Phi(x,1-y) + \Phi(1-x,1-y) \\
 \pi_{o,e}\Phi(x,y) &= -\Phi(x,y) + \Phi(1-x,y) - \Phi(x,1-y) + \Phi(1-x,1-y) \\
 \pi_{e,o}\Phi(x,y) &= \Phi(x,y) + \Phi(1-x,y) - \Phi(x,1-y) - \Phi(1-x,1-y) \\
 \pi_{o,o}\Phi(x,y) &= -\Phi(x,y) + \Phi(1-x,y) + \Phi(x,1-y) - \Phi(1-x,1-y)
 \end{aligned}
 \tag{10.9.1}$$

Define further

$$f^{(e)}(z) = \frac{f^{(e)}(z) + f^{(e)}(-z)}{2} \quad f^{(o)}(z) = \frac{f^{(o)}(z) - f^{(o)}(-z)}{2}$$

With this definition the transformed boundary conditions take the following form.

If

$$\tilde{\psi}(\partial\Omega) = \begin{aligned} &a(x) \text{ on } y = 1 \\ &b(y) \text{ on } x = 1 \\ &c(x) \text{ on } y = 0 \\ &d(y) \text{ on } x = 0 \end{aligned}$$

then for example $\pi_{i,j}\tilde{\psi}(x,1) = \psi_0^{(i,j)}(x,1)$ and

$$\psi_0^{(i,j)}(x,1) = \begin{array}{cccc} \text{i=e, j=e} & \text{i=o, j=e} & \text{i=e, j=o} & \text{i=o, j=o} \\ \frac{a^{(e)}(x) + c^{(e)}(x)}{2} & \frac{a^{(o)}(x) + c^{(o)}(x)}{2} & \frac{a^{(e)}(x) - c^{(e)}(x)}{2} & \frac{a^{(o)}(x) - c^{(o)}(x)}{2} \end{array}$$

Chapter 11

High dimensional Fourier transform inversion with lattice quadrature rules

11.1 Introduction

In this chapter we present an algorithm which can accurately invert Fourier transforms up to dimension 10. The algorithm combines the quadrature rule with lattice quadrature rules. With the algorithm we can price Basket options up to dimension 10. A typical running time of the algorithm is within 1 second.

The Fourier transform presented in chapter 2, can also be defined for d -dimensional functions as

$$\widehat{f}(\mathbf{s}) = \int_{\mathbb{R}^d} e^{-\langle \mathbf{s}, \mathbf{t} \rangle} f(\mathbf{t}) d\mathbf{t}$$

where

$$\langle \mathbf{s}, \mathbf{t} \rangle = \sum_{k=1}^d s_k t_k$$

11.2 Extending the quadrature rule to higher dimensions

The quadrature rule can easily be extended to higher dimensions by taking the tensor product of the univariate quadrature rule. More precisely, the n -dimensional quadrature rule is defined by

$$\langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v}, n} = \sum_{j \in I_n} \frac{a_j^{\mathbf{v}}}{|\mu_j^{\mathbf{v}}|^2} \widehat{f} \left(\frac{1}{\mu_j^{\mathbf{v}}} \right)$$

Chapter 11. High dimensional Fourier transform inversion with lattice quadrature rules with $I_n = \{0, \dots, n\}^d$, $\mathbf{j} = (\mathbf{j}_1, \dots, \mathbf{j}_d)$, $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_d)$ and

$$\frac{\alpha_{\mathbf{j}}^{\mathbf{v}}}{|\mu_{\mathbf{j}}^{\mathbf{v}}|^2} = \prod_{k=1}^d \frac{\alpha_{\mathbf{j}_k}^{\mathbf{v}_k}}{|\mu_{\mathbf{j}_k}^{\mathbf{v}_k}|^2}$$

$$\widehat{f}\left(\frac{1}{\mu_{\mathbf{j}}^{\mathbf{v}}}\right) = \widehat{f}\left(\frac{1}{\mu_{\mathbf{j}_1}^{\mathbf{v}_1}}, \dots, \frac{1}{\mu_{\mathbf{j}_d}^{\mathbf{v}_d}}\right)$$

We have the following multi-dimensional version of theorem 4.4.2:

Theorem 11.2.1. *The inner product satisfies the following Parseval identity*

$$\int_{[0,1]^d} \langle \Psi \widehat{f}, \Psi \widehat{g} \rangle_{\mathbf{v},n} d\mathbf{v} = \langle f, g \rangle$$

In particular,

$$\int_0^1 \langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n} dv = \int_0^1 \langle \Psi \widehat{f}, \Psi 1 \rangle_{\mathbf{v},n} dv = f(\mathbf{0})$$

Proof. The proof is similar as for the one dimensional case. □

Since $\langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n}$ is a smooth d -dimensional periodic function we can represent $\langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n}$ in the Fourier series

$$\langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n} = \sum_{\mathbf{k} \in \mathbb{Z}^d} f_{\mathbf{k}} \exp(2\pi i \mathbf{k} \cdot \mathbf{v})$$

where

$$f_{\mathbf{k}} = \int_{[0,1]^d} \exp(-2\pi i \mathbf{k} \cdot \mathbf{v}) \langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n} d\mathbf{v}$$

Similar as in the one dimensional case we have

$$f_{\mathbf{0}} = \int_{[0,1]^d} \langle \Gamma \widehat{f}, 1 \rangle_{\mathbf{v},n} d\mathbf{v} = f(\mathbf{0}) \tag{11.2.1}$$

In the following section we discuss the so-called Lattice rules. These quasi-Monte Carlo algorithms are special designed for integrating a periodic d dimensional function f over the hyper cube $[0, 1]^d$. Especially if the Fourier coefficients converge fast to zero, these Lattice rules are very efficient.

11.3 Lattice rules

11.3 Lattice rules

A lattice rule is a quasi-Monte Carlo algorithm special designed for integrating a periodic d dimensional function f over the hyper cube $[0, 1]^d$.

$$I(f) := \int_{[0,1]^d} f(x) dx$$

is approximated with the sum

$$I_M(f) = \frac{1}{M} \sum_{m=1}^M f(t_m)$$

For a Lattice rule the sample points are given by

$$t_m = \left\{ \frac{mg}{M} \right\}$$

where $\{x\}$ denotes the (component-wisely) fractional part of x . We denote with $I_n(f; g)$ a n points Lattice rule with generator g .

The vector $g \in \{0, 1, \dots, n-1\}^d$ is called the generator of the lattice rule with n assumed be a prime number.

We assume that f have a Fourier series representation

$$f(x) = \sum_{h \in \mathbb{Z}^d} \hat{f}(h) \exp(2\pi i \langle h, x \rangle)$$

with $\langle h, x \rangle = \sum_{k=1}^d h_k x_k$ and the Fourier coefficients \hat{f} are given by

$$\hat{f}(h) = \int_{[0,1]^d} f(x) \exp(-2\pi i \langle h, x \rangle) dx$$

Introduce the inner product of f and g by

$$\langle f, g \rangle_w = \sum_{h \in \mathbb{Z}^d} \frac{1}{w_h} \hat{f}(h) \hat{g}^*(h),$$

with the w_h given positive numbers with $w_0 = 1$. We say that f belongs to the space $H(K)$ if $\|f\|_w$ is finite.

11.3.1 The dual Lattice

The Lattice rule can be interpreted as applying a trapezium integration rule over a slice. More precisely, since f is periodic,

$$\begin{aligned} \frac{1}{M} \sum_{m=1}^n f\left(\left\{\frac{mg}{M}\right\}\right) &= \frac{1}{M} \sum_{m=1}^M f\left(\frac{m}{M}g_1, \dots, \frac{m}{M}g_d\right) \\ &= \frac{1}{M} \sum_{m=1}^M f_g\left(\frac{m}{M}\right) \end{aligned}$$

where the function f_g is given by

$$f_g(v) = f(vg_1, \dots, vg_d)$$

From this we obtain the representation

$$\begin{aligned} \frac{1}{M} \sum_{m=1}^M f\left(\left\{\frac{mg}{M}\right\}\right) &= \sum_{h \in \mathbb{Z}} \hat{f}_g(hM) \\ &= \hat{f}(0) + \sum_{h \in L_g} \hat{f}(h) \end{aligned}$$

with L_g the so called dual Lattice

$$L_g = \left\{ h \in \mathbb{Z}^d \setminus \{0\} : \langle h, g \rangle \bmod M = 0 \right\}$$

11.4 Reproducing kernels

The reproducing kernel K is given by

$$\begin{aligned} K(x, y) &= K(x - y) \\ &= \sum_{h \in \mathbb{Z}^d} w_h \exp(2\pi i h \cdot (x - y)) \end{aligned}$$

K is called the reproducing kernel since

$$f(y) = \langle f, K(\cdot - y) \rangle_w$$

Since $I(f) = \langle f, 1 \rangle_w$ we obtain for the approximation error

$$\begin{aligned} I(f) - I_M(f) &= \langle f, 1 \rangle_w - \frac{1}{M} \sum_{m=1}^M \langle f, K(\cdot - t_m) \rangle_w \\ &= \left\langle f, 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m) \right\rangle_w \end{aligned}$$

11.4 Reproducing kernels

Therefor the squared worst error e_M^2 is given by

$$\frac{\left\langle 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m), 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m) \right\rangle_w}{\left\| 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m) \right\|_w} = \left\| 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m) \right\|_w$$

Since K is a reproducing kernel we obtain that

$$\begin{aligned} \sup_{\|f\|_w \leq 1} |I(f) - I_M(f)|^2 &= \left\| 1 - \frac{1}{M} \sum_{m=1}^M K(\cdot - t_m) \right\|_w^2 \\ &= 1 - \frac{2}{M} \sum_{m=1}^M \langle 1, K(\cdot - t_m) \rangle + \frac{1}{M^2} \sum_{k,m=1}^M \langle K(\cdot - t_k), K(\cdot - t_m) \rangle_w \\ &= \frac{1}{M^2} \sum_{k,m=1}^M K(t_m - t_k) - 1 \end{aligned}$$

Since K is periodic, we finally obtain

$$\begin{aligned} \frac{1}{M^2} \sum_{k,m=1}^M K(t_m - t_k) - 1 &= \frac{1}{M^2} \sum_{k,m=1}^M K\left(\frac{m-g}{M}\right) - 1 \\ &= \frac{1}{M} \sum_{k=1}^M K(t_k) - 1 \end{aligned}$$

Hence, we obtain that

$$\begin{aligned} \sup_{\|f\|_w \leq 1} |I(f) - I_M(f)|^2 &= I_M(K) - I(K) \\ &= I_M(K) - 1 \end{aligned}$$

Thus the worst error e_M is the approximation error for the kernel K .

