Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

The Gibbs phenomenon in option pricing methods

Filtering and other techniques applied to the COS method

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Mark Versteegh

Delft, Netherlands
October 2012

Copyright © 2012 by Mark Versteegh. All rights reserved

TU Delft Delft University of Technology

MSc thesis APPLIED MATHEMATICS

The Gibbs phenomenon in option pricing methods

Filtering and other techniques applied to the COS method

MARK VERSTEEGH

Delft University of Technology

Daily Supervisors

Prof.dr.ir. C.W. Oosterlee
M.J. Ruijter, MSc.

Responsible professor

Prof.dr.ir. C.W. Oosterlee

Other thesis committee members

Dr.ir. R.J.Fokkink
Dr.ir. J.K. Ryan

October, 2012

Delft

Table of Contents

Preface

This thesis has been written for the degree of Master of Science in Applied Mathematics, faculty of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology. The graduation has been done in the unit of Numerical Analysis, taking about ten months of work.

The first three months have been used for literature study to get an overview of the main research topics, analized with an intermediate report and presentation. This thesis is based upon that report, and the results of the remaining seven months of research.

I would like to thank everyone that has supported me and specifically Kees Oosterlee and Marjon Ruijter for the valuable and fortifying advise and input during this period.

Delft, University of Technology M.Versteegh
October 3, 2012

Chapter 1

Introduction

1-1 Option pricing

An option is the right (but not the obligation) to buy or sell a risky asset at a prespecified price after or within a prespecified period. It is a contract between two parties: the *writer*, who fixes the terms of the contract and sells the option, and a *holder* who purchases the option at the stock market, paying the *option price*.

Options have a limited lifetime, after the *maturity date* T the rights of the holder expire and for later times the option is worthless. There are two basic types of option: the *put* option gives the holder the right to *sell* the underlying asset for an agreed price K (per unit), called the strike price, by the date T. The *call* option gives the holder the right to *buy* for the agreed price K at or before time T.

At time $t = T$ the option holder can check the current price of the underlying asset, denoted by $S = S_T$ and decide to exercise the option or not. For example, for a call option the holder can buy the asset at costs K exercising the option, or he can buy on the spot market for costs S. Therefore the holder will exercise only when $S > K$, which gains $S - K$ per share (ignoring transaction costs). The option value at maturity is therefore equal to its intrinsic value, which is the payoff that would be received if the underlying is at its current level when the option expires [31].
This brings us to the *payoff function* which is the value $v(S, T)$ of an option at the expiration date T. For a call option this is given by:

$$v_C(S_T, T) = (S_T - K)^+ := \max\{S_T - K, 0\}. \tag{1-1}$$

In a similar way the payoff of a put at expiration time T is

$$v_P(S_T, T) := (K - S_T)^+. \tag{1-2}$$

These payoff functions are displayed in figure 1-1 as the solid lines.

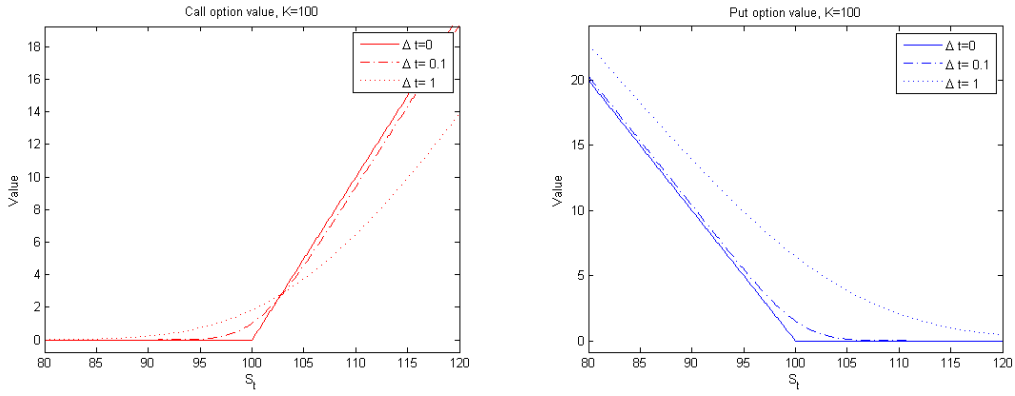

Figure 1-1: The value of plain vanilla European options with exercise price $K = 100$ for different times to maturity.

Prices have a large element of randomness, see figure 1-2 for an example of how the prices behave in practice. This means that the modeling must be done in a probabilistic sense. For times $t < T$ the option prices will differ from the intrinsic value, since the price of the underlying can still change. The option price is based on the expected value at time of maturity, given the current situation. Figure 1-1 shows option values for $\Delta t := T - t = 0.1$ year and $\Delta t = 1$ year, as dashed and dotted lines.

An option can be *in the money*, which means the option has positive intrinsic value, for example a call option for which the asset price is above strike. *Out of the money* options have no intrinsic value and an option is *at the money* when the strike price is equal or close to the current asset level. How to calculate an option price, which should be a fair value of the premium, is one of the central themes in financial mathematics. Mathematical models can serve as approximations and idealizations of the financial world.

The option pricing formula is [31]

$$v_C(S,t) = e^{-r\Delta t}\mathbb{E}[(S_T - K)^+] = e^{-r\Delta t}\int_K^\infty (S_T - K) f(S_T|S_t)\, dS_T, \tag{1-3}$$

where $f(S_T|S_t)$ is the probability density function of S_T under the risk-neutral measure, given S_t and Δt is the time to maturity. Under the assumption that the asset price follows geometric Brownian motion (GBM) $f(S_T|S_t)$ is the lognormal probability density function where the *drift* μ under the risk neutral measure equals the *interest rate* r: $\mu = r$.

One of the most important parameters in almost every pricing model available is *volatility*: a statistical measure of the tendency of a market or security to rise or fall sharply within a period of time. Volatility is typically calculated by using variance of the price or return. A highly volatile market means that prices have huge swings in very short periods of time. High volatility generally results in higher option prices, which is sometimes referred to as *the price of risk*.

There are different classes, sometimes called styles or families of options. Distinguished by the dates on which the option may be exercised one has *European* and *American* options. European options can only be exercised at the expiration date T, while American options can

be exercised at any time up to and including T. As an intermediate between these two there are *Bermudan* options which can be exercised at a prescribed number of times.

The aforementioned options, that include only an expiration date and straightforward strike price are often called *vanilla* options, as opposed to *exotic* options, that have more complex contract details. Some well known exotic examples are the *Asian* option, for which the payoff depends on the average underlying price over some pre-set period of time, the *basket* option, which depends on some combination of different assets and the *barrier* option, for which the value depends on whether the asset price S_t hits a prescribed barrier during its life time.

One of the most successful and widely accepted models is the Black-Scholes model of the market, in which the asset prices are modeled to behave as geometric Brownian motion. See section 2-5-1 for more details about this model. Nowadays there are many other, more complicated models available. A selection of these models, the ones that will be used in the numerical examples later on, will be discussed in section 2-5.

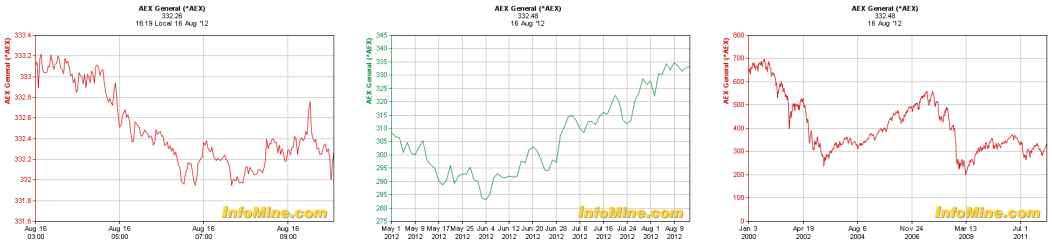

Figure 1-2: AEX index chart over different periods of time. *Charts from Infomine.com.*

Efficient option pricing is very important for the calibration of models, especially those where many options with different strike prices and different dates of maturity need to be priced at the same time. Therefore, fast and accurate pricing methods are a necessity for banks and trading companies. It is generally difficult to find analytic solutions for the problems that arise in practice, so numerical methods are being used to find approximate solutions. Most of the pricing methods that are being used in the financial world nowadays can be divided into three groups:

- Monte-Carlo methods: estimate the option value by generating a large amount of sample paths. These methods are fairly straightforward to implement but are computationally expensive, especially when high accuracy is demanded.
- Partial (integro-)differential equation (PIDE) methods, which use numerical schemes, such as finite element discretization, to directly solve the PIDE.
- Numerical integration methods, where the option price is modeled as the discounted expected value of the payoff at maturity. The numerical integration methods are attractive because of the fast computational speed, especially for plain vanilla options. Usually those methods are combined with the (Fast) Fourier transform or Hilbert transform, and therefore, the numerical integration methods are often referred to as the *transform methods*. Representatives of transform methods are the Carr-Madan method, the CONV method and the Hilbert transform method and the recently developed COS method as proposed in [7]. See [36] and the references there for more about the other methods.

The rest of this thesis focuses on the COS method, which will be described in detail in section 2-4.

1-2 Contents of the thesis

Chapter 2 will introduce the used mathematical tools, namely the Fourier transform and Fourier series, its convergence properties and the Gibbs phenomenon. The COS method for option pricing will also be introduced along with some financial models that will be used later on as test cases. For each of these models there are situations in which relatively slow convergence as a consequence of Gibbs oscillations occurs.

The next chapter, 3, focuses on various methods to improve the convergence rate of the so called spectral methods. Note that we are not just interested in an accurate recovery, but that we also want to be able to perform fast option pricing, so the computational costs should remain relatively low. The chapter contains a brief description of a wide range of possibilities and some of the more promising methods for our subject will be analyzed in depth.

After discussing the possible improvement methods, the numerical examples in chapter 4 will put them to test in practical situations such as density recovery and option pricing for the financial models listed earlier.

Chapter 5 contains conclusions, possible improvements and suggestions for further research.

Chapter 2

Problem description

The Fourier series and Fourier transform, named after the 18th century French mathematician Joseph Fourier, are mathematical tools with applications in many fields, such as engineering, physics, chemistry and applied mathematics. Fourier theory also form the basis of the COS method, therefore we start with the definitions and some useful properties of both the discrete series and the continous transform in section 2-1.

Analysis of the convergence properties of Fourier series in sections 2-2 naturally leads to the Gibbs phenomenon, which will be the subject of section 2-3. From there we move back to the financial modelling, which is done by the COS method defined in section 2-4. Section 2-5 defines some of the financial models in which the Gibbs phenomenon plays a role.

2-1 Introduction to Fourier series and the Fourier transform

The Fourier series and the Fourier transform are actually discrete and continuous counterparts, so in principle both can be chosen as a starting point. Here we start with the Fourier series.

2-1-1 Fourier series

We use the following definition of the Fourier series of a function defined on $[-1, 1]$ as a starting point:

$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}_n e^{in\pi x}. \tag{2-1}$$

The Fourier coefficients, here denoted by $\hat{f}_n$ are given by:

$$\hat{f}_n = \int_{-1}^{1} f(x) e^{-in\pi x}\, dx \tag{2-2}$$

A simple change of variables can be used to transform the interval $[-1, 1]$ to any other interval $[a, b]$.

Using Eulers equation,

$$e^{i\pi x} = \cos(\pi x) + i\sin(\pi x), \tag{2-3}$$

we can transform (2-1) to the equivalent notation using sine and cosine:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\pi x) + b_n \sin(n\pi x), \tag{2-4}$$

with

$$\begin{aligned} a_n &= \int_{-1}^{1} f(x)\cos(n\pi x)\,dx, \\ b_n &= \int_{-1}^{1} f(x)\sin(n\pi x)\,dx. \end{aligned} \tag{2-5}$$

The relation between the coefficients a_n, b_n and $\hat{f}_n$ also follow from 2-3, which can also be formulated as:

$$\begin{aligned} \cos(n\pi x) &= \frac{e^{in\pi x} + e^{-in\pi x}}{2}, \\ \sin(n\pi x) &= \frac{e^{in\pi x} - e^{-in\pi x}}{2i}, \end{aligned} \tag{2-6}$$

and leads to:

$$\begin{aligned} a_n &= \hat{f}_n + \hat{f}_{-n}, \\ b_n &= i(\hat{f}_n - \hat{f}_{-n}). \end{aligned} \tag{2-7}$$

Since we use only non-negative a_n and b_n the reversed version of this relation is:

$$\hat{f}_n = \begin{cases} \frac{a_0}{2} & \text{if } n = 0, \\ \frac{1}{2}(a_{|n|} - i\,\mathrm{sgn}(n) b_{|n|}) & \text{otherwise.} \end{cases} \tag{2-8}$$

The choice for the exponential or the sine/cosine notation is just a matter of taste. Most of the literature about filters, mollifiers and other methods discussed in upcoming chapters is written in exponential notation because in general it is more compact. The COS method on the other hand uses a pure cosine series thus writing it in exponential form would require more terms. In this thesis we use both notations, noting that it is easy to convert from one to another.

In practice we do not use the infinite series, but approximate it with its partial sum:

$$S_N(f)(x) = \sum_{n=-N}^{N} \hat{f}_n e^{\pi i n x}, \tag{2-9}$$

or equivalently:

$$S_N(f)(x) = \frac{a_0}{2} + \sum_{n=1}^{N} a_n \cos(n\pi x) + b_n \sin(n\pi x). \tag{2-10}$$

Since the Fourier basis-functions are orthogonal, the product of two Fourier series satisfies:

$$||fg|| = \int_{-1}^{1} \left[\left(\sum_{n=-\infty}^{\infty} \hat{f}_n e^{in\pi x} \right) \left(\sum_{n=-\infty}^{\infty} \hat{g}_n e^{in\pi x} \right) \right] dx = \sum_{n=-\infty}^{\infty} \hat{f}_n \hat{g}_n \tag{2-11}$$

2-1-2 Fourier transform

Roughly speaking the Fourier transform is a continuous version of the discrete Fourier series. Replacing the integer $n \in \mathbb{Z}$ in the coefficients and the frequencies by a continuous $\omega \in \mathbb{R}$ and changing the integration domain from $[-1, 1]$ to all of $\mathbb{R}$.

The Fourier-transform is defined as

$$\hat{f}(\omega) = \int_{\mathbb{R}} e^{i\omega x} f(x)\, dx \tag{2-12}$$

and the inverse Fourier transform as:

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\omega x} \hat{f}(\omega)\, d\omega. \tag{2-13}$$

Different papers use different definitions, not only regarding the interval, but also regarding the sign of the exponent, thereby switching the roles of forward and reverse transform. This is a matter of taste however, it will not affect any results as long as one is consistent in this choice.

The Fourier transform has some properties that are quite useful in practice, for example for solving differential equations. For Fourier pairs $f(x)$, $\hat{f}(\omega)$ and $g(x)$, $\hat{g}(\omega)$ we have [33]:

(i) $f(x+h) \longrightarrow \hat{f}(\omega)e^{-ih\omega}$

(ii) $f(x+h)e^{ihx} \longrightarrow \hat{f}(\omega+h)$

(iii) $f(cx) \longrightarrow \frac{1}{c}\hat{f}(\frac{\omega}{c})$

(iv) $f'(x) \longrightarrow -i\omega\hat{f}(\omega)$

(v) $ixf(x) \longrightarrow \hat{f}'(\omega)$

(vi) $(f * g)(x) \longrightarrow \hat{f}(\omega)\phi(\omega)$ and $f(x)g(x) \longrightarrow (\hat{f} * \hat{g})(\omega)$

Where $(f*g)(x) := \int_{-\infty}^{\infty} f(y)g(x-y)\, dy$, so the last property states that the Fourier transform interchanges pointwise products with convolutions.

If $f(x)$ and $g(x)$ are integrable and square integrable, Parseval's theorem states that

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}\,dx = \int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}(\omega)}\,d\omega, \tag{2-14}$$

where the bar denotes complex conjugation. Equivalently, Plancherels theorem states that for such $f(x)$

$$||f|| = ||\varphi||. \tag{2-15}$$

So basically we can calculate the product of $f(x)$ and $g(x)$ directly from the Fourier transforms. The discrete analog of this which will be used in the COS pricing formula that will be defined later on in section 2-4.

If we now assume that $f(x)$ is normalized such that $\int_{-\infty}^{\infty} |f(x)|^2\,dx = 1$ then Heisenberg uncertainty principle states that the second moment of a function and it's Fourier transform cannot both be arbitrarily small at some point

$$\left(\int_{-\infty}^{\infty} (x-x_0)^2|f(x)|^2\,dx\right)\left(\int_{-\infty}^{\infty} (\omega-\omega_0)^2|\hat{f}(\omega)|^2\,dx\right) \geq \frac{1}{16\pi^2} \tag{2-16}$$

for any x_0, $\omega_0 \in \mathbb{R}$. This implies that one cannot at the same time localize the value of a function and its Fourier transform. If for example we take a perfectly localized Dirac delta function (neglecting the fact that formally this is not a function), its Fourier transform never decays, it is constant. The Fourier transform of a box-function, is a sinc-function, which decays very slowly, and so on. The best we can get in this regard is exponential decay in both domains, which is the case for a Gaussian $\exp(-at^2)$, as the Fourier transform of a Gaussian is again a Gaussian. For any other function the decay in one of the domains will be slower. This will play a role in the filtering theory later on, since it gives a principle bound to the precision of filtering.

2-1-3 Characteristic functions

The characteristic function (CF) of a random variable X is defined as

$$\varphi(\omega) = \mathbb{E}[e^{i\omega X}]. \tag{2-17}$$

This is equal to the Fourier transform of its distribution. Therefore many probabilistic properties of random variables correspond to analytical properties of their characteristic functions.

The CF of a random variable completely characterizes its behaviour. Two functions with equal CF's are identically distributed. The n-the moment of a random variable X on $\mathbb{R}$ is defined by $m_n(X) = \mathbb{E}[X^n]$.

For a Gaussian random variable X with mean μ and variance σ^2, as used in the Black-Scholes model, the characteristic function is given by:

$$E[\exp(i\omega X)] = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega x} e^{(x-\mu)^2/2\sigma^2}\,dx = e^{i\omega\mu - \frac{1}{2}\omega^2\sigma^2} \tag{2-18}$$

The characteristic function of the log-asset price, $\ln(S_t/K)$ is given in Table 1.1 of [5] for various models.

2-2 Convergence

Now that we know about the existence of the Fourier series, for which $f(x)$ only has to be integrable on $[a, b]$ so that the coefficients in (3-2) are defined, we want to be certain that the series converges and want to know how fast this happens. For discussing the convergence of the used methods we need some definitions, which are taken from [1].

2-2-1 Convergence rates

These definitions are formulated in a general form, so they will also be useful later on in discussing convergence for more general spectral methods.

Definition 1. *The algebraic index of convergence k is the largest number for which*

$$\lim_{n\to\infty} |a_n| n^k < \infty \tag{2-19}$$

where the a_n are the coefficients of the series.

Alternatively, we have convergence of order k if the coefficients a_n decay at rate $a_n \sim O(|n|^{-(k+1)})$ for $n \to \infty$.

Definition 2. *If the algebraic index of convergence k is unbounded then the series is said to have exponential, or spectral convergence.*

Different rates of exponential convergence can be further categorized as geometric, sub- and supergeometric. Precise definitions are given by Boyd in [1], see figure 2-1 for a visual representation, also taken from [1].

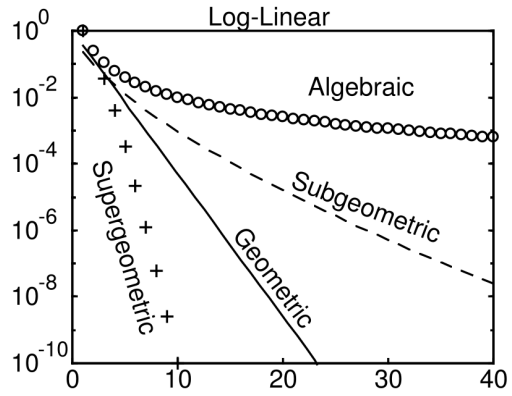

Figure 2-1: $\log|a_n|$ versus n for different rates of convergence. Circles: algebraic convergence, such as $a_n \sim 1/n^2$. Dashed: subgeometric convergence, such as $a_n \sim \exp(-1.5n^{2/3})$. Solid: geometric convergence, such as $a_n \sim exp(-\mu n)$ for any positive μ. Pluses: supergeometric, such as $a_n \sim \exp(-n\log(n))$ or faster decay

2-2-2 Convergence of the Fourier series

Convergence in norm is guaranteed when f is square integrable, for $f \in L^2[-1,1]$ we have:

$$\lim_{n\to\infty} \int_{-1}^{1} |f(x) - S_N(f)(x)|^2 \, dx = 0 \tag{2-20}$$

In practice this is often all we need, but to be able to say something about the convergence rate we also need to look at pointwise convergence. The following theorems are taken from [33].

Theorem 1. ***Uniqueness theorem:*** *If f is a continuous periodic function on $[-1,1]$, so that $f(-1) = f(1)$, and that the Fourier series of f is absolutely convergent: $\sum_{n=-\infty}^{\infty} |\hat{f}_n| < \infty$, then this Fourier series converges uniformly to f:*

$$\lim_{N\to\infty} S_N(f)(x) = f(x) \text{ uniformly for } x \in [-1,1]. \tag{2-21}$$

For non-continuous functions we already know that there is no uniform convergence, but the Fourier expansion of any function in L^2 converges almost everywhere.

In case there are discontinuities of any order the regular, unmodified, Fourier series expansion only has algebraic convergence. For jump discontinuities, 0th order, this leads to the Gibbs-overshoot.

How fast the coefficients converge depends on the smoothness of the function, as is stated in the following theorem:

Theorem 2. ***Decay of coefficients:*** *Suppose that f is a k times continuously differentiable function, $f \in C^k[-1,1]$, with $f^{(l)}(-1) = f^{(l)}(1)$ for $l = 0..k$, then*

$$\hat{f}_n \sim O(\frac{1}{|n|^k}) \text{ as } |n| \to \infty \tag{2-22}$$

In other words, the Fourier coefficients of a k-times differentiable function decay algebraically faster than $|n|^{-k}$ as $n \to \pm\infty$. The proof of this boils down to integration by parts: the nth Fourier coefficient of $\frac{d^k f}{dt^k}$ is given by $(in)^k \hat{f}(n)$, If $f \in C^k$, the Fourier coefficients of the continuous function $\frac{d^k f}{dt^k}$ go to zero, so $|n|^k |\hat{f}(n)|$ goes to zero.

The exact speed of the decay also depends on the size of the discontinuity. If you have a small discontinuity in the kth derivative, you might get an error term like $\epsilon \frac{1}{|n|^{k-\frac{1}{2}}}$. For any $\epsilon > 0$ this will still be the dominant term if $|n|$ is large enough, but for lower n the effect might not be that big.

For $f(x) \in C^\infty[-1,1]$ this results in exponential convergence of the Fourier series. An example of this is shown in figure 2-2. Figures 2-3 and 2-4 show the recovery for functions with a jump and a first order discontinuity.

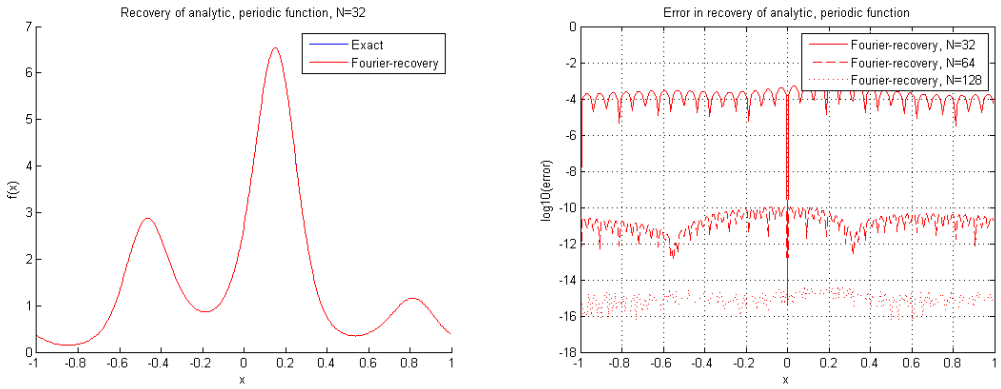

Figure 2-2: *Left:* Recovery of analytic periodic function $f(x) = \exp(\sin(3\pi x) + \cos(\pi x))$ using partial Fourier-sum $S_N f$ with $N = 32$. *Right:* Log-error in the recovery, for $N = 32$, $N = 64$ and $N = 128$.

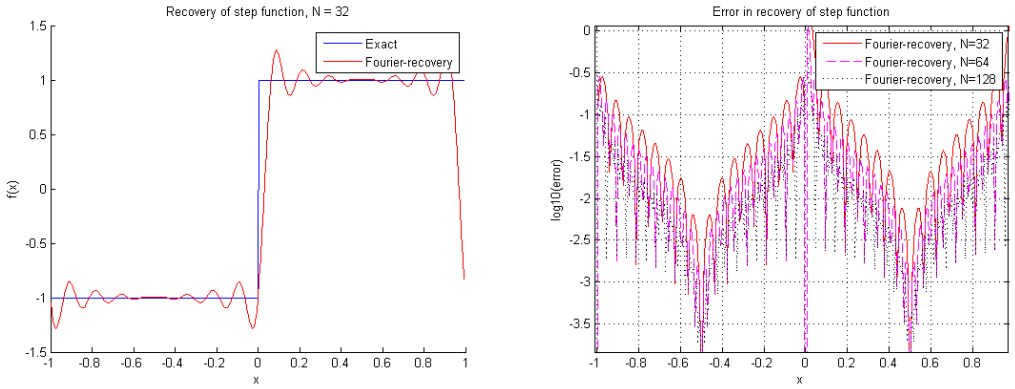

Figure 2-3: *Left:* Recovery of stepfunction $f(x) = -1$ for $x \in [-1, 0)$, 1 for $x \in [0, 1]$ using partial Fourier-sum $S_N f$ with $N = 32$. *Right:* Log-error in the recovery, for $N = 32$, $N = 64$ and $N = 128$.

2-3 Gibbs phenomenon

2-3-1 Introduction

The Gibbs phenomenon was already described in 1848, but became well known in 1898 when Albert Michelson observed it via a mechanical graphing machine that he used to compute and re-synthesize the Fourier series of a square wave. Soon after this observation J. Willard Gibbs published a paper about it, but only in 1906 a detailed mathematical description of the phenomenon was given and named after Gibbs.