11.4.1 Finding good lattice rules

Theoretically we can find a good Lattice rule by

$$\arg \min_g [I_M(K; g) - 1]$$

However the number of possible generating vectors is $O(M^s)$, which makes the problem intractable. In Korobov (1960) the search space is limited to generating vectors of the form

$$g = \left(1, a, \dots, a^{d-1}\right) \bmod M$$

This research the number of possible generating vectors to n . In the search can be even reduced to $a \in [1, \dots, \frac{M}{2}]$. Computing $I_M(K; g)$ cost $O(dM)$ time, thus the running time

Chapter 11. High dimensional Fourier transform inversion with lattice quadrature rules is $O(dM^2)$ time. The search algorithm is tractable for moderate large M , say $M = 10^5$, for larger M the search become practical infeasible.

In the next section we discuss a search algorithm which can find good Lattice points in $O(dM \log(M))$ time.

11.4.2 Component by Component search

If the weights are multiplicative then the Kernel is of tensor product form. This is seen as follows: Let $w_h = \prod_{j=1}^d w_{h_j}$ then $K(x)$ is equal to

$$\begin{aligned} \sum_{h \in \mathbb{Z}^d} \prod_{j=1}^d w_{h_j} \exp(2\pi i h \cdot x) &= \sum_{h \in \mathbb{Z}^d} \prod_{j=1}^d w_{h_j} \exp(2\pi i h_j x_j) \\ &= \prod_{j=1}^d K_j(x_j) \end{aligned}$$

where

$$\begin{aligned} K_j(x) &= \sum_{h_j \in \mathbb{Z}} w_{h_j} \exp(2\pi i h_j x_j) \\ &= 1 + \sum_{h_j \in \mathbb{Z} \setminus \{0\}} w_{h_j} \exp(2\pi i h_j x_j) \end{aligned}$$

In Sloan and Reztsov (2002) the component by component search algorithm is introduced:

Algorithm:

for $j=2$ to d do

$$g_j = \arg \min_z e^2(g_1, \dots, g_{j-1}, z), z = 1, 2, \dots, \frac{M}{2}$$

The running time of the algorithm is $O(d^2 M^2)$. In Kuo (2003) it is shown that this algorithm find lattice rules which achieve the optimal convergence rate.

In Nuyens and Cools (2006) an algorithm based on the fast Fourier transform is introduced that performs the component by component search algorithm in $O(dM \log(M))$ time.

11.5 The inverse Fourier transform

We get the following inversion formula

$$F_{N,g}^{-1} \hat{f}(t) := \frac{1}{M} \sum_{k=1}^M \sum_{j=1}^n \alpha_j \left(\left\{ \frac{kg}{M} \right\} \right) \exp \left(\left\langle i \lambda_j \left(\frac{kg}{M} \right), t \right\rangle \right) \hat{f} \left(\lambda_j \left(\frac{kg}{M} \right) \right)$$

11.6 Numerical experiments for the inverse Fourier transform

with $j = (j_1, \dots, j_d)$, $\left| \alpha_j \left(\left\{ \frac{k_g}{M} \right\} \right) \right| = \prod_{m=1}^d \alpha_{j_m} \left(\left\{ \frac{k_{g_m}}{M} \right\} \right)$, $\lambda_j \left(\frac{k_g}{M} \right) = (\lambda_{j_1} \left(\frac{k_{g_1}}{M} \right), \dots, \lambda_{j_d} \left(\frac{k_{g_d}}{M} \right))$ and $\sum_{j=1}^n = \sum_{j_1, \dots, j_d=1}^n$.

Theorem 11.5.1. *The operator $F_{N,g}^{-1}$ satisfies the following properties*

1. $F_{N,g}^{-1} \widehat{f} * g = F_{N,g}^{-1} \widehat{f * g} = F_{N,g}^{-1} (\widehat{f} \widehat{g})$
2. Let D be a differential operator and M be a multiplication operator. Then $D^m F_{N,g}^{-1} \widehat{f} = F_{N,g}^{-1} M^m \widehat{f}$
3. Let S_a be given by $S_a f(t) = f(t - a)$ and $E_a f(t) = \exp(a * t) f(t)$. Then $S_a F_{N,g}^{-1} \widehat{f} = F_{N,g}^{-1} E_{-a} \widehat{f}$.

Proof. This can be verified by direct computation. □

11.6 Numerical experiments for the inverse Fourier transform

In the numerical experiments we take $n = 2$ for the Gaussian quadrature rule. For the weights we take $w_{h_j} = \frac{1}{5^{h_j}}$. We use the fast component by component of Nuyens to compute the Lattice rules. This yields the following Lattice rules:

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
1	37	43	8	25	28	5	39	45	21
1	260	461	163	234	56	197	48	174	133
1	1752	316	3297	4367	2711	2412	2601	1330	2889
1	44785	44069	4476	15331	22437	15903	16561	3322	12402
1	336303	493004	448989	274878	195816	374701	476110	322056	54794
1	1851280	3361998	2298758	3720761	1152663	2417695	3287941	488233	1586589
1	16587296	5938319	47026596	33214608	37713680	4976566	8807103	20269879	8546740

for $M = (97, 997, 9973, 99991, 999983, 9999991, 99999989)$.

As a first example we invert the Fourier transform of the standard multi variate density. The Fourier transform is given by

$$\widehat{f}(s) = \prod_{j=1}^d e^{\frac{1}{2}s_j^2}$$

We have normalized the results with factor $(2\pi)^{\frac{d}{2}}$, and computed $f(\mathbf{t})$ for $\mathbf{t} = \mathbf{0}$. The computation is done for $d = 1, \dots, 10$, while using the lattice points of table 1. The results are presented in the tables below: for all dimensions the Lattice transform algorithm converges extremely fast.

$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
1.0000000000000000	1.000000039059137	1.000009707784457	1.000073302625905	1.000392759396817
1.0000000000000000	0.999999999891985	0.999999999371404	1.000000006469789	1.000001387813120
1.0000000000000000	1.0000000000000000	1.0000000000000085	0.99999999997800	1.000000001540061
1.0000000000000000	0.999999999999990	0.999999999999990	1.000000000000823	1.000000000000455
1.0000000000000000	0.999999999999999	0.999999999999999	0.999999999999996	0.999999999999995
0.999999999999999	1.000000000000002	1.000000000000002	0.999999999999999	1.000000000000002
$d = 6$	$d = 7$	$d = 8$	$d = 9$	$d = 10$
1.001550145735646	1.003497006483246	1.007181337321017	1.014960448056923	1.025877016982776
0.999998574015691	1.000004871064538	1.000103695485601	1.000229366870632	1.000712722324263
0.999999905726270	1.000000326198758	1.000002530350383	1.000005394756590	1.000012888730967
0.99999999950871	0.99999999704772	0.999999988489276	1.000000009063984	0.999999933953289
0.9999999999951	1.000000000000412	1.000000000030561	1.000000000188339	0.999999996576185
1.000000000000002	1.000000000000000	0.99999999999838	1.000000000000295	1.000000000000262

We can also use the lattice transform inversion algorithm to compute high dimensional integrals. As a second experiment we price a d -dimension Basket put option. The payoff is given by

$$\left(K - \frac{1}{d+1} \sum_{k=0}^d S_k(T) \right)^+$$

Here S_k follows a geometric Brownian motion, that is

$$S_k(T) = S_0 \exp \left(\left(r - q_k - \frac{1}{2} \sigma_k^2 \right) T + \sigma_k W_k(T) \right)$$

We take for the interest rate $r = 0.03$, the volatility $\sigma_k = 0.2$, the dividend rate $q_k = 0.01$ and the expiration time $T = 5$. The price of the Basket put option can be written as

$$\begin{aligned} \mathbb{E} \left[e^{-rT} \left(K - \frac{1}{d} \sum_{k=1}^d S_k(T) \right)^+ \middle| F_0 \right] &= \mathbb{E} (h(Z_2, \dots, Z_d)) \\ &= \int_{\mathbb{R}^d} h(z_2, \dots, z_d) f(z_2, \dots, z_d) dz_2 \dots dz_d \end{aligned}$$

where

$$h(Z_2, \dots, Z_d) = \mathbb{E}_{Z_1} e^{-rT} \left(K - \sum_{k=1}^d \frac{e^{(r-q_k-\frac{1}{2}\sigma_k^2)T}}{d} \exp(\sigma_k \sqrt{T} Z_k) \right)^+$$

and f is the density of (Z_2, \dots, Z_d) . The function \hat{V} can easily be computed with the Black-Scholes formula. The function

$$\hat{V}(z_2, \dots, z_d) := h(z_2, \dots, z_d) f(z_2, \dots, z_d)$$

11.6 Numerical experiments for the inverse Fourier transform

is bounded by $e^{-rT} K f(z_2, \dots, z_d)$, hence \widehat{V} is the transform of a smooth function, say V . The price of the Basket option is given by

$$\int_{\mathbb{R}^d} \widehat{V}(z_2, \dots, z_d) dz_2 \dots dz_d = (2\pi)^{d-1} V(0)$$

We did the computation for $d = 1, \dots, 10$, and use the lattice points of table 1. The result are given in the tables below:

$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
0.073710017337385	0.034957207922192	0.020980266606449	0.014220539215257	0.010622515890196
0.073710017375368	0.034957628265605	0.020980648655084	0.014381869605335	0.010731895291859
0.073710017375368	0.034957635407280	0.020980541519274	0.014381129049913	0.010732209097494
0.073710017375368	0.034957635407379	0.020980541158816	0.014381156495638	0.010732228979009
0.073710017375368	0.034957635407373	0.020980541884721	0.014381156110129	0.010732248390596
0.073710017375368	0.034957635407373	0.020980541884722	0.014381156378695	0.010732248402242
$d = 6$	$d = 7$	$d = 8$	$d = 9$	$d = 10$
0.008233183239675	0.006850745789601	0.005839839672656	0.005089309012977	0.004376361169893
0.008453622261798	0.006995035016534	0.006030977702936	0.005253261049618	0.004667699810195
0.008493262471232	0.007014055051896	0.005979576497185	0.005223361487011	0.004648768589918
0.008493339379634	0.007013901085378	0.005979806873743	0.005223846002662	0.004649646789869
0.008493365415407	0.007013901865435	0.005979825692899	0.005223817677378	0.004649569299573
0.008493365245934	0.007013903186801	0.005979825552844	0.005223817595580	0.004649576121149

Bibliography

- J. Abate and W. Whitt. Numerical inversion of probability generating functions. *Operations Research Letters*, 12:245–251, 1992a.
- J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10:5–88, 1992b.
- J. Abate and W. Whitt. Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing*, 7:36–43, 1995.
- J. Abate, G.L. Choudhury, and W. Whitt. On the Laguerre method for numerically inverting Laplace transforms. *INFORMS Journal on Computing*, 8:413–427, 1996.
- J. Abate, G.L. Choudhury, and W. Whitt. Numerical inversion of multidimensional Laplace transforms by the Laguerre method. *Performance Evaluation*, 31:216–243, 1998.
- N. Asghari, P. den Iseger, and Mandjes M. Numerical techniques in lévy fluctuation theory. *Methodology and Computing in Applied Probability*, 16:31–52, 2014.
- S. Asmussen. *Applied Probability and Queues*. John Wiley & Sons, New York, 1987.
- D.H. Bailey and P.N Swartztrauber. The fractional Fourier transform and applications. *Siam Review*, 33 no. 3:389–404, 1991.
- O. Barndorff-Nielsen. Processes of normal inverse Gaussian type. *Finance and Stochastics*, 2:41–68, 1998.
- E Bázsza and P. den Iseger. Wide sense one-dependent processes with embedded harris chains and their applications in inventory management. *Econometric Institute*, 2002-44, 2002a.
- E Bázsza and P. den Iseger. A two level decentralized distribution system with compound renewal demand. *Econometric Institute*, EI 2002-45, 2002b.
- E Bázsza and P. den Iseger. Single item inventory models - a time- and event- averages approach. *Econometric Institute*, EI 2002-46, 2002c.

- E Bázsza and P. den Iseger. Optimal continuous order quantity (s, s) policies - the 45-degrees algorithm. *Econometric Institute*, EI 2002-47, 2002d.
- E Bázsza, H. Frenk, and P. den Iseger. Inventory control and regenerative processes: theory. *Econometric Institute*, 9931/A, 1999a.
- E Bázsza, H. Frenk, and P. den Iseger. Inventory control and regenerative processes: Computations. *Econometric Institute*, 9932/A, 1999b.
- E Bázsza, H. Frenk, and P. den Iseger. Inventory modeling of inventory control with regenerative processes. *International Journal of Production Economics*, 71:263–276, 2001.
- G. Beylkin, C. Kurcz, and L Monzón. Fast algorithms for helmholtz greens functions. *Proc. R. Soc. A*, pages 3301–3326, 2008.
- P. Billingsley. *Probability and measure*. John Wiley & Sons, New York, 3rd edition, 1995.
- M.J. Brennan and E.S. Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis*, 13(3):461–474, 1978.
- M. Broadie and P. Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42(2):269–285, 1996.
- M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *PaineWebber Working Papers in Money, Economics and Finance*, (# PW9804), 1997. Columbia Business School, New York.
- M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7(4):35–72, 2004.
- M. Broadie and Y. Yamamoto. A double-exponential fast Gauss transform algorithm for pricing discrete path-dependent options. *Operations Research*, 2005. to appear.
- P. Carr. Deriving derivatives of derivative securities. *Journal of Computational Finance*, 4(2):5–29, 2000.
- P. Carr and D.B. Madan. Option pricing and the Fast Fourier Transform. *Journal of Computational Finance*, 2(4):61–73, 1999.
- P. Carr and M. Schröder. On the valuation of arithmetic-average Asian options: The Geman-Yor Laplace transform revisited. February 2001. <http://arxiv.org/abs/math.CA/0102080> [11 April 2006].

BIBLIOGRAPHY

- P. Carr, H. Geman, D.B. Madan, and M. Yor. The fine structure of asset returns: An empirical investigation. *Journal of Business*, 75(2):305–332, 2002.
- P. Carr, H. Geman, D. Madan, and M. Yor. Stochastic volatility for Lévy processes. *Mathematical Finance*, 13(3):345–382, July 2003.
- G.L. Choudhury and W. Whitt. Probabilistic scaling for the numerical inversion of non-probability transforms. *INFORMS J. Computing*, 9:175–184, 1997.
- J.B. Conway. *A course in Functional analysis*. Springer Verlag, 1990.
- J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Math. Comput.*, 19:297–301, 1965.
- J.C. Cox, S.A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–264, 1979.
- M. Craddock, D. Heath, and E. Platen. Numerical inversion of Laplace transforms: A survey of techniques with applications to derivatives pricing. *Computational Finance*, 4(1):57–81, 2000.
- M. Craddock, D. Heath, and E. Platen. Numerical inversion of Laplace transforms: A survey of techniques with applications to derivative pricing. *Journal of Computational Finance*, 4(1):57–81, 2001.
- B. Davies and B.L. Martin. Numerical inversion of the Laplace transform: A survey and comparison of methods. *J. Comp. Phys.*, 33:1–32, 1979.
- S. de Groot, P. Den Iseger, and van der Weide H. A fast lévy ladder algorithm for pricing derivatives under lévy processes. *Working Paper*, 2006.
- P. Den Iseger. Numerical transform inversion using Gaussian quadrature. *Probability in the Engineering and Informational Sciences*, 20(1):1–44, 2006a.
- P. Den Iseger. Numerical Laplace transform inversion on the entire line. *Working Paper*, download from SSRN, 2006b.
- P. Den Iseger and E. Oldenkamp. Pricing guaranteed return rate products and discretely sampled Asian options. *Journal of Computational Finance*, 9(3), 2006a.
- P. Den Iseger and E. Oldenkamp. Computing Greeks: A drift-adjustment technique for European and Asian style derivatives. *Working Paper*, download from SSRN, 2006b.
- P. Den Iseger and E. Oldenkamp. Cliquet options: Pricing and Greeks in deterministic and stochastic volatility models. *working paper*, 2006c. download from SSRN.

- P. Den Iseger and E. Oldenkamp. High Dimensional Transformation Algorithms. *Working Paper, download from SSRN*, 2008.
- A.A. Drăgulescu and V.M. Yakovenko. Probability distribution of returns in the Heston model with stochastic volatility. *arXiv*, (cond-mat/0203046), November 2002. <http://arxiv.org/abs/cond-mat/0203046>.
- H. Dubner and J. Abate. Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *J. ACM*, 15:115–123, 1968.
- D. Duffy. *Transform Methods for Solving Partial Differential Equations*. Chapman & Hall/CRC, 2004.
- D. Dufresne. Laguerre series for Asian and other options. *Mathematical Finance*, 10:407–428, 2000.
- E. Eberlein and U. Keller. Hyperbolic distributions in finance. *Bernoulli*, 1(1994):281–299, 1995.
- E. Eberlein and W. Kluge. Exact pricing formulae for caps and swaptions in a Lévy term structure model. *Journal of Computational Finance*, 9(2):99–125, 2006.
- J. Zheng et al. Efficient parameter computation of 2d multiconductor interconnection lines in layered media by convergence acceleration of dielectric green’s function via pade approximation. *IEEE Trans. on Microwave Theory and Techn.*, 46:1339–1343, 1998.
- A. Eydeland and H. Geman. Domino effect. *RISK*, 8(4):65–67, 1995.
- F. Fang and C.W. Oosterlee. A novel pricing method for european options based on fourier-cosine series expansions. *SIAM Sci. Comput.*, 31:826–848, 2008.
- N. Marcuvitz Felsen, L. *Radiation and Scattering of Waves*. Wiley–IEEE Press, 1994.
- M.C. Fu, D.B. Madan, and T. Wang. Pricing continuous Asian options: a comparison of Monte Carlo and Laplace inversion methods. *Journal of Computational Finance*, 2:49–74, 1998.
- Sz. Gaál. Surface integral method to determine guided modes in uniaxially anisotropic embedded waveguides. *J. of Opt. and Quantum Electron.*, 31:763–7804, 1999.
- Sz. B. Gaál and P. den Iseger. A fast laplace inversion method for the solution of linear partial differential equations. *working paper*, 2009.
- D.P. Gaver. Observing stochastic processeand approximate transform inversion. *Operation Research*, 14:444–459, 1966.