The Gibbs phenomenon reflects the difficulty in approximating a discontinuous function by a finite Fourier series. Although the limit of the partial sums represents the original function exactly, in the finite there is always an overshoot at a jump discontinuity. The width of this overshoots decreases with n, but the height of the maximum does not.

In case of a jump discontinuity we have pointwise convergence almost everywhere, but not uniform convergence. For a piecewise continuous function the Fourier series converges to

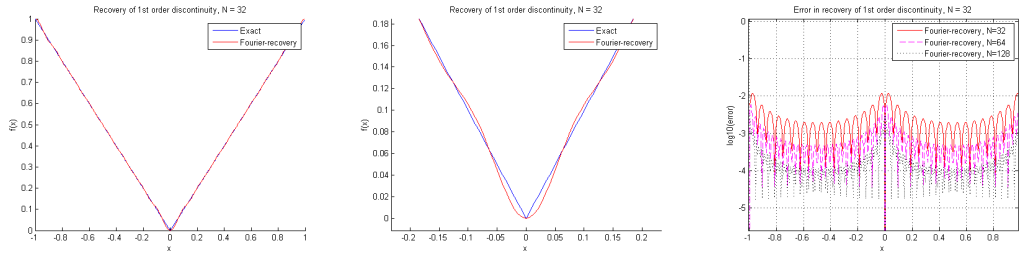

Figure 2-4: *Left:* Recovery of continuous periodic function $f(x) = |x|$ using partial Fourier-sum $S_N f$ with $N = 32$. *Middle:* Closeup of recovery near first order discontinuity, at $x = 0$. *Right:* Log-error in the recovery, for $N = 32$, $N = 64$ and $N = 128$.

the function at every point except at the jump discontinuities. At the jump discontinuities themselves the limit will converge to the average of the values of the function on either side of the jump.

The Gibbs phenomenon is also closely related to the principle that the decay of the Fourier coefficients of a function at infinity is controlled by the smoothness of that function; very smooth functions will have very rapidly decaying Fourier coefficients (resulting in the rapid convergence of the Fourier series), whereas discontinuous functions will have very slowly decaying Fourier coefficients (causing the Fourier series to converge very slowly).

The local effect of the Gibbs phenomenon results in oscillations near the jumps, thereby causing a lack of uniform convergence. But there also is a global effect: although the error decays away from the jumps, the decay rate is only first order. Thus the existence or one or more discontinuities slows down the convergence rate over the whole domain. Spectral accuracy is lost [16], [33].

2-3-2 Gibbs constant

The slow decay of the coefficients, resulting in slow convergence on the whole domain has already been discussed in section 2-2, so here we focus on the local convergence near the jump.

If we take a simple step function, the same function as is shown in figure 2-3, for $x \in [-1, 1]$ defined as:

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases} \tag{2-23}$$

and calculate the coefficients, which is pretty straightforward in this case, we get:

$$f(x) = \frac{1}{2} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{\sin((2n-1)\pi x)}{2n-1} \tag{2-24}$$

Since this function is obviously integrable, we know that the series converges in L^2:

$$\lim_{n\to\infty} \int_{-1}^{1} |f(x) - S_N(f)(x)|^2 \, dx = \lim_{n\to\infty} ||f(x) - S_N(f)(x)||^2 = 0 \tag{2-25}$$

But since there is a discontinuity, there can be no pointwise convergence everywhere. At $x = 0$ the series converges to $(f(0^-) + f(0^+))/2$, so now we are interested in the points close to this discontinuity.

If we truncate the series (2-24), because we are using the coefficients $\{1, 3, \ldots, 2N-1\}$ and the others happen to be zero here, we get the partial sum $S_{2N}(f)(x)$. The first local maximum of the partial sum is located at $x = 1/2N$, if we evaluate the partial sum at this point we get:

$$S_{2N}(f)\left(\frac{1}{2N}\right) = \frac{1}{2} + \frac{2}{\pi}\sum_{n=1}^{N}\frac{\sin((2n-1)\pi/2N)}{2n-1} \tag{2-26}$$

Using the normalized sinc-function, defined as

$$\operatorname{sinc}(x) := \frac{\sin(\pi x)}{\pi x}, \tag{2-27}$$

we can rewrite (2-26) to get:

$$S_{2N}(f)\left(\frac{1}{2N}\right) = \frac{1}{2} + \frac{1}{N}\sum_{n=1}^{N}\frac{\sin\left(\frac{(2n-1)\pi}{2N}\right)}{\frac{(2n-1)\pi}{2N}} = \frac{1}{2} + \frac{1}{N}\sum_{n=1}^{N}\operatorname{sinc}\left(\frac{(2n-1)\pi}{2N}\right) \approx \frac{1}{2} + \int_0^1 \operatorname{sinc}(x)\,dx \tag{2-28}$$

In the last step we used the fact that the sum here can be seen as a first order numerical approximation of the integral of the sinc-function. Now the last term on the right hand determines the height of the overshoot. This contant is known as the Gibbs constant:

$$2\int_0^1 \operatorname{sinc}(x)\,dx \approx 1.17689796 \tag{2-29}$$

Similar arguments can be used in a general case to prove that the overshoot in case of a discontinuity is always approximately 9% in both directions.

2-3-3 Gibbs phenomenon in applications

There are numerous applications in which the Gibbs phenomenon occurs. For instance, many fields use the Fourier-transform as a tool to solve differential equations. This works so well because of some specific properties of the transformation. However, once one has obtained a solution in Fourier-space one often wants to know the corresponding solution in a time or space domain. For most practical problems this inversion is not known analytically and thus has to be calculated numerically. If such a solution is not perfectly smooth, the Gibbs phenomenon occurs, slowing down the convergence significantly. If no additional measures are taken this results in extra computation time and/or inaccurate results. An example of Gibbs oscillations in a numerical simulation of a conduction problem is seen in figure 2-6.

In signal processing, the Gibbs phenomenon is undesirable because it causes artifacts, namely clipping from the overshoot and undershoot, and ringing artifacts from the oscillations. In

the case of low-pass filtering, these can be reduced or eliminated by using different low-pass filters.

In MRI, the Gibbs phenomenon causes artifacts in the presence of adjacent regions of markedly differing signal intensity. This is most commonly encountered in spinal MR imaging, where the Gibbs phenomenon may simulate the appearance of syringomyelia, see figures 2-5, 2-7 [8].

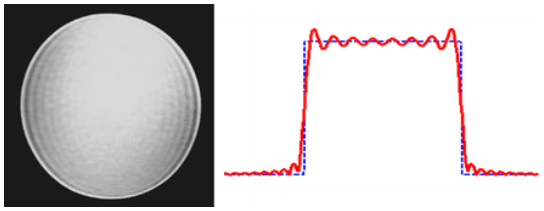

Figure 2-5: Gibbs artefact in MRI image, taken from *http://radiographics.rsna.org/*

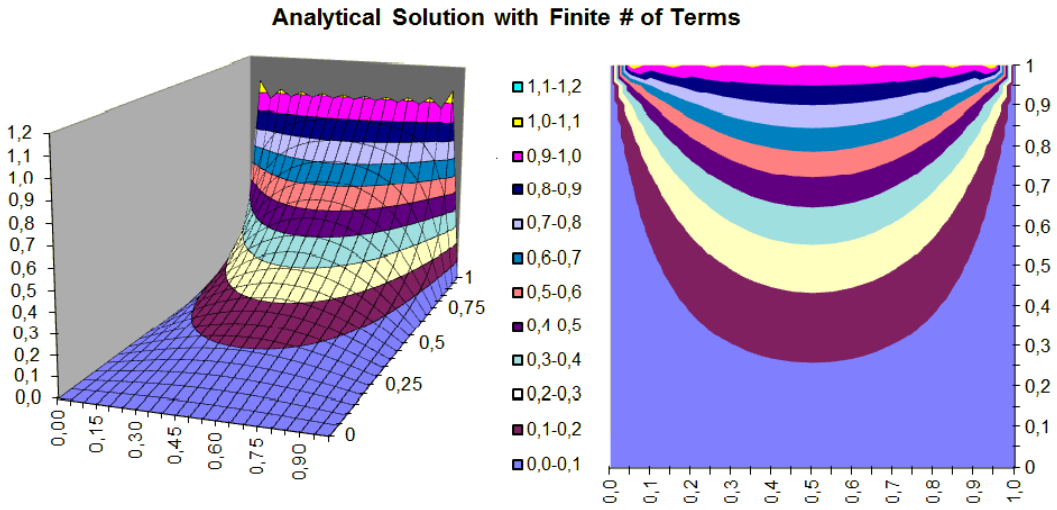

Figure 2-6: 2-Dimensional steady-state conduction problem. The back-edge is at T=1, the others at T=0. Analytical solution with a finite number of terms. The Gibbs phenomenon is clearly visible near the edges in the upper region. Image generated with module from: *http://www.faculty.virginia.edu/ribando/modules/xls/*

2-4 COS method

We now focus on the appearance of Gibbs phenomena in option pricing with Fourier techniques. The COS method, short for Fourier cosine pricing method, is based on the risk-neutral valuation formula for an option value v:

$$v(x,t_0) = e^{-r\Delta t}\mathbb{E}_{\mathbb{Q}}[v(y,T)|x] = e^{-r\Delta t}\int_{\mathbb{R}} v(y,T)f(y|x)dy. \tag{2-30}$$

Here $\mathbb{E}_Q$ is the expectation under risk-neutral measure $\mathbb{Q}$; Δt is the difference between the time of maturity T and the initial time t_0; $f(y|x)$ is the conditional probability density and r is the risk-neutral interest rate [5].

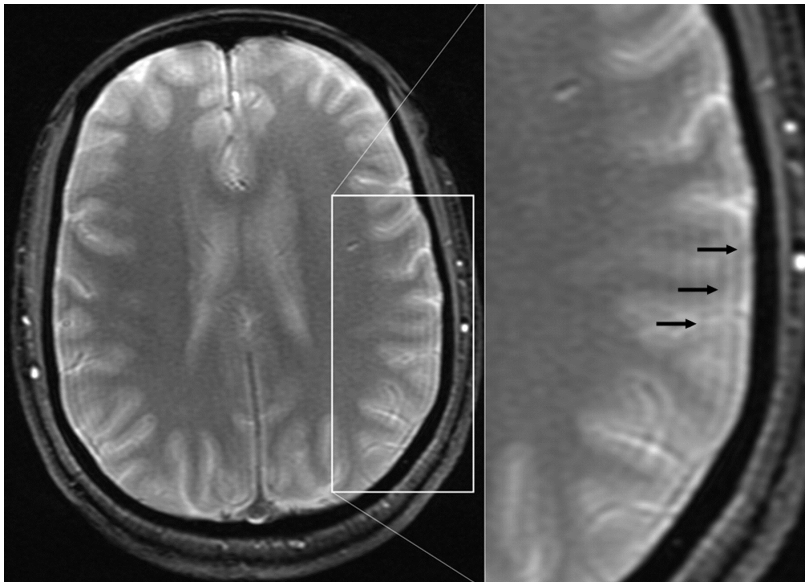

Figure 2-7: Gibbs artifact (also called truncation or ringing artifact). Axial gradient-echo image of brain obtained at 256 Œ 160 matrix. Gibbs artifact near inner table of calvarium manifests as subtle hypointense lines overlying cortex (arrows), image from [8].

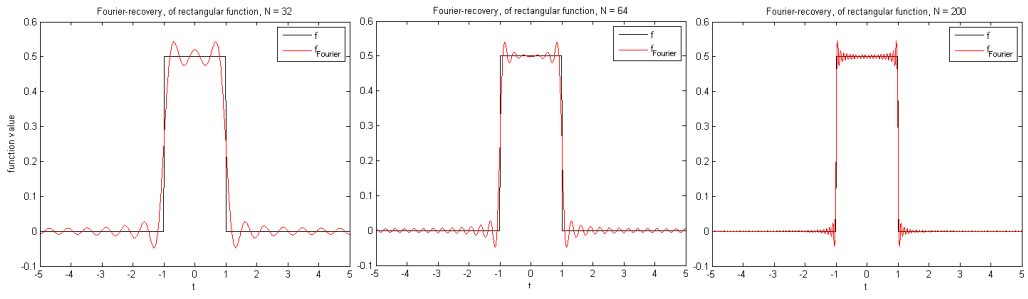

Figure 2-8: Fourier-recovery of rectangular function for N=32, N=64 and N=200. The overshoot at the discontinuities remains around 9% in both directions

x and y are state variables, which can be any monotone function of the underlying asset, but from here it is taken to be the log asset-price, denoted as $x = \log(S/K)$.
In the type of problems we are handling an explicit expression for the conditional probability density $f(y|x)$ is often not known, or it involves some mathematical special functions, which makes it impractical to calculate. Instead the characteristic function $\varphi(\omega, x, t)$ of $f(y|x)$, defined as

$$\varphi(\omega, x, \Delta t) := \mathbb{E}(e^{i\omega X_T} | X_{t_0} = x, T - t_0 = \Delta t). \tag{2-31}$$

is often known.

The density and the characteristic function (CF) are a Fourier pair, where the characteristic function is the Fourier transform of the density:

$$\varphi(\omega, x, \Delta t) = \int_{\mathbb{R}} e^{i\omega y} f(y|x)\, dy, \tag{2-32}$$

and the density is the inverse Fourier transform of the CF:

$$f(y|x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\omega y} \varphi(\omega, x, \Delta t)\, d\omega. \tag{2-33}$$

So we could just take this inverse Fourier transform and plug it into equation (2-30) to calculate $v(x, t_0)$, but that would be a computationally expensive operation.

The COS method evaluates the inverse Fourier integral as a Fourier cosine series expansion, which for a real-valued function $f(x)$ supported on a finite interval $[a, b]$ we can write down as

$$f(y|x) = \sum_{n=0}^{\infty}{}' A_n(x) \cos\left(n\pi \frac{y-a}{b-a}\right), \tag{2-34}$$

where the prime $'$ indicates dividing the first term (A_0) by 2. The Fourier-coefficients are given by

$$A_n(x) = \frac{2}{b-a} \int_a^b f(y|x) \cos\left(n\pi \frac{y-a}{b-a}\right) dy. \tag{2-35}$$

This is equivalent to the series defined in (2-4), with $b_n = 0$ for all n, using a linear transformation from $[0, 1]$ to $[a, b]$ and the fact that we only need half the interval to evaluate the integral of an even function.

So this can be seen as a regular Fourier cosine series of a function $f(x) = f(|x - a|)$, on the interval $[2a - b, b]$, which is mirrored around the midpoint a to make it an even function.

Chebychev series expansions usually give an optimal approximation of functions with a finite support [1]. For functions where the odd derivatives at the boundaries are zero, the cosine expansion of $f(x)$ as used in the COS method is equivalent to the Chebyshev series expansion of $f(cos^{-1}(t))$ in t [5].

In a similar way we can also replace the payoff function by its Fourier series:

$$v(x,T) = \sum_{n=0}^{\infty}{}' V_n \cos\left(n\pi \frac{x-a}{b-a}\right), \tag{2-36}$$

with the coefficients V_n given by

$$V_n = \frac{2}{b-a}\int_a^b v(y,T)\cos\left(n\pi\frac{y-a}{b-a}\right)\,dy. \tag{2-37}$$

For several payoff functions, including plain vanilla puts and calls, the cosine coefficients are available in closed form [5].

Plugging the series defined in (2-34) and (2-36) into equation (2-30) and using the orthogonality of the cosines we get a formula for $v(x,t_0)$ written as the product of the Fourier coefficients:

$$v(x,t_0) = \frac{b-a}{2}e^{-r\Delta t}\sum_{n=0}^{\infty}{}' A_n(x)V_n \tag{2-38}$$

where the A_n and V_n are defined as in (2-35) and (2-37)

The key step of the COS method is approximating the Fourier-coefficients A_n of the density function $f(y|x)$ using its characteristic function $\varphi(\omega,x,t)$. If the density function $f(y|x)$ decays rapidly to zero for $y \to \pm\infty$, the integration range can be truncated without losing any significant accuracy:

$$\varphi(\omega) = \int_{\mathbb{R}} e^{i\omega x}f(x)\,dx \approx \int_a^b e^{i\omega x}f(x)\,dx = \varphi_1(\omega) \tag{2-39}$$

Therefore the characteristic function (which we assume to be known) can be used to efficiently calculate the Fourier-coefficients. Combining (2-35) and (2-39):

$$A_n(x) \equiv \frac{2}{b-a}\mathrm{Re}\left\{\varphi_1(\frac{n\pi}{b-a},x,\Delta t)e^{-i\frac{na\pi}{b-a}}\right\} \approx \frac{2}{b-a}\mathrm{Re}\left\{\varphi(\frac{\pi}{b-a},x,\Delta t)e^{-i\frac{na\pi}{b-a}}\right\} = F_n(x) \tag{2-40}$$

where Re denotes taking the real part.

Truncating the series, we get our next approximation, the COS-pricing formula:

$$v(x,t_0) \approx e^{-r\Delta t}\sum_{n=0}^{N-1}{}' \mathrm{Re}\left\{\varphi(\frac{n\pi}{b-a},x,\Delta t)e^{-in\pi\frac{a}{b-a}})\right\}V_n \tag{2-41}$$

For processes with independent increments, such as exponential Levy processes, which include the log-versions of Geometric Brownian Motion (GBM), Variance-Gamma, and CGMY, the characteristic function can be written as the product of $e^{i\omega x}$ and a part independent of x:

$$\varphi(\omega,x,\Delta t) = e^{i\omega x}\phi(\omega,\Delta t). \tag{2-42}$$

Inserting this in (2-41) gives

$$v(x,t_0) \approx e^{-r\Delta t} \sum_{n=0}^{N-1}{}' \operatorname{Re}\left\{\phi(\frac{n\pi}{b-a},\Delta t)e^{in\pi\frac{x-a}{b-a}})\right\} V_n, \tag{2-43}$$

which allows us to calculate option values for many strike-prices at the same time, at little extra computational cost, because if we now input a vector $\boldsymbol{x}$ we have to evaluate the $\phi(\frac{n\pi}{b-a},\Delta t)$ only once (for each n), because it is independent of x [7].

2-4-1 Truncation Range

The integration range $[a,b]$ must be chosen carefully to avoid significant errors. An interval which is too small or too wide will result in significant integration- or truncation range errors. This will be further covered in the upcoming error analysis section.

For now we mention the results given by [5], which give a rule of thumb for choosing the integration range:

$$[a,b] := \left[(\xi_1+x_0) - L\sqrt{\xi_2+\sqrt{\xi_4}}, (\xi_1+x_0) + L\sqrt{\xi_2+\sqrt{\xi_4}}\right] \tag{2-44}$$

This interval is centered around $x_0 := \log(S_0/K)$, $\xi_1,...,\xi_4$ are the cumulants of the underlying stochastic process, given in table 1.2 of [5] and $L \in [6,12]$ depends on a user-defined tolerance level.

2-4-2 Error Analysis

The error is composed of three parts: The series truncation error, the integration range truncation error and the error related to approximating A_n by F_n [5]:

The integration range error ϵ_1 depends on the payoff function and the integration interval:

$$\epsilon_1 := e^{-r\Delta t} \int_{\mathbb{R}\setminus[a,b]} v(y,T) f(y|x)\, dy. \tag{2-45}$$

The series truncation error ϵ_2 depends on the smoothness of the underlying probability density function $f(y|x)$:

$$\epsilon_2 := e^{-r\Delta t} \sum_{n=N}^{\infty} \operatorname{Re}\left\{e^{-i\frac{n\pi a}{b-a}} \int_a^b e^{i\frac{n\pi}{b-a}y} f(y|x)\, dy\right\} V_n \tag{2-46}$$

ϵ_3 is the error related to approximating the real Fourier cosine coefficients A_n by the coefficients F_n that are calculated using the Fourier transform:

$$\epsilon_3 := e^{-r\Delta t} \sum_{n=0}^{N-1}{}' \operatorname{Re}\left\{\int_{\mathbb{R}\setminus[a,b]} e^{in\pi\frac{y-a}{b-a}} f(y|x)\, dy\right\} V_n \tag{2-47}$$

In general the convergence rate of the Fourier series depends on the smoothness of the function on the expansion interval. See section 2-1 and [5] for more details. A detailed error analysis is given in [7], where it is shown that if $f(x) \in \mathbb{C}^{\infty}[a,b]$ then with a properly chosen truncation range, its Fourier cosine series expansion on $[a,b]$ has geometric convergence.

Put-Call Relations

When pricing call options, the solutions accuracy is very sensitive to the width of the integration domain [36],[5].

The payoff of a call grows exponentially in log-stock price, which may result in larger errors, especially under fat-tailed distributions and in models involving multiple time-steps such as Bermudan options.

Often it is possible to avoid some of these resulting complications by using the *put-call parity*. Using

$$(S-K)^+ - (K-S)^+ \equiv S-K, \tag{2-48}$$

we can derive the following relation between the call-price v_C and the put price v^P:

$$v_C(x,t) = v_P(x,t) + S_t e^{-q\Delta t} - Ke^{-r\Delta t}. \tag{2-49}$$

Here r is the rate of interest and q is the dividend rate. In [36] a significant improvement calculating European call option values using the put-call parity compared to the direct calculation with the COS method is shown.

Although good results are obtained using put-call relations, this approach is not always possible. There are still problems for which no such relations exist or they have not been found yet. In these cases the integration range error in ϵ_1 and ϵ_3 remains a problem, so we also want to see if our suggested methods offer any help in these situations.

2-5 Models of interest and Gibb's phenomenon

At present there are many different asset pricing models available. Here we briefly discuss the ones used in the numerical experiments later on.

2-5-1 The Black-Scholes Model

In the Black-Scholes model the asset price S_t follows a *geometric Brownian motion* (GBM) and satisfies the following differential equation:

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t. \tag{2-50}$$

Here W_t is a Brownian Motion or Wiener process, μ is the drift and σ the volatility, which are both assumed to be constant.

The option value v can be obtained by solving the famous Black-Scholes equation, derived with the help of Itô's lemma [31]:

$$\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + rS\frac{\partial v}{\partial S} - rv = 0. \tag{2-51}$$

For the Black-Scholes model a closed form solution of the option price for plain vanilla options is known. For a call option for a non-dividend paying underlying stock the value is:

$$\begin{aligned} v_C(S,t) &= N(d_1)S - N(d_2)Ke^{-r(T-t)}, \\ d_1 &= \frac{ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma\sqrt{T-t} = \frac{ln(\frac{S}{K}) + (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}. \end{aligned} \tag{2-52}$$

Here K is the strike price and $N(.)$ is the standard normal cumulative distribution function.

The value of a put option is obtained by applying the put-call parity on (2-52):

$$\begin{aligned} v_P(S,t) &= Ke^{-r(T-t)} - S + C(S,t), \\ &= -N(-d_1)S + N(-d_2)Ke^{-r(T-t)}. \end{aligned} \tag{2-53}$$

The characteristic function of the log asset price, $X_t = \log(S_t)$ under the GBM model is given by:

$$\varphi(\omega, t) = \exp(i\omega\bar{\mu}t - \frac{1}{2}\sigma^2\omega^2 t), \tag{2-54}$$

where $\bar{\mu} = \mu - \frac{1}{2}\sigma^2$.

The Black-Scholes model has turned out to be very popular, but is based on several crucial assumptions, such as no market friction, constraints on stock holdings, constant drift and volatility. Because empirical data shows that the log returns do not really behave according to a normal distribution many other models have been developed.

Nonetheless, the Black-Scholes model serves as a good starting point. The availability of closed form solutions makes it easy to verify results generated with numerical pricing methods. The COS method shows exponential convergence under GBM, which can be explained by the fact that the normal density function decays fast and is infinitely differentiable.

2-5-2 The Heston Model

Stephen L. Heston [20] proposed a 2-dimensional model in which the volatility is not constant, but instead follows a stochastic process, which can be correlated to the price process to take into account the leverage effect:

$$\begin{aligned} dS_t &= \mu S_t\,dt + \sqrt{\nu_t} S_t\,dW_t^1, \\ d\nu_t &= \kappa(\theta - \nu_t)\,dt + \eta\sqrt{\nu_t}\,dW_t^2, \\ dW_t^1\,dW_t^2 &= \rho\,dt. \end{aligned} \tag{2-55}$$

Here S_t is the price process, ν_t is the variance and W_t^1, W_t^2 are correlated Brownian motion processes with correlation parameter ρ. ν_t is modelled by a square root mean reverting process with long-run mean θ and rate reversion κ and η is the volatility of the volatility. This process is also known as a Cox-Ingersoll-Ross (CIR)-process and will be described in some more detail in section 2-5-4.

One of the main advantages of the Heston model is that the volatility surfaces generated by it are often looking similar to the ones empirically observed in the Black-Scholes model.

On the downside, the model predictions still are not consistent with market prices. As a result financial engineers have to recalibrate model parameters very often to new market data. Furthermore, the generalization of Heston model for variable (time dependent) parameters is not straightforward [19].

If the underlying densities for both processes are smooth, the COS method converges exponentially as is shown in [5]. However, typically this is not the case. In section 4-4 some numerical examples using the Heston model will be discussed.

2-5-3 Variance Gamma Model

The VG process is obtained by evaluating Brownian motion with drift μ and volatility σ at a random time given by a gamma process with mean 1 and variance rate ν, [25],[30]:

$$X(t;\sigma,\nu,\theta) = \theta\gamma(t;1,\nu) + \sigma W(\gamma(t;1,\nu)). \tag{2-56}$$

The VG process is of bounded variation, has independent increments and governs an infinite arrival of jumps:

$$\int_{-\infty}^{\infty} \nu_{Levy}\,dx = \infty, \tag{2-57}$$

where ν_{Levy} is the Levy-measure.

The VG density has a *fat tail*: it is suitable to model phenomena where relatively large values are more probable than would be the case for the normal distribution.

The characteristic function $\phi_{VG}(\omega) = \mathbb{E}[e^{i\omega X_t}]$ is given by [30],[25]:

$$\phi_{VG}(\omega,\Delta t) = (1 - i\omega\theta\nu + \frac{1}{2}\sigma^2\nu\omega^2)^{-\Delta t/\nu} = \left(\frac{GM}{GM + (M-G)i\omega + \omega^2}\right)^{C\Delta t}, \tag{2-58}$$

where

$$\begin{aligned} C &= 1/\nu > 0 \\ G &= (\sqrt{\frac{1}{4}\theta^2\nu^2 + \frac{1}{2}\sigma^2\nu} - \frac{1}{2}\theta\nu)^{-1} > 0, \\ M &= (\sqrt{\frac{1}{4}\theta^2\nu^2 + \frac{1}{2}\sigma^2\nu} + \frac{1}{2}\theta\nu)^{-1} > 0, \end{aligned} \tag{2-59}$$

This can be split in two parts, giving

$$\nu_{VG}(\,dx) = \begin{cases} Ce^{Gx}|x|^{-1}\,dx & \text{for x} < 0, \\ Ce^{-Mx}x^{-1}\,dx & \text{for x} > 0. \end{cases} \tag{2-60}$$

In [25] the following expression for the VG-density function is derived:

$$f_{VG}(x) = \int_0^\infty \frac{1}{\sigma\sqrt{2\pi g}} \exp\left(-\frac{(x-\theta g)^2}{2\sigma^2 g}\right) \frac{g^{\frac{t}{\nu}-1}\exp(-\frac{g}{\nu})}{\nu^{\frac{t}{\nu}}\Gamma(\frac{t}{\nu})}\,dg. \tag{2-61}$$

So in theory it is possible to use this expression for option pricing, but it is computationally expensive to evaluate this function for each point in the domain one is interested in.

The VG-density can be seen as a special case of the *CGMY-model*, named after the authors Car, Geman, Madan and Yor [30]. This is a continuous time model that allows for both diffusions and for jumps of finite and infinite activity, depending on the choice of parameters.

Figures 2-9 and 2-10 show the VG-density for different values of T. For smaller values of T the peak gets really sharp and therefore hard to approximate using a cosine series.

Figures 2-11, 2-12 show the influence of parameters ν and σ. For increasing ν the peak gets sharper and higher, while increasing σ results in a lower, peak and higher tails, without altering the smoothness of the function around the peak.

The sharp peak, a very large first order discontinuity, that can be seen in these figures results in algebraic decay of the Fourier coefficients and therefore slow convergence in the COS method. Table 2.7 in [5], $T = 0.1$ shows results for the COS method with slow convergence. In the numerical section 4-2, improvements methods will be applied to speed up this convergence.

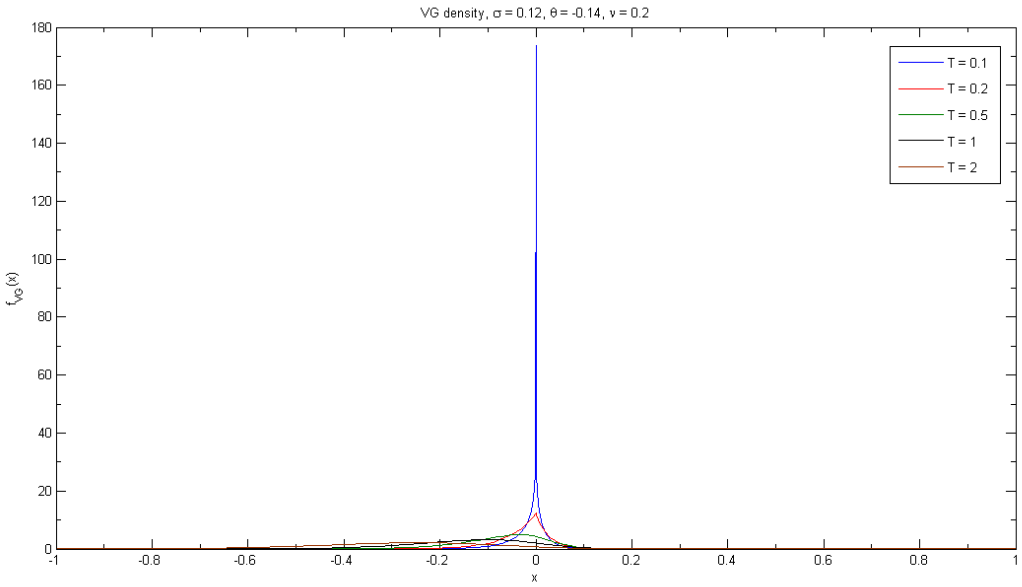

Figure 2-9: VG-density for different values of T

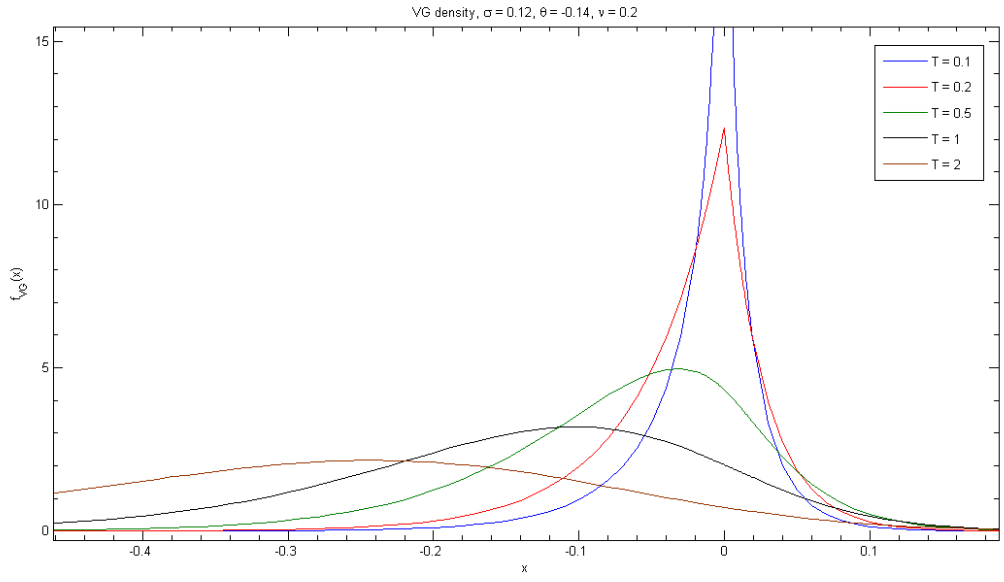

Figure 2-10: VG density: closeup of peaks

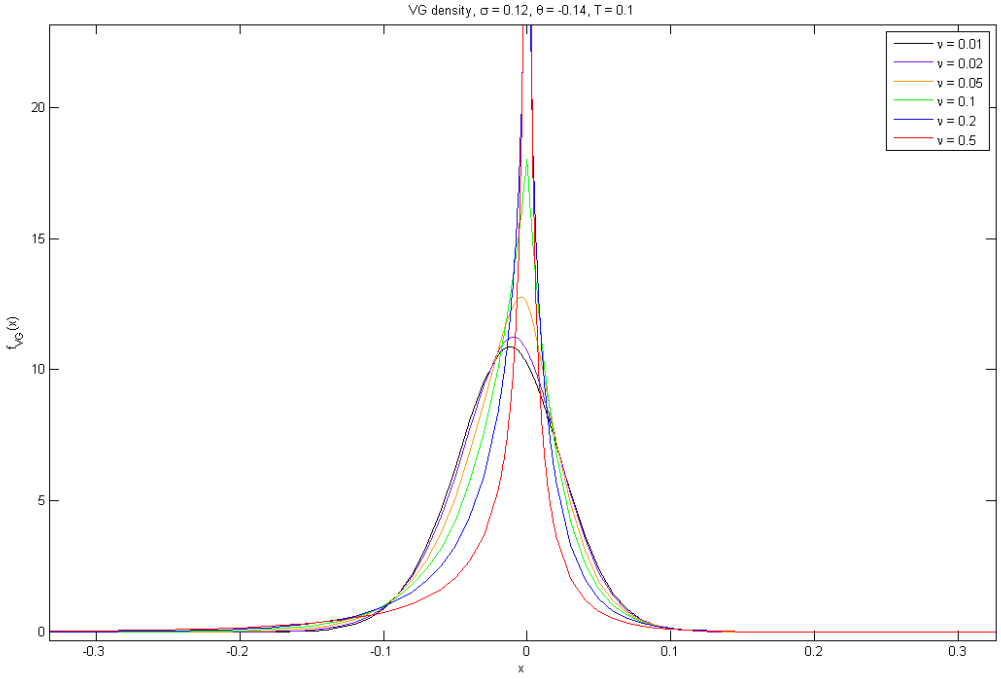

Figure 2-11: VG-density for different values of ν

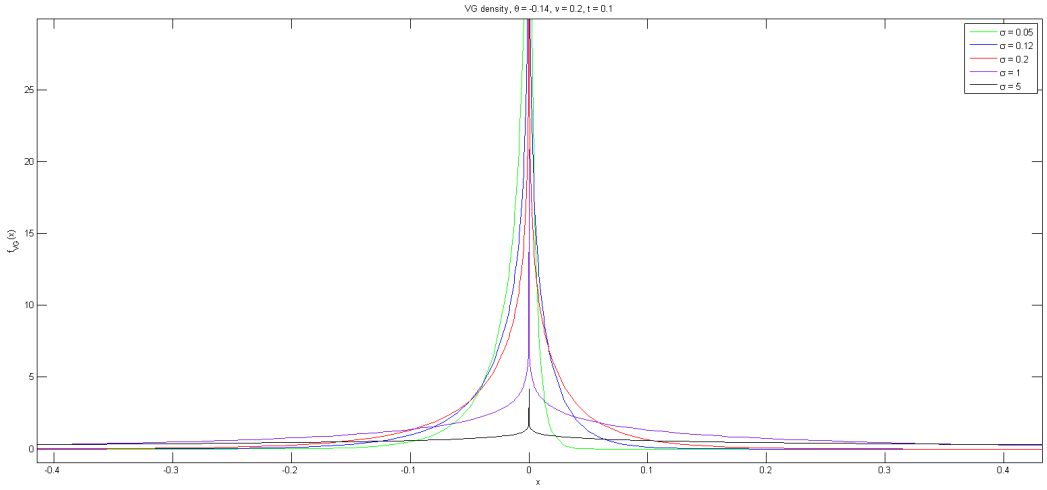

Figure 2-12: VG-density for different values of σ

2-5-4 CIR Model

The Cox-Ingersoll-Ross density can be used to describes a stochastic interest rate, or a stochatic volatility, as is done in the Heston model,

$$d\nu_t = \kappa(\theta - \nu_t)\,dt + \eta\sqrt{\nu_t}\,dW_t. \tag{2-62}$$

Here ν_t can be used to represent the volatility, as is done in the Heston model, but is also often used to model a stochastic interest rate. W_t is again Brownian motion. θ corresponds to the long run mean of the volatility over time. The drift term $\kappa(\theta-\nu_t)$ is positive for $\theta > \nu_t$ and negative for $\theta < \nu_t$, which causes a pull to θ, which is called reversion. The strength of the reversion is determined by the frequency parameter κ, and η is the volatility of the volatility.

As long as the Feller condition, $2\kappa\theta \geq \eta^2$, is met, the process will stay strictly positive. If the Feller condition is not met the process might reach zero, causing the density of ν_t, given ν_0 to have a sharp peak for ν_t approaching zero. In these cases the COS method converges very slowly.

See section 4-3 for numerical examples using the CIR-density.

2-5-5 Portfolio loss distribution

For a bank it is very important to manage the risks originated from its business activities, since the credit risk underlying the credit portfolio is often the largest risk in a bank.

For quantifying losses in credit portfolios one often looks at the Value at Risk (VaR), which is the maximum loss not exceeded with a given probability defined as the confidence level, over a given period of time.

The value at risk (VaR) of a portfolio at the confidence level α is given by the smallest number l such that the probability that the loss L exceeds l is at most $(1-\alpha)$:

$$\text{VaR}_\alpha = \inf\left\{l \in \mathbb{R} : P(L > l) \leq (1-\alpha)\right\} = \inf\left\{l \in \mathbb{R} : F_L(l) \geq \alpha\right\}, \tag{2-63}$$

where F_L is the cumulative loss distribution function.

The Vasicek model can be used to find an approximation to the loss distribution and the VaR. Under this model losses occur when an obliger defaults in a fixed time horizon.

In section 4-5 the COS method is used to approximate the cumulative distribution function of the loss function and calculate the VaR. The probability density function is build from a discrete set of events, resulting in a stepwise cumulative distribution function, which causes the COS method to suffer from the Gibbs phenomenon: if the loss distribution is very discontinuous, then relatively bigger errors appear around the discontinuous points [6]. We will show how the results from this method can be improved.

Chapter 3

Improvements for spectral methods

For smooth functions spectral methods yield extremely accurate approximations relatively fast, hence their popularity in applications such as numerical differential equations. However when functions are only piecewise smooth the Gibbs phenomenon, already introduced in chapter 2 occurs, resulting in oscillations near the jumps and only first order accuracy away from the jumps.

Much of the theory in this chapter can be applied to spectral methods in general, but this thesis focuses on the Fourier series in particular. See Appendix A-4 for some background information about spectral methods and the relation with other numerical methods.

One way to reduce the Gibbs phenomenon is to reproject the Fourier series on another set of basis-functions. For a piecewise smooth function one can project each smooth subinterval on another basis and then combine these parts to restore the function on the whole interval. These *reprojection methods* will be the subject of section 3-1.

The next section, 3-2, will deal with *mollifiers*, which are used to smoothen the sharp features of a function by convolving it with the mollifier, while still remaining close to the original non-smooth function.

Mollifiers are closely related with filters, which are the subject of section 3-3. The idea of filtering is to alter the expansion coefficients in such a way that they decay faster, because not only does a smooth function result in fast decaying Fourier-coefficients, the opposite statement is also true.

The rest of the thesis focuses on the methods mentioned above, but these are selections from a wide range of possibilities. Some of the alternatives will be discussed briefly in section 3-5.

3-1 Reprojection methods

Reprojection methods attempt to reduce the Gibbs phenomenon by projecting onto another basis. The first method we discuss is the Gegenbauer reconstruction, as developed in [14], [15], [17].

The Gegenbauer transform, just as the regular Fourier transform can be seen as a special case of the generalized Fourier transform, which is a projection on a set of orthogonal basis functions. The Gegenbauer projection uses the Gegenbauer-polynomials which are described in 3-1-3, as a basis.

Notation
Using the notation from [24] we use the overbar symbol to denote coefficients from the direct Gegenbauer method, $\bar{g}$, the tilde for the inverse polynomial reconstruction method (IPRM), which will be defined in the next section, $\tilde{g}$, whereas the hat is used for the Fourier transform, $\hat{f}$.

3-1-1 The direct Gegenbauer reconstruction method

This method will be referred to as the *direct* Gegenbauer reconstruction, as opposed to the *inverse* methods which will be discussed later on.

Let $f(x) \in L^2[-1,1]$ be a piecewise analytic function, and let $[a,b] \subset [-1,1]$ be a sub-interval on which f is analytic. Our goal is to approximate $f(x)$ in $[a,b]$ from its truncated Fourier series given by

$$S_N f(x) := \sum_{|n|\leq N} \hat{f}_n e^{i\pi n x}, \tag{3-1}$$

with the Fourier coefficients given by

$$\hat{f}_n = \frac{1}{2}\int_{-1}^{1} f(x) e^{-i\pi n x}\, dx. \tag{3-2}$$

Given this Fourier series approximation of a function, the idea is to calculate a certain amount of Gegenbauer-coefficients from the partial Fourier sum and reconstruct the function as a Gegenbauer-sum. That is, the representation

$$G_m^{\lambda}(f)(x) := \sum_{l=0}^{m} \bar{g}_l C_l^{\lambda}(x) \tag{3-3}$$

is sought, in which $\bar{g}_l^{\lambda}$ are the Gegenbauer coefficients and $C_l^{\lambda}(x)$ are the Gegenbauer polynomials for a parameter λ.

The coefficients are given by:

$$\bar{g}_l^{\lambda} = \frac{1}{h_l^{\lambda}} \int_{-1}^{1} (1-x^2)^{\lambda-\frac{1}{2}} S_N(f)(x) C_l^{\lambda}(x)\, dx, \tag{3-4}$$

where h_l^{λ} is given by (3-12).

For the reconstruction on the whole interval (with $[a,b] = [-1,1]$) there is an explicit formula for the coefficients, so we do not have to evaluate the integral in (3-4):

$$\bar{g}_l^{\lambda} = \delta_{0l}\hat{f}_0 + \Gamma(\lambda) i^l (l+\lambda) \sum_{0<|n|\leq N} J_{l+\lambda}(\pi n) (\frac{2}{\pi n})^{\lambda} \hat{f}_n. \tag{3-5}$$

This formula can be used to construct the Gegenbauer expansion directly from the Fourier coefficients. In [17], it is shown that for an analytic, but non-periodic $f(x)$ the series converges at exponential rate:

$$\max_{x\in[-1,1]} |f(x) - G_m^\lambda(f)(x)| \leq C \cdot m\sqrt{\frac{m+\lambda}{m}}\sqrt{\frac{(m+\lambda)^{m+2\lambda}}{m^m(2\lambda)^{2\lambda}}}r^m, \tag{3-6}$$

for some constants C, and $r < 1$.

As an example, the function $f(x) = x$ on $[-1, 1]$, which gives a sawtooth function if made periodic, can be reconstructed almost exactly, from just a few Fourier coefficients.

Reconstruction on a sub-interval

So far the Gegenbauer method can remove the Gibbs oscillations from non-periodic analytic functions, thus effectively handling a single discontinuity.

One can also create a Gegenbauer reconstruction of a function which is analytic on a subinterval $[a, b] \subseteq [-1, 1]$, using the Fourier-coefficients calculated over the whole domain. Using the linear transformation of $x \in [a, b]$ to $\xi \in [-1, 1]$,

$$\xi = -1 + \frac{2(x-a)}{b-a}, \tag{3-7}$$

we can apply the projection $G_m^\lambda(S_N f)(\xi(x))$ defined by (3-3):

$$\bar{g}^\lambda = \frac{1}{h_l^\lambda}\int_{-1}^{1}(1-\xi^2)^{\lambda-\frac{1}{2}}S_N(f)(x(\xi))C_l^\lambda(\xi)\,d\xi. \tag{3-8}$$

In this case the explicit formula (3-5) does not work, because the integration domains do not fully overlap. Instead the Gegenbauer coefficients can be obtained by numerically evaluating the integrals (3-8).

This evaluation is more costly, but the idea remains the same: the Gegenbauer coefficients are directly calculated, using the given truncated Fouries-expansion.

Given the locations of discontinuities, possibly by an edge detection method, we can reconstruct partially analytic functions. See for example figure 4-4.

3-1-2 Convergence

The error for the Gegenbauer reconstruction can be split into two parts:

$$f(x) - G_m^\lambda(S_N f)(\xi(x)) = f(x) - G_m^\lambda(f)(\xi(x)) + G_m^\lambda(f - S_N)(\xi(x)) \tag{3-9}$$

The first component converges exponentially, as in (3-6), whereas for fixed λ the second term only decays at approximately the rate $(m^2/N)^\lambda$. To achieve exponential decay in (3-9) both the degree m and the weight order λ have to be selected proportional to N. This however

inhibits the convergence rate in (3-6), so some optimum has to be found. In [15] it was shown that to ensure exponential convergence it is sufficient to select $\lambda = \gamma m$, with constant γ.

In general it is important to use the right amount of coefficients m: if m is too low the reconstruction is not accurate, but if m is too large, then reconstructed function will exhibit some of the Gibbs phenomena, since $\lim_{m\to\infty} \hat{f}(x) = S_N(f)(x)$. See figure 4-3. m has to have a value for which the truncation error and the regularization error both decay exponentially. Values that are close to optimal and often used in the literature, are $m = N/4$ and $m = N/5$.

With properly chosen parameters a piecewise analytic function can be restored with exponential accuracy. The method depends heavily on the order M of the polynomials and a Gegenbauer-parameter λ.

3-1-3 Gegenbauer Polynomials

The Gegenbauer polynomials $C_n^\lambda(x)$ are solutions to the Gegenbauer differential equations:

$$(1-x^2)y'' - 2(\lambda+1)xy' + n(n+2\lambda)y = 0 \tag{3-10}$$

The Gegenbauer polynomials are polynomials of order n which are orthogonal with respect to their weight function $(1-x^2)^{\lambda-\frac{1}{2}}$:

$$\int_{-1}^{1} (1-x^2)^{\lambda-\frac{1}{2}} C_k^\lambda(x) C_n^\lambda(x)\, dx = \delta_{k,n} h_n^\lambda, \tag{3-11}$$

$$h_n^\lambda = \pi^{\frac{1}{2}} C_n^\lambda(1) \frac{\Gamma(\lambda+\frac{1}{2})}{\Gamma(\lambda)(n+\lambda)}. \tag{3-12}$$

An explicit expression for the Gegenbauer polynomials is given by the Rodrigues' formula:

$$(1-x^2)^{\lambda-\frac{1}{2}} C_n^\lambda(x) = H(\lambda,n) \frac{d^n}{dx^n}[(1-x^2)^{n+\lambda-\frac{1}{2}}] \tag{3-13}$$

with $H(\lambda, n)$ given by

$$H(\lambda,n) = \frac{(-1)^n \Gamma(\lambda+\frac{1}{2})\Gamma(n+2\lambda)}{2^n n! \Gamma(2\lambda)\Gamma(n+\lambda+\frac{1}{2})}. \tag{3-14}$$

For calculating our projections it is useful to note that the Gegenbauer coefficients can also be expressed directly in terms of the Fourier coefficients, using the formula given in [17]:

$$\bar{g}^\lambda(l) = \delta_{0l}\hat{f}(0) + \Gamma(\lambda) i^l (l+\lambda) \sum_{0<|k|\le N} J_{l+\lambda}(\pi k)(\frac{2}{\pi k})^\lambda \hat{f}(k) \tag{3-15}$$

Note however, that for this formula to apply both expansions must use the same interval.

Here are some explicit expressions for the first couple of lower-order Gegenbauer polynomials:

$$
\begin{aligned}
C_0^\lambda(x) &= 1,\\
C_1^\lambda(x) &= 2\lambda x,\\
C_2^\lambda(x) &= 2\lambda(\lambda+1)x^2 - \lambda,\\
C_3^\lambda(x) &= \frac{4\lambda}{3}(\lambda+1)(\lambda+2)x^3 - 2\lambda(\lambda+1)x\lambda.
\end{aligned} \tag{3-16}
$$

Relations with Jacobi, Chebyshev and Legendre polynomials

The Chebyshev and Legendre polynomials are special cases of the Gegenbauer polynomials, which themselves are special cases of the Jacobi polynomials, also known as hypergeometric polynomials.
The Jacobi polynomials $P_n^{(\alpha,\beta)}$ are orthogonal with respect to the weight $(1-x)^\alpha(1+x)^\beta$ on the interval $[-1,1]$. More explicitly:

$$
\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta P_m^{(\alpha,\beta)}P_n^{(\alpha,\beta)}\,dx = \delta_{m,n}\frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1}\frac{\Gamma(n+\alpha+\beta)\Gamma(n+\beta+1)}{n!\Gamma(n+\alpha+\beta+1)}. \tag{3-17}
$$

If we take $\alpha = \beta = \lambda - \frac{1}{2}$ we get the Gegenbauer polynomials from equation (3-11).

Taking $\lambda = 0$ we get the Chebyshev polynomials of the first kind and with $\lambda = 1$ we get the Chebyshev polynomials of the second kind.

Another way to define the Chebyshev polynomials T_n of the first kind is through the identity

$$
T_n(\cos(\theta)) \equiv \cos(n\theta), \tag{3-18}
$$

which shows that a Chebyshev series for a periodic function are actually just Fourier cosine expansions with a change of variables [1].

The Gegenbauer polynomials, and thus also the Chebyshev and Legendre polynomials inherit all the symmetry relations, explicit formulas for derivatives, recurrency relations, asymptotics and generating functions from the Jacobi polynomials which are well known in the field of special functions.

Legendre polynomials

If we take $\lambda = \frac{1}{2}$, we get the *Legendre polynomials*. Since the weight function in (3-11) is equal to 1, these polynomials satisfy an orthogonality relation:

$$
\int_{-1}^{1} P_n(x)P_m(x)dx = \delta_{mn}\frac{2}{2n+1}. \tag{3-19}
$$

From here on we use the *normalized* Legendre polynomials:

$$
\bar{P}_n(x) := P_n\sqrt{\frac{2}{2n+1}}, \tag{3-20}
$$

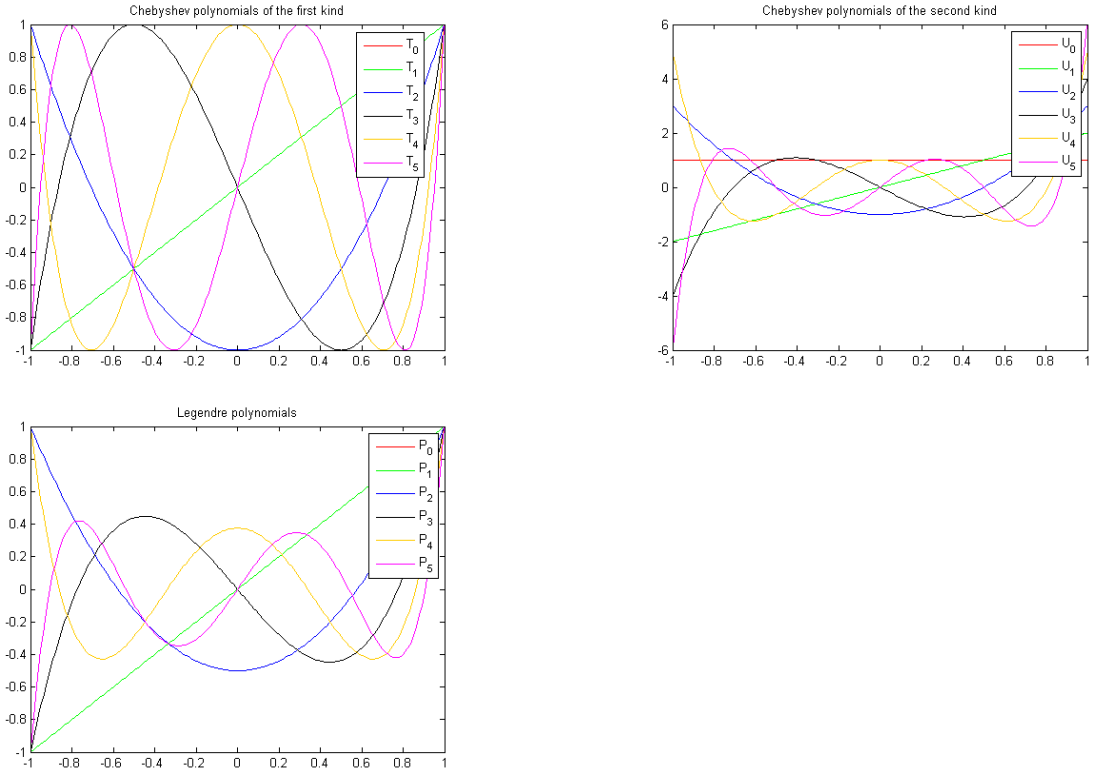

Figure 3-1: The Chebyshev polynomials of the first ($\lambda = 0$) and second kind ($\lambda = 1$) and the Legendre polynomials ($\lambda = \frac{1}{2}$)

For suitable $f(x)$ on $[-1,1]$ we have the *generalized Fourier series*

$$f(x) = \sum_{0}^{\infty} a_n \bar{P}_n(x), \tag{3-21}$$

with the expansion coefficients given by

$$a_n = \int_{-1}^{1} f(x)\bar{P}_n(x)\,dx. \tag{3-22}$$

Since the Legendre polynomials are orthogonal, we can calculate the integral of a product of two functions by multiplying its Legendre-coefficients:

$$\begin{aligned} \int_{-1}^{1} f(x)g(x)\,dx &= \int_{-1}^{1} \left(\sum_{m=1}^{M} a_m \bar{P}_m(x)\right)\left(\sum_{n=1}^{N} b_n \bar{P}_n(x)\right) dx \\ &= \sum_{k=1}^{\min(M,N)} a_k b_k. \end{aligned} \tag{3-23}$$

This is similar to the calculation in the COS method, in which the Fourier-coefficients are used to calculate the integral of the product of two functions. This works particularly fast if one of the series consists of only a couple of terms, or the coefficients of at least of the series decay exponentially.