BIBLIOGRAPHY

- H. Geman. Pure jump Lévy processes for asset price modelling. *Journal of Banking and Finance*, 26(7):1297–1316, 2002.
- H. Geman and M. Yor. Bessel processes, Asian options and perpetuities. *Mathematical Finance*, 3:349–375, 1993a.
- H. Geman and M. Yor. Bessel processes, Asian options and perpetuities. *Mathematical Finance*, 3:349–375, 1993b.
- P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2003. ISBN 0-387-00451-3.
- P. Glasserman and N. Merener. Cap and swaption approximations in Libor market models with jumps. *Computational Finance*, 7(1):1–36, Fall 2003.
- P. Glasserman and X. Zhao. Fast greeks by simulation in forward libor models. *Computational Finance*, 3(1):5–39, Fall 1999.
- P. Gruntjes, P. den Iseger, and Mandjes M. Numerical techniques in lévy fluctuation theory. *Methodology and Computing in Applied Probability*, 16:31–52, 2013.
- S.L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993.
- A. Hirska and D. Madan. Pricing American options under variance gamma. *Journal of Computational Finance*, 7(2), 2003.
- J. Hull and A. White. Valuing derivative securities using the explicit finite difference method. *Journal of Financial and Quantitative Analysis*, 25(1):87–100, 1990.
- J. Jacod and A.N. Shiryaev. *Limit Theorems for Stochastic Processes*. Springer-Verlag, Berlin, 1987.
- C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, 2009.
- R. E. Jorgenson and R. Mittra. Efficient calculation of the free-space periodic greens function. *IEEE Trans. Antennas Propag.*, 38:633–642, 1990.
- N.M. Korobov. Properties and calculation of optimal coefficients. *Dokl. Akad. Nauk SSSR*, 132:1009–1012, 1960.
- F.Y. Kuo. Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted korobov and sobolev spaces. *J. Complexity*, 19:301–320, 2003.

- D. Lamberton and B. Lapeyre. *Introduction to stochastic calculus applied to finance*. Chapman & Hall/CRC, 1996.
- R.W. Lee. Option pricing by transform methods: Extensions, unification, and error control. *Journal of Computational Finance*, 7(3):51–86, 2004.
- V. Linetsky. Spectral expansions for Asian (average price) options. *Operations Research*, 52(6):856–867, November–December 2004.
- C. M. Linton. Rapidly convergent representations for green’s functions for laplace’s equation. *Proc. R. Soc. Lond. A*, pages 1767–1797, 1999.
- A. Lipton. *Mathematical Methods For Foreign Exchange: A Financial Engineer’s Approach*. World Scientific Publishing Company, 2002.
- R. Lord and C. Kahl. Complex logarithms in heston-like models. *Mathematical Finance*, 20(4), 2010.
- R. Lord, F Fang, F Bervoets, and CW Oosterlee. *A Fast Method for Pricing Early-Exercise Options with the FFT*, pages 415–422. Computational Science - ICCS. Springer, Heidelberg, 2007.
- D. Madan, P. Carr, and E. Chang. The variance gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.
- S. Mallat. *A wavelet tour of signal processing*. Academic Press, London, second edition, 2001.
- S. L. Marshall. A rapidly convergent modified green’s function for laplace’s equation in a rectangular region.
- R. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–44, 1976.
- M.V. Moorthy. Numerical inversion of two-dimensional Laplace transforms - Fourier series representation. *Applied Numerical Mathematics*, 17:119–127, 1995.
- A. Murli and M. Rizzardi. Algorithm 682. Talbot’s method for the Laplace inversion problem. *ACM Trans. Math. Soft.*, 16:158–168, 1990.
- D. Nuyens and Cools R. Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel hilbert spaces. *Math. Comp.*, 75:903–920, 2006.

BIBLIOGRAPHY

- C.A. O’Cinneide. Euler summation for Fourier series and Laplace transform inversion. *Stochastic Models*, 13:315–337, 1997.
- A. Papoulis. *systems and Transforms with Application in Optics*. Kreiger, 1981.
- N. Patel. The evolving art of pricing cliquets. *Risk*, 15(7), 2002.
- R. Piessens. Gaussian quadrature formulas for the numerical inversion of Laplace transform. *J. of Engr. Math.*, 5:1–9, 1971a.
- R. Piessens. Some aspects of gaussian quadrature formulae for the numerical inversion of the Laplace transform. *The Computer Journal*, 14:433–436, 1971b.
- V.V. Piterbarg. Computing deltas of callable libor exotics in forward libor models. *Journal of Computational Finance*, 7(3):107–143, 2003.
- A. D. Polyanin. *Handbook of Linear Partial Differential Equations for Engineers and Scientists*. Chapman & Hall/CRC, 2002.
- P.E. Protter. *Stochastic Integration and Differential Equations*. Springer-Verlag, Berlin, 2nd edition, 2005.
- R. Richtmyer and K. Morton. *Difference Methods for Initial-Value Problems*. Krieger Pub Co, 1994.
- H. Rogier. Shanks-transform accelerated pml-based series expansions for the 1-d periodic 3-d greens functions of multilayered media. In *U.R.S.I, XXIX General Assembly, 7-16 August 2008, Chicago, USA*, 2008.
- W. Rudin. *Real and Complex Analysis*. Mathematics Series. McGraw-Hill, Singapore, third edition, 1987.
- W.J. Runggaldier. *Handbook of Heavy Tailed Distributions in Finance*, volume Book 1 of *Handbooks in Finance*, chapter Jump Diffusion Models, pages 169–209. Elsevier/North-Holland, Amsterdam, 2003.
- T. Sakurai. Numerical inversion of Laplace transform of functions with discontinuities. *Adv. in Appl. Probab.*, 20(2):616642, 2004.
- D.F. Schrager and A.A.J. Pelsser. Pricing rate of return guarantees in regular premium unit linked insurance. *Insurance: Mathematics and Economics*, 35(2):369–398, 2004.
- M. Schröder. Fact and fantasy in recent applications of transform methods to the valuation of exotic options. *Working paper*, September 1999. Universität Mannheim.

- R Seydel. *Tools for Computational Finance*. Springer-Verlag, Berlin, 2nd edition, 2004.
- W. Shaw. Pricing Asian options by contour integration, including asymptotic methods for low volatility. *Working paper*, 2002.
- S.E. Shreve. *Stochastic Calculus for Finance: Continuous-Time Models*, volume II. Springer - Verlag, New York, 8 edition, 2008. ISBN 978-0-387-40101-0.
- I.H. Sloan and Reztsov A.V. Component-by-component construction of good lattice rules. *Math. Comp.*, 71:263–273, 2002.
- H. Stehfest. Algorithm 368: numerical inversion of Laplace transforms. *Commun. ACM*, 13:47 – 49, 624, 1970.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Text in Applied Mathematics Vol. 12. Springer-Verlag, New York, 3rd edition, 2000.
- G. Szegő. *Orthogonal Polynomials*. American Math. Soc. Colloq. Pub. 23, fourth edition, 1975.
- A. Talbot. The accurate inversion of Laplace transforms. *J. Inst. Maths. Applics.*, 23: 97–120, 1979.
- H.C. Tijms. *A first course in stochastic models*. John Wiley & Sons, 2003. ISBN 0-471-49881-5.
- G. Valerio, P Baccarelli, P Burghignoli, and A. Galli. Comparative analysis of acceleration techniques for 2-d and 3-d greens functions in periodic structures along one and two directions. *IEEE Trans. Antennas Propag.*, 55:1630–1643, 2007.
- J. Vecer. A new PDE approach for pricing arithmetic average Asian options. *Journal of Computational Finance*, 4(4):105–113, 2001.
- J. Vecer. Unified Asian pricing. *RISK*, 15(6):113–116, June 2002.
- W.T. Weeks. Numerical inversion of Laplace transforms using Laguerre functions. *J. ACM*, 13:419–426, 1966.
- P. Wilmott. Cliquet options and volatility models. *Wilmott Magazine*, pages 78–83, December 2002.
- H.A. Windcliff, P.A. Forsyth, and K.R. Vetzal. Numerical methods for valuing cliquet options. Technical report, University of Waterloo, School of Computer Science, <http://www.scicom.uwaterloo.ca/paforsyt/cliquet.pdf>, August 2003.