More properties of the Gegenbauer, Chebyshev and Legendre polynomials are described in [1].

3-1-4 Inverse reprojection methods

If we interchange the sum and integral in formula (3-1) and (3-8) we are calculating the Gegenbauer coefficients for each Fourier mode, which is in fact calculating a basis transformation.

This is guaranteed to work because in the infinite case the Gegenbauer space can be mapped onto the Fourier space and vice-versa, since both are orthogonal bases for the same space.

The inverse polynomial reconstruction method (IPRM), developed in [32], calculates the coefficients by solving a system of equations instead of calculating the coefficients directly.

If we denote the matrix of Gegenbauer coefficients of each Fourier mode by W_{kl}:

$$W_{kl} = \frac{1}{2}\int_{-1}^{1} G_l^{\lambda}(x) e^{-ik\pi x}\, dx, \tag{3-24}$$

we must have

$$\sum_{l=0}^{m} W_{kl} = \hat{f}_k \tag{3-25}$$

and we can find the Gegenbauer coefficients by solving

$$W\tilde{g} = \hat{f}. \tag{3-26}$$

Originally the Gegenbauer polynomials were used as reconstruction basis, but later this method was generalized toward using any set of basis functions [22].

One of the advantages of this method is that the Gegenbauer parameter λ does not play the same role anymore as it does in the original Gegenbauer method. We can now use a fixed λ without losing convergence. It only affects the condition number of the transformation matrix W, which is best for small λ.

However, even for small λ the condition number of W grows exponentially with the number of Fourier coefficients N. The numerical convergence is described detail in [23]. The method and its numerical aspects are investigated further by Pasquetti in [27]. More numerical results for this method are given by Greene in [18].

For polynomial testfunctions this method finds the exact solution, see for example figure 4-5.

3-1-5 Modified IPRM

The IPRM, discussed in the previous section, does not directly project the function (its trunctated Fourier series) on the Gegenbauer-space, but calculates a transformation matrix between the 2 spaces and then restores the function by solving the matrix equation.

A drawback is that the condition number of the underlying $N \times N$ transformation matrix shows exponential (in the number of fourier-coefficients) growth, it grows approximately like $O(e^{0.4N})$, which quickly leads to ill-conditioning.

For this reason, IPRM fails to converge for example in the case of a meromorphic [1] function, whose poles are located sufficiently close to the interval where the function is defined. This happens because the function's Legendre series does not converge fast enough to mitigate the exponential growth of the condition number.

See for example figure 3-2, showing the function $f(x) = \exp(\sin(3x) + \cos(x))$ in the complex plane. Here we have $f(x) \in \mathbb{C}^{\infty}[a, b]$, but the Gegenbauer expansion fails to converge because of the behaviour just next to the real axis [3].

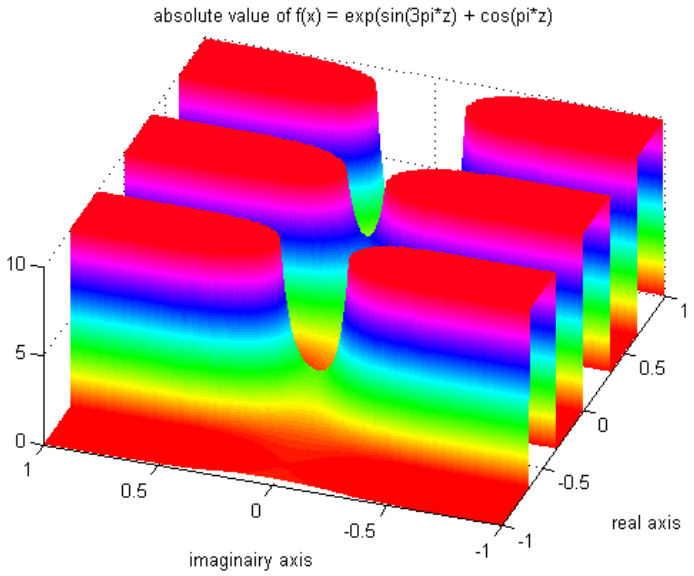

Figure 3-2: This function, although analytic and smooth on the real axis causes problems with the Gegenbauer expansion, because of the behaviour in the complex plane just away from the real axis

Recent research has shown that we can dramatically improve the conditioning of the system by using more Fourier coefficients: to reconstruct the function as a polynomial of order $(n-1)$, we use m Fourier-coefficients, with m (much) larger than n. This means that the matrix equation (3-26) is replaced by a least-squares minimization problem of the form

$$\min_{\bar{g} \in \mathbb{R}^n} ||W\bar{g} - \hat{f}||. \tag{3-27}$$

In [21] it is proven that if $m > \alpha N^2$, the condition number of the underlying least squares problem does not exceed $\sqrt{\frac{m}{m-\alpha_0 N^2}}$, with $\alpha_0 = 4\sqrt{2}/\pi^2 \approx 0.573$. This method, the *modified IPRM*, to resolve the problem with the high condition number is described in detail in [21].

The modified IPRM uses n Fourier-coefficients to create a reconstruction of m Legendre polynomials, $m \geq n$ and for $m \sim n^2$ the condition number of the transformation matrix A is limited close to 1.

[1] A meromorphic function is a single-valued function that is analytic in all but possibly a discrete subset of its domain, and at those singularities it must go to infinity like a polynomial (i.e., these exceptional points must be poles and not essential singularities).

The success of both the projection and direct-inverse methods is that they reconstruct the spectral approximation with high accuracy in sufficiently smooth regions or in the region sufficiently far from the discontinuity. However, the crucial point is the convergence behavior near or up to the discontinuity points. Fourier filtering methods and spectral mollification methods intrinsically yield only $O(1)$ convergence near the singular points although one can obtain fast convergence in the smooth region from the given Fourier data.

This is because these methods still use the Fourier space where the original spectral data is obtained. The Gegenbauer reconstruction method, and IPRM utilize polynomials, to reconstruct the function represented with the Fourier data. By seeking a reconstruction in a different space, these methods obtain fast convergence even near the discontinuity.

These methods are different from the filtering or mollification methods as they overcome the $O(1)$ convergence up to the discontinuity if the exact location of the discontinuity is known.

However, the accurate reconstruction with these methods near or at singularities is very sensitive to the numerical conditioning of the projection or inversion operators. As the number of the given Fourier coefficients increases, the accuracy in the interior region away from the singularity is highly enhanced, but the accuracy near the singular points decreases if N is larger than a certain value.

3-1-6 Possible other improvements IPRM

Other adjustments to the IPRM that have been suggested to resolve the conditioning problem are the Freud projection and the Truncated IPRM.

Freud projection

Although we did not end up using the Freud projection, the article describing it gives some additional insight about the failed convergence of the Gegenbauer reprojection in specific cases. In the article three requirements for a reprojection basis are formulated, which leads to a so called robust Gibbs' complementary basis [13]. The Gegenbauer-polynomials only satisfy the first two of them:

1. For a function, analytic on the interval $[-1, 1]$, the function's expansion in the orthogonal reprojection must be exponentially convergent.
2. The projection of the high modes in the original basis (Fourier or Chebyshev) on the low modes in the new basis is exponentially small.
3. As the order of the original projection N increases, the weight function of the reprojection bases converges to a weight function whose associated basis satisfies requirement 1.

Because in the Gegenbauer projection parameters m and λ must both be proportional to N, the weight-function becomes increasingly narrow and the polynomial amplitude rapidly increases. As N approaches infinity, the Gegenbauer projection approaches the power series expansion, which does not satisfy the additional requirement and is guaranteed to converge

only when the underlying function is analytic in the whole complex area surrounding the interval: $|z - \frac{b+a}{2}| \leq \frac{b-a}{2}$

As an example of a robust Gibbs complement Gelb and Tanner use a basis resulting from Freud weight functions: the Gegenbauer weight $w_\alpha(x) = (1-x^2)^{\alpha-\frac{1}{2}}$ is replaced by the Freud weight $w_m(x) = e^{-cx^{2m}}$.

We did not implement the Freud reprojection, since many of the numerical round-off problems are already resolved in another way by using the modified IPRM. Furthermore, the weighting function implies that the Freud-basis-functions are not orthogonal in itself, so we cannot use this orthogonality to formulate a fast pricing formula similar to the COS-pricing formula.

Truncated IPRM

Jung and Shizgal propose a truncation method with Gaussian elimination in order to prevent the exponential growth of the condition number of the transformation matrix W between Fourier and Legendre basisfunctions [23].

3-1-7 Edge detection

The polynomial reconstruction methods discussed so far focus on recovering a function from its Fourier expansion on an interval on which the function is continuous but non-periodic. Given the locations of the discontinuities it is fairly straightforward to extend those methods to recover piecewise-continuous functions. For most of the densities and pricing functions used in the examples here, those locations are known, but in general this is not the case, and then we need some edge-detection methods.

In the Gegenbauer-reconstruction and IPRM implementations edge detection with a linear kernel was used, which gives reasonable results on the test functions used so far.

There is a lot of literature available on variouse edge detection methods, see for example [10] [11] [12] [12] [34].

We start out with partial Fourier sum, in sin/cos notation because this illustrates the ideas better:

$$S_N(f)(x) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(k\pi x) + b_k \sin(k\pi x). \tag{3-28}$$

The *generalized conjugate partial sum* is defined as:

$$\tilde{S}_N^\sigma(f)(x) = \sum_{k=1}^{N} \sigma\left(\frac{k}{N}\right)(a_k \sin(k\pi x) - b_k \cos(k\pi x)). \tag{3-29}$$

If we take $\sigma(x) \equiv 1$ we get the standard conjugate sum, $\tilde{S}_N(f)(x)$. This standard conjugate sum, multiplied by a factor, $\frac{-\pi}{\log N}\tilde{S}_N(f)(x)$ is known to convert to the jump function, defined by

$$(f)(x) := f(x+) - f(x-), \tag{3-30}$$

The convergence is however slow, at the rate of order $1/\log N$. This convergence can be accelerated by using a concentration kernel $\sigma = \sigma(k/N)$.

To be admissible, in the sense that the corresponding generalized conjugate sum satisfies the concentration property

$$\tilde{S}_N^{\sigma}(f)(x) \to (f)(x), \text{ for } N \to \infty, \tag{3-31}$$

the concentration kernel can be an arbitrary nondecreasing $\sigma \in C^2[0, \pi]$, satisfying

$$\int_{1/n}^{1} \frac{\sigma(x)}{x}\, dx \to -\pi. \tag{3-32}$$

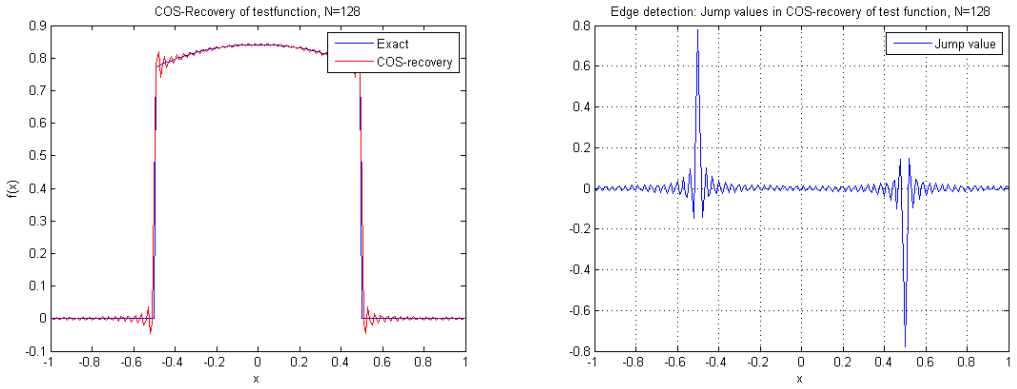

Figure 3-3: *Left:* Recovery of testfunction $f(x) = \sin(\cos(x))$, for $x \in [-0.5, 0.5]$, 0 elsewhere, using Fourier cosine series. *Right:* Edge detection, using linear concentration kernel

An example of a second degree polynomial concentration factor is [12]:

$$\sigma^{p_2}(k, N) := -\frac{2\pi n^2}{N(N+1)}. \tag{3-33}$$

In [11] and [10] examples of improved edge detection methods based on enhanced concentration factors are given. But in our case, for the polynomial reconstruction we only need to know the location of the jumps, since the height of the function on the left and the right side of these jumps follows naturally from the reconstruction.

3-2 Mollifiers

Mollifiers work in the time domain (as opposed to filters which work in the Fourier-domain). We discuss them first because they are more intuitive and many of the ideas carry over to filters. By convolving a function with a suitable mollifier the function's sharp features are smoothed, while still remaining close to the original non-smooth function.

Definition 3. *A mollifier is a smooth function Φ on $\mathbb{R}^n$ satisfying the following 3 requirements:*

1. *it is compactly supported;*
2. $\int \Phi(x)\,dx = 1$*;*
3. $\lim_{\epsilon\to 0} \Phi_\epsilon = \lim_{\epsilon\to 0} \epsilon^{-n}\Phi(x/\epsilon) = \delta(x)$*;*

As an example, we can choose a weighted Gegenbauer polynomial of arbitrary degree p, (see (6.12) [34]). Another frequently used example is:

$$\Phi(x) = \begin{cases} k\exp\left(\frac{-1}{1-|x|.^2}\right) & \text{for } |x| < 1, \\ 0 & \text{for } |x| \geq 1, \end{cases} \tag{3-34}$$

where k is a constant such that condition 2 from definition 3 is satisfied.

The *mollification* of a function is the convolution of the function with the mollifier, so if we insert a Fourier partial sum, we get

$$\Phi(x) * S_N(f)(x) = \int_{\mathbb{R}} \Phi(x-y)S_N(f)(y)\,dy. \tag{3-35}$$

Given a mollifier that satisfies the requirements from 3, one can create a family of molifiers by dilating:

$$\Phi_\theta(x) := \frac{1}{\theta}\Phi\left(\frac{x}{\theta}\right). \tag{3-36}$$

If we let θ tend to zero, convolving with this mollifier returns the original function.

Mollifying a function at a jump location would smoothen out this discontinuity as well, so mollifying works best on interior points (interior on the smooth subinterval). This naturally brings us the idea of adaptive mollifiers, which adjust their width depending on how far from the edge they are.

3-2-1 Adaptive mollifiers

The idea of adaptive mollifiers is to make the dilation parameter $\theta = \theta(x)$ as large as possible while maintaining smoothness of $\rho(x-\theta)f(.)$. As an example, we take a two-parameter family of spectral mollifiers, defined by

$$\Phi_{p,\theta}(x) = \frac{1}{\theta}\rho\left(\frac{x}{\theta}\right) D_p\left(\frac{x}{\theta}\right), \tag{3-37}$$

where $\rho(.)$ is an arbitrary C_0^∞ function which localizes the p-degree Dirichlet kernel [1] $D_p(y) := (sin(p+1/2)y)/(2\pi sin(y/2))$.

Letting $d(x)$ denote the distance to the nearest edge, we can set $\theta(x) \sim d(x)$ so that $\Phi_{p,\theta} * S_N(f)(x)$ incorporates the largest smooth neighborhood around x.

The degree p, which is responsible for the overall high accuracy by enforcing an intricate cancellation also has to be chosen carefully. Analysis in [34] leads to an optimal choice of an adaptive degree of order $p = p(x) \sim d(x)N$.

Tadmor describes a family of root-exponential accurate mollifiers:

$$\Phi_{p,\delta} = \frac{1}{\delta}\rho_2\left(\frac{x}{\delta}\right) D_p\left(\frac{x}{\delta}\right), \rho_2 = \exp\left(\frac{cx^2}{x^2-\pi^2}\right) 1_{(-\pi,\pi)}(x), \;\; c > 0. \tag{3-38}$$

An error bound, which is given by

$$|\Phi_{p,\theta} * S_N f(x) - f(x)| \leq Const \cdot d(x)N \cdot e^{-\eta\sqrt{d(x)N}}, \tag{3-39}$$

shows that the adaptive mollifier is exponentially accurate at all x,except for the immediate neighborhood of the jumps, where $d(x) \sim 1/N$.

3-2-2 Error analysis

We can decompose the error into three terms, one of which vanishes by orthogonoality:

$$\begin{aligned} E(N,p,\theta;f(x)) &= E(N,p,\theta) := \Phi_{p,\theta} * S_N f(x) - f(x) \\ &= (f * \Phi_{p,\theta} - f) + (S_N f - f) * (\Phi_{p,\theta} - S_N\Phi_{p,\theta}) + (S_N f - f) * S_N\Phi_{p,\theta} \\ &= (f * \Phi_{p,\theta} - f) + (S_N f - f) * (\Phi_{p,\theta} - S_N\Phi_{p,\theta}) \\ &=: R(N,p,\theta + T(N,p,\theta) \end{aligned} \tag{3-40}$$

The resulting two terms are referred to as the regularization and the truncation errors. Detailed analysis in [35] results in upper bounds for both these errors:

$$|R(N,p,\theta)| \leq Const \cdot ||\rho||_{C^s} ||f^{(s)}||_{L^\infty_{loc}} \cdot p\left(\frac{2}{p}\right)^s, \tag{3-41}$$

$$|T(N,p,\theta)| \leq Const \cdot ||\rho||_{C^s} \cdot N\left(\frac{1+p}{N\theta}\right)^{s+1}. \tag{3-42}$$

[1] See appendix A-5 for some more information about the Dirichlet kernel.

One can use different orders of degrees of smoothness for the truncation and the regularization, say r and s, so combining both errors yields:

$$|E(N,p,\theta;f(x)))| \leq Const \cdot \left[||\rho||_{C^r} \cdot N \left(\frac{1+p}{N\theta} \right)^{r+1} + ||\rho||_{C^s} \cdot p \left(\frac{2}{p} \right)^{s} ||f^{(s)}||_{L^{\infty}_{loc}} \right]. \quad (3\text{-}43)$$

Mollification can give really good results for reconstructing functions from their Fourier-coefficients. With *adaptive mollifiers*, which take the locations of the discontinuities into account one can reach exponential accuracy.

Since mollification takes place in the time domain it is not as easy to implement in the COS method. There is however an equivalence relation with filters: for every filter there is an associated mollifier and vice-versa, which still makes the theory significant for our subject.

3-3 Filters

The idea of filtering is to alter the expansion coefficients in such a way that they decay faster. If the filter is well chosen this will improve the convergence rate away from the discontinuity. Note that since the approximation will be smoothened, convergence very close to the jump will not improve.

Filtering takes place in the Fourier-domain and therefore works very well in combination with the COS-pricing formula and does this without adding significant computational costs. The technique can be used in 1D, but also in higher dimensional problems.

We define a family of general filters $\sigma(.)$ in Fourier-space:

$$S_N^{\sigma} f(x) := \sum_{|n|\leq N} \sigma(n,N)\hat{f}_n e^{inx}, \tag{3-44}$$

where σ must be a real and even function, with rapid decay.

Truncation of a spectral series can be seen as a special case of filtering in which all the filter weights are either one or zero:

$$\sigma_{trunc}(n,N) = \begin{cases} 1, & \text{if } |n| \leq N, \\ 0, & \text{otherwise.} \end{cases} \tag{3-45}$$

but the decay is not smooth at all, and some form of smoothness is what we need to resolve the Gibbs oscillations.

Note that filtering does not affect the total mass of the resulting approximation (which we want to be 1 for a probability distribution), since the first coefficient is never altered.

The accuracy of a filter is described by its *order*:

Definition 4. *A filter is of order p if for any fixed $n > 0$*

$$\sigma\left(\frac{n}{N}\right) \sim O\left(\frac{1}{N^p}\right) \text{ for } N \to \infty. \tag{3-46}$$

Equivalently a filter is p-order accurate if it satisfies

$$\sigma^{(n)}(0) = \delta_{n0} \text{ for } n = 0, 1, .., p. \tag{3-47}$$

The order determines the rate at which the error produced by the filtering decays. For a function that is already analytic, so that its spectral series is converging exponentially, one needs a filter of order p to make sure the filtered spectral series converges at order p, that is that the errors produced by the filtering are $O(1/N^p)$. Therefore it is desirable to use a filter of high order to keep the advantages of having a high order spectral method when computing smooth solutions.

For a function that is not smooth, so that its unfiltered spectral series converge only at algebraic rate, Vandeven [1] has shown that one can recover the true solution with $O(1/N^p)$ accuracy at any fixed point away from the discontinuities by using a filter of order p.

3-3-1 Non-adaptive filters

Exponential filter

To be truly p-th order, a filter $\sigma(\omega)$ must have also have a $(p-1)$th order zero at $\omega = 1$ [1]. Most filters however satisfy this condition only approximately. In the exponential filter for example the constant is chosen so that the filter reaches zero within machine accuracy:

$$\sigma^{exponential}(\omega) = e^{(\ln \epsilon_m)\omega^p}, \tag{3-48}$$

where p must be even and ϵ_m represents the machine zero and the argument ω is used to denote the relative frequency which in our discrete case is n/N.

Vandeven Filter

The Vandeven filter which satisfies all the conditions in his theorem is given by:

$$\sigma^{Vandeven}(\omega) = 1 - \frac{(2p-1)!}{(p-1)!}\int_0^{|\omega|} t^{p-1}(1-t)^{p-1}\,dt, \tag{3-49}$$

Unfortunately this filter is rather difficult to implement exactly, so that in practice one still ends up using an approximation.

Erfc-Log Filter

Boyd [1] showed that the Vandeven's filter can be approximated quite accurately by an analytical function that satisfies the conditions, this is the "Erfc-Log"-filter:

$$\sigma^{ErfcLog}(\omega) = \frac{1}{2}erfc\left(2\sqrt{p}(|\omega| - \frac{1}{2}\sqrt{\frac{-ln(1-4(|\omega|-1/2)^2)}{4(|\omega|-1/2)^2}}\right). \tag{3-50}$$

Here erfc$(\cdot)$ is the complimentary Gauss error function, erfc$(x) = 1 - $ erf(x), with erf$(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}\,dy$.

Figure 3-4 shows the shape of the filters in this section, for various values of the filter order p. The difference between the Erf-Log and Vandeven filter is very small.

3-3-2 Relation between filters and mollifiers

Filters and mollifiers are closely related, since multiplication in the Fourier-space corresponds with convolution in physical space:

$$S_N^\sigma f(x) := \sum_{|n|\leq N} \sigma(n,N)\hat{f}_n e^{inx} \equiv \Phi * S_N f(x) = \int_{-\pi}^{\pi} \Phi(y) S_N f(x-y)\,dy. \tag{3-51}$$

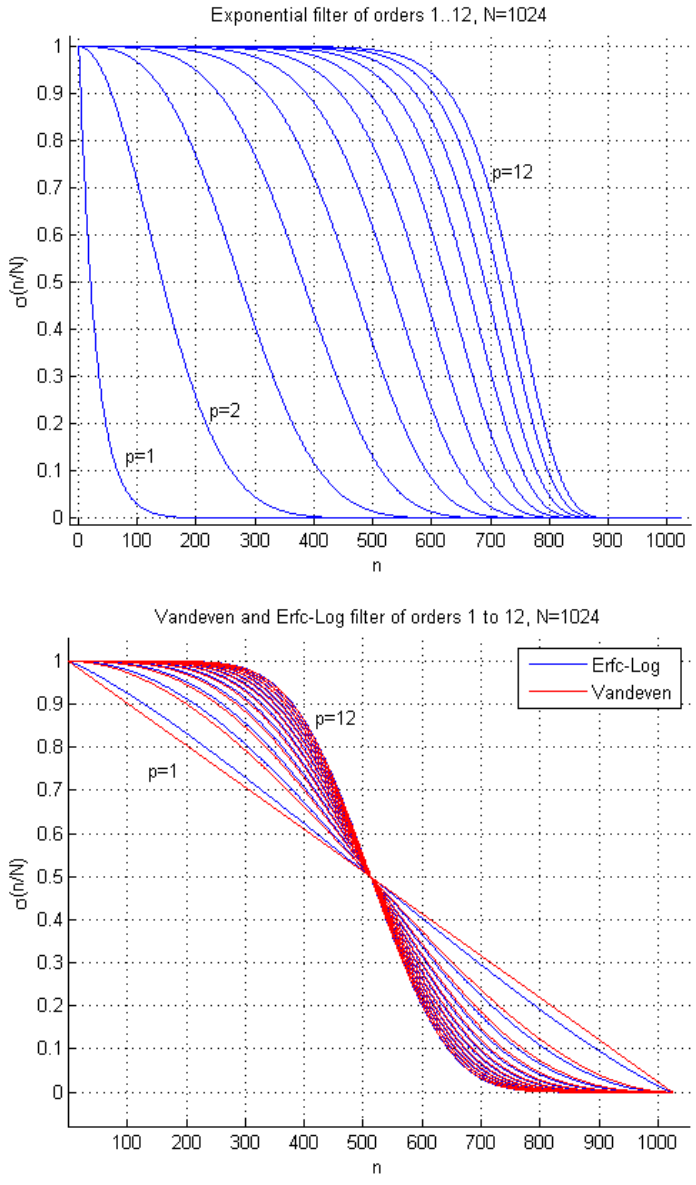

Figure 3-4: *Top:* Exponential filters of orders 1 to 12. *Bottom:* Erfc-Log and Vandeven filters of orders 1 to 12.

Therefore for a filter $\sigma(n, N)$ the associated mollifier is defined as:

$$\Phi^{\sigma}(x) := \frac{1}{2\pi} \sum_{-\infty}^{\infty} \sigma(n, N) e^{inx}. \tag{3-52}$$

The reversed statement: for a mollifier defined in the physical space its associated filter's samples are the mollifiers's Fourier coefficients, $\sigma(n, N) := \hat{\Phi}_n$.

It follows that if the filter σ is p-th order accurate, its associated mollifier, here denoted by Φ_N^{σ} satisfies the moment condition to order p. Thus the smoother σ, the better Φ_N^{σ} is localized [34].

Now that we know that filters and mollifiers are equivalent, we can use the idea of adaptive mollifiers and apply it to filters.

3-3-3 Adaptive filters

The optimal filter order p for a function with discontinuities is an increasing function of the distance to the nearest discontinuity. Denoting this distance from x to the nearest discontinuity by $d(x)$, the optimal filter order for recovering the function f at point x is

$$p_{optimal}(x) = 1 + N \frac{d(x)}{2\pi}. \tag{3-53}$$

The idea of adaptive filtering is to vary the filter order so that it is close to this optimal value. Tadmor [34] describes adaptive filters recovering root-exponential accuracy:

$$S_N^{\sigma_{p,\delta}} = \sum_{|n| \leq N} \sigma_{p,\delta} \left(\frac{|n|}{N} \right) \hat{f}_n e^{inx}, \tag{3-54}$$

where δ depends on the analytic neighborhood enclosing x:

$$\delta_x = \sqrt{\theta d(x) N}. \tag{3-55}$$

Tadmor [34] gives an error decomposition into the usual truncation and regularization terms and proves that this adaptive filter recovers the point values $f(x)$ with root-exponential accuracy:

$$|S_N^{\sigma_{p,\delta}} f(x) - f(x)| \leq d_x N e^{-\eta \sqrt{d_x N}}, \tag{3-56}$$

in which the constant $\eta = \eta_{\sigma,f}$ depends on the analyticity properties of σ and f.

3-4 The use in the COS method

Since mollifiers work in the time domain, implementation in the COS method would require reconstruction to the time-domain. This is a step we would like to avoid, since it is computationally expensive.

Non-adaptive filters work strictly in the Fourier domain and therefore can be used directly in the COS pricing formula (2-41).

3-4-1 Pricing with reprojection methods

Any method that can be used for recovery can in principle be used for pricing, since one can just calculate the prices from the recovered functions, but this approach is expensive. One of the reasons the COS method is so fast is because the pricing formula works directly with the coefficients, without recovery of the functions.

With reprojection methods we can use a formula similar to the COS pricing formula if we use a reconstruction basis which is orthogonal. This is the case for the Legendre polynomials, in which case the product of the density and value function can again be calculated by multiplying the coefficients as in equation (3-23).

In fact in practice the payoff function is almost always known, so we can calculate the Legendre coefficients of the payoff directly, which is easier than using the COS coefficients. So we only have to calculate the Legendre coefficients of the density function by a reprojection algorithm, such as the IPRM or modified IPRM.

3-4-2 Pricing with filters

The big advantage of filters is that they are easily implemented in the COS pricing formula. We can basically just replace the coefficients and replace them with a filtered version in the pricing formula (2-41):

$$v(x,t_0) \approx e^{-r\Delta t} {\sum_{n=0}^{N-1}}' \sigma\left(\frac{n}{N}\right) \mathrm{Re}\left\{\varphi(\frac{n\pi}{b-a}, x, \Delta t) e^{-in\pi\frac{a}{b-a}})\right\} V_n. \tag{3-57}$$

Here σ can be any non-adaptive filter. For processes with independent increments we can apply the filter to 2-43.

Unfortunately, the same does not work for adaptive filters, because if we vary the coefficients depending on the position in the interval we cannot use the orthogonality relation anymore.

3-5 Alternative methods

3-5-1 Subtracting Discontinuities

Perhaps one of the most intuitive ways to increase smoothness is subtracting discontinuities. When a function is for example the sum of a smooth function and a step function one can subtract the step function, whose Fourier-coefficients can be determined analytically, focus on recovering the remaining smooth part and add the step function again in the reconstruction later on. This can be done with jumps, but the same idea can be applied to higher order discontinuities until the Fourier coefficients of the smoothened remainder decay fast enough. This method is described in more detail in for example [4].

This method is somewhat complicated and expensive to implement in combination with the COS method, because it involves a transformation to the time-domain, which makes it harder to use directly in combination with the COS-pricing formula (2-41).

When one can assume a function is having a certain shape, which is almost always the case for pricing formulas and often the case for the probability densities, the techniques in this method might be useful.

3-5-2 Digital Total Variation Filtering

Digital Total Variation (DTV) filtering is a physical space filtering process that can be applied to data on arbitrary grids. Originally DTV filters were designed to remove noise from images within the accuracy of which the human eye can detect [2].

In [29] it is shown that DTV filters can also be used as a post-processing method for pseudo-spectral approximations. The article concludes that the DTV filter as a post-processing for pseudospectral methods is a very good choice when other existing methods, such as the Gegenbauer Reconstruction procedure, either fail, are cumbersome to apply, or are too computationally expensive.

DTV works strictly in the time-domain, which makes it unsuitable for fast pricing using the COS method.

3-5-3 Rational reconstruction: Padé approximations

While all the methods described in section 3-1 focus on polynomial reprojection bases, Padé methods use rational functions instead of simple polynomials for the approximation. Rational functions are often better suited for approximating functions with steep gradients or discontinuities.

With no knowledge of the singularity, Padé reconstruction successfully recovers a nonoscillatory solution with a reduced (but still existent) overshoot at the singularity. Numerical results show the successful mitigation of the Gibbs oscillations away from the singularity with spectral convergence, as the grids are refined and the degree of the denominator is increased [26].

Combined with knowledge of the location of discontinuities Fourier Padé approximations can restore spectral accuracy up to those discontinuities [3].

3-5-4 Hybrid methods

The idea of hybrid methods is to use a polynomial reconstruction only where needed and filtering elsewhere. This way one can reach spectral accuracy up to the discontinuities. The method is described in detail in [9], concluding that it gives good results with relatively low computational costs in suitable situations.

However, it is complicated to combine this hybrid approach with the COS pricing formula, since the decision on where to use which method (filter or polynomial reconstruction) is based on the distance to the nearest discontinuity in a similar manner as with adaptive filters and mollifiers, which are described in sections 3-3 and 3-2.

3-6 Comparison of the available methods

Any method that can be used for recovery can in principle be used for pricing, since one can just calculate the prices from the recovered functions. This approach is very natural, but quite expensive. One of the reasons the COS method is so fast is because it allows pricing without actually calculating the density function recovery: the pricing formula works directly in the Fourier domain.

Table 3-1 summarizes the results of this chapter, the main criteria are accuracy and speed/-computational cost in combination with the COS method for pricing.

Table 3-1: Improvements for spectral methods

Mollifiers	Relatively easy to implement, works in time domain, therefore relatively expensive COS pricing
Filters	Easy to implement, very accurate away from jumps, works in frequency domain, therefore fast COS pricing
Adaptive Filters	Somewhat harder to implement, improves accuracy of filters close to jumps. Adaptive parts works in time domain, therefore relatively expensive COS pricing
Reprojection methods	Relatively hard to implement - very accurate for simple functions, lots of terms needed for general functions, computationally expensive. Numerical tability issues, parameters must be carefully chosen
Inverse reprojection methods	Resolves stability issues and simplifies parameter choice of Reprojection methods. Allows for use of orthogonal basis, which can be used for relatively fast pricing. Disadvantage: dependency between all subdomains
Padé methods	Relatively hard to implement, very accurate function recovery, standard basis is not orthogonal, which makes pricing a lot more difficult
Hybrid methods	potentially interesting for recovery, but hard to implement and unsuitable for COS pricing

Chapter 4

Numerical examples

4-1 Recovery of test functions

4-1-1 Recovery using Gegenbauer-reconstruction and IPRM

Step functions

Throughout this chapter N is used to denote the number of Fourier coefficients and m will be the polynomial order in the reconstruction, thus $m+1$ the number Gegenbauer coefficients. Later on, with the IPRM method, the number of Gegenbauer coefficients can be varied for each subinterval, this will be denoted as $m = [1, 1, 1]$ for using polynomial order 1 on each of the 3 subintervals.Calculations were performed multiple times to get accurate cpu-time estimates.

The first test function is a uniform density function, a rectangle, defined on $[-1, 1]$, of the form $f(x) = \frac{1}{2L}$ for $x \in [-L, L]$. Here $L = 0.6$ is chosen.

Figure 4-1 shows the Gegenbauer and the Fourier reconstruction with $N = 10$ and $m = 2$. We see that that Gegenbauer reconstruction looks good on the middle part, but at the side parts, the number of Fourier coeffients is a little too low, causing an error. Figure 4-2 shows an improved version, and if we increase N a little more, $N = 32$ for example, the Gegenbauer reconstruction will match the original function almost up to machine precision.

What happens if we choose m too high can be seen in figure 4-3, eventually the polynomial reconstruction will converge to the partial sum, including the Gibbs phenomenon.

Figure 4-4 shows another example of a successful Gegenbauer reconstruction with more discontinuities.

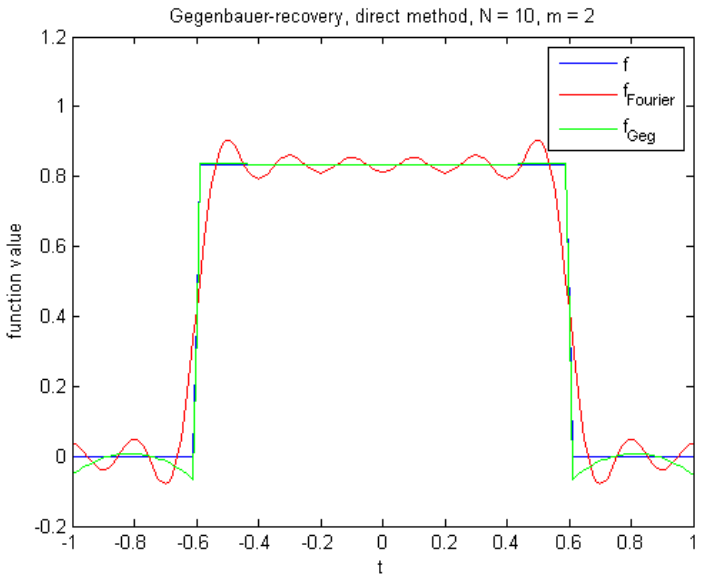

Figure 4-1: Gegenbauer reconstruction: not enough Fourier coefficients.

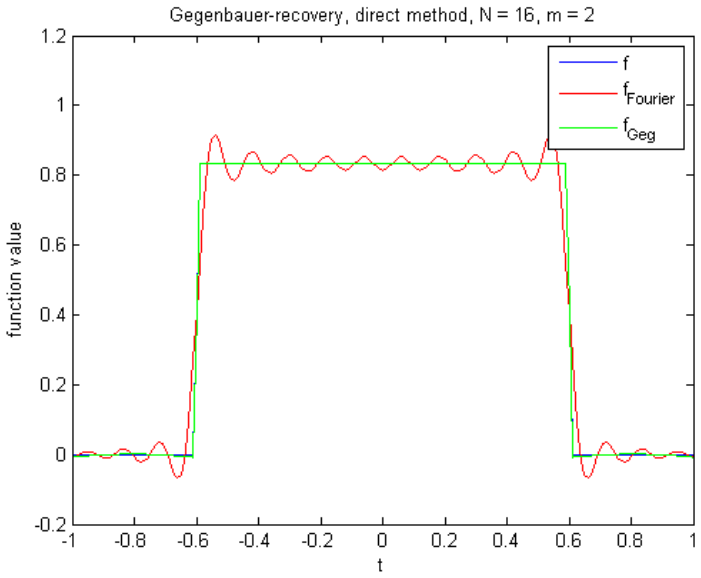

Figure 4-2: Gegenbauer reconstruction: if the parameters are well chosen the Gegenbauer reprojection matches the original function for a fairly low number of coefficients

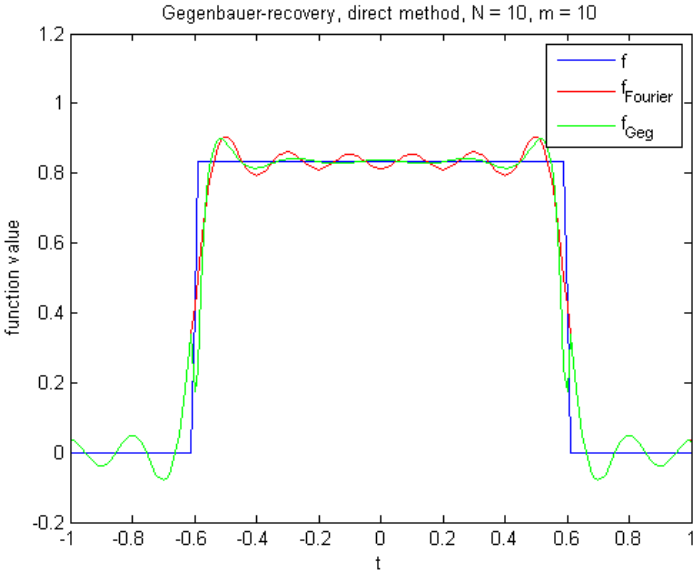

Figure 4-3: If we choose m too high the Gegenbauer reconstruction will mimic the partial Fourier sum, including the Gibbs phenomenon

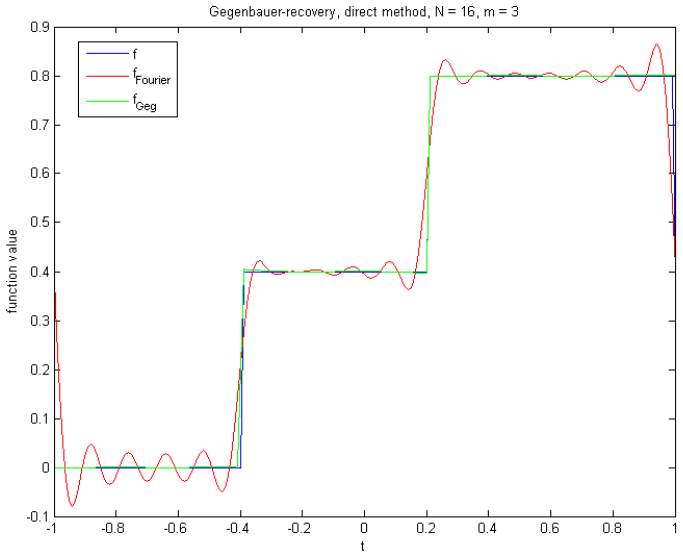

Figure 4-4: Some more discontinuies, here 3rd order polynomials are used for each subinterval, although in this case 0th order would give the same result.

Piecewise analytic functions

The polynomial reconstruction works extremely well for these step functions, for but this is not a realistic scenario.

The next example is a piecewise analytic function on $[-1, 1]$ of the form $f(x) = -2$ for $x < -0.2$, $f(x) = 10x^3$ for $-0.2 < x < 0.2$, $f(x) = \cos(8x)$ for $x > 0.2$.

Here I compared the reconstruction from the COS method, with the polynomial reconstruction using the modified IPRM algorithm, and a direct Legendre projection. The direct Legendre projection is calculated using the function definition instead of the Fourier coefficient and is for comparison only (since we normally do not know this function definition).

Here we can see that the the accuracy of the reconstruction is limited by the possible accuracy in approximating a cosine with polynomials of low order. If we increase the polynomial order the approximation gets better and the accuracy is again limited by the accuracy of the direct Legendre projection. This shows that for these cases almost no precision is lost using the Fourier coefficients instead of the original function.

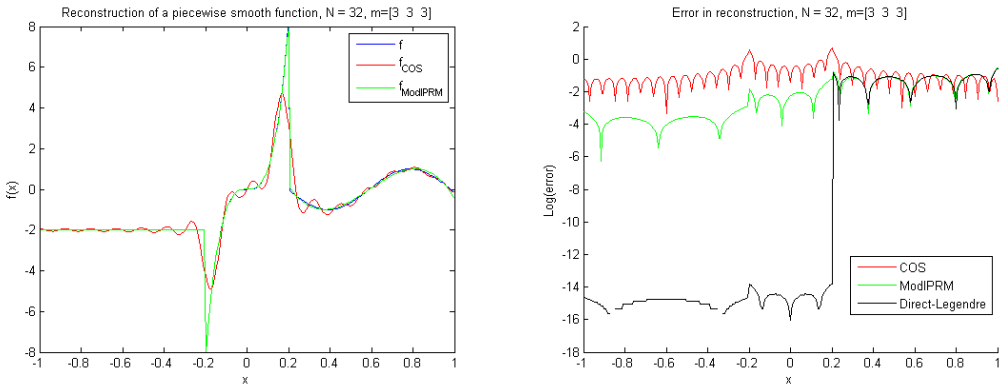

Figure 4-5: *Left:* Recovery of piecewise smooth test function $f(x) = -2$ for $x < -0.2$, $f(x) = 10x^3$ for $-0.2 < x < 0.2$, $f(x) = \cos(8x)$ for $x > 0.2$, using partial Fourier-sum $S_N f$ with $N = 32$ and modified IPRM, with polynomial order 3 in each subinterval. *Right:* Log-error of the reconstructions.

The p-th order Legendre expansion is (semi-) exact for polynomials of order $\leq p$, as can be seen in the error-plots in figure 4-7 for $x < 0.2$. Since the IPRM-method calculates a Legendre expansion from the Fourier coefficients, its accuracy is limited by the accuracy of the direct Legendre expansion on each subinterval.

The bottom right graph in figure 4-7 shows greatly improved accuracy as a result of increasing the polynomial order on the subinterval $[0.2, 1]$. For the IPRM-method the accuracy on the rest of the interval also improves significantly, which is a result of the algorithm solving a system of equations for the whole function. This would not occur if the direct Gegenbauer-expansion method was used, because that method calculates all the subintervals independently. The direct Gegenbauer method is however only stable for $\lambda \sim N$, [15], and we would prefer a fixed $\lambda = \frac{1}{2}$, because then we have orthogonality (which results in fast pricing).

The IPRM reaches machine accuracy for very low N as on the subintervals where the test function is of either lower or equal polynomial order.

The regular Fourier-reconstruction is the fastest, but due to the Gibbs phenomenon there is a large error of $O(1)$.

Filtering is also fast and reduces the effect of the Gibbs phenomenon for certain functions, but does not work for all discontinuities at the edges.

The Gegenbauer methods all give good results for this test function as far as the error is concerned.

There are however functions for which the direct-method does not converge. For higher values of m the IPRM also becomes unstable because of the conditioning of the transformation matrix W. The modified IPRM is more robust, but takes longer to calculate since more Fourier-coefficients are used. Using an analytical expression instead of Gaussian-quadrature (as is used in the literature) significantly speeds up this process, but the method is still expensive.

The direct Gegenbauer method uses $m = \lambda = N/4$ which is often used in literature because it is close to optimal and for this choice root-exponential convergence is proven in [16].

Table 4-1 shows the cpu-time and error for four different methods, for various values of N, with $m = N/4$ for all methods [1] and $\lambda = N/4$ for the Gegenbauer method and $\lambda = \frac{1}{2}$ for the IPRM and modified IPRM. The used testfunction is: $f(x) = 0$ for $(x < 0.2)$, $f(x) = \exp(3(x-0.2))-1$ for $(x > 0.2)$. Here the measured error is the maximum absolute difference between the reconstruction and the original function over all points on which the function is calculated, except for the exact jump locations. Here we see that for equal N the error of the reprojection methods is many orders of magnitude lower than the error in the original Fourier sum. The cpu time is much higher, especially for the IPRM using Gauss quadrature (column 3).

Table 4-1: CPU-time and absolute error for some reconstruction methods

	Fourier sum		Gegenbauer direct		IPRM		Mod IPRM	
N	time	error	time	error	time	error	time	error
16	0,00214502	3,44291987	0,02685392	3,00788294	0,02812818	0,51878284	0,02467405	0,44182827
32	0,00265266	1,94566399	0,02717494	0,23548899	0,04028051	0,01185878	0,02689393	0,00749478
48	0,00293307	0,90224215	0,03137006	0,00809256	0,08052723	0,00011890	0,03079027	0,00004990
64	0,00334000	0,57105346	0,04153481	0,00015368	0,20604503	0,00000319	0,03812816	0,00000237
80	0,00381706	0,69583951	0,05905931	0,00000185	0,56588951	0,00000320	0,05309502	0,00000244
96	0,00411160	0,89579248	0,09267043	0,00000010	1,65620262	0,00000121	0,08746878	0,00000079

Table 4-2 shows cpu-time and error for various values of N, keeping m constant at 0. Since the function is piecewise constant, higher order polynomials do not give any extra accuracy for the inverse method, which should be exact up to machine accuracy if the given Fourier coefficients are exact.

4-1-2 Recovery using filters

Figures 4-9 and 4-10 show the result of filtering in recovery of testfunctions with the COS method. As expected, the filters work the same way as they did for the regular Fourier series, so these are very similar to the Fourier recovery case in figure 4-8.

[1] Note that we here use a regular Fourier sum, and not the COS method series, which would perform better in this case.

Table 4-2: CPU-time and error for rectangle function

	Fourier		Gegenbauer		Mod IPRM	
N	time	error	time	error	time	error
16	2,16E-03	3,42E-01	2,63E-02	6,57E-05	2,36E-02	4,96E-08
32	2,57E-03	1,97E-01	2,73E-02	2,53E-05	2,37E-02	4,96E-08
64	3,35E-03	5,84E-02	3,43E-02	2,11E-08	2,37E-02	4,95E-08
128	4,99E-03	5,84E-02	1,07E-01	1,24E-10	2,38E-02	4,95E-08

4-1-3 COS density recovery and pricing

Figure 4-11 shows the result of the IPRM recovery used to improve the recovery of a uniform (squared) density function and a pricing function (called f and g in this figure) and their product. Figure 4-12 shows the filtered version.

Because the IPRM method is exact for polynomials up to order m for each subinterval, there is only an error due to the error in the approximated Fourier-coefficients, but this error can be made arbitrarily small in the COS method. In this case $m = [0, 0, 0]$ would have sufficed, since the density function is piecewise constant. The filtered version is not as accurate as the IPRM, but still shows a very clear improvement over the original COS approximation.

Table 4-3: Errors in recovery of rectangle function for COS method and filtered COS method

	COS		Filtered COS	
N	error$\|\|.\|\|_1$	error$\|\|.\|\|_2$	error$\|\|.\|\|_1$	error$\|\|.\|\|_2$
16	0,064488	0,0024348	0,0861038	0,0033241
32	0,037156	0,0016737	0,0427315	0,0023120
64	0,020681	0,0011135	0,0206423	0,0015805
128	0,011099	0,0006936	0,0096251	0,0010425
256	0,005701	0,0003769	0,0041680	0,0006358
512	0,002856	0,0001739	0,0015353	0,0003216

Comparing 1 norm and 2 norm errors, for the COS and filtered COS recovery of a block-function in table 4-3, it looks like there is not much of an improvement. This is because the reconstruction close to the jumps does not improve by filtering, and the error there dominates everything on the rest of the domain. If we look at the error away from these discontinuities in figure 4-10, results look much better for the filtered method.

In option pricing inaccuracies on the outside are most times far worse than inaccuracies around discontinuities, because the value function is generally large on the far left or the far right side, due to the definition of the put and call payoff functions.

Furthermore, in pricing, overshoots on the left and the right of a discontinuity in the density approximation cancel out each other, unless the value function also differs greatly around that point. This was already the case for the original COS method and still holds for the filtered version.

Digital total variation filtering

The DTV filtered recovery of the test function is shown in figure 4-13. The DTV removes the oscillations fairly well, close to the discontinuities the accuracy is a little better than that

of the lower order filters. The problem with this method for our purposes is that it works strictly in the time-domain, which makes it hard to combine with the pricing formula.

Padé methods

The Padé methods performs very good in function recovery with steep gradients and therefore recovers our test-function with great accuracy, as can be seen in figure 4-14.

Note that this result was obtained without using any knowledge of the discontinuities. The method can be improved upon by implementing it on sub-intervals, combined with edge detection (or given locations of discontinuities), as was done with the polynomial reconstruction.

This method is very useful as a high accuracy recovery method and in some sense is superior to the polynomial reconstruction, since it deals way better with steep gradients.

However, since the used rational basis is not orthogonal, fast pricing is again very difficult. If one can construct an orthogonal basis from these rational functions, this disadvantage would be resolved.