Index

- Abate and Whitt [1992a], 255
Abate and Whitt [1992b], 255, 265
Abate and Whitt [1995], 255, 265
Abate et al. [1996], 255, 266
Abate et al. [1998], 255, 266
Asghari et al. [2014], 255
Asmussen [1987], 255
Bázsa and den Iseger [2002a], 255
Bázsa and den Iseger [2002b], 255
Bázsa and den Iseger [2002c], 255
Bázsa and den Iseger [2002d], 255
Bázsa et al. [1999a], 256
Bázsa et al. [1999b], 256
Bázsa et al. [2001], 256
Bailey and Swartztrauber [1991], 255
Barndorff-Nielsen [1998], 255
Beylkin et al. [2008], 256
Billingsley [1995], 256
Brennan and Schwartz [1978], 256
Broadie and Glasserman [1996], 256
Broadie and Glasserman [1997], 256
Broadie and Glasserman [2004], 256
Broadie and Yamamoto [2005], 256
Carr and Madan [1999], 256, 265
Carr and Schröder [2001], 256
Carr et al. [2002], 256
Carr et al. [2003], 257
Carr [2000], 256
Choudhury and Whitt [1997], 257, 265
Conway [1990], 257
Cooley and Tukey [1965], 257
Cox et al. [1979], 257
Craddock et al. [2000], 257
Craddock et al. [2001], 257
Davies and Martin [1979], 257
Den Iseger and Oldenkamp [2006a], 257
Den Iseger and Oldenkamp [2006b], 257
Den Iseger and Oldenkamp [2006c], 257
Den Iseger and Oldenkamp [2008], 257
Den Iseger [2006a], 257, 266
Den Iseger [2006b], 257
Drăgulescu and Yakovenko [2002], 258
Dubner and Abate [1968], 258, 265
Duffy [2004], 258
Dufresne [2000], 258
Eberlein and Keller [1995], 258
Eberlein and Kluge [2006], 258
Eydeland and Geman [1995], 258
Fang and Oosterlee [2008], 258
Felsen [1994], 258
Fu et al. [1998], 258
Gaál and den Iseger [2009], 258
Gaál [1999], 258
Gaver [1966], 258, 266
Geman and Yor [1993a], 259
Geman and Yor [1993b], 259
Geman [2002], 258
Glasserman and Merener [2003], 259
Glasserman and Zhao [1999], 259
Glasserman [2003], 259
Gruntjes et al. [2013], 259
Heston [1993], 259
Hirsa and Madan [2003], 259
Hull and White [1990], 259

- Jacod and Shiryaev [1987], 259
 Johnson [2009], 259
 Jorgenson and Mittra [1990], 259
 Korobov [1960], 259
 Kuo [2003], 259
 Lamberton and Lapeyre [1996], 259
 Lee [2004], 260
 Linetsky [2004], 260
 Linton [1999], 260
 Lipton [2002], 260
 Lord and Kahl [2010], 260
 Lord et al. [2007], 260
 Madan et al. [1998], 260
 Mallat [2001], 260
 Marshall [], 260
 Merton [1976], 260
 Moorthy [1995], 260
 Murli and Rizzardi [1990], 260, 266
 Nuyens and R. [2006], 260
 O’Cinneide [1997], 260, 265
 Papoulis [1981], 261
 Patel [2002], 261
 Piessens [1971a], 261
 Piessens [1971b], 261
 Piterbarg [2003], 261
 Polyanin [2002], 261
 Protter [2005], 261
 Richtmyer and Morton [1994], 261
 Rogier [2008], 261
 Rudin [1987], 261
 Runggaldier [2003], 261
 Sakurai [2004], 261, 266
 Schröder [1999], 261
 Schrage and Pelsser [2004], 261
 Seydel [2004], 261
 Shaw [2002], 262
 Shreve [2008], 262, 267
 Sloan and A.V. [2002], 262
 Stehfest [1970], 262, 266
 Stoer and Bulirsch [2000], 262
 Szegő [1975], 262
 Talbot [1979], 262, 266
 Tijms [2003], 262
 Valerio et al. [2007], 262
 Vecer [2001], 262
 Vecer [2002], 262
 Weeks [1966], 262, 266
 Wilmott [2002], 262
 Windcliff et al. [2003], 262
 de Groot et al. [2006], 257
 et al [1998], 258

Nederlandse Samenvatting

Operations research problemen worden vaak gemodelleerd door stochastische modellen, waar kansverdelingen een grote rol in spelen. Een efficiënt algoritme voor het doorrekenen van die modellen kan pas gemaakt worden als die kansverdelingen ook efficiënt berekend kunnen worden. Voor dit doeleinde zijn verschillende types numeriek inversie algoritmen voor Fourier (Laplace) transformaties ontworpen; kansverdelingen kunnen namelijk gekarakteriseerd worden door hun Fourier (Laplace) transformaties.

Fourier transformaties hebben net als de exponentiele functie, een aantal zeer waardevolle eigenschappen. Het is vanwege deze eigenschappen dat er veel doelfuncties in termen van hun Fourier transformaties uitgedrukt kunnen worden. De uitdaging is dan om deze Fourier transformatie te inverteren om de grootheid van belang te kunnen verkrijgen.

Dit proefschrift presenteert een geavanceerde techniek voor het inverteren van Fourier transformaties, samen met vernieuwende toepassingen voornamelijk in kwantitatieve financiering. Deze techniek is ook met succes toegepast voor problemen in de logistiek, wachtrijtheorie en kansrekening; echter de focus ligt hier vanwege omvang overwegingen op toepassingen in de financiering. Fourier inversie algoritmen waren voor het eerst toegepast voor problemen in kwantitatieve financiering door Carr en Madan Carr and Madan [1999]. Hun methode zette het bekende FFT (Fast Fourier Transform) algoritme in om de rekentijd aanzienlijk in te korten; zij gebruiken de trapezode regel voor het discreet maken van de Fourier inversie integraal. Het discreet maken van de Fourier inversie integraal staat bekend in de vakliteratuur als de Fourier reeks methode. De wiskundige rechtvaardiging waarop die methodes gebaseerd zijn, is de zogenaamde Poisson Sommatie Formule (PSF). De PSF relateert de oneindige som van Laplace transformatie waarden aan de Fourier serie van functie waarden $f(k\Delta)$, $k = 0, 1, \dots$. Het probleem is dat die oneindige som van Laplace transformatie waarden heel langzaam convergeert. Abate and Whitt Abate and Whitt [1992b] zetten de zogenaamde Euler sommatie techniek in om de convergentie te versnellen. De meest succesvolle Fourier series methodes zijn Dubner and Abate [1968], Abate and Whitt [1992b], Abate and Whitt [1995], Choudhury and Whitt [1997], O'Connell [1997] en

Sakurai [2004].

Hoofdstuk 3 introduceert een nieuw Gaussian kwadratuur methode Den Iseger [2006a], die vervangt die langzaam convergerende oneindige serie met een eindige som, resulterend in een efficiënt inversie algoritme. Dit algoritme berekent alle benodigde functie waarden tegelijk, binnen $O(N \log(N))$ tijd (de functie waarden in de punten $k\Delta$, $k = -N, -(N-1), \dots, N$, waar N en Δ vrij gekozen parameters zijn). Het algoritme is niet alleen heel snel, het is ook heel nauwkeurig. De meest gebruikte transformaties kunnen met dit algoritme geïnverteerd worden met een resultaat dat net zo nauwkeurig is als de computer precisie.

Een tweede type numerieke inversie methode is het zogenaamde Gaver-Stehfest algoritme (zie Gaver [1966] en Stehfest [1970]). Een derde type is de Weeks methode, die gebaseerd is op Laguerre expansies Weeks [1966] en Abate, Choudhury en Whitt Abate et al. [1996], Abate et al. [1998]. Ten vierde noemen we een inversie techniek genaamd de Talbot methode, Talbot [1979] en Murli and Rizzardi [1990]). Hoewel deze methoden bij de meeste toepassingen uitstekende resultaten geven, zijn ze allemaal erg gevoelig voor singulariteiten en discontinuïteiten. n enkel discontinuïteit kan de convergentie van de bovengenoemde methodes helemaal doen mislukken. De inversie methodes van dit proefschrift zijn noch gevoelig voor singulariteiten noch voor discontinuïteiten. Een andere gemeenschappelijke eigenschap van de bekende inversiemethodes is dat de inversie puntgewijs wordt gedaan, dat wil zeggen in een enkel punt. In de praktijk heeft men veel functiewaarden nodig, of zelfs functiewaarden op een heel interval. Bijvoorbeeld bij het calibreren van modellen op marktprijzen in de kwantitatieve financiering. Om zo'n model te berekenen die gebaseerd is op een inversie algoritme die de functiewaarden punt voor punt inverteert, kan erg lang duren. Dit proefschrift presenteert een methode, gebaseerd op een Gaussian kwadratuur formule, die een uniform grid tegelijk inverteert. Het resultaat is zeer nauwkeurig, als van computer precisie. Hoewel deze methode heel nauwkeurig en snel is, de beperking van een uniform grid in plaats van een willekeurig grid (punten willekeurig kunnen kiezen in plaats van op een vaste afstand) kan nadelig zijn bij sommige toepassingen. Zo een toepassing is bijvoorbeeld iteratieve, dat wil zeggen recursieve algoritmen (voorkomend bij dynamisch programmeren bijvoorbeeld) of partiele differentiaalvergelijkingen.