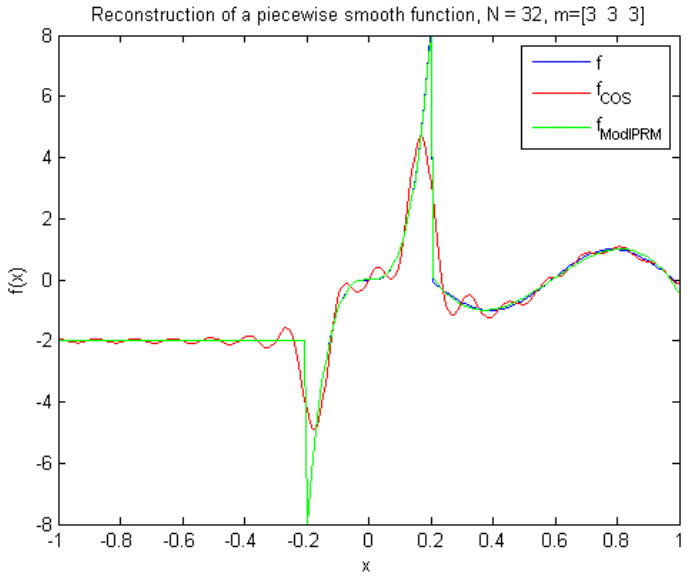

Figure 4-6: Recovery of piecewise smooth testfunction, $f(x) = -2$ for $x < -0.2$, $f(x) = 10x^3$ for $-0.2 < x < 0.2$, $f(x) = \cos(8x)$ for $x > 0.2$ using COS method and IPRM approximations

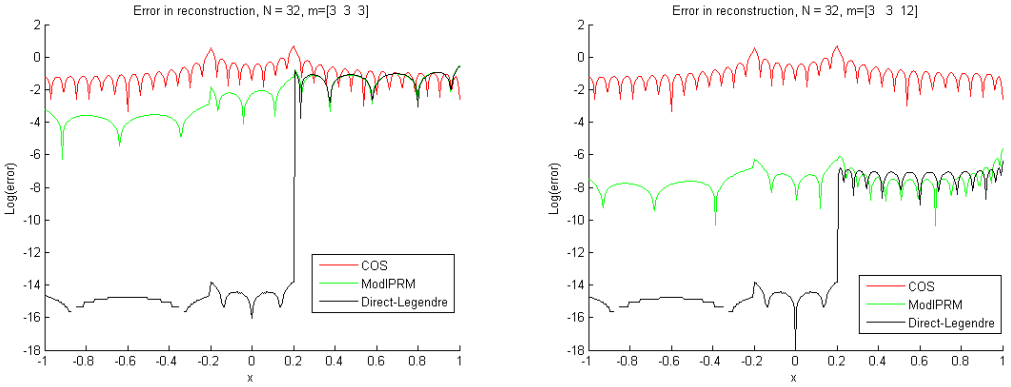

Figure 4-7: Error in function recovery for different polynomial orders, $m = [3, 3, 3]$ (left) and $m = [3, 3, 12]$ (right). The black line represents the error using a direct Legendre expansion (using $f(x)$ as input)

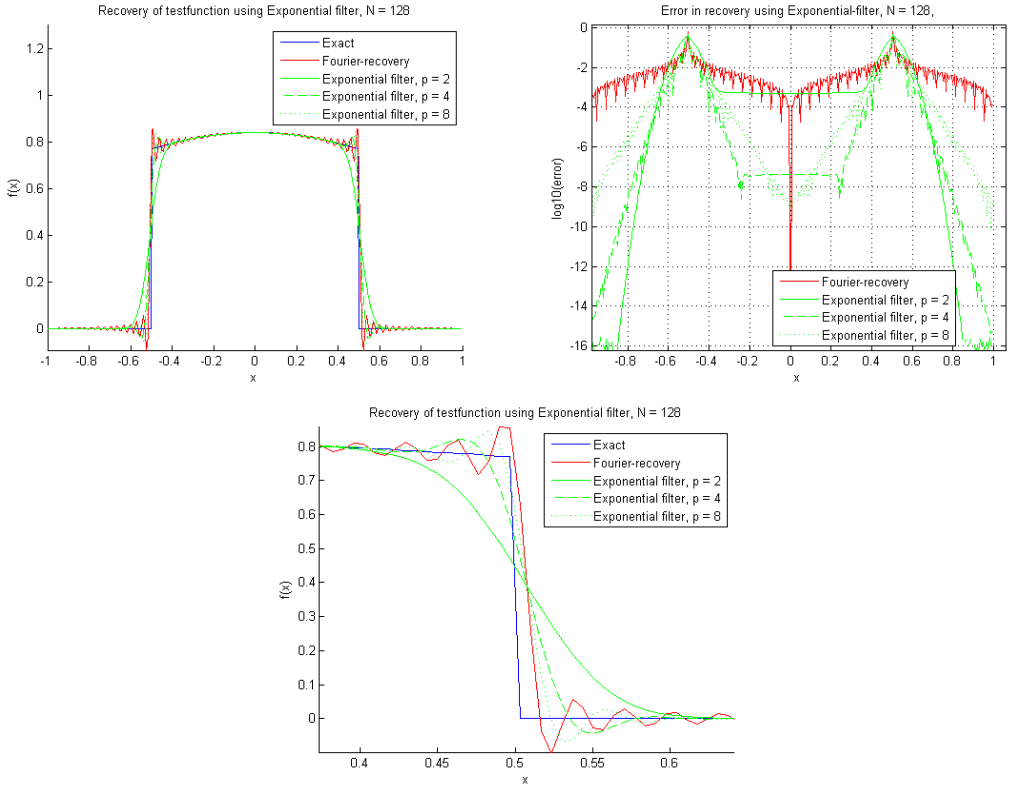

Figure 4-8: *Top Left:* Fourier recovery of testfunction $f(x) = \sin(\cos(x))$, for $x \in [-0.5, 0.5]$, 0 elsewhere. Using partial Fourier-sum $S_N f$ with $N = 128$ and filtered versions, using the exponential filter with $p = 2$, $p = 4$ and $p = 8$. *Top Right:* Log-error in the recovery, for $N = 32$, $N = 64$ and $N = 128$. *Bottom:* Closeup near discontinuity at $x = 0.5$

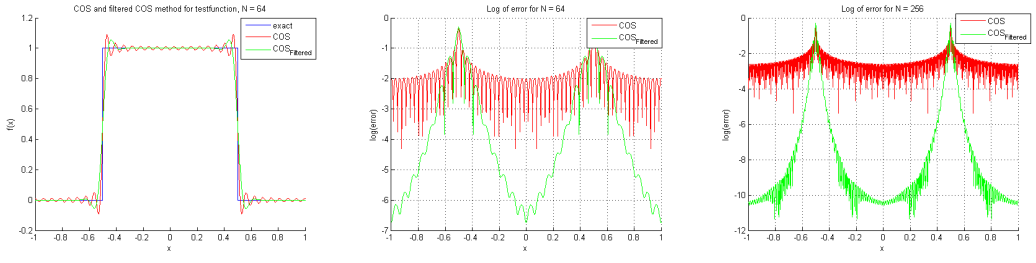

Figure 4-9: *Left:* COS method recovery of testfunction $f(x) = 1$, for $x \in [-0.5, 0.5]$, 0 elsewhere. Using COS method and filtered versions, using the exponential filter with $p = 6$; *Middle:* Log-error in the recovery, $N = 64$. *Right:* Log-error in the recovery, $N = 256$.

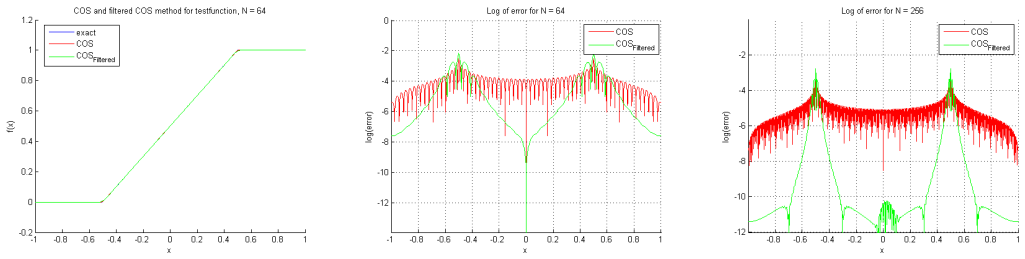

Figure 4-10: *Left:* COS method recovery of testfunction $f(x) = 1$, for $x \in [-0.5, 0.5]$, 0 elsewhere. Using COS method and filtered versions, using the exponential filter with $p = 6$; *Middle:* Log-error in the recovery, $N = 64$. *Right:* Log-error in the recovery, $N = 256$.

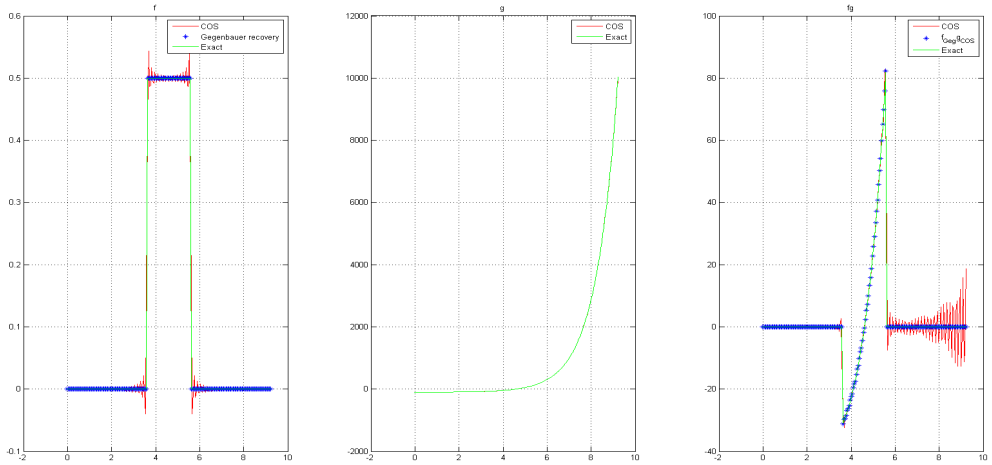

Figure 4-11: IPRM recovery applied to COS density-function recovery $N = 128$, $m = [2, 2, 2]$

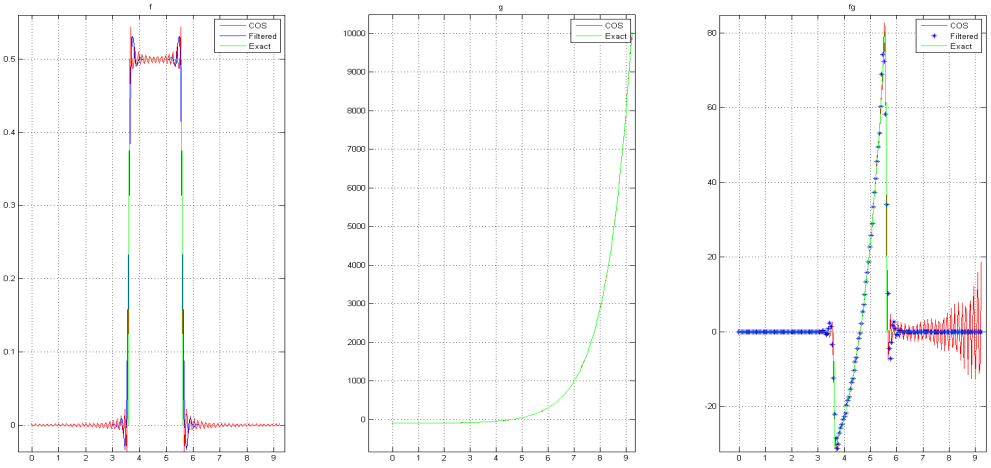

Figure 4-12: COS recovery and filtered version using exponential filter ($p = 6$) applied to COS density-function recovery $N = 128$.

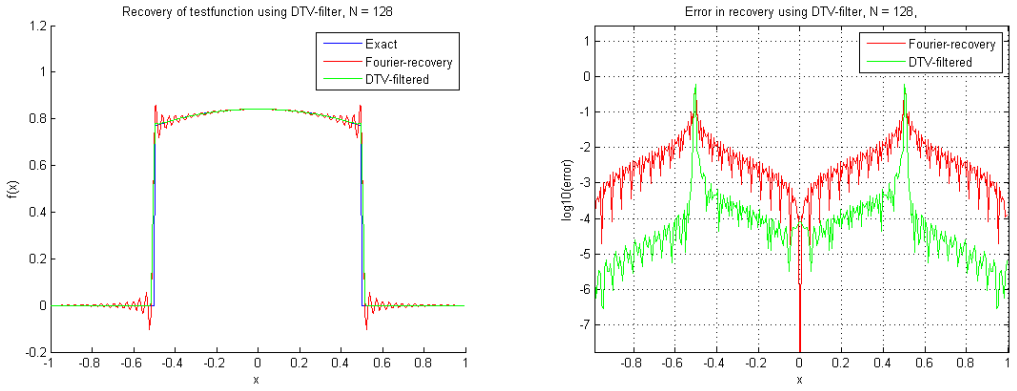

Figure 4-13: *Left:* Recovery of continuous periodic function $f(x) = \sin(\cos(x))$, for $x \in [-0.5, 0.5]$, 0 elsewhere. Using partial Fourier-sum $S_N f$ with $N = 128$ and DTV-filter using the same Fourier-coefficients. *Right:* Log-error of the reconstructions.

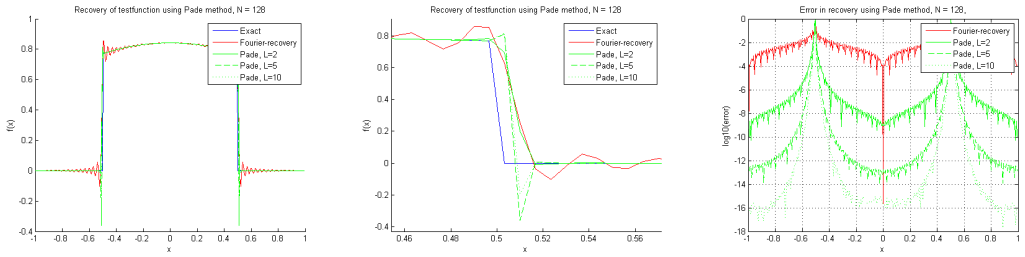

Figure 4-14: *Left:* Recovery of continuous periodic function $f(x) = \sin(\cos(x))$, for $x \in [-0.5, 0.5]$, 0 elsewhere. Using partial Fourier-sum $S_N f$ with $N = 128$ and Padé-method using the same Fourier-coefficients, for $L = 2$,$L = 5$ and $L = 10$. *Middle:* Closeup of recovery near the jump at $x = 0.5$. *Right:* Log-error of the reconstructions.

4-1-4 Applying the analytic expression to the IPRM

The direct Gegenbauer reconstruction calculates the Gegenbauer-coefficients for a given Fourier-sum. The inverse method instead calculates the Fourier-coefficients for each Gegenbauer-polynomial, and finds the Gegenbauer coefficients by solving a matrix equation.
For $\lambda = \frac{1}{2}$, the Legendre polynomials, the Gegenbauer weight-function is equal to 1. Therefore, for this weight, the k-th Fourier-coefficient of Gegenbauer-polynomial $C_n^{\frac{1}{2}}$ is essentially the same as the n-th Gegenbauer-coefficient of a trigonometric function $e^{ik\pi x}$. As a consequence we can use an explicit expression to determine each matrix element, instead of calculating them by Gaussian quadrature.

Table 4-4 shows a comparision between the modified IPRM using Gaussian Quadrature with 600 points and the Modified IPRM using the analytic expression. Parameter values are $\lambda = \frac{1}{2}$ and $m = N/4$ for each of the subintervals.

	Gaussian Quadrature		Direct Expression	
N	time	error	time	error
16	0,06052954	0,44182827	0,02467747	0,44182827
32	0,12018392	0,00749478	0,02689599	0,00749478
48	0,36126305	0,00004990	0,03079555	0,00004990
64	1,13474527	0,00000237	0,03797554	0,00000237
80	3,52090225	0,00000244	0,05324331	0,00000244
96	10,72400166	0,00000079	0,08783075	0,00000079

Table 4-4: CPU-time and error for the Modified IPRM algorithm, using Gauss-Legendre quadrature vs Direct Expression

4-2 Variance Gamma density recovery

Problems with recovering the the VG density-function:

As mentioned in [5], the COS method converges very slowly for the Variance Gamma distribution with short time to maturity ($T = 0.1$ is used here). Parameters are the same as in [5]: $K = 90$, $S_0 = 100$, $r = 0.1$, $q = 0$, $\sigma = 0.12$, $\theta = -0.14$, $\nu = 0.2$, $L = 10$

We tried to apply various methods to improve this convergence. In the following sections a brief discussion of the applied methods and the remaining problems is given.

4-2-1 Filtering

Filtering works well away from the peak in the VG density, but right at the peak the approximation becomes a little worse. This can theoretically be handled with using adaptive filtering, where the filter order depends on the distance to the discontinuity, but even then the approximation at the peak can at best get as good as the Fourier-approximation.

This would in itself be a good result: improving the accuracy everywhere except the region close to the peak. But a drawback of adaptive filtering methods is that they are hard to combine with the COS-pricing formula, where the option price is calculated directly from the Fourier-coefficients of the distribution and price function.

For recovering the density, adaptive filtering is still an option, since it does succeed in removing the oscillations at the tails of the density.

Pricing with the filtered method

Implementation of a filter in the COS method is actually quite straigthforward: once a suitable filter and order is chosen one can multiply each of the Fourier-coefficients with a factor depending only on n and work with the COS method as before.

Below are some results of the filter applied to the pricing of a call option. Figure 4-17 shows the error for a call price for various values of the strike price K. Only for low values of N and option values S close to the strike price K the filtered COS method performs slightly worse. This can be explained by the flattening of the peak in the VG density-function as can be seen in figure 4-16 For all other values the advantages (the reduction of the oscillations at the tails) outweight these disadvantages, resulting in faster algebraic convergence.

Different filters

I used 3 different filters: the exponential filter, the VandeVen filter and the erfclog filter, as described in section 3-3. In figures 4-18 and 4-19 one can see that in this case the exponential filter performs better for low values of N and the other two perform slightly better for larger

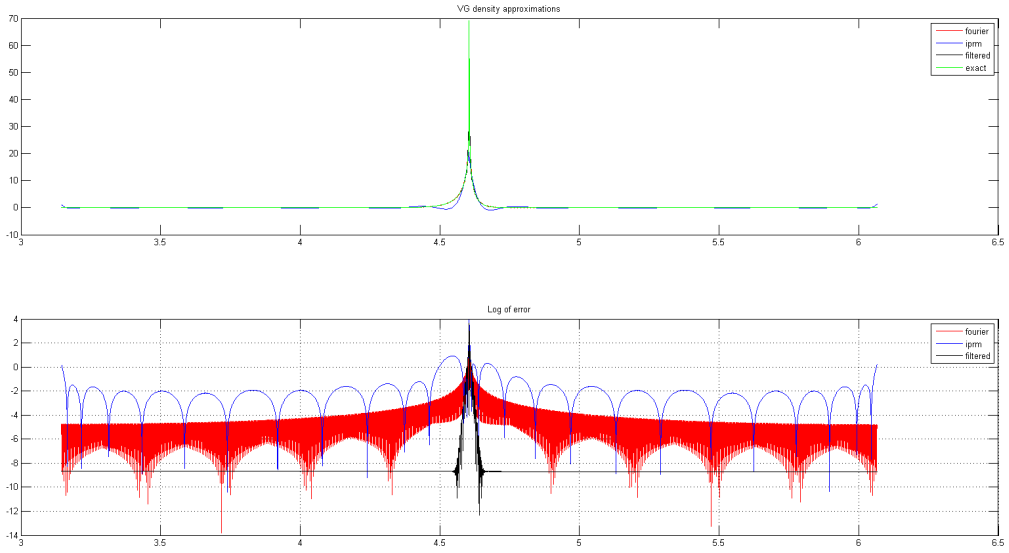

Figure 4-15: Recovery of VG-density, using Fourier-recovery, $N = 1000$, filtered version with exponential filter of order 6 and IPRM $m = [10, 10]$. The 'exact' function is actually a Fourier-sum using $N = 50000$

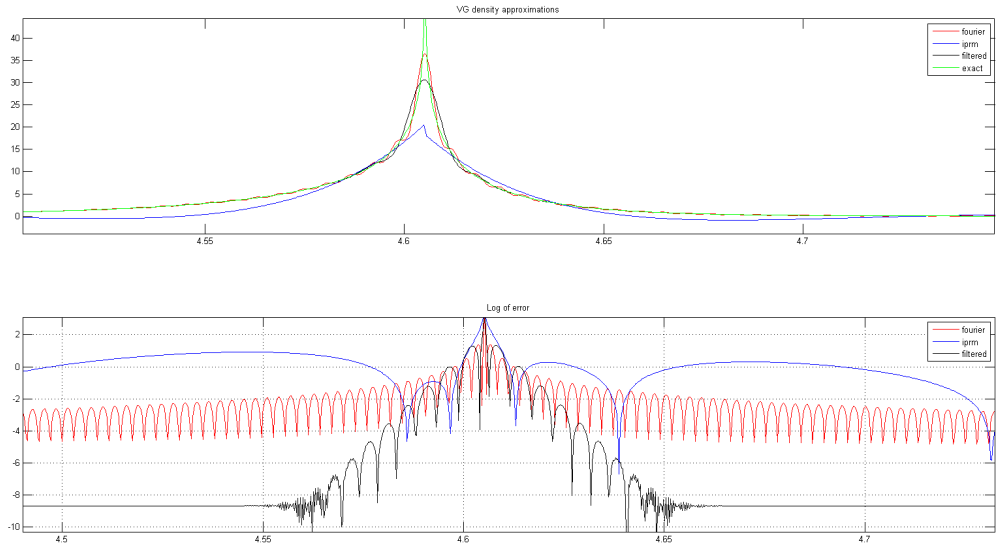

Figure 4-16: Recovery of VG-density: closeup of peak

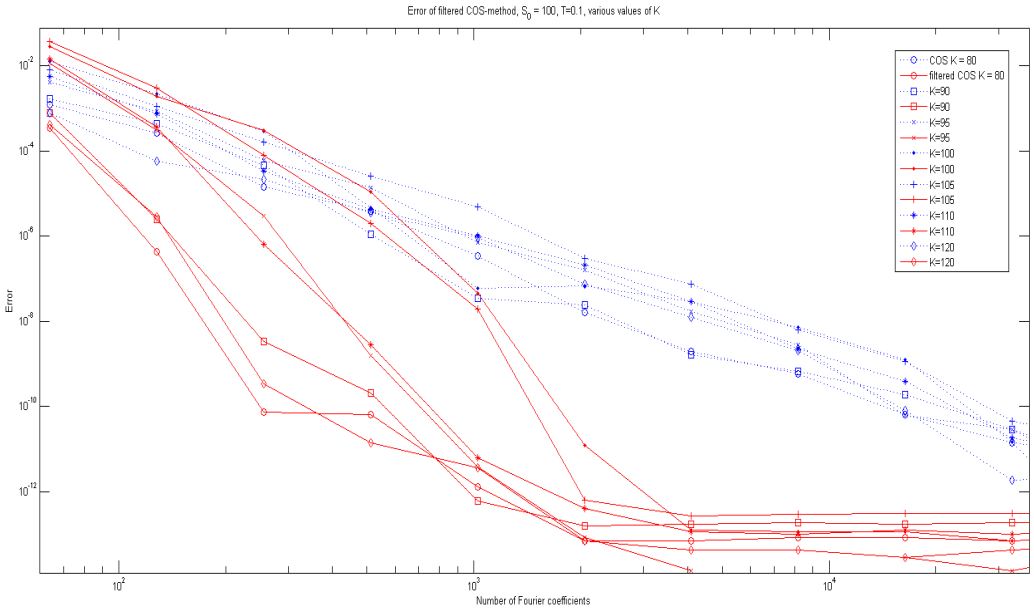

Figure 4-17: Error of the filtered and original COS method for a call price in the VG-model for various values of K: $K = 80$ (O); $K = 90$ (□); $K = 95$ (x); $K = 100$ (.); $K = 105$ (+); $K = 110$ (*); $K = 120$(⋄). For all values the used filter is the exponential filter or order 10.

values. These differences are however pretty small, so based on these results there is no reason to prefer one over the others yet.

We also experimented with different filter orders, for this function a relatively high order $p = 10$ gave the best results, for all the filters. A short explanation for this is that for this specific case the lower order filters smoothen out the sharp peak too much.

4-2-2 IPRM

Although strictly speaking there is only a first order discontinuity which causes the Gibbs-oscillations, the real problem here is that the VG density is very hard to approximate using polynomials only, even when split into two or more parts (the left and right tail), because of the very steep peak and flat tails.

We also tried using the polynomial method on multiple sub-intervals, large intervals at the end of the tails and smaller intervals near the peak. This improves the result, but gives problems at the new edges: the function is no longer continuous at those points. The fundamental problem here is that the VG density behaves much like an exponential function, and we are trying to approximate it with polynomials. Cutting the interval in pieces does change the magnitude, but the principle remains the same. One could still improve this approximation somewhat by using even more intervals, but unless we also increase the number of Fourier coefficients, this will result in instability.

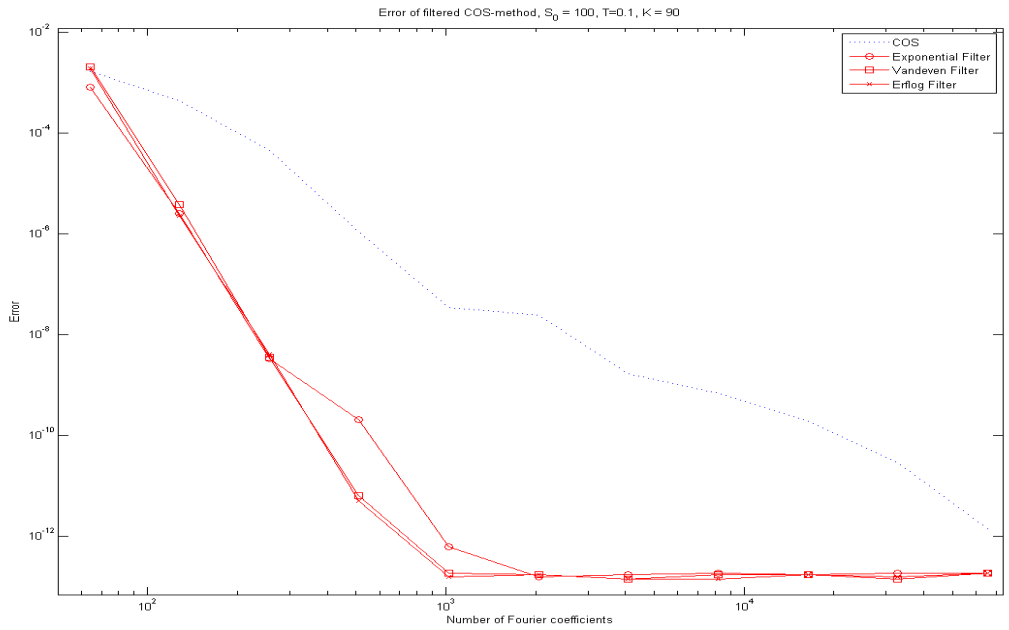

Figure 4-18: Error of the filtered and original COS method, for a call-price in the money, VG-model, using different filters, all of order 10

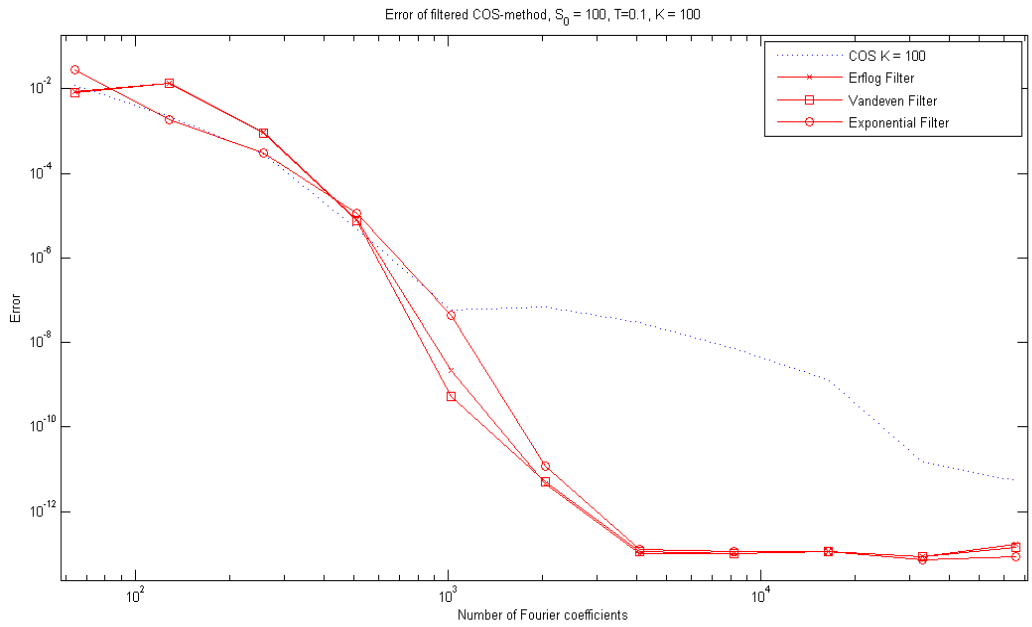

Figure 4-19: Error of the filtered and original COS method, for a call-price at the money, VG-model, using different filters, all of order 10

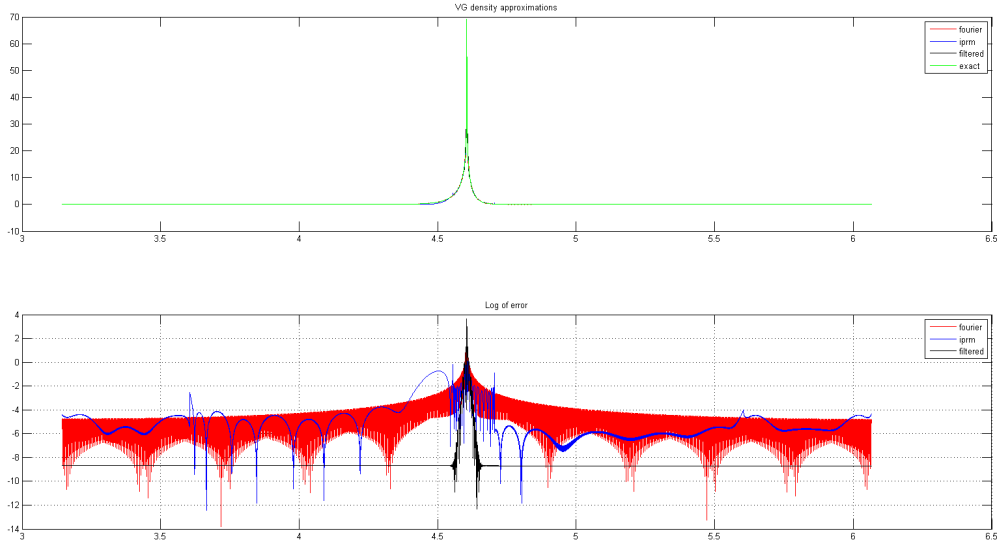

Figure 4-20: Approximation of VG-density, using Fourier-recovery, N=1000, filtered version with exponential filter of order 6 and IPRM, m=[10,10,10,10,10,10].

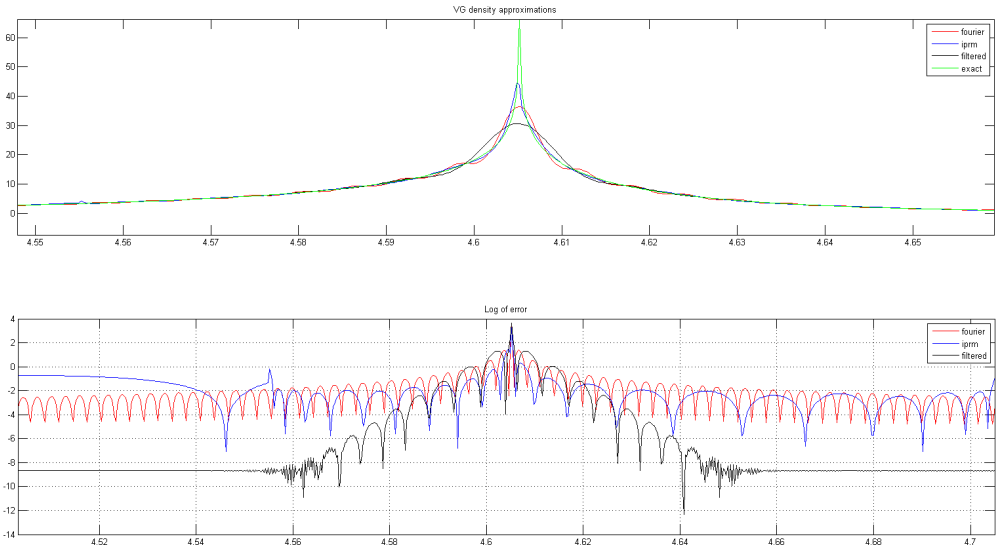

Figure 4-21: Closeup of peak

4-3 CIR density recovery

4-3-1 Results of filtering

The COS method recovery of the CIR density function converges very slowly due to the oscillations which are mainly caused by the sharp peak on the left hand side which is hard to approximate using a cosine series.
As was the case with the VG-density, filtering does not improve the reconstruction of the sharp peak on the left side, in fact the approximation is slightly worse in the close vicinity of the peak. However, on the rest of the interval, the recovery is greatly improved due to the removal of the oscillations. See figure 4-22.

Pricing results will be presented in the next section, where the CIR density will be used in the 2D Heston model.

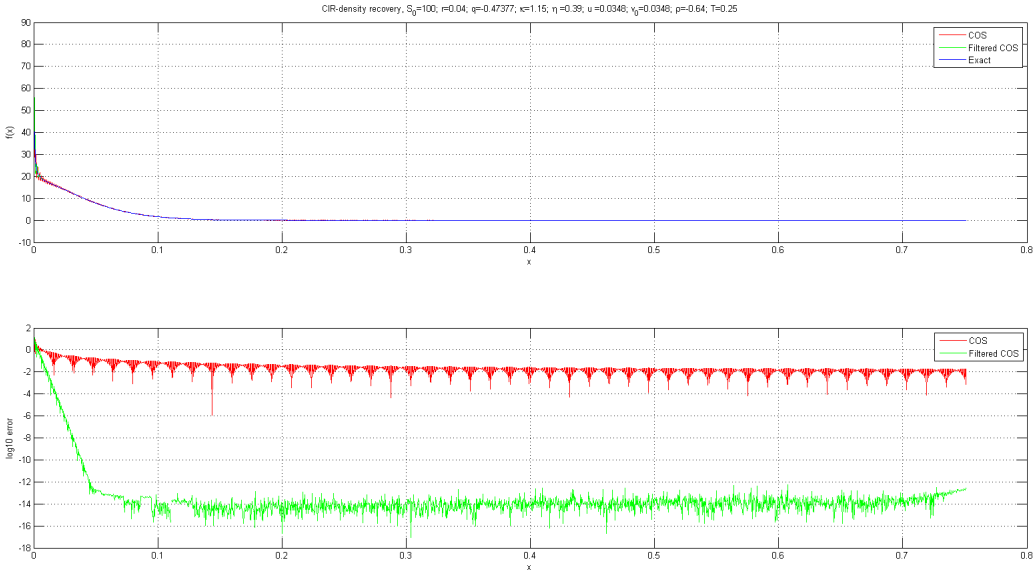

Figure 4-22: CIR density recovery, using COS method, $N = 1024$ and filtered COS with exponential filter of order $p = 10$

4-4 The 2D Heston Model

The density surface is shown in figure 4-23. In one direction we can recognize the CIR density from the previous section, while in the other direction the shape is that of a Gaussian.

Figures 4-24 and 4-25 show the function evaluated at lines parallel to the axes, through the peak of the density. The dashed lines show the COS-version before filtering, whereas the solid lines show the filtered version. The close-up versions in figures 4-26, clearly show the improvements.

As in the 1 dimensional case, the recovery is improved everywhere except for the close vicinity of the sharp peak. The most significant improvements are shown on the tail, seen in figure

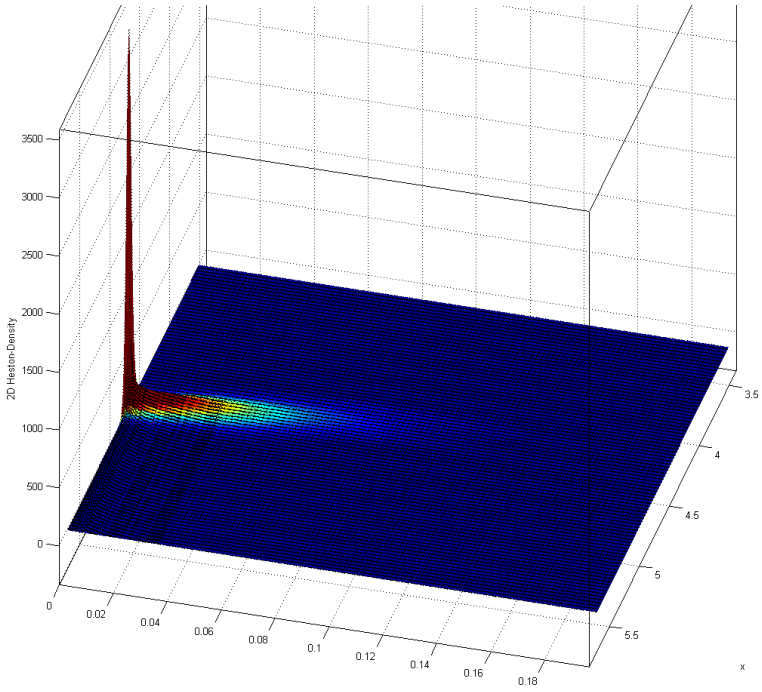

Figure 4-23: 2D Heston Density surface

4-27, where the non-filtered version oscillates with an amplitude that is way bigger than the actual function value.

In the X direction the density function is smooth, Gaussian-like, so in that direction the COS method already converges exponentially.

In the Y direction we see the sharp peak of the CIR-density, resulting in Gibbs-oscillations, which can be seen in figures 4-25, 4-26, 4-27. Notice that at the tail the oscillations amplitude is greater than the actual height of the tail, causing negative values.
The filter is only applied in one direction. Filtering in the X-direction does not improve the reconstruction since the functions are already smooth in that direction. See figure 4-24

4-5 Portfolio loss distribution

The following example is taken from [6], the new part is the application of filters.

In this example here we are given a hypothetical portfolio:

- There are 20 names in total.
- All have the same default probability of 5% and the same asset correlation of 23%.
- One issuer has the exposure of 1 million euros, the rest have 10k each.

The upper truncation range of the COS method is set as $\sum_n V_n$, and the lower truncation range is set as 0. The intuition behind is this is that the default loss cannot exceed the total exposures and will be larger than 0.

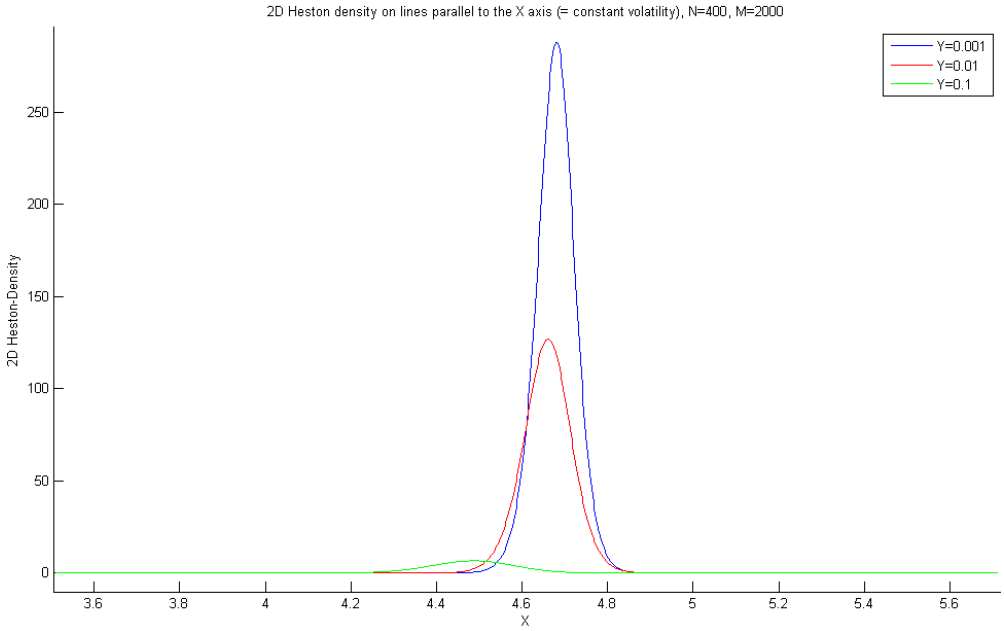

Figure 4-24: Recovery of 2D Heston Density along lines parallel to the X-axis (constant volatility).

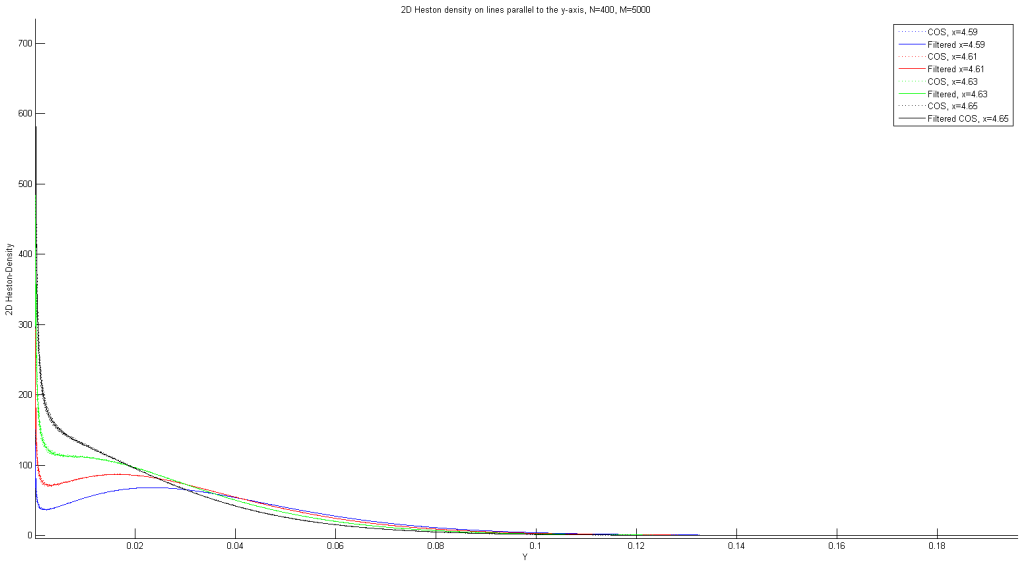

Figure 4-25: Recovery of 2D Heston Density along the Y axis, using COS method and filtered version, using the exponential filter with $p = 8$

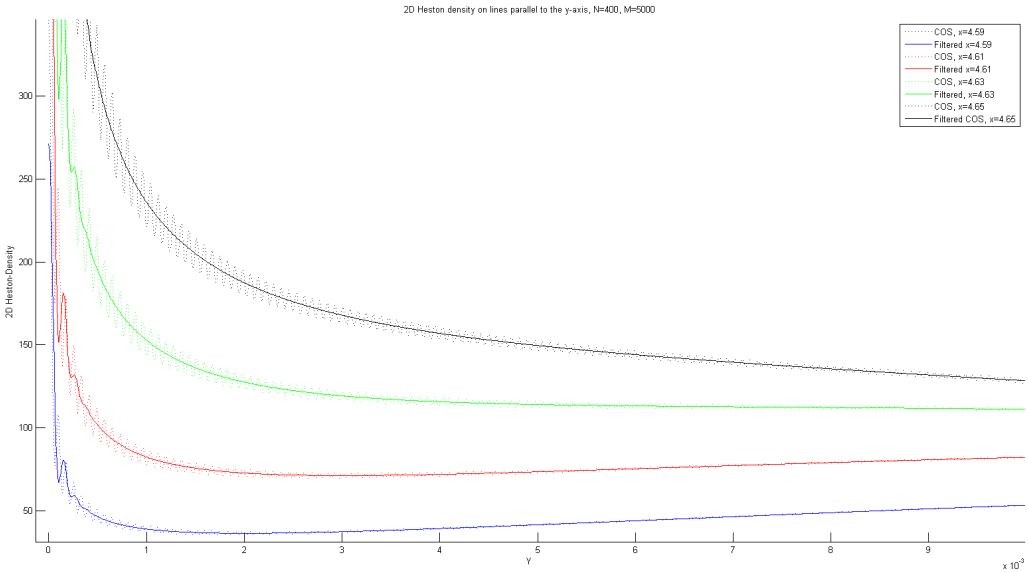

Figure 4-26: Recovery of 2D Heston Density: Closeup of left edge

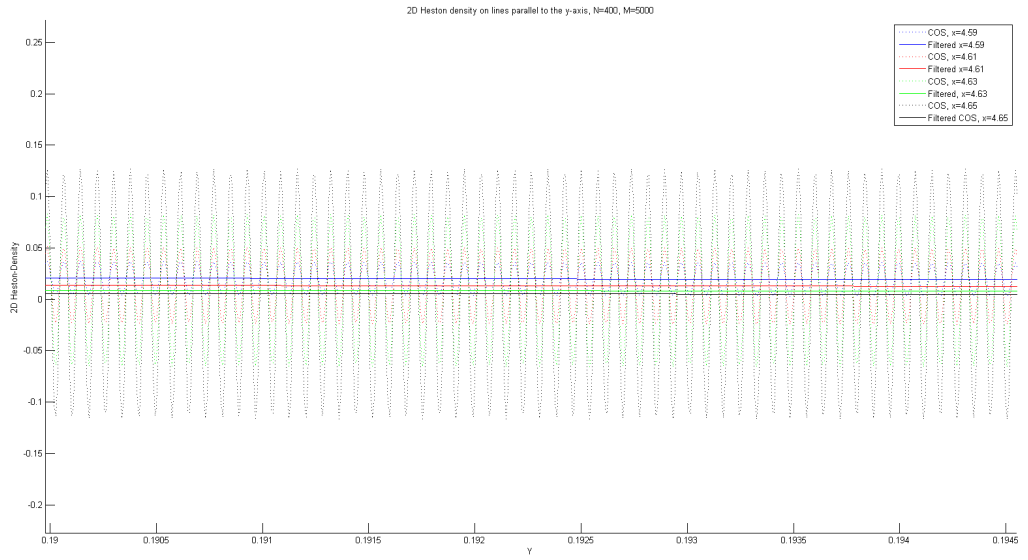

Figure 4-27: Recovery of 2D Heston Density: Closeup of right edge

The computational complexity of the COS method is linear in

- the number of cosine expansion terms,
- the number of discrete points in the function recovery,
- the number of issuers.

In the example above, the number of cosine series expansion is 1024. The number of discrete points in Y is 5000. VaR(99.9%) returned by the COS method equals 1080460. The reference value from Monte Carlo simulation is 1080000. Thus, the relative error is 4.3×10^{-4}.

In this case the cpu time of the COS method is about one third of Monte Carlo (with 1 million simulations). The difference in cpu time becomes much bigger as the number of issuers increases, or as the VaR level is higher. The advantages of applying the COS method are that [6]:

- the computational time does not depend on the VaR level, unlike the Monte Carlo simulation, where the high the VaR level the more simulation is needed due to Monte Carlo noise.
- the more issuers there are in the portfolio, the less cosine expansion terms are needed to obtain the same level of accuracy. The reason is that the more issuers there are, the smoother the loss distribution, and the faster the COS method converges.

The loss distribution of this example portfolio is plotted in figures 4-28, 4-29 and 4-30. The blue line is the result of a Monte Carlo simulation and serves as our reference solution here. The red-dotted line is the COS method approximation, and the green line is the filtered-COS version, using an exponential filter with $p = 6$.

Near the lower right corner of figure 4-30 there is some distance between the filtered-COS and the Monte Carlo approximation. For the figure 10^7 simulations were used. This distance gets smaller if the number of Monte Carlo simulations is increased.

Measured this way the maximum error in value at risk (VaR) close to a step will always be the order of the step size. In that sence the VaR calculation is not too sensitive for the Gibbs' oscillations as long as they do not overreach the height of the steps. Filtering ensures that the error does not become any larger than that even for percentages very close to 1, see figure 4-31.

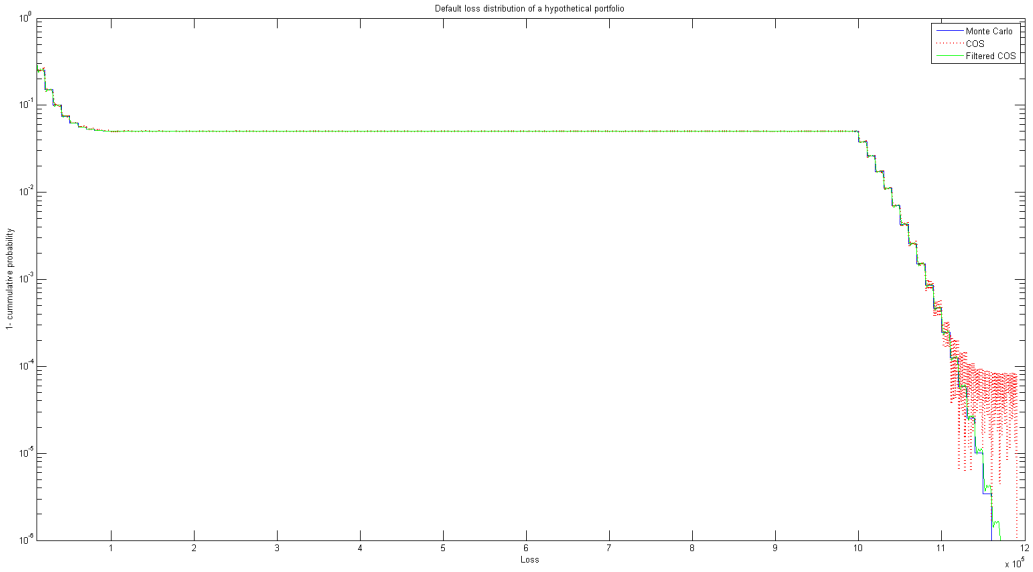

Figure 4-28: Recovery of portfolio loss distribution function using COS method and filtered version, using the exponential filter with $p = 6$

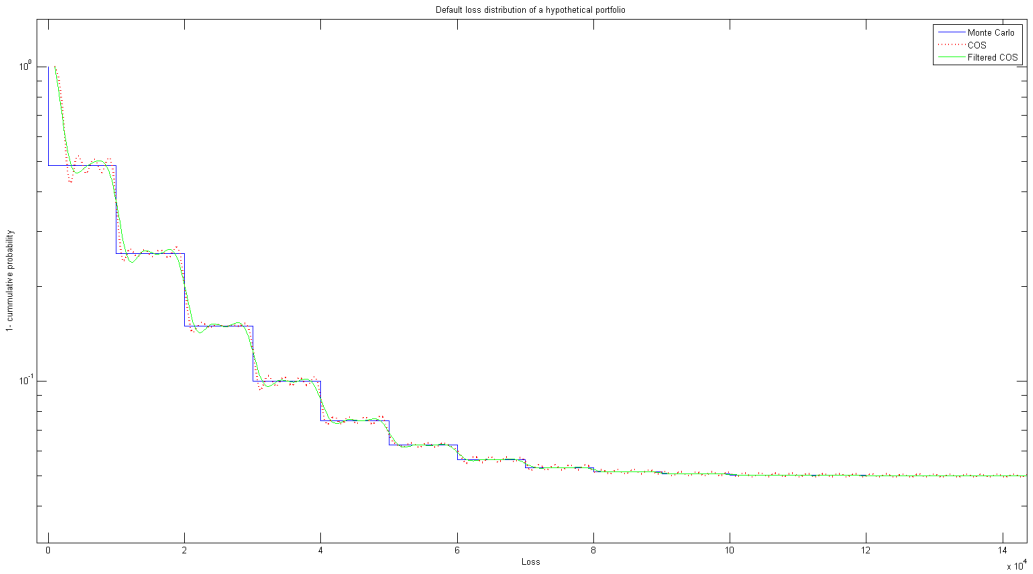

Figure 4-29: Portfolio loss distribution: Closeup of the left edge

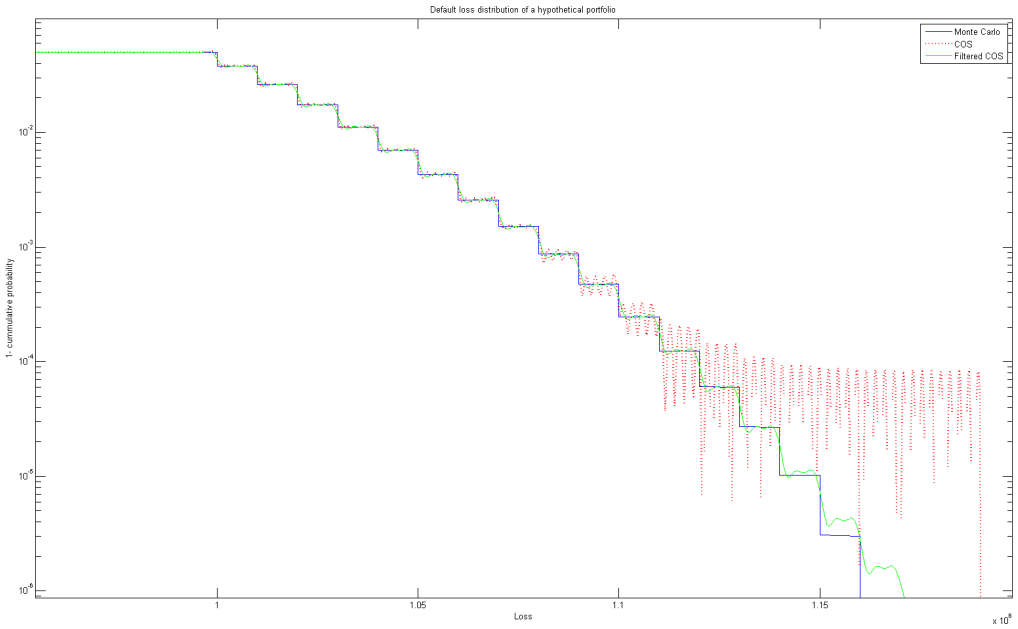

Figure 4-30: Portfolio loss distribution: Closeup of the right edge

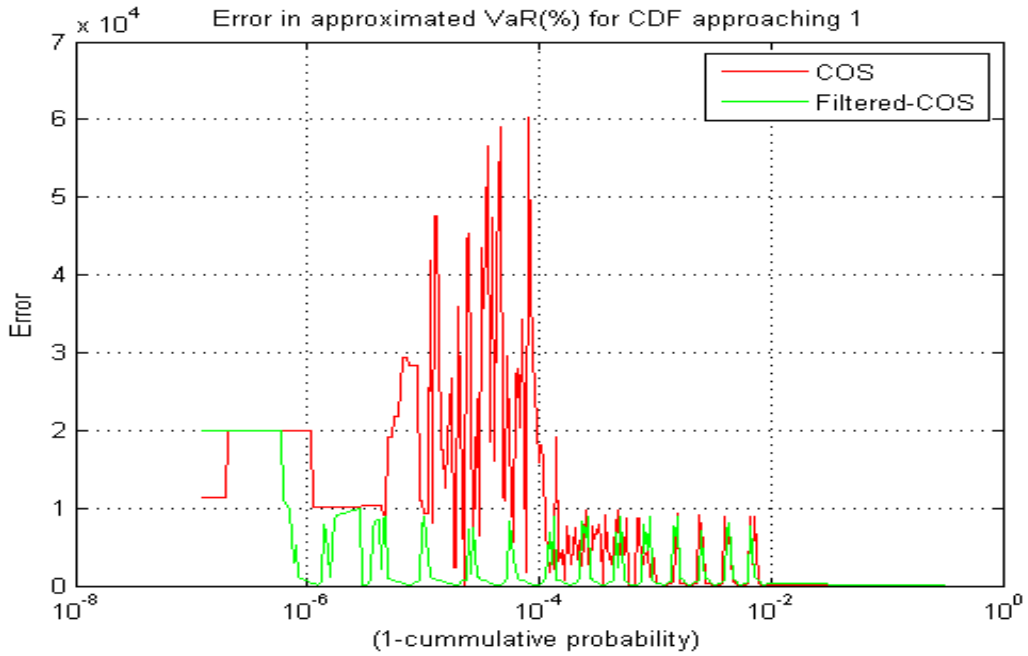

Figure 4-31: Value at Risk, calculated from the recovered default loss distribution. The error is measured using the Monte Carlo values as a reference

4-6 Boundary issues

4-6-1 Filtering

In [5], section 4.1.1, an example of a significant error close to the spatial boundaries is presented. The problem occurs when using the cos-approximation of the payoff of a call option at terminal time T, with log-asset price $y = log(S_T)$, is given by the function

$$g(y) = (e^y - K)^+ \tag{4-1}$$

In [28] a solution using extrapolation is proposed, but this is somewhat hard to implement, especially in higher dimensions.

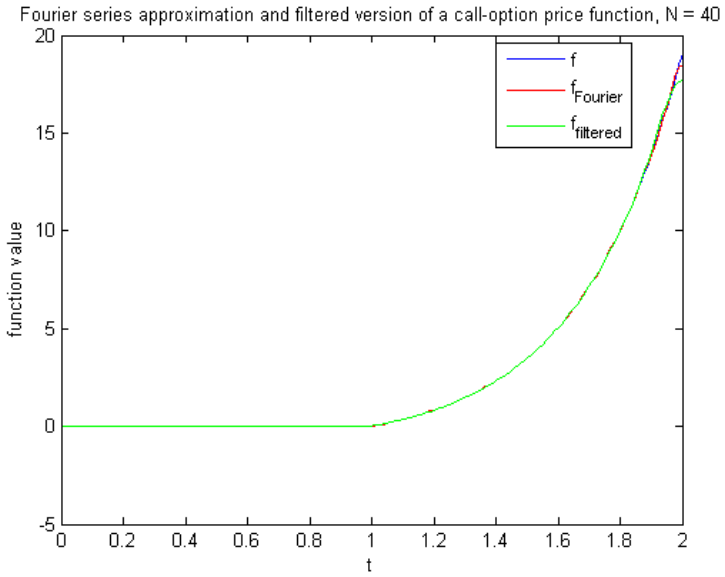

Figure 4-32: Call-price function, Fourier-approximation and filtered version, using the Erfc-Log filter with $p = 4$

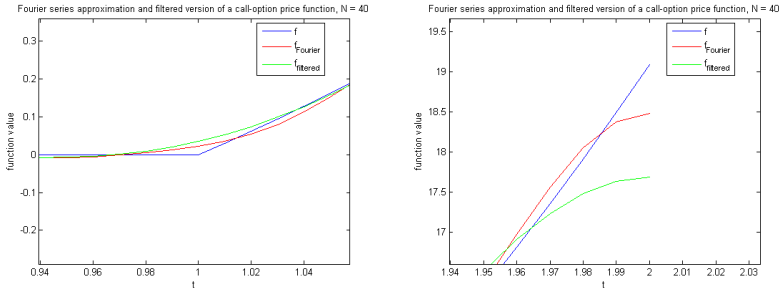

Figure 4-33: Closer view of error in middle and at right edge

The periodic version of the payoff-function, which is what the cosine-expansion is approximating, is symmetrical around $x = 0$ and $x = 2$. Two first order discontinuities can be observed, at $x = 1$ and $x = 2$, the two areas areas at which the COS method has most difficulties are

around these discontinuities. Note however that at the first order discontinuities the pointwise convergence is only of order n and due to this, the overall convergence (in the smooth regions) also is only algebraic.

Applying a normal filter does not solve this problem close to the edges, it actually makes it worse, since the filter reduces the higher Fourier-coefficients and therefore softens up the steepest gradients. Away from the edges the approximation becomes much smoother and the error is reduced.

We tried combining this with the COS-pricing method, but with no success. When using a process with a smooth underlying density such as GBM, the coefficients F_k already decay exponentially, so the product $F_k V_k$ also decays exponentially as long as the V_k are bounded (which they must be for the method to work at all).

For a non-smooth density, such as the VG-density, filtering does significantly improve the decay of the coefficients, but this is exactly the same effect that was reached by filtering the F_k, since $[\varphi(k/N)F_k]V_k = F_k[\varphi(k/N)V_k]$.

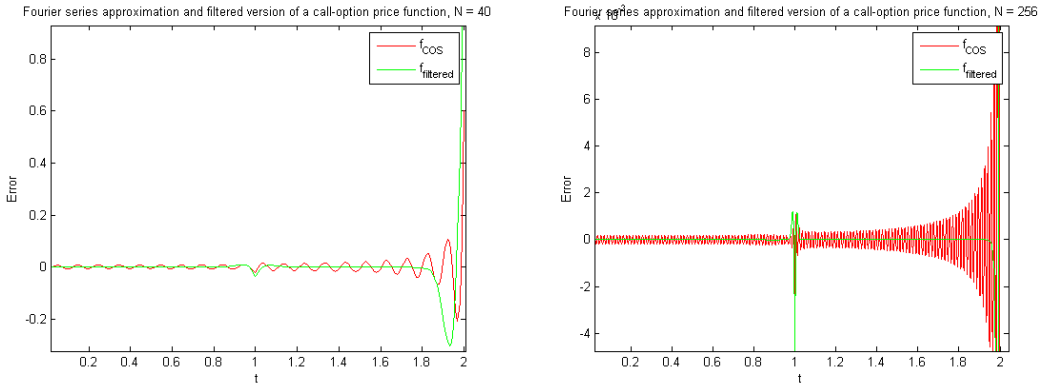

Figure 4-34: Error between Call-price and COS-approximation and filtered version, using the Erfc-Log filter with $p = 4$

Chapter 5

Conclusions and Outlook

5-1 Conclusions

The COS method is a relatively new option pricing method that performs very well compared to other option pricing methods in suitable cases. When the underlying density is smooth enough we even have exponential convergence.

When the underlying density is not so smooth, the case the method suffers from the Gibbs phenomenon and the convergence is only of low algebraic order. In this research various methods to improve the convergence rate for these cases were investigated.

Reprojection methods are very suitable for recovering piecewise smooth density functions from its Fourier coefficients with great accuracy up to the discontinuities. While it is possible to calculate option prices with reprojection methods, in practice the computational costs are too high compared to those of other methods.

Filters on the other hand are not as accurate up to the discontinuities, but on the rest of the domain they perform very well. In the COS method we are in fact not really interested in pointwise values of the probability density, but instead want to evaluate integrals over the complete interval.

If the COS method converges for a given case, we can be certain that the filtered COS method also converges. In practical cases where the COS method suffers from Gibbs oscillations, this convergence will be significantly faster in terms of the number of required Fourier coefficients as well as in cpu time.

The improved convergence is illustrated by examples involving the VG density, the CIR density in 1 dimension, the 2 dimensional Heston model and a VaR calculation based on a portfolio loss distribution.

This improved convergence at no additional computational cost and the fact that the filtering is very easy to implement, even in multiple dimensions, makes the filtered COS method a natural solution to some of the problems with the COS method.

5-2 Future Research

While the improved results are easily measured for specific cases, we have not formulated any objective criteria for the method to work, established error bounds, et cetera. There is a trade-off between local and global errors, which depends heavily on the underlying density and the payoff function.

There are still many other possible improvements which have not been investigated in depth. The investigated methods could be tailored more specific for the problems at hand, for example by making use of prior knowledge, such as the shape of the function to be reconstructed, which is often known. Using a special, wavelet-like basis with functions that are closer to the functions we are trying to approximate can make a great difference.

In reprojection methods, for example results from previous calculations could be recycled to save calculation time. Also, the boundary issue needs to be investigated further.

Table 5-1: Improvements for spectral methods

Mollifiers	Relatively easy to implement, works in time domain, therefore relatively expensive COS pricing
Filters	Easy to implement, very accurate away from jumps, works in frequency domain, therefore fast COS pricing
Adaptive Filters	Somewhat harder to implement, improves accuracy of filters close to jumps. Adaptive parts works in time domain, therefore relatively expensive COS pricing
Reprojection methods	Relatively hard to implement - very accurate for simple functions, lots of terms needed for general functions, computationally expensive. Numerical tability issues, parameters must be carefully chosen
Inverse reprojection methods	Resolves stability issues and simplifies parameter choice of Reprojection methods. Allows for use of orthogonal basis, which can be used for relatively fast pricing. Disadvantage: dependency between all subdomains
Padé methods	Relatively hard to implement, very accurate function recovery, standard basis is not orthogonal, which makes pricing a lot more difficult
Hybrid methods	potentially interesting for recovery, but hard to implement and unsuitable for COS pricing

Appendix A

Appendix

A-1 Gamma function

The gamma function, which occurs in the explicit expressions, is defined via an improper integral:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt \tag{A-1}$$

The Gamma function is well defined for all complex numbers except the non-positive integers. In our application we only use positive real values, on which the function is well defined and smooth. For the positive integers we have $\Gamma(n) = (n-1)!$

The asymptotic behavior of the Gamma-function is described by the Stirling formula, which is often useful in error analysis and approximations:

$$(2\pi)^{\frac{1}{2}} x^{x+\frac{1}{2}} \leq \Gamma(x+1) \leq (2\pi)^{\frac{1}{2}} x^{x+\frac{1}{2}} e^{\frac{1}{12x}} \tag{A-2}$$

A-2 Bessel functions

Some other functions that occur frequently in the used formulas are the Bessel functions which are solutions of the differential equation

$$x^2 y'' + xy' + (x^2 - p^2)y = 0, \tag{A-3}$$

known as the *Bessel equation* of order p. For general p we have the series solution

$$J_p(x) = \sum_{j=0}^{\infty} \frac{(-1)^j}{\Gamma(j+1)\Gamma(j+p+1)} \left(\frac{x}{2}\right)^{2j+p}, \tag{A-4}$$

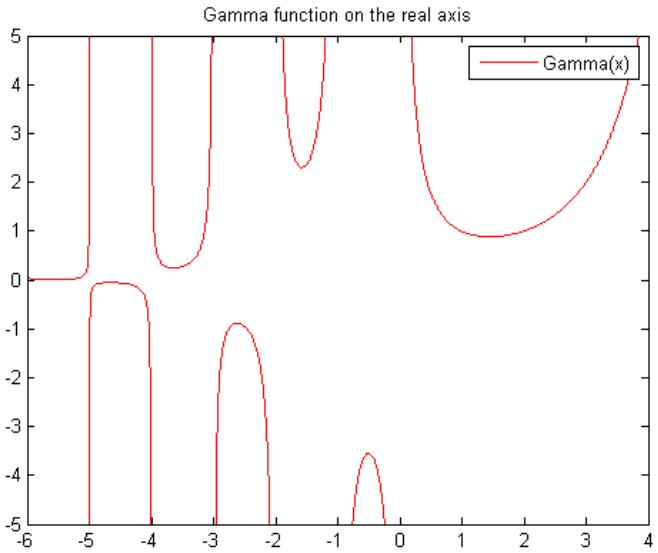

Figure A-1: The Gamma funcion on the real axis. The function is defined for all complex numbers except the non-positive integers. For the positive integers $\Gamma(n) = (n-1)!$.

the *Bessel function of the first kind*, with $\Gamma(z)$ the Gamma function.

Since the Bessel equation is a second order equation there is a second independent solution. If p is *not an integer* there is a second independent solution $J_{-p}(x)$, but if p is an integer then $J-n = (-1)^n J_n$, so these solutions are linearly dependent.

Although it is not used in this thesis, for completeness we mention that the *Bessel function of the second kind* is given by:

$$Y_p(x) = \frac{J_p \cos(p\pi - J_{-p}(x))}{\sin(p\pi)}. \tag{A-5}$$

A-3 Measures

To every *cadlag* (right continuous with left limits) process X one can associate a random measure on $\mathbb{R}^d \times [0, \infty]$ describing the jumps of X:

Definition 5. *Jump measure*

$$J_X(B) = \#\{t : (X_t - X_{t-}) \in B\} \tag{A-6}$$

So for every measurable set $A \subset \mathbb{R}^d$, $J_X([t_1, t_2] \times A)$ counts the number of jump times of X between t_1 and t_2 such that their jump times are in A.

Definition 6. *Levy measure*

Let $\{X_t\}_{t \geq 0}$ be a Levy process on $\mathbb{R}^d$. The measure ν on $\mathbb{R}^d$ defined by

$$\nu(A) = E[\#\{t \in [0,1] : \Delta X_t \neq 0, \Delta X_t \in A\}] \tag{A-7}$$

is called the Levy measure of X: $\nu(A)$ is the expected number, per unit time, of jumps whose size belongs to A.

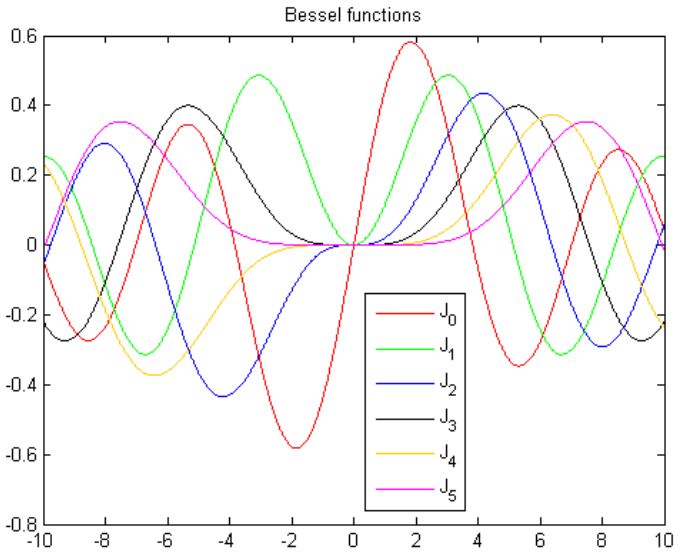

Figure A-2: Bessel functions of the first kind

A-4 Spectral methods

Spectral methods are used to approximate functions by projecting them onto a space of orthogonal polynomials. When the underlying function is periodic it is most natural to use trigonometric (Fourier) polynomials. For non-periodic functions the Chebyshev polynomials are a popular choice. Spectral methods based on functions such as Hermite, Laguerre of rational Chebyshev functions are possible as well, but the above mentioned are most popular because of the availability of fast transform methods [1].

Relation with other numerical methods

Spectral methods are also closely related to other numerical methods such as *finite element* and *finite difference* methods.

The basic idea of any approximation method is to assume that a function $f(x)$ can be approximated by a sum of basis functions:

$$f(x) \approx f_N(x) = \sum_{n=0}^{N} a_n \varphi_n(x) \tag{A-8}$$

Substituting this approximation into an equation of the form

$$Lf = g(x) \tag{A-9}$$

where L is a given differential of integral operator, one wants to find series coefficients a_n that minimize the residual function defined by

$$R(x, a_0, ..., a_N) = Lf_N - f. \tag{A-10}$$

Now there are a lot of choices that distinguish the different methods, such as [1]:

- Choice of basis functions
- What residual-minimizing condition to use
- Optimum collocation points
- How to solve the resulting system

The main difference lies in the used set of basis functions used as *trial functions*. Finite difference uses overlapping local polynomials of low (typically first or second) order.

Finite element method use local smooth functions, polynomials of fixed degree which are non-zero only on a subdomain.

Spectral methods use global smooth functions, such as the trigonometric polynomials used in the Fourier series.

The numerical methods can be further classified according to the *test functions* that are used. The well known Galerkin and tau methods use the trial functions as test functions, whereas *pseudospectral methods* use delta functions at special points, and thus evaluate the approximation only at those so-called *collocation points*.

A-5 The Dirichlet kernel

The Dirichlet kernel is defined by:

$$D_N(y) := \frac{1}{2\pi} \sum_{n=-N}^{N} e^{ikx} \equiv (sin(N + 1/2)y)/(2\pi sin(y/2)) \tag{A-11}$$

The relation with the Fourier series is that the convolution of $D_N(x)$ with any function $f(x)$ of period 2π is the N-th degree Fourier approximation of $f(x)$:

$$(D_N * f)(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(y) D_N(x - y)\, dy = \sum_{k=-N}^{N} \hat{f} e^{ikx} = S_N(f)(x) \tag{A-12}$$

Figure A-3 shows the first 6 Dirichlet kernels.

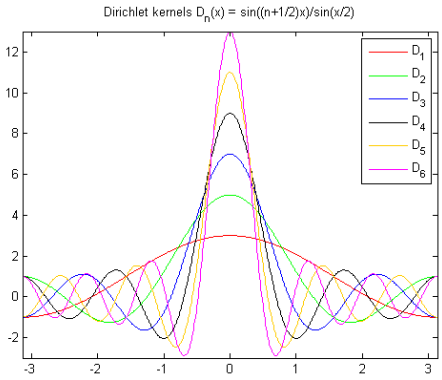

Figure A-3: The first few dirichlet Kernels

List of Figures

List of Tables

Bibliography

[1] Boyd, J., *Chebyshev and Fourier spectral methods*, Dover Publications, New York, 2001.

[2] Chan, T., Osher, S. & Shen, J., (2001), *The digital TV filter and nonlinear denoising*, Image Processing, IEEE Transactions on, 10 (2), 231–241.

[3] Driscoll, T. & Fornberg, B., (2001), *A Padé-based algorithm for overcoming the Gibbs phenomenon*, Numerical Algorithms, 26 (1), 77–92.

[4] Eckhoff, K., (1998), *On a high order numerical method for functions with singularities*, Mathematics of Computation, 67 (223), 1063–1088.

[5] Fang, F., *The COS method: An Efficient Fourier Method for Pricing Financial Derivatives*, Delft University of Technology, 2010.

[6] Fang, F., *Use COS method to calculate portfolio credit loss*, tech. report, Working paper, 2012.

[7] Fang, F. & Oosterlee, C., (2008), *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM Journal on Scientific Computing, 31 (2), 826–848.

[8] Gallagher, T. A., Nemeth, A. J. & Hacein-Bey, L., (2008), *An Introduction to the Fourier Transform: Relationship to MRI*, American Journal of Roentgenology, 190 (5), 1396–1405.

[9] Gelb, A., (2000), *A hybrid approach to spectral reconstruction of piecewise smooth functions*, Journal of Scientific Computing, 15 (3), 293–322.

[10] Gelb, A. & Cates, D., (2007), *Detection of Edges in Spectral Data III. Refinement of the Concentration Method*, Journal of Scientific Computing, 36 (1), 1–43.

[11] Gelb, A. & Tadmor, E., (2001), *Detection of Edges in Spectral Data II. Nonlinear Enhancement*, SIAM Journal on Numerical Analysis, 38 (4).

[12] Gelb, A., Tadmor, E., et al., (1999), *Detection of edges in spectral data*, Applied and computational harmonic analysis, 7 (1), 101.

[13] Gelb, A. & Tanner, J., (2006), *Robust reprojection methods for the resolution of the Gibbs phenomenon*, Applied and Computational Harmonic Analysis, 20 (1), 3–25.

[14] Gottlieb, D. & Shu, C.-W., (1995), *On the Gibbs Phenomenon IV: Recovering Exponential Accuracy in a Subinterval from a Gegenbauer Partial Sum of a Piecewise Analytic Function*, Mathematics of Computation, 64 (211), 1081–1095.

[15] Gottlieb, D. & Shu, C.-W., (1996), *On the Gibbs Phenomenon III: Recovering Exponential Accuracy in a Sub- Interval from a Spectral Partial Sum of a Piecewise Analytic Function*, SIAM Journal on Numerical Analysis, 33 (1), 280–290.

[16] Gottlieb, D. & Shu, C.-W., (1997), *On the Gibbs Phenomenon and its Resolution*, SIAM Review, 39 (4), 644–668.

[17] Gottlieb, D., Shu, C.-W., Solomonoff, A. & Vandeven, H., (1992), *On the Gibbs phenomenon I: recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function*, Journal of Computational and Applied Mathematics, 43 (1-2), 81–98.

[18] Greene, N., (2008), *Inverse Wavelet Reconstruction for Resolving the Gibbs Phenomenon*, International Journal of Circuits, Systems and Signal Processing, 2 (2), 73–77.

[19] Grzelak, L. & Oosterlee, C., (2010), *On the Heston Model with Stochastic Interest Rates*, SSRN eLibrary, 2 (1), 255–286.

[20] Heston, S., (1993), *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Review of financial studies, 6 (2), 327–343.

[21] Hrycak, T. & Gröchenig, K., (2010), *Pseudospectral Fourier reconstruction with the modified Inverse Polynomial Reconstruction Method*, Journal of Computational Physics, 229 (3), 933–946.

[22] Jung, J. & Shizgal, B., (2004), *Generalization of the inverse polynomial reconstruction method in the resolution of the Gibbs phenomenon*, Journal of computational and applied mathematics, 172 (1), 131–151.

[23] Jung, J. & Shizgal, B., (2007), *Short Note: On the numerical convergence with the inverse polynomial reconstruction method for the resolution of the Gibbs phenomenon*, Journal of Computational Physics, 224 (2), 477–488.

[24] Jung, J.-H. & Shizgal, B. D., (2005), *Inverse Polynomial Reconstruction of Two Dimensional Fourier Images*, Journal of Scientific Computing, 25 (3), 367–399.

[25] Madan, D., Carr, P. & Chang, E., (1998), *The Variance Gamma Process and Option Pricing*, European Finance Review, 2), 79–105.

[26] Min, M., Kaber, S. & Don, W., (2007), *Fourier-Padé approximations and filtering for spectral simulations of an incompressible Boussinesq convection problem*, Mathematics of computation, 76 (259), 1275.

[27] Pasquetti, R., (2004), *On inverse methods for the resolution of the Gibbs phenomenon*, Journal of computational and applied mathematics, 170 (2), 303–315.

[28] Ruijter, M., Oosterlee, C. & Aalbers, R., *On the Fourier cosine series expansion (COS) method for stochastic control problems*, tech. report, Working paper, 2011.

[29] Sarra, S., (2006), *Digital total variation filtering as postprocessing for Chebyshev pseudospectral methods for conservation laws*, Numerical Algorithms, 41 (1), 17–33.

[30] Schoutens, W., *Lévy processes in finance: pricing financial derivatives*, vol. 534, Wiley, 2003.

[31] Seydel, R., *Tools for computational finance*, Springer Verlag, 2009.

[32] Shizgal, B. & Jung, J., (2003), *Towards the resolution of the Gibbs phenomena*, Journal of Computational and Applied Mathematics, 161 (1), 41–65.

[33] Stein, E. & Shakarchi, R., *Fourier analysis: an introduction*, vol. 1, Princeton University Press, 2003.

[34] Tadmor, E., (2007), *Filters, Mollifiers and the Computation of the Gibbs Phenomenon*, Acta Numerica, 16), 305–378.

[35] Tadmor, E. & Tanner, J., (2005), *Adaptive filters for piecewise smooth spectral data**, IMA journal of numerical analysis, 25 (4), 635–647.

[36] Zhang, B., *Efficient Pricing of Early-Exercise and Exotic Options Based on Fourier Cosine Expansions*, PhD thesis, TUDelft, Delft, 2012.