Deze toepassingen vragen om een inversiealgoritme dat in staat is een Fourier transformatie te inverteren op een heel interval. De tweede inversiealgoritme van dit proefschrift is een inversiealgoritme, dat ook genaamd is "entire line algorithm", dat wil zeggen, een methode die de inversie tegelijk op een heel interval realiseert. Deze methode berekent de coëfficiënten van de Legendre expansie voor de functie. Een bijkomende pr van deze methode is dat die zowel de oorspronkelijke functie kan berekenen als er numerieke waarden van de Fourier (Laplace) transformatie gegeven zijn, als de

Laplace transformatie kan berekenen als er numerieke waarden van de oorspronkelijke functie beschikbaar zijn. Zeer belangrijk om hier te op te merken is dat er geen analytische vorm van de functie noch van de Laplace transformatie nodig is, alleen numerieke waarden.

Deze technieken zijn vervolgens toegepast om bekende modellen te berekenen: Aziatische optie, gegarandeerd rendementsoptie en de Grieken, de gevoeligheid van opties naar veranderingen in de markt (recursief!). Er worden ook zgn. ladder algoritmen besproken en berekend; algoritmen waarmee problemen uit dynamisch programmeren opgelost kunnen worden. Een toepassing hiervan is het prijzen van Bermuda opties. Met deze methode kunnen ook Lévy processen behandeld worden. Er worden in hoofdstuk 8 cliquet-opties geprijsd en hun Grieken berekend, een ware uitdaging in de vakliteratuur, hier nauwkeurig berekend met de nieuwe methoden. De berekeningen zijn zowel voor constante volatiliteit als voor stochastisch (Hestonian) volatiliteit en voor jump processen uitgevoerd. Een natuurlijke stap in het uitbreiden van de inversiealgoritmen is het ontwerpen van een inversie algoritme voor hogere dimensionale transformaties. Zon Multi-dimensionaal inversiealgoritme kan bijvoorbeeld ingezet worden in kwantitatieve financiering voor het prijzen van derivaten die afhankelijk zijn van meerdere onderliggende factoren. De vakliteratuur op het gebied van hogere dimensionale inversie algoritmen is gering. In plaats van een dergelijk inversiealgoritme wordt meestal Monte Carlo simulatie gebruikt: langzaam en niet heel nauwkeurig. De ladder en fast convolution inversiealgoritmen zijn ook uit te breiden tot hogere dimensies. Deze uitbreiding wordt beschreven in hoofdstuk 9.

In hoofdstuk 10 worden de hogere dimensionale inversiealgoritmen toegepast voor het oplossen van partiele differentiaalvergelijkingen en de Heat vergelijking. Met de bekende Feynman-Kac formule Shreve [2008], kan het prijzen van vele financie instrumenten geformuleerd worden als partiele differentiaalvergelijkingen. We passen die technieken toe om de prijs van barrier opties te berekenen: het resultaat is een veel snellere, nauwkeurigere en stabielere berekening. Het laatste hoofdstuk combineert de nieuwe inversiealgoritmen met de zogenaamde Lattice integratie methoden. Lattice integratie methoden zijn quasi-Monte Carlo methoden die heel efficiënt hoge dimensionale periodieke functies kunnen integreren. Het resulterende zeer nauwkeurige algoritme kan in een paar seconden (tot) tien dimensionale Fourier transformaties inverteren. Een mooie toepassing hiervan is het prijzen van de zogenaamde Basket opties, een optie op een "mandje" van verschillende aandelen. Zulke opties zijn belangrijk in het bijzonder voor pensioenfondsen. De enige andere methode bekend van de vakliteratuur voor het prijzen van Basket opties is Monte Carlo simulatie, en die is heel tijdrovend.

Laat ons deze samenvatting met het antwoord op de volgende vraag conclude-

ren. Dit proefschrift presenteert twee verschillende basis- inversiealgoritmen, dat wil zeggen, eendimensionale inversie algoritmen. Is er wellicht n van deze algoritmen voldoende? Het is inderdaad waar dat de twee methoden veel gemeen hebben met elkaar. We zouden zelfs kunnen stellen dat het enige verschil tussen de twee methoden is dat de ene de transformatie op een grid inverteert terwijl de ander het op een heel interval inverteert. Het inverteren op een heel interval tegelijkertijd lijkt dus superieur aan de inversie op een grid (in enkele punten), dus het algoritme uit hoofdstuk 4 lijkt voldoende. De methode uit hoofdstuk 3 geeft wat nauwkeuriger berekeningen en die is ook wat sneller. Deze is ook makkelijker aan te passen voor singulariteiten. Met name bij singulariteiten rond 0, zoals voor de functie \sqrt{x} . Bovendien kan de methode uitgebreid worden tot ongeveer 10 dimensies (zodat het nog nauwkeurig en snel blijft) terwijl de methode 4 tot ongeveer vier dimensies uitgebreid worden.

About the author

Peter den Iseger holds a masters degrees in Econometrics from Erasmus University Rotterdam. Peters main research interest are applied probability and numerical mathematics, with in particular applications in quantitative finance. He published a number of publications about the numerical inversion of Fourier and Laplace transforms and application of these algorithms in quantitative finance. He is a specialist in designing computational methods for price and risk calculations for derivative portfolios in various asset classes. He works as a quant and quantitative researcher for many years in the financial industry.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the [Erasmus Research Institute of Management \(ERIM\)](http://hdl.handle.net/1765/1). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: <http://hdl.handle.net/1765/1> ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

DISSERTATIONS LAST FIVE YEARS

Acciario, M., *Bundling Strategies in Global Supply Chains*, Promotor(s): Prof.dr. H.E. Haralambides, EPS-2010-197-LIS, <http://hdl.handle.net/1765/19742>

Akpinar, E., *Consumer Information Sharing; Understanding Psychological Drivers of Social Transmission*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2013-297-MKT, <http://hdl.handle.net/1765/50140>

Alexiev, A., *Exploratory Innovation: The Role of Organizational and Top Management Team Social Capital*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-208-STR, <http://hdl.handle.net/1765/20632>

Akin Ates, M., *Purchasing and Supply Management at the Purchase Category Level: Strategy, Structure, and Performance*, Promotor: Prof.dr. J.Y.F. Wynstra, EPS-2014-300-LIS, <http://hdl.handle.net/1765/1>

Almeida, R.J.de, *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*, Promotor Prof.dr.ir. Uzay Kaymak, EPS-2014-310-LIS, <http://hdl.net/1765/1>

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-Frequency Data*, Promotor: Prof.dr.D.J.C. van Dijk, EPS-2013-273-F&A, <http://hdl.handle.net/1765/38240>

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2011-241-MKT, <http://hdl.handle.net/1765/23670>

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2013-278-S&E, <http://hdl.handle.net/1765/39128>

Berg, W.E. van den, *Understanding Salesforce Behavior Using Genetic Association Studies*, Promotor: Prof.dr. W.J.M.I. Verbeke, EPS-2014-311-MKT, <http://hdl.handle.net/1765/51440>

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promotor: Prof.dr. B. Krug, EPS-2012-262-ORG, <http://hdl.handle.net/1765/32345>

Binken, J.L.G., *System Markets: Indirect Network Effects in Action, or Inaction*, Promotor: Prof.dr. S. Stremersch, EPS-2010-213-MKT, <http://hdl.handle.net/1765/21186>

Blitz, D.C., *Benchmarking Benchmarks*, Promotor(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, <http://hdl.handle.net/1765/226244>

Boons, M., *Working Together Alone in the Online Crowd: The Effects of Social Motivations and Individual Knowledge Backgrounds on the Participation and Performance of Members of Online Crowdsourcing Platforms*, Promotor: Prof.dr. H.G. Barkema, EPS-2014-306-S&E, <http://hdl.net/1765/50711>

Borst, W.A.M., *Understanding Crowdsourcing: Effects of Motivation and Rewards on Participation and Performance in Voluntary Online Activities*, Promotor(s): Prof.dr.ir. J.C.M. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2010-221-LIS, <http://hdl.handle.net/1765/21914>

Budiono, D.P., *The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2010-185-F&A, <http://hdl.handle.net/1765/18126>

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.R. Commandeur, EPS-2011-243-ORG, <http://hdl.handle.net/1765/26178>

Byington, E., *Exploring Coworker Relationships: Antecedents and Dimensions of Interpersonal Fit, Coworker Satisfaction, and Relational Models*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2013-292-ORG, <http://hdl.handle.net/1765/41508>

Camacho, N.M., *Health and Marketing: Essays on Physician and Patient Decision-making*, Promotor: Prof.dr. S. Stremersch, EPS-2011-237-MKT, <http://hdl.handle.net/1765/23604>

Cankurtaran, P. *Essays On Accelerated Product Development*, Promotor: Prof.dr.ir. G.H. van Bruggen, EPS-2014-317-MKT, <http://hdl.handle.net/1765/1>

Caron, E.A.M., *Explanation of Exceptional Values in Multi-dimensional Business Databases*, Promotor(s): Prof.dr.ir. H.A.M. Daniels & Prof.dr. G.W.J. Hendrikse, EPS-2013-296-LIS, <http://hdl.handle.net/1765/50005>

Carvalho, L., *Knowledge Locations in Cities; Emergence and Development Dynamics*, Promotor: Prof.dr. L. van den Berg, EPS-2013-274-S&E, <http://hdl.handle.net/1765/38449>

Carvalho de Mesquita Ferreira, L., *Attention Mosaics: Studies of Organizational Attention*, Promotor(s): Prof.dr. P.M.A.R. Heugens & Prof.dr. J. van Oosterhout, EPS-2010-205-ORG, <http://hdl.handle.net/1765/19882>

Cox, R.H.G.M., *To Own, To Finance, and to Insure; Residential Real Estate Revealed*, Promotor: Prof.dr. D. Brounen, EPS-2013-290-F&A, <http://hdl.handle.net/1765/40964>

Defilippi Angeldonis, E.F., *Access Regulation for Naturally Monopolistic Port Terminals: Lessons from Regulated Network Industries*, Promotor: Prof.dr. H.E. Haralambides, EPS-2010-204-LIS, <http://hdl.handle.net/1765/19881>

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2012-255-ORG, <http://hdl.handle.net/1765/31174>

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promotor: Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2011-232-ORG, <http://hdl.handle.net/1765/23268>

Dietvorst, R.C., *Neural Mechanisms Underlying Social Intelligence and Their Relationship with the Performance of Sales Managers*, Promotor: Prof.dr. W.J.M.I. Verbeke, EPS-2010-215-MKT, <http://hdl.handle.net/1765/21188>

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promotor:
Prof.dr. A.P.M. Wagelmans, EPS-2013-272-LIS, <http://hdl.handle.net/1765/38241>

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promotor(s): Prof.dr. J.J.P. Jansen,
Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-258-STR,
<http://hdl.handle.net/1765/32166>

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory*, Promotor:
Prof.dr. W.F.C. Verschoor, EPS-2012-257-F&A, <http://hdl.handle.net/1765/31914>

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promotor: Prof.dr. A. de Jong,
EPS-2011-240-F&A, <http://hdl.handle.net/1765/26041>

Duursema, H., *Strategic Leadership; Moving Beyond the Leader-follower Dyad*, Promotor:
Prof.dr. R.J.M. van Tulder, EPS-2013-279-ORG, <http://hdl.handle.net/1765/39129>

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promotor:
Prof.dr.ir. R. Dekker, EPS-2011-247-LIS, <http://hdl.handle.net/1765/26509>

Essen, M. van, *An Institution-Based View of Ownership*, Promotor(s): Prof.dr. J. van Oosterhout &
Prof.dr. G.M.H. Mertens, EPS-2011-226-ORG, <http://hdl.handle.net/1765/22643>

Feng, L., *Motivation, Coordination and Cognition in Cooperatives*, Promotor:
Prof.dr. G.W.J. Hendrikse, EPS-2010-220-ORG, <http://hdl.handle.net/1765/21680>

Fourné, S. P. L. *Managing Organizational Tensions: A Multi-level Perspective on Exploration, Exploitation, and Ambidexterity*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr. S.J. Magala & dr. T.J.M.Mom, EPS-2014-318-S&E, <http://hdl.handle.net/1765/21680>

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promotor:
Prof.dr.ir. M.B.M. de Koster, EPS-2012-269-LIS, <http://hdl.handle.net/1765/37779>

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2012-270-ORG, <http://hdl.handle.net/1765/38028>

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2012-256-MKT, <http://hdl.handle.net/1765/31913>

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promotor(s): Prof.dr. G.M.H. Mertens &
Prof.dr. E. Peek, EPS-2012-264-F&A, <http://hdl.handle.net/1765/37170>

Hakimi, N.A., *Leader Empowering Behaviour: The Leader's Perspective: Understanding the Motivation behind Leader Empowering Behaviour*, Promotor: Prof.dr. D.L. van Knippenberg,
EPS-2010-184-ORG, <http://hdl.handle.net/1765/17701>

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promotor(s): Prof.dr. A. Jolink & Prof.dr. S.J. Magala, EPS-2010-193-ORG, <http://hdl.handle.net/1765/19494>

- Hernandez Mireles, C., *Marketing Modeling for New Products*, Promotor: Prof.dr. P.H. Franses, EPS-2010-202-MKT, <http://hdl.handle.net/1765/19878>
- Heyde Fernandes, D. von der, *The Functions and Dysfunctions of Reminders*, Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2013-295-MKT, <http://hdl.handle.net/1765/41514>
- Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-259-STR, <http://hdl.handle.net/1765/32167>
- Hoever, I.J., *Diversity and Creativity: In Search of Synergy*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, <http://hdl.handle.net/1765/37392>
- Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet*, Promotor(s): Prof.dr. H.P.G. Pennings & Prof.dr. A.R. Thurik, EPS-2011-246-STR, <http://hdl.handle.net/1765/26447>
- Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promotor(s): Prof.dr. D. De Cremer & Dr. M. van Dijke, EPS-2011-244-ORG, <http://hdl.handle.net/1765/26228>
- Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesign*, Promotor(s): Prof.dr. M.J.C.M. Verbeek & Prof.dr. R.J. Mahieu, EPS-2010-196-F&A, <http://hdl.handle.net/1765/19674>
- Hytönen, K.A. *Context Effects in Valuation, Judgment and Choice*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2011-252-MKT, <http://hdl.handle.net/1765/30668>
- Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2013-288-LIS, <http://hdl.handle.net/1765/39933>
- Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics*, Promotor(s): Prof.dr. L.G. Kroon, EPS-2011-222-LIS, <http://hdl.handle.net/1765/22156>
- Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems*, Promotor(s): Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, <http://hdl.handle.net/1765/19532>
- Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promotor(s): Prof.dr. S. Stremersch, EPS-2011-239-MKT, <http://hdl.handle.net/1765/23610>
- Karreman, B., *Financial Services and Emerging Markets*, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.P.G. Pennings, EPS-2011-223-ORG, <http://hdl.handle.net/1765/22280>

- Kil, J.C.M., *Acquisitions Through a Behavioral and Real Options Lens*, Promotor(s): Prof.dr. H.T.J. Smit, EPS-2013-298-F&A, <http://hdl.handle.net/1765/50142>
- Klooster, E. van't, *Travel to Learn: The Influence of Cultural Distance on Competence Development in Educational Travel*, Promotors: Prof.dr. F.M. Go & Prof.dr. P.J. van Baalen, EPS-2014-312-MKT, <http://hdl.handle.net/1765/151460>
- Koendjibiharie, S.R., *The Information-Based View on Business Network Performance Revealing the Performance of Interorganizational Networks*, Promotors: Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2014-315-LIS, <http://hdl.handle.net/1765/1>
- Konter, D.J., *Crossing borders with HRM: An inquiry of the influence of contextual differences in the adaption and effectiveness of HRM*, Promotor: Prof.dr. J. Paauwe, EPS-2014-305-ORG, <http://hdl.handle.net/1765/1>
- Korkmaz, E., *Understanding Heterogeneity in Hidden Drivers of Customer Purchase Behavior*, Promotors: Prof.dr. S.L. van de Velde & dr. R.Kuik, EPS-2014-316-LIS, <http://hdl.handle.net/1765/1>
- Lam, K.Y., *Reliability and Rankings*, Promotor(s): Prof.dr. P.H.B.F. Franses, EPS-2011-230-MKT, <http://hdl.handle.net/1765/22977>
- Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, <http://hdl.handle.net/1765/30682>
- Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promotor(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, <http://hdl.handle.net/1765/23504>
- Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, Promotor(s): Prof.dr.ir. J. Dul & Prof.dr. M.B.M. de Koster, EPS-2010-217-LIS, <http://hdl.handle.net/1765/21527>
- Leunissen, J.M., *All Apologies: On the Willingness of Perpetrators to Apologize*, Promotor: Prof.dr. D. De Cremer, EPS-2014-301-ORG, <http://hdl.handle.net/1765/1>
- Liang, Q., *Governance, CEO Identity, and Quality Provision of Farmer Cooperatives*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2013-281-ORG, <http://hdl.handle.net/1765/1>
- Liket, K.C., *Why 'Doing Good' is not Good Enough: Essays on Social Impact Measurement*, Promotor: Prof.dr. H.R. Commandeur, EPS-2014-307-S&E, <http://hdl.handle.net/1765/51130>
- Loos, M.J.H.M. van der, *Molecular Genetics and Hormones; New Frontiers in Entrepreneurship Research*, Promotor(s): Prof.dr. A.R. Thurik, Prof.dr. P.J.F. Groenen & Prof.dr. A. Hofman, EPS-2013-287-S&E, <http://hdl.handle.net/1765/40081>
- Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promotor(s): Prof.dr. J. Spronk & Prof.dr.ir. U. Kaymak, EPS-2011-229-F&A, <http://hdl.handle.net/1765/22814>
- Lu, Y., *Data-Driven Decision Making in Auction Markets*, Promotors: Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. W.Ketter, EPS-2014-314-LIS, <http://hdl.handle.net/1765/1>
- Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promotor: Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, <http://hdl.handle.net/1765/22744>

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, <http://hdl.handle.net/1765/34930>

Meur, J., *Configurations of Inter-Firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promotor: Prof.dr. B. Krug, EPS-2011-228-ORG, <http://hdl.handle.net/1765/22745>

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, <http://hdl.handle.net/1765/32343>

Milea, V., *New Analytics for Financial Decision Support*, Promotor: Prof.dr.ir. U. Kaymak, EPS-2013-275-LIS, <http://hdl.handle.net/1765/38673>

Naumovska, I. *Socially Situated Financial Markets: a Neo-Behavioral Perspective on Firms, Investors and Practices*, Promoter(s) Prof.dr. P.P.M.A.R. Heugens & Prof.dr. A.de Jong, EPS-2014-319-S&E, <http://hdl.handle.net/1765/1>

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in Short-term Planning and in Disruption Management*, Promotor: Prof.dr. L.G. Kroon, EPS-2011-224-LIS, <http://hdl.handle.net/1765/22444>

Nijdam, M.H., *Leader Firms: The Value of Companies for the Competitiveness of the Rotterdam Seaport Cluster*, Promotor(s): Prof.dr. R.J.M. van Tulder, EPS-2010-216-ORG, <http://hdl.handle.net/1765/21405>

Noordegraaf-Eelens, L.H.J., *Contested Communication: A Critical Analysis of Central Bank Speech*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2010-209-MKT, <http://hdl.handle.net/1765/21061>

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promotor: Prof.dr. G. van der Pijl & Prof.dr. H. Commandeur & Prof.dr. M. Keil, EPS-2012-263-S&E, <http://hdl.handle.net/1765/34928>

Oosterhout, M., van, *Business Agility and Information Technology in Service Organizations*, Promotor: Prof.dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, <http://hdl.handle.net/1765/19805>

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on Bureaucracy and Formal Rules*, Promotor: Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, <http://hdl.handle.net/1765/23250>

Otgaar, A.H.J., *Industrial Tourism: Where the Public Meets the Private*, Promotor: Prof.dr. L. van den Berg, EPS-2010-219-ORG, <http://hdl.handle.net/1765/21585>

Ozdemir, M.N., *Project-level Governance, Monetary Incentives and Performance in Strategic R&D Alliances*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, <http://hdl.handle.net/1765/23550>

Peers, Y., *Econometric Advances in Diffusion Models*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2011-251-MKT, <http://hdl.handle.net/1765/30586>

Pince, C., *Advances in Inventory Management: Dynamic Models*, Promotor: Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, <http://hdl.handle.net/1765/19867>

Porck, J.P., *No Team is an Island*, Promotor: Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-299-ORG, <http://hdl.handle.net/1765/50141>

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, <http://hdl.handle.net/1765/30848>

Potthoff, D., *Railway Crew Rescheduling: Novel Approaches and Extensions*, Promotor(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. L.G. Kroon, EPS-2010-210-LIS, <http://hdl.handle.net/1765/21084>

Poruthiyil, P.V., *Steering Through: How Organizations Negotiate Permanent Uncertainty and Unresolvable Choices*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. S. Magala, EPS-2011-245-ORG, <http://hdl.handle.net/1765/26392>

Pourakbar, M., *End-of-Life Inventory Decisions of Service Parts*, Promotor: Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, <http://hdl.handle.net/1765/30584>

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promotor(s): Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2013-282-S&E, <http://hdl.handle.net/1765/39654>

Retel Helmrich, M.J., *Green Lot-Sizing*, Promotor: Prof.dr. A.P.M. Wagelmans, EPS-2013-291-LIS, <http://hdl.handle.net/1765/41330>

Rietveld, C.A., *Essays on the Intersection of Economics and Biology*, Promotor(s): Prof.dr. P.J.F. Groenen, Prof.dr. A. Hofman & Prof.dr. A.R. Thurik, EPS-2014-320-S&E, <http://hdl.handle.net/1765/1>

Rijssenbilt, J.A., *CEO Narcissism; Measurement and Impact*, Promotor: Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, <http://hdl.handle.net/1765/23554>

Roelofsens, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promotor(s): Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, <http://hdl.handle.net/1765/18013>

Roza, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of Innovation, Absorptive Capacity and Firm Size*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, <http://hdl.handle.net/1765/22155>

Rubbaniy, G., *Investment Behavior of Institutional Investors*, Promotor: Prof.dr. W.F.C. Verschoor, EPS-2013-284-F&A, <http://hdl.handle.net/1765/40068>

Schellekens, G.A.C., *Language Abstraction in Word of Mouth*, Promotor: Prof.dr.ir. A. Smidts, EPS-2010-218-MKT, <http://hdl.handle.net/1765/21580>

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promotor: Prof.dr. G.M.H. Mertens, EPS-2013-283-F&A, <http://hdl.handle.net/1765/39655>

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization*, Promotor(s): Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, <http://hdl.handle.net/1765/19714>

Sousa, M., *Servant Leadership to the Test: New Perspectives and Insight*, Promotors: Prof.dr. D. van Knippenberg & Dr. D. van Dierendonck, EPS-2014-313-ORG, <http://hdl.net/1765/1>

Spliet, R., *Vehicle Routing with Uncertain Demand*, Promotor: Prof.dr.ir. R. Dekker, EPS-2013-293-LIS, <http://hdl.handle.net/1765/41513>

Srouf, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promotor: Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, <http://hdl.handle.net/1765/18231>

Staat, J.L., *Leading Public Housing Organisation in a Problematic Situation: A Critical Soft Systems Methodology Approach*, Promotor: Prof.dr. S.J. Magala, EPS-2014-308-ORG, <http://hdl.handle.net/1765/50712>

Stallen, M., *Social Context Effects on Decision-Making; A Neurobiological Approach*, Promotor: Prof.dr.ir. A. Smidts, EPS-2013-285-MKT, <http://hdl.handle.net/1765/39931>

Tarakci, M., *Behavioral Strategy; Strategic Consensus, Power and Networks*, Promotor(s): Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-280-ORG, <http://hdl.handle.net/1765/39130>

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promotor: Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, <http://hdl.handle.net/1765/37265>

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes*, Promotor(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, <http://hdl.handle.net/1765/18457>

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments*, Promotor(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, <http://hdl.handle.net/1765/19868>

Tröster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, <http://hdl.handle.net/1765/23298>

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision Making*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, <http://hdl.handle.net/1765/37542>

Tzioti, S., *Let Me Give You a Piece of Advice: Empirical Papers about Advice Taking in Marketing*, Promotor(s): Prof.dr. S.M.J. van Osselaer & Prof.dr.ir. B. Wierenga, EPS-2010-211-MKT, hdl.handle.net/1765/21149

Vaccaro, I.G., *Management Innovation: Studies on the Role of Internal Change Agents*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-212-STR, hdl.handle.net/1765/21150

Vagias, D., *Liquidity, Investors and International Capital Markets*, Promotor: Prof.dr. M.A. van Dijk, EPS-2013-294-F&A, <http://hdl.handle.net/1765/41511>

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains*, Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, <http://hdl.handle.net/1765/19594>

Venus, M., *Demystifying Visionary Leadership; In Search of the Essence of Effective Vision Communication*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2013-289-ORG, <http://hdl.handle.net/1765/40079>

Visser, V., *Leader Affect and Leader Effectiveness; How Leader Affective Displays Influence Follower Outcomes*, Promotor: Prof.dr. D. van Knippenberg, EPS-2013-286-ORG, <http://hdl.handle.net/1765/40076>

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, <http://hdl.handle.net/1765/30585>

Ward, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, <http://hdl.handle.net/1765/18012>

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promotor(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, <http://hdl.handle.net/1765/26564>

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2011-242-F&A, <http://hdl.handle.net/1765/26066>

Wang, Y., *Corporate Reputation Management; Reaching Out to Find Stakeholders*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2013-271-ORG, <http://hdl.handle.net/1765/38675>

Weenen, T.C., *On the Origin and Development of the Medical Nutrition Industry*, Promotors: Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2014-309-S&E, <http://hdl.handle.net/1765/51134>

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value of Dutch Firms*, Promotor: Prof.dr. A. de Jong, EPS-2013-277-F&A, <http://hdl.handle.net/1765/39127>

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, <http://hdl.handle.net/1765/18125>

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2013-276-LIS, <http://hdl.handle.net/1765/38766>

Zhang, D., *Essays in Executive Compensation*, Promotor: Prof.dr. I. Dittmann, EPS-2012-261-F&A, <http://hdl.handle.net/1765/32344>

Zhang, X., *Scheduling with Time Lags*, Promotor: Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, <http://hdl.handle.net/1765/19928>

Zhou, H., *Knowledge, Entrepreneurship and Performance: Evidence from Country-level and Firm-level Studies*, Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. L.M. Uhlaner, EPS-2010-207-ORG, <http://hdl.handle.net/1765/20634>

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, <http://hdl.handle.net/1765/23422>

FOURIER AND LAPLACE TRANSFORM INVERSION WITH APPLICATIONS IN FINANCE

It is becoming standard for modeling and analysis to include algorithms for computing probability distributions of interest. Therefore, advances in computational probability increase the value of stochastic models in among others quantitative finance, queueing, reliability, and inventory management. Several tools have been developed for this purpose, of which one of the most powerful tools in numerical Fourier transform inversion. The reason is that probability distributions can be characterized in terms of Laplace transforms, or more generally, Fourier transforms. Many results in among others quantitative finance, queueing and reliability theory, are given in the form of transforms and become amenable for practical computations once fast and accurate methods for numerical Laplace and Fourier transform inversion are available.

This thesis presents advanced numerical inversion techniques of Laplace and Fourier transforms and their innovative application in quantitative finance. Although our applications are mainly focused on quantitative finance, the presented inversion techniques also tackles important problems from logistics, queueing, probability theory.

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.



ERIM PhD Series Research in Management

Erasmus Research Institute of Management - ERiM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